МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

X.509

(08/2005)

СЕРИЯ X: СЕТИ ПЕРЕДАЧИ ДАННЫХ,
ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И
БЕЗОПАСНОСТЬ

Справочник

Информационные технологии – Взаимосвязь открытых систем – Справочник: Структуры сертификатов открытых ключей и атрибутов

Рекомендация МСЭ-Т X.509

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ X

СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ

СЕТИ ПЕРЕДАЧИ ДАННЫХ ОБЩЕГО ПОЛЬЗОВАНИЯ	
Службы и услуги	X.1–X.19
Интерфейсы	X.20–X.49
Передача, сигнализация и коммутация	X.50–X.89
Сетевые аспекты	X.90–X.149
Техническое обслуживание	X.150–X.179
Административные предписания	X.180–X.199
ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ	
Модель и обозначение	X.200–X.209
Определения служб	X.210–X.219
Спецификации протоколов с установлением соединений	X.220–X.229
Спецификации протоколов без установления соединений	X.230–X.239
Проформы PICS	X.240–X.259
Идентификация протоколов	X.260–X.269
Протоколы обеспечения безопасности	X.270–X.279
Управляемые объекты уровня	X.280–X.289
Испытание на соответствие	X.290–X.299
ВЗАИМОДЕЙСТВИЕ МЕЖДУ СЕТЯМИ	
Общие положения	X.300–X.349
Спутниковые системы передачи данных	X.350–X.369
Сети, основанные на протоколе Интернет	X.370–X.379
СИСТЕМЫ ОБРАБОТКИ СООБЩЕНИЙ	X.400–X.499
СПРАВОЧНИК	**X.500–X.599**
ОРГАНИЗАЦИЯ СЕТИ ВОС И СИСТЕМНЫЕ АСПЕКТЫ	
Организация сети	X.600–X.629
Эффективность	X.630–X.639
Качество обслуживания	X.640–X.649
Наименование, адресация и регистрация	X.650–X.679
Абстрактно-синтаксическая нотация 1 (ASN.1)	X.680–X.699
УПРАВЛЕНИЕ В ВОС	
Структура и архитектура управления системами	X.700–X.709
Служба и протокол связи для общего управления	X.710–X.719
Структура управляющей информации	X.720–X.729
Функции общего управления и функции ODMA	X.730–X.799
БЕЗОПАСНОСТЬ	X.800–X.849
ПРИЛОЖЕНИЯ ВОС	
Фиксация, параллельность и восстановление	X.850–X.859
Обработка транзакций	X.860–X.879
Удаленные операции	X.880–X.889
Общие приложения ASN.1	X.890–X.899
ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА	X.900–X.999
БЕЗОПАСНОСТЬ ЭЛЕКТРОСВЯЗИ	X.1000–

Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.

МЕЖДУНАРОДНЫЙ СТАНДАРТ ИСО/МЭК 9594-8
РЕКОМЕНДАЦИЯ МСЭ-Т X.509

Информационные технологии – Взаимосвязь открытых систем – Справочник: Структуры сертификатов открытых ключей и атрибутов

Резюме

В настоящей Рекомендации | Международном стандарте представлена структура для сертификатов открытых ключей и сертификатов атрибутов. Данные структуры могут использоваться другими телами стандартов для профилирования их применения к инфраструктурам открытых ключей (PKI) и инфраструктурам управления привилегиями (PMI). Также, в настоящей Рекомендации | Международном стандарте определена структура для предоставления Справочником услуг аутентификации его пользователям. Описываются два уровня аутентификации: простая аутентификация, с использованием пароля в качестве подтверждения заявленной идентификационной информации, и строгая аутентификация, включающая удостоверения, созданные с использованием криптографических методов. Поскольку простая аутентификация предлагает несколько ограниченную защиту от несанкционированного доступа, в качестве основы для предоставления услуг безопасности должна использоваться только строгая аутентификация.

Источник

Рекомендация МСЭ-Т X.521 утверждена 29 августа 2005 года 17-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т A.8. Идентичный текст также опубликован как стандарт ИСО/МЭК 9594-8.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

На Всемирной ассамблее по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяются темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соблюдение положений данной Рекомендации носит добровольный характер. Однако в Рекомендации могут содержаться определенные обязательные положения (например, для обеспечения возможности взаимодействия или применимости), и соблюдение положений данной Рекомендации достигается в случае выполнения всех этих обязательных положений. Для выражения необходимости выполнения требований используется синтаксис долженствования и соответствующие слова (такие, как "должен" и т. п.), а также их отрицательные эквиваленты. Использование этих слов не предполагает, что соблюдение положений данной Рекомендации является обязательным для какой-либо из сторон.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или реализация этой Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, обоснованности или применимости заявленных прав интеллектуальной собственности, независимо от того, отстаиваются ли они членами МСЭ или другими сторонами вне процесса подготовки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: http://www.itu.int/ITU-T/ipr/.

СОДЕРЖАНИЕ

Стр.

Стр.

Введение

Настоящая Рекомендация | Международный стандарт вместе с другими Рекомендациями | Международными стандартами разработана для упрощения взаимосвязи систем обработки информации в целях обеспечения справочных служб. Совокупность таких систем вместе с содержащейся в них справочной информацией можно рассматривать как единое целое, называемое *Справочником*. Содержащаяся в Справочнике информация, называемая в совокупности информационной базой Справочника (DIB), обычно используется для содействия обеспечению связи между объектами, с объектами или относительно объектов, примерами которых могут служить объекты прикладного уровня, люди, оконечные устройства и списки рассылки.

Справочник играет существенную роль во взаимосвязи открытых систем, назначение которой заключается, при минимальных технических соглашениях, не входящих в сами стандарты взаимосвязи, в обеспечении взаимосвязи систем обработки информации:

- поставляемых разными производителями;
- находящихся под различным управлением;
- различных уровней сложности; и
- разных поколений.

Многие приложения предъявляют требования к безопасности для защиты от угроз доступа к информации. Фактически, все услуги безопасности зависят от того, что идентификационная информация взаимодействующих сторон является достоверно известной, т. е. аутентификации.

В данной Рекомендации | Международном стандарте определена структура для сертификатов открытых ключей. В данную структуру включена спецификация объектов данных, используемых для представления как самих сертификатов, так и уведомлений об аннулировании выданных сертификатов, которые более не должны признаваться истинными. Структура сертификатов открытых ключей определена в данной Спецификации, хотя в ней определены некоторые важные компоненты инфраструктуры открытых ключей (PKI), в ней не определяется PKI в своей полноте. Тем не менее, в данной Спецификации предоставляются основы, на которых будут построены полные PKI и их спецификации.

Аналогично, в данной Рекомендации | Международном стандарте определена структура для сертификатов атрибутов. В данную структуру включена спецификация объектов данных, используемых для представления как самих сертификатов, так и уведомлений об аннулировании выданных сертификатов, которые более не должны признаваться истинными. Структура сертификатов атрибутов определена в данной Спецификации, хотя в ней определены некоторые важные компоненты инфраструктуры управления привилегиями (PMI), в ней не определяется PMI в своей полноте. Тем не менее, в данной Спецификации предоставляются основы, на которых будут построены полные PMI и их спецификации.

Также определены информационные объекты для хранения объектов PKI и PMI в Справочнике, а также для сравнения представленных значений с хранящимися значениями.

В данной Рекомендации | Международном стандарте также определена структура для предоставления Справочником услуг аутентификации его пользователям.

В данной Рекомендации | Международном стандарте предоставлены фундаментальные основы для определения отраслевых профилей другими группами по разработке стандартов и отраслевыми форумами. Многие из функций, определенных в этих основах как необязательные, могут определяться как обязательные к использованию в определенных средах с помощью профилей. Настоящее пятое издание является результатом пересмотра и усовершенствования в техническом аспекте четвертого издания данной Рекомендации | Международного стандарта, но не заменяет его. Реализации могут по-прежнему соответствовать четвертому изданию. Тем не менее, вместе с тем, с некоторого момента четвертое издание поддерживаться не будет (то есть более не будут устраняться сообщаемые дефекты). Рекомендуется обеспечить в возможно краткие сроки соответствие реализаций настоящему пятому изданию.

В данном пятом издании определяются 1, 2 и 3 версии сертификатов открытых ключей и 1 и 2 версии списков аннулированных сертификатов. В данном издании также определяется 2 версия сертификатов атрибутов.

В более раннюю редакцию с 3 версией сертификата открытого ключа и со 2 версией списка аннулированных сертификатов была добавлена функция расширяемости и была включена в сертификат атрибутов с самого его начала. Данная функция определена в пункте 7. Ожидается, что с использованием данной функции можно приспособиться к любым улучшениям данного издания, а также избежать необходимости в создании новых версий.

В Приложении A, которое является неотъемлемой частью настоящей Рекомендации | Международного стандарта, представлены модули ASN.1, содержащие все определения, связанные с данной структурой.

В Приложении B, которое является неотъемлемой частью настоящей Рекомендации | Международного стандарта, представлены правила создания и обработки списков аннулированных сертификатов.

В Приложении C, которое не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, представлены примеры выдачи дельта-CRL.

В Приложении D, которое не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, представлены примеры синтаксисов политики привилегий и атрибутов привилегий.

Приложение E, которое не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, является введением в криптографию с открытым ключом.

В Приложении F, которое является неотъемлемой частью настоящей Рекомендации | Международного стандарта, определены идентификаторы объектов, присвоенные алгоритмам аутентификации и шифрования при отсутствии формального реестра.

В Приложении G, которое не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, содержатся примеры использования ограничений тракта сертификации.

В Приложении H, которое не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, представлено руководство для приложений, разрешенных PKI, по обработке политики сертификатов в процессе проверки подлинности тракта сертификата.

В Приложении I, которое не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, представлено руководство по применению бита contentCommitment в расширении сертификата keyUsage.

В Приложении J, которое не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, содержится алфавитный указатель определений единиц информации в настоящей Спецификации.

В Приложении K, которое не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, перечислены поправки и сообщения о дефектах, которые были включены при составлении данного издания Рекомендации | Международного стандарта.

МЕЖДУНАРОДНЫЙ СТАНДАРТ
РЕКОМЕНДАЦИЯ МСЭ-Т

Информационные технологии – Взаимосвязь открытых систем – Справочник: Структуры сертификатов открытых ключей и атрибутов

РАЗДЕЛ 1 – ОБЩИЕ ПОЛОЖЕНИЯ

1 Сфера применения

В данной Рекомендации | Международном стандарте рассматриваются некоторые требования безопасности в сферах аутентификации и других услуг безопасности путем предоставления совокупности структур, на которых может основываться полный комплекс услуг. В частности, в данной Рекомендации | Международном стандарте определяются структуры для:

- сертификатов открытых ключей;
- сертификатов атрибутов;
- услуг аутентификации.

Структура сертификатов открытых ключей, определенная в данной Рекомендации | Международном стандарте, включает в себя определение информационных объектов для инфраструктуры открытых ключей (PKI), включая сертификаты открытых ключей и списки аннулированных сертификатов (CRL). Структура сертификатов атрибутов включает в себя определение информационных объектов для инфраструктуры управления привилегиями (PMI), включая сертификаты атрибутов и списки аннулированных сертификатов атрибутов (ACRL). В данной Спецификации также представлена структуры для издания, управления, использования и аннулирования сертификатов. Механизм расширяемости включен в определенные форматы как для типов сертификатов, так и для всех схем списков аннулирования. В данную Рекомендацию | Международный стандарт также включен набор стандартных расширений для каждого из них, который ожидается быть в целом полезен для некоторого количества применений PKI и PMI. Компоненты схемы, включая классы объектов, типа атрибутов и правила согласования для хранения объектов PKI и PMI в Справочнике, включены в данную Рекомендацию | Международный стандарт. Ожидается, что прочие элементы PKI и PMI, находящиеся вне этих структур, такие как протоколы управления ключами и сертификатами, рабочие протоколы, дополнительные расширения сертификатов и CRL, будут определены другими телами стандартов (например, ИСО ТС 68, IETF и т. д.).

Схема аутентификации, определенная в данной Рекомендации | Международном стандарте, является общей и может применяться к множеству применений и сред.

Справочник составляет использование сертификатов открытых ключей и сертификатов атрибутов, и структура для использования Справочником этих средств также определена в данной Рекомендации | Международном стандарте. Технология открытого ключа, включая сертификаты, используется Справочником для возможности строгой аутентификации, подписанных и/или зашифрованных операций, а также для хранения подписанных и/или зашифрованных данных в Справочнике. Сертификаты атрибутов могут использоваться Справочником для возможности управления доступом на основе правил. Несмотря на то, что структура для этого предоставляется в данной Спецификации, полное определение применения Справочником данных структур, а также связанных услуг, предоставляемых Справочником и его компонентами, приводится во всей совокупности спецификаций Справочника.

В данной Рекомендации | Международном стандарте, в структуре услуг аутентификации, также:

- определяется форма информации для аутентификации, содержащейся в Справочнике;
- описывается, как информация для аутентификации может быть получена из Справочника;
- формулируются предположения о том, как формируется и помещается в Справочник информация для аутентификации;
- определяются три способа, которыми данная информация может использоваться приложениями для аутентификации, и описывается использование аутентификации для обеспечения прочих услуг безопасности.

В данной Рекомендации | Международном стандарте описываются два уровня аутентификации: простая аутентификация, использующая пароль для подтверждения заявленной идентификационной информации, и строгая аутентификация, включающая удостоверения, созданные с использованием криптографических методов. Поскольку простая аутентификация предлагает несколько ограниченную защиту от несанкционированного доступа, в качестве основы для предоставления услуг безопасности должна использоваться только строгая аутентификация. Не предполагается устанавливать это в качестве

общей структуры для аутентификации, но этот подход может рассматриваться как общий для применений, считающих данные методы соответствующим их требованиям.

Аутентификация и другие услуги безопасности могут предоставляться только в контексте определенной политики безопасности. Пользователи могут определять свою собственную политику безопасности, которая может ограничиваться услугами, предоставляемыми стандартом.

В задачу определяющих стандарты приложений, которые используют структуру аутентификации, входит определение обменов протоколами, которые необходимо выполнять для выполнения аутентификации, основанной на информации для аутентификации, получаемой из Справочника. Протоколом, используемым приложениями для получения из Справочника удостоверений, является протокол доступа к Справочнику (DAP), определенный в Рекомендации МСЭ-Т X.519 | ИСО/МЭК 9594-5.

2 Нормативные справочные документы

В перечисленных ниже Рекомендациях МСЭ-Т и Международных стандартах содержатся положения, которые посредством ссылок на них в этом тексте составляют основные положения данной Рекомендации | Международного стандарта. На момент публикации, действовали указанные редакции документов. Все Рекомендации и Стандарты являются предметом корректировки, в связи с чем сторонам соглашений, основанных на данной Рекомендации | Международном стандарте, предлагается изучить возможность применения последнего издания Рекомендаций и Стандартов, перечисленных ниже. Члены МЭК и ИСО ведут регистры действующих в настоящее время Международных стандартов. Бюро стандартизации электросвязи МСЭ ведет список действующих в настоящее время Рекомендаций МСЭ-Т.

2.1 Идентичные Рекомендации | Международные стандарты

- ITU-T Recommendation X.411 (1999) | ISO/IEC 10021-4:2003, *Information technology – Message Handling Systems (MHS) – Message transfer system: Abstract service definition and procedures*.
- Рекомендация МСЭ-Т X.500 (2005 г.) | ИСО/МЭК 9594-1:2005, *Информационные технологии – Справочник: Обзор понятий, моделей и услуг*.
- Рекомендация МСЭ-Т X.501 (2005 г.) | ИСО/МЭК 9594-2:2005, *Информационные технологии – Взаимосвязь открытых систем – Справочник: Модели*.
- Рекомендация МСЭ-Т X.511 (2005 г.) | ИСО/МЭК 9594-3:2005, *Информационные технологии – Взаимосвязь открытых систем – Справочник: Определение абстрактной службы.*
- Рекомендация МСЭ-Т X.518 (2005 г.) | ИСО/МЭК 9594-4:2005, *Информационные технологии – Взаимосвязь открытых систем – Справочник: Процедуры распределенных операций.*
- ITU-T Recommendation X.519 (2005) | ISO/IEC 9594-5:2005, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications*.
- Рекомендация МСЭ-Т X.520 (2005 г.) | ИСО/МЭК 9594-6:2005, *Информационные технологии – Взаимосвязь открытых систем – Справочник: Избранные типы атрибутов.*
- Рекомендация МСЭ-Т X.521 (2005 г.) | ИСО/МЭК 9594-7:2005, *Информационные технологии – Взаимосвязь открытых систем – Справочник: Избранные объектные классы.*
- Рекомендация МСЭ-Т X.525 (2005 г.) | ИСО/МЭК 9594-9:2005, *Информационные технологии – Взаимосвязь открытых систем – Справочник: Копирование.*
- ITU-T Recommendation X.530 (2005) | ISO/IEC 9594-10:2005, *Information technology – Open Systems Interconnection – The Directory: Use of systems management for administration of the Directory*.
- ITU-T Recommendation X.660 (2004) | ISO/IEC 9834-1:2005, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures, and top arcs of the ASN.1 Object Identifier tree*.
- ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.681 (2002) | ISO/IEC 8824-2:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.
- ITU-T Recommendation X.682 (2002) | ISO/IEC 8824-3:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.
- ITU-T Recommendation X.683 (2002) | ISO/IEC 8824-4:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications*.
- ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.
- ITU-T Recommendation X.691 (2002) | ISO/IEC 8825-2:2002, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

- ITU-T Recommendation X.812 (1995) | ISO/IEC 10181-3:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework.*
- ITU-T Recommendation X.813 (1996) | ISO/IEC 10181-4:1997, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Non-repudiation framework.*
- ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations: Concepts, model and notation.*
- ITU-T Recommendation X.881 (1994) | ISO/IEC 13712-2:1995, *Information technology – Remote Operations: OSI realizations – Remote Operations Service Element (ROSE) service definition.*

2.2 Парные Рекомендации | Международные стандарты, эквивалентные по техническому содержанию

- CCITT Recommendation X.800 (1991), *Security Architecture for Open Systems Interconnection for CCITT applications*.
- ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*.

3 Определения

Для целей настоящей Рекомендации | Международного стандарта используются следующие определения.

3.1 Определения архитектуры безопасности эталонной модели OSI

Следующие термины определены в Рекомендации CCITT X.800 | ИСО 7498-2:

a) асимметричное (шифрование);
b) обмен аутентификацией;
c) информация для аутентификации;
d) секретность;
e) удостоверения;
f) криптография;
g) аутентификация источника данных;
h) расшифровка;
i) цифровая подпись;
j) шифрование;
k) ключ;
l) пароль;
m) аутентификация равноправных-логических-объектов;
n) симметричное (шифрование).

3.2 Определения модели Справочника

Следующие термины определены в Рекомендации МСЭ-Т X.501 | ИСО/МЭК 9594-2:

a) атрибут;
b) информационная база Справочника;
c) информационное дерево Справочника;
d) системный агент Справочника;
e) агент пользователя Справочника;
f) выделенное время;
g) запись;
h) объект;
i) корень.

3.3 Определения

В данной Рекомендации | Международном стандарте определены следующие термины:

3.3.1 сертификат атрибута (AC): Структура данных с цифровой подписью органа атрибутов, связывающая определенные значения атрибута с идентификационной информацией о его держателе.

3.3.2 орган атрибутов (AA): Орган, присваивающий привилегии путем выдачи сертификатов атрибутов.

3.3.3 список аннулированных органов атрибутов (AARL): Список аннулирования, содержащий перечень ссылок на сертификаты атрибутов, выданных AA, которые более не считаются действительными выдавшим органом.

3.3.4 список аннулированных сертификатов атрибутов (ACRL): Список аннулирования, содержащий перечень ссылок на сертификаты атрибутов, которые более не считаются действительными выдавшим органом.

3.3.5 маркер аутентификации; (маркер): Информация, передаваемая в процессе обмена строгой аутентификацией, которая может быть использована для аутентификации отправителя.

3.3.6 орган: Объект, ответственный за выдачу сертификатов. В данной Спецификации определены два типа: орган по сертификации, выдающий сертификаты открытых ключей, и орган атрибутов, выдающий сертификаты атрибутов.

3.3.7 сертификат органа: Сертификат, выданный органу (например, или органу по сертификации, или органу атрибутов).

3.3.8 основной CRL: CRL, выданный в качестве основы в поколении dCRL.

3.3.9 сертификат CA: Сертификат для CA, выданный другим CA.

3.3.10 политика сертификатов: Поименованная совокупность правил, обозначающая применимость сертификата к конкретному сообществу/классу применения с общими требованиями к безопасности. Например, конкретная политика сертификатов может обозначать применимость типа сертификата к аутентификации операций электронного обмена данными для торговли товарами в пределах заданного ценового диапазона.

3.3.11 заявление практического применения сертификации (CPS): Заявление практического применения, которое орган по сертификации использует при выдаче сертификатов.

3.3.12 список аннулированных сертификатов (CRL): Подписанный список, отражающий совокупность сертификатов, которые более не признаются действительными выдавшим их органом. Помимо общего понятия CRL, определены некоторые конкретные типы CRL для особенных областей применения.

3.3.13 пользователь сертификата: Объект, которому необходимо достоверно знать атрибуты или открытый ключ другого объекта.

3.3.14 порядковый номер сертификата: Целое число, уникальное в пределах выдающего сертификаты органа, однозначно связанное с сертификатом, выданным данным органом.

3.3.15 система, использующая сертификат: Реализация функций, определенных в данной спецификации Справочника, используемых пользователем сертификата.

3.3.16 проверка сертификата: Процесс проверки того, что сертификат был действителен на заданный момент времени, возможно, включающий создание и обработку тракта сертификации, и проверку того, что все сертификаты в данном тракте были действительны (т. е. не были просрочены или аннулированы) на тот заданный момент времени.

3.3.17 орган по сертификации (CA): Орган, которому один или несколько пользователей доверили разработку и присвоение сертификатов открытых ключей. Возможен вариант, при котором орган по сертификации разрабатывает ключи для пользователей.

3.3.18 список аннулированных органов по сертификации (CARL): Список аннулирования, содержащий перечень ссылок на сертификаты открытого ключа, выданных органам по сертификации, которые более не считаются действительными выдавшим сертификаты органом.

3.3.19 тракт сертификации: Упорядоченная последовательность сертификатов открытых ключей в DIT, которые вместе с открытым ключом начального объекта в тракте могут быть обработаны для получения открытого ключа конечного объекта в тракте.

3.3.20 точка распределения CRL: Запись в справочнике или другой источник распределения для CRL; CRL, распределенные через точку распределения, могут содержать записи аннулирования только для одного подмножества полного множества сертификатов, выданных одним CA, или могут содержать объекты аннулирования, выданные несколькими CA.

3.3.21 перекрестный сертификат: Сертификат открытого ключа или атрибута, в котором выдавший орган и субъект/держатель являются различными CA или AA соответственно. CA и AA выдают перекрестные сертификаты другим CA или AA соответственно в качестве механизма авторизации существования CA субъекта (например, в строгой иерархии) или для распознавания существования CA субъекта или AA держателя (например, в распределенной модели доверия). Структура перекрестных сертификатов используется в обоих случаях.

3.3.22 криптографическая система, криптосистема: Совокупность преобразований открытого теста в криптограмму и наоборот, при этом использование конкретного(ых) преобразования(ий) выбирается при помощи ключей. Обычно преобразования определяются при помощи математического алгоритма.

3.3.23 конфиденциальность данных: Данная услуга может использоваться для обеспечения защиты данных от несанкционированного раскрытия. Услуга конфиденциальности данных поддерживается структурой аутентификации. Она может использоваться для защиты от перехвата данных.

3.3.24 делегирование: Передача привилегии от объекта, держащего данную привилегию, к другому объекту.

3.3.25 тракт делегирования: Упорядоченная последовательность сертификатов, которая вместе с идентификационной информацией заявителя привилегии может быть обработана для проверки подлинности заявителя привилегии.

3.3.26 дельта-CRL (dCRL): Частичный список аннулирования, содержащий только записи для сертификатов, статус аннулирования которых был изменен с момента выдачи соответствующего основного CRL.

3.3.27 оконечный объект: Или субъект сертификата открытого ключа, использующий свой частный ключ для целей, отличных от подписания сертификатов, или держатель сертификата атрибутов, использующий свои атрибуты для получения доступа к ресурсу, или объект, являющийся зависимой стороной.

3.3.28 список аннулированных сертификатов атрибутов оконечных объектов (EARL): Список аннулирования, содержащий список сертификатов атрибутов, выданных их держателям, не являющимся также AA, которые более не считаются действительными органом, выдавшим сертификат.

3.3.29 список аннулированных сертификатов открытых ключей оконечных объектов (EPRL): Список аннулирования, содержащий список сертификатов открытых ключей, выданных субъектам, не являющимся также CA, которые более не считаются действительными органом, выдавшим сертификат.

3.3.30 переменные среды: Те аспекты политики, требуемые для принятия решения об авторизации, которые не содержатся в статических структурах, но являются доступными верификатору привилегий при помощи некоторых локальных средств (например, время дня или текущий платежный баланс).

3.3.31 полный CRL: Полный список аннулирования, содержащий записи для всех сертификатов, которые были аннулированы для заданной области применения.

3.3.32 хэш-функция: (Математическая) функция, отображающая значения из большой (возможно, очень большой) области в меньший диапазон значений. Хэш-функция считается "хорошей", если результаты ее применения к (большому) множеству значений в области будут равномерно (и, очевидно, случайным образом) распределены по диапазону.

3.3.33 держатель: Объект, которому делегирована некоторая привилегия, либо непосредственно Источником органа, или косвенно через другой орган атрибутов.

3.3.34 непрямой CRL (iCRL): Список аннулирования, который содержит по меньшей мере информацию аннулирования о сертификатах, выданных органами, отличными от выдавшего данный CRL.

3.3.35 соглашение о ключах: Метод интерактивного согласования значения ключа без передачи ключа даже в шифрованной форме, например, метод Диффи-Хэллмана (дополнительную информацию о механизмах соглашения о ключах см. также в ИСО/МЭК 11770-1).

3.3.36 метод объекта: Действие, которое может быть вызвано на ресурсе (например, файловая система может иметь методы объектов чтения, записи и выполнения).

3.3.37 однонаправленная функция: (Математическая) функция f, которую легко вычислить, но для которой трудно найти такое значение x из области определения функции, для которого $f(x) = y$, где y – любое значение из области значений. Может существовать лишь небольшое число значений y, нахождение x для которых не представляет трудности в вычислении.

3.3.38 отображение политики: Признается, что когда CA в одной области сертифицирует CA в другой области, конкретная политика сертификатов во второй области может рассматриваться органом первой области как эквивалентная (но необязательно идентичная во всех отношениях) конкретной политике сертификатов в первой области.

3.3.39 частный ключ; секретный ключ (исключено): (В криптосистеме с открытыми ключами) тот ключ пары ключей пользователя, который известен только данному пользователю.

3.3.40 привилегия: Атрибут или свойство, присвоенное органом объекту.

3.3.41 заявитель привилегии: Держатель привилегий, использующий их сертификат атрибутов или сертификат открытого ключа для заявления привилегии.

3.3.42 инфраструктура управления привилегиями (PMI): Инфраструктура, способная поддерживать управление привилегиями в поддержке комплексной услуги авторизации и во взаимосвязи с инфраструктурой открытых ключей.

3.3.43 политика привилегий: Политика, описывающая условия для верификаторов привилегий при предоставлении/выполнении чувствительных услуг квалифицированным заявителям привилегий. Политика привилегий касается атрибутов, связанных с услугами, а также атрибутов, связанных с заявителями.

3.3.44 верификатор привилегий: Объект, проверяющий сертификаты согласно политике привилегий.

3.3.45 открытый ключ: (В криптосистеме с открытыми ключами) тот ключ пары ключей пользователя, который общеизвестен.

3.3.46 сертификат открытого ключа (PKC): Открытый ключ пользователя в совокупности с некоторой дополнительной информацией, воспроизводимой без возможности фальсификации при помощи цифровой подписи с открытым ключом выдавшего его органа по сертификации.

3.3.47 инфраструктура открытых ключей (PKI): Инфраструктура, способная поддерживать управление открытыми ключами для поддержки услуг аутентификации, шифрования, целостности или фиксации авторства.

3.3.48 зависимая сторона: Пользователь или агент, зависящий от данных сертификата при принятии решений.

3.3.49 сертификат присвоения роли: Сертификат, содержащий атрибут роли, присваивающий одну или несколько ролей субъекту/держателю сертификата.

3.3.50 сертификат спецификации роли: Сертификат, содержащий присвоение привилегий роли.

3.3.51 чувствительность: Характеристика ресурса, подразумевающая его значение или важность.

3.3.52 простая аутентификация: Аутентификация при помощи простых соглашений о паролях.

3.3.53 политика безопасности: Совокупность правил, установленных органом безопасности, управляющим использованием и предоставлением услуг и средств безопасности.

3.3.54 автоматически выданный АС: Сертификат атрибута, когда выдавший орган и субъект являются одним и тем же органом атрибутов. Орган атрибутов может использовать автоматически выданный АС, например, для опубликования информации о политике.

3.3.55 автоматически выданный сертификат: Сертификат открытого ключа, когда выдавший орган и субъект являются одним и тем же СА. СА может использовать автоматически выданные сертификаты, например, в течение операции перебора ключа для обеспечения доверия от старого ключа новому ключу.

3.3.56 автоматически подписанный сертификат: Особый случай автоматически выданных сертификатов, когда частный ключ, используемый СА для подписания сертификата, соответствует открытому ключу, сертифицированному в сертификате. СА может использовать автоматически подписанный сертификат, например, для рекламы своих открытых ключей или другой информации о своей деятельности.

ПРИМЕЧАНИЕ. – Использование автоматически выданных сертификатов и автоматически подписанных сертификатов, выданных не СА, выходит за рамки области применения данной Рекомендации | Международного стандарта.

3.3.57 источник органа (SOA): Орган атрибутов, которому верификатор привилегий для конкретного ресурса доверяет как последней инстанции при присвоении набора привилегий.

3.3.58 строгая аутентификация: Аутентификация при помощи криптографически полученных удостоверений.

3.3.59 доверие: В общем случае можно сказать, что один объект "доверяет" другому объекту, если первый объект предполагает, что поведение второго объекта будет в точности соответствовать ожиданиям первого объекта. Данное доверие может применяться только к некоторой конкретной функции. Ключевая роль понятия "доверие" в этой структуре заключается в описании взаимосвязи между объектом аутентификации и органом, объект должен быть уверен, что может доверять органу и что орган создает только действительные и надежные сертификаты.

3.3.60 опора доверия: Опора доверия представляет собой совокупность следующей информации в дополнение к открытому ключу: идентификатор алгоритма, параметры открытого ключа (если применимо), выделенное имя держателя соответствующего открытого ключа (например, субъект СА) и (необязательно) период действия. Опора доверия может быть представлена в форме автоматически подписанного сертификата. Система, использующая сертификат, доверяет опоре доверия, которая применяется для проверки сертификатов в тракте сертификации.

4 Сокращения

Для целей настоящей Рекомендации | Международного стандарта используются следующие сокращения.

AA	Attribute Authority	Орган атрибутов
AARL	Attribute Authority Revocation List	Список аннулированных органов атрибутов
AC	Attribute Certificate	Сертификат атрибута
ACRL	Attribute Certificate Revocation List	Список аннулированных сертификатов атрибутов
CA	Certification Authority	Орган по сертификации
CARL	Certification Authority Revocation List	Список аннулированных органов по сертификации
CRL	Certificate Revocation List	Список аннулированных сертификатов

dCRL	Delta Certificate Revocation List	Список аннулированных дельта-сертификатов
DIB	Directory Information Base	Информационная база Справочника
DIT	Directory Information Tree	Информационное дерево Справочника
DSA	Directory System Agent	Системный агент Справочника
DUA	Directory User Agent	Агент пользователя Справочника
EARL	End-entity Attribute certificate Revocation List	Список аннулированных сертификатов атрибутов оконечных объектов
EPRL	End-entity Public-key certificate Revocation List	Список аннулированных сертификатов открытых ключей оконечных объектов
iCRL	Indirect Certificate Revocation List	Непрямой список аннулированных сертификатов
OCSP	Online Certificate Status Protocol	Интерактивный протокол состояния сертификата
PKC	Public-Key Certificate	Сертификат открытого ключа
PKCS	Public-Key Cryptosystem	Криптосистема открытых ключей
PKI	Public-Key Infrastructure	Инфраструктура открытых ключей
PMI	Privilege Management Infrastructure	Инфраструктура управления привилегиями
SOA	Source of Authority	Источник органа

5 Соглашения

За небольшими исключениями, эта спецификация Справочника была подготовлена в соответствии с "*Правилами представления общего текста МСЭ-Т | ИСО/МЭК*", ноябрь 2001 г.

Термин "спецификация Справочника" (как и "эта спецификация Справочника") означает Рекомендацию МСЭ-Т X.509 | ИСО/МЭК 9594-8. Термин "спецификация Справочника" должен означать Рекомендации серии X.500 и все части стандарта ИСО/МЭК 9594.

В данной спецификации Справочника используется термин *системы первого издания* для указания на системы, соответствующие первому изданию спецификаций Справочника, т. е. изданию 1988 года Рекомендаций МККТТ серии X.500 и изданию стандарта ИСО/МЭК 9594:1990. В этой спецификации Справочника используется термин *системы второго издания* для указания на системы, соответствующие второму изданию спецификаций Справочника, т. е. изданию 1993 года Рекомендаций МСЭ-Т серии X.500 и изданию стандарта ИСО/МЭК 9594:1995. В этой спецификации Справочника используется термин *системы третьего издания* для указания на системы, соответствующие третьему изданию спецификаций Справочника, т. е. изданию 1997 года Рекомендаций МСЭ-Т серии X.500 и изданию стандарта ИСО/МЭК 9594:1998. В этой спецификации Справочника используется термин *системы четвертого издания* для указания на системы, соответствующие четвертому изданию спецификаций Справочника, т. е. изданиям 2001 года Рекомендаций МСЭ-Т X.500, X.501, X.511, X.518, X.519, X.520, X.521, X.525 и X.530, изданию 2000 года Рекомендаций МСЭ-Т X.509 и частям 1–10 издания стандарта ИСО/МЭК 9594:2001.

В настоящей спецификации Справочника используется термин *системы пятого издания* для ссылки на системы, соответствующие пятому изданию спецификаций Справочника, т. е. изданиям 2005 года Рекомендаций МСЭ-Т X.500, X.501, X.509, X.511, X.518, X.519, X.520, X.521, X.525 и X.530 и частей 1–10 издания стандарта ИСО/МЭК 9594:2005.

В данной спецификации Справочника нотация на языке ASN.1 дается полужирным шрифтом Helvetica. Когда типы и значения ASN.1 приводятся в обычном тексте, они выделяются полужирным шрифтом Helvetica. Названия процедур, упоминаемых при определении семантики обработки, выделяются в тексте полужирным шрифтом Times. Разрешения на управление доступом предоставляются курсивом шрифта Times.

Если элементы в списке пронумерованы (либо против них указаны "–" или буквы), то эти пункты должны рассматриваться как шаги в процедуре.

Обозначения, используемые в данной спецификации Справочника, определены в таблице 1, приведенной ниже.

Таблица 1 – Обозначения

Обозначение	Значение
Xp	Открытый ключ пользователя X.
Xs	Частный ключ пользователя X.
Xp[I]	Шифрование некоторой информации I с использованием открытого ключа пользователя X.
Xs[I]	Шифрование некоторой информации I с использованием частного ключа пользователя X.
X{I}	Подписание информации I пользователем X. Состоит из I с приложенным зашифрованным кратким содержанием.

CA(X)	Орган по сертификации пользователя X.
$CA^n(X)$	(где n>1): CA(CA(...n раз...(X)))
$X_1<<X_2>>$	Сертификат пользователя X_2, выданный органом по сертификации пользователя X_1.
$X_1<<X_2>>$ $X_2<<X_3>>$	Цепочка сертификатов (может быть произвольной длины), в которой каждый элемент является сертификатом для органа по сертификации, выдавшего следующий сертификат. Функционально данная цепочка эквивалентна следующему сертификату $X1<<X_{n+1}>>$. Например, владение цепочкой A<<B>>B<<C>> обеспечивает те же возможности, что и A<<C>>, а именно возможность выяснить Cp по заданному Ap.
X_1p ° $X_1<<X_2>>$	Операция по развертыванию сертификата (или цепочки сертификатов) для извлечения открытого ключа. Представляет собой инфиксный оператор, левым операндом которого является открытый ключ органа по сертификации, а правым операндом является сертификат, выданный данным органом по сертификации. Результатом операции является открытый ключ пользователя, сертификатом которого является правый операнд. Например: Ap ° A<<B>> B<<C>> означает операцию применения открытого ключа A для получения открытого ключа Bp пользователя B из его сертификата, сопровождаемую применением Bp для развертывания сертификата пользователя C. Результатом данной операции является открытый ключ Cp пользователя C.
A→B	Тракт сертификации от A к B, образованный цепочкой сертификатов, начинающейся с $CA(A)<<CA^2(A)>>$ и заканчивающейся CA(B)<<B>>.
ПРИМЕЧАНИЕ. – В данной таблице символы X, X_1, X_2 и т. д. расположены на местах для имен пользователей, тогда как символ I расположен на месте, указывающем на произвольную информацию.	

6 Обзор структуры

В данной Спецификации определена структура для получения и доверия в отношении открытого ключа объекта для шифрования информации, которая должна быть расшифрована данным объектом, или для проверки цифровой подписи данного объекта. В структуру включены выдача сертификата открытого ключа органом по сертификации (CA) и проверка подлинности этого сертификата выдавшим органом. Проверка подлинности включает в себя:

- установление доверяемого тракта сертификатов между пользователем сертификата и субъектом сертификата;
- проверку цифровой подписи каждого сертификата в тракте; и
- проверку подлинности всех сертификатов на протяжении данного тракта (т. е., что они не были просрочены или аннулированы на заданный момент времени).

В данной Спецификации определена структура для получения и доверия в отношении атрибутов привилегий объекта для определения того, являются ли они авторизованными для доступа к определенному ресурсу. В структуру включены выдача сертификата органом атрибутов (AA) и проверка подлинности этого сертификата верификатором привилегий. Проверка подлинности включает в себя:

- обеспечение того, что привилегии в сертификате являются достаточными при сопоставлении с политикой привилегий;
- установление доверенного тракта делегирования сертификатов при необходимости;
- проверку цифровой подписи на каждом сертификате в тракте;
- обеспечение того, что каждый выдавший орган был авторизован для делегирования привилегий; и
- проверку того, что сертификаты не были просрочены или аннулированы выдавшими их органами.

Несмотря на то, что PKI и PMI являются отдельными инфраструктурами и могут быть созданы независимо одна от другой, они связаны. А данной Спецификации рекомендуется, чтобы держатели и пользователи сертификатов атрибутов были определены в сертификатах атрибутов указателями на их соответствующие сертификаты открытых ключей. Аутентификация выдавших сертификаты атрибутов органов и их пользователей для подтверждения того, что объекты, заявляющие о привилегии и выдающие привилегию, являются теми, кем себя представляют, выполняется с использованием обычных процессов PKI для аутентификации идентификационной информации. Данный процесс аутентификации не дублируется в пределах структуры сертификатов атрибутов.

6.1 Цифровые подписи

Цифровые подписи используются как в PKI, так и в PMI в качестве механизма, при помощи которого орган, выдающий сертификат, сертифицирует связывание в сертификате. В PKI, цифровая подпись выдающего CA на сертификате открытого ключа сертифицирует связывание между материалом открытого ключа и субъектом сертификата. В PMI, цифровая подпись выдающего AA на сертификате открытого ключа сертифицирует связывание между атрибутами (привилегиями) и держателем сертификата. В данном подпункте описаны цифровые подписи в общем. Во 2 и 3 Разделах данной Спецификации более конкретно обсуждается использование цифровых подписей в рамках PKI и PMI соответственно.

Данный подпункт не предназначен в общем случае для определения стандарта для цифровых подписей, в нем определяются средства, при помощи которых подписываются маркеры в PKI, PMI и Справочнике.

Информация (Info) подписывается посредством присоединения к ней зашифрованного резюме данной информации. Резюме составляется при помощи однонаправленной хэш-функции, в то время как шифрование выполняется при использовании частного ключа подписавшей стороны (см. рисунок 1). Таким образом:

X{Info} = Info, Xs[h (Info)].

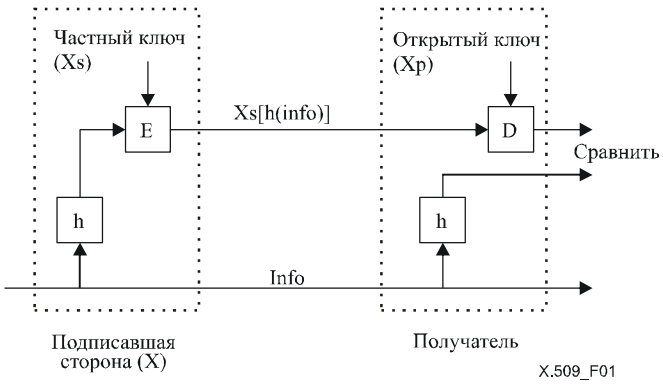


Рисунок 1 – Цифровые подписи

ПРИМЕЧАНИЕ 1. – Шифрование при использовании частного ключа обеспечивает невозможность фальсификации подписи. Однонаправленный характер хэш-функции обеспечивает невозможность подстановки ошибочной информации, которая сгенерирована для получения того же результата хэширования (и таким образом, подписи).

Получатель подписанной информации проверяет подпись путем:

- применения к информации однонаправленной хэш-функции;
- сравнения результата со значением, полученным при дешифровании подписи с использованием открытого ключа подписавшей стороны.

В данной Спецификации не утверждается обязательность применения одной определенной однонаправленной хэш-функции при подписании. Предполагается, что структура должна быть применимой к любой соответствующей хэш-функции, и должна таким образом поддерживать изменения в используемых методах, которые появятся в результате будущих достижений в криптографии, математических методах или вычислительных возможностях. Тем не менее, два пользователя, которые желают пройти аутентификацию, для правильного выполнения аутентификации должны поддерживать одну и ту же хэш-функцию. Таким образом, в контексте совокупности связанных приложений, выбор единой функции должен способствовать максимизации сообщества пользователей, которым доступна конфиденциальная аутентификация и связь.

Подписанная информация включает в себя индикаторы, которые определяют алгоритм хэширования и алгоритм шифрования, применяемый при вычислении цифровой подписи.

Шифрование некоторого элемента данных может быть описано с использованием следующей ASN.1:

```
ENCRYPTED { ToBeEnciphered } ::=   BIT STRING ( CONSTRAINED BY {
    -- должен быть результат применения процедуры шифрования --
    -- для октетов, кодированным с использованием BER, имеющих значения -- ToBeEnciphered } )
```

Значение строки битов генерируется путем выбора октетов, формирующих полное кодирование (с использованием Основных правила кодирования ASN.1 – Рек. МСЭ-Т X.690 (2002) | ИСО/МЭК 8825-1:2002) значения типа **ToBeEnciphered**, и применения процедуры шифрования к этим октетам.

ПРИМЕЧАНИЕ 2. – Процедура шифрования требует соглашения об алгоритме, который должен применяться, включая любые параметры алгоритма, такие как необходимые ключи, значения инициализации и правила заполнения. В обязанности процедур шифрования входит определение средств, при помощи которых достигается синхронизация между отправителем и получателем данных, которая может включать информацию в битах, которые должны быть переданы.

ПРИМЕЧАНИЕ 3. – Процедуре шифрования требуется принять в качестве входного параметра строку октетов и как результат сгенерировать единую строку битов.

ПРИМЕЧАНИЕ 4. – Механизмы для конфиденциального соглашения об алгоритме шифрования и его параметров между отправителем и получателем данных находятся вне области применения данной спецификации Справочника.

Подпись некоторого элемента данных формируется при помощи шифрования укороченной или "хэшированной" информации об элементе и может быть описано следующей ASN.1:

```
HASH {ToBeHashed}          ::=   SEQUENCE {
    algorithmIdentifier          AlgorithmIdentifier,
    hashValue                    BIT STRING ( CONSTRAINED BY {
        -- должен быть результат применения процедуры хэширования для октетов, кодированным с --
        -- использованием DER, имеющим значения -- ToBeHashed } ) }
```

```
ENCRYPTED-HASH { ToBeSigned }    ::=        BIT STRING ( CONSTRAINED BY {
    -- должен быть результат применения процедуры хэширования для октетов, кодированным с --
    -- использованием DER, имеющим значения -- ToBeSigned – и затем применения процедуры--
    -- шифрования к этим октетам -- })
SIGNATURE { ToBeSigned }        ::=        SEQUENCE {
    algorithmIdentifier         AlgorithmIdentifier,
    encrypted                   ENCRYPTED-HASH { ToBeSigned }}
```

ПРИМЕЧАНИЕ 5. – Процедура шифрования требует соглашений, указанных в Примечании 2, а также соглашения в отношении того, происходит ли шифрование хэшированных октетов непосредственно или только после дальнейшего кодирования их в виде **BIT STRING** с использованием Основных правила кодирования ASN.1.

В случае, когда подпись дополняется до типа данных, следующая ASN.1 может быть использована для определения типа данных, получаемого при применении подписи к заданному типу данных.

```
SIGNED { ToBeSigned }                    ::=        SEQUENCE {
    toBeSigned                  ToBeSigned,
    COMPONENTS OF               SIGNATURE { ToBeSigned }}
```

Для того чтобы сделать возможной проверку типов **SIGNED** и **SIGNATURE** в распределенной среде, требуется отличительное кодирование. Отличительное кодирование значения данных **SIGNED** и **SIGNATURE** должно быть получено при применении основных правил кодирования, определенных в Рек. МСЭ-Т X.690 (2002) | ИСО/МЭК 8825-1:2002, со следующими ограничениями:

a) должна использоваться определенная форма кодирования длины, кодированная при помощи минимального числа октетов;

b) для типов строки, не должна использоваться созданная форма кодирования;

c) если значение типа является его значением по умолчанию, оно должно отсутствовать;

d) все компоненты типа Set должны кодироваться в порядке возрастания значения их метки;

e) все компоненты типа Set должны кодироваться в порядке возрастания значения их октетов;

f) если значение данных булевого типа является истинным, то при кодировании значение октета содержимого должно быть установлено в "FF"16;

g) каждый неиспользованный бит в последнем октете кодирования значения строки битов, при наличии, должен быть установлен в значение 0;

h) кодирование данных действительного типа должно быть таким, что не должны использоваться основы 8, 10 и 16, и двоичный масштабирующий коэффициент должен быть равен 0;

i) кодирование времени UTC должно соответствовать Рек. МСЭ-Т X.690 (2002) | ИСО/МЭК 8825-1:2002;

j) кодирование обобщенного времени должно соответствовать Рек. МСЭ-Т X.690 (2002) | ИСО/МЭК 8825-1:2002.

Генерирование отличительного кодирования требует, чтобы абстрактный синтаксис данных был кодирован с возможностью полного понимания. Может существовать необходимость, чтобы Справочник подписал данные или проверил подпись данных, содержащих неизвестные расширения протокола или неизвестные синтаксисы атрибутов. Справочник должен следовать следующим правилам:

– он должен зафиксировать кодирование полученной информации, абстрактный синтаксис которой не является для него полностью известным и которую он планирует впоследствии подписать;

– при подписании данных для отправки, он должен отправлять данные, синтаксис которых является для него полностью известным при помощи отличительного кодирования, а также любые другие данные с зафиксированным кодированием, и должен подписать фактическое кодирование, которое отправляет;

– при проверке подписей в полученных данных, он должен проверить подпись для фактически полученных данных, скорее чем свое преобразование полученных данных в отличительное кодирование.

РАЗДЕЛ 2 – СТРУКТУРА СЕРТИФИКАТОВ ОТКРЫТЫХ КЛЮЧЕЙ

Структура сертификатов открытых ключей, определенная в данном разделе, может использоваться приложениями с требованиями к аутентификации, целостности, конфиденциальности и фиксации авторства.

Связывание открытого ключа и объекта предоставляется органом при помощи структуры данных с цифровой подписью, называемой сертификатом открытого ключа. Здесь определяется формат сертификата открытого ключа, включая механизм расширяемости и совокупность конкретных расширений сертификатов. Если по какой-либо причине орган аннулирует ранее выданный сертификат открытого ключа, пользователям нужно иметь возможность узнать о том, что аннулирование имело место, чтобы не использовать ненадежный сертификат. Списки аннулирования являются одной из схем, которые могут быть использованы для уведомления пользователей об аннулированиях. Здесь определяется формат списков аннулирования, включая механизм расширяемости и совокупность расширений списков аннулирования. Как для сертификатов, так и для списков аннулирования, другие тела могут также определять дополнительные расширения, являющиеся полезными для конкретных сред.

Система, использующая сертификаты открытых ключей, должна осуществлять проверку подлинности сертификата, прежде чем использовать данный сертификат для приложения. Процедуры для выполнения этой проверки также определяются здесь, включая проверку целостности непосредственно сертификата, его статус аннулирования, а также его подлинность относительно планируемого использования.

Справочник использует сертификаты открытых ключей при предоставлении услуг безопасности, включая:

- строгую аутентификацию между и среди компонентами справочника;
- аутентификацию, целостность и конфиденциальность операций справочника; а также
- целостность и аутентификацию хранимых данных.

7 Открытые ключи и сертификаты открытых ключей

Чтобы пользователь мог доверить открытый ключ другому пользователю, например, для аутентификации идентификационной информации того пользователя, открытый ключ должен быть получен из доверенного источника. Такой источник, называемый органом по сертификации (CA), сертифицирует открытый ключ путем выдачи сертификата открытого ключа, который связывает открытый ключ с объектом, который держит соответствующий частный ключ. Процедуры, используемые CA для обеспечения того, что объект на самом деле имеет владеет частным ключом, и другие процедуры, относящиеся к выдаче сертификатов открытых ключей, находятся вне области применения данной спецификации. Сертификат, форма которого определена ниже в данном пункте, обладает следующими свойствами:

- любой пользователь с доступом к открытому ключу органа по сертификации может возвратить открытый ключ, который был сертифицирован;
- никакая сторона кроме органа по сертификации не может изменить сертификат без обнаружения (сертификаты невозможно фальсифицировать).

Поскольку сертификаты невозможно фальсифицировать, они могут быть опубликованы путем размещения в Справочнике, без необходимости последнему прилагать особые усилия по их защите.

ПРИМЕЧАНИЕ 1. – Несмотря на то, что CA являются однозначно определенными при помощи выделенного имени в DIT, это не подразумевает, что между организацией CA и DIT существует какая-либо взаимосвязь.

Орган по сертификации производит сертификат пользователя путем подписания (см. п. 6.1) набора информации, включая выделенное имя пользователя и открытый ключ, а также дополнительный *уникальный идентификатор*, содержащий дополнительную информацию о пользователе. Точная форма содержания уникального идентификатора здесь не определяется и оставлена на усмотрение органа по сертификации, и может быть, например, идентификатором объекта, сертификатом, датой или любой другой формой сертификации по проверке подлинности выделенного имени. В частности, сертификат пользователя с выделенным именем A и уникальным идентификатором UA, произведенным органом по сертификации с именем CA и уникальным идентификатором UCA, имеет следующую форму:

$$CA\langle\langle A\rangle\rangle = CA\{V,SN,AI,CA,UCA,A,UA,Ap,T^A\},$$

где V – версия сертификата, SN – порядковый номер сертификата, AI – идентификатор алгоритма, использованного для подписания сертификата, UCA – дополнительный уникальный идентификатор CA, UA – дополнительный уникальный идентификатор пользователя A, T^A обозначает период действия сертификата и состоит их двух дат, первой и последней, когда сертификат действителен. Период действия сертификата представляет собой интервал времени, в течение которого CA гарантирует, что он будет поддерживать информацию о статусе сертификата, т. е. публиковать данные об аннулировании. Так как предполагается, что T^A будет изменяться в сроки не менее 24 часов, ожидается, что системы будут использовать Универсальное глобальное время в качестве эталонной временной оси. Подпись в сертификате может быть проверена на подлинность любым пользователем, знающем CAp. Следующий тип данных ASN.1 может использоваться для представления сертификатов:

```
Certificate                     ::=     SIGNED { SEQUENCE {
    version                 [0] Version DEFAULT v1,
    serialNumber                CertificateSerialNumber,
    signature                   AlgorithmIdentifier,
    issuer                      Name,
    validity                    Validity,
    subject                     Name,
    subjectPublicKeyInfo        SubjectPublicKeyInfo,
    issuerUniqueIdentifier  [1] IMPLICIT UniqueIdentifier OPTIONAL,
                                -- при наличии, версия должна быть v2 или v3
    subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL,
                                -- при наличии, версия должна быть v2 или v3
    extensions              [3] Extensions OPTIONAL
                                -- при наличии, версия должна быть v3 -- } }

    Version                     ::=     INTEGER { v1(0), v2(1), v3(2) }

    CertificateSerialNumber     ::=     INTEGER

AlgorithmIdentifier         ::=     SEQUENCE {
algorithm           ALGORITHM.&id ({SupportedAlgorithms}),
parameters              ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }
```

```
-- Определение следующей совокупности информационных объектов передается, возможно, в стандартизированные
-- профили или в свидетельства о соответствии протокольной реализации. Эта совокупность необходима для
-- определения табличного ограничения, накладываемого на компонент parameters AlgorithmIdentifier.
-- SupportedAlgorithms        ALGORITHM        ::=  { ... }

Validity                    ::=     SEQUENCE {
    notBefore       Time,
    notAfter        Time }

SubjectPublicKeyInfo                ::=     SEQUENCE {
    algorithm               AlgorithmIdentifier,
    subjectPublicKey        BIT STRING }

Time  ::=  CHOICE {
    utcTime             UTCTime,
    generalizedTime     GeneralizedTime }

Extensions ::= SEQUENCE OF Extension

Extension ::=   SEQUENCE {
    extnId      EXTENSION.&id ({ExtensionSet}),
    critical    BOOLEAN DEFAULT FALSE,
    extnValue   OCTET STRING
                -- содержит кодирование DER значения типа &ExtnType
                -- для объекта расширения, определенного extnId -- }

ExtensionSet   EXTENSION        ::=     { ... }
```

Прежде чем значение **Time** используется в любой операции сравнения, например, как часть правила соответствия в поиске, и если синтаксис **Time** был выбран в качестве типа **UTCTime**, значение двузначного поля года должно быть рационализировано в четырехзначное значение года следующим образом:

- Если двузначное значение находится в пределах от 00 до 49 включительно, к значению нужно прибавить 2000.
- Если двузначное значение находится в пределах от 50 до 99 включительно, к значению нужно прибавить 1900.

ПРИМЕЧАНИЕ 2. – Использование **GeneralizedTime** может предотвратить организацию межсетевого взаимодействия с реализациями, неосведомленными о возможности выбора между **UTCTime** или **GeneralizedTime**. Это входит обязанности определяющих области, в которых будут использоваться сертификаты, определенные в данной спецификации Справочника, например, профилирование групп, в отношении того, когда может использоваться **GeneralizedTime**. **UTCTime** не может использоваться ни в каком случае для представления дат после 2049 года.

version представляет собой версию кодированного сертификата. Если в сертификате присутствует компонент **extensions**, должна быть версия v3. Если присутствует компонент **issuerUniqueIdentifier** или **subjectUniqueIdentifier**, должна быть версия v2 or v3.

serialNumber представляет собой целое число, присваиваемое CA каждому сертификату. Значение **serialNumber** должно быть уникальным для каждого сертификата, выданного заданным CA (т. е. имя выдавшего органа и порядковый номер однозначно определяют сертификат).

signature содержит идентификатор алгоритма для алгоритма и хэш-функции, используемой CA при подписании сертификата (например, md5WithRSAEncryption, sha-1WithRSAEncryption, id-dsa-with-sha1 и т. д.)

issuer определяет объект, подписавший и выдавший сертификат.

validity представляет собой интервал времени, в течение которого CA гарантирует, что он будет поддерживать информацию о статусе сертификата.

subject определяет объект, связанный с открытым ключом, содержащимся в поле открытого ключа субъекта.

subjectPublicKeyInfo используется для хранения сертифицируемого открытого ключа и для определения алгоритма, примером которого является данный открытый ключ (например, rsaEncryption, dhpublicnumber, id-dsa и т. д.)

issuerUniqueIdentifier используется для однозначного определения выдавшего органа в случае повторного использования имени.

subjectUniqueIdentifier используется для однозначного определения субъекта в случае повторного использования имени.

ПРИМЕЧАНИЕ 3. – В ситуациях, когда выделенное имя может быть повторно присвоено другому пользователю органом по присвоению имен, CA могут использовать уникальный идентификатор для различения повторно использованных экземпляров. Тем не менее, если одному и тому же пользователю предоставляются сертификаты многими CA, рекомендуется, чтобы CA согласовывали присвоение уникальных идентификаторов как часть своих процедур по регистрации пользователя.

Поле **extensions** делает возможным добавление новых полей в структуру без изменения определения ASN.1. Поле расширения состоит из идентификатора расширения, флага критичности и кодирования значения данных типа ASN.1, связанного с определенным расширением. Для расширений, для которых значим порядок индивидуальных расширений в **SEQUENCE**, спецификация должна включать правила значимости порядка в них. Когда реализация обработки сертификата не узнает расширение, если значение флага критичности равно **FALSE**, она может игнорировать данное расширение. Если

значение флага критичности равно **FALSE**, неузнанные критические расширения должны вызвать признание структуры недействительной, т. е. в сертификате, неузнанное критическое расширение вызовет проверку подлинности подписи, используя отклонение сертификата. Когда использующая сертификат реализация узнает и способна обработать расширение, использующая сертификат реализация должна обработать расширение несмотря на значение флага критичности. Отметим, что любое расширение, помеченное как некритическое, вызовет несовместимое поведение между использующими сертификаты системами, которые будут обрабатывать расширение, и использующими сертификаты системами, которые не узнают данное расширение и будут игнорировать его.

Если внутри расширения появляются неизвестные элементы и расширение не отмечено как критическое, эти неизвестные элементы должны игнорироваться в соответствии в правилами расширяемости, документированными в Рек. МСЭ-Т X.519 | ИСО/МЭК 9594-5.

У СА имеется три возможности относительно расширения:

i) он может исключить расширение из сертификата;

ii) он может включить расширение и пометить его как некритическое;

iii) он может включить расширение и пометить его как критическое.

У устройства проверки подлинности имеется два возможных варианта действия относительно расширения:

i) оно может игнорировать расширение и принять сертификат (все прочие эквивалентные варианты);

ii) оно может обработать расширение и принять или отклонить сертификат в зависимости от содержимого расширения и условий, при которых происходит обработка (например, текущие значения переменных обработки тракта).

Некоторые расширения могут быть помечены только как критические. В этих случаях устройство проверки подлинности, понимающее расширение, обрабатывает его; принятие/отклонение сертификата зависит (по крайней мере частично) от содержимого расширения. Устройство проверки подлинности, не понимающее расширение, отклоняет сертификат.

Некоторые расширения могут быть помечены только как некритические. В этих случаях устройство проверки подлинности, понимающее расширение, обрабатывает его, и принятие/отклонение сертификата зависит (по крайней мере частично) от содержимого расширения. Устройство проверки подлинности, не понимающее расширение, принимает сертификат (если факторы, отличные от данного расширения, не приводят к его отклонению).

Некоторые расширения могут быть помечены как критические или некритические. В этих случаях устройство проверки подлинности, понимающее расширение, обрабатывает его, независимо от флага критичности. Устройство проверки подлинности, не понимающее расширение, принимает сертификат, если расширение помечено как некритическое (если факторы, отличные от данного расширения, не приводят к его отклонению), и отклоняет сертификат, если расширение помечено как критическое.

СА рассматривает включение расширения в сертификат, ожидая, что его цель будет соблюдаться, когда это возможно. Если необходимо, чтобы содержимое расширения рассматривалось до какого-либо доверия сертификату, СА должен пометить расширение как критическое. Это выполняется вместе с пониманием того, что любое устройство проверки подлинности, которое не обрабатывает расширение, отклонит сертификат (возможно, ограничив набор приложений, которые могут проверить сертификат). СА может пометить определенное расширение как некритическое для получения обратной совместимости с приложениями проверки подлинности, которые не могут обработать расширения. В случаях, когда возможность обратной совместимости и способности к взаимодействию с приложениями проверки подлинности, неспособными обработать расширения, более необходима, чем способность СА укреплять расширения, эти необязательно критические расширения должны быть помечены как некритические. Более вероятно, что СА установят необязательно критические расширения как некритические в течение периода перехода, в то время как приложения верификаторов по обработке сертификатов будут усовершенствоваться до способности обрабатывать расширения.

Определенные расширения могут быть определены в Рекомендациях МСЭ-Т | Международных стандартах или любой организацией, имеющей в этом потребность. Идентификатор объекта, который определяет расширение, должен быть определен в соответствии в Рек. МСЭ-Т X.660 | ИСО/МЭК 9834-1. Стандартные расширения для сертификатов определены в пункте 8 данной спецификации Справочника.

Следующий класс объектов используется для определения определенных расширений.

```
EXTENSION ::= CLASS {
    &id     OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX              &ExtnType
    IDENTIFIED BY       &id }
```

Существует два основных типа сертификатов открытого ключа, сертификаты оконечных объектов и СА-сертификаты.

Сертификат оконечного объекта представляет собой сертификат, выданный СА субъекту, не являющемуся органом, выдающим другие сертификаты открытых ключей.

СА-сертификат представляет собой сертификат, выданный СА субъекту, являющемуся СА и потому способному выдавать сертификаты открытых ключей. СА-сертификаты можно классифицировать при помощи следующих типов:

- Автоматически выданный сертификат – представляет собой сертификат, когда выдавший орган и субъект являются одним и тем же СА. СА может использовать автоматически выданные сертификаты, например, в течение операции перебора ключа для обеспечения доверия от старого ключа новому ключу.
- Автоматически подписанный сертификат – представляет собой особый случай автоматически выданных сертификатов, когда частный ключ, используемый СА для подписания сертификата, соответствует открытому ключу, сертифицированному в сертификате. СА может использовать автоматически подписанный сертификат, например, для рекламы своих открытых ключей или другой информации о своей деятельности.
- Перекрестный сертификат – представляет собой сертификат, когда выдавший орган и субъект являются различными СА. СА выдают сертификаты другим СА в качестве механизма авторизации существования СА субъекта (например, в строгой иерархии) или для распознавания существования СА субъекта (например, в распределенной модели доверия). Структура перекрестных сертификатов используется в обоих случаях. В некоторых ситуациях, противоречивые или перекрывающиеся требования к ограничениям, такие как ограничения имен, могут привести к тому, что СА будет выдавать более одного перекрестного сертификата другому СА.

Запись справочника любого пользователя А, который участвует в строгой аутентификации, состоит из сертификата(ов) А. Такой сертификат генерируется органом по сертификации пользователя А, который является записью в DIT. Орган по сертификации пользователя А, который может не быть уникальным, обозначается CA(A) или просто СА, если А подразумевается. Открытый ключ пользователя А может таким образом быть обнаружен любым пользователем, знающим открытый ключ СА. Таким образом, обнаружение открытого ключа является рекурсивным.

Если пользователь А, пытаясь получить открытый ключ пользователя В, уже получил открытый ключ CA(B), то процесс является законченным. Чтобы дать возможность А получить открытый ключ CA(B), запись справочника каждого органа по сертификации X содержит несколько сертификатов. Эти сертификаты являются сертификатами двух типов: Во-первых, существуют прямые сертификаты X, сгенерированные другими органами по сертификации. Во-вторых, существуют обратные сертификаты, сгенерированные непосредственно X, которые являются сертифицированными открытыми ключами других органов по сертификации. Существование этих сертификатов дает пользователям возможность создавать тракты сертификации от одной точки к другой.

Список сертификатов, необходимых для разрешения отдельному пользователю получить открытый ключ другого, известен как *тракт сертификации*. Каждый элемент списка является сертификатом органа по сертификации следующего элемента списка. Тракт сертификации от А к В (обозначается A→B):

- начинается с сертификата, произведенного CA(A), а именно CA(A)<<X1>> для некоторого объекта X1;
- продолжается дальнейшими сертификатами Xi<<Xi+1>>;
- заканчивается сертификатом пользователя В.

Поля **выдавший орган** и **субъект** каждого сертификата используются отчасти для определения действительного тракта. Для каждой пары смежных сертификатов в действительном тракте сертификации, значение поля **субъект** в одном сертификате должно соответствовать значению поля **выдавший орган** в последующем сертификате. Более того, значение поля **субъект** в первом сертификате должно соответствовать DN опоры доверия. Только имена в этих полях используются при проверке подлинности тракта сертификации. Имена в расширениях сертификатов не используются для этой цели. Тракт сертификации логически формирует непрерывную цепочку доверяемых точек в информационном дереве Справочника между двумя пользователями, желающими пройти аутентификацию. Точный метод, используемый пользователями А и В для получения трактов сертификации A→B и B→A, может меняться. Одним способом упрощения этого является создание иерархии СА, которая может или не может совпадать со всей или частью иерархии DIT. Преимуществом этого является то, что пользователи, чьи СА есть в иерархии, могут установить тракт сертификации между собой, используя Справочник без какой-либо предварительной информации. Чтобы сделать это возможным, каждый СА может хранить один сертификат и один обратный сертификат, обозначенные как соответствующие своему вышестоящему СА. Правило соответствия **distinguishedNameMatch**, определенное в п. 13.5.2 Рек. МСЭ-Т X.501 | ИСО/МЭК 9594-2, должно использоваться при сравнении Выделенного имени (DN) в поле **выдавший орган** одного сертификата с DN в поле **субъект** другого.

Пользователь может получить один или несколько сертификатов от одного или нескольких органов по сертификации. Каждый сертификат несет имя органа по сертификации, который его выдал. Следующие типы данных ASN.1 могут использоваться для представления сертификатов и тракта сертификации:

```
Certificates                ::=  SEQUENCE {
    userCertificate             Certificate,
    certificationPath           CertPath OPTIONAL }

CertificationPath           ::=  SEQUENCE {
    userCertificate             Certificate,
    theCACertificates           SEQUENCE OF CertificatePair OPTIONAL }
```

Более того, следующий тип данных ASN.1 может использоваться для представления прямого тракта сертификации. Данный компонент содержит тракт сертификации, который может указать на своего создателя.

```
CertPath ::= SEQUENCE OF CrossCertificates

CrossCertificates ::= SET OF Certificate

PkiPath ::= SEQUENCE OF Certificate
```

PkiPath используется для представления тракта сертификации. В последовательности порядок сертификатов таков, что субъект первого сертификата является выдавшим второй сертификат и т. д.

Каждый сертификат в тракте сертификации должен быть уникален. Ни один сертификат не может встретиться более чем однократно в значении компонента **theCACertificates CertificationPath** или в значении **Certificate** в компоненте **CrossCertificates** в **CertPath** или в значении **Certificate** в **PkiPath**.

7.1 Генерация пары ключей

Общая политика управления безопасностью реализации должна определять жизненный цикл пары ключей, и, таким образом, находится вне области применения данной структуры. Тем не менее, для общей безопасности жизненно важно, чтобы все частные ключи оставались известными только пользователю, которому они принадлежат.

Для человека-пользователя запомнить данные ключа не является простым, поэтому необходимо использовать соответствующий метод для хранения их удобным переносным способом. Одним возможным механизмом является использование "Смарт-карты". В ней будут содержаться все частные и (по выбору) открытые ключи пользователя, сертификат пользователя и копия открытого ключа органа по сертификации. Использование такой карты должно быть дополнительно защищено, например, по меньшей мере использованием Персонального идентификационного номера (PIN), повышающего безопасность системы путем предъявления к пользователю требования иметь при себе карточку и уметь получить доступ к ней. Конкретный метод, выбранный для хранения таких данных, тем не менее, находится вне области применения данной спецификации Справочника.

Три способа, которым можно создать пару ключей пользователя, состоят в следующем:

a) Пользователь генерирует свою собственную пару ключей. Преимущество данного метода заключается в том, что частный ключ пользователя никогда не подлежит раскрытию другому объекту, но данный метод требует определенного уровня умений пользователя.

b) Пара ключей генерируется третьей стороной. Третья сторона должна раскрыть частный ключ пользователю физически конфиденциальным способом, затем активно уничтожить всю информацию, связанную с созданием пары ключей, в том числе сами ключи. Должны применяться соответствующие физические меры безопасности для обеспечения того, что третья сторона и операции с данными защищены от фальсификации.

c) Пара ключей генерируется CA. Представляет собой особый случай пункта b), и все рассуждения применимы в данном случае.

ПРИМЕЧАНИЕ. – Орган по сертификации уже представляет доверяемые функциональные возможности в отношении пользователя и должен являться предметом необходимых физических мер безопасности. Преимущество данного метода состоит в том, что для сертификации не требуется конфиденциальная передача данных в CA.

Используемая криптосистема налагает особые ограничения на генерацию ключей.

7.2 Создание сертификата открытого ключа

Сертификат открытого ключа связывает открытый ключ и уникальное выделенное имя пользователя, которого он описывает. Таким образом:

a) орган по сертификации должен быть удовлетворен идентификационной информацией пользователя, прежде чем создать сертификат для него;

b) орган по сертификации не должен выдавать сертификаты двум пользователям с одним и тем же именем.

Важно, чтобы передача информации органу по сертификации не рассекречивалась, и должны предприниматься соответствующие физические меры безопасности. В этом отношении:

a) Было бы серьезным нарушением безопасности, если бы CA выдал сертификат пользователю с открытым ключом, который был фальсифицирован.

b) Если используется способ генерации пар ключей 7.1 b) или 7.1 c), то частный ключ пользователя должен передаваться пользователю конфиденциальным способом.

c) Если используется способ генерации пар ключей 7.1 a) или 7.1 b), то пользователь может использовать различные методы (диалоговые или автономные) для сообщения своего открытого ключа CA конфиденциальным способом. Диалоговые методы могут обеспечить некоторую дополнительную гибкость для удаленных операций, выполняемых между пользователем и CA.

Сертификат открытого ключа является общедоступным блоком информации, и нет необходимости предпринимать особенные меры безопасности для его перемещения в Справочник. Так как это производится в автономном режиме органом по сертификации от имени пользователя, которому должна быть предоставлена его копия, пользователю нужно только сохранить эту информацию в своей записи справочника при последующем доступе к справочнику. В качестве альтернативы CA может вписать сертификат за пользователя, тогда данному агенту должны быть предоставлены соответствующие права доступа.

7.3 Период действия сертификата

Орган, который выдает сертификат (открытого ключа или атрибута), также должен указывать период действия сертификата, который выдает. В общем, сертификаты являются предметом для возможного последующего аннулирования. Данное аннулирование и уведомление об аннулировании могут выполняться непосредственно органом, выдавшим сертификат, или косвенно, другим органом, надлежащим образом авторизованным органом, выдавшим сертификат. Орган, выдающий сертификат, должен заявить, возможно путем опубликованного заявления о своих действиях, при помощи самих сертификатов или другим определенным образом, где:

- сертификаты не могут быть аннулированы; или
- сертификаты могут быть аннулированы тем же органом, выдающим сертификат, непосредственно; или
- орган, выдающий сертификат, авторизует другой объект на выполнение аннулирования.

Органы, аннулирующие сертификаты, должны заявить похожими способами, какой(ие) механизм(ы) могут использоваться зависимыми сторонами для получения информации о статусе аннулирования в отношении сертификатов, выданных данным органом. В данной Спецификации определяется механизм списка аннулированных сертификатов (CRL), но не исключается применение альтернативных механизмов. К таким альтернативным механизмам относится, например, диалоговый протокол статуса сертификата (OCSP), определенный в IETF RFC 2560[1)]. Применяя этот протокол, зависимая сторона (клиент) запрашивает статус аннулирования сертификата на сервере OCSP. Сервер может использовать **CRL** или другие механизмы для проверки статуса сертификата и ответа клиенту соответственно. Если OCSP может быть использован зависимыми сторонами для проверки статуса сертификата, в IETF RFC 3280[2)] содержится расширение сертификата (доступ к информации органа), которое включается в такие сертификаты и предоставляет достаточную информацию для доступа к соответствующему серверу OCSP. Зависимые стороны проверяют информацию о статусе аннулирования надлежащим образом для всех сертификатов, рассматриваемых в течение процедуры обработки тракта, описанной в п. 10, и процедуры обработки тракта делегирования, описанной в п. 16, для проверки подлинности сертификата.

Только CA, авторизованный для выдачи CRL, может принять решение о делегировании данного органа другому объекту. Если делегирование выполнено, оно должно проверяться в момент проверки сертификата/CRL. Для этой цели может использоваться расширение **cRLDistributionPoints**. Поле **cRLIssuer** данного расширения должно быть заполнено именем (именами) объектов, отличных от самого выдавшего сертификат, которые были авторизованы для выдачи CRL в отношении статуса аннулирования рассматриваемых сертификатов.

Сертификаты, включая сертификаты открытых ключей, а также сертификаты атрибутов, должны иметь связанную с ними продолжительность существования, в конце которой их срок истекает. Для обеспечения непрерывности обслуживания, орган должен обеспечить временную доступность замещающий сертификатов для замены сертификатов с истекшим/истекающим сроком действия. Дата уведомления об аннулировании представляет собой дату и время, когда уведомление об аннулировании для сертификата впервые появляется в CRL, несмотря на то, является он основным CRL или dCRL. В CRL, уведомление об аннулировании является значением, содержащимся в поле **thisUpdate**. Дата аннулирования представляет собой дату и время, когда CA фактически аннулировал сертификат и может отличаться от даты его первого появления в CRL. В CRL, дата аннулирования является значением, содержащимся в компоненте **revocationDate**. Дата недействительности представляет собой дату и время, когда становится известно или подозревается, что частный ключ был рассекречен или что сертификат должен считаться недействительным по иным причинам. Данная дата может предшествовать дате аннулирования. В CRL, дата недействительности является значением, содержащимся в расширении записи **invalidityDate**.

Два связанных момента описаны ниже:

- Период действия сертификата может быть разработан таким образом, что каждый становится действительным в момент истечения срока своего предшественника, или может допускаться перекрывание этих сроков. Последний способ избавляет орган от необходимости вводить и распространять большое количество сертификатов, срок действия которых может истечь одновременно.
- Сертификаты с истекшим сроком действия обычно будут удаляться из Справочника. Задачей политики безопасности и обязанностью органа является хранение старых сертификатов в течение периода времени, если предоставляется услуга фиксации авторства данных.

Сертификаты могут быть аннулированы до срока их истечения, например, если предполагается, что частный ключ пользователя был фальсифицирован или пользователь более не должен сертифицироваться органом, или предполагается, что был фальсифицирован сертификат органа. Об аннулировании сертификата пользователя или сертификата органа орган должен сообщить и при необходимости сделать доступным новый сертификат. Орган может затем информировать держателя сертификата об его аннулировании при помощи некоторых процедур в автономном режиме.

Орган, который выдает и впоследствии аннулирует сертификаты:

a) может иметь требование поддерживать и проверять запись своих событий аннулирования для всех типов сертификатов, выданных данным органом (например, сертификаты открытых ключей, сертификаты атрибутов, выданные оконечным объектам, а также другим органам);

---

1) IETF RFC 2560, *X.509 Интернет Диалоговый протокол статуса сертификата (OCSP) Структура открытых ключей,* июнь 1999 г.

2) IETF RFC 3280, *Интернет X.509 Сертификат структуры открытых ключей и профиль списка аннулированных сертификатов (CRL)*, апрель 2002 г.

b) должен предоставлять информацию о статусе аннулирования зависимым сторонам с использованием CRL, диалогового протокола статуса сертификата или какого-либо другого механизма публикации информации о статусе аннулирования;

c) при использовании CRL должен поддерживать и публиковать CRL, даже если списки аннулированных сертификатов пусты;

d) при использовании только частичных CRL, должен выдавать полное множество частичных CRL, охватывающее всю совокупность сертификатов, о статусе аннулирования которых будет сообщено с использованием механизма CRL. Таким образом, полное множество частичных CRL должно быть эквивалентно полному CRL для той же совокупности сертификатов, при условии что выдавший CRL орган не использовал частичные CRL.

Зависимые стороны могут использовать несколько механизмов для определения местонахождения информации о статусе аннулирования, предоставляемой органом. Например, непосредственно в сертификате может существовать указатель, который направляет зависимую сторону в местоположение, где предоставлена информация об аннулировании. В списке аннулирования может находиться указатель, переадресовывающий зависимую сторону в другое местоположение. Зависимая сторона может определять местонахождение информации об аннулировании в хранилище (например, справочнике) или при помощи других средств, находящихся вне области применения данной Спецификации (т. е. конфигурируемых локально).

Поддержка записей Справочника, на которые влияют списки аннулирования органа, является обязанностью Справочника и его пользователей, действующих в соответствии с политикой безопасности. Например, пользователь может изменить запись своего объекта путем замещения старого сертификата новым. Последний затем должен использоваться при аутентификации пользователя к Справочнику.

Если списки аннулирования публикуются в Справочнике, они хранятся в записях как атрибуты следующих типов:

– список аннулированных сертификатов;
– список аннулированных органов;
– список аннулированных дельт;
– список аннулированных сертификатов атрибутов;
– список аннулированных органов атрибутов.

```
CertificateList          ::=    SIGNED { SEQUENCE {
    version                     Version OPTIONAL,
                                -- при наличии, версия должна быть v2
    signature                   AlgorithmIdentifier,
    issuer                      Name,
    thisUpdate                  Time,
    nextUpdate                  Time OPTIONAL,
    revokedCertificates         SEQUENCE OF SEQUENCE {
        serialNumber                CertificateSerialNumber,
        revocationDate              Time,
        crlEntryExtensions      Extensions OPTIONAL } OPTIONAL,
    crlExtensions      [0]      Extensions OPTIONAL }}
```

version представляет собой версию кодированного списка аннулирования. Если компонент **extensions** помеченный как критический, присутствует в списке аннулирования, версия должна быть v2. Если компонента **extensions**, помеченного как критический, нет в списке аннулирования, версия может либо отсутствовать, либо быть v2.

signature содержит идентификатор алгоритма, используемого органом для подписания списка аннулирования.

issuer определяет объект, подписавший и выдавший список аннулирования.

thisUpdate представляет собой дату/время, когда был выдан данный список аннулирования.

nextUpdate, при наличии, указывает дату/время, к которой будет выдан следующий список аннулирования в данной серии. Следующий список аннулирования может быть выдан до наступления указанной даты, но не может быть выдан после ее наступления.

revokedCertificates определяет сертификаты, которые были аннулированы. Аннулированные сертификаты определяются своими порядковыми номерами. Если не был аннулирован ни один сертификат, охватываемый данным CRL, настойчиво рекомендуется, чтобы параметр **revokedCertificates** был изъят из CRL, а не включен с пустым значением **SEQUENCE**.

crlExtensions, при наличии, содержит одно или несколько расширений CRL.

ПРИМЕЧАНИЕ 1. – Проверка полного списка сертификатов является локальной задачей. Не должно предполагаться, что список будет иметь какой-то определенный порядок, если только выдавшим органом не были определены конкретные правила упорядочения, например, в политике данного органа.

ПРИМЕЧАНИЕ 2. – Если услуга фиксации авторства зависит от ключей, предоставляемых органом, услуга должна обеспечивать, что все соответствующие ключи органа (аннулированные или с истекшим сроком действия) и списки аннулирования со временной отметкой заархивированы и сертифицированы данным органом.

ПРИМЕЧАНИЕ 3. – Если любые расширения, включенные в **CertificateList**, определены как критические, должен присутствовать элемент версии **CertificateList**. Если не включены расширения, помеченные как критические, элемент версии может отсутствовать. Отсутствие **version** может позволить реализации, поддерживающей только CRL версии 1, все еще использовать CRL, если при проверке последовательности **revokedCertificates** в CRL она не встречает расширение. Реализация, поддерживающая CRL версии 2 и выше, при отсутствии версии может также оптимизировать его обработку, если на ранних этапах обработки может определить, что в CRL нет критических расширений.

ПРИМЕЧАНИЕ 4. – Когда реализация, обрабатывающая список аннулированных сертификатов, не узнает критическое расширение в поле **crlEntryExtensions**, она должна предполагать, что по меньшей мере определенный сертификат был аннулирован и более не является действительным, и производить дополнительные действия, связанные с данным аннулированным сертификатом, как предписано локальной политикой. Когда реализация не узнает критическое расширение в поле **crlEntryExtensions**, она должна предполагать, что определенные сертификаты были аннулированы и более не являются действительными. Тем не менее, в последнем случае, так как список может быть неполным, сертификаты, которые не были определены как аннулированные, не могут предположительно быть действительными. В таком случае, локальная политика должна предписывать действия, которые должны быть предприняты. В любом случае, локальная политика может предписывать действия, дополняющие и/или усиливающие указанные в данной Спецификации.

ПРИМЕЧАНИЕ 5. – Если расширение влияет на обработку списка (например, многие CRL нужно сканировать для проверки полного списка аннулированных сертификатов, или одна запись может представлять ряд сертификатов), тогда расширение должно быть обозначено как критическое в поле **crlExtensions** независимо от того, где расширение размещено в CRL. Расширение, обозначенное в поле **crlEntryExtensions** записи, должно быть помещено в эту запись и должно влиять только на сертификат(ы), определенный(ые) в этой записи.

ПРИМЕЧАНИЕ 6. – Стандартные расширения для CRL определены в п. 8 данной спецификации Справочника.

Если в расширении появляются неизвестные элементы и расширение не помечено как критическое, эти неизвестные элементы должны игнорироваться в соответствии с правилами расширяемости, документированными в п. 12.2.2 Рек. МСЭ-Т X.519 | ИСО/МЭК 9594-5

7.4 Отказ от цифровой подписи

Любой участник события может впоследствии решить отказаться от чего-либо, что участник подписал цифровой подписью в том событии. Например, кто-то может оспорить чье-либо участие в установлении ключа авторство подписанного электронного почтового сообщения так же легко, как кто-то может оспорить чье-либо подписание документа с целью быть связанным с содержимым данного документа. Отказ может не быть успешным. В структуре фиксации авторства, Рек. МСЭ-Т X.813 | ИСО/МЭК 10181-4, процесс разрешения оспаривания описывается следующим образом:

1) генерация доказательств;
2) передача, хранение и поиск доказательств;
3) проверка доказательств; и
4) разрешение оспаривания.

Сгенерированное доказательство может включать, но не ограничивается:

– записи проверки, имеющие отношение к событию и утверждению намерения;
– заявления, сделанные нотариусами третьей стороны;
– заявления политики;
– информация, подписанная цифровой подписью, включая записи проверки и нотариальные заявления;
– временные отметки информации, подписанной цифровой подписью;
– сертификаты, поддерживающие цифровую подпись;
– соответствующая информация об аннулировании, опубликованная и доступная на момент оспариваемого события; и
– любые аннулирования сертификатов, произошедшие после времени события, указывающие на фальсификацию ключа, произошедшее до наступления события.

Целостность хранимых данных, которые могут быть представлены как доказательство, может поддерживаться разными способами, например, управление доступом, хранение хэшей доверенной третьей стороной, цифровая подпись. Также может потребоваться периодическое усиление защиты хранимых данных для противодействия улучшениям в компьютерной обработке и/или крипто-анализе.

ПРИМЕЧАНИЕ. – В данной спецификации Справочника не определяется ни тип и количество сгенерированных доказательств, ни уровень целостности. Тем не менее, ожидается, что уровень усилий будет соизмерим со связанным риском.

Проверка доказательств может потребовать повторную проверку подлинности данных, например, сообщений, документов, сертификатов, CRL и временных отметок, которые использовались в первоначальном процессе проверки подлинности. Факт истечения срока действия сертификата не должен предотвращать его использование для повторной проверки подлинности подписей, созданных в течение периода действия данного сертификата. Сертификат, который был аннулирован, может быть использован, если возможно определить, что сертификат был действителен на момент оспариваемого события.

Даже если все цифровые доказательства, описанные выше, признаны технически действительными, другие условия, например, намерение, понимание или умение подписавшего лица могут позвонить ему успешно отказаться от них.

8 Расширения сертификатов открытых ключей и CRL

Расширения сертификатов, определенные в данном пункте, могут использоваться с сертификатами открытых ключей, если не обусловлено иное. Расширения для использования с сертификатами атрибутов определены в п. 15. Расширения CRL, определенные в данном пункте, могут использоваться в CRL, CARL, а также для ACRL и AARL, определенных в п. 17.

В данном пункте определены расширения в следующих областях:

a) *Информация о ключах и политике*: Эти расширения сертификатов и CRL передают дополнительную информацию о связанных ключах, включая идентификаторы ключей для субъекта и ключи выдавшего органа, индикаторы планируемого или ограниченного использования ключей, а также индикаторы политики сертификатов.

b) *Атрибуты субъекта и выдавшего органа*: Эти расширения сертификатов и CRL поддерживают альтернативные имена, различных форм имен, для субъекта сертификата, для выдавшего сертификат или выдавшего CRL. Эти расширения могут также передавать дополнительную информацию атрибутов о субъекте сертификата, для того чтобы помочь пользователю сертификата быть уверенным, что субъектом сертификата является определенное лицо или объект.

c) *Ограничения тракта сертификации*: Эти расширения сертификата позволяют включить спецификации ограничений в CA-сертификаты, т. е. сертификаты для CA, выданные другими CA, для упрощения автоматизированной обработки трактов сертификации, когда используются несколько политик сертификатов. Несколько политик сертификатов имеют место, когда политики меняются для различных применений в среде, или когда возникает взаимодействие со внешними средами. Ограничения могут ограничить типы сертификатов, которые могут выдаваться субъектом CA или которые могут появляться впоследствии в тракте сертификации.

d) *Основные расширения CRL*: Эти расширения CRL позволяют CRL включить указание причины аннулирования, предоставить временную приостановку сертификата, а также включить последовательные номера выдачи CRL, для того чтобы позволить пользователям сертификата обнаруживать пропущенные CRL в последовательности от одного выдавшего CRL органа.

e) *Точки распределения CRL и дельта-CRL*: Эти сертификаты и расширения CRL позволяют разделить полную совокупность информации об аннулировании от одного CA на отдельные CRL, а также позволяют объединить в один CRL информацию об аннулировании от многих CA. Данные расширения также поддерживают использование частичных CRL, указывая только изменения с момента выдачи последнего CRL.

Включение любого расширения в сертификат или CRL выполняется на усмотрение органа, выдавшего данный сертификат или CRL.

В сертификате или CRL расширение помечается как критическое или некритическое. Если расширение помечено как критическое и система, использующая сертификат, не узнает тип поля расширения или не выполняет семантику расширения, тогда эта система должна считать сертификат недействительным. Если расширение помечено как некритическое, то система, использующая сертификат, которая не узнает или не выполняет данный тип расширения, может обработать остаток сертификата, игнорируя расширение. Если расширение помечено как некритическое, то система, использующая сертификат, которая не узнает расширение, должна обработать расширение. Определения типов расширений в данной спецификации Справочника указывают, является ли расширение всегда критическим, всегда некритическим или решение о критичности может быть принято выдавшим сертификат или CRL органом. Причина, по которой некоторые расширения должны быть всегда некритическими, заключается в позволении реализациям, использующим сертификаты, которым не нужно использовать такие расширения, исключить их поддержку без подвергания опасности возможности к взаимодействию со всеми органами по сертификации.

ПРИМЕЧАНИЕ. – Система, использующая сертификат, может потребовать наличия в сертификате определенных некритических расширений, для того чтобы тот сертификат был рассмотрен как приемлемый. Необходимость включения таких расширений может быть связана с правилами локальной политики пользователя сертификата или может быть правилом политики CA, указанным для системы, использующей сертификат, путем включения определенного идентификатора политики сертификатов в расширение политик сертификатов с данным расширением, помеченным как критическое.

Для всех расширений сертификатов, расширений CRL и расширений записи CRL, определенных в данной спецификации Справочника, должно существовать не более одного экземпляра каждого типа расширения в любом сертификате, CRL или записи CRL соответственно.

8.1 Обработка политики

8.1.1 Политика сертификатов

В данной структуре содержится три типа объекта: пользователь сертификата, орган по сертификации и субъект сертификата (оконечный объект). Каждый объект действует в условиях обязательств перед другими двумя объектами и, в ответ, получает ограниченные гарантии, предлагаемые ими. Данные обязательства и гарантии определяются в политике сертификатов. Политика сертификатов представляет собой документ (обычно на языке без шифрования). На нее может даваться ссылка путем уникального идентификатора, который может быть включен в расширение политик безопасности сертификата, выданного органом по сертификации оконечному объекту, и от которого зависит пользователь сертификата. Сертификат может быть выдан в соответствии в одной или несколькими политиками. Определение политики и присвоение идентификатора производится органом по политике. Совокупность политик, управляемых органом по политике, называется областью политики. Все сертификаты выдаются в соответствии с политикой, даже если политика нигде не записана или если на нее нет ссылок в сертификате. В данной Спецификации не предписывается стиль содержания политики сертификатов.

Пользователь сертификатов может быть привязан к своим обязательствам в соответствии с политикой сертификатов путем действия по импорту открытого ключа органа и использования его в качестве основы доверия, или путем зависимости от сертификата, который включает соответствующий идентификатор политики. Орган по сертификации может быть привязан к своим обязательствам в соответствии с политикой путем действий по выдаче сертификата, который включает соответствующий идентификатор политики. Оконечный объект может быть привязан к своим обязательствам в соответствии с политикой путем действий запроса и принятия сертификата, который включает соответствующий

идентификатор политики, а также использованием соответствующего частного ключа. Реализации, которые не используют расширение политик сертификатов, должны получить требуемое связывание при помощи каких-либо других средств.

Для объекта просто заявить о соответствии политике не полностью удовлетворяет требованиям гарантии других объектов в структуре. Им требуется какая-либо причина, чтобы поверить, что другие стороны действуют в надежной реализации политики. Тем не менее, если так явно указано в политике, пользователи сертификатов могут принять гарантии органа по сертификации, что его оконечные объекты согласны быть связанными своими обязательствами согласно политике, без необходимости непосредственного подтверждения ими. Данный аспект политики сертификатов находится вне области применения данной Спецификации.

Орган по сертификации может разместить ограничения на использование своих сертификатов, чтобы контролировать риск того, что, как он предполагает, является результатом выдачи сертификатов. Например, он может ограничить сообщество пользователей сертификатов, цели, для которых они могут использовать его сертификаты и/или тип данных и степень повреждений, которые он готов исправить в случае повреждения его части или части оконечных объектов. Данные вопросы должны быть определены в политике сертификатов.

Дополнительная информация для помощи затрагиваемым объектам в понимании положений политики может быть включена в расширение политик сертификатов в форме квалификаторов политики.

8.1.2 Перекрестная сертификация

Орган по сертификации может является субъектом сертификата, выданного другим органом по сертификации. В этом случае, сертификат называется перекрестным сертификатом, орган по сертификации, являющийся субъектом сертификата, называется субъектом-органом по сертификации, а орган по сертификации, который выдает перекрестный сертификат, называется промежуточным органом по сертификации (см. рисунок 2). Как перекрестный сертификат, так и сертификат оконечного объекта могут содержать расширение политик сертификата.

Гарантии и обязательства, несомые субъектом-органом по сертификации, промежуточным органом по сертификации и пользователем сертификата, определяются в политике сертификатов, определенной в перекрестном сертификате, в соответствии с которой субъект-орган по сертификации может действовать как оконечный объект или от имени оконечного объекта. И гарантии и обязательства, несомые субъектом сертификата, субъектом-органом по сертификации и промежуточным органом по сертификации, определяются в политике сертификатов, определенной в сертификате оконечного объекта, в соответствии с которой промежуточный орган по сертификации может действовать как пользователь сертификата или от имени пользователя сертификата.

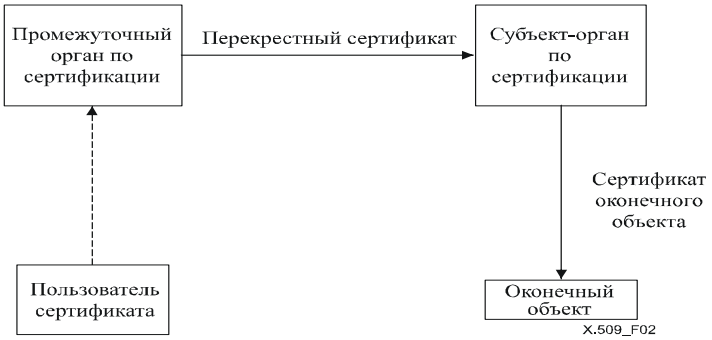


Рисунок 2 – Перекрестная сертификация

Говорят, что тракт сертификации действителен в условиях совокупности политик, являющихся общими для всех сертификатов в тракте.

Промежуточный орган по сертификации может в свою очередь быть субъектом сертификата, выданного другим органом по сертификации, таким образом создавая тракты сертификации, длиной превышающие два сертификата. И так как доверие испытывает ослабление из-за того, что тракты сертификации увеличиваются в длине, при контроле необходимо обеспечивать, чтобы сертификаты оконечных объектов с неприемлемо низким соответствующим уровнем доверия были отклонены пользователем сертификата. Это является частью функции процедуры обработки тракта сертификации.

В дополнение к описанной выше ситуации, необходимо рассмотреть два особых случая:

a) орган по сертификации не использует расширение политик сертификатов для передачи своих требований к политике пользователям сертификатов; и

b) пользователь сертификата или промежуточный орган по сертификации делегирует работу по контролю политики следующему сертификату в тракте.

В первом случае, сертификат не должен совсем содержать расширение политик сертификации. В результате, совокупность политик, в условиях которых тракт действителен, будет нулевой. Но, тем не менее, тракт может быть действителен. Пользователи сертификатов все еще должны обеспечивать, что они используют сертификат в соответствии с политиками органов в тракте.

Во втором случае, пользователь сертификата или промежуточный орган по сертификации должен включить специальное значение *любая-политика* в *начальный-набор-политик* или перекрестный сертификат. Когда сертификат включает специальное значение *любая-политика,* он не должен включать любые другие идентификаторы политики сертификатов. Идентификатор *any-policy* не должен иметь любых соответствующих идентификаторов политики.

Пользователь сертификата может обеспечить, что все его обязательства переносятся в соответствии со стандартом, путем установки индикатора *начальная-явная-политика*. При этом способе только органы, использующие стандартное расширение политик сертификатов в качестве способа получения связывания, принимаются в тракте, и у пользователей сертификатов нет дополнительных обязательств. Поскольку органы также привлекают обязательства, когда они действуют как пользователь сертификата или от имени пользователя сертификата, они могут обеспечивать, что все их обязательства передаются в соответствии со стандартом путем установки **requireExplicitPolicy** в перекрестном сертификате.

8.1.3 Отображение политики

Некоторые тракты сертификации могут пересекать границы между областями политик. Гарантии и обязательства, в соответствии с которыми выдается перекрестный сертификат, могут быть фактически эквивалентны некоторым или всем гарантиям и обязательствами, в соответствии с которыми субъект-орган по сертификации выдает сертификаты оконечным объектам, даже несмотря на то, что органы по политике, под руководством которых действуют оба органа по сертификации, могут выбрать другие уникальные идентификаторы для этих фактически эквивалентных политик. В таком случае, промежуточный орган по сертификации может включить расширение отображений политики в перекрестный сертификат. В расширении отображений политики, промежуточный орган по сертификации гарантирует пользователю сертификата, что он продолжит получать привычные гарантии и что он должен продолжать выполнять свои привычные обязательства, даже несмотря на то, что последующие объекты в тракте сертификации действуют в другой области политик. Промежуточный орган по сертификации должен включить одно или несколько отображений для каждого подмножества политик, согласно которым он выдал перекрестный сертификат, и не должен включать отображения для любых других политик. Если одна или несколько политик сертификата, в соответствии с которыми действует субъект-орган по сертификации, идентична тем, в соответствии с которыми действует промежуточный орган по сертификации (т. е. имеет тот же самый уникальный идентификатор), то эти идентификаторы должны быть исключены их расширения отображений политики, но включены в расширение политик сертификатов.

Отображение политики имеет эффект преобразования всех идентификаторов политики в сертификатах дальше вниз по тракту сертификации в идентификатор эквивалентной политики, как узнаваемой пользователем сертификата.

Политики не должны отображаться в специальное значение или из специального значения *any-policy.*

Пользователи сертификатов могут определить, что сертификатам, выданным в области политик, отличной от их собственной, не должно оказываться доверие, даже несмотря на то, что доверенный промежуточный орган по сертификации может определить его политику как фактически эквивалентную его собственной. Он может выполнить это путем установки *initial-policy-mapping-inhibit input* в процедуру проверки подлинности тракта. Кроме того, промежуточный орган по сертификации может выполнить аналогичное определение от имени своих пользователей сертификатов. Чтобы обеспечить, что пользователи сертификатов правильно внедряют данное требование, он может установить **inhibitPolicyMapping** в расширение ограничений политики.

8.1.4 Обработка тракта сертификации

Пользователь сертификата сталкивается с необходимостью выбора между двумя стратегиями:

a) он может требовать, чтобы тракт сертификации был действителен согласно по меньшей мере одной совокупности политик, предопределенных пользователем; или

b) он может запросить модуль проверки подлинности тракта сообщить совокупность политик, для которых действителен тракт сертификации.

Первая стратегия может быть более подходящей, когда пользователь сертификата знает, *априори*, совокупность политик, приемлемых для его предполагаемого использования.

Вторая стратегия может быть более подходящей, когда пользователь сертификата не знает, *априори*, совокупности политик, приемлемых для его предполагаемого использования.

В первом примере, процедура проверки подлинности тракта сертификации будет обозначать, что тракт, является действительным, только если он действителен согласно одной или нескольким политикам, определенных в *начальном-наборе-политик*, и будет возвращать подмножество *начальный-набор-политик,* согласно которому тракт является действительным. Во втором примере, процедура проверки подлинности тракта сертификации будет обозначать, что тракт не является действительным согласно *начальному-набору-политик*, но является действительным согласно непересекающемуся множеству *набору-политик-ограниченных-органом*. Тогда пользователь сертификата должен определить, совместимо ли планируемое им использование сертификата с одной или несколькими политиками сертификата, согласно которым действителен тракт. Путем установки *начального-набора-политик* в значение *любая-политика,* пользователь сертификата может привести к тому, что процедура будет возвращать действительный результат, если тракт является действительным согласно любой (неопределенной) политике.

8.1.5 Автоматически выданные сертификаты

Существует три обстоятельства, при которых орган по сертификации может выдать сертификат самому себе:

a) в качестве удобного способа кодирования открытого ключа, связанного с частным ключом, используемым при подписании сертификата, таким образом, что он может быть передан и сохранен в качестве основы доверия системами, использующими его сертификаты;

b) для сертификации дополнительных открытых ключей CA, используемых для целей, отличных от охватываемых категорией a) (таких, как OCSP и возможно подписание CRL); и

c) для замены своих собственных сертификатов с истекшим сроком действия.

Данные типы сертификатов называются автоматически выданными сертификатами и могут быть узнаны по факту, что присутствующие в нем имена выдавшего органа и субъекта совпадают. Для целей проверки подлинности тракта, автоматически выданные сертификаты категории a) являются автоматически подписанными сертификатами и поэтому проверяются при помощи открытых ключей, содержащихся в них, и если они встречаются в тракте, они должны игнорироваться.

Автоматически выданные сертификаты типа b) могут существовать только как оконечные сертификаты тракта и должны обрабатываться как оконечные сертификаты.

Автоматически выданные сертификаты типа c) (также известные как автоматически выданные промежуточные сертификаты) могут существовать как промежуточные сертификаты в тракте. Что касается хорошей практики, при замене ключа, находящегося на точке истечения срока действия, орган по сертификации должен запросить выдачу любых-в-границах перекрестных сертификатов, которые ему требуются для замены открытого ключа до использования ключа. Тем не менее, если автоматически выданные сертификаты данной категории встречаются в тракте, они должны обрабатываться как промежуточные сертификаты, со следующим исключением: они не увеличивают длину тракта для целей обработки компонента **pathLenConstraint** расширения **basicConstraints** и значений *пропущенные-сертификаты*, связанных в индикаторами *обработка-запрета-отображения-политик* и *обработка-явной-политики.*

Если орган использует один и тот же ключ для подписания сертификатов и CRL, должен использоваться один автоматически выданный сертификат категории a). Если орган для подписания CRL использует ключ, отличный от ключа для подписания сертификатов, орган может принять решение о выдаче двух автоматически выданных сертификатов категории a), по одному для каждого ключа. В такой ситуации, пользователям сертификатов потребуется доступ к обоим автоматически выданным сертификатам для установления отдельных опор доверия для сертификатов и CRL, подписанных данным органом. В качестве альтернативы, орган может выдать один автоматически выданный сертификат категории a) для подписания сертификатов и один автоматически выданный сертификат категории b) для подписания CRL. В такой ситуации, пользователи сертификатов используют ключ, сертифицированный в сертификате категории a), как единую опору доверия как для сертификатов, так и для CRL, подписанных данным органом. В этом случае, если автоматически выданный сертификат категории b) должен быть использован для проверки подписей на CRL, в данном стандарте не определены средства для проверки подлинности такого сертификата.

Если автоматически выданные сертификаты категории b) встречаются в тракте, они должны игнорироваться.

ПРИМЕЧАНИЕ. – Другие механизмы для распределения открытых ключей CA находятся вне области применения данной спецификации Справочника.

8.2 Расширения информации о ключах и политике

8.2.1 Требования

Следующие требования относятся к информации о ключах и политике:

a) Обновление пары ключей CA может происходить через регулярные промежутки времени или при особых обстоятельствах. Поле сертификата должно передавать идентификатор открытого ключа, который должен использоваться для проверки подписи сертификата. Система, использующая сертификат, может использовать такие идентификаторы при нахождении правильного CA-сертификата для проверки подлинности открытого ключа органа, выдавшего сертификат.

b) В общем случае, у субъекта сертификата есть различные открытые ключи и соответственно различные сертификаты для различных целей, например, цифровая подпись и соглашение о шифровании ключа. Поле сертификата необходимо для помощи пользователю сертификата в выборе правильного сертификата для заданного субъекта для конкретного использования или для разрешения CA оговаривать, что сертифицированный ключ может только быть использован для конкретной цели.

c) Обновление пары ключей субъекта может происходить через регулярные промежутки времени или при особых обстоятельствах. Поле сертификата необходимо для передачи идентификатора для различения между разными открытыми ключами для одного и того же субъекта, используемыми в различные моменты времени. Система, использующая сертификат, может использовать такие идентификаторы для нахождения правильного сертификата.

d) Частный ключ, соответствующий сертифицированному открытому ключу, обычно используется в течение периода, отличного от периода действия открытого ключа. С ключами цифровой подписи, период использования для подписания частного ключа обычно короче периода использования проверяющего открытого ключа. Период действия сертификата указывает период, в течение которого открытый ключ может использоваться и который необязательно совпадает с периодом использования частного ключа. В случае фальсификации частного ключа, период воздействия может быть ограничен, если верификатор подписи знает законный период использования для частного ключа. Таким образом, существует требование к способности указывать период использования частного ключа в сертификате.

e) В связи с тем, что сертификаты могут использоваться в средах, где применимы несколько политик сертификатов, необходимо проводить приготовления ко включению в сертификаты информации о политике сертификатов.

f) В случае перекрестной сертификации от одной организации к другой, иногда может быть достигнуто соглашение, что определенная политика одной из организаций может считаться эквивалентной. CA-сертификат должен позволить органу, выдавшему сертификат, указать, что одна из его собственных политик сертификатов является эквивалентной другой политике сертификатов в области CA субъекта. Это известно как отображение политики.

g) Пользователь системы шифрования или цифровой подписи, использующий сертификаты, определенные в данной спецификации Справочника, должен обладать возможностью заранее определять алгоритмы, поддерживаемые другими пользователями.

8.2.2 Поля расширений сертификатов открытого ключа и CRL

Определены следующие поля расширений:

a) *идентификатор ключа органа*;

b) *идентификатор ключа субъекта*;

c) *использование ключа*;

d) *расширенное использование ключа*;

e) *период использования частного ключа*;

f) *политики сертификатов*;

g) *отображения политики*.

Данные поля расширений могут использоваться только в качестве расширений сертификатов, исключая идентификатора ключа органа, который может также использоваться в качестве расширения CRL. Если не указано иное, эти расширения могут использоваться как в CA-сертификатах, так и в сертификатах оконечных объектов.

8.2.2.1 Расширение идентификатора ключа органа

Данное поле, которое может использоваться как расширение сертификата или расширение CRL, определяет открытый ключ, который должен использоваться для проверки подписи на данном сертификате или CRL. Оно позволяет различать разные ключи, используемые одним и тем же CA (например, когда происходит обновление ключа). Это поле определяется следующим образом:

```
authorityKeyIdentifier EXTENSION ::= {
    SYNTAX              AuthorityKeyIdentifier
    IDENTIFIED BY       id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier               OPTIONAL,
    authorityCertIssuer         [1] GeneralNames                OPTIONAL,
    authorityCertSerialNumber   [2] CertificateSerialNumber     OPTIONAL }
    ( WITH COMPONENTS           {..., authorityCertIssuer PRESENT,
                                authorityCertSerialNumber PRESENT} |
     WITH COMPONENTS            {..., authorityCertIssuer ABSENT,
                                authorityCertSerialNumber ABSENT} )

KeyIdentifier ::= OCTET STRING
```

Ключ может определяться при помощи явного идентификатора ключа в компоненте **keyIdentifier**, определения сертификата для ключа (предоставляя орган, выдавший сертификат в компоненте **keyIdentifier** и порядковый номер сертификата в компоненте **authorityCertSerialNumber**) или явного идентификатора ключа и определения сертификата для ключа одновременно. Если используются обе формы идентификации, то орган, выдавший сертификат или CRL, должен обеспечивать их совместимость. Идентификатор ключа должен быть уникальным относительно всех идентификаторов ключей для выдавшего органа для сертификата или CRL, содержащего расширение. Реализация, поддерживающая расширение, не обязательно должна быть способна обработать все формы имени в компоненте **authorityCertIssuer**. (Подробная информация о типе **GeneralNames** содержится в п. 8.3.2.1.)

Органы по сертификации должны присваивать порядковые номера сертификатам таким образом, чтобы любая пара (выдавший орган, порядковый номер сертификата) однозначным образом определяла единственный сертификат. Форма **keyIdentifier** может использоваться для выбора CA-сертификатов в течение создания тракта. Пара **authorityCertIssuer**, **authoritySerialNumber** может использоваться только для предоставления привилегии одному сертификату над другими в течение создания тракта.

Данное расширение всегда является некритическим.

8.2.2.2 Расширение идентификатора ключа субъекта

Данное поле определяет сертифицируемый открытый ключ. Оно дает возможность различать разные ключи, используемые одним и тем же субъектом (например, когда происходит обновление ключа). Это поле определяется следующим образом:

```
subjectKeyIdentifier EXTENSION ::= {
    SYNTAX              SubjectKeyIdentifier
    IDENTIFIED BY       id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier
```

Идентификатор ключа должен быть уникальным относительно всех идентификаторов ключей для субъекта, с которым он используется. Данное расширение всегда является некритическим.

8.2.2.3 Расширение использования ключа

Данное поле определяет предполагаемое использование, для которого был выдан сертификат. Предполагаемое использование может быть в дальнейшем ограничено политикой. Политика может быть установлена в определении политики сертификата, контракте или другой спецификации. Тем не менее, политика не должна превышать ограничения, указанные битом **KeyUsage**, т. е. политика сертификатов не может разрешить сертификату быть использованным для цифровой подписи, если **KeyUsage** указывает, что он может быть использован только для соглашения о ключах.

Установка конкретного значения **KeyUsage** в сертификате не сигнализирует само по себе, для примера соединения, что взаимодействующие стороны действуют в соответствии с этой установкой, например, при подписании документа. Определение методов, которыми стороны могут сигнализировать о своих намерениях для конкретного примера соединения (например, фиксация содержимого для данного конкретного примера), находится вне области применения данной спецификации Справочника, но допускается существование многих методов. Хотя и не рекомендуется, но является возможным использовать содержимое сертификата, например, политики сертификата, для сигнализации о намерениях подписания. Тем не менее, так как сигнализация производится при выдаче сертификата CA, такое применение может не удовлетворить требованию о том, что заявление о намерениях должно производиться в момент подписания подписавшим лицом.

В примере расширения **keyUsage** можно установить более чем один бит. Установка нескольких битов не должно изменить значение каждого отдельного бита, но должно указывать, что сертификат может быть использован для всех целей, указанных множеством битов. Установка нескольких битов может подвергаться рискам. Обзор данных рисков приведен в Приложении I.

Это поле определяется следующим образом:

```
keyUsage EXTENSION ::= {
    SYNTAX              KeyUsage
    IDENTIFIED BY       id-ce-keyUsage }

KeyUsage ::= BIT STRING {
    digitalSignature        (0),
    contentCommitment       (1),
    keyEncipherment         (2),
    dataEncipherment        (3),
    keyAgreement            (4),
    keyCertSign             (5),
    cRLSign                 (6),
    encipherOnly            (7),
    decipherOnly            (8) }
```

Биты в **KeyUsage** означают следующее:

a) **digitalSignature**: для проверки цифровых подписей, которые используются с услугой аутентификации объекта, услугой аутентификации источника данных и/или услугой целостности;

b) **contentCommitment**: для проверки цифровых подписей, цель которых состоит в сигнализации о том, что подписавшее лицо фиксируется в подписываемом содержимом. Тип фиксации, для поддержки которого может использоваться сертификат, может в дальнейшем быть ограничен CA, например, посредством политики сертификатов. Точный тип фиксации подписавшего лица, например, "рассмотрено и одобрено" или "с намерением быть связанным", может сигнализироваться посредством подписываемого содержимого, например, самого подписанного документа или дополнительной подписанной информации.

   Так как подписание фиксации содержимого рассматривается как операция, подписанная цифровой подписью, нет необходимости устанавливать в сертификате бит **digitalSignature**. Если он установлен, это не влияет на уровень фиксации, предоставленный подписавшим лицом в подписанном содержимом.

   Отметим, что является неправильным относить к данному биту **keyUsage** идентификатор **nonRepudiation**. Тем не менее, использование данного идентификатора было исключено. Несмотря на используемый идентификатор, семантика данного бита является такой же, как определенная в данной спецификации Справочника;

c) **keyEncipherment**: для шифрования ключей или другой информации безопасности, например, для транспортировки ключа;

d) **dataEncipherment**: для шифрования данных пользователя или другой информации безопасности, см. выше в c);

e) **keyAgreement**: для использования в качестве ключа соглашения об открытых ключах;

f) **keyCertSign**: для проверки подписи CA на сертификатах.

   Так как подписание сертификатов рассматривается как фиксация CA к содержимому сертификата, ни бит **digitalSignature**, ни бит **contentCommitment** не должны устанавливаться в данном сертификате. Если

какой-либо из них (или оба) установлены, это не влияет на уровень фиксации, предоставленный подписавшим лицом в подписанном сертификате;

g) **cRLSign**: для проверки подписи органа на CRL.

Так как подписание CRL рассматривается как фиксация органа, выдавшего CRL к содержимому CRL, ни бит **digitalSignature**, ни бит **contentCommitment** не должны устанавливаться в данном сертификате. Если какой-либо из них (или оба) установлены, это не влияет на уровень фиксации, предоставленный подписавшим лицом в подписанном CRL;

h) **encipherOnly**: ключ соглашения об открытых ключах для использования только для шифрования данных при совместном использовании с установленным битом **keyAgreement** (значение с другой установкой бита использования ключа не определено);

i) **decipherOnly**: ключ соглашения об открытых ключах для использования только для дешифрования данных при совместном использовании с установленным битом **keyAgreement** (значение с другой установкой бита использования ключа не определено).

Спецификации приложений должны указывать, какой из битов **digitalSignature** или **contentCommitment** является подходящим для их использования. Если подписывающее приложение не знает о намерениях подписавшего лица в отношении фиксации к содержимому, приложение должно подписать и поддерживать это подписание с сертификатом, у которого значение бита **digitalSignature** установлено в расширении **keyUsage** сертификата.

Даже несмотря на то, что цифровая подпись была проверена с использованием сертификата, в котором установлен только бит **digitalSignature**, другие факторы, внешние по отношению к проверке, также могут играть роль в определении намерения подписания. И наоборот, даже несмотря на то, что цифровая подпись была проверена с использованием сертификата, в котором установлен только бит **contentCommitment**, внешние факторы могут использоваться подписавшим лицом для отказа от фиксации к подписанному содержимому.

Бит **keyCertSign** предназначен для использования только в CA-сертификатах. Если **KeyUsage** установлен в **keyCertSign**, значение компонента cA в расширении **basicConstraints** должно быть установлено в **TRUE**. CA также могут использовать другие определенные биты использования ключа в **KeyUsage**, например, **digitalSignature** для предоставления аутентификации и целостности операций администрации в диалоговом режиме.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим.

Если расширение помечено как критическое или расширение помечено как некритическое, но система, использующая сертификат, узнает его, то сертификат должен использоваться только для цели, на которую установлен соответствующий бит использования ключа. Если расширение помечено как некритическое и система не узнает его, то данное расширение должно игнорироваться. Установка бита в нулевое значение указывает, что ключ не предназначен для данной цели. Если в расширении все биты установлены в нулевое значение, то данный ключ предназначен для какой-либо цели, отличной от перечисленных выше.

8.2.2.4 Расширенное расширение использования ключа

Данное поле указывает одну или несколько целей, для которых может использоваться сертифицированный открытый ключ, помимо или вместо основных целей, указанных в поле расширения использования ключа. Данное поле определяется следующим образом:

```
extKeyUsage EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF KeyPurposeId
    IDENTIFIED BY       id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER
```

CA может заявит об использовании-любого-расширенного-ключа при помощи идентификатора **anyExtendedKeyUsage**. Это дает CA возможность выдать сертификат, содержащий OID для расширенных использований ключа, которые могут потребоваться приложениям, использующим сертификаты, без ограничения сертификата только этими использованиями ключа. Если использование расширенного ключа ограничило использование ключа, то включение OID устраняет данное ограничение.

```
anyExtendedKeyUsage  OBJECT IDENTIFIER  ::= { 2 5 29 37 0 }[S9]
```

Цели ключа могут определяться любой организацией, у которой есть в этом необходимость. Идентификаторы объектов, используемые для определения целей ключа, должны быть присвоены в соответствии с Рек. МСЭ-Т X.660 | ИСО/МЭК 9834-1.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим.

Если расширение помечено как критическое, то сертификат должен использоваться только для одной из указанных целей.

Если расширение помечено как некритическое, то оно указывает на предполагаемую цель или цели ключа и может быть использовано при поиске правильного ключа/сертификата объекта, имеющего несколько ключей/сертификатов. Если расширение присутствует и система, использующая сертификат, узнает и обрабатывает тип расширения **extendedKeyUsage**, то система, использующая сертификат, должна обеспечивать, что сертификат должен быть использован только для одной из указанных целей. (Использование приложений может, тем не менее, потребовать, чтобы была указана конкретная цель, для того чтобы сертификат был приемлем для данного приложения.)

Если сертификат содержит как критическое поле использования ключа, так и критическое поле расширенного использования ключа, то оба поля должны обрабатываться независимо и сертификат должен быть использован только для цели, совместимой с обоими полями. Если цели, совместимой с обоими полями, нет, то сертификат не должен использоваться ни для одной цели.

В данной Спецификации определяется следующая цель ключа, которая может быть включена в расширение расширенного использования ключа. Другие цели, которые также могут быть включены, определены в других спецификациях, таких как IETF RFC 3280.

```
keyPurposes          OBJECT IDENTIFIER  ::=    {ds 38 1}
```

8.2.2.5 Расширение периода использования частного ключа

Данное поле указывает период использования частного ключа, соответствующий сертифицированному открытому ключу. Оно применимо только к ключам цифровой подписи. Данное поле определяется следующим образом:

```
privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX              PrivateKeyUsagePeriod
    IDENTIFIED BY       id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore   [0]   GeneralizedTime OPTIONAL,
    notAfter    [1]   GeneralizedTime OPTIONAL }
    ( WITH COMPONENTS {..., notBefore PRESENT} |
    WITH COMPONENTS       {..., notAfter PRESENT} )
```

Компонент **notBefore** указывает на самую раннюю дату и время, когда частный ключ может использоваться для подписания. Если компонент **notBefore** не присутствует, то не предоставляется информация о том, когда начинается период допустимого использования частного ключа. Компонент **notAfter** указывает на самую позднюю дату и время, когда частный ключ может использоваться для подписания. Если компонент **notAfter** не присутствует, то не предоставляется информация о том, когда заканчивается период допустимого использования частного ключа.

Данное расширение всегда является некритическим.

ПРИМЕЧАНИЕ 1. – Период допустимого использования частного ключа может отличаться от сертифицированного периода действия открытого ключа, как указано в периоде действия сертификата. Для ключей цифровых подписей, период использования для подписания частного ключа как правило короче, чем этот период для проверки открытого ключа.

ПРИМЕЧАНИЕ 2. – Если верификатор цифровой подписи хочет проверить, что сертификат не был аннулирован, например, по причине фальсификации ключа, до момента проверки, то действительный сертификат будет все еще существовать для открытого ключа во время проверки. По истечении периода действия сертификата(тов) для открытого ключа, верификатор подписи не может полагаться на фальсификации, уведомление о которых происходит через CRL.

8.2.2.6 Расширение политик сертификатов

В данном поле перечислены политики сертификатов, узнаваемые выдавшим CA, которые применяются к сертификату, вместе с информацией о дополнительных квалификаторах, имеющих отношение к данным политикам сертификатов. Список политик сертификатов используется при определении периода действия тракта сертификации, как описано в п. 10. Дополнительные квалификаторы не используются в процедуре обработки тракта сертификации, но соответствующие квалификаторы предоставляются как результат этой обработки приложению, использующему сертификат, для помощи при определении того, является ли действительный тракт подходящим для определенной операции. Как правило, различные политики сертификатов будут относиться к различным приложениям, которые могут использовать сертифицированный ключ. Наличие данного расширения в сертификате оконечного объекта указывает политики сертификатов, для которых данный сертификат является действительным. Наличие данного расширения в сертификате, выданном одним CA другому CA, указывает политики сертификатов, для которых тракты сертификации, содержащие данный сертификат, могут быть действительными. Данное поле определяется следующим образом:

```
certificatePolicies EXTENSION ::= {
    SYNTAX              CertificatePoliciesSyntax
    IDENTIFIED BY       id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier        CertPolicyId,
    policyQualifiers        SEQUENCE SIZE (1..MAX) OF
                                PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId       CERT-POLICY-QUALIFIER.&id
                            ({SupportedPolicyQualifiers}),
    qualifier               CERT-POLICY-QUALIFIER.&Qualifier
                            ({SupportedPolicyQualifiers}{@policyQualifierId})
                            OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }
```

Значение типа **PolicyInformation** определяет и передает информацию квалификаторов для одной политики сертификатов. Компонент **policyIdentifier** содержит идентификатор политики сертификатов, а компонент **policyQualifiers** содержит значения квалификаторов политики для данного элемента.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим.

Если расширение помечено как критическое, это указывает, что сертификат должен использоваться только для цели и в соответствии с правилами, подразумеваемыми одной из указанных политик сертификатов. Правила определенной политики могут требовать, чтобы система, использующая сертификат, обрабатывала значение квалификатора определенным способом.

Если расширение помечено как некритическое, использование этого расширения не обязательно ограничивает использование сертификата перечисленными политиками. Тем не менее, пользователь сертификата может потребовать, чтобы присутствовала определенная политика, для того чтобы использовать сертификат (см. п. 10). Квалификаторы политики могут, по усмотрению пользователя сертификата, обрабатываться или игнорироваться.

Политики сертификатов и типы квалификаторов политик сертификатов могут определяться любой организацией, у которой есть в этом необходимость. Идентификаторы объектов, используемые для определения политик сертификатов и типов квалификаторов политик сертификатов, должны быть присвоены в соответствии с Рек. МСЭ-Т X.660 | ИСО/МЭК 9834-1. СА может заявить любую-политику путем использования идентификатора **anyPolicy**, для того чтобы доверять сертификату для всех возможных политик. Из-за необходимости в определении этого специального значения для применения вне зависимости от приложения или среды, данный идентификатор объекта присваивается в данной Спецификации. В данной Спецификации не будут присваиваться идентификаторы для конкретных политик сертификатов. Данное присвоение является обязанностью объекта, который определяет политику сертификатов.

```
anyPolicy       OBJECT IDENTIFIER  ::=          { 2 5 29 32 0 }
```

Идентификатор **anyPolicy** не должен иметь связанных с ним квалификаторов политики.

Следующий класс объектов ASN.1 используется при определении типов квалификаторов политик сертификатов:

```
CERT-POLICY-QUALIFIER ::= CLASS {
    &id                 OBJECT IDENTIFIER UNIQUE,
    &Qualifier          OPTIONAL }
WITH SYNTAX {
    POLICY-QUALIFIER-ID     &id
    [QUALIFIER-TYPE     &Qualifier] }
```

Определение типов квалификаторов политик сертификатов должно включать:

- заявление семантики возможных значений; и
- указание того, может ли идентификатор квалификатора появиться в расширении политик сертификатов без сопровождающего значения, и если да, то подразумеваемая в таком случае семантика.

ПРИМЕЧАНИЕ. – Квалификатор может быть определен как имеющий любой тип ASN.1. Если ожидается использование квалификатора сперва с приложениями, не имеющими функции декодирования ASN.1, рекомендуется, чтобы был определен тип **OCTET STRING**. Значение **OCTET STRING** ASN.1 может в таком случае передать значение квалификатора, кодированное в соответствии с любым соглашением, определенным организацией, определяющей элементы политики.

8.2.2.7 Расширение отображений политики

Данное поле, которое должно использоваться только в СА-сертификатах, позволяет выдавшему сертификат указать, что для целей пользователя тракта сертификации, содержащего данный сертификат, одна из политик сертификатов выдавшего органа может считаться эквивалентной другой политике сертификатов, используемой в области СА субъекта. Данное поле определяется следующим образом:

```
policyMappings EXTENSION ::= {
    SYNTAX              PolicyMappingsSyntax
    IDENTIFIED BY       id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy      CertPolicyId,
    subjectDomainPolicy     CertPolicyId }
```

Компонент **issuerDomainPolicy** указывает на политику сертификатов, которую он узнает в области выдающего СА и которая может считаться эквивалентной политике сертификатов, указанной в компоненте **subjectDomainPolicy**, которая узнается в области СА субъекта.

Политики не должны отображаться из специального значения **anyPolicy** или в специальное значение **anyPolicy**.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Рекомендуется, чтобы оно было критическим, иначе пользователь сертификата может неправильно интерпретировать условие выдавшего СА.

ПРИМЕЧАНИЕ 1. – Примером отображения политик может служить следующий Область правительства США может иметь политику, называемую Торговля с Канадой, а Канадское правительство может иметь политику, называемую Торговля с США. Хотя обе эти политики являются четко идентифицированными и определенными, между двумя правительствами может существовать соглашение о принятии трактов сертификации, расширяющих границы соприкосновения в пределах правил, подразумеваемых этими политиками для соответствующих целей.

ПРИМЕЧАНИЕ 2. – Отображение политик подразумевает значительные административные накладные расходы и вовлечение соответственно подготовленного и уполномоченного персонала для принятия решений. В общем случае, предпочтительным является соглашение о более глобальном использовании общих политик, чем применять отображение политик. В приведенном выше примере, для США, Канады и Мексики было бы предпочтительнее договориться об общей политике для Северо-Американской торговли.

ПРИМЕЧАНИЕ 3. – Ожидается, что отображение политик будет иметь практический смысл только в ограниченных средах, в которых положения политик являются очень простыми.

8.3 Расширения информации о субъекте и выдавшем органе

8.3.1 Требования

Следующие требования относятся к атрибутам субъекта сертификата и органа, выдавшего сертификат:

a) Сертификаты должны иметь возможность использоваться приложениями, которые используют разнообразные формы имен, включая имена электронной почты Интернета, имена доменов Интернета, адреса источника/приемника X.400 и имена сторон EDI. Поэтому необходимо обеспечить возможность конфиденциального связывания множества имен разнообразных форм имен с субъектом сертификата или органом, выдавший сертификат или CRL.

b) Пользователю сертификата может потребоваться конфиденциально узнать некоторую идентификационную информацию о субъекте, для того чтобы иметь уверенность в том, что субъект на самом деле является подразумеваемой личностью или вещью. Например, может потребоваться такая информация, как почтовый адрес, должность в корпорации или изображение картинки. Такая информация может быть удобно представлена в виде атрибутов справочника, но эти атрибуты необязательно являются частью выделенного имени. Таким образом, поле сертификата требуется для передачи дополнительных атрибутов справочника помимо присутствующих в выделенном имени.

8.3.2 Поля расширения сертификатов и CRL

Определяются следующие поля расширений:

a) *Альтернативное имя субъекта*;

b) *Альтернативное имя выдавшего органа*;

c) *Атрибуты справочника субъекта*.

Данные поля должны использоваться только как расширения сертификатов, за исключением альтернативного имени выдавшего органа, которое также может использоваться в качестве расширения CRL. В качестве расширений сертификата, они могут присутствовать в CA-сертификатах или сертификатах оконечных объектов.

8.3.2.1 Расширение альтернативного имени субъекта

Данное поле содержит одно или несколько альтернативных имен с использованием любой из разнообразных форм имен для объекта, который связан CA с сертифицированным открытым ключом. Данное поле определяется следующим образом:

```
subjectAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0]     INSTANCE OF OTHER-NAME,
    rfc822Name                  [1]     IA5String,
    dNSName                     [2]     IA5String,
    x400Address                 [3]     ORAddress,
    directoryName               [4]     Name,
    ediPartyName                [5]     EDIPartyName,
    uniformResourceIdentifier   [6]     IA5String,
    iPAddress                   [7]     OCTET STRING,
    registeredID                [8]     OBJECT IDENTIFIER }

OTHER-NAME ::= TYPE-IDENTIFIER

EDIPartyName ::= SEQUENCE {
    nameAssigner    [0]     DirectoryString {ub-name} OPTIONAL,
    partyName       [1]     DirectoryString {ub-name} }
```

Значения альтернатив типа **GeneralName** представляют собой имена различных форм, как например:

– **otherName** является именем любой формы, определенной как экземпляр класса информационного объекта **OTHER-NAME**;

– **rfc822Name** является адресом электронной почты интернета, определенным в соответствии с Internet RFC 822;

– **dNSName** является именем домена интернета, определенным в соответствии с Internet RFC 1035;

– **x400Address** является адресом O/R, определенным в соответствии с Рек. МСЭ-Т X.411 | ИСО/МЭК 10021-4;

– **directoryName** является именем Справочника, определенным в соответствии с Рек. МСЭ-Т X. 501 | ИСО/МЭК 9594-2;

- **ediPartyName** является именем формы, согласованной взаимодействующими партнерами электронного обмена данными; компонент **nameAssigner** определяет орган, который присваивает уникальные значения имен в компоненте **partyName**;
- **uniformResourceIdentifier** является универсальным идентификатором ресурса для Всемирной паутины, определенным в соответствии с Internet RFC 1630;
- **iPAddress** является адресом протокола Интернет, определенным в соответствии с Internet RFC 791, представленным в виде бинарной строки;
- **registeredID** является идентификатором любого зарегистрированного объекта, присвоенным в соответствии с Рек. МСЭ-Т X.660 | ISO/IEC 9834-1.

Для любой формы имени, используемой в типе **GeneralName**, должна существовать система регистрации имен, которая обеспечивает, что любое имя однозначно определяет один объект как для органа, выдавшего сертификат, так и для пользователей сертификата.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Реализация, поддерживающая данное расширение, не обязательно должна быть способной обрабатывать все формы имен. Если расширение помечено как критическое, по меньшей мере одна из присутствующих форм имен должна быть узнана и обработана, иначе сертификат должен считаться недействительным. Независимо от указанного выше ограничения, системе, использующей сертификат, разрешается игнорировать любое имя с неузнаваемой или неподдерживаемой формой имени. Рекомендуется, чтобы при условии, что поле субъекта сертификата содержит имя справочника, которое однозначно определяет субъект, данное поле было помечено как некритическое.

ПРИМЕЧАНИЕ 1. – Использование класса **TYPE-IDENTIFIER** описано в Приложениях A и C Рек. МСЭ-Т X.681 | ИСО/МЭК 8824-2.

ПРИМЕЧАНИЕ 2. – Если данное поле расширения присутствует и помечено как критическое, то поле **subject** сертификата может содержать нулевое имя (например, последовательность нулевых относительных выделенных имен), в этом случае субъект определяется только именем или именами в данном расширении.

8.3.2.2 Расширение альтернативного имени выдавшего органа

Данное поле содержит одно или несколько альтернативных имен с использованием любой из разнообразных форм имен для органа, выдавшего сертификат или CRL. Данное поле определяется следующим образом:

```
issuerAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-issuerAltName }
```

Данное расширение может по усмотрению органа, выдавшего сертификат или CRL, быть либо критическим, либо некритическим. Реализация, поддерживающая данное расширение, не обязательно должна быть способной обрабатывать все формы имен. Если расширение помечено как критическое, по меньшей мере одна из присутствующих форм имен должна быть узнана и обработана, иначе сертификат или CRL должен считаться недействительным. Независимо от указанного выше ограничения, системе, использующей сертификат, разрешается игнорировать любое имя с неузнаваемой или неподдерживаемой формой имени. Рекомендуется, чтобы при условии, что поле выдавшего сертификат или CRL органа содержит имя справочника, которое однозначно определяет выдавший орган, данное поле было помечено как некритическое.

ПРИМЕЧАНИЕ. – Если данное поле расширения присутствует и помечено как критическое, то поле **issuer** сертификата или CRL может содержать нулевое имя (например, последовательность нулевых относительных выделенных имен), в этом случае выдавший орган определяется только именем или именами в данном расширении.

8.3.2.3 Расширение атрибутов справочника субъекта

Данное поле передает любые желательные значения атрибутов справочника субъекта сертификата. Данное поле определяется следующим образом:

```
subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX              AttributesSyntax
    IDENTIFIED BY       id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute
```

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Система, использующая сертификат, обрабатывающая данное расширение, не обязательно должна понимать все типы атрибутов, включенные в расширение. Если расширение помечено как критическое, по меньшей мере один из типов атрибутов, содержащихся в расширении, должен быть понят для того, чтобы сертификат был принят. Если расширение помечено как критическое и ни один из содержащихся типов атрибутов не может быть понят, сертификат должен быть отклонен.

Если данное расширение присутствует в сертификате открытого ключа, то могут также присутствовать некоторые из расширений, определенных в п. 15.

8.4 Расширения ограничения тракта сертификации

8.4.1 Требования

Для обработки тракта сертификации:

a) Сертификаты оконечных объектов должны быть отличимыми от СА-сертификатов для защиты от возможности самостоятельной установки оконечных объектов в СА без авторизации. Также для СА необходимо иметь возможность ограничивать длину последовательной цепочки, получаемой из сертифицированного субъекта СА, например, не более чем один дополнительный сертификат или не более чем два дополнительных сертификата.

b) СА должен иметь возможность определять ограничения, которые позволяют пользователю сертификата проверить, что менее доверенные СА в тракте сертификации (например, СА далее вниз по тракту сертификации, начиная с СА, с открытого ключа которого начинает пользователь сертификата) не нарушают доверие, выдавая сертификаты субъектам в несоответствующем пространстве имен. Строгое соблюдение этих ограничений должно быть автоматически проверяемо пользователем сертификата.

c) Обработка тракта сертификации должна быть реализуема в автоматическом автономном модуле. Это необходимо, чтобы разрешить реализацию доверенных аппаратных и программных модулей, выполняющих функции обработки тракта сертификации.

d) Должно быть возможно реализовывать обработку тракта сертификации независимо от взаимодействия с локальным пользователем в режиме реального времени.

e) Должно быть возможно реализовывать обработку тракта сертификации независимо от использования доверенных локальных баз данных с информацией об описании политики. (Некоторая доверенная локальная информация – первоначальный открытый ключ, по меньшей мере – необходима для обработки тракта сертификации, но количество такой информации должно быть минимальным.)

f) Тракты сертификации должны работать в средах, где узнаются несколько политик сертификатов. СА должен иметь возможность оговаривать, каким СА в других областях он доверяет и в каких целях. Должно поддерживаться сцепление по нескольким областям политик.

g) В моделях доверия требуется полная гибкость. Строгая иерархическая модель, подходящая для одной организации, не подходит при рассмотрении потребностей нескольких взаимосвязанных предприятий. Гибкость требуется в выборе первого доверенного СА в тракте сертификации. В частности, должно быть возможно требовать, чтобы тракт сертификации начинался в локальной области безопасности системы пользователя открытого ключа.

h) Структуры присвоения имен не должны быть ограничены необходимостью использовать имена в сертификатах, т. е. структуры имен Справочника, считающиеся естественными для организаций или географических зон, не должны требовать урегулирования с целью соответствия требованиям органа по сертификации.

i) Поле расширения сертификата должно быть обратно-совместимо с системой, использующей неограниченный подход к тракту сертификации, как определено в ранних изданиях Рек. МСЭ-Т X.509 | ИСО/МЭК 9594-8.

j) СА должен иметь возможность запрещать использование отображения политики и требовать, чтобы в последующих сертификатах в тракте сертификации присутствовали идентификаторы явных политик сертификатов.

   ПРИМЕЧАНИЕ. – В любой системе, использующей сертификаты, обработка тракта сертификации требует соответствующего уровня гарантии. В данной Спецификации Справочника определяются функции, которые могут использоваться в реализациях, от которых требуется соответствие конкретным положениям о гарантиях. Например, требование о гарантии может устанавливать, что обработка тракта сертификации должна быть защищена от подверсий процесса (таких, как фальсификация программного обеспечения или модификации данных). Уровень гарантий должен быть соизмерим с коммерческим риском. Например:

   – обработка, внутренняя по отношению к соответствующему криптографическому модулю, может потребоваться для открытых ключей, используемых при проверке подлинности передачи средств в больших объемах, где

   – обработка в программном обеспечении может быть подходящей для запросов банковского баланса из дома.

   Следовательно, функции обработки тракта сертификации должны соответствовать реализациям в аппаратных криптографических модулях или криптографических маркерах, как одного из вариантов.

k) СА должен иметь возможность препятствовать рассмотрению специального значения любая-политика в качестве действительного в последующих сертификатах в тракте сертификации.

8.4.2 Поля расширения сертификата

Определяются следующие поля расширений:

a) *основные ограничения*;

b) *ограничения имен*;

c) *ограничения политик*;

d) *запретить любую политику*.

Данные поля расширений должны использоваться только в качестве расширений сертификатов. Ограничения имен и ограничения политик должны использоваться только в СА-сертификатах, основные ограничения могут также использоваться в сертификатах оконечных объектов. Примеры использования данных расширений приведены в Приложении G.

8.4.2.1 Расширение основных ограничений

Данное поле указывает, может ли субъект действовать как CA с использованием сертифицированного открытого ключа для проверки подписей сертификата. Если это так, то может быть также определено ограничение длины тракта сертификации. Данное поле определяется следующим образом:

```
basicConstraints EXTENSION ::= {
    SYNTAX              BasicConstraintsSyntax
    IDENTIFIED BY       id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
    cA                      BOOLEAN DEFAULT FALSE,
    pathLenConstraint       INTEGER (0..MAX) OPTIONAL }
```

Компонент **cA** указывает, может ли сертифицированный открытый ключ использоваться для проверки подписей сертификата.

Компонент **pathLenConstraint** должен присутствовать, только если **cA** установлен в истинное значение. Он задает максимальное количество CA-сертификатов, которые могут следовать за этим сертификатом в тракте сертификации. Значение 0 указывает, что субъект данного сертификата может выдавать сертификаты только оконечным объектам, а не дальнейшим CA. Если поле **pathLenConstraint** не встречается ни в одном сертификате в тракте сертификации, ограничения на допустимую длину тракта сертификации нет. Ограничение начинает действовать, начиная со следующего сертификата в тракте. Ограничение ограничивает длину сегмента тракта сертификации между сертификатом, содержащим это расширение, и сертификатом оконечного объекта. Оно не оказывает влияния на количество CA-сертификатов в тракте сертификации между опорой доверия и сертификатом, содержащим данное расширение. Поэтому длина полного тракта сертификации может превышать максимальную длину сегмента, ограниченную данным расширением. Ограничение контролирует количество не автоматически выданных CA сертификатов между сертификатом CA, содержащим ограничение, и сертификатом оконечного объекта. Поэтому общая длина данного сегмента тракта, исключая автоматически выданные сертификаты, может превышать значение ограничения на два сертификата. (Сюда включены сертификаты на двух оконечных точках сегмента, а также сертификаты CA между двумя оконечными точками, ограниченными значением данного расширения.)

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Рекомендуется, чтобы оно было помечено как критическое, иначе объект, неавторизованный как CA, может выдавать сертификаты и система, использующая сертификаты, может непреднамеренно использовать такой сертификат.

Если данное расширение присутствует и помечено как критическое или помечено как некритическое, но узнается системой, использующей сертификаты, то:

- если значение **cA** не установлено в истинное, то сертифицированный открытый ключ не должен использоваться для проверки подписи сертификата;
- если значение **cA** установлено в истинное и присутствует **pathLenConstraint**, то система, использующая сертификат, должна проверить, что обрабатываемый тракт сертификации совместим со значением **pathLenConstraint**.

ПРИМЕЧАНИЕ 1. – Если данное расширение не присутствует или помечено как некритическое и не узнается системой, использующей сертификаты, то сертификат должен считаться сертификатом оконечного объекта и не может использоваться для проверки подписей сертификатов.

ПРИМЕЧАНИЕ 2. – Для того чтобы ограничить субъект сертификата только оконечным объектом, т. е. не CA, выдавший орган может включить данное поле расширения, содержащее только пустое значение **SEQUENCE**.

8.4.2.2 Расширение ограничений имен

Данное поле, которое должно использоваться только в CA-сертификате, указывает пространство имен, в пределах которого должны находиться все имена субъектов в последующих сертификатах в тракте сертификации. Данное поле определяется следующим образом:

```
nameConstraints EXTENSION ::= {
    SYNTAX          NameConstraintsSyntax
    IDENTIFIED BY       id-ce-nameConstraint }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees       [0]   GeneralSubtrees OPTIONAL,
    excludedSubtrees        [1]   GeneralSubtrees OPTIONAL,
    requiredNameForms       [2]   NameForms OPTIONAL  }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base                    GeneralName,
    minimum  [0]   BaseDistance DEFAULT 0,
    maximum  [1]   BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)

NameForms  ::= SEQUENCE {
   basicNameForms    [0]   BasicNameForms OPTIONAL,
```

```
    otherNameForms     [1]   SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL }
(ALL EXCEPT ({ -- никакой; т. е. должен присутствовать по меньшей мере один компонент -- }))

BasicNameForms  ::= BIT STRING {
    rfc822Name                (0),
    dNSName                   (1),
    x400Address               (2),
    directoryName             (3),
    ediPartyName              (4),
    uniformResourceIdentifier (5),
    iPAddress                 (6),
    registeredID              (7) }  (SIZE (1..MAX))
```

При наличии, каждый из компонентов **permittedSubtrees** и **excludedSubtrees** определяют одно или несколько поддеревьев присвоения имен, каждое из которых определяется именем корня поддерева и, необязательно, в поддереве, областью, ограниченной верхним и нижнем уровнями. Если присутствует компонент **permittedSubtrees**, имена субъектов в пределах этих поддеревьев являются приемлемыми. Если присутствует компонент **excludedSubtrees**, любой сертификат, выданный CA субъекта или последующими CA в тракте сертификации, имеющий имя субъекта в пределах этих поддеревьев, является неприемлемым. Если присутствуют оба компонента **permittedSubtrees** и **excludedSubtrees**, и пространства имен перекрываются, то положение об исключении имеет преимущество для перекрывающихся имен. Если для формы имени не определены ни разрешенные, ни исключенные поддеревья, то любое имя в пределах данной формы имени является приемлемым. Если присутствует **requiredNameForms**, то все последующие сертификаты в тракте сертификации должны включать имя по меньшей мере в одной из требуемых форм.

Если присутствует **permittedSubtrees**, к последующим сертификатам в тракте сертификации применяется следующее. Если какой-либо сертификат содержит имя субъекта (в поле **subject** или расширении **subjectAltNames**) формы имени, для которой определены разрешенные поддеревья, то имя должно попадать по меньшей мере в одно из определенных поддеревьев. Если какой-либо сертификат содержит только имена субъектов форм имен, отличных от тех, для которых определены разрешенные поддеревья, то имена субъектов не должны попадать в какое-либо из определенных поддеревьев. Например, предположим, что определены два разрешенных поддерева, одно для формы имени DN и одно для формы имени rfc822, не определены исключенные поддеревья, но определены **requiredNameForms** с присутствующими битами **directoryName** и **rfc822Name**. Сертификат, который содержал бы только имена, отличные от имени справочника или имени rfc822, был бы неприемлемым. Если бы requiredNameForms не были определены, тем не менее, такой сертификат был бы приемлемым. Например, предположим, что определены два разрешенных поддерева, одно для формы имени DN и одно для формы имени rfc822, не определены исключенные поддеревья, и не присутствует **requiredNameForms**. Сертификат, который содержал бы только DN, находящийся в пределах определенного разрешенного поддерева, был бы приемлемым. Сертификат, который содержал бы как DN, так и имя rfc822, и в котором только одно из них находилось бы в пределах своего определенного разрешенного поддерева, был бы неприемлемым. Сертификат, который содержал бы только имена, отличные от DN и имени rfc822, также был бы приемлемым.

ПРИМЕЧАНИЕ. – Данный пример приведен исключительно в целях иллюстрации. Как управлять именами в формах имен типа **GeneralName**, исключая форму имени **directoryName**, в их иерархической структуре, не определяется в данной Рекомендации | Международном стандарте.

Если присутствует **excludedSubtrees**, любой сертификат, выданный CA субъекта или последующими CA в тракте сертификации, имеющий имя субъекта (в поле **subject** или расширении **subjectAltNames**) в пределах данных поддеревьев, является неприемлемым. Например, предположим, что определены два исключенных поддерева, одно для формы имени DN и одно для формы имени rfc822. Сертификат, который содержал бы только DN в пределах определенного исключенного поддерева, был бы неприемлемым. Сертификат, который содержал бы как DN, так и имя rfc822 и в котором по меньшей мере из них находилось бы в пределах своего определенного исключенного поддерева, был бы неприемлемым.

Когда субъект сертификата имеет несколько имен одной формы (включая в случае формы имени **directoryName** имя в поле субъекта сертификата, если не нулевое), тогда все подобные имена должны быть проверены на совместимость с ограничением имени данной формы имени.

Если присутствует **requiredNameForms**, все последующие сертификаты в тракте сертификации должны включать имя субъекта по крайней мере одной из требуемых для имени форм.

Среди форм имен, доступных через тип **GeneralName**, в полях **permittedSubtrees** и **excludedSubtrees** могут использоваться только те формы имен, которые имеют хорошо определенную иерархическую структуру. Форма имени **directoryName** удовлетворяет данному требованию при использовании данной формы имени поддерево, присваивающее имена, соответствует поддереву DIT.

Поле **minimum** определяет верхнюю границу области в поддереве. Все имена, у которых компонент конечного имени находится выше определенного уровня, не содержатся в данной области. Значение **minimum**, равное нулю (по умолчанию), соответствует основе, т. е. верхнему узлу поддерева. Например, если **minimum** установлено в 1, то поддерево, присваивающее имена, исключает основной узел, но включает подчиненные узлы.

Поле **maximum** определяет нижнюю границу области в поддереве. Все имена, у которых компонент конечного имени находится ниже определенного уровня, не содержатся в данной области. Значение **maximum**, равное нулю (по умолчанию), соответствует основе, т. е. верхнему узлу поддерева. Отсутствие компонента **maximum** указывает, что для

области в поддереве не должна устанавливаться нижняя граница. Например, если **maximum** установлено в 1, то поддерево, присваивающее имена, исключает все узлы, кроме основного узла и его непосредственных подчиненных.

Для формы имени **directoryName**, **certificate** считается подчиненным **base** (и поэтому кандидатом быть в пределах поддерева), если **SEQUENCE RDN**, которая формирует полный **DN** в **base**, является идентичной начальному **SEQUENCE** того же самого номера **RDN**, который формирует первую часть **DN** в поле **subject certificate**. **DN** в поле **subject certificate** может иметь дополнительные замыкающие **RDN** в своей последовательности, который не встречается в **DN base**. Правило соответствия **distinguishedNameMatch** используется для сравнения значения **base** с начальной последовательностью **RDN** в **DN** в поле **subject** сертификата.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Рекомендуется, чтобы оно было помечено как критическое, иначе пользователь сертификата может не проверить, что последующие сертификаты в тракте сертификации расположены в пространстве имен, намеченном выдающим CA.

Совместимые реализации не должны узнавать все возможные формы имени.

Если расширение присутствует и помечено как критическое, реализация, использующая сертификаты, должна узнать и обработать все формы имени, для которых существуют как спецификация поддерева (разрешенная или исключенная) в расширении, так и соответствующее значение в поле **subject** или расширении **subjectAltNames** любого последующего сертификата в тракте сертификации. Если как в спецификации поддерева, так и в последующем сертификате встречается неузнанная форма имени, данный сертификат должен управляться, как если бы встретилось неузнанное критическое расширение. Если какое-либо имя субъекта в сертификате попадает в исключенное поддерево, сертификат является неприемлемым. Если поддерево определено для формы имени, которая не содержится ни в одном последующем сертификате, данное поддерево может игнорироваться. Если компонент **requiredNameForms** определяет только неузнанные формы имени, сертификат должен управляться, как если бы встретилось неузнанное критическое расширение. Иначе, по меньшей мере одно из неузнанных форм имен должно встретиться во всех последующих сертификатах в тракте.

Если расширение присутствует и помечено как некритическое, а реализация, использующая сертификат, не узнает форму имени, используемую в каком-либо компоненте **base**, тогда данная спецификация поддерева может игнорироваться. Если расширение помечено как некритическое и ни одна из форм имен, определенных компонентом **requiredNameForms**, не узнается реализацией, использующей сертификаты, то сертификат должен управляться как если бы компонент **requiredNameForms** отсутствовал.

Отметим, что в некоторых случаях может потребоваться, чтобы более, чем один сертификат был выдан CA другому CA, чтобы получить желаемые результаты при конфликте некоторых требований к ограничениям имен. Например, предположим, что у Корпорации Acme в США есть 20 филиалов.

Корпорация Widget хочет перекрестно сертифицировать центральный CA Корпорации Acme, но только хочет, чтобы сообщество Widget использовало сертификаты Acme для субъектов, которые удовлетворяют следующим критериям:

- Филиалы с 1 по 19 Корпорации Acme, все отделы приемлемы как субъект;
- Филиал 20 Корпорации Acme, все отделы неприемлемы как субъект, исключая субъект в Отделе закупок.

Этого можно достичь путем выдачи двух сертификатов следующим образом: первый сертификат должен иметь **permittedSubtrees** {base: C=US, O=Acme} и **excludedSubtrees** {base: C=US, O=Acme, OU=branch20}. Второй сертификат должен иметь **permittedSubtrees** {base: C=US, O=Acme, OU=branch20, OU=Purchasing}.

В Приложении G содержатся примеры использования расширения ограничения имени.

8.4.2.3 Расширение ограничений политики

Данное поле определяет ограничения, которые могут потребовать идентификации явной политики сертификатов или запретить отображение политик для оставшейся части тракта сертификации. Данное поле определяется следующим образом:

```
policyConstraints EXTENSION ::= {
    SYNTAX              PolicyConstraintsSyntax
    IDENTIFIED BY       id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
    requireExplicitPolicy       [0]   SkipCerts OPTIONAL,
    inhibitPolicyMapping        [1]   SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)
```

Если присутствует компонент **requireExplicitPolicy** и тракт сертификации включает сертификат, выданный назначенным CA, необходимо, чтобы все сертификаты в тракте содержали в расширении политик сертификата приемлемый идентификатор политики. Приемлемым идентификатором политики является идентификатор политики сертификатов, требуемый пользователем тракта сертификации, идентификатор политики, которая была заявлена как эквивалентная одной их данных политик при помощи отображения политик, или *любая-политика*. Назначенный CA является либо CA, выдавшим

сертификат, содержащий данное расширение (если значение **requireExplicitPolicy** равно 0), или CA, выдавшим последующий сертификат в тракте сертификации (как указывается ненулевым значением).

Если присутствует компонент **inhibitPolicyMapping**, он указывает, что отображение политик не запрещено во всех сертификатах, начиная с назначенного CA в тракте сертификации и до конца тракта сертификации. Назначенный CA является либо CA субъекта сертификата, содержащего данное расширение (если значение **inhibitPolicyMapping** равно 0), либо CA, являющимся субъектом последующего сертификата в тракте сертификации (как указывается ненулевым значением).

Значение типа **SkipCerts** указывает количество сертификатов в тракте сертификации, которое нужно пропустить до того, как ограничение вступит в действие.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Рекомендуется, чтобы оно было критическим, иначе пользователь сертификата может неправильно интерпретировать условие выдавшего CA.

8.4.2.4 Расширение запретить любую политику

Данное поле определяет ограничение, которое указывает, что любая-политика не рассматривается как явное соответствие для других политик сертификатов для всех не автоматически выданных сертификатов в тракте сертификации, начиная с назначенного CA. Назначенный CA является ибо CA субъекта сертификата, содержащего данное расширение (если значение **inhibitAnyPolicy** равно 0), либо CA, являющимся субъектом последующего сертификата в тракте сертификации (как указывается ненулевым значением).

```
inhibitAnyPolicy      EXTENSION   ::= {
    SYNTAX            SkipCerts
IDENTIFIED BY         id-ce-inhibitAnyPolicy }
```

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Рекомендуется, чтобы оно было критическим, иначе пользователь сертификата может неправильно интерпретировать условие выдавшего CA.

8.5 Основные расширения CRL

8.5.1 Требования

Следующие требования относятся к CRL:

a) Пользователи сертификата должны иметь возможность отслеживать все CRL, выданные выдающим CRL или точкой распределения CRL (см. п. 8.6), а также обнаруживать пропущенный CRL в последовательности. В связи с этим требуются порядковые номера CRL.

b) Некоторые пользователи CRL могут пожелать отвечать по-разному на аннулирование, в зависимости от причины аннулирования. Поэтому запись CRL должна указывать причину аннулирования.

c) Орган должен иметь возможность временно приостанавливать действие сертификата и впоследствии либо аннулировать, либо восстановить его. К возможным причинам для такого действия относятся:

   – желание уменьшить ответственность за ошибочные аннулирования, когда запрос на аннулирование не является аутентифицированным и нет адекватной информации, для того чтобы определить, является ли он действительным;

   – другие производственные потребности, такие как временная блокировка сертификата объекта при аудите или расследовании.

d) Для каждого аннулированного сертификата CRL содержит дату, когда орган выполнил аннулирование. Может быть получена дополнительная информация о том, когда произошла фактическая или подозреваемая фальсификация ключа, и данная информация может быть ценной для пользователя сертификата. Даты аннулирования недостаточно для разрешения некоторых вопросов, потому что в худшем случае все подписи, выданные в течение периода действия сертификата, должны считаться недействительными. Тем не менее, для пользователя может быть важно, чтобы подписанный документ был признан как действительный даже несмотря на то, что ключ, использованный для подписания сообщений, был фальсифицирован после произведения подписи. Чтобы помочь при решении данной проблемы, запись CRL может включать вторую дату, указывающую, когда стало известно или стало подозреваться, что частный ключ был фальсифицирован.

e) Пользователи сертификата должны иметь возможность определить непосредственно из CRL дополнительную информацию, включая границы сертификатов, на которые распространяется данный список, порядок уведомлений об аннулировании, а также поток CRL, в котором номер CRL является уникальным.

f) Выдавшим органам необходимо иметь возможность динамически изменять разбиение CRL и адресовать пользователей сертификатов к новому расположению для соответствующих CRL при изменении разбиения.

g) Также могут быть доступны дельта-CRL, которые обновляют заданный основной CRL. Пользователи сертификатов должны иметь возможность определить из заданного CRL, доступны ли дельта-CRL, где они расположены и когда будет выдан следующий дельта-CRL.

h) В дополнение к CRL, публикующим информацию о том, что сертификаты были аннулированы, необходимо публиковать уведомления о том, что сертификаты будут аннулированы на определенную дату и время в будущем.

i) Необходимо предоставлять более эффективные способы для указания в CRL, что ряд сертификатов был аннулирован.

8.5.2 Поля расширения CRL и записи CRL

Определяются следующие поля расширений:

a) *номер CRL*;
b) *код причины*;
c) *код инструкции удержания*;
d) *дата недействительности*;
e) *границы CRL*;
f) *передача статуса*;
g) *идентификатор потока CRL*;
h) *упорядоченный список*;
i) *информация дельта*.

Номер CRL, границы CRL, передача статуса, идентификатор потока CRL, упорядоченный список и информация дельта должны использоваться только как поля расширений CRL, а остальные поля должны использоваться только как поля расширений записей CRL.

8.5.2.1 Расширение номера CRL

Данное поле расширения CRL передает монотонно возрастающие порядковые номера для каждого CRL, выданного заданным выдающим CRL посредством заданного атрибута справочника или точки распределения CRL. Это позволяет пользователю CRL определить, были ли CRL, выданные до обработки одного, также замечены и обработаны. Данное поле определяется следующим образом:

```
cRLNumber EXTENSION ::= {
    SYNTAX              CRLNumber
    IDENTIFIED BY       id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)
```

Данное расширение всегда является некритическим.

8.5.2.2 Расширение кода причины

Данное поле расширения записи CRL определяет причину аннулирования сертификата. Код причины может использоваться приложениями при принятии решения о том, как реагировать на произошедшее аннулирование, основываясь на локальной политике. Данное поле определяется следующим образом:

```
reasonCode EXTENSION ::= {
    SYNTAX              CRLReason
    IDENTIFIED BY       id-ce-reasonCode }

CRLReason ::= ENUMERATED {
    unspecified             (0),
    keyCompromise           (1),
    cACompromise            (2),
    affiliationChanged      (3),
    superseded              (4),
    cessationOfOperation    (5),
    certificateHold         (6),
    removeFromCRL           (8),
    privilegeWithdrawn      (9),
    aaCompromise            (10) }
```

Следующие значения кода причины указывают, почему сертификат был аннулирован:

– **unspecified** может использоваться для аннулирования сертификатов по причинам, отличным от конкретных кодов;
– **keyCompromise** используется при аннулировании сертификата оконечного объекта; указывает, что известно или подозревается факт фальсификации частного ключа субъекта или других аспектов субъекта, проверенных в сертификате;
– **cACompromise** используется при аннулировании CA-сертификата; указывает, что известно или подозревается факт фальсификации частного ключа субъекта или других аспектов субъекта, проверенных в сертификате;
– **affiliationChanged** указывает, что имя субъекта или другая информация в сертификате была модифицирована, но нет причин подозревать, что частный ключ был фальсифицирован;

- **superseded** указывает, что сертификат был заменен, но нет причин подозревать, что частный ключ был фальсифицирован;
- **cessationOfOperation** указывает, что сертификат более не нужен для цели, для которой был выдан, но нет причин подозревать, что частный ключ был фальсифицирован;
- **privilegeWithdrawn** указывает, что сертификат (открытого ключа или сертификат атрибута) был аннулирован, потому что была отозвана привилегия, содержащаяся в данном сертификате;
- **aACompromise** указывает, что известно или подозревается факт фальсификации аспектов AA, проверенных в сертификате атрибута.

Сертификат может быть помещен на удержание путем выдачи записи CRL с кодом причины **certificateHold**. Уведомление об удержании сертификата может включать необязательный код инструкции удержания для передачи дополнительной информации для пользователей сертификата (см. п. 8.5.2.3). Если удержание было выдано, им можно управлять одним из трех способов:

a) оно может остаться в CRL без дополнительных действий, заставляя пользователей отклонять операции, выданные в течение периода удержания; или

b) его можно заменить (окончательным) аннулированием для того же сертификата, тогда причиной должна быть одна из стандартных причин аннулирования, датой аннулирования должна быть дата, когда сертификат был помещен на удержание, и необязательное поле расширения кода инструкции не должно встречаться; или

c) оно может быть явно освобождено, и запись удалена из CRL.

Код причины **removeFromCRL** используется только с дельта-CRL (см. п. 8.6) и указывает, что существующая запись CRL должна сейчас быть удалена по причине истечения периода действия сертификата или отмены удержания. Запись с таким кодом причины должна использоваться в дельта-CRL, для которого соответствующий основной CRL или любой последующий (дельта или полный для границ) CRL содержит запись для того же сертификата с кодом причины **certificateHold**.

Данное расширение всегда является некритическим.

8.5.2.3 Расширение кода инструкции удержания

Данное поле расширения записи CRL используется для включения идентификатора зарегистрированной инструкции для указания действия, которое должно быть предпринято при обнаружении удерживаемого сертификата. Оно применимо только к записи, имеющей код причины **certificateHold**. Данное поле определяется следующим образом:

```
holdInstructionCode EXTENSION ::= {
    SYNTAX              HoldInstruction
    IDENTIFIED BY       id-ce-instructionCode }

HoldInstruction ::= OBJECT IDENTIFIER
```

Данное расширение всегда является некритическим. В данной спецификации Справочника не определяется стандартные коды инструкций удержания.

ПРИМЕЧАНИЕ. – К примерам инструкций удержания относятся "пожалуйста, свяжитесь с CA" или "снова используйте маркер пользователя".

8.5.2.4 Расширение даты недействительности

Данное поле расширения записи CRL указывает дату, на которую стало известно или стало подозреваться, что частный ключ был фальсифицирован или что сертификат должен по другой причине считаться недействительным. Данная дата может предшествовать дате аннулирования в записи CRL, являющейся датой, на которую орган обработал аннулирования. Данное поле определяется следующим образом:

```
invalidityDate EXTENSION ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-invalidityDate }
```

Данное расширение всегда является некритическим.

ПРИМЕЧАНИЕ 1. – Дата в данном расширении сама по себе не является достаточной для целей фиксации авторства. Например, эта дата может быть датой, рекомендованной держателем частного ключа, и такое лицо может мошеннически заявить, что ключ был фальсифицирован некоторое время назад, чтобы отказаться от правомерно сгенерированной подписи.

ПРИМЕЧАНИЕ 2. – Когда аннулирование впервые заносится органом в CRL, дата недействительности может предшествовать дате выдачи более ранних CRL. Дата аннулирования не должна предшествовать дате выдачи более ранних CRL.

8.5.2.5 Расширение границ CRL

ПРИМЕЧАНИЕ. – Использование расширения границ CRL исключено.

Границы CRL указываются в данном CRL с использованием следующего расширения CRL. Чтобы избежать атаку замещения CRL против приложения, которое не поддерживает расширение границ, при наличии, расширение границ должно быть помечено как критическое.

Данное расширение может использоваться для предоставления положений об границах разнообразных типов CRL, включая:

- простые CRL, которые предоставляют информацию об аннулировании сертификатов, выданных одним органом;
- непрямые CRL, которые предоставляют информацию об аннулировании сертификатов, выданных несколькими органами;
- дельта-CRL, которые обновляют ранее выданную информацию об аннулировании;
- непрямые дельта-CRL, которые предоставляют информацию об аннулировании, которая обновляет несколько основных CRL, выданных одним или несколькими органами.

```
crlScope EXTENSION ::= {
    SYNTAX                  CRLScopeSyntax
    IDENTIFIED BY           id-ce-cRLScope }

CRLScopeSyntax ::=          SEQUENCE SIZE (1..MAX) OF PerAuthorityScope

PerAuthorityScope ::= SEQUENCE {
    authorityName               [0]     GeneralName OPTIONAL,
    distributionPoint           [1]     DistributionPointName OPTIONAL,
    onlyContains                [2]     OnlyCertificateTypes OPTIONAL,
    onlySomeReasons             [4]     ReasonFlags OPTIONAL,
    serialNumberRange           [5]     NumberRange OPTIONAL,
    subjectKeyIdRange           [6]     NumberRange OPTIONAL,
    nameSubtrees                [7]     GeneralNames OPTIONAL,
    baseRevocationInfo          [9]     BaseRevocationInfo OPTIONAL
    }

OnlyCertificateTypes ::= BIT STRING {
    user            (0),
    authority       (1),
    attribute       (2) }

NumberRange ::= SEQUENCE {
    startingNumber      [0]     INTEGER OPTIONAL,
    endingNumber        [1]     INTEGER OPTIONAL,
    modulus                     INTEGER OPTIONAL }

BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier     [0]     CRLStreamIdentifier     OPTIONAL,
    cRLNumber               [1]     CRLNumber,
    baseThisUpdate          [2]     GeneralizedTime }
```

Если CRL является непрямым CRL, который предоставляет информацию о статусе аннулирования для нескольких органов, то расширение будет включать несколько структур **PerAuthorityScope**, по одной или несколько для каждого органа, для которого включена информация об аннулировании. Каждый экземпляр **PerAuthorityScope**, который имеет отношение к органу, отличному от выдавшего данный CRL, должен содержать компонент **authorityName**. Если CRL является dCRL, который предоставляет информацию о статусе аннулирования для нескольких основных CRL, выданных одним органом, то расширение будет включать несколько структур **PerAuthorityScope**, по одной для каждого основного CRL, для которого данный dCRL предоставляет обновления. Даже несмотря на то, что будет несколько экземпляров структуры **PerAuthorityScope**, значение компонента **authorityName** при наличии должно быть одним и тем же для всех экземпляров.

Если CRL является непрямым dCRL, который предоставляет информацию о статусе аннулирования для нескольких основных CRL, выданных несколькими органами, то расширение будет включать несколько структур **PerAuthorityScope**, по одной для каждого основного CRL, для которого данный dCRL предоставляет обновления. Каждый экземпляр **PerAuthorityScope**, имеющий отношение к органу, отличному от выдавшего данный непрямой dCRL, должен включать компонент **authorityName**.

Для каждого компонента **PerAuthorityScope**, присутствующего в данном расширении, поля используются следующим образом. Отметим, что в случае непрямых CRL и непрямых dCRL, каждый экземпляр **PerAuthorityScope** может содержать различные сочетания данных полей и различных значений.

Поле **authorityName**, при наличии, определяет орган, выдавший сертификаты, для которых предоставляется информация об аннулировании. Если **authorityName** пропущен, он принимает как значение по умолчанию имя выдавшего CRL.

Поле **distributionPoint**, при наличии, используется как описано в расширении **issuingDistributionPoint**.

Поле **onlyContains**, при наличии, указывает тип(ы) сертификатов, для которых CRL содержит информацию о статусе аннулирования. Если данное поле отсутствует, то CRL содержит информацию и всех типах сертификатов.

Поле **onlySomeReasons**, при наличии, используется как описано в расширении **issuingDistributionPoint**.

Элемент **serialNumberRange**, при наличии, используется следующим образом. Когда присутствует значение модуля, порядковый номер уменьшается по модулю до заданного значения до проверки присутствия в диапазоне. Затем, считается, что сертификат с (уменьшенным) порядковым номером находится в пределах границ данного CRL, если он:

- равен или превышает **startingNumber** и меньше, чем **endingNumber**, если оба присутствуют; или
- равен или превышает **startingNumber**, если **endingNumber** не присутствует; или
- меньше, чем **endingNumber**, если **startingNumber** не присутствует.

Элемент **subjectKeyIdRange**, при наличии, толкуется так же, как и **serialNumberRange**, за исключением того, что используемый номер является значением в расширении **subjectKeyIdentifier** сертификата. Кодирование **BIT STRING** по правилам DER (опуская метку, длину и неиспользованный октет битов) должно рассматриваться как значение кодирования **INTEGER** по правилам DER. Если 0 бит **BIT STRING** установлен, то должен быть присоединен дополнительный нулевой октет для обеспечения того, чтобы итоговое кодирование представляло положительное **INTEGER**, например:

03 02 01 f7 (представляет множество битов 0-6)

отображается в

02 02 00 f7 (т. е. десятичное 247).

Поле **nameSubtrees**, при наличии, использует те же соглашения для форм имен, как определено в расширении **nameConstraints**.

Поле **baseRevocationInfo**, при наличии, указывает, что CRL является dCRL в отношении сертификатов, на которые распространяется структура **PerAuthorityScope**. Использование расширения **crlScope** для определения CRL как dCRL отличается от использования расширения **deltaCRLIdentifier** следующим образом. В случае **crlScope**, информация в компоненте **baseRevocationInfo** указывает момент времени, начиная с которого CRL, содержащий данное расширение, предоставляет обновления. Хотя это выполняется путем обращения к CRL, данный CRL может быть или не быть полным для соответствующих границ, тогда как расширение **deltaCRLIdentifier** обращается к выданному CRL, который является полным для соответствующих границ. Тем не менее, обновленная информация, предоставленная в dCRL, содержащем расширение **crlScope**, является обновлением для информации об аннулировании, полной для соответствующих границ независимо от того, был ли CRL, к которому обращался **baseRevocationInfo**, фактически выдан как полный для тех же границ. Данный механизм предоставляет больше гибкости, чем расширение **deltaCRLIndicator**, т.к. пользователи могут создавать полные CRL локально и создавать их на основе времени, а не выдачи основных CRL, которые являются полными для соответствующих границ. В обоих случаях, dCRL всегда предоставляет обновления для статуса аннулирования сертификатов в пределах заданных границ с конкретного момента времени. Тем не менее, в случае **deltaCRLIndicator** данный момент времени должен быть тем, на который CRL, являющийся полным для данных границ, был выдан и адресован. В случае **crlScope**, данный момент времени может быть тем, на который адресуемый выданный CRL может быть полным или неполным для данных границ.

В зависимости от политики ответственного органа, несколько dCRL могут быть опубликованы до опубликования нового основного CRL. dCRL, содержащие расширение **crlScope** для обращения к их точке построения, не обязательно должны ссылаться на **cRLNumber** самого последнего выданного основного CRL в поле **BaseRevocationInfo**. Тем не менее, **cRLNumber**, адресуемый в поле **BaseRevocationInfo** dCRL, должен быть меньше или равен **cRLNumber** самого последнего выданного CRL, являющегося полным для соответствующих границ.

Отметим, что расширение **issuingDistributionPoint** и расширение **crlScope** могут конфликтовать друг с другом и не предназначены для совместного использования. Тем не менее, если CRL содержит как расширение **issuingDistributionPoint**, так и расширение **crlScope**, то сертификат открытого ключа попадает в границы CRL тогда и только тогда, когда он удовлетворяет критериям обоих расширений. Если CRL содержит расширение **AAissuingDistributionPoint**, но не содержит расширения **issuingDistributionPoint** или **crlScope**, то границы не включают сертификаты открытых ключей. Если CRL не содержит ни расширение **issuingDistributionPoint**, ни **AAissuingDistributionPoint**, ни **crlScope**, то границы совпадают с полными границами данного органа, и CRL может быть выдан для любого сертификата данного органа. Аналогично, расширение **AAissuingDistributionPoint** и расширение **crlScope** могут конфликтовать друг с другом и не предназначены для совместного использования. Тем не менее, если CRL содержит как расширение **AAissuingDistributionPoint**, так и расширение **crlScope**, то сертификат атрибута попадает в границы CRL тогда и только тогда, когда он удовлетворяет критериям обоих расширений. Если CRL содержит расширение **issuingDistributionPoint**, но не содержит расширения **AAissuingDistributionPoint** или **crlScope**, то границы не включают сертификаты атрибутов. Если CRL не содержит ни расширение **issuingDistributionPoint**, ни **AAissuingDistributionPoint**, ни **crlScope**, то границы совпадают с полными границами данного органа, и CRL может быть выдан для любого сертификата данного органа.

Когда система, использующая сертификат, использует CRL, содержащий расширение **crlScope** для проверки статуса сертификата, она должна проверить, что сертификат и рассматриваемые коды причин попадают в границы CRL, как определено расширением **crlScope** следующим образом:

a) Система, использующая сертификат, должна проверить, что сертификат попадает в границы, указанные пересечением границ **serialNumberRange**, **subjectKeyIdRange** и **nameSubtrees**, и является совместимым с **distributionPoint** и **onlyContains** при наличии, для соответствующей структуры **PerAuthorityScope**.

b) Если CRL содержит компонент **onlySomeReasons** в расширении **crlScope**, то система, использующая сертификат, должна проверить, что коды причин, охватываемые данным CRL, соответствуют целям применения. Если нет, то могут потребоваться дополнительные CRL. Отметим, что если CRL содержит как расширение **crlScope**, так и расширение **issuingDistributionPoint**, и оба содержат компонент **onlySomeReasons**, то данным CRL охватываются только те коды причин, которые включены в компоненты **onlySomeReasons** обоих расширений.

8.5.2.6 Расширение передачи статуса

Данное расширение CRL предназначено для использования в структуре CRL в качестве средства передачи информации о уведомлениях об аннулировании пользователям сертификатов. Как таковой, он должен присутствовать в структуре CRL, которая сама не содержит уведомлений об аннулировании сертификатов. Структура CRL, содержащая данное расширение, не должна использоваться пользователями сертификатов или зависимыми сторонам как источник уведомлений об аннулировании, но скорее как средство обеспечения того, что используется соответствующая информация об аннулировании. Любой CRL, содержащий данное расширение, не должен использоваться как источник для зависимых сторон для проверки статуса какого-либо сертификата. Вернее, CRL, содержащий данное расширение, может использоваться зависимой стороной в качестве дополнительного средства для определения местоположения соответствующего CRL для проверки статуса аннулирования.

Данное расширение обслуживает две первичные функции:

– Данное расширение предоставляет механизм для опубликования доверенного "списка CRL", включая всю соответствующую информацию для помощи зависимым сторонам в определении того, могут ли они получить информацию об аннулировании, достаточную для их потребностей. Например, орган может выдавать новый аутентифицированный список CRL периодически, как правило, с относительно высокой частотой повторной выдачи (в сравнении в другими частотами повторной выдачи CRL). Данный список может включать дату/время последнего обновления для каждого адресуемого CRL. Пользователь сертификата, по получении данного списка, может быстро определить, являются ли еще новейшими кэшированные копии CRL. Это может исключить много ненужных восстановлений CRL. Более того, при использовании данного механизма, пользователи сертификатов получают информацию о CRL, выданных органом помимо его обычного цикла обновления, таким образом улучшая своевременность системы CRL.

– Данное расширение также предоставляет механизм для переадресации зависимой стороны от предварительного местоположения (например, кому-либо указано расширение точки распределения CRL или запись справочника выдающего органа) в другое местоположение информации об аннулировании. Данная функция дает органам возможность модифицировать схему разбиения CRL, используемую ими, без воздействия на существующие сертификаты или пользователей сертификатов. Для достижения этого, орган должен включать каждое новое местоположение и границы CRL, который находится в данном местоположении. Зависимая сторона должна сравнить рассматриваемый сертификат с положениями границ и следовать за указателем в соответствующее новое местоположение информации об аннулировании, соответствующей сертификату, подлинность которого проверяется.

Данное расширение само расширяемо, и в будущем другие схемы аннулирования, не основанные на CRL, могут также быть адресованы, используя данное расширение.

```
statusReferrals EXTENSION ::= {
    SYNTAX              StatusReferrals
    IDENTIFIED BY       id-ce-statusReferrals }

StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral

StatusReferral ::= CHOICE {
    cRLReferral         [0]     CRLReferral,
    otherReferral       [1]     INSTANCE OF OTHER-REFERRAL}


CRLReferral ::= SEQUENCE {
    issuer              [0]     GeneralName OPTIONAL,
    location            [1]     GeneralName OPTIONAL,
    deltaRefInfo        [2]     DeltaRefInfo OPTIONAL,
    cRLScope                    CRLScopeSyntax,
    lastUpdate          [3]     GeneralizedTime OPTIONAL,
    lastChangedCRL      [4]     GeneralizedTime OPTIONAL}
```

```
DeltaRefInfo ::= SEQUENCE {
    deltaLocation       GeneralName,
    lastDelta           GeneralizedTime OPTIONAL }

OTHER-REFERRAL ::= TYPE-IDENTIFIER
```

Поле **issuer** определяет объект, который подписывает CRL; по умолчанию равно имени органа, выдавшего охватывающий CRL.

Поле **location** предоставляет местоположение, в которое нужно направить передачу, и по умолчанию равно тому же значению, что и имя **issuer**.

Поле **deltaRefInfo** предоставляет альтернативное местоположение, из которого можно получить dCRL, а также (необязательно) дату предыдущей дельты.

Поле **cRLScope** предоставляет границы CRL, который будет обнаружен по адресованному местоположению.

Поле **lastUpdate** является значением поля **thisUpdate** в самом последнем выданном адресном CRL.

lastChangedCRL является значением поля **thisUpdate** самом последнем выданном CRL, содержимое которого изменилось.

OTHER-REFERRAL предоставляет расширяемость для возможности приспособления в будущем других схем аннулирования, не основанных на CRL.

Данное расширение всегда помечено как критическое для обеспечения того, что системы, использующие сертификаты, не зависят непреднамеренно от CRL, содержащего данное расширение, как источника информации о статусе аннулирования сертификатов.

Если данное расширение присутствует и узнается системой, использующей сертификат, данная система не должна использовать CRL как источник информации о статусе аннулирования. Система должна использовать либо информацию, содержащуюся в данном расширении, либо другие средства, выходящие за границы данной Спецификации, для определения местоположения информации о статусе аннулирования.

Если данное расширение присутствует, но не узнается системой, использующей сертификат, данная система не должна использовать CRL как источник информации о статусе аннулирования. Система должна использовать другие средства, выходящие за границы данной Спецификации, для определения местоположения информации о статусе аннулирования.

8.5.2.7 Расширение идентификатора потока CRL

Поле идентификатора потока CRL используется для определения контекста, в котором номер CRL является уникальным.

```
cRLStreamIdentifier EXTENSION       ::= {
    SYNTAX              CRLStreamIdentifier
    IDENTIFIED BY       id-ce-cRLStreamIdentifier }

CRLStreamIdentifier ::= INTEGER (0..MAX)
```

Данное расширение всегда является некритическим.

Каждое значение данного расширения должно быть уникальным для органа. Идентификатор потока CRL в сочетании с Номером CRL служат уникальным идентификатором для каждого CRL, выданного каким-либо заданным органом, независимо от типа CRL.

8.5.2.8 Расширение упорядоченного списка

Расширение упорядоченного списка указывает, что последовательность аннулированных сертификатов в поле **revokedCertificates** CRL находится в возрастающем порядке либо по порядковому номеру сертификата, либо по дате аннулирования. Данное поле определяется следующим образом:

```
orderedList EXTENSION       ::= {
    SYNTAX              OrderedListSyntax
    IDENTIFIED BY       id-ce-orderedList }

OrderedListSyntax ::= ENUMERATED {
ascSerialNum        (0),
ascRevDate          (1) }
```

Данное расширение всегда является некритическим.

- **ascSerialNum** указывает, что последовательность аннулированных сертификатов в CRL находится в возрастающем порядке по порядковому номеру сертификата, основанному на значении компонента **serialNumber** в каждой записи списка;
- **ascRevDate** указывает, что последовательность аннулированных сертификатов в CRL находится в возрастающем порядке по дате аннулирования, основанной на значении компонента **revocationDate** в каждой записи списка.

Если **orderedList** отсутствует, то не предоставляется информации, касающейся упорядочивания, при наличии, списка аннулированных сертификатов в CRL.

8.5.2.9 Расширение информации дельта

Данное расширение CRL предназначено для использования в CRL, которые не являются dCRL, и используется для указания зависимым сторонам, что dCRL также являются доступными для CRL, содержащих данное расширение. В данном расширении предоставлено местоположение, в котором находится соответствующие dCRL, а также (необязательно) время выдачи следующего dCRL.

```
deltaInfo EXTENSION        ::=  {
    SYNTAX                 DeltaInformation
    IDENTIFIED BY          id-ce-deltaInfo }

DeltaInformation           ::=        SEQUENCE {
    deltaLocation          GeneralName,
    nextDelta              GeneralizedTime OPTIONAL }
```

Данное расширение всегда является некритическим.

8.5.2.10 Расширение быть аннулированным

Данное расширение CRL учитывает при уведомлении, что сертификаты будут аннулированы в определенную дату и время в будущем. Расширение **toBeRevoked** используется для определения причины аннулирования сертификата, даты и времени, когда сертификат будет аннулирован, а также группу сертификатов, которые должны быть аннулированы. Каждый список может содержать один порядковый номер сертификата, ряд порядковых номеров сертификатов или поименованное **subtree**. Данные сертификаты могут быть сертификатами открытых ключей или сертификатами атрибутов.

```
toBeRevoked  EXTENSION  ::= {
  SYNTAX               ToBeRevokedSyntax
  IDENTIFIED BY        id-ce-toBeRevoked }

ToBeRevokedSyntax::=       SEQUENCE SIZE(1..MAX) OF ToBeRevokedGroup

ToBeRevokedGroup ::=       SEQUENCE {
  certificateIssuer    [0] GeneralName OPTIONAL,
  reasonInfo           [1] ReasonInfo     OPTIONAL,
  revocationTime           GeneralizedTime,
  certificateGroup         CertificateGroup }

ReasonInfo                 ::=     SEQUENCE {
  reasonCode               CRLReason,
  holdInstructionCode      HoldInstruction OPTIONAL }

CertificateGroup      ::=  CHOICE {
  serialNumbers            [0]     CertificateSerialNumbers,
  serialNumberRange        [1]     CertificateGroupNumberRange,
  nameSubtree              [2]     GeneralName }

CertificateGroupNumberRange ::= SEQUENCE {
  startingNumber     [0]     INTEGER,
  endingNumber       [1]     INTEGER }

CertificateSerialNumbers ::= SEQUENCE SIZE(1..MAX) OF CertificateSerialNumber
```

Поле **certificateIssuer**, при наличии, определяет орган (CA или AA), который выдал все сертификаты, перечисленные в данном **ToBeRevokedGroup**. Если **certificateIssuer** пропущено, по умолчанию оно принимает значение имени органа, выдавшего CRL.

Поле **reasonInfo**, при наличии, определяет причину аннулирований сертификатов. При наличии, данное поле указывает, что все сертификаты, определенные в **ToBeRevokedGroup**, будут аннулированы по причине, указанной в данном поле. Если **reasonCode** содержит значение **certificateHold**, также может присутствовать **holdInstructionCode**. При наличии, **holdInstructionCode** указывает действие, которое должно быть предпринято при обнаружении любого из сертификатов, определенных в **RevokedGroup**. Данное действие должно быть предпринято только после наступления времени аннулирования, указанного в **revocationTime**.

Поле **revocationTime** указывает дату и время, когда данная группа сертификатов будет аннулирована и потому должна считаться недействительной. Данная дата должна быть позже, чем время **thisUpdate** CRL, содержащего данное расширение. Если **revocationTime** раньше, чем время **nextUpdate** CRL, содержащего данное расширение, сертификаты должны считаться аннулированными между временем **revocationTime** и **nextUpdate** зависимой стороной, использующей CRL, содержащий данное расширение. Иначе, это является уведомлением, что в определенное время в будущем данные сертификаты будут аннулированы. При наступлении времени аннулирования CA либо аннулирует сертификат, либо нет. Если он аннулировал сертификат, будущие CRL должны включить это в список аннулированных сертификатов, по меньшей мере до истечения периода действия сертификата. Если CA не аннулировал сертификат, но все еще планирует аннулировать его в будущем, он может включить сертификат в данное расширение в последующих CRL с исправленным **revocationTime**. Если CA более не планирует аннулировать сертификат, оно может быть исключен из всех последующих CRL, и сертификат не должен считаться аннулированным.

В поле **certificateGroup** перечислены сертификаты, которые должны быть аннулированы. Данное поле определяет сертификаты, выданные органом, определенном в **certificateIssuer**, которые должны быть аннулированы на дату/время, определенное в **revocationTime**. Данное множество сертификатов в дальнейшем не уточняется каким-либо внешним контролем (например, **issuingDistributionPoint**).

Если присутствует **serialNumbers**, сертификат(ы) с порядковыми номерами, указанными в данном поле и выданные определенным органом, будет(ут) аннулирован(ы) в определенное время.

Если присутствует **serialNumberRange**, все сертификаты в диапазоне, начиная с начального порядкового номера и заканчивая конечным порядковым номеров и выданные определенным органом, выдающим сертификаты, будут аннулированы в определенное время.

Если присутствует **nameSubtree**, все сертификаты с именем субъекта/держателя, являющие подчиненными определенному имени и выданные определенным органом, выдающим сертификаты, будут аннулированы в определенное время. Если **nameSubtree** содержит DN, то должны рассматриваться все DN, связанные с субъектом сертификата открытого ключа (т. е. поле **subject** и расширение **subjectAltNames**) или полем **holder** сертификата атрибутов. Для других форм имен, должны рассматриваться расширение **subjectAltNames** сертификатов открытого ключа и поля **holder** сертификатов атрибутов. Если по меньшей мере одно из имен, связанных с субъектом/держателем, содержащимся в сертификате, находится в поддереве, определенном в **nameSubtree**, данный сертификат будет аннулирован в определенное время. Как и с расширением **nameConstraints**, не все формы имени подходят для спецификации **subtree**. В данном расширении должны использоваться только те, которые узнали правила подчинения.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Так как информация, предоставляемая данным расширением, применяется к аннулированию, которое произойдет в будущем, рекомендуется, чтобы оно было помечено как некритическое, снижая риск проблем с возможностями взаимодействия и обратной совместимостью.

8.5.2.11 Расширение аннулированной группы сертификатов

Множество сертификатов, которые были аннулированы, может быть опубликовано с использованием следующего расширения CRL. Каждый список сертификатов, которые должны быть аннулированы, связан с определенным органом, выдающим сертификаты, и временем аннулирования. Каждый список может содержать ряд порядковых номеров сертификатов или поименованное поддерево. Данные сертификаты могут быть сертификатами открытых ключей или сертификатами атрибутов.

```
revokedGroups  EXTENSION  ::= {
  SYNTAX            RevokedGroupsSyntax
  IDENTIFIED BY     id-ce-RevokedGroups }

RevokedGroupsSyntax ::=   SEQUENCE SIZE (1..MAX) OF RevokedGroup

RevokedGroup ::=   SEQUENCE {
  certificateIssuer             [0]   GeneralName OPTIONAL,
  reasonInfo                    [1]   ReasonInfo OPTIONAL,
  invalidityDate                [2]   GeneralizedTime OPTIONAL,
  revokedcertificateGroup       [3]   RevokedCertificateGroup }

RevokedCertificateGroup ::= CHOICE {
  serialNumberRange         NumberRange,
  nameSubtree               GeneralName }
```

Поле **certificateIssuer**, при наличии, определяет орган (CA или AA), который выдал все сертификаты, перечисленные в данном **ToBeRevokedGroup**. Если **certificateIssuer** пропущено, по умолчанию оно принимает значение имени органа, выдавшего CRL.

Поле **reasonInfo**, при наличии, определяет причину аннулирований сертификатов. При наличии, данное поле указывает, что все сертификаты, определенные в **ToBeRevokedGroup**, будут аннулированы по причине, указанной в данном поле. Если **reasonCode** содержит значение **certificateHold**, также может присутствовать **holdInstructionCode**. При наличии, **holdInstructionCode** указывает действие, которое должно быть предпринято при обнаружении любого из сертификатов, определенных в **RevokedGroup**.

Поле **invalidityDate**, при наличии, указывает время, начиная с которого все сертификаты, определенные в **RevokedGroup**, должны считаться недействительными. Данная дата должна быть раньше, чем дата, содержащаяся в поле **thisUpdate** CRL. Если пропущено, то все сертификаты, определенные в **RevokedGroup**, должны считаться недействительными по меньшей мере со времени, указанного в поле **thisUpdate** CRL. Если статус сертификата прежде времени **thisUpdate** является критическим для системы, использующей сертификаты (например, чтобы определить, появилась ли цифровая подпись, созданная ранее данной выдачи CRL, когда сертификат был все еще действительным или после его аннулирования), потребуются дополнительные методы проверки статуса аннулирования для определения фактической даты/времени, с которой заданный сертификат должен считаться недействительным.

В поле **revokedCertificateGroup** перечислено множество сертификатов, которые были аннулированы. Данное поле определяет сертификаты, выданные органом, определенным в **certificateIssuer**, аннулированные в определенных условиях. Данное множество сертификатов в дальнейшем не уточняется каким-либо внешним контролем (например, **issuingDistributionPoint**).

Если присутствует **serialNumberRange**, все сертификаты, содержащие порядковые номера сертификатов в определенном диапазоне, выданные определенным выдающим органом, являются применимыми.

Если присутствует **nameSubtree**, все сертификаты с именем субъекта/держателя, являющие подчиненными определенному имени и выданные определенным органом, выдающим сертификаты, будут аннулированы в определенное время. Если **nameSubtree** содержит DN, то должны рассматриваться все DN, связанные с субъектом сертификата открытого ключа (т. е. поле **subject** и расширение **subjectAltNames**) или полем **holder** сертификата атрибутов. Для других форм имен, должны рассматриваться расширение **subjectAltNames** сертификатов открытого ключа и поля **holder** сертификатов атрибутов. Если по меньшей мере одно из имен, связанных с субъектом/держателем, содержащимся в сертификате, находится в поддереве, определенном в **nameSubtree**, данный сертификат был аннулирован. Как и с расширением **nameConstraints**, не все формы имени подходят для спецификации **subtree**. В данном расширении должны использоваться только те, которые узнали правила подчинения.

Данное расширение всегда помечено как критическое. Иначе система, использующая сертификат, может ошибочно предположить, что сертификаты, определенные как аннулированные в данном расширении, не являются аннулированными. Когда данное расширение присутствует, оно может быть единственным указанием на аннулированные сертификаты в CRL (т. е. **revokedCertificates** может быть пустым) или оно может перечислять аннулированные сертификаты, дополняющие указанные в поле **revokedCertificates**. Аннулированный сертификат не должен быть указан одновременно и в поле **revokedCertificates**, и в данном расширении.

8.5.2.12 Расширение истекших сертификатов в CRL

Данное поле расширения CRL указывает, что CRL включает уведомления об аннулировании для истекших сертификатов.

```
expiredCertsOnCRL EXTENSION  ::= {
  SYNTAX      ExpiredCertsOnCRL
  IDENTIFIED BY      id-ce-expiredCertsOnCRL }

ExpiredCertsOnCRL  ::= GeneralizedTime
```

Данное расширение всегда является некритическим.

Границы CRL, содержащего данное расширение, расширены для включения статуса аннулирования сертификатов, истекших в точное время, определенное в расширении, или после этого времени. Если определены ограничения границ CRL (либо кодом причины, либо точками распределения), это также применяется к истекшим сертификатам. Статус аннулирования сертификата не должен быть обновлен после истечения периода действия сертификата.

8.6 Расширения точек распределения CRL и дельта-CRL

8.6.1 Требования

Поскольку существует возможность, что списки аннулирования станут большими и громоздкими, требуется возможность представления частичных CRL. Для двух различных типов реализаций, которые обрабатывают CRL, требуются различные подходы.

Первым типом реализации являются отдельные рабочие станции, возможно, в прикрепленном криптографическом маркере. Данные реализации, вероятно, ограничили мощность доверенного хранилища. Поэтому весь CRL должен исследоваться на предмет определения его действительности, а затем определения действительности сертификата. Данная обработка может быть продолжительной, если CRL длинный. Для устранения данной проблемы для этих реализаций требуется разбиение CRL.

Вторым типом реализации являются высокопроизводительные серверы обрабатывающие большие объемы сообщений, например, сервер по обработке операций. В данной среде CRL обычно обрабатываются как фоновые задачи, когда после проверки подлинности CRL, содержимое CRL хранится локально в представлении, которое направляет его исследование, например, один бит для каждого сертификата, указывающий, если он был аннулирован. Данное представление хранится в доверенном хранилище. Данному типу сервером как правило требуются новейшие CRL от большого количества органов. Так как у него уже имеется список ранее аннулированных сертификатов, ему необходимо только получить список новых аннулированных сертификатов. Данный список, называемый dCRL, будет меньше и потребует меньше ресурсов для получения и обработки, чем полный CRL.

Таким образом, следующие требования относятся к точкам распределения CRL и dCRL:

a) Чтобы контролировать размеры CRL, необходима возможность присваивать подмножества из множества всех сертификатов, выданных одним органом различным CRL. Этого можно достичь путем связывания каждого сертификата с точкой распределения CRL, которая является:
   - записью Справочника, атрибут CRL которого будет содержать запись об аннулировании для данного сертификата, если он был аннулирован; или
   - местоположением, таким как адрес электронной почты или универсальный идентификатор ресурса Интернет, откуда может быть получен соответствующий CRL.

b) Для повышения производительности, желательно сокращать количество CRL, которые необходимо проверять при проверке подлинности нескольких сертификатов, например, тракта сертификации. Этого можно достичь, если один выдающий CRL будет подписывать и выдавать CRL, содержащие аннулирование от нескольких органов.

c) Необходимо, чтобы отдельные CRL охватывали аннулированные сертификаты органов и аннулированные сертификаты оконечных объектов. Это упрощает обработку трактов сертификации, так как CRL для аннулированных сертификатов органов предполагаются очень короткими (обычно пустыми). Для этой цели были определены атрибуты **authorityRevocationList** и **certificateRevocationList**. Тем не менее, для обеспечения безопасности данного разбиения, необходимо в CRL иметь один указатель, определяющий, чей это список. Иначе, не сможет быть обнаружена незаконная замена одного списка другим.

d) Необходимо обеспечение существования различных CRL для возможных ситуаций фальсификации (когда есть значительный риск неправильного использования частного ключа), чем один, включающий все типовые завершения привязывания (когда нет значительного риска неправильного использования частного ключа).

e) Также обеспечение необходимо для частичных CRL (известных как dCRL), которые содержат только записи для сертификатов, которые были аннулированы с момента выдачи основного CRL.

f) Для дельта-CRL обеспечение требуется для указания даты/времени, по наступлении которых данный список содержит обновления.

g) Необходимо указывать в сертификате, где находится наиболее новый CRL (например, самый последний дельта).

8.6.2 Поля расширений точек распределения CRL и дельта-CRL

Определяются следующие поля расширений:

a) *точки распределения CRL*;

b) *выдающая точка распределения*;

c) *AA, выдающий точку распределения;*

d) *выдающий сертификат*;

e) *индикатор дельта CRL*;

f) *обновление основы*;

g) *наиболее новый CRL*.

Точки распределения CRL и наиболее новый CRL должны использоваться только как расширение сертификата. Выдающая точка распределения, AA, выдающий точку распределения, индикатор дельта CRL и обновление основы должны использоваться только как расширения CRL. Выдающий сертификат должен использоваться только как расширение записи CRL.

Хотя расширение выдающей точки распределения и AA, выдающего точки распределения, обслуживают аналогичные цели, они применяются к различным сертификатам. Расширение выдающей точки распределения применяется только к сертификатам открытых ключей, выданных пользователям и/или CA. Расширение AA, выдающего точки распределения, применяется только к сертификатам атрибутов, выданных пользователям и AA, а также сертификатам открытых ключей, выданных SOA. Если один CRL охватывает типы сертификатов, которые их перекрывают, тогда данный CRL должен включить оба расширения.

8.6.2.1 Расширение точек распределения CRL

Расширение точек распределения CRL должно использоваться только как расширение сертификата и может использоваться в сертификат органов, сертификатах открытого ключа оконечного объекта и в сертификатах атрибутов. Данное поле определяет точку или точки распределения CRL, к которым должен обратиться пользователь сертификата, чтобы установить, был ли аннулирован сертификат. Пользователь сертификата может получить CRL из соответствующей точки распределения или может иметь возможность получить текущий полный CRL из записи справочника органа.

Данное поле определяется следующим образом:

```
cRLDistributionPoints EXTENSION ::= {
    SYNTAX          CRLDistPointsSyntax
    IDENTIFIED BY   id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
    distributionPoint   [0]   DistributionPointName OPTIONAL,
    reasons             [1]   ReasonFlags OPTIONAL,
    cRLIssuer           [2]   GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
    fullName                    [0]   GeneralNames,
    nameRelativeToCRLIssuer     [1]   RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
    unused              (0),
    keyCompromise       (1),
    cACompromise        (2),
```

affiliationChanged (3),
superseded (4),
cessationOfOperation (5),
certificateHold (6),
privilegeWithdrawn (7),
aACompromise (8) }

Компонент **distributionPoint** определяет местоположение, из которого можно получить CRL. Если данный компонент отсутствует, то имя точки распределения по умолчанию принимает значение имени органа, выдавшего CRL.

Когда используется альтернатива **fullName** или когда применяется значение по умолчанию, имя точки распределения может иметь несколько форм имен. То же имя, по меньшей мере одна из его форм имен, должно присутствовать в поле **distributionPoint** расширения выдающей точки распределения CRL. Система, использующая сертификаты, не обязательно должна иметь возможность обрабатывать все формы имен. Она может использовать точку распределения, если по меньшей мере одна форма имен может быть обработана. Если ни одна форма имен для точки распределения не может быть обработана, то система, использующая сертификаты, по-прежнему может использовать сертификат, если необходимая информация об аннулировании может быть получена из другого источника, например, другой точки распределения или записи справочника органа.

Компонент **nameRelativeToCRLIssuer** может использоваться, только если точке распределения CRL присвоено имя справочника, непосредственно подчиненное имени справочника выдавшего CRL. В таком случае, компонент **nameRelativeToCRLIssuer** передает относительное выделенное имя относительно имени справочника выдавшего CRL.

Компонент **reasons** указывает причины аннулирования, охватываемые данным CRL. Если компонент **reasons** отсутствует, то соответствующая точка распределения CRL распределяет CRL, который будет содержать запись для данного сертификата, если сертификат был аннулирован, независимо от причины аннулирования. Иначе значение указывает, какие причины аннулирования охватываются соответствующей точкой распределения CRL.

Компонент **cRLIssuer** определяет орган, выдающий и подписывающий CRL. Если данный компонент отсутствует, то имя выдающего CRL по умолчанию принимает значение имени выдающего сертификаты.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. В целях возможности взаимодействия, рекомендуется, чтобы оно было помечено как некритическое.

Если данное расширение помечено как критическое, то система, использующая сертификаты, не должна использовать сертификаты без предварительного поиска и проверки CRL от одной из названных точек распределения, охватывающих рассматриваемые коды причин. В случае, если точки распределения используются для распределения информации о CRL для всех кодов причин аннулирования и все сертификаты, выданные CA, включают **cRLDistributionPoints** как критическое расширение, CA не должен также публиковать полный CRL в записи CA.

Если данное расширение помечено как некритическое и система, использующая сертификаты, не узнает тип поля расширения, то данная система должна использовать сертификат, только если:

– она может получить и проверить полный CRL от органа (на полноту последнего CRL указывает отсутствием поля расширения выдающей точки распределения в CRL);
– согласно локальной политике, не требуется проверка аннулирования; или
– проверка аннулирования выполняется при помощи других средств.

ПРИМЕЧАНИЕ 1. – Возможно, чтобы CRL для одного сертификата были выданы более, чем одним выдающим CRL. Согласование действий данных выдающих CRL и выдающего органа является областью политики органа.

ПРИМЕЧАНИЕ 2. – Значение каждого кода причины определяется в поле код причины в п. 8.5.2.2 данной Спецификации.

8.6.2.2 Расширение выдающей точки распределения

Данное поле расширения CRL определяет точку распределения CRL для сертификатов открытого ключа для данного определенного CRL и указывает, если CRL является непрямым или ограниченным, охватывая только подмножество информации об аннулировании. При использовании только частичных CRL, полная совокупность частичных CRL должна охватывать полное множество сертификатов, о статусе аннулирования которых будет сообщено с использованием механизма CRL. Таким образом, полная совокупность частичных CRL должна быть эквивалентной полному CRL для того же множества сертификатов, если орган, выдающий CRL, не использовал частичные CRL. Ограничение может быть основано на подмножестве совокупности сертификатов или подмножестве причин аннулирования. CRL подписывается частным ключом органа, выдающего CRL – точки распределения CRL не имеют своих собственных пар ключей. Тем не менее, для CRL, распределенного через Справочник, CRL хранится в записи точки распределения CRL, которая может не быть записью справочника органа, выдавшего CRL. Если поле выдающей точки распределения, поле AA, выдающего точку распределения, а также поле границ CRL отсутствуют одновременно, CRL должен содержать записи для всех аннулированных не истекших сертификатов открытых ключей, выданных органом, выдавшим CRL. Если поле выдающей точки распределения и поле границ CRL отсутствуют одновременно, а поле AA, выдающего точку распределения, присутствует, то границы CRL не включают сертификаты открытых ключей.

После появления сертификата в CRL, он может быть удален из последующего CRL после истечения периода действия сертификата. Данное поле определяется следующим образом:

```
issuingDistributionPoint EXTENSION  ::= {
    SYNTAX IssuingDistPointSyntax
    IDENTIFIED BY id-ce-issuingDistributionPoint }

IssuingDistPointSyntax ::= SEQUENCE {
    -- Если onlyContainsUserPublicKeyCerts и onlyContainsCACerts оба являются FALSE,
    -- CRL охватывает оба типа сертификатов
    distributionPoint                [0] DistributionPointName OPTIONAL,
    onlyContainsUserPublicKeyCerts   [1] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts              [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons                  [3] ReasonFlags OPTIONAL,
    indirectCRL                      [4] BOOLEAN DEFAULT FALSE }
```

Компонент **distributionPoint** содержит имя точки распределения в одной или нескольких формах имен. Если **onlyContainsUserPublicKeyCerts** является истинным, то CRL содержит аннулирования для сертификатов открытых ключей оконечных объектов. Если **onlyContainsCACerts** является истинным, то CRL содержит аннулирования для CA-сертификатов. Если оба **onlyContainsUserPublicKeyCerts** и **onlyContainsCACerts** являются ложными, то CRL содержит аннулирования как для сертификатов открытых ключей оконечных объектов, так и для CA-сертификатов. Если присутствует **onlySomeReasons**, CRL содержит только аннулирования сертификатов открытых ключей по определенной причине или причинам, иначе CRL содержит аннулирования по всем причинам. Если **indirectCRL** является истинным, то CRL может содержать уведомления об аннулировании сертификатов открытых ключей от органов, отличных от органа, выдавшего CRL. Определенный орган, ответственный за каждую запись, находится в данной записи, как указано в расширении записи CRL выдавшего сертификат, или в соответствии с правилами по умолчанию, описанными в п. 8.6.2.3. В таком CRL в обязанности выдавшего CRL входит обеспечение того, что CRL является полным и что он содержит все записи аннулирования, совместимые с индикаторами **onlyContainsUserPublicKeyCerts**, **onlyContainsCACerts** и **onlySomeReasons**, ото всех органов, которые определяют данный орган, выдающий CRL, в своих сертификатах открытых ключей.

Если CRL разбиваются по коду причины и код причины меняется для аннулированных сертификатов (приводя к перемещению сертификата из одного потока CRL в другой), необходимо продолжать включать сертификат в поток CRL для старой причины аннулирования до времени **nextUpdate** всех CRL, которые не перечисляют сертификат, для нового кода причины.

Если CRL содержит расширение **issuingDistributionPoint** с присутствующим полем **distributionPoint**, по меньшей мере одно имя для точки распределения в сертификате (например, **cRLDistributionPoints**, **freshestCRL**, **issuer**) должно соответствовать имени для точки распределения в CRL. Также может иметь мест случай, когда присутствует только поле **nameRelativeToCRLIssuer**. Тогда сравнение имен должно производиться на основе полного DN, созданного путем дополнения значения **nameRelativeToCRLIssuer** до DN, находящегося в поле **issuer** CRL. Если сравниваемые имена являются DN (в противоположность именам других форм в структуре **GeneralNames**), для сравнения двух DN на предмет эквивалентности используется правило соответствия **distinguishedNameMatch**.

Для CRL, распределенных через Справочник, применяются следующие правила. Если CRL является dCRL, он должен быть распределен через атрибут **deltaRevocationList** соответствующей точки распределения или, если не определено точки распределения, через атрибут **deltaRevocationList** записи выдавшего CRL, независимо от настроек для типов сертификатов, охватываемых CRL. Если CRL не является dCRL:

- CRL с установленным **onlyContainsCACerts**, не содержащий расширение **AAissuingDistributionPoint**, должен распределяться через атрибут **authorityRevocationList** соответствующей точки распределения или, если не определено точки распределения, через атрибут **authorityRevocationList** записи выдавшего CRL;
- CRL с установленным **onlyContainsCACerts**, и содержащий расширение **AAissuingDistributionPoint** с **containsUserAttributeCerts**, установленным в ложное значение, должен распределяться через атрибут **authorityRevocationList** соответствующей точки распределения или, если не определено точки распределения, через атрибут **authorityRevocationList** записи выдавшего CRL;
- CRL с **onlyContainsCACerts,** установленным в ложное значение, должен распределяться через атрибут **certificateRevocationList** соответствующей точки распределения или, если не определено точки распределения, через атрибут **certificateRevocationList** записи выдавшего CRL;
- CRL, содержащий как расширение **issuingDistributionPoint**, так и расширение **AAissuingDistributionPoint** с установленным **containsUserAttributeCerts**, должен распределяться через атрибут **certificateRevocationList** соответствующей точки распределения или, если не определено точки распределения, через атрибут **certificateRevocationList** записи выдавшего CRL.

Данное расширение всегда является критическим. Пользователь сертификата, который не понимает данное расширение, не может полагать, что CRL содержит полный список аннулированных сертификатов определенного органа. CRL, не содержащие критических расширение, должны содержать все текущие записи CRL для выдающего органа, включая записи для всех аннулированных сертификатов пользователей и сертификатов органов.

ПРИМЕЧАНИЕ 1. – Средства, при помощи которых информация об аннулировании передается органами к выдающим CRL, находится вне области применения данной спецификации Справочника.

ПРИМЕЧАНИЕ 2. – Если орган публикует CRL с **onlyContainsUserPublicKeyCerts** или **onlyContainsCACerts**, установленными в истинное значение, то орган должен обеспечить, что все сертификаты, охватываемые данным CRL, содержат расширение **basicConstraints**.

8.6.2.3 Расширение выдающего сертификат

Данное расширение записи CRL определяет выдающего сертификат, связанного с записью в непрямом CRL, т. е. CRL, в котором индикатор **indirectCRL** установлен в свое расширение выдающей точки распределения. Если данное расширение не присутствует в первой записи непрямого CRL, то выдающий сертификат по умолчанию принимает значение выдающего CRL. В последующих записях непрямого CRL, если данное расширение не присутствует, выдающий сертификат для записи является тем же, что и для предыдущей записи.

Данное поле определяется следующим образом:

```
certificateIssuer EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-certificateIssuer }
```

Данное расширение всегда является критическим. Если реализация проигнорировала данное расширение, она не может передавать сертификатам атрибуты статей CRL.

8.6.2.4 Расширение индикатора дельта CRL

Поле идентификатора дельта CRL определяет CRL как являющийся дельта CRL (dCRL), предоставляющим обновления для адресуемого основного CRL. Адресуемый основной CRL представляет собой CRL, который был явно выдан как CRL, полный для данных границ. CRL, содержащий расширение индикатора дельта CRL, содержит обновления для статуса аннулирования сертификатов для тех же границ. Данные границы необязательно включают все причины аннулирования или все сертификаты, выданные CA, особенно в случае, когда CRL является точкой распределения CRL. Тем не менее, сочетание CRL, содержащего расширение индикатора дельта CRL, а также CRL, адресуемый в компоненте **BaseCRLNumber** данного расширения, является эквивалентным полному CRL, для соответствующих границ, во время публикации dCRL.

Данное поле определяется следующим образом:

```
deltaCRLIndicator EXTENSION ::= {
    SYNTAX              BaseCRLNumber
    IDENTIFIED BY       id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber
```

Значение **BaseCRLNumber** определяет номер CRL основного CRL, который был использован как основа при генерации данного dCRL. Адресуемый CRL должен быть CRL, являющимся полным для соответствующих границ.

Данное расширение всегда является критическим. Пользователь сертификата, который не понимает использование dCRL, не должен использовать CRL, содержащий данное расширение, так как CRL может быть недостаточно полным относительно ожиданий пользователя.

8.6.2.5 Расширение обновления основы

Поле обновления основы предназначено для использования в dCRL и используется для определения даты/времени, по наступлении которых данный дельта предоставляет обновления для статуса аннулирования. Данное расширение должно использоваться только в dCRL, который содержит расширение **deltaCRLIndicator**. dCRL, вместо него содержащий расширение **crlScope**, не требует данное расширение, так как поле **baseThisUpdate** расширения **crlScope** может использоваться с той же целью.

```
baseUpdateTime EXTENSION        ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-baseUpdateTime }
```

Данное расширение всегда является некритическим.

8.6.2.6 Расширение наиболее нового CRL

Расширение наиболее нового CRL может использоваться как расширение или сертификата, или CRL. Для сертификатов, данное расширение может использоваться в сертификатах, выданных органам, а также сертификатах, выданных пользователям. Данное поле определяет CRL, к которому должен обратиться пользователь сертификата для получения наиболее новой информации об аннулировании (например, последний CRL). Данное поле определяется следующим образом:

```
freshestCRL EXTENSION     ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-freshestCRL }
```

Значение типа **CRLDistPointsSyntax** определено в расширении точек распределения CRL в п. 8.6.2.1.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Если расширение наиболее нового CRL является критическим, система, использующая сертификаты, не должна использовать сертификаты без предварительного поиска и проверки наиболее нового CRL. Если расширение помечено как

некритическое, система, использующая сертификаты, может использовать локальные средства для определения необходимости проверки наиболее нового CRL.

8.6.2.7 Расширение AA, выдающего точки распределения

Данное поле расширения CRL определяет точку распределения CRL для сертификатов атрибутов для данного определенного CRL и указывает, является ли CRL непрямым или указывает, если CRL является непрямым или ограниченным, охватывая только подмножество информации об аннулировании. Ограничение может быть основано на подмножестве совокупности сертификатов или подмножестве причин аннулирования. CRL подписывается частным ключом органа, выдающего CRL – точки распределения CRL не имеют своих собственных пар ключей. Тем не менее, для CRL, распределенного через Справочник, CRL хранится в записи точки распределения CRL, которая может не быть записью справочника органа, выдавшего CRL. Если расширение выдающей точки распределения, расширение AA, выдающего точку распределения, а также поле границ CRL отсутствуют одновременно, CRL должен содержать записи для всех аннулированных не истекших сертификатов открытых ключей, выданных органом, выдавшим CRL. Если поле AA, выдающего точку распределения, и поле границ CRL отсутствуют одновременно, а поле выдающей точки распределения присутствует, то границы CRL не включают сертификаты атрибутов.

После появления сертификата в CRL, он может быть удален из последующего CRL после истечения периода действия сертификата.

Данное поле определяется следующим образом:

```
AAissuingDistributionPoint EXTENSION  ::= {
    SYNTAX AAIssuingDistPointSyntax
    IDENTIFIED BY id-ce-AAissuingDistributionPoint }

AAIssuingDistPointSyntax  ::= SEQUENCE {
    distributionPoint              [ 0 ] DistributionPointName OPTIONAL,
    onlySomeReasons                [ 1 ] ReasonFlags OPTIONAL,
    indirectCRL                    [ 2 ] BOOLEAN DEFAULT FALSE,
    containsUserAttributeCerts     [ 3 ] BOOLEAN DEFAULT TRUE,
    containsAACerts                [ 4 ] BOOLEAN DEFAULT TRUE,
    containsSOAPublicKeyCerts      [ 5 ] BOOLEAN DEFAULT TRUE }
```

Компонент **distributionPoint** содержит имя точки распределения в одной или нескольких формах имен. Если присутствует **onlySomeReasons**, CRL содержит только аннулирования сертификатов атрибутов по определенной причине или причинам, иначе CRL содержит аннулирования по всем причинам.

Если **indirectCRL** является истинным, то CRL может содержать уведомления об аннулировании сертификатов атрибутов от органов, отличных от органа, выдавшего CRL. Определенный орган, ответственный за каждую запись, находится в данной записи, как указано в расширении записи CRL выдавшего сертификат, или в соответствии с правилами по умолчанию, описанными в п. 8.6.2.3. В таком CRL в обязанности выдавшего CRL входит обеспечение того, что CRL является полным и что он содержит все записи аннулирования, совместимые с индикаторами **containsUserAttributeCerts**, **containsAACerts**, **containsSOAPublicKeyCerts** и **onlySomeReasons**, ото всех органов, которые определяют данный орган, выдающий CRL, в своих сертификатах открытых ключей.

Если **containsUserAttributeCerts** является истинным, то CRL содержит аннулирования для сертификатов атрибутов, выданных оконечным объектам, самим не являющимся AA. Если **containsAACerts** является истинным, то CRL содержит аннулирования для сертификатов атрибутов, выданных субъектам, самим являющимся AA.

Если **containsSOAPublicKeyCerts** является истинным, то CRL содержит аннулирования для сертификатов открытых ключей, выданных объекту, самому не являющемуся SOA для целей управления привилегиями (т. е. сертификатов, содержащих расширение **SOAIdentifier**). Для CRL, распределенных через Справочник, применяются следующие правила. Если CRL является dCRL, он должен быть распределен через атрибут **deltaRevocationList** соответствующей точки распределения или, если не определено точки распределения, через атрибут **deltaRevocationList** записи выдавшего CRL, независимо от настроек для типов сертификатов, охватываемых CRL. Если CRL не является dCRL:

- CRL, не содержащий расширение **issuingDistributionPoint**, с установленным только **containsAACerts** и/или **containsSOAPublicKeyCerts**, должен распределяться через атрибут **attributeAuthorityRevocationList** соответствующей точки распределения или, если не определено точки распределения, через атрибут **attributeAuthorityRevocationList** записи выдавшего CRL;
- CRL, не содержащий расширение **issuingDistributionPoint**, с установленным **containsUserAttributeCerts**, (с также установленными значениями **containsAACerts** и/или **containsSOAPublicKeyCerts** или без них), должен распределяться через атрибут **attributeCertificateRevocationList** соответствующей точки распределения или, если не определено точки распределения, через атрибут **attributeCertificateRevocationList** записи выдавшего CRL;
- CRL, содержащий расширение **issuingDistributionPoint**, должен распределяться как определено в п. 8.6.2.2.

Данное расширение всегда является критическим. Пользователь сертификата, который не понимает данное расширение, не может полагать, что CRL содержит полный список аннулированных сертификатов определенного органа. CRL, не содержащие критических расширение, должны содержать все текущие записи CRL для выдающего органа, включая записи для всех аннулированных сертификатов пользователей и сертификатов органов.

ПРИМЕЧАНИЕ 1. – Средства, при помощи которых информация об аннулировании передается органами к выдающим CRL, находится вне области применения данной спецификации Справочника.

ПРИМЕЧАНИЕ 2. – Если орган публикует CRL с **containsAACerts**, установленным в **true**, и **containsUserAttributeCerts**, не установленным в **true**, то орган должен обеспечить, что все сертификаты АА, охватываемые данным CRL, содержат расширение **basicAttConstraints**.

ПРИМЕЧАНИЕ 3. – Если орган публикует CRL с **containsSOAPublicKeyCerts**, установленным в **true**, то орган должен обеспечить, что все сертификаты SOA, охватываемые данным CRL, содержат расширение **SOAIdentifier**.

9 Отношение дельта CRL к основному

dCRL включает или расширение **deltaCRLIndicator**, или расширение **crlScope** для указания основной информации об аннулировании, которая обновляется данным dCRL.

Если в dCRL присутствует **deltaCRLIndicator**, основная обновляемая информация об аннулировании является основным CRL, адресуемом в данном расширении. Основной CRL, адресуемый расширением **deltaCRLIndicator**, должен быть CRL, выданным как полный для данных границ (т. е. сам не является dCRL).

Если расширение **crlScope** присутствует и содержит компонент **baseRevocationInfo** для обращения к основной обновляемой информации об аннулировании, это является обращением к определенному моменту времени, по наступлении которого данный dCRL предоставляет обновления. Компонент **baseRevocationInfo** адресует CRL, который может быть или не быть выдан как полный для данных границ (т. е. адресуемый CRL может быть выдан только как dCRL). Тем не менее, dCRL, содержащий компонент **baseRevocationInfo**, обновляет информацию об аннулировании, являющуюся полной для границ адресуемого dCRL на момент выдачи адресуемого CRL. Пользователь сертификата может применить dCRL к CRL, который является полным для заданных границ и который был выдан одновременно или позже с выдачей CRL, адресуемого в dCRL, содержащем компонент **baseRevocationInfo**.

По причине возможности конфликта информации, CRL не должен содержать одновременно расширение **deltaCRLIndicator** и расширение **crlScope** с компонентом **baseRevocationInfo**. CRL может содержать одновременно расширение **deltaCRLIndicator** и расширение **crlScope**, только если в расширении **crlScope** не присутствует компонент **baseRevocationInfo**.

dCRL также может быть непрямым CRL, так как может содержать обновленную информацию об аннулировании, относящуюся к основному CRL, выданному одним или несколькими органами. Расширение **crlScope** должно использоваться как средство определения CRL как непрямого dCRL. Расширение **crlScope** должно содержать по одному экземпляру компонента **PerAuthorityScope** для каждого основного CRL, для которого непрямой dCRL предоставляет обновленную информацию.

Применение dCRL к адресуемой основной информации об аннулировании должно точно отражать текущий статус аннулирования.

- Уведомление об аннулировании сертификата по причине аннулирования **certificateHold** может появиться либо в dCRL, либо в CRL, полном для заданных границ. Данный код причины служит для указания временного аннулирования сертификата а ожидании дальнейшего решения о том, аннулировать ли сертификат окончательно или восстановить его как не аннулированный.
  - Если сертификат был перечислен как аннулированный по причине аннулирования **certificateHold** в CRL (либо dCRL, либо CRL, полном для заданных границ), **cRLNumber** которого равен *n*, и удержание впоследствии отменяется, то сертификат должен быть включен во все dCRL, выданные после отмены удержания, где **cRLNumber** адресуемого основного CRL меньше или равен *n*. В зависимости от расширения, используемого для указания, что данный CRL является dCRL, номер адресуемого основного CRL является либо значением компонента **BaseCRLNumber** расширения **deltaCRLIndicator**, либо элементом **cRLNumber** компонента **BaseRevocationInfo** расширения **cRLScope**. Сертификат должен быть перечислен с причиной аннулирования **removeFromCRL**, если сертификат не был впоследствии снова аннулирован по одной из причин аннулирования, охватываемой dCRL, в этом случае сертификат должен быть перечислен с причиной аннулирования, соответствующей последующему аннулированию.
  - Если сертификат не был удален из удержания, а был окончательно аннулирован, тогда он должен быть перечислен во всех последующих dCRL, где **cRLNumber** адресуемого основного CRL меньше, чем **cRLNumber** CRL (либо dCRL, либо CRL, полный для заданных границ), в котором впервые появилось уведомление о постоянном аннулировании. В зависимости от расширения, используемого для указания, что данный CRL является dCRL, номер адресуемого основного CRL является либо значением компонента **BaseCRLNumber** расширения **deltaCRLIndicator**, либо элементом **cRLNumber** компонента **BaseRevocationInfo** расширения **cRLScope**.
- Уведомление об аннулировании сертификата может впервые появиться в dCRL, и возможно, что период действия сертификата истечет до выдачи следующего CRL, полного для заданных границ. В таком случае, данное уведомление об аннулировании должно быть включено во все последующие dCRL то тех пор, пока данное уведомление об аннулировании не попадет в по меньшей мере один выданный CRL, являющийся полным для границ данного сертификата.

CRL, являющийся полным для заданных границ, в настоящее время может быть создан локально или следующими способами:

– путем поиска текущего CRL для данных границ и объединения его с выданным CRL, полным для данных границ, с **cRLNumber**, большим или равным **cRLNumber** основного CRL, адресуемого в dCRL; или

– путем поиска текущего CRL для данных границ и объединения его с локально созданным CRL, полным для данных границ, который был создан с dCRL с **cRLNumber**, большим или равным **cRLNumber** основного CRL, адресуемого в текущем dCRL.

10 Процедура обработки тракта сертификации

Обработка тракта сертификации выполняется в системе, которой необходимо использовать открытый ключ удаленного оконечного объекта, например, системе, проверяющей цифровую подпись, сгенерированную удаленным объектом. Расширения политик сертификатов, основных ограничений, ограничений имен и ограничений политик была разработаны для упрощения автоматической самодостаточной реализации логики обработки тракта сертификации.

Ниже приведены основные принципы процедуры проверки подлинности трактов сертификации. Реализация должна быть функционально эквивалентной внешнему поведению, вызываемому данной процедурой. Алгоритм, используемый определенной реализацией для получения правильного(ых) результата(ов) из заданных входных данных, не стандартизируется.

10.1 Входные данные для обработки тракта

К входным данным для процедуры обработки тракта сертификации относятся:

a) множество сертификатов, составляющих тракт сертификации;

ПРИМЕЧАНИЕ. – Каждый сертификат в тракте сертификации является уникальным. Тракт, содержащий один сертификат два или боле раз, не является действительным трактом сертификации.

b) доверенное значение отрытого ключа или идентификатор ключа (если ключ хранится внутри модуля обработки тракта сертификации) для использования при проверке первого сертификата в тракте сертификации;

c) *начальный-набор-политик,* включающий один или несколько идентификаторов политик сертификатов, указывающий, что любая из этих политик будет приемлема для пользователя сертификата для целей обработки тракта сертификации: данное входное значение также может принять специальное значение *любая-политика,* но оно не может быть нулевым;

d) значение индикатора *начальная-явная-политика,* которое указывает, должен ли идентификатор приемлемой политики явно встретиться в поле расширения политик сертификатов для всех сертификатов в тракте;

e) значение индикатора *начальный-запрет-отображения-политик*, которое указывает, запрещено ли отображение политик в тракте сертификации;

f) значение индикатора *начальное-запретить-политику*, которое указывает, считается ли специальное значение **anyPolicy**, при наличии в расширении политик сертификатов, равносильным любому определенному значению политики сертификатов в ограниченном множестве;

g) текущая дата/время (если не доступно внутри модуля обработки тракта сертификации);

h) *начальное-множество-разрешенных-поддеревьев,* содержащее начальное множество спецификаций поддеревьев, определяющих поддеревья, в которых разрешены имена субъектов (или формы имен, используемые для определения поддеревьев). В сертификатах в тракте сертификации все имена субъектов заданной формы имени, для которых определены начальные разрешенные поддеревья, должны попадать в множество разрешенных поддеревьев для данной заданной формы имен. Данное входное значение может также содержать специальное значение *неограниченно* для указания, что изначально все имена субъектов являются приемлемыми. Для пункта 10, имена субъектов являются значениями, встречающимися в поле субъект или расширении subjectAltName;

i) *начальное-множество-запрещенных-поддеревьев,* содержащее начальное множество спецификаций поддеревьев, определяющих поддеревья, в которые на могут попадать имена субъектов в сертификатах в тракте сертификации. Данное входное значение может также быть пустым множеством для указания, что первоначально не действуют никакие исключения поддеревьев;

j) *начальные-требуемые-формы-имен,* содержащие начальное множество форм имен, указывающее, что все сертификате в тракте должны включать имя субъекта по меньшей мере одной из определенных форм имен. Данное входное значение может также быть пустым множеством для указания, что не требуются никакие определенные формы имен для имен субъектов в сертификатах.

Значения c), d), e) и f) будут зависть от требований политики сочетания пользователь-приложение, которое должно использовать сертифицированный открытый ключ оконечного объекта.

Отметим, что эти данные являются индивидуальным исходными данными для процесса проверки подлинности тракта, пользователь сертификата может ограничить доверие, которое он помещает в какой-либо заданный доверенный открытый

ключ или заданное множество политик сертификатов. Этого можно достичь путем обеспечения, что заданный открытый ключ является входным значением для обработки, только когда входное значение начальный-набор-политик включает политики, которым пользователь сертификата доверяет данный открытый ключ. Так как другим входным значением для обработки является сам тракт сертификации, данный контроль может выполняться на основе операции за операцией.

10.2 Результаты обработки тракта

К результатам процедуры относятся:

a) указание успеха или неудачи проверки подлинности тракта сертификации;

b) при неудаче проверки, диагностический код, указывающий причину неудачи;

c) набор политик, ограниченных органами, и их соответствующие квалификаторы, в соответствии с которыми тракт сертификации является действительным, или специальное значение *любая-политика*;

d) набор политик, ограниченных пользователем, полученный при пересечении *набора-политик-ограниченных-органами* и *начального-набора-политик*;

e) *индикатор-явной-политики,* указывающий, требует ли пользователь сертификата или орган в тракте, чтобы приемлемая политика была определена в каждом сертификате в тракте; и

f) подробности отображения политик, произошедшего при обработке тракта сертификации.

ПРИМЕЧАНИЕ. – Если проверка подлинности прошла успешно, то система, использующая сертификат, может по-прежнему принять решение о том, чтобы не использовать сертификат, по результатам значений квалификаторов политик или другой информации в сертификате.

10.3 Переменные обработки тракта

Процедура использует следующий набор переменных состояния:

a) *набор-политик-ограниченных-органами*: Таблица идентификаторов и квалификаторов политик из сертификатов тракта сертификации (в строках представлены политики, их квалификаторы и история отображений, а в столбцах – сертификаты в тракте сертификации);

b) *разрешенные-поддеревья*: Набор спецификаций поддеревьев, определяющих поддеревья, в которые должны попадать все имена субъектов в последующих сертификатах в тракте сертификации, или может принимать специально значение *неограниченно*;

c) *исключенные-поддеревья*: (Возможно, пустой) набор спецификаций поддеревьев (каждая состоит из основного имени поддерева и индикаторов максимального и минимального уровня), определяющих поддеревья, в которые не должны попадать имена субъектов в последующем сертификате в тракте сертификации;

d) *требуемые-формы-имен*: (Возможно, пустой) набор множеств форм имен. Для каждого множества форм имен, каждый последующий сертификат должен содержать имя одной из форм имен в множестве.

e) *индикатор-явной-политики*: Указывает, должна ли приемлемая политика быть явно определена в каждом сертификате в тракте;

f) *глубина-тракта*: Целое число, на один превышающее количество сертификатов в тракте сертификации, обработка которого была завершена;

g) *индикатор-запрета-отображения-политик*: Указывает, запрещено ли отображение политик;

h) *индикатор-запретить-любую-политику*: Указывает, считается ли специальное значение **anyPolicy** равносильным любой определенной политике сертификатов;

i) *обработка-ограничений*: Подробности ограничений *явная-политика*, *запрет-отображения-политик* и/или *запретить-любую-политику*, уже обусловленных, но еще не действующих. Существует три однобитовых индикаторы, называемых *обработка-явной-политики*, *обработка-запрета-отображения-политик* и *обработка-запретить-любую-политику*, а также для каждого из них целое число, называемое *пропущенные-сертификаты*, которое задает количество сертификатов, пропускаемым до начала действия ограничения.

10.4 Этап инициализации

Данная процедура включает в себя этап инициализации, за которым следует ряд этапов по обработке сертификатов. Этап инициализации включает следующие действие:

a) записать *любая-политика* в нулевой и первой строках нулевого ряда таблицы *набор-политик-ограниченных-органами*;

b) инициализировать переменную *разрешенные-поддеревья* в значение *начальное-множество-разрешенных-поддеревьев*;

c) инициализировать переменную *исключенные-поддеревья* в значение *начальное-множество-запрещенных-поддеревьев*;

d) инициализировать переменную *требуемые-формы-имен* в значение *начальные-требуемые-формы-имен*;

e) инициализировать *индикатор-явной-политики* в значение *начальная-явная-политика*;

f) инициализировать *глубину-тракта* в значение один;

g) инициализировать *индикатор-запрета-отображения-политик* в значение *начальный-запрет-отображения-политик*;

h) инициализировать *индикатор-запретить-любую-политику* в значение *начальное-запретить-политику*;

i) инициализировать три индикатора *обработка-ограничений* в неустановленное значение.

10.5 Обработка сертификатов

Затем каждый сертификат по очереди обрабатывается, начиная с сертификата, подписанного с использованием входного доверенного открытого ключа. Последний сертификат считается конечным сертификатом, все остальные сертификаты считаются промежуточными сертификатами.

10.5.1 Проверки основных сертификатов

К сертификатам применяются следующие проверки. Автоматически подписанные сертификаты, если встречаются в тракте, игнорируются.

a) Проверить, что подпись действительна, даты действительны, имена субъекта сертификата и выдавшего сертификат сцепляются правильно и что все сертификаты не были аннулированы.

b) Для промежуточного сертификата версии 3, проверить, что **basicConstraints** присутствует и что компонент **cA** в расширении **basicConstraints** принимает значение **TRUE**. Если присутствует компонент **pathLenConstraint**, проверить, что текущий тракт сертификации не нарушает это ограничение (игнорируя промежуточные автоматически выданные сертификаты).

c) Если расширение политик сертификатов не присутствует, то установить значения *набор-политик-ограниченных-органами* в ноль путем удаления всех строк из таблицы *набор-политик-ограниченных-органами*.

d) Если расширение политик сертификатов присутствует, то для каждой политики, *P*, в расширении, отличном от **anyPolicy**, присоединить квалификаторы политик, соответствующие *P,* в каждой строке таблицы *набор-политик-ограниченных-органами*, запись столбца [*глубина-тракта*] в которой содержит значение *P*. Если ни одна строка в таблице *набор-политик-ограниченных-органами* не содержит *P* в записи столбца [*глубина-тракта*], но значением в таблице *набор-политик-ограниченных-органами* [0, *глубина-тракта*] является *любая политика*, то добавить в таблицу новую строку путем удвоения нулевой строки и записи идентификатора политики *P* вместе с ее квалификаторами в запись столбца [*глубина-тракта*] новой строки.

e) Если расширение политик сертификатов присутствует и не включает значение **anyPolicy** или если установлен *индикатор-запретить-любую-политику* и сертификат не является автоматически выданным промежуточным сертификатом, то удалить любую строку, для которой запись столбца [*глубина-тракта*] содержит значение *любая-политика*, а также любую строку, для которой запись столбца [*глубина-тракта*] не содержит одно из значений в расширении политик сертификатов.

f) Если расширение политик сертификатов присутствует и включает значение **anyPolicy** и если не установлен *индикатор-запретить-любую-политику,* то присоединить квалификаторы политик, соответствующих **anyPolicy**, к каждой строке в таблице *набор-политик-ограниченных-органами,* запись столбца [*глубина-тракта*] в которой содержит значение или содержит значение *любая-политика*, которое не встречается в расширении политик сертификатов.

g) Если сертификат не является промежуточным автоматически выданным сертификатом, проверить, что имя субъекта находится в пространстве имен, заданном значением *разрешенные-поддеревья*, и не находится в пространстве имен, заданном значением *исключенные-поддеревья*.

h) Если сертификат не является промежуточным автоматически выданным сертификатом, и если *требуемые-формы-имен* не является пустым множеством, то для каждого множества форм имен в переменной требуемые-формы-имен проверить, что в сертификате находится имя субъекта в одной из форм имен в множестве.

10.5.2 Обработка промежуточных сертификатов

Для промежуточного сертификата, затем выполняются следующие действия по ограничению регистрации, чтобы правильно установить переменные состояния для обработки следующего сертификата. Автоматически подписанные сертификаты, если встречаются в тракте, игнорируются.

a) Если расширение **nameConstraints** с компонентом **permittedSubtrees** присутствует в сертификате, установить переменную состояния *разрешенные-поддеревья* в пересечение предыдущего значения и значения, указанного в расширении сертификата.

b) Если расширение **nameConstraints** с компонентом **excludedSubtrees** присутствует в сертификате, установить переменную состояния *исключенные-поддеревья* в объединение предыдущего значения и значения, указанного в расширении сертификата.

c) Если расширение **nameConstraints** с компонентом **requiredNameForms** присутствует в сертификате, установить переменную *требуемые-формы-имен* в объединение предыдущего значения и множества, состоящего из множества форм имен, определенных в расширении сертификата. Если компонент **requiredNameForms** содержит более, чем одну форму имен, то переменная должна сигнализировать, что имя по меньшей мере одной из указанных форм имен в данном расширении должно присутствовать во всех последующих сертификатах. Объединение значения переменной *требуемые-формы-имен* со значением из расширения текущего сертификата является набором множеств, сигнализирующих требования для всех

последующих сертификатов. Например, если текущее значение *требуемые-формы-имен* установлено в требование, чтобы либо DNS, либо имя rfc822 присутствовали в сертификатах, и текущее расширение в обрабатываемом сертификате указывает, что требуются либо имена rfc822, либо имена DN, итоговое объединение, являющееся новыми требуемыми-формами-имен, указывает, что каждый ил последующих сертификатов должен иметь либо имя rfc822, либо одновременно DN и имя DNS.

d) Если *индикатор-запрета-отображения-политик* установлен:

– обработать любое расширение отображения политик для каждого отображения, определенного в расширении, путем определения местоположения всех строк в таблице *набор-политик-ограниченных-органами*, запись столбца [*глубина-тракта*] в которых равна значению политики области выдавшего органа в расширении, и удалить эту строку.

e) Если *индикатор-запрета-отображения-политик* не установлен:

– обработать любое расширение отображения политик для каждого отображения, определенного в расширении, путем определения местоположения всех строк в таблице *набор-политик-ограниченных-органами*, запись столбца [*глубина-тракта*] в которых равна значению политики области выдавшего органа в расширении, и записать значение политики области выдавшего органа из расширения в запись столбца [*глубина-тракта*+1] в той же строке. Если расширение отображает политику области выдавшего органа более, чем в одну политику области субъекта, то задействованная строка копируется и к каждой строке добавляется новая запись. Если значение в таблице *набор-политик-ограниченных-органами* [0, *глубина-тракта*] является *любая политика*, то записать идентификатор каждой политики области выдавшего при необходимости и сохраняя квалификаторы при наличии, а также записать значение политики области субъекта из расширения в запись столбца [*глубина-тракта*+1] в той же строке;

– если установлен индикатор *обработка-запрета-отображения-политик* и сертификат не является автоматически выданным, то уменьшить соответствующее значение *пропущенные-сертификаты,* и, если данное значение обратится в ноль, установить *индикатор-запрета-отображения-политик*;

– если в сертификате присутствует ограничение **inhibitPolicyMapping**, выполнить следующее. Для нулевого значения **SkipCerts** установить *индикатор-запрета-отображения-политик.* Для любого другого значения **SkipCerts**, установить индикатор *обработка-запрета-отображения-политик* и установить соответствующее значение *пропущенные-сертификаты* в меньшее из значений **SkipCerts** и предыдущего значения *пропущенные-сертификаты* (если индикатор *обработка-запрета-отображения-политик* уже был установлен).

f) Для любой строки, не измененной ни в вышеописанном шаге c), ни в d), записать идентификатор политики из столбца [*глубина-тракта*] в столбец [*глубина-тракта*+1] строки.

g) Если не установлен *индикатор-запретить-любую-политику*:

– Если индикатор *обработка-запретить-любую-политику* установлен и сертификат не является автоматически выданным, то уменьшить соответствующее значение *пропущенные-сертификаты*, и, если данное значение обратится в ноль, установить *индикатор-запретить-любую-политику*.

– Если в сертификате присутствует ограничение **inhibitAnyPolicy**, выполнить следующее. Для нулевого значения **SkipCerts** установить *индикатор-запретить-любую-политику.* Для любого другого значения **SkipCerts**, установить индикатор *обработка-запретить-любую-политику* и установить соответствующее значение *пропущенные-сертификаты* в меньшее из значений **SkipCerts** и предыдущего значения *пропущенные-сертификаты* (если индикатор *обработка-запретить-любую-политику* уже был установлен).

h) Увеличить значение [*глубина-тракта*].

10.5.3 Обработка индикатора точной политики

Затем для всех сертификатов выполняются следующие действия:

a) Если *индикатор-явной-политики* не установлен:

– Если индикатор *обработка-явной-политики* установлен и сертификат не является автоматически выданным то уменьшить соответствующее значение *пропущенные-сертификаты*, и, если данное значение обратится в ноль, установить *индикатор-явной-политики*.

– Если в сертификате присутствует ограничение **requireExplicitPolicy**, выполнить следующее. Для нулевого значения **SkipCerts** установить *индикатор-явной-политики.* Для любого другого значения **SkipCerts**, установить индикатор *обработка-явной-политики* и установить соответствующее значение *пропущенные-сертификаты* в меньшее из значений **SkipCerts** и предыдущего значения *пропущенные-сертификаты* (если индикатор *обработка-явной-политики* уже был установлен).

– Если присутствует ограничение **requireExplicitPolicy** и тракт сертификации включает сертификат, выданный названным CA, для всех сертификатов в тракте не является обязательным содержать в расширении политик сертификата идентификатор приемлемой политики. Идентификатор приемлемой политики представляет собой идентификатор политики сертификата, требуемой пользователем тракта сертификации, идентификатор политики, которая была заявлена как эквивалентная ей путем отображения политик, или *любая-политика*. Названный CA является либо CA, выдавшим сертификат,

содержащий данное расширение (если значение **requireExplicitPolicy** равно 0), либо CA, являющимся субъектом последующего сертификата в тракте сертификации (на что указывает ненулевое значение).

10.5.4 Окончательная обработка

Как только все сертификаты прошли обработку, выполняются следующие действия:

a) Определить *набор-политик-ограниченных-органами* из таблицы *набор-политик-ограниченных-органами*. Если таблица пуста, то *набор-политик-ограниченных-органами* является пустым или нулевым множеством. Если *набор-политик-ограниченных-органами [0, глубина-тракта]* является *любая-политика,* то *набор-политик-ограниченных-органами* является *любой-политикой.* Иначе *набор-политик-ограниченных-органами* для каждой строки в таблице является значением в самой левой ячейке, которая не содержит идентификатор *любая-политика*.

b) Вычислить *набор-политик-ограниченных-пользователем* путем формирования пересечения *набора-политик-ограниченных-органами* и *начального-набора-политик*.

c) Если установлен *индикатор-явной-политики*, проверить, что ни *набор-политик-ограниченных-органами*, ни *набор-политик-ограниченных-пользователем* не являются пустыми.

Если любая из вышеуказанных проверок не удается, то процедура должна завершиться, возвращая указание о неудаче, соответствующий код причины, *индикатор-явной-политики*, *набор-политик-ограниченных-органами* и *набор-политик-ограниченных-пользователем.* Если неудача произошла по причине пустого *набора-политик-ограниченных-пользователем*, тогда тракт является действительным согласно политике(кам)-ограниченным-органами, но ни одна не приемлема для пользователя.

Если ни одна из вышеуказанных проверок терпит неудачу в конечном сертификате, то процедура должна завершиться, возвращая указание об успехе, а также *индикатор-явной-политики*, *набор-политик-ограниченных-органами* и *набор-политик-ограниченных-пользователем.*

11 Схема Справочника PKI

В данном пункте определяются элементы схемы справочника, используемые для представления в Справочнике информации PKI. В него включена спецификация соответствующих классов объектов, атрибуты и правила соответствия значений атрибутов.

11.1 Классы объектов и формы имен справочника PKI

В данный подпункт включено определение классов объектов, используемых для представления объектов PKI в Справочнике.

11.1.1 Класс объектов пользователь PKI

Класс объекта пользователя PKI используется при определении записей для объектов, которые могут являться субъектом сертификатов открытых ключей.

```
pkiUser    OBJECT-CLASS   ::= {
    SUBCLASS OF       {top}
    KIND              auxiliary
    MAY CONTAIN       {userCertificate}
    ID                id-oc-pkiUser }
```

11.1.2 Класс объектов CA PKI

Класс объектов CA PKI используется при определении записей для объектов, которые действуют как органы по сертификации.

```
pkiCA      OBJECT-CLASS   ::= {
    SUBCLASS OF       {top}
    KIND              auxiliary
    MAY CONTAIN       {cACertificate |
                       certificateRevocationList |
                       authorityRevocationList |
                       crossCertificatePair }
ID                    id-oc-pkiCA }
```

11.1.3 Класс объектов и форма имени точек распределения CRL

Класс объектов точка распределения CRL используется при определении записей для объектов, которые действуют как точки распределения CRL.

```
cRLDistributionPoint      OBJECT-CLASS      ::= {
    SUBCLASS OF           { top }
    KIND                  structural
    MUST CONTAIN          { commonName }
```

```
    MAY CONTAIN         { certificateRevocationList |
                          authorityRevocationList |
                          deltaRevocationList }
    ID                  id-oc-cRLDistributionPoint }
```

Форма имени точки распределения CRL определяет, как могут быть названы записи класса объекта **cRLDistributionPoint**.

```
cRLDistPtNameForm       NAME-FORM ::= {
    NAMES               cRLDistributionPoint
    WITH ATTRIBUTES     { commonName}
    ID                  id-nf-cRLDistPtNameForm }
```

11.1.4 Класс объектов дельта CRL

Класс объектов дельта CRL используется при определении записей для объектов, которые содержат дельта списки аннулирования (например, CA, AA и т. д.).

```
deltaCRL                OBJECT-CLASS    ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {deltaRevocationList}
    ID                  id-oc-deltaCRL }
```

11.1.5 Класс объектов политика сертификатов и CPS

Класс объектов CP CPS используется при определении записей для объектов, которые содержат политику сертификатов и/или информацию о выполнении сертификации.

```
cpCps                   OBJECT-CLASS    ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {certificatePolicy |
                         certificationPracticeStmt}
    ID                  id-oc-cpCps }
```

11.1.6 Класс объектов тракт сертификации PKI

Класс объектов тракт сертификации PKI используется при определении записей для объектов, которые содержат тракты PKI. Как правило, он будет использоваться совместно с записями структурного **pkiCA** или **pkiUser**.

```
pkiCertPath             OBJECT-CLASS    ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         { pkiPath }
    ID                  id-oc-pkiCertPath }
```

11.2 Атрибуты справочника PKI

В данный подпункт включено определение атрибутов справочника для хранения элементов информации PKI в Справочнике.

11.2.1 Атрибут сертификата пользователя

Пользователь может получить один или несколько сертификатов открытых ключей от одного или нескольких CA. Тип атрибута **userCertificate** содержит сертификаты открытого ключа, которые пользователь получил от одного или нескольких CA.

```
userCertificate                 ATTRIBUTE   ::=  {
    WITH SYNTAX                 Certificate
    EQUALITY MATCHING RULE      certificateExactMatch
    ID                          id-at-userCertificate}
```

11.2.2 Атрибут сертификата CA

Атрибут **cACertificate** записи справочника CA должен использоваться для хранения автоматически выданных сертификатов (при наличии) и сертификатов, выданных данному CA CA из той же области. В случае сертификатов v3, данные сертификаты должны включать расширение **basicConstraints** со значением **cA**, установленным в **TRUE**. Определение области является задачей исключительно локальной политики.

```
cACertificate                   ATTRIBUTE   ::=  {
    WITH SYNTAX                 Certificate
    EQUALITY MATCHING RULE      certificateExactMatch
    ID                          id-at-cAcertificate }
```

11.2.3 Атрибут пары перекрестной сертификации

Элементы **issuedToThisCA** атрибута **crossCertificatePair** записи справочника CA должны использоваться для хранения всех сертификатов, выданных данному CA, за исключением автоматически выданных. По выбору, элементы **issuedToThisCA** атрибута **crossCertificatePair** записи справочника CA могут содержать подмножество сертификатов, выданных данным CA другим CA. Если CA выдает сертификат другому CA и CA субъекта не является подчиненным CA выдающего в иерархии, то CA выдающего должен поместить данный сертификат в элемент **issuedByThisCA** атрибута **crossCertificatePair** записи своего собственного справочника. Когда в одном значении атрибута присутствуют одновременно элементы **issuedToThisCA** и **issuedByThisCA**, имя выдающего в одном сертификате должно соответствовать имени субъекта в другом и наоборот, и открытый ключ субъекта в одном сертификате должен иметь возможность проверки цифровой подписи на другом сертификате и наоборот. Термин **forward** был использован в предыдущих изданиях вместо **issuedToThisCA**, а термин **reverse** был использован в предыдущих изданиях вместо **issuedByThisCA**.

Когда присутствует элемент **issuedByThisCA**, значения элементов **issuedToThisCA** и **issuedByThisCA** не обязательно должны храниться в одном и том же значении атрибута; другими словами, они могут храниться или в одном значении атрибута, или в двух значениях атрибута.

В случае сертификатов v3, они должны включать расширение **basicConstraints** со значением **cA**, установленным в **TRUE**.

```
crossCertificatePair                ATTRIBUTE  ::=  {
    WITH SYNTAX                     CertificatePair
    EQUALITY MATCHING RULE          certificatePairExactMatch
    ID                              id-at-crossCertificatePair }

CertificatePair         ::=     SEQUENCE {
issuedToThisCA          [0]     Certificate OPTIONAL,
issuedByThisCA          [1]     Certificate OPTIONAL
                                -- должна быть представлена как минимум одна пара -- }
(WITH COMPONENTS { ..., issuedToThisCA PRESENT} |
WITH COMPONENTS { ..., issuedByThisCA PRESENT})
```

11.2.4 Атрибут списка аннулированных сертификатов

Следующий атрибут содержит список аннулированных сертификатов.

```
certificateRevocationList           ATTRIBUTE  ::=  {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-certificateRevocationList }
```

11.2.5 Атрибут списка аннулированных органов

Следующий атрибут содержит список аннулированных сертификатов органов.

```
authorityRevocationList             ATTRIBUTE  ::=  {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-authorityRevocationList }
```

11.2.6 Атрибут дельта списка аннулирования

Следующий тип атрибута определен для хранения dCRL в записи справочника.

```
deltaRevocationList                 ATTRIBUTE ::= {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-deltaRevocationList }
```

11.2.7 Атрибут поддерживаемых алгоритмов

Атрибут справочника определен для поддержки выбора алгоритма, используемого при соединении с удаленным оконечным объектом, используя сертификаты, как определено в данной спецификации Справочника. Следующая ASN.1 определяет данный (многозначный) атрибут:

```
supportedAlgorithms ATTRIBUTE ::= {
    WITH SYNTAX                     SupportedAlgorithm
    EQUALITY MATCHING RULE          algorithmIdentifierMatch
    ID                              id-at-supportedAlgorithms }
```

```
SupportedAlgorithm ::= SEQUENCE {
   algorithmIdentifier                  AlgorithmIdentifier,
   intendedUsage                 [0]    KeyUsage OPTIONAL,
   intendedCertificatePolicies   [1]    CertificatePoliciesSyntax OPTIONAL }
```

Все значения многозначного атрибута должны иметь разные значения **algorithmIdentifier**. Значение компонента **intendedUsage** предоставляет указание планируемого использования алгоритма (см. п. 8.2.2.3 для распознанного использования). Значение компонента **intendedCertificatePolicies** определяет политики сертификатов и, необязательно, квалификаторы политик сертификатов, с которыми может использоваться определенный алгоритм.

11.2.8 Атрибут утверждения выполнения сертификации

Атрибут **certificationPracticeStmt** используется для хранения информации об утверждении выполнения сертификации органа.

```
certificationPracticeStmt     ATTRIBUTE  ::=  {
   WITH SYNTAX        InfoSyntax
   ID                 id-at-certificationPracticeStmt }

InfoSyntax           ::=       CHOICE {
   content           DirectoryString {ub-content},
   pointer           SEQUENCE {
      name                        GeneralNames,
      hash                        HASH { HashedPolicyInfo } OPTIONAL } }

   POLICY   ::=  TYPE-IDENTIFIER

HashedPolicyInfo      ::=   POLICY.&Type( {Policies} )
```

Policies POLICY ::= {...} – *Определяется использующими данную Рекомендацию --*

Если присутствует **content**, включается полное содержимое утверждения о выполнении сертификации органа.

Если присутствует **pointer,** компонент **name** обращается к одному или нескольким местоположениям, где может быть расположена копия утверждения о выполнении сертификации органа. Если присутствует компонент **hash**, он содержит HASH содержимого утверждения о выполнении сертификации, которое должно находиться по адресуемому местоположению. Данный хэш может использоваться для выполнения проверки целостности адресуемого документа.

11.2.9 Атрибут политики сертификатов

Атрибут **certificatePolicy** используется для хранения информации о политике сертификатов.

```
certificatePolicy          ATTRIBUTE  ::=  {
   WITH SYNTAX             PolicySyntax
   ID                      id-at-certificatePolicy }

PolicySyntax   ::=         SEQUENCE {
   policyIdentifier        PolicyID,
   policySyntax            InfoSyntax
}

   PolicyID   ::=  CertPolicyId
```

Компонент **policyIdentifier** включает идентификатор объекта, зарегистрированный для определенной политики сертификатов.

Если присутствует **content** включается полное содержимое политики сертификатов.

Если присутствует **pointer**, компонент **name** обращается к одному или нескольким местоположениям, где может быть расположена копия политики сертификатов. Если присутствует компонент **hash**, он содержит HASH содержимого политики сертификатов, которое должно находиться по адресуемому местоположению. Данный хэш может использоваться для выполнения проверки целостности адресуемого документа.

ПРИМЕЧАНИЕ. – Возможность включить хэш в данный атрибут дана исключительно для проверки целостности в отношении данных, расположенных от другого источника, отличного от справочника. HASH, хранящийся в Справочнике, нуждается в защите. Для этой цели могут использоваться службы безопасности Справочника, включая строгую аутентификацию, управление доступом и/или подписанные атрибуты. Более того, даже если HASH соответствует первоначальному документу CP/CPS, существуют дополнительные требования безопасности для обеспечения того, что первоначальная спецификация сама является правильным документом (т. е. документ пописан соответствующим органом).

11.2.10 Атрибут тракта PKI

Атрибут тракта PKI используется для хранения трактов сертификации, каждый из которых состоит из последовательности сертификатов.

```
pkiPath    ATTRIBUTE  ::= {
   WITH SYNTAX        PkiPath
   ID                 id-at-pkiPath }
```

Данный атрибут может храниться в записи справочника класса объектов **pkiCA** или **pkiUser**.

При хранении в записях **pkiCA**, значения данного атрибута содержат тракты сертификации, исключая сертификаты оконечных объектов. Как таковой, атрибут используется для хранения трактов сертификации, часто используемых зависимыми сторонами, связанными с данным CA. Значение данного атрибута может использоваться совместно с любым сертификатом оконечного объекта, выданным субъектом последнего сертификата в значении атрибута.

При хранении в записях **pkiUser**, значения данного атрибута содержат тракты сертификации, которые включают сертификат оконечного объекта. В данном случае, оконечным объектом является пользователь, запись которого содержит данный атрибут. Значения атрибута представляют полные тракты сертификации для сертификатов, выданных пользователю.

11.3 Правила соответствия справочника PKI

В данной спецификации Справочника определяются правила соответствия для использования с атрибутами типов **Certificate**, **CertificatePair**, **CertificateList**, **CertificatePolicy** и **SupportedAlgorithm**, соответственно. В данном пункте также определяются правила соответствия для упрощения выбора сертификатов или CRL с определенными характеристиками из многозначных атрибутов, содержащих несколько сертификатов или CRL. Расширенное правило соответствия предоставляет возможность для выполнения более сложного соответствия в отношении сертификатов, содержащихся в записях справочника.

11.3.1 Точное соответствие сертификатов

Правило точного соответствия сертификатов сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **Certificate**. Оно однозначно выбирает единственный сертификат.

```
certificateExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificateExactAssertion
    ID          id-mr-certificateExactMatch }

CertificateExactAssertion ::= SEQUENCE {
    serialNumber    CertificateSerialNumber,
    issuer          Name }
```

Данное правило соответствия возвращает TRUE, если компоненты в значении атрибута соответствуют значениям в представленном значении.

11.3.2 Соответствие сертификатов

Правило соответствия сертификатов сравнивает представленное значение и значение атрибута типа **Certificate**. Оно выбирает один или несколько сертификатов на основе различных характеристик.

```
certificateMatch MATCHING-RULE ::= {
    SYNTAX      CertificateAssertion
    ID          id-mr-certificateMatch }

CertificateAssertion ::= SEQUENCE {
    serialNumber            [0]     CertificateSerialNumber     OPTIONAL,
    issuer                  [1]     Name                        OPTIONAL,
    subjectKeyIdentifier    [2]     SubjectKeyIdentifier        OPTIONAL,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier      OPTIONAL,
    certificateValid        [4]     Time                        OPTIONAL,
    privateKeyValid         [5]     GeneralizedTime             OPTIONAL,
    subjectPublicKeyAlgID   [6]     OBJECT IDENTIFIER           OPTIONAL,
    keyUsage                [7]     KeyUsage                    OPTIONAL,
    subjectAltName          [8]     AltNameType                 OPTIONAL,
    policy                  [9]     CertPolicySet               OPTIONAL,
    pathToName              [10]    Name                        OPTIONAL,
    subject                 [11]    Name                        OPTIONAL,
    nameConstraints         [12]    NameConstraintsSyntax       OPTIONAL
}

AltNameType ::= CHOICE {
    builtinNameForm     ENUMERATED {
                            rfc822Name              (1),
                            dNSName                 (2),
                            x400Address             (3),
                            directoryName           (4),
                            ediPartyName            (5),
                            uniformResourceIdentifier (6),
                            iPAddress               (7),
                            registeredId            (8) },
    otherNameForm       OBJECT IDENTIFIER }

CertPolicySet ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId
```

Данное правило соответствия возвращает TRUE, если все компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам значения атрибута следующим образом:

serialNumber соответствует, если значение данного компонента в значении атрибута эквивалентно данному значению в представленном значении;

issuer соответствует, если значение данного компонента в значении атрибута эквивалентно данному значению в представленном значении;

subjectKeyIdentifier соответствует, если значение данного компонента в сохраненном значении атрибута эквивалентно данному значению в представленном значении; соответствия нет, если сохраненное значение атрибута не содержит расширения идентификатора ключа субъекта;

authorityKeyIdentifier соответствует, если значение данного компонента в сохраненном значении атрибута эквивалентно данному значению в представленном значении; соответствия нет, если сохраненное значение атрибута не содержит расширения идентификатора ключа органа или если не все компоненты в представленном значении присутствуют в сохраненном значении атрибута;

certificateValid соответствует, если представленное значение попадает в период действия сохраненного значения атрибута;

privateKeyValid соответствует, если представленное значение попадает в период, указанный расширением периода использования частного ключа, или если в сохраненном значении атрибуте нет расширения периода использования частного ключа;

subjectPublicKeyAlgID соответствует, если эквивалентно компоненту **algorithm algorithmIdentifier** или компоненту **subjectPublicKeyInformation** сохраненного значения атрибута;

keyUsage соответствует, если все биты, установленные в представленном значении, также установлены в расширении использования ключа в сохраненном значении атрибута; или если в сохраненном значении атрибуте нет расширения использования ключа;

subjectAltName соответствует, если сохраненное значение атрибута содержит расширение альтернативного имени субъекта с компонентом **AltNames** того же типа, как указано в представленном значении;

policy соответствует, если по меньшей мере один член представленного **CertPolicySet** появляется в расширении политик сертификатов в сохраненном значении атрибута, или если либо представленный, либо сохраненный сертификат содержит специальное значение **anyPolicy** в компоненте **policy**. Соответствия нет, если в сохраненном значении атрибута нет расширения политик сертификатов;

pathToName соответствует, если сертификат не содержит расширения ограничения имен, запрещающего создание тракта сертификации к представленному значению имени;

subject соответствует, если значение данного компонента в значении атрибута эквивалентно значению в представленном значении;

nameConstraints соответствует, если имена субъектов в сохраненном значении атрибута находятся в пространстве имен, заданном компонентом разрешенные-поддеревья представленного значения, и не находятся в пространстве имен, заданном компонентом исключенные-поддеревья представленного значения.

11.3.3 Точное соответствие пар сертификатов

Правило точного соответствия пар сертификатов сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **CertificatePair**. Оно однозначно выбирает единственную пару перекрестных сертификатов.

```
certificatePairExactMatch MATCHING-RULE          ::=  {
    SYNTAX     CertificatePairExactAssertion
    ID         id-mr-certificatePairExactMatch }

CertificatePairExactAssertion ::= SEQUENCE {
    issuedToThisCAAssertion     [0] CertificateExactAssertion OPTIONAL,
    issuedByThisCAAssertion     [1] CertificateExactAssertion OPTIONAL }
    ( WITH COMPONENTS               {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS               {..., issuedByThisCAAssertion PRESENT} )
```

Данное правило соответствия возвращает TRUE, если компоненты, присутствующие в компонентах **issuedToThisCAAssertion** и **issuedByThisCAAssertion** представленного значения, соответствуют соответствующим компонентам **issuedToThisCA** и **issuedByThisCA** соответственно в сохраненном значении атрибута.

11.3.4 Соответствие пар сертификатов

Правило соответствия пар сертификатов сравнивает представленное значение и значение атрибута типа **CertificatePair**. Оно выбирает одну или несколько пар перекрестных сертификатов на основе различных характеристик или **issuedToThisCA**, или **issuedByThisCA** сертификата пары.

```
certificatePairMatch MATCHING-RULE ::=  {
    SYNTAX     CertificatePairAssertion
    ID         id-mr-certificatePairMatch }

CertificatePairAssertion ::= SEQUENCE {
    issuedToThisCAAssertion     [0] CertificateAssertion OPTIONAL,
    issuedByThisCAAssertion     [1] CertificateAssertion OPTIONAL }
```

```
( WITH COMPONENTS        {..., issuedToThisCAAssertion PRESENT} |
WITH COMPONENTS          {..., issuedByThisCAAssertion PRESENT} )
```

Данное правило соответствия возвращает TRUE, если компоненты, присутствующие в компонентах **issuedToThisCAAssertion** и **issuedByThisCAAssertion** представленного значения, соответствуют соответствующим компонентам **issuedToThisCA** и **issuedByThisCA** соответственно в сохраненном значении атрибута.

11.3.5 Точное соответствие списков сертификатов

Правило точного соответствия списков сертификатов сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **CertificateList**. Оно однозначно выбирает единственный CRL.

```
certificateListExactMatch MATCHING-RULE ::=     {
    SYNTAX    CertificateListExactAssertion
    ID        id-mr-certificateListExactMatch }

CertificateListExactAssertion ::= SEQUENCE {
    issuer              Name,
    thisUpdate          Time,
    distributionPoint   DistributionPointName OPTIONAL }
```

Данное правило возвращает TRUE, если все компоненты в сохраненном значении атрибута соответствуют компонентам в представленном значении. Если присутствует компонент **distributionPoint**, то он должен соответствовать по меньшей мере по одной форме имени.

11.3.6 Соответствие списков сертификатов

Правило соответствия списков сертификатов сравнивает представленное значение и значение атрибута типа **CertificateList**. Оно выбирает один или несколько CRL на основе различных характеристик.

```
certificateListMatch MATCHING-RULE ::= {
    SYNTAX    CertificateListAssertion
    ID        id-mr-certificateListMatch }

CertificateListAssertion ::= SEQUENCE {
    issuer                  Name                    OPTIONAL,
    minCRLNumber      [0]   CRLNumber               OPTIONAL,
    maxCRLNumber      [1]   CRLNumber               OPTIONAL,
    reasonFlags             ReasonFlags             OPTIONAL,
    dateAndTime             Time                    OPTIONAL,
    distributionPoint [2]   DistributionPointName   OPTIONAL,
    authorityKeyIdentifier [3] AuthorityKeyIdentifier OPTIONAL }
```

Данное правило соответствия возвращает TRUE, если все компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам сохраненного значения атрибута следующим образом:

issuer соответствует, если значение данного компонента в значении атрибута эквивалентно данному значению в представленном значении;

minCRLNumber соответствует, если его значение меньше или равно значению в расширении номера CRL в сохраненном значении атрибута; соответствия нет, если сохраненное значение не содержит расширения номера CRL;

maxCRLNumber соответствует, если его значение больше или равно значению в расширении номера CRL в сохраненном значении атрибута; соответствия нет, если сохраненное значение не содержит расширения номера CRL;

reasonFlags соответствует, если любой из битов, установленных в представленном значении, также установлен в компонентах **onlySomeReasons** расширения выдающей точки распределения в сохраненном значении атрибута; соответствие также есть, если сохраненное значение атрибута не содержит **reasonFlags** в расширении выдающей точки распределения или если сохраненное значение атрибута не содержит расширения выдающей точки распределения;

ПРИМЕЧАНИЕ. – Даже несмотря на то, что CRL соответствует определенному значению **reasonFlags**, CRL может не содержать уведомлений об аннулировании с данным кодом ошибки.

dateAndTime соответствует, если значение равно или позже значения в компоненте **thisUpdate** сохраненного значения атрибута и раньше, чем значение в компоненте **nextUpdate** сохраненного значения атрибута; соответствия нет, если сохраненное значение атрибута не содержит компонента **nextUpdate**;

distributionPoint соответствует, если сохраненное значение атрибута содержит расширение выдающей точки распределения и значение данного компонента в представленном значении эквивалентно соответствующему значению в данном расширении, по меньшей мере, в одной из форм имен.

authorityKeyIdentifier соответствует, если значение данного компонента в сохраненном значении атрибута эквивалентно значению в представленном значении; соответствия нет, если сохраненное значение атрибута не содержит расширения идентификатора ключа органа или если не все компоненты в представленном значении присутствуют в сохраненном значении атрибута.

11.3.7 Соответствие идентификаторов алгоритмов

Правило соответствия идентификаторов алгоритмов сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **SupportedAlgorithms**.

```
algorithmIdentifierMatch MATCHING-RULE::=  {
    SYNTAX      AlgorithmIdentifier
    ID          id-mr-algorithmIdentifierMatch }
```

Правило возвращает TRUE, если представленное значение эквивалентно компоненту **algorithmIdentifier** сохраненного значения атрибута.

11.3.8 Соответствие политик

Правило соответствия политик сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **CertificatePolicy** или значение атрибута типа **privPolicy**.

```
policyMatch MATCHING-RULE     ::= {
    SYNTAX      PolicyID
    ID          id-mr-policyMatch }
```

Правило возвращает TRUE, если представленное значение эквивалентно компоненту **policyIdentifier** сохраненного значения атрибута.

11.3.9 Соответствие трактов PKI

Правило соответствия **pkiPathMatch** сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **pkiPath**. Система, использующая сертификат, может использовать данное правило соответствия для выбора тракта, начинающегося с сертификата, выданного CA, которому она доверяет, и заканчивая сертификатом, выданным определенному субъекту.

```
pkiPathMatch MATCHING-RULE  ::= {
    SYNTAX      PkiPathMatchSyntax
    ID          id-mr-pkiPathMatch }

PkiPathMatchSyntax      ::=  SEQUENCE {
    firstIssuer         Name,
    lastSubject         Name }
```

Данное правило соответствия возвращает TRUE, если представленное значение в компоненте **firstIssuer** соответствует соответствующим элементам поля **issuer** первого сертификата в **SEQUENCE** в сохраненном значении, и представленное значение в компоненте **lastSubject** соответствует соответствующим элементам поля **subject** последнего сертификата в **SEQUENCE** в сохраненном значении. Данное правило соответствия возвращает FALSE, если любое из соответствий терпит неудачу.

11.3.10 Расширенное соответствие сертификатов

Правило расширенного соответствия сертификатов сравнивает представленное значение и значение атрибута типа **Certificate**. Оно выбирает один или несколько сертификатов на основе различных характеристик.

```
enhancedCertificateMatch MATCHING-RULE  ::= {
    SYNTAX      EnhancedCertificateAssertion
    ID          id-mr-enhancedCertificateMatch }

EnhancedCertificateAssertion  ::= SEQUENCE {
    serialNumber            [0]  CertificateSerialNumber    OPTIONAL,
    issuer                  [1]  Name                       OPTIONAL,
    subjectKeyIdentifier    [2]  SubjectKeyIdentifier       OPTIONAL,
    authorityKeyIdentifier  [3]  AuthorityKeyIdentifier     OPTIONAL,
    certificateValid        [4]  Time                       OPTIONAL,
    privateKeyValid         [5]  GeneralizedTime            OPTIONAL,
    subjectPublicKeyAlgID   [6]  OBJECT IDENTIFIER          OPTIONAL,
    keyUsage                [7]  KeyUsage                   OPTIONAL,
    subjectAltName          [8]  AltName                    OPTIONAL,
    policy                  [9]  CertPolicySet              OPTIONAL,
    pathToName              [10] GeneralNames               OPTIONAL,
    subject                 [11] Name                       OPTIONAL,
    nameConstraints         [12] NameConstraintsSyntax      OPTIONAL
}

(ALL EXCEPT ({ -- отсутствует; должен присутствовать по меньшей мере один компонент -- }))

AltName  ::= SEQUENCE {
  altnameType       AltNameType,
  altNameValue      GeneralName OPTIONAL }
```

Операция поиска справочника позволяет сочетать несколько значений **EnhancedCertificateAssertion** в спецификациях фильтра, включая логику и/или. Данное правило соответствия возвращает TRUE, если все компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам значения атрибута следующим образом:

Соответствие для компонентов **serialNumber**; **issuer**; **subjectKeyIdentifier**; **authorityKeyIdentifier**; **certificateValid**, **privateKeyValid**, **policy**, **subject** и **nameConstraints** определено в правиле соответствия **certificateMatch**.

Компонент **subjectAltName** содержит поле **altNameType** и (необязательно) поле **altNameValue**. Если присутствует **altNameValue**, значение должно иметь форму имен, указанную в **altNameType**.

subjectAltName соответствует, если выполняется по меньшей мере одно из следующих условий:

- представленное значение содержит только компонент **altNameType**, и сохраненное значение атрибута содержит расширение альтернативного имени субъекта с компонентом **AltNames** типа, указанного в представленном значении;
- представленное значение содержит как компонент **altNameType**, так и компонент **altNameValue**, и сохраненное значение атрибута содержит расширение альтернативного имени субъекта с компонентом **AltNames** типа и значения, указанных в представленном значении;

subjectAltName не соответствует, если выполняется по меньшей мере одно из следующих условий:

- сохраненное значение атрибута не содержит расширения альтернативного имени субъекта;
- сохраненное значение атрибута содержит расширение альтернативного имени субъекта, но компонент **AltNames** не включает тип, определенный в представленном значении;
- представленное значение содержит как компонент **altNameType**, так и компонент **altNameValue**, и сохраненное значение атрибута содержит расширение альтернативного имени субъекта с компонентом **AltNames** типа, указанного в представленном значении, но сохраненное значение не содержит того же значения данного типа, как и в представленном значении.

Соответствие **subjectAltName** не определено, если представленное значение содержит как компонент **altNameType**, так и компонент **altNameValue**, и сохраненное значение атрибута содержит расширение альтернативного имени субъекта с компонентом **AltNames** типа, указанного в представленном значении, но для данного типа справочник не имеет возможности сравнить значения для целей определения соответствия. Причиной этого может быть форма имен, не подходящая для соответствия, или невозможность справочника выполнить требуемые сравнения.

pathToName соответствует, если сертификат не содержит расширения ограничения имен, запрещающего создание тракта сертификации к любому из представленных значений имен. Например, при попытке поиска сертификатов, которые формируют тракт к сертификату пользователя, значение субъекта которого равно "dc=com; dc=corporate; cn=john.smith", может быть полезно включить в операцию поиска утверждение, содержащее данный DN в компонент **pathToName**. Сохраненный сертификат, содержащий расширение ограничения имени, исключающее полное поддерево ниже основы "dc=com; dc=company A", должен потерпеть неудачу при проверке подлинности тракта сертификации к данному сертификату пользователя и поэтому не будет являться соответствующим значением для данного выборочного утверждения.

РАЗДЕЛ 3 – СТРУКТУРА СЕРТИФИКАТОВ АТРИБУТОВ

Структура сертификатов атрибутов, определенная здесь, предоставляет основу для построения инфраструктуры управления привилегиями (PMI). Данные инфраструктуры могут поддерживать приложения, такие как управление доступом.

Связывание привилегии и объекта предоставляется органом при помощи структуры данных с цифровой подписью, называемой сертификатом атрибута, или путем сертификата открытого ключа, содержащего для этой цели явно определенное расширение. Здесь определяется формат сертификатов атрибутов, включая механизм расширяемости и набор характерных расширений сертификатов. Аннулирование сертификатов атрибутов может потребоваться, а может не потребоваться. Например, в некоторых средах периоды действия сертификатов атрибутов могут быть очень короткими (например, минуты), отрицая необходимость в схеме аннулирования. Если по какой-либо причине орган аннулирует предварительно выданный сертификат атрибута, то пользователи должны иметь возможность узнать, что данное аннулирование имело место, чтобы не использовать ненадежный сертификат. Списки аннулирования являются одной из схем, которые могут использоваться для уведомления пользователей об аннулированиях. Формат списков аннулирования определен во 2-м Разделе данной Спецификации, включая механизм расширяемости и набор расширений списков аннулирования. Здесь определяются дополнительные расширения. Как в случае сертификатов, так и списков аннулирования, другие тела также могут определять дополнительные расширения, полезные для своих определенных сред.

Система, использующая сертификаты атрибутов, должна проверить подлинность сертификата до использования данного сертификата для приложения. Процедуры для выполнения данной проверки также определяются здесь, включая проверку целостности самого сертификата, его статуса аннулирования и его подлинности в отношении планируемого использования.

В данную структуру включено несколько дополнительных элементов, подходящих только для некоторых сред. Хотя модели определяются как полные, данная структура может использоваться в средах, где используются не все компоненты определенных моделей. Например, существуют среды, где не требуется аннулирование сертификатов атрибутов. Делегирование привилегий и использование ролей также представляют собой аспекты данной структуры, не являющиеся универсально применимыми. Тем не менее, они включены в данную Спецификацию, поэтому также могут поддерживаться требующие их среды.

Справочник использует сертификаты атрибутов для предоставления основанного на правилах управления доступом к информации Справочника.

12 Сертификаты атрибутов

Сертификаты открытых ключей в основном предназначены для предоставления услуги идентичности, на которой могут быть построены другие услуги безопасности, такие как целостность данных, аутентификация объекта, конфиденциальность и авторизация. В данной Спецификации представлено два разных механизма для привязывания атрибута привилегии к держателю.

Сертификаты открытых ключей, используемые совместно с услугой аутентификации объекта, могут предоставлять услугу авторизации непосредственно, если привилегии связаны с субъектом посредством действий выдающего CA. Сертификаты открытых ключей могут содержать расширение **subjectDirectoryAttributes**, содержащее привилегии, связанные в субъектом сертификата открытого ключа. Данный механизм является подходящим в ситуациях, когда орган, выдающий сертификат открытого ключа (CA), является также органом для делегирования привилегий (AA) и период действия привилегии совпадает с периодом действия сертификата открытого ключа. Оконечные объекты не могут действовать как AA. Если какое-либо расширение, определенное в пункте 15, включено в сертификат открытого ключа, то данное расширение применяется в равной степени ко всем привилегиям, присвоенным в расширении **subjectDirectoryAttributes** данного сертификата открытого ключа.

В более общем случае, продолжительность жизни привилегий объектов не будет совпадать с периодом действия сертификата открытого ключа. Продолжительность жизни привилегий часто много более коротка. Орган по присвоению привилегий будет часто отличаться от органа, выдающего данному объекту сертификат открытого ключа, и различные привилегии могут присваиваться различными органами атрибутов (AA). Привилегии также могут быть присвоены на основе временного контекста, и аспект привилегий 'включить/выключить' может быть далеко не синхронным с продолжительностью жизни сертификата открытого ключа и/или не синхронным с привилегиями объектов, выданными другим AA. Использование сертификатов атрибутов, выданных AA, предоставляет гибкую инфраструктуру управления привилегиями (PMI), которая может быть установлена и управляема независимо от PKI. В то же время, существует связь между этими структурами, в силу которой PKI используется для аутентификации идентификационной информации выдающих органов и держателей в сертификатах атрибутов.

12.1 Структура сертификата атрибутов

Сертификат атрибута является структурой, отдельной от сертификата открытого ключа субъекта. Субъект может иметь несколько сертификатов атрибутов, связанных с каждым из своих сертификатов открытых ключей. Не требуется, чтобы один орган создавал как сертификат открытого ключа, так и сертификат(ы) атрибутов для пользователя, фактически, иначе часто будет предписываться разделение обязанностей. В средах, где различные органы несут ответственность за выдачу сертификатов открытых ключей и сертификатов атрибутов, сертификат(ы) открытого ключа, выданные органом по сертификации (CA), и сертификат(ы) атрибутов, выданные органом атрибутов (AA), будут подписаны с использованием различные частных подписывающих ключей. В средах, где один объект является одновременно CA, выдающим сертификаты открытых ключей, и AA, выдающим сертификаты атрибутов, строго рекомендуется, чтобы для подписи сертификатов атрибутов использовался другой ключ, чем для подписания сертификатов открытых ключей. Обмены между выдающим органом и объектом, получающим сертификат, находятся вне области применения данной Спецификации.

Сертификат атрибута определяется следующим образом:

```
AttributeCertificate ::= SIGNED {AttributeCertificateInfo}
AttributeCertificateInfo ::= SEQUENCE
    {
    version                 AttCertVersion, -- версия v2
    holder                  Holder,
    issuer                  AttCertIssuer,
    signature               AlgorithmIdentifier,
    serialNumber            CertificateSerialNumber,
    attrCertValidityPeriod  AttCertValidityPeriod,
    attributes              SEQUENCE OF Attribute,
    issuerUniqueID          UniqueIdentifier OPTIONAL,
    extensions              Extensions    OPTIONAL
    }
AttCertVersion ::= INTEGER { v2(1) }
Holder     ::=   SEQUENCE
        {
        baseCertificateID       [0] IssuerSerial          OPTIONAL,
```

```
        -- выдавший орган и порядковый номер держателя сертификата открытого ключа
    entityName          [1] GeneralNames      OPTIONAL,
        -- имя объекта или роль
    objectDigestInfo    [2] ObjectDigestInfo  OPTIONAL
        -- используется для прямой аутентификации держателя, например, выполнимый
        -- по меньшей мере один из baseCertificateID, entityName или objectDigestInfo должен
        -- присутствовать--}

ObjectDigestInfo  ::= SEQUENCE {
    digestedObjectType  ENUMERATED {
        publicKey           (0),
        publicKeyCert       (1),
        otherObjectTypes    (2) },
    otherObjectTypeID       OBJECT IDENTIFIER  OPTIONAL,
    digestAlgorithm         AlgorithmIdentifier,
    objectDigest            BIT STRING }

AttCertIssuer ::=   [0]  SEQUENCE {
issuerName                  GeneralNames OPTIONAL,
baseCertificateID       [0] IssuerSerial OPTIONAL,
objectDigestInfo        [1] ObjectDigestInfo OPTIONAL }
-- Должен присутствовать по меньшей мере один компонент
    ( WITH COMPONENTS { ..., issuerName PRESENT } |
    WITH COMPONENTS { ..., baseCertificateID PRESENT } |
    WITH COMPONENTS { ..., objectDigestInfo PRESENT } )

IssuerSerial  ::=  SEQUENCE {
    issuer      GeneralNames,
    serial      CertificateSerialNumber,
    issuerUID   UniqueIdentifier OPTIONAL }

AttCertValidityPeriod  ::= SEQUENCE {
    notBeforeTime       GeneralizedTime,
    notAfterTime        GeneralizedTime }
```

version проводит различия между разными версиями сертификата атрибутов. Для сертификатов атрибутов, выданных в соответствии с синтаксисом в данной Спецификации, **version** должна быть **v2**.

Поле **holder** передает идентификационную информацию о держателе сертификата атрибута.

Компонент **baseCertificateID,** при наличии, определяет определенный сертификат открытого ключа, который должен использоваться для аутентификации идентификационной информации держателя при утверждении привилегий с данным сертификатом атрибутов.

Компонент **entityName**, при наличии, определяет одно или несколько имен для держателя. Если **entityName** является единственным компонентом, присутствующим в **holder**, любой сертификат открытого ключа, имеющий одно из этих имен в качестве своего субъекта, может использоваться для аутентификации данного держателя при утверждении привилегий с данным сертификатом атрибутов. Если присутствуют как **baseCertificateID**, так и **entityName**, может использоваться только сертификат, определенный **baseCertificateID**. В данном случае **entityName** включается только как средство для помощи верификатору привилегий в определении местоположения определенного сертификата открытого ключа.

ПРИМЕЧАНИЕ 1. – Существует риск исключительного использования **GeneralNames** для определения держателя в части того, что он указывает только имя держателя. Этого как правило недостаточно для возможности аутентификации идентификационной информации держателя для целей выдачи привилегий данному держателю. Использование имени выдавшего органа и порядкового номера определенного сертификата открытого ключа, тем не менее, дает возможность органу, выдающему сертификаты атрибутов, полагаться на процесс аутентификации, выполняемый CA при выдаче данного определенного сертификата открытого ключа. Также, некоторые из возможностей в **GeneralNames** (например, **IPAddress**) являются неподходящими для использования в присвоении имени держателю сертификата атрибута, особенно когда держатель является ролью, а не индивидуальным объектом. Другая проблема в **GeneralNames** как единственном идентификаторе держателя заключается в том, что многие формы имен в данной структуре не имеют строгих органов по регистрации или процессов для присвоения имен.

Компонент **objectDigestInfo**, при наличии, используется для аутентификации идентификационной информации держателя, включая выполняемого держателя (например, прикладная мини-программа). Аутентификация держателя происходит путем сравнения сборника соответствующей информации, созданного верификатором привилегий при помощи алгоритма, определенного в **objectDigestInfo**, с содержимым **objectDigest**. Если они идентичны, то держатель аутентифицируется для целей утверждения привилегий с данным сертификатом атрибутов.

– **publicKey** должен быть указан, когда включается хэш открытого ключа объекта. Хэширование открытого ключа может не однозначно определять один сертификат (т. е. идентичное значение ключа может встретиться в нескольких сертификатах). Чтобы привязать сертификат атрибута к открытому ключу, вычисляется хэш по представлению данного открытого ключа, который должен присутствовать в сертификате открытого ключа. В особенности, входным данным для алгоритма хэширования должно быть кодирование по правилам DER представления **SubjectPublicKeyInfo** ключа. Отметим, что сюда включается **AlgorithmIdentifier**, а также **STRING**. Отметим, что если значение открытого ключа, использованное как входное значение для хэш-функции, было получено из сертификата открытого ключа, то возможно, что впоследствии (например, если были унаследованы параметры для Алгоритма цифровой

подписи) этого входного значения будет недостаточно для HASH. Правильные входные данные для хэширования в данном контексте будут включать значение унаследованных параметров, и, таким образом, они могут отличаться от **SubjectPublicKeyInfo**, присутствующего в сертификате открытого ключа.

- **publicKeyCert** должен быть указан, когда хэшируется сертификат открытого ключа; хэш выполняется по полному кодированию по правилам DER сертификата открытого ключа, включая биты подписи.
- **otherObjectTypes** должен быть указан, когда хэшируются объекты, отличные от открытых ключей или сертификатов открытых ключей (например, объекты программного обеспечения). Дополнительно может предоставляться идентификационная информация типа объекта. Хэшируемую часть объекта можно определить либо из явно утвержденного идентификатора типа, либо, если идентификатор не предоставляется, либо из контекста, в котором используется объект.

Поле **issuer** передает идентификационную информацию AA, выдавшего данный сертификат.

- Компонент **issuerName**, при наличии, определяет одно или несколько имен для выдавшего органа.
- Компонент **baseCertificateID**, при наличии, определяет выдавший орган путем обращения к определенному сертификату открытого ключа, субъектом для которого является данный выдавший орган.
- Компонент **objectDigestInfo**, при наличии, определяет выдавший орган путем предоставления хэша идентификационной информации выдавшего органа.

signature определяет криптографический алгоритм, используемый для подписания цифровой подписью сертификата атрибута.

serialNumber представляет собой порядковый номер, который однозначно определяет сертификат атрибута в границах своего выдавшего органа.

Поле **attrCertValidityPeriod** передает период времени, в течение которого сертификат атрибута считается действительным, выраженный в формате **GeneralizedTime**.

Поле **attributes** содержит сертифицируемые атрибуты, связанные с держателем (например, привилегии).

ПРИМЕЧАНИЕ 2. – В случае сертификатов атрибутов дескрипторов атрибута, данная последовательность атрибутов может быть пустой.

issuerUniqueID может использоваться для определения органа, выдавшего сертификат атрибута, в случаях, когда компонента выдавший орган недостаточно.

Поле **extensions** дает возможность добавления новых полей в сертификат атрибута.

Если в расширении появляются неизвестные элементы и расширение не помечено как критическое, данные неизвестные элементы должны игнорироваться в соответствии с правилами расширяемости, документированными в п. 12.2.2 Рек. МСЭ-Т X.519 | ИСО/МЭК 9594-5.

Структура для сертификатов, описанная в данном разделе, в основном фокусируется на модели, в которой привилегия помещается в сертификаты атрибутов. Тем не менее, как упоминалось ранее, расширения сертификатов, определенные в данном разделе, также могут помещаться в сертификат открытого ключа с использованием расширения **subjectDirectoryAttributes**.

12.2 Тракты сертификатов атрибутов

Как и с сертификатами открытых ключей, может требоваться передавать тракт сертификатов атрибутов (например, в протоколе приложения для утверждения привилегий). Следующий тип данных ASN.1 может использоваться для представления тракта сертификатов атрибутов:

```
AttributeCertificationPath ::= SEQUENCE {
    attributeCertificate        AttributeCertificate,
    acPath                      SEQUENCE OF ACPathData OPTIONAL }

ACPathData ::= SEQUENCE {
    certificate             [0] Certificate OPTIONAL,
    attributeCertificate    [1] AttributeCertificate OPTIONAL }
```

13 Связь органа атрибутов, SOA и органа по сертификации

Орган атрибутов (AA) и орган по сертификации (CA) являются логически (и во многих случаях физически) полностью независимыми. Создание и поддержка "идентификационной информации" может (и часто должна) быть отделена от PMI. Таким образом, полная PKI, включая CA, может существовать и работать до установления PMI. CA, хотя и является источником органа для идентификационной информации в своей области, не является автоматически источником органа для привилегий. Поэтому CA не обязательно сам должен быть AA и, путем логической импликации, не обязательно будет ответственен за решение в отношении того, что другие объекты смогут функционировать как AA (например, путем включения такого назначения в их идентификационные сертификаты).

Источник органа (SOA) является объектом, которому верификатор привилегий доверяет как объекту с наибольшей ответственностью за присвоение совокупности привилегий. Ресурс может ограничить орган SOA путем доверия определенным SOA конкретных функций (например, одному чтение привилегий, а другому – запись привилегий). SOA сам является AA, так как он выдает сертификаты другим объектам, в которых данным объектам присваиваются привилегии. SOA аналогичен 'корневому CA' или 'опоре доверия' в PKI, так как верификатор привилегий доверяет сертификатам, подписанным SOA. В некоторых средах необходимо, чтобы CA осуществляли строгий контроль над объектами, которые могут действовать как SOA. В данной структуре предоставлен механизм для поддержки данного требования. В других средах данный контроль не является необходимым, и механизмы для определения объектов, которые могут действовать как SOA в таких средах, находятся вне области применения данной Спецификации.

Данная структура является гибкой и может удовлетворять требованиям различных типов сред.

a) Во многих средах все привилегии будут присвоены непосредственно индивидуальным объектам одним CA, именуемым SOA.

b) В других средах может потребоваться поддержка дополнительной возможности ролей, посредством которой лицам выдаются сертификаты, которые присваивают им различные роли. Таким лицам неявно присваиваются привилегии, связанные с ролью. Привилегии ролей могут сами быть присвоены с сертификате атрибута, выданном самой роли, или при помощи некоторых других средств (например, локально настроенных).

c) Другой дополнительной возможностью данной структуры является поддержка делегирования привилегий. Если делегирование выполнено, SOA присваивает привилегии объекту, которому также разрешено действовать как AA и далее делегировать привилегию. Делегирование может продолжаться через несколько промежуточных AA, до тех пор пока она не будет окончательно присвоена оконечному объекту, который не может делегировать привилегию далее. Промежуточные AA могут иметь или не иметь возможности действовать как заявители привилегии для привилегий, которые они делегируют.

d) В некоторых средах один и тот же физический объект может действовать и как AA, и как CA. Данная двойная логическая роль для одного и того же физического объекта всегда встречается, когда привилегия передается в расширении **subjectDirectoryAttributes** сертификата открытого ключа. В других средах, различные физические объекты действуют как CA и AA. В последнем случае привилегия присваивается при использовании сертификатов атрибутов вместо сертификатов открытых ключей.

Когда сертификаты атрибутов указывают на сертификаты открытых ключей их выдавшим органам и держателям, PKI используется для аутентификации держателей (заявителей привилегии) и проверки цифровых подписей выдавших органов.

В данной Спецификации описаны две модели делегирования. В первой модели делегирования делегирующим привилегию органом является AA, который может выдавать сертификаты, делегируя данную привилегию другим. Вторая модель допускает независимую Службу делегирования (DS), в которой объект выдает сертификаты от имени другого AA (которые может иметь или не иметь возможности сам выдавать AC). Данный DS не может сам действовать как претендент на данную привилегию. Модель DS является особенно уместной в средах, где желательно установление некоторого центрального управления совокупностью привилегий, делегированных в их области. Например, совокупность из одного или нескольких выполняющих делегирование серверов DS дает возможность лучше, чем индивидуальные держатели привилегий, определить при помощи централизованных средств общую совокупности привилегий, делегированных в пределах среды, а также дает возможность соответствующим образом модифицировать решения в области политики и управленческие решения. Для серверов DS возможны две разные модели применения. В одном модели привилегия держателям привилегий присваивается SOA, и эти держатели авторизованы для делегирования данной привилегии другим. Тем не менее, скорее чем выдавать сертификаты атрибутов, которые сами делегируют привилегии, держатель привилегии запрашивает DS от его имени делегировать данную привилегию. DS сам не держит данную привилегию и потому не может действовать как претендент на нее; тем не менее, DS авторизован SOA выдавать сертификаты атрибутов от имени других держателей привилегий. Вторая модель применения похожа на первую с одним исключением. DS фактически является держателем, которому присвоена привилегия, которая должна быть делегирована, но DS не авторизован действовать как претендент на нее, а только как делегирующий орган. В данном случае, в AC, выданном DS SOA должно быть установлено расширение noAssertion. Данный DS называется непрямым выдающим органом.

В обеих моделях применения, SOA выдает атрибуты/привилегии подчиненным AA. Затем AA запрашивает DS выдать подмножество данных атрибутов привилегий другим держателям. Во второй модели применения DS может проверить, что AA делегирует в пределах всех границ, установленных SOA; в первой модели применения DS не может проверить, и зависимая сторона должна будет проверить, что делегирование было выполнено правильно.

13.1 Привилегия в сертификатах атрибутов

Объекты могут получать привилегии двумя способами:

– AA может в одностороннем порядке присвоить привилегию объекту путем создания сертификата атрибутов (возможно, полностью по собственной инициативе или по запросу третьей стороны). Данный сертификат может храниться в общественно доступном хранилище и впоследствии может быть обработан одним или несколькими верификаторами привилегий для принятия решения об авторизации. Все это может случиться без знания объекта или явных действий.

– Альтернативно, объект может запросить привилегию у какого-либо AA. После создания сертификат может быть возвращен (только) запрашивающей стороне, которая явно предоставляет его при запросе доступа к какому-либо защищенному ресурсу.

Отметим, что в обеих процедурах AA должен выполнить тщательную проверку для обеспечения того, что объекту на самом деле следует присвоить данную привилегию. Это может повлечь некоторые внеполосные механизмы, аналогичные сертификации связывания CA идентификационной информации/пары ключей.

PMI, основанная на сертификатах атрибутов, подходит для сред, где выполняется любое из перечисленного:

– за присвоение определенной привилегии держателю ответственным является объект, отличный от объекта, ответственного за выдачу сертификатов открытых ключей тому же субъекту;
– существует несколько атрибутов привилегий, которые должны быть присвоены держателю несколькими органами;
– срок действия привилегии отличается от периода действия сертификата открытого ключа держателя (как правило, продолжительность жизни привилегии намного меньше); или
– привилегия является действительной только в течение определенных интервалов времени, не синхронных с периодом действия открытого ключа данного пользователя или периодом действия других привилегий.

13.2 Привилегия в сертификатах открытых ключей

В некоторых средах привилегии связаны с субъектом путем действий CA. Такая привилегия может быть помещена непосредственно в сертификаты открытых ключей (таким образом повторно используя многое из уже созданной инфраструктуры), вместо того, чтобы выдавать сертификаты атрибутов. В таких случаях привилегия включается в расширение **subjectDirectoryAttributes** сертификата открытого ключа.

Данный механизм подходит для сред, где выполняется одно или несколько из перечисленного:

– один и тот же физический объект действует и как CA, и как AA;
– срок действия привилегии выровнен со сроком действия открытого ключа, включенного в сертификат;
– делегирование привилегий не разрешается; или
– делегирование разрешается, но для любого отдельного делегирования все привилегии в сертификате (в расширении **subjectDirectoryAttributes**) имеют те же параметры делегирования, и все расширения, соответствующие делегированию, применяются в равной степени ко всем привилегиям в сертификате.

14 Модели PMI

14.1 Общая модель

Общая модель управления привилегиями состоит из трех объектов: объект, заявитель привилегии и верификатор привилегии.

Объект может быть защищаемым ресурсом, например, в приложении управления доступом. Защищаемый ресурс именуется объектом. Данный тип объекта имеет методы, которые могут быть вызваны (например, объект может быть брандмауэром с методом объекта "разрешить запись", или объект может быть файлом в файловой системе с методами объекта читать, записывать и исполнять). Другой тип объекта в данной модели может быть объектом, подписанным в приложении фиксации авторства.

Заявителем привилегий является объект, держащий определенную привилегию и заявляет свои привилегии для определенного контекста использования.

Верификатор привилегий является объект, который выполняет определение того, являются ли заявленные привилегии достаточными для заданного контекста использования.

Определение прохождения/неудачи, выполняемое верификатором привилегий, зависит от четырех условий:

– привилегии заявителя привилегий;
– местной политики привилегий;
– текущие переменные среды, если значимы; и
– чувствительность метода объекта, если значим.

Привилегия держателя привилегий отражает степень доверия, помещенного в данного держателя органом, выдавшим сертификат, что держатель привилегий будет придерживаться тех аспектов политики, которые не были внедрены при помощи технических средств. Данная привилегия инкапсулируется в сертификат(ы) атрибутов держателя привилегии (или расширение **subjectDirectoryAttributes** его сертификата открытого ключа), который может быть предоставлен

верификатору привилегий в запросе вызова или может быть распространен другими способами, такими как используя Справочник. Кодифицирование привилегии выполняется при использовании структуры **Attribute**, содержащей **AttributeType** и **SET OF AttributeValue**. Некоторые типы атрибутов, используемые для определения привилегий, могут иметь очень простой синтаксис, такие как один **INTEGER** или **OCTET STRING**. Другие могут иметь более сложные синтаксисы. Пример приведен в Приложении D.

Политика привилегий определяет степень привилегии, которая считается достаточной для заданной чувствительности метода объекта или контекста использования. Политика привилегий должна быть защищена на целостность и подлинность. Для передачи политики существует несколько возможностей. Одним из предельных вариантов является мысль, что политика в действительности вообще не передается, а просто определяется и всегда хранится только локально в среде верификатора привилегий. Другим предельным вариантом является мысль, что некоторые политики являются универсальными и должны передаваться и быть известным всем объектам в системе. Между этими пределами находится много оттенков разновидностей. Компоненты схемы для хранения информации о политике привилегий в Справочнике определяются в данной Спецификации.

Политика привилегий определяет порог принятия заданного набора привилегий. Если определить точно, это когда верификатор привилегий должен заключить, что представленный набор привилегий является "достаточным", а именно может предоставить доступ (к запрашиваемому объекту, ресурсу, приложению и т. д.) заявителю привилегий.

Синтаксис для определения политики привилегий не стандартизируется в данной Спецификации. В Приложении D содержится несколько примеров синтаксисов, которые могут использоваться для этой цели. Тем не менее, это просто примеры. Для этой цели может использоваться любой синтаксис, включая пустой текст. Независимо от синтаксиса, используемого для определения политики привилегий, каждый экземпляр политики привилегий должен быть однозначно определен. Для данной цели используются идентификаторы объектов.

PrivilegePolicy ::= OBJECT IDENTIFIER

Переменные среды, если значимы, охватывает аспекты политики, требуемые для определения прохождения/неудачи (например, время дня или текущий платежный баланс), доступные верификатору привилегий при помощи локальных средств. Представление переменных среды является полностью локальной задачей.

Чувствительность метода объекта, если значима, может отражать атрибуты обрабатываемого документа или запроса, такие как денежная стоимость перевода средств, которую он намерен авторизовать, или конфиденциальность содержимого документа. Чувствительность метода объекта может быть явно закодирована в соответствующей метке безопасности или в сертификате атрибута, который держит метод объекта, или может быть неявно инкапсулирован в структуру и содержимое соответствующего объекта данных. Она может быть закодирована одним из нескольких различных способов. Например, она может быть закодирована вне границ PMI в метке X.411, связанной с документом, в полях взаимного обмена EDIFACT, или жестко закодирована в приложении верификатора привилегий. Альтернативно, это может выполняться в пределах PMI, в сертификате атрибута, связанного с методом объекта. Для некоторых контекстов использования чувствительность метода объекта не используется.

Не обязательно существует отношения связывания между верификатором привилегий и каким-либо определенным AA. Так же, как держатели привилегий могут иметь сертификаты атрибутов, выданные им несколькими различными AA, верификаторы привилегий для предоставления доступа к определенному ресурсу могут принимать сертификаты, выданные многочисленными AA, которые не должны иерархически подчиняться друг другу.

Структура сертификатов атрибутов может использоваться для управления привилегиями различных типов и для различных целей. Термины, использованные в данной Спецификации, такие как заявитель привилегий, верификатор привилегий и т. д., не зависят от определенного приложения или использования.

14.1.1 PMI в контексте управления доступом

Для контроля доступа существует стандартная структура (Рек. МСЭ-Т X.812 | ИСО/МЭК 10181-3), которая определяет соответствующий набор терминов, характерных для приложения управления доступом. Здесь предоставлено отображение характерных терминов, использованных в данной Спецификации, в термины структуры контроля доступа для прояснения отношения данной модели и той Спецификации.

Заявитель привилегий в данной Спецификации действовал бы в роли 'инициатора' в структуре управления доступом.

Заявитель привилегий в данной Спецификации действовал бы в роли 'функции решения об управления доступом (ADF)' в структуре управления доступом.

Метод объекта, для которого заявляется привилегия, в данной Спецификации соответствовал бы 'цели', определенной в структуре управления доступом.

Переменные среды в данной Спецификации соответствовали бы 'контекстуальной информации' в структуре управления доступом.

Политика привилегий, обсуждаемая в данной Спецификации, могла включать 'политику управления доступом' и 'правила политики управления доступом', как определено в структуре управления доступом.

Данная модель позволяет достаточно плавно наложить PMI на существующую структуру защищаемых ресурсов. В особенности, действия верификатора привилегий в качестве шлюза к чувствительному методу объекта, предоставление или запрет запросов для вызова данного метода объекта позволяет защищать объект с малым воздействием на сам объект или без этого воздействия. Верификатор привилегий тщательно отбирает все запросы, и только правильно авторизованные получают доступ к соответствующим методам объектов.

14.1.2 PMI в контексте фиксации авторства

Для фиксации авторства существует стандартная структура (Рек. МСЭ-Т X. 813 | ИСО/МЭК 10181-4), которая определяет соответствующий набор терминов, характерных для фиксации авторства. Здесь предоставлено отображение характерных терминов, использованных в данной Спецификации, в термины структуры фиксации авторства для прояснения отношения данной модели и той Спецификации.

Заявитель привилегий в данной Спецификации действовал бы в роли 'субъекта доказательства' или 'источника' в структуре фиксации авторства.

Заявитель привилегий в данной Спецификации действовал бы в роли 'пользователя доказательства' или 'получателя' в структуре фиксации авторства.

Метод объекта, для которого заявляется привилегия, в данной Спецификации соответствовал бы 'цели', определенной в структуре фиксации авторства.

Переменные среды в данной Спецификации соответствовали бы 'дате и времени, когда было сгенерировано или проверено доказательство' в структуре фиксации авторства.

Политика привилегий, обсуждаемая в данной Спецификации, могла включать 'политику безопасности фиксации авторства' в структуре фиксации авторства.

14.2 Модель управления

Модель контроля иллюстрирует, как управление осуществляется над доступом к чувствительному методу объекта. В модель включено пять компонентов: заявитель привилегий, верификатор привилегий, метод объекта, политика привилегий и переменные среды (см. рисунок 3). Заявитель привилегий имеет привилегию, метод объекта имеет чувствительность. Методы, описанные здесь, позволяют верификатору привилегий управлять доступом к методу объекта при помощи заявителя привилегий в соответствии с политикой привилегий. Как привилегия, так и чувствительность могут быть многозначными параметрами.

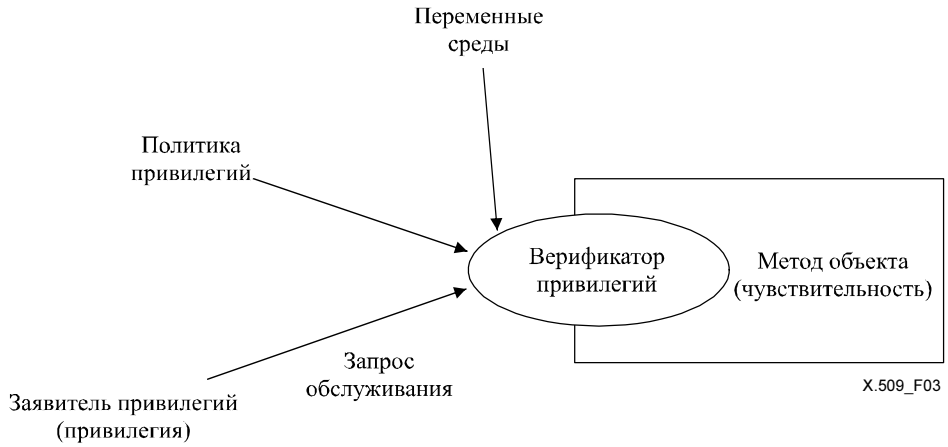


Рисунок 3 – Модель управления

Заявитель привилегий может быть объектом, определенным сертификатом открытого ключа, или выполнимым объектом, определенным на основе обзора изображения диска и т. д.

14.3 Модель делегирования

В некоторых средах может существовать необходимость делегирования привилегий, тем не менее, она является дополнительным аспектом структуры и не требуется во всех средах. В модель делегирования включено четыре компонента: верификатор привилегий, SOA, другие AA и заявитель привилегий (см. рисунок 4).

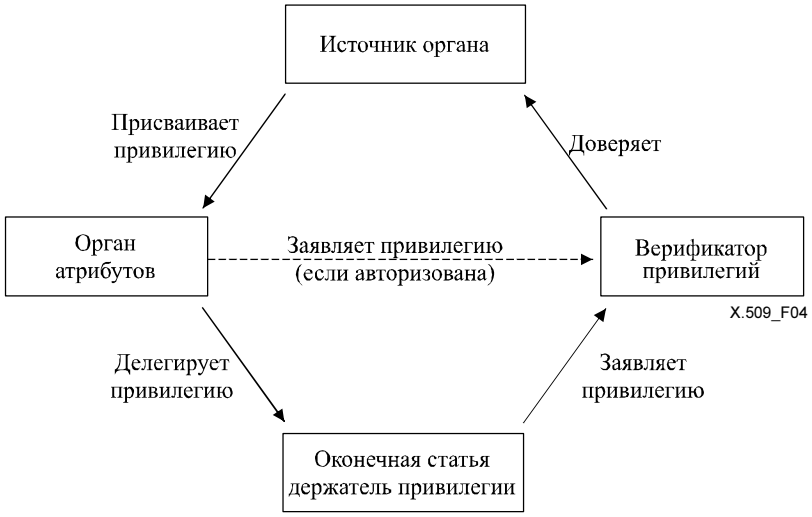


Рисунок 4 – Модель делегирования

Как и в средах, где делегирование не используется, SOA является начальным органом, выдающим сертификаты, который присваивает привилегии держателям привилегий. Тем не менее, в данном случае SOA авторизует держателя привилегии действовать как AA и делегировать привилегию далее другим объектам путем выдачи сертификатов, содержащих ту же привилегию (или ее подмножество). SOA может наложить ограничения на делегирование, которое может быть выполнено (например, ограничить длину тракта, ограничить пространство имен, в пределах которого может выполняться делегирование). Каждый из таких промежуточных AA может в сертификатах, которые выдает дальнейшим держателям привилегий, авторизовать выполнение дальнейшего делегирования данными держателями, действующими как AA. Универсальным ограничением на делегирования является то, что ни один AA не может делегировать больше привилегий, чем он держит. Делегирующий орган может также дальше ограничить возможность нисходящего потока AA.

При использовании делегирования, верификатор привилегий доверяет SOA делегировать некоторые или все данные привилегии держателям, некоторые из которых могут далее делегировать некоторые или все привилегии другим держателям.

Верификатор привилегий доверяет SOA как органу для заданного набора привилегий для ресурса. Если сертификат заявителя привилегий не является выданных данным SOA, то верификатор привилегий должен определить местоположение тракта делегирования сертификатов от сертификата заявителя привилегии до сертификата, выданного SOA. Проверка подлинности данного тракта делегирования включает проверку того, что каждый AA имеет достаточные привилегии и был должным образом авторизован для делегирования данных привилегий.

Для случая, когда привилегии передаются при помощи сертификатов атрибутов, тракт делегирования отличается от тракта проверки подлинности сертификатов, используемого для проверки подлинности сертификатов открытых ключей объектов, включенных в процесс делегирования. Тем не менее, качество подлинности, предлагаемое процессом проверки подлинности сертификата открытого ключа, должно быть соизмеримым с чувствительностью защищаемого метода объекта.

Тракт делегирования должен состоять или полностью из сертификатов атрибутов, или полностью из сертификатов открытых ключей. Делегирующий орган, получающий привилегию в сертификате атрибута, может делегировать только, если авторизован, путем выдачи последующих сертификатов атрибутов. Аналогично, делегирующий орган, получающий привилегию в сертификате открытого ключа, если авторизован, может делегировать только путем выдачи последующих сертификатов открытых ключей. Делегировать привилегию могут только AA. Оконечные объекты не могут.

14.4 Модель ролей

Роли предоставляют средство для непрямого присвоения привилегий лицам. Лицам выдаются сертификаты присвоения ролей, которые присваивают им одну или несколько ролей при помощи атрибута роли, содержащегося в сертификате. Определенные привилегии назначаются имени роли путем сертификатов спецификаций роли, скорее чем индивидуальным держателям привилегий путем сертификатов атрибутов. Данный уровень косвенности делает возможным, например, обновление привилегий, присвоенных роли, без воздействия на сертификаты, которые присваивают роли лицам. Сертификаты присвоения ролей могут быть сертификатами атрибутов или сертификатами открытых ключей. Сертификаты спецификации ролей могут быть сертификатами атрибутов, но не сертификатами открытых ключей. Если сертификаты спецификации ролей не используются, то присвоение привилегий роли может выполняться при помощи других средств (например, может быть локально настроено у верификатора привилегий).

Возможно все из перечисленного ниже:

- любое количество ролей может быть определено любым AA%;
- сама роль и члены роли могут определяться и управляться раздельно, разными AA;
- членство роли, как любая другая привилегия, может делегироваться; и

– ролям и членству может быть присвоена любая продолжительность жизни.

Если сертификат присвоения роли является сертификатом атрибута, то атрибут **role** содержится в компоненте **attributes** сертификата атрибута. Если сертификат присвоения роли является сертификатом открытого ключа, то атрибут **role** содержится в расширении **subjectDirectoryAttributes**. В последнем случае, любые дополнительные привилегии, содержащиеся в сертификате открытого ключа, представляют собой привилегии, непосредственно назначенные субъекту сертификата, но не привилегии, назначенные роли.

Таким образом, заявитель привилегии может представить сертификат присвоения роли верификатору привилегий, демонстрируя только то, что у заявителя привилегий есть определенная роль (например, "управляющий" или "покупатель"). Верификатор привилегий может знать, *априори*, или может быть вынужден выяснить при использовании других средств привилегии, связанные с заявленной ролью, чтобы принять решение о прохождении/неудаче авторизации. Для этой цели может использоваться сертификат спецификации роли.

Верификатор привилегий должен обладать пониманием привилегий, определенных для роли. Присвоение данных привилегий роли может быть выполнено в пределах PMI в сертификате спецификации роли или вне PMI (например, локально настроено). Если привилегии роли заявляются в сертификате присвоения роли, то механизмы для связывания данного сертификата с соответствующим сертификатом присвоения роли для заявителя привилегий представлены в данной Спецификации. Сертификат спецификации роли не может делегироваться любому другому объекту. Орган, выдавший сертификат присвоения роли, может быть независимым от органа, выдавшего сертификат спецификации роли, и они могут управляться (истекать, аннулироваться и т.п.) полностью независимо. Один и тот же сертификат (сертификат атрибута или сертификат открытого ключа) может быть сертификатов присвоения роли, а также содержать присвоение других привилегий непосредственно тому же лицу. Тем не менее, сертификат спецификации роли должен быть отдельным сертификатом.

ПРИМЕЧАНИЕ. – Использование ролей в структуре авторизации может повысить сложность обработки тракта, так как такая функциональная возможность главным образом определяет другой тракт делегирования, которому необходимо следовать. Тракт делегирования для сертификата присвоения роли может включать различные AA и может быть независимым от AA, который выдал сертификат спецификации роли.

14.4.1 Атрибут роли

Спецификация типов атрибутов привилегий в общем случае является вопросом, связанным с конкретным применением, и находится вне области применения данной Спецификации. Единственным исключением является атрибут, определенный здесь для присвоения держателя роли. Спецификация значений для атрибута роли находится вне области применения данной Спецификации.

```
role  ATTRIBUTE  ::= {
      WITH SYNTAX          RoleSyntax
      ID                   id-at-role }

RoleSyntax ::= SEQUENCE {
roleAuthority       [0]    GeneralNames    OPTIONAL,
roleName            [1]    GeneralName }
```

Данный атрибут привилегий использовался бы для заполнения поля **attributes** сертификата присвоения роли. Если сертификат присвоения роли является сертификатом открытого ключа, то атрибут использовался бы для заполнения расширения **subjectDirectoryAttributes** данного сертификата открытого ключа.

roleAuthority, при наличии, определяет узнанный орган, ответственный за выдачу сертификата спецификации роли.

Если **roleAuthority** присутствует и верификатор привилегий использует сертификат спецификации роли для определения привилегий, присвоенных роли, то по меньшей мере одно из имен в **roleAuthority** должно присутствовать в поле **issuer** данного сертификата спецификации роли. Если верификатор привилегий использует средства, отличные от сертификата спецификации роли для определения привилегий, присвоенных роли, то механизмы обеспечения того, что эти привилегии были присвоены органом, названным в данном компоненте, находятся вне области применения данной Спецификации

Если **roleAuthority** отсутствует, то идентификационная информация ответственного органа должна определяться при помощи других средств. Расширение **roleSpecCertIdentifier** в сертификате присвоения роли представляет собой один из способов для получения данного связывания, когда сертификат спецификации роли использовался для присвоения привилегий роли.

Компонент **roleName** определяет роль, которой присвоен держатель сертификата присвоения роли, содержащего данный атрибут. Если верификатор привилегий использует сертификат спецификации роли для определения привилегий, присвоенных роли, то данное имя роли также должно появиться в поле **holder** сертификата спецификации роли.

14.5 Атрибут информации привилегии XML

Спецификация привилегий в общем случае является вопросом, связанным с конкретным применением, и находится вне области применения данной Спецификации. Несмотря на то, что данный атрибут не определяет какую-либо определенную информацию о привилегии, он предоставляет атрибут контейнера, в котором могут передаваться привилегии, закодированные в XML в сертификатах атрибутов.

```
xmlPrivilegeInfo   ATTRIBUTE  ::= {
  WITH SYNTAX      UTF8String – содержит информацию о привилегии закодированную в XML
  ID               id-at-xMLPrivilegeInfo }
```

Схема XML для типа атрибута роли может быть определена либо с ASN.1, либо с XSD.

XML, содержащийся в **UTF8String**, должен быть автоматически определяющимся.

Ниже представлена схема ASN.1, определяющая тип атрибута роли XML. Ниже следует спецификация XSD для того же самого типа атрибута, а также примерный экземпляр XML. Данный примерный экземпляр является действительным экземпляром как для экземпляра схемы ASN.1, так и схемы XSD, и может быть проверен средствами либо ASN.1, либо XSD.

Примерная схема определяет атрибут роли с ID, выдающим органом и именем роли.

```
CERTIFICATE-ATTRIBUTE DEFINITIONS ::=
BEGIN
  Role ::= [UNCAPITALIZED] SEQUENCE {
     id            [ATTRIBUTE] XML-ID,
     authorities     SEQUENCE (1..MAX) OF
                    authority UTF8String,
     name           UTF8String }

  XML-ID  ::= UTF8String
END
```

Следующая схема XSD является альтернативным (точно эквивалентным) определением:

```
<schema xmlns="http://www.w3.org/2000/08/XMLSchema">
 <element name="role">
  <attribute name="id" type="ID"/>
  <complexType>
   <sequence>
    <element name="authorities">
     <complexType>
      <sequence>
       <element name="authority" type="string" minOccurs="1" maxOccurs="*"/>
      </sequence>
     </complexType>
    </element>
    <element name="name" type="string"/>
   </sequence>
  </complexType>
 </element>
</schema>
```

Примером экземпляра, соответствующего вышеописанным определениям схемы, которая была бы значением типа атрибута **xMLPrivilegeInfo**, мог бы быть:

```
<role id="123" xmlns="http://www.example.org/certificates/attribute">
<authorities>
<authority>Fictitious Organization</authority>
</authorities>
<name>manager</name>
</role>
```

15 Расширения сертификатов управления привилегиями

Следующие расширения сертификатов могут быть включены в сертификаты для целей управления привилегиями. Вместе с определением самих расширений, также предоставлены правила для типов сертификатов, в которых может присутствовать расширение.

Исключая расширение идентификатора SOA, любое из расширений, которое может быть включено в сертификат открытого ключа, должно быть включено, только если данный сертификат открытого ключа присваивает привилегии своему субъекту (т. е. должно присутствовать расширение **subjectDirectoryAttributes**). Если какое-либо из этих расширений присутствует в сертификате открытого ключа, данное расширение применяется ко ВСЕМ привилегиям, присутствующим в расширении **subjectDirectoryAttributes**.

Списки аннулирования, использованные для опубликования уведомлений об аннулировании сертификатов атрибутов (ACRL и AARL), могут содержать любые расширения CRL или записей CRL, как определено в использовании CRL и CARL во 2-м Разделе данной Спецификации.

В данном пункте определяются расширения в следующих областях:

a) *Основное управление привилегиями*: Данные расширения сертификатов передают информацию, соответствующую заявлению привилегии.

b) *Аннулирование привилегий*: Данные расширения сертификатов передают информацию относительно местоположения информации о статусе аннулирования.

c) *Источник органа*: Данные расширения сертификатов относятся к доверенному источнику присвоения привилегий при помощи верификатора для заданного ресурса.

d) *Роли*: Данные расширения сертификатов передают информацию относительно местоположения связанных сертификатов спецификаций ролей.

e) *Делегирование*: Данные расширения сертификатов позволяют установить ограничения на последующее делегирование присвоенных привилегий.

15.1 Расширения основного управления привилегиями

15.1.1 Требования

Следующие требования относятся к основному управлению привилегиями:

a) выдающие органы должны иметь возможность налагать ограничения на время, в течение которого привилегия может быть завялена;

b) выдающие органы должны иметь возможность направлять сертификаты атрибутов на определенные сервера/услуги;

c) выдающим органам может быть необходимо передавать информацию, предназначенную для отображения заявителям привилегий и/или верификаторам привилегий с использованием сертификата;

d) выдающим органам может быть необходимо налагать ограничения на политики привилегий, с которыми может использоваться присвоенная привилегия.

15.1.2 Поля основных расширений управления привилегиями

Определяются следующие поля расширений:

a) *спецификация времени*;

b) *направление информации*;

c) *уведомление пользователя*;

d) *приемлемые политики привилегий*;

e) *непрямой выдающий орган*;

f) *нет заявления.*

15.1.2.1 Расширение спецификации времени

Расширение спецификации времени может использоваться AA для ограничения определенных периодов времени, в течение которых привилегия, присвоенная в сертификате, содержащем данное расширение, может быть заявлена держателем привилегии. Например, AA может выдать сертификат, присваивающий привилегии, который может быть заявлен только с понедельника по пятницу с 9 час. 00 мин. до 17 час. 00 мин. Другим примером, в случае делегирования, может быть управляющий, делегирующий право подписи подчиненному на время своего отъезда в отпуск.

Данное поле определяется следующим образом:

```
timeSpecification EXTENSION  ::= {
    SYNTAX              TimeSpecification
    IDENTIFIED BY       id-ce-timeSpecification }
```

Данное расширение может присутствовать в сертификатах атрибутов или сертификатах открытых ключей, выданных AA, включая SOA, объектам, которые могут действовать как заявители привилегий, включая другие AA и оконечные объекты. Данное расширение не должно включаться в сертификаты, содержащие расширение идентификатора SOA, или в сертификаты, выданные AA, которые не могут также действовать как заявители привилегий.

Если данное расширение присутствует в сертификате, выданном объекту, являющемуся AA, оно применяется только к заявлению тем объектом привилегий, содержащихся в сертификате. Оно не влияет на период времени, в течение которого AA может выдавать сертификаты.

Так как данное расширение эффективно определяет уточнение периода действия его содержащего сертификата, данное расширение должно быть помечено как критическое (т. е. выдающий орган путем включения данного расширения явно определяет, что присвоение привилегий является недействительным вне определенного времени).

Если данное расширение присутствует, но не понимается верификатором привилегий, сертификат должен быть отклонен.

15.1.2.1.1 Соответствие спецификаций времени

Правило соответствия спецификации времени сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
timeSpecificationMatch MATCHING-RULE ::= {
    SYNTAX          TimeSpecification
    ID              id-mr-timeSpecMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **timeSpecification** и если компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам сохраненного значения.

15.1.2.2 Расширение направления информации

Расширение направления информации делает возможным направление сертификата атрибута на определенный набор серверов/услуг. Сертификат атрибута, содержащий данное расширение, должен иметь возможность использоваться только определенными серверами/услугами.

Данное поле определяется следующим образом:

```
targetingInformation EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF Targets
    IDENTIFIED BY       id-ce-targetInformation }

Targets    ::=      SEQUENCE SIZE (1..MAX) OF Target

Target     :  :=    CHOICE {
    targetName          [0]     GeneralName,
    targetGroup         [1]     GeneralName,
    targetCert          [2]     TargetCert }

TargetCert      ::= SEQUENCE {
    targetCertificate   IssuerSerial,
    targetName          GeneralName OPTIONAL,
    certDigestInfo      ObjectDigestInfo OPTIONAL }
```

Компонент **targetName**, при наличии, предоставляет имя целевых серверов/услуг, на которые направлен сертификат, содержащий атрибут.

Компонент **targetGroup**, при наличии, предоставляет имя целевой группы, на которую направлен сертификат, содержащий атрибут. Определение членства цели в **targetGroup** находится вне области применения данной Спецификации.

Компонент **targetCert**, при наличии, определяет целевые сервера/услуги путем обращения к их сертификату.

Данное расширение может присутствовать в сертификатах атрибутов, выданных AA, включая SOA, объектам, которые могут действовать как заявители привилегий, включая другие AA и оконечные объекты. Данное расширение не должно включаться в сертификаты открытых ключей или сертификаты атрибутов, выданные AA, которые не могут также действовать как заявители привилегий.

Если данное расширение присутствует в сертификате атрибута, выданном объекту, являющемуся AA, оно применяется только к заявлению тем объектом привилегий, содержащихся в сертификате. Оно не влияет на возможность AA выдавать сертификаты.

Данное расширение всегда является критическим.

Если данное расширение присутствует, но верификатора привилегий нет среди определенных объектов, то сертификат атрибута должен быть отклонен.

Если данное расширение не присутствует, то сертификат атрибута не направлен и может быть принят любым сервером.

15.1.2.3 Расширение уведомления пользователя

Расширение уведомления пользователя позволяет АА включить уведомление, которое должно отображаться держателю при заявлении привилегии, и/или верификатору привилегий при использовании сертификата атрибута, содержащего данное расширение.

Данное поле определяется следующим образом:

```
userNotice  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF UserNotice
    IDENTIFIED BY       id-ce-userNotice }
```

Данное расширение может присутствовать в сертификатах атрибутов или сертификатах открытых ключей, выданных АА, включая SOA, объектам, которые могут действовать как заявители привилегий, включая другие АА и оконечные объекты. Данное расширение не должно включаться в сертификаты, содержащие расширение идентификатора SOA, или в сертификаты, выданные АА, которые не могут также действовать как заявители привилегий.

Если данное расширение присутствует в сертификате, выданном объекту, являющемуся АА, оно применяется только к заявлению тем объектом привилегий, содержащихся в сертификате. Оно не влияет на возможность АА выдавать сертификаты.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим.

Если данное расширение помечено как критическое, то уведомление пользователя должно отображаться верификатору привилегий при каждом заявлении привилегии. Если заявитель привилегий предоставляет сертификат атрибута верификатору привилегий (т. е. верификатор привилегий не ищет его непосредственно в хранилище), то уведомление пользователя также должно отображаться заявителю привилегий.

Если данное расширение помечено как некритическое, то привилегия, заявленная в сертификате, может предоставляться верификатором привилегий независимо от того, отображались ли уведомления пользователя заявителю привилегий и/или верификатору привилегий.

15.1.2.4 Приемлемое расширение политик привилегий

Поле приемлемого расширения политик привилегий используется для ограничения заявления присвоенных привилегий использованием с определенным набором политик привилегий.

Данное поле определяется следующим образом:

```
acceptablePrivilegePolicies EXTENSION ::= {
    SYNTAX              AcceptablePrivilegePoliciesSyntax
    IDENTIFIED BY       id-ce-acceptablePrivilegePolicies }

AcceptablePrivilegePoliciesSyntax       ::=     SEQUENCE SIZE (1..MAX) OF PrivilegePolicy
```

Данное расширение может присутствовать в сертификатах атрибутов или сертификатах открытых ключей, выданных АА, включая SOA, другим АА и оконечным объектам. Если данное расширение содержится в сертификате открытого ключа, оно относится только к возможности субъекта действовать как заявитель привилегий для привилегий, содержащихся в расширении **subjectDirectoryAttributes**.

При наличии, данное расширение должно быть помечено как критическое.

Если данное расширение присутствует и верификатор привилегий понимает его, верификатор должен обеспечить, что политика привилегий, с которой сравниваются данные привилегии, является одной из политик, определенных в данном расширении.

Если данное расширение но не понимается верификатором привилегий, сертификат должен быть отклонен.

15.1.2.5 Расширение непрямого выдающего органа

В некоторых средах привилегия может делегироваться косвенно. В таких случаях делегирующий орган запрашивает, чтобы АА выдал сертификат, делегирующий привилегию, от его имени другому объекту. Поле непрямого выдающего органа используется либо в сертификате атрибута, либо в сертификате открытого ключа, выданного SOA для АА. Наличие данного расширения означает, что АА субъекта авторизован данным SOA действовать как его заместитель и выдавать сертификаты, которые делегируют привилегии, от имени других делегирующих органов.

```
indirectIssuer EXTENSION ::= {
  SYNTAX              BOOLEAN
  IDENTIFIED BY       id-ce-indirectIssuer }
```

Данное расширение всегда является некритическим.

Правило соответствия непрямого выдающего органа сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
indirectIssuerMatch MATCHING-RULE  ::= {
SYNTAX  BOOLEAN
ID id-mr-indirectIssuerMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **timeSpecification** и если значение, присутствующее в представленном значении, соответствует сохраненному значению.

15.1.2.6 Расширение нет заявления

При наличии, данное расширение указывает, что держатель AA не может заявлять привилегии, указанные в атрибутах AC. Данное поле может быть вставлено только в AC AA, но не в AC оконечного объекта. При наличии, данное расширение всегда должно быть помечено как критическое.

```
noAssertion EXTENSION ::= {
SYNTAX NULL
IDENTIFIED BY  id-ce-noAssertion }
```

15.2 Расширения аннулирования привилегий

15.2.1 Требования

Следующие требования относятся к аннулированию сертификатов атрибутов:

a) чтобы контролировать размеры CRL, необходима возможность присваивать подмножества из множества всех сертификатов, выданных одним AA различным CRL;

b) органы, выдающие сертификаты атрибутов, должны иметь возможность указывать в сертификате атрибута, что для данного сертификата нет доступной информации об аннулировании.

15.2.2 Поля расширений аннулирования привилегий

Определяются следующие поля расширений:

a) *точки распределения CRL*;

b) *нет информации об аннулировании*.

15.2.2.1 Расширение точек распределения CRL

Расширение точек распределения CRL определено во 2-м Разделе данной Спецификации для использования в сертификатах открытых ключей. Данное поле также может быть включено в сертификат атрибута. Оно может присутствовать в сертификатах, выданных AA, включая SOA, а также в сертификатах, выданных оконечным объектам.

При наличии в сертификате, верификатор привилегий должен обработать данное расширение тем же способом, как описано во 2-м Разделе для сертификатов открытых ключей.

15.2.2.2 Расширение нет информации об аннулировании

В некоторых средах (например, где сертификаты атрибутов выдаются с очень коротким периодом действия), может не быть необходимости аннулировать сертификаты. AA может использовать данное расширение для указания, что для данного сертификата атрибута информация о статусе аннулирования не предоставляется. Данное поле определяется следующим образом:

```
noRevAvail EXTENSION ::=  {
   SYNTAX              NULL
   IDENTIFIED BY       id-ce-noRevAvail }
```

Данное расширение может присутствовать в сертификатах, выданных AA, включая SOA, оконечным объектам. Данное расширение не должно включаться в сертификаты открытых ключей или сертификаты атрибутов, выданные AA.

Данное расширение всегда является некритическим.

Если данное расширение присутствует в сертификате атрибута, верификатор привилегий не должен искать информацию о статусе аннулирования.

15.3 Источник расширения органа

15.3.1 Требования

Следующие требования относятся к источникам органов:

a) в некоторых средах существует необходимость жесткого контроля CA объектов, которые могут действовать как SOA;

b) существует необходимость создания действующих определений синтаксисов и правил доминирования для атрибутов привилегий, доступных при помощи ответственного SOA.

15.3.2 Поля расширений SOA

Определяются следующие поля расширений:

a) *идентификатор SOA*;

b) *дескриптор атрибута*.

15.3.2.1 Расширение идентификатора SOA

Расширение идентификатора SOA указывает, что субъект сертификата может действовать как SOA для целей управления привилегиями. Как таковой, субъект сертификата может определять атрибуты, которые присваивают привилегии, выдают сертификаты дескрипторов атрибутов этим атрибутам и используют открытый ключ, соответствующий сертифицируемому открытому ключу, для выдачи сертификатов, присваивающих привилегии держателям. Данные последующие сертификаты могут быть сертификатами атрибутов или сертификатами открытых ключей с расширением **subjectDirectoryAttributes**, содержащим привилегии.

В некоторых средах данное расширение не требуется, и для определения объектов, которые могут действовать как SOA, могут использоваться другие механизмы. Данное расширение требуется только в средах, где требуется жесткий централизованный контроль CA для управления объектами, действующими как SOA.

Данное поле определяется следующим образом:

```
sOAIdentifier EXTENSION  ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-sOAIdentifier }
```

Если данное расширение не присутствует в сертификате, то способность субъекта/держателя действовать как SOA должна определяться при помощи других средств.

Данное поле может присутствовать только в сертификате открытого ключа, выданного SOA. Оно не должно включаться в сертификаты атрибутов или сертификаты открытых ключей, выданные другим AA или держателям привилегий оконечных объектов.

Перекрестная сертификация применяется только к сертификатам открытых ключей, но не к сертификатам атрибутов. Таким образом, перекрестный сертификат, выданный CA, выдавшему сертификат, содержащий расширение идентификатора SOA, не обеспечивает переходного доверия к SOA, определенному в данном расширении.

Данное расширение всегда является некритическим.

15.3.2.1.1 Соответствие идентификаторов SOA

Правило соответствия идентификаторов SOA сравнивает представленное значение и значение атрибута типа **Certificate**.

```
sOAIdentifierMatch MATCHING-RULE ::= {
  SYNTAX     NULL
  ID         id-mr-sOAIdentifierMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение идентификатора SOA.

15.3.2.2 Расширение дескриптора атрибута

Определение атрибута привилегии и правила доминирования, руководящих последующим делегированием данной привилегии, необходимы верификаторам привилегий для обеспечения того, что авторизация выполнена правильно. Данные определения и правила могут предоставляться верификаторам привилегий разными способами, находящимися вне области применения данной Спецификации (например, могут быть локально настроены верификатором привилегий).

Данное расширение предоставляет один из механизмов, который может использоваться SOA для создания определений атрибутов привилегий и соответствующих правил доминирования, доступных для верификаторов привилегий. Сертификат атрибута, содержащий данное расширение, называется сертификатом дескриптора атрибута и представляет собой особый тип сертификата атрибута. Будучи синтаксически идентичным **AttributeCertificate**, сертификат дескриптора атрибута:

- содержит пустое **SEQUENCE** в поле **attributes**;
- является автоматически выданным сертификатом (т. е. выдавший орган и держатель являются одним и тем же объектом); и
- включает расширение дескриптора атрибута.

Данное поле определяется следующим образом:

```
attributeDescriptor EXTENSION  ::= {
    SYNTAX              AttributeDescriptorSyntax
    IDENTIFIED BY       {id-ce-attributeDescriptor } }
```

```
AttributeDescriptorSyntax  ::= SEQUENCE {
    identifier                  AttributeIdentifier,
    attributeSyntax             OCTET STRING (SIZE(1..MAX)),
    name                [0]     AttributeName   OPTIONAL,
    description         [1]     AttributeDescription    OPTIONAL,
    dominationRule              PrivilegePolicyIdentifier}

AttributeIdentifier  ::= ATTRIBUTE.&id({AttributeIDs})

AttributeIDs ATTRIBUTE  ::= {...}

AttributeName  ::= UTF8String(SIZE(1..MAX))

AttributeDescription  ::= UTF8String(SIZE(1..MAX))

PrivilegePolicyIdentifier ::=   SEQUENCE {
    privilegePolicy             PrivilegePolicy,
    privPolSyntax               InfoSyntax }
```

Компонент **identifier** значения расширения **attributeDescriptor** является идентификатором объекта, определяющим тип атрибута.

Компонент **attributeSyntax** содержит определение ASN.1 синтаксиса атрибута. Такое определение ASN.1 должно быть задано как определенное для информационного компонента рабочего атрибута Правил соответствия, определенного в Рек. МСЭ-Т X.501 | ИСО/МЭК 9594-2.

Компонент **name** дополнительно содержит удобное для пользователя имя, по которому может быть узнан атрибут.

Компонент **description** дополнительно содержит удобное для пользователя описание атрибута.

Компонент **dominationRule** определяет, для атрибута, что для делегируемой привилегии значит быть "меньше чем" соответствующая привилегия, удерживаемая делегирующим органом. Компонент **privilegePolicy** определяет экземпляр политики привилегий, которая содержит правила, при помощи идентификатора объекта. Компонент **privPolSyntax** содержит либо саму политику привилегий, либо указатель на местоположение, где она может быть расположена. Если включен указатель, то также может быть включен дополнительный хэш политики привилегий для возможности проверки целостности по адресуемой политике привилегий.

Данное расширение может присутствовать только в сертификатах дескрипторов атрибутов. Данное расширение не должно присутствовать в сертификатах открытых ключей или сертификатах атрибутов, отличные от автоматически выданных сертификатов SOA.

Данное расширение всегда должно являться некритическим.

Сертификат дескриптора атрибута, созданный SOA в момент создания/определения соответствующего типа атрибутов, является средством, при помощи которого может быть понято и внедрено в структуру универсальное ограничение делегирования "вниз". В Справочнике сертификаты атрибутов, содержащие данное расширение, должны храниться в атрибуте **attributeDescriptorCertificate** записи справочника SOA.

15.3.2.2.1 Соответствие дескрипторов атрибутов

Правило соответствия дескрипторов атрибутов сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
attDescriptor MATCHING-RULE  ::= {
    SYNTAX          AttributeDescriptorSyntax
    ID              id-mr-attDescriptorMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **attributeDescriptor** и если компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам сохраненного значения.

15.4 Расширения ролей

15.4.1 Требования

Следующие требования относятся к ролям:

– Если сертификат является сертификатом присвоения ролей, верификатор привилегий должен иметь возможность определить местоположение соответствующего сертификата спецификации роли, содержащего определенные привилегии, присвоенные самой роли.

15.4.2 Поля расширений ролей

Определяется следующее поле расширения:

– *Идентификатор сертификата спецификации роли*.

15.4.2.1 Расширение идентификатора сертификата спецификации роли

Данное расширение может использоваться АА в качестве указателя на сертификат спецификации роли, содержащий присвоение привилегий роли. Оно может присутствовать в сертификате присвоения роли (т. е. сертификате, содержащем атрибут **role**).

Верификатор привилегий при работе с сертификатом присвоения роли должен получить набор привилегий данной роли, чтобы определить, прошла ли успешно или неуспешно верификация. Если привилегии были присвоены роли в сертификате спецификации роли, данное поле может использоваться для определения местоположения данного сертификата.

Данное поле определяется следующим образом:

```
roleSpecCertIdentifier EXTENSION ::=
    {
    SYNTAX              RoleSpecCertIdentifierSyntax
    IDENTIFIED BY       { id-ce-roleSpecCertIdentifier }
}

RoleSpecCertIdentifierSyntax  ::= SEQUENCE SIZE (1..MAX) OF RoleSpecCertIdentifier

RoleSpecCertIdentifier  ::= SEQUENCE {
    roleName                    [0]     GeneralName,
    roleCertIssuer              [1]     GeneralName,
    roleCertSerialNumber        [2]     CertificateSerialNumber     OPTIONAL,
    roleCertLocator             [3]     GeneralNames                OPTIONAL
    }
```

roleName определяет роль. Данное имя будет таким же, как и **holder** в компоненте сертификата спецификации роли, к которому обращается данное расширение.

roleCertIssuer определяет АА, который выдал адресуемый сертификат спецификации роли.

roleCertSerialNumber, при наличии, содержит порядковый номер сертификата спецификации роли. Отметим, что если присвоенные самой роли привилегии меняются, то роли должен быть выдан новый сертификат спецификации роли. Любые сертификаты, содержащие данное расширение, включая компонент **roleCertSerialNumber**, затем должны быть заменены сертификатами, которые обратились к новому порядковому номеру. Хотя данное поведение требуется в некоторых средах, оно нежелательно во многих других. Как правило, данный компонент будет отсутствовать, делая возможным автоматическое обновление привилегий, присвоенных самой роли, без влияния на сертификаты присвоения ролей.

roleCertLocator, при наличии, содержит информацию, которая может использоваться для определения местоположения сертификата спецификации роли.

Данное расширение может присутствовать в сертификатах присвоения ролей, являющихся сертификатами атрибутов или сертификатами открытых ключей, выданных АА, включая SOA, другим АА или держателям привилегий оконечных объектов. Данное расширение не должно включаться в сертификаты, содержащие расширение идентификатора SOA.

При наличии, данное расширение может использоваться верификатором привилегий для определения местоположения сертификата спецификации роли.

Если данное расширение не присутствует, либо

a) другие средства будут использоваться для определения местоположения сертификата спецификации роли; либо

b) механизмы, отличные от сертификата спецификации роли, были использованы для присвоения привилегий роли (например, привилегии роли могут быть локально настроены верификатором привилегий).

Данное расширение всегда является некритическим.

15.4.2.1.1 Соответствие ID сертификата спецификации ролей

Правило соответствия идентификаторов сертификатов спецификации ролей сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
roleSpecCertIdMatch MATCHING-RULE  ::= {
    SYNTAX          RoleSpecCertIdentifierSyntax
    ID              id-mr-roleSpecCertIdMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **roleSpecCertIdentifier** и если компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам сохраненного значения.

15.5 Расширения делегирования

15.5.1 Требования

Следующие требования относятся к делегированию привилегий:

a) сертификаты привилегий оконечных объектов должны быть отличимыми от сертификатов АА для защиты от того, что оконечные статьи устанавливают себя как АА без авторизации. Также АА должен иметь возможность ограничивать длину последующего тракта делегирования;

b) АА должен иметь возможность определять соответствующее пространство имен, в пределах которого может иметь место делегирование привилегий. Соблюдение данных ограничений должно иметь возможность быть проверенным верификатором привилегий;

c) АА должен иметь возможность определять приемлемые политики сертификатов, которые должны использовать заявители привилегий далее вниз по тракту делегирования для собственной аутентификации при заявлении о делегировании привилегий данным АА;

d) верификатор привилегий должен иметь возможность определять местонахождение соответствующего сертификата атрибута для выдающего органа, чтобы обеспечить, что выдающий орган имел достаточную привилегию для делегирования привилегии в данном сертификате;

e) независимая служба делегирования (DS) должна выдавать сертификаты, которые делегируют привилегии, пока сервер DS не может сам действовать как претендент на данные привилегии.

15.5.2 Поля расширений делегирования

Определяются следующие поля расширений:

a) *основные ограничения атрибутов*;

b) *ограничения делегированных имен*;

c) *приемлемые политики сертификатов*;

d) *идентификатор атрибута органа*;

e) *непрямой выдающий орган*;

f) *выдано от имени*.

15.5.2.1 Расширение основных ограничений атрибутов

Данное поле указывает, разрешено ли последующее делегирование привилегий, присвоенных в сертификате, содержащем данное расширение. Если так, то также может быть определено ограничение длины тракта делегирования.

Данное поле определяется следующим образом:

```
basicAttConstraints EXTENSION ::=
{
        SYNTAX                  BasicAttConstraintsSyntax
        IDENTIFIED BY           { id-ce-basicAttConstraints }
}

BasicAttConstraintsSyntax ::= SEQUENCE
    {
        authority               BOOLEAN DEFAULT FALSE,
        pathLenConstraint       INTEGER (0..MAX) OPTIONAL
    }
```

Компонент **authority** указывает, является ли держатель авторизованным для последующего делегирования привилегии. Если **authority** принимает значение **TRUE,** держатель также является АА и авторизован далее делегировать привилегию, в зависимости от соответствующих ограничений. Если **authority** принимает значение **FALSE**, держатель является оконечным объектом и не авторизован делегировать привилегию.

Компонент **pathLenConstraint** имеет значение, только если **authority** установлен в **TRUE**. Он задает максимальное количество сертификатов АА, которые могут следовать за данным сертификатом в тракте делегирования. Значение **0** указывает, что субъект данного сертификата может выдавать сертификаты только оконечным объектам, а не АА. Если ни в одном сертификате в тракте делегирования не встречается поле **pathLenConstraint**, то разрешенная длина тракта делегирования не ограничена. Отметим, что ограничение вступает в действие, начиная со следующего сертификата в тракте. Ограничение контролирует количество сертификатов АА между сертификатом АА, содержащим ограничение, и сертификатом оконечного объекта. Ограничение ограничивает длину сегмента тракта делегирования между сертификатом, содержащим данное расширение, и сертификатом оконечного объекта. Оно не оказывает влияния на количество сертификатов в тракте делегирования между опорой доверия и сертификатом, содержащим данное расширение. Поэтому длина полного тракта делегирования может превышать максимальную длины сегмента, ограниченного данным расширением. Ограничение контролирует количество сертификатов АА между сертификатом АА, содержащим ограничение, и сертификатом оконечного объекта. Поэтому общая длина данного сегмента тракта может превышать значение ограничения на два сертификата. (Сюда включаются сертификаты на двух оконечных точках сегмента плюс сертификаты АА между двумя оконечными точками, ограниченными значением данного расширения.)

Данное расширение может присутствовать с сертификатах атрибутов и сертификатах открытых ключей, выданных AA, включая SOA, другим AA или оконечным объектам. Данное расширение не должно быть включено в сертификаты, содержащие расширение идентификатора SOA.

Если данное расширение присутствует в сертификате атрибута и значением **authority** является **TRUE**, держатель авторизован выдавать последующие сертификаты атрибутов, делегирующие содержащуюся привилегию другим объектам, но не сертификаты открытых ключей.

Если данное расширение присутствует в сертификате открытого ключа и если расширение **basicConstraints** указывает, что субъект также является CA, то субъект авторизован выдавать последующие сертификаты открытых ключей, делегирующие данные привилегии другим объектам, но не сертификаты атрибутов. Если включено ограничение длины тракта, то субъект может делегировать только в пределах пересечения ограничения, определенного в данном расширении, и определенного в расширении **basicConstraints**. Если данное расширение присутствует в сертификате открытого ключа, но расширение **basicConstraints** отсутствует или указывает, что субъект является оконечным объектом, то субъект не авторизован делегировать привилегии.

Данное расширение может по усмотрению органа, выдавшего сертификат, быть либо критическим, либо некритическим. Рекомендуется, чтобы оно было помечено как критическое, иначе держатель, не авторизованный быть AA, может выдавать сертификаты, и верификатор привилегий может непреднамеренно использовать такой сертификат.

Если данное расширение присутствует и помечено как критическое, то:

- если значение **authority** не установлено в **TRUE**, то делегированный атрибут не должен использоваться для дальнейшего делегирования;
- если значение **authority** установлено в **TRUE** и присутствует **pathLenConstraint**, то верификатор привилегий должен проверить, что обрабатываемый тракт делегирования совместим со значением **pathLenConstraint**.

Если данное расширение присутствует, помечено как некритическое и не узнается верификатором привилегий, то данная система должна использовать другие способы для определения того, может ли делегируемый атрибут использоваться для дальнейшего делегирования.

Если данное расширение не присутствует или присутствует с пустым значением **SEQUENCE**, то держатель ограничен быть только оконечным объектом, а не органом атрибутов, и держателем не разрешено делегирование привилегий, содержащихся в сертификате атрибута.

15.5.2.1.1 Соответствие основных ограничений атрибутов

Правило соответствия основных ограничений атрибутов сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
basicAttConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          BasicAttConstraintsSyntax
    ID              id-mr-basicAttConstraintsMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **basicAttConstraints** и если компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам сохраненного значения.

15.5.2.2 Расширение ограничений делегированного имени

Поле ограничений делегированного имени указывает пространство имен, в котором должны быть расположены все имена держателя в последующих сертификатах в тракте делегирования.

Данное поле определяется следующим образом:

```
delegatedNameConstraints EXTENSION ::= {
    SYNTAX              NameConstraintsSyntax
    IDENTIFIED BY       id-ce-delegatedNameConstraints }
```

Данное расширение обрабатывается таким же образом, как и расширение **nameConstraints** для сертификатов открытых ключей. Если присутствует **permittedSubtrees**, из всех сертификатов атрибутов, выданных AA держателя и последующими AA в тракте делегирования, приемлемыми являются только сертификаты атрибутов с именами держателей в пределах этих поддеревьев. Если присутствует **excludedSubtrees**, любой сертификат атрибута, выданный AA держателя или последующими AA в тракте делегирования, имеющий имя держателя пределах этих поддеревьев, является неприемлемым. Если присутствуют одновременно **permittedSubtrees** и **excludedSubtrees**, и пространства имен перекрываются, то положение об исключении имеет преимущество.

Данное расширение может присутствовать с сертификатах атрибутов и сертификатах открытых ключей, выданных AA, включая SOA, другим AA. Данное расширение не должно быть включено в сертификаты, выданные оконечным объектам, а также сертификаты, содержащие расширение идентификатора SOA.

Если данное расширение присутствует в сертификате открытого ключа и расширение **nameConstraints** также присутствует, то субъект может делегировать только в пределах пересечения ограничения, определенного в данном расширении, и определенного в расширении **nameConstraints**.

Данное расширение может по усмотрению органа, выдавшего сертификат атрибута, быть либо критическим, либо некритическим. Рекомендуется, чтобы оно было помечено как критическое, иначе пользователь сертификата атрибута может не проверить, что последующие сертификаты атрибутов в тракте делегирования расположены в пространстве имен, намеченном выдающим АА.

15.5.2.2.1 Соответствие ограничений делегированных имен

Правило соответствия ограничений делегированных имен сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
delegatedNameConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          NameConstraintsSyntax
    ID              id-mr-delegatedNameConstraintsMatch}
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **attributeNameConstraints** и если компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам сохраненного значения.

15.5.2.3 Расширение приемлемых политик сертификатов

Поле приемлемых политик сертификатов используется при делегировании с сертификатами атрибутов для контроля приемлемых политик сертификатов, согласно которым должны быть выданы сертификаты открытых ключей для последующих держателей в тракте делегирования. Путем перечисления набора политик в данном поле АА требует, чтобы последующие выдающие органы в тракте делегирования делегировали содержащиеся привилегии только держателям, имеющим сертификаты открытых ключей, выданные согласно одной или нескольким из перечисленных политик сертификатов. Политики, перечисленные здесь, не являются политиками, согласно которым был выдан сертификат атрибутов, а политиками, согласно которым должны быть выданы приемлемые сертификаты открытых ключей для последующих держателей.

Данное поле определяется следующим образом:

```
acceptableCertPolicies EXTENSION ::= {
    SYNTAX              AcceptableCertPoliciesSyntax
    IDENTIFIED BY       id-ce-acceptableCertPolicies }

AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER
```

Данное расширение может присутствовать только в сертификатах атрибутов, выданных АА, включая SOA, другим АА. Данное расширение не должно включаться в сертификаты атрибутов оконечных объектов или в сертификаты открытых ключей. В случае делегирования с использованием сертификатов открытых ключей, та же функциональная возможность предоставляется **certificatePolicies** и другими связанными расширениями.

При наличии, данное расширение должно быть помечено как критическое.

Если данное расширение присутствует и верификатор привилегий понимает его, то верификатор должен обеспечить, что все последующие заявители привилегий в тракте делегирования аутентифицированы с сертификатом открыто ключа согласно одной или нескольким из перечисленных политик сертификатов.

Если данное расширение присутствует, но не понимается верификатором привилегий, то сертификат должен быть отклонен.

15.5.2.3.1 Соответствие приемлемых политик сертификатов

Правило соответствия приемлемых политик сертификатов сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
acceptableCertPoliciesMatch MATCHING-RULE  ::= {
    SYNTAX          AcceptableCertPoliciesSyntax
    ID              id-mr-acceptableCertPoliciesMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **acceptableCertPolicies** и если компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам сохраненного значения.

15.5.2.4 Расширение идентификатора атрибута органа

При делегировании привилегий, АА, делегирующий привилегии, должен сам иметь по меньшей мере одну привилегию и орган, которому делегировать данную привилегию. АА, делегирующий привилегию другому АА или оконечному объекту, может поместить данное расширение в сертификат АА или оконечного объекта, который он выдает. Расширение является обратным указателей на сертификат, в котором органу, выдавшему сертификат, содержащий данное расширение, была присвоена соответствующая ему привилегия. Данное расширение может использоваться верификатором привилегий для обеспечения того, что выдающий АА имел достаточную привилегию для возможности делегирования держателю сертификата, содержащего данное расширение.

Данное поле определяется следующим образом:

```
authorityAttributeIdentifier EXTENSION  ::=
    {
    SYNTAX              AuthorityAttributeIdentifierSyntax
    IDENTIFIED BY       { id-ce-authorityAttributeIdentifier }
    }
AuthorityAttributeIdentifierSyntax   ::=      SEQUENCE SIZE (1..MAX) OF AuthAttId
AuthAttId       ::=     IssuerSerial
```

Сертификат, содержащий данное расширение, может включать делегирование нескольких привилегий держателю сертификата. Если присвоение данных привилегий АА, выдавшему данный сертификат, было выполнено в нескольких сертификатах, то данное расширение должно включать несколько указателей.

Данное расширение может присутствовать в сертификатах атрибутов или сертификатах открытых ключей, выданных АА другим АА или держателям привилегий оконечных объектов. Данное расширение не должно включаться в сертификат, содержащий расширение идентификатора SOA.

Данное расширение всегда является некритическим.

15.5.2.4.1 Соответствие идентификаторов АА

Правило соответствия идентификаторов атрибутов органа сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
authAttIdMatch MATCHING-RULE  ::= {
    SYNTAX          AuthorityAttirbuteIdentifierSyntax
    ID              id-mr-authAttIdMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **authorityAttributeIdentifier** и если компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам сохраненного значения.

15.5.2.5 Расширение непрямого выдающего органа

В некоторых средах привилегия может делегироваться косвенно. В таких случаях делегирующий орган запрашивает, чтобы сервер DS выдал сертификат, делегирующий привилегию, от его имени другому объекту. Поле непрямого выдающего органа используется либо в сертификате атрибута, либо в сертификате открытого ключа, выданного SOA для сервера DS. Наличие данного расширения означает, что АА субъекта (сервер DS) авторизован данным SOA действовать как его заместитель и выдавать сертификаты, которые делегируют привилегии, от имени других делегирующих органов.

```
indirectIssuer EXTENSION ::= {
    SYNTAX          NULL
    IDENTIFIED BY       id-ce-indirectIssuer }
```

Данное расширение всегда является некритическим.

Правило соответствия непрямого выдающего органа сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
indirectIssuerMatch MATCHING-RULE  ::= {
    SYNTAX   NULL
    ID  id-mr-indirectIssuerMatch }
```

Данное правило соответствия возвращает TRUE, если сохраненное значение содержит расширение **indirectIssuer** и если значение, присутствующее в представленном значении, соответствует сохраненному значению.

15.5.2.6 Выдано от имени

Данное расширение вставляется в АС непрямым выдающим органом (сервером DS). Оно указывает АА, который запросил сервер DS выдать АС, и позволяет создавать и проверять подлинность цепочки делегирования.

```
issuedOnBehalfOf EXTENSION ::= {
    SYNTAX GeneralName
    ID id-ce-issuedOnBehalfOf }
```

`GeneralName` является именем АА, который запросил непрямой выдающий орган (сервер DS) выдать данный АС.

Органу, выдавшему данный АС, должна быть предоставлена привилегия выдавать АС от имени других АА посредством SOA, при помощи расширения IndirectIssuer в своем АС.

Данное расширение может быть критическим или некритическим, как необходимо для обеспечения проверки подлинности тракта делегирования.

16 Процедура обработки тракта привилегий

Обработка тракта привилегий выполняется верификатором привилегий. Правила обработки тракта аналогичны правилам для сертификатов открытых ключей.

Другие компоненты обработки тракта, которые не рассматриваются в данном пункте, включают проверку подписей сертификатов, проверку подлинности периодов действия сертификатов и т. д.

Для трактов привилегий, состоящих из одного сертификата (т. е. привилегии были присвоены SOA непосредственно заявителю привилегий) требуется только основная процедура, описанная в п. 16.1, ниже, если привилегия не назначена роли. В данном случае, верификатор привилегий не настраивается согласно определенным привилегиями роли, ему может потребоваться получить сертификат спецификации роли, присваивающий определенные привилегии роли, как описано в п. 16.2, ниже. Если привилегия была делегирована заявителю привилегий промежуточным AA, также требуется процедура тракта делегирования, описанная в п. 16.3. Данные процедуры не выполняются последовательно. Процедура обработки роли и процедура обработки делегирования выполняются перед определением того, являются ли заявленные привилегии достаточными для контекста использования в основной процедуре.

16.1 Основная процедура обработки

Подпись на каждом сертификате в тракте должна быть проверена. Процедуры, относящиеся к проверке подлинности подписей и сертификатов открытого ключа, в данном пункте не повторяются. Верификатор привилегий должен проверить идентификационную информацию каждого объекта в тракте, используя процедуры пункта 10. Отметим, что проверка подписи на сертификате атрибута неизбежно вызывает проверку адресуемого сертификата открытого ключа на подлинность. Если привилегии присваиваются при использовании сертификатов атрибутов, то устройства обработки тракта должны будут учитывать элементы как PMI, так и PKI в ходе окончательного определения подлинности сертификата атрибута заявителя привилегий. По подтверждении подлинности, привилегии, содержащиеся в сертификате, *могут* использоваться в зависимости от сравнения с соответствующей политикой привилегий и другой информацией, соответствующей контексту, в котором используется сертификат.

Контекст использования должен определить, намерен ли на самом деле держатель привилегий заявить содержащуюся привилегию к использованию с данным контекстом. Факт существования цепочки сертификатов к доверенному SOA сам по себе не является достаточным для выполнения данного определения. Желание держателя привилегий использовать данный сертификат должно быть четко указано и проверено. Тем не менее, механизмы обеспечения того, что такое заявление привилегии было должным образом показано держателем привилегий, находятся вне области применения данной Спецификации. В качестве примера, такое заявление привилегий может быть проверяемо, если держатель привилегий подписал ссылку на данный сертификат, таким образом указывая желание использовать данный сертификат для данного контекста.

Для каждого сертификата атрибута в тракте, который не содержит расширение **noRevAvail**, верификатор привилегий должен обеспечить, что сертификат атрибута не был аннулирован.

Верификатор привилегий должен обеспечивать, что заявленная привилегия является действительной на время, называемое "время оценки", которая может быть выполнена в *любое* время, т. е. текущее время проверки или любое время в прошлом. В контексте услуги управления доступом, проверка всегда выполняется на настоящее время. Тем не менее, в контексте фиксации авторства, проверка может быть выполнена на время в прошлом или текущее время. Когда сертификаты прошли проверку подлинности, верификатор привилегий должен обеспечить, что время оценки попадает во все периоды действия всех сертификатов, использованных в тракте. Также, если какой-либо сертификат в тракте содержит расширение **timeSpecification**, ограничения, наложенные на время, когда привилегия может быть заявлена, также должны позволять заявлению привилегии быть действительными на момент оценки.

Если в сертификате, использованном для заявления привилегии, присутствует расширение **targetingInformation**, верификатор привилегии должен проверить, что сервер/услуга, для которой производится проверка, включена в список целей.

Если сертификат является сертификатом присвоений ролей, то процедура обработки, описанная в п. 16.2, должна обеспечивать, что соответствующие привилегии определены. Если привилегия была делегирована объекту, а не присвоена непосредственно SOA, доверенным верификатором привилегий, то процедура обработки, описанная в п. 16.3, должна обеспечивать, что делегирование было выполнено должным образом.

Верификатор привилегий также должен определить, являются ли заявляемые привилегии достаточными для контекста использования. Политика привилегий устанавливает правила для совершения данного определения и включает определение любых переменных среды, которые необходимо принимать во внимание. Присвоенные привилегии, включая полученные в процедуре ролей в п. 16.2 и в процедуре делегирования в п. 16.3, а также любые соответствующие переменные среды (например, время дня или текущий платежный баланс) сравниваются с политикой привилегий для определения того, являются ли они достаточными для контекста использования. Если присутствует расширение **acceptablePrivilegePolicies**, заявление привилегий может быть успешным, только если политика привилегий, сравнение с которой производит верификатор привилегий, содержится в данном расширении.

Если сравнение проходит успешно, верификатору привилегий предоставляются какие-либо соответствующие уведомления пользователя.

16.2 Процедура обработки ролей

Если заявленный сертификат является сертификатом присвоения ролей, верификатор привилегий должен получить определенные привилегии, присвоенные данной роли. Имя роли, которой присвоен заявитель привилегии, содержится в атрибуте **role** сертификата. Верификатору привилегий, если он еще не настроен на привилегии названной роли, может потребоваться необходимость определить местоположение сертификата спецификации роли, присваивающего привилегии данной роли. Информация в атрибуте **role** и в расширении **roleSpecCertIdentifier** может использоваться для определения местоположения данного сертификата.

Привилегии, присвоенные роли, неявно присваиваются заявителю привилегий и поэтому включаются в заявленные привилегии, сравниваемые с политикой привилегий в основной процедуре в п. 16.1 для определения того, являются ли заявленные привилегии достаточными для контекста использования.

16.3 Процедура обработки делегирования

Если привилегии заявляются и делегируются заявителю привилегий промежуточным AA, верификатор привилегий должен обеспечивать, что тракт является действительным трактом делегирования, обеспечивая, что:

- каждый AA, выдавший сертификат в тракте делегирования, был авторизован для этого;
- каждый сертификат в тракте делегирования является действительным в отношении тракта и ограничений имени, налагаемы на него;
- каждый объект в тракте делегирования аутентифицирован с сертификатом открытого ключа, который является действительным в соответствии с любыми ограничениями, налагаемыми политикой;
- не существует привилегии делегирования AA больше, чем привилегия, удерживаемая данным AA.

До начала проверки подлинности тракта делегирования, верификатор привилегий должен получить следующее. Любое из этого может быть предоставлено заявителем привилегий или получено верификатором привилегий из какого-либо другого источника, такого как Справочник. Атрибуты услуги могут быть предоставлены верификатору привилегий в виде структурированного документа или каким-либо другим способом.

- Установленное доверие в открытом ключе верификации, использованном для проверки подлинности подписи SOA. Данное доверие может быть установлено или при помощи внеполосных средств, или путем сертификата открытого ключа, выданного SOA CA, в котором верификатор привилегий уже установил доверие. Такой сертификат должен содержать расширение **sOAIdentifier**.
- Привилегия верификатора привилегий, закодированная в его сертификате атрибута или в расширении атрибутов справочника субъекта его сертификата открытого ключа.
- Тракт делегирования сертификатов от заявителя привилегий к доверенному SOA.
- Правило доминирования для заявляемой привилегии; может быть получено из дескриптора атрибута, выданного SOA, ответственного за рассматриваемый атрибут, или может быть получено посредством внеполосных средств.
- Политика привилегий; может быть получена из Справочника или посредством каких-либо внеполосных средств.
- Переменные среды, включая например текущую дату/время, текущий платежный баланс и т. д.

Реализация должна быть функционально эквивалентной внешнему поведению, вытекающему из данной процедуры; тем не менее, алгоритм, используемый определенной реализацией для получения правильного(ых) результата(ов) из заданных входных данных, не стандартизируется.

В случае, когда сертификаты атрибутов выдаются непрямым выдающим органом (DS), зависимая сторона должна полностью проверять подлинность цепочки делегирования следующим образом:

i) начиная с AC оконечного объекта, RP извлекает имя пользователя и имя issuedOnBehalfOf;

ii) RP ищет AC выдающего органа и проверяет, что выдающий орган является непрямым выдающим органом SOA (т. е. имеет расширение indirectIssuer);

iii) RP ищет AC AA issuedOnBehalfOf и проверяет, что AA содержит расширенное множество атрибутов привилегий, выданных данному оконечному объекту.

RP возвращается к шагу ii), используя AC AA, и таким образом поднимается по цепочке до тех пор, пока не дойдет до AC AA, выданного SOA.

16.3.1 Проверить целостность правила доминирования

Правило доминирования связано с делегируемой привилегией. Синтаксис и метод получения правила делегирования не стандартизируется. Тем не менее, целостность восстановленного правила доминирования может быть проверена. Сертификат дескриптора атрибута, выданный SOA, ответственным за делегируемый атрибут, может содержать HASH правила доминирования. Верификатор привилегий может воспроизвести функцию HASH восстановленной копии правила доминирования и сравнить оба хэша. Если они идентичны, верификатор привилегий имеет верное правило доминирования.

16.3.2 Установить действительный тракт делегирования

Верификатор привилегий должен найти тракт делегирования и получить сертификаты для каждого объекта в тракте. Тракт делегирования распространяется от непосредственного заявителя привилегий к SOA. Каждый промежуточный сертификат в тракте делегирования должен содержать расширение **basicAttConstraints** с компонентом органа, установленным в **TRUE**. Орган, выдающий каждый сертификат, должен быть таким же, как и держатель/субъект сертификата, смежного с ним в тракте делегирования. Расширение **authorityAttributeIdentifier** используется для определения местоположения соответствующего сертификата смежной записи в тракте делегирования. Количество сертификатов в тракте от каждого объекта до непосредственного заявителя привилегий (включительно) не должно превышать значения **pathLenConstraint** в расширении **basicAttConstraints** объекта более чем на 2. Это связано с тем, что **pathLenConstraint** ограничивает количество промежуточных сертификатов между двумя оконечными точками (т. е. сертификатом, содержащим ограничение, и сертификатом оконечного объекта), поэтому максимальная длина равна значению данного ограничения плюс сертификаты, являющиеся оконечными точками.

Если расширение **delegatedNameConstraints** присутствует в каком-либо сертификате в тракте делегирования, ограничения обрабатываются тем же способом, как обрабатывается расширение **nameConstraints** в процедуре обработки тракта сертификации в пункте 10.

Если расширение **acceptableCertPolicies** присутствует в каком-либо сертификате в тракте делегирования, верификатор привилегий должен обеспечить, что аутентификация каждого последующего объекта в тракте делегирования выполнена с сертификатом открытого ключа, содержащим по меньшей мере одну из приемлемых политик.

16.3.3 Проверить делегирование привилегий

Ни один делегирующий орган не может делегировать привилегию, большую, чем привилегия, которой он обладает. Правило доминирования в атрибуте дескриптора атрибута предоставляет правила, в соответствии с которыми заданное значение "меньше чем" другое значение для делегируемого атрибута.

Для каждого сертификата в тракте делегирования, включая сертификат непосредственного заявителя привилегии, верификатор привилегий должен обеспечивать, что делегирующий орган был авторизован делегировать привилегию, которой он обладает, и что делегированная привилегия не была больше, чем привилегия, которой он обладает.

Для каждого из данных сертификатов, верификатор привилегий должен сравнить делегированную привилегию с привилегией, которой обладает делегирующий орган, в соответствии с правилом доминирования для привилегии. Привилегия, которой обладает делегирующий орган, получается из смежного сертификата в тракте делегирования, как описано в п. 16.2. Сравнение двух привилегий выполняется на основе правила доминирования, описанного в п. 16.3.1.

16.3.4 Определение прохождения/неудачи

Допуская, что установлен действительный тракт делегирования, привилегии непосредственного заявителя привилегий предоставляются как входное значение для сравнения с политикой привилегий, как описано в п. 16.1, для определения того, имеет ли непосредственный заявитель привилегии достаточную привилегию для контекста использования.

17 Схема справочника PMI

В данном пункте определяются элементы схемы справочника, используемые для представления в Справочнике информации PMI. В него включена спецификация соответствующих классов объектов, атрибуты и правила соответствия значений атрибутов.

17.1 Классы объектов справочника PMI

В данном подпункте определяются определения классов объектов, используемых для представления объектов PMI в Справочнике.

17.1.1 Класс объектов пользователь PMI

Класс объектов пользователя PMI используется при определении записей для объектов, которые могут являться держателями сертификатов атрибутов.

```
pmiUser OBJECT-CLASS ::= {
-- пользователь PMI (т. е. "держатель")
  SUBCLASS OF     {top}
  KIND            auxiliary
  MAY CONTAIN     {attributeCertificateAttribute}
  ID              id-oc-pmiUser }
```

17.1.2 Класс объектов AA PMI

Класс объектов AA PMI используется при определении записей для объектов, которые действуют как органы атрибутов.

```
pmiAA OBJECT-CLASS ::= {
-- AA PMI
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    {aACertificate |
                  attributeCertificateRevocationList |
                  attributeAuthorityRevocationList}
  ID             id-oc-pmiAA }
```

17.1.3 Класс объектов SOA PMI

Класс объектов SOA PMI используется при определении записей для объектов, которые действуют как источники органов. Отметим, что если объект был авторизован действовать как SOA путем выдачи сертификата открытого ключа, содержащего расширение **sOAIdentifier**, то запись справочника, представляющая данный объект, также должна содержать класс объекта **pkiCA**.

```
pmiSOA OBJECT-CLASS ::= {  -- источник органа PMI
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    {attributeCertificateRevocationList |
                  attributeAuthorityRevocationList |
                  attributeDescriptorCertificate}
  ID             id-oc-pmiSOA }
```

17.1.4 Класс объектов точка распределения CRL сертификата атрибута

Класс объектов точка распределения CRL сертификата атрибута используется при определении записей для объектов, которые содержат сегменты списков аннулирования сертификатов атрибутов и/или органов атрибутов. Данный вспомогательный класс предназначен для сочетания со структурный классом объектов **crlDistributionPoint** при обработке записей. Так как атрибуты **certificateRevocationList** и **authorityRevocationList** являются необязательными в данном классе, возможно создавать записи, содержащие, например, только один список аннулирования органов атрибутов, или записи, содержащие списки аннулирования нескольких типов, в зависимости от требований.

```
attCertCRLDistributionPt     OBJECT-CLASS ::= {
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    { attributeCertificateRevocationList |
                   attributeAuthorityRevocationList }
  ID             id-oc-attCertCRLDistributionPts }
```

17.1.5 Тракт делегирования PMI

Класс объектов тракта делегирования PMI используется при определении записей для объектов, которые могут содержать тракты делегирования. В общем случае он будет использоваться в сочетании с записями структурного класса объектов **pmiAA**.

```
pmiDelegationPath      OBJECT-CLASS   ::= {
   SUBCLASS OF         {top}
   KIND                auxiliary
   MAY CONTAIN         { delegationPath }
   ID                  id-oc-pmiDelegationPath }
```

17.1.6 Класс объектов политика привилегий

Класс объектов политика привилегий используется при определении записей для объектов, которые содержат информацию о политике привилегий.

```
privilegePolicy      OBJECT-CLASS   ::= {
   SUBCLASS OF       {top}
   KIND              auxiliary
   MAY CONTAIN       {privPolicy }
   ID                id-oc-privilegePolicy }
```

17.1.7 Класс объектов защищенная политика привилегий

Класс объектов защищенная политика привилегий используется при определении записей для объектов, которые содержат политики привилегий, защищенные в сертификатах атрибутов.

```
protectedPrivilegePolicy          OBJECT-CLASS   ::= {
   SUBCLASS OF       {top}
   KIND              auxiliary
```

```
    MAY CONTAIN     {protPrivPolicy }
    ID              id-oc-protectedPrivilegePolicy }
```

17.2 Атрибуты Справочника PMI

В данном подпункте определяются атрибуты справочника, используемые для хранения данных PMI в записях справочника.

17.2.1 Атрибут сертификата атрибута

Следующий атрибут содержит сертификаты атрибутов, выданные определенному держателю, и хранится в записи справочника держателя.

```
attributeCertificateAttribute ATTRIBUTE  ::= {
    WITH SYNTAX                 AttributeCertificate
    EQUALITY MATCHING RULE      attributeCertificateExactMatch
    ID                          id-at-attributeCertificate }
```

17.2.2 Атрибут сертификата AA

Следующий атрибут содержит сертификаты атрибутов, выданные AA, и хранится в записи справочника AA держателя.

```
aACertificate       ATTRIBUTE   ::=  {
    WITH SYNTAX                 AttributeCertificate
    EQUALITY MATCHING RULE      attributeCertificateExactMatch
    ID                          id-at-aACertificate }
```

17.2.3 Атрибут сертификата дескриптора атрибута

Следующий атрибут содержит сертификаты атрибутов, выданные SOA, содержащие расширение **attributeDescriptor**. Данные сертификаты атрибутов содержат действующий синтаксис и спецификацию правила доминирования атрибутов привилегий и хранятся в записи справочника выдающего SOA.

```
attributeDescriptorCertificate      ATTRIBUTE  ::= {
    WITH SYNTAX                 AttributeCertificate
    EQUALITY MATCHING RULE      attributeCertificateExactMatch
    ID                          id-at-attributeDescriptorCertificate }
```

17.2.4 Атрибут списка аннулированных сертификатов атрибутов

Следующий атрибут содержит список аннулированных сертификатов атрибутов. Данные списки могут храниться в записи справочника выдающего органа или другой записи справочника (например, точке распределения).

```
attributeCertificateRevocationList      ATTRIBUTE ::= {
    WITH SYNTAX                 CertificateList
    EQUALITY MATCHING RULE      certificateListExactMatch
    ID                          id-at-attributeCertificateRevocationList}
```

17.2.5 Атрибут списка аннулированных сертификатов AA

Следующий атрибут содержит список аннулированных сертификатов атрибутов, выданных AA. Данные списки могут храниться в записи справочника выдающего органа или другой записи справочника (например, точке распределения).

```
attributeAuthorityRevocationList        ATTRIBUTE   ::= {
    WITH SYNTAX                 CertificateList
    EQUALITY MATCHING RULE      certificateListExactMatch
    ID                          id-at-attributeAuthorityRevocationList }
```

17.2.6 Атрибут тракта делегирования

Атрибут тракта делегирования содержит тракты делегирования, каждый из которых состоит из последовательности сертификатов атрибутов.

```
delegationPath      ATTRIBUTE   ::= {
    WITH SYNTAX             AttCertPath
    ID                      id-at-delegationPath }

AttCertPath     ::=  SEQUENCE OF AttributeCertificate
```

Данный атрибут может храниться в записи справочника AA и может содержать некоторые тракты делегирования от данного AA к другим AA. Данный атрибут, если используется, дает возможность более эффективного поиска сертификатов атрибутов, формирующих часто используемые тракты делегирования. К данному атрибуту не предъявляются определенные требования как таковые, которые должны использоваться, и множество значений, хранящихся в атрибуте, вряд ли представляет полный набор трактов делегирования для любого заданного AA.

17.2.7 Атрибут политики привилегий

Атрибут политики привилегий содержит информацию о политиках привилегий.

```
privPolicy ATTRIBUTE  ::= {
    WITH SYNTAX       PolicySyntax
    ID                id-at-privPolicy }
```

Компонент **policyIdentifier** включает идентификатор объекта, зарегистрированный за определенной политикой привилегий.

Если присутствует **content**, то включено полное содержимое политики привилегий.

Если присутствует **pointer** то компонент **name** обращается к одному или нескольким местоположениям, где может находиться копия политики привилегий. Если присутствует компонент **hash**, он содержит HASH содержимого политики привилегий, которое должно находиться по адресуемому местоположению. Данный хэш может использоваться при выполнении проверки целостности адресуемого документа.

17.2.8 Атрибут защищенной политики привилегий

Атрибут защищенной политики привилегий содержит политики привилегий, защищенные в сертификатах атрибутов.

```
protPrivPolicy      ATTRIBUTE   ::= {
    WITH SYNTAX                 AttributeCertificate
    EQUALITY MATCHING RULE      attributeCertificateExactMatch
    ID                          id-at-protPrivPolicy }
```

Отметим, что в отличие от обычных сертификатов атрибутов, сертификаты с атрибутом **protPrivPolicy** содержат политики привилегий, а не привилегии. Компоненты выдавший орган и держатель данных сертификатов атрибутов определяют один и тот же объект. Атрибут, включенный в сертификат атрибута, содержащийся в атрибуте **protPrivPolicy**, является либо атрибутом **privPolicy**, либо атрибутом **xmlPrivPolicy**.

17.2.9 Атрибут защищенной политики привилегий XML

Атрибут защищенной политики привилегий XML содержит закодированную с XML информацию о политике привилегий.

```
xmlPrivPolicy     ATTRIBUTE   ::= {
    WITH SYNTAX   UTF8String -- содержит закодированную с XML информацию о политике привилегий
    ID            id-at-xMLPprotPrivPolicy }
```

17.3 Общие правила соответствия справочника PMI

В данном подпункте определяются правила соответствия для атрибутов справочника PMI.

17.3.1 Точное соответствие сертификатов атрибутов

Правило точного соответствия сертификатов атрибутов сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **AttributeCertificate**.

```
attributeCertificateExactMatch MATCHING-RULE  ::= {
    SYNTAX          AttributeCertificateExactAssertion
    ID              id-mr-attributeCertificateExactMatch }

AttributeCertificateExactAssertion      ::=  SEQUENCE {
    serialNumber            CertificateSerialNumber,
    issuer                  AttCertIssuer }
```

Данное правило соответствия возвращает TRUE, если компоненты в значении атрибута соответствуют значениям в представленном значении.

17.3.2 Соответствие сертификатов атрибутов

Правило соответствия сертификатов атрибутов сравнивает представленное значение и значение атрибута типа **AttributeCertificate**. Данное правило соответствия делает возможным более комплексное соответствие, чем **certificateExactMatch**.

```
attributeCertificateMatch  MATCHING-RULE  ::=  {
    SYNTAX          AttributeCertificateAssertion
    ID              id-mr-attributeCertificateMatch  }

AttributeCertificateAssertion  ::=  SEQUENCE  {
    holder          [0]  CHOICE {
                         baseCertificateID      [0]  IssuerSerial,
                         holdertName            [1]  GeneralNames} OPTIONAL,
    issuer          [1]  GeneralNames OPTIONAL,
    attCertValidity [2]  GeneralizedTime OPTIONAL,
    attType         [3]  SET OF AttributeType OPTIONAL}
```

-- Должен присутствовать по меньшей мере один из компонентов последовательности

Данное правило соответствия возвращает TRUE, если все компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам значения атрибута следующим образом:

- **baseCertificateID** соответствует, если эквивалентно компоненту **IssuerSerial** сохраненного значения атрибута;
- **holderName** соответствует, если сохраненное значение атрибута содержит расширение имени с тем же типом имени, как указано в представленном значении;
- **issuer** соответствует, если сохраненное значение атрибута содержит компонент имени того же типа имени, как указано в представленном значении;
- **attCertValidity** соответствует, если попадает в определенный период действия сохраненного значения атрибута; и
- для каждого **attType** в представленном значении, в компоненте **attributes** сохраненного значения присутствует атрибут данного типа.

17.3.3 Соответствие выдавших органов держателя

Правило соответствия выдавших органов держателей сертификатов атрибутов сравнивает на предмет эквивалентности представленное значение компонентов представленного значения держателя и/или выдавшего органа и значение атрибута типа **AttributeCertificate**.

```
holderIssuerMatch MATCHING-RULE  ::= {
    SYNTAX          HolderIssuerAssertion
    ID              id-mr-holderIssuerMatch }

HolderIssuerAssertion   ::=  SEQUENCE {
    holder          [0]      Holder          OPTIONAL,
    issuer          [1]      AttCertIssuer   OPTIONAL }
```

Данное правило соответствия возвращает TRUE, если все компоненты, присутствующие в представленном значении, соответствуют соответствующим компонентам значения атрибута.

17.3.4 Соответствие трактов делегирования

Правило соответствия **delegationPathMatch** сравнивает на предмет эквивалентности представленное значение и значение атрибута типа **delegationPath**. Верификатор привилегий может использовать данное правило соответствия для выбора тракта, начинающегося с сертификата, выданного его SOA, и заканчивающегося сертификатом, выданным AA, который выдал проверяемый сертификат держателя оконечного объекта.

```
delegationPathMatch MATCHING-RULE  ::= {
    SYNTAX          DelMatchSyntax
    ID              id-mr-delegationPathMatch }

DelMatchSyntax      ::=  SEQUENCE {
    firstIssuer     AttCertIssuer,
    lastHolder      Holder }
```

Данное правило соответствия возвращает TRUE, если представленное значение в компоненте **firstIssuer** соответствует соответствующим элементам поля выдавший орган первого сертификата в **SEQUENCE** в сохраненном значении, и представленное значение в компоненте **lastHolder** соответствует соответствующим элементам поля держатель последнего сертификата в **SEQUENCE** в сохраненном значении. Данное правило соответствия возвращает FALSE, если любое из соответствий терпит неудачу.

РАЗДЕЛ 4 – ИСПОЛЬЗОВАНИЕ СПРАВОЧНИКА СТРУКТУР СЕРТИФИКАТОВ ОТКРЫТЫХ КЛЮЧЕЙ И АТРИБУТОВ

Справочник использует структуру сертификатов открытых ключей в качестве основы для нескольких услуг безопасности, включая строгую аутентификацию и защиту операций Справочника, а также защиту хранящихся данных. Справочник использует структуру сертификатов атрибутов в качестве основы для основанной на правилах схемы управления доступом. Здесь определяется отношение элементов структуры сертификатов открытых ключей и структуры сертификатов атрибутов к различным услугам безопасности Справочника. Особые услуги безопасности, предоставляемые Справочником, полностью определяются в полном наборе спецификаций Справочника.

18 Аутентификация Справочника

Справочник поддерживает аутентификацию пользователей, получающих доступ к Справочнику через DUA, а также аутентификацию систем справочников (DSA) пользователям и другим DSA. В зависимости от среды, может использоваться либо простая, либо строгая аутентификация. Процедуры, которые должны использоваться для простой и строгой аутентификации в Справочнике, описаны в последующих подпунктах.

18.1 Простая процедура аутентификации

Простая аутентификация предназначена для обеспечения локальной авторизации, основанной на выделенном имени пользователя, двусторонне согласованном (необязательно) пароле и двустороннего понимания способов использования и управления данным паролем в пределах одной области. Использование простой аутентификации в основном предназначено для локального использования, т. е. для аутентификации равноправных элементов между одним DUA и одним DSA или между одним DSA и одним DSA. Простая аутентификация может выполняться несколькими способами:

a) передачей выделенного имени пользователя и (необязательно) пароля в открытой (незащищенной) форме получателю для оценки;

b) передачей выделенного имени пользователя, пароля и случайного числа и/или временной отметки, все из которых защищены применением однонаправленной функции;

c) передачей защищенной информации, описанной в b), вместе со случайным числом и/или временной отметкой, все из которых защищены применением однонаправленной функции.

ПРИМЕЧАНИЕ 1. – Не требуется, чтобы применяемые однонаправленные функции были различными.

ПРИМЕЧАНИЕ 2. – Процедуры сигнализации для защиты паролей могут быть вопросом для расширения документа.

Если пароли не защищены, то предоставляется минимальная степень защиты для предотвращения несанкционированного доступа. Это не должно считаться основой для услуг безопасности. Защита выделенного имени и пароля пользователя предоставляет большие степени безопасности. Алгоритмы, используемые для механизма защиты, как правило представляют собой незашифрованные однонаправленные функции, очень простые для реализации.

Общая процедура для выполнения простой аутентификации представлена на рисунке 5.

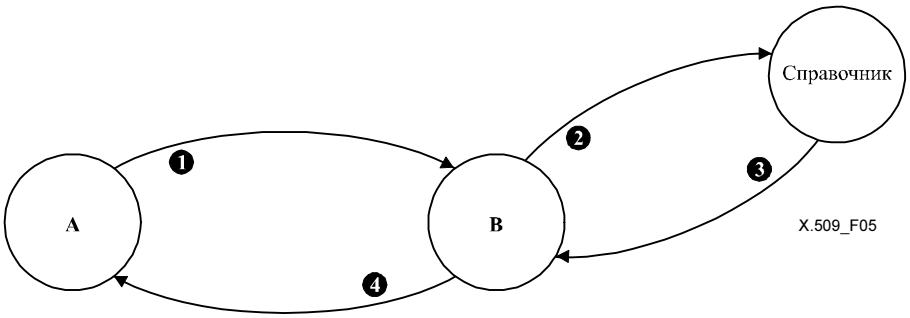


Рисунок 5 – Процедура незащищенной простой аутентификации

Она включает следующие шаги:

1) исходящий пользователь A посылает свое выделенное имя и пароль принимающему пользователю B;

2) B посылает заявленное выделенное имя и пароль A в Справочник, где пароль проверяется с содержащимся в атрибуте **UserPassword** в записи справочника для A (используя операцию сравнения Справочника);

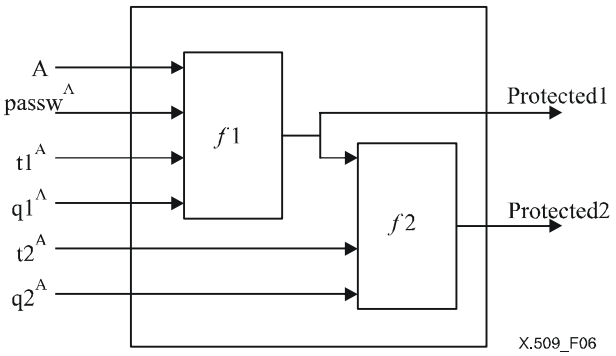
3) Справочник подтверждает (или отрицает) B, что удостоверения являются действительными;

4) успех (или неудача) аутентификации может передаваться A.

Самая основная форма простой аутентификации включает только шаг 1), и после проверки пользователем B выделенного имени и пароля, может включать шаг 4).

18.1.1 Генерация защищенной идентификационной информации

На рисунке 6 показано два подхода, при помощи которых может генерироваться защищенная идентификационная информация. *f*1 и *f*2 представляют собой однонаправленные функции (идентичные или различные), а временные отметки и случайные числа являются необязательными и применяются по двустороннему соглашению.



A Выделенное имя пользователя
T^A Временные отметки
$passw^A$ Пароль A
q^A Случайные числа, по выбору включенные с счетчиком

Рисунок 6 – Защищенная простая аутентификация

18.1.2 Процедура защищенной простой аутентификации

На рисунке 7 показана процедура защищенной простой аутентификации.

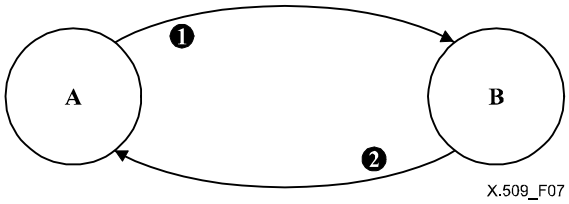


Рисунок 7 – Процедура защищенной простой аутентификации

Данная процедура включает следующие шаги (изначально используя только *f*1):

1) Исходящий пользователь, пользователь A, посылает свою защищенную идентификационную информацию (Authenticator1) пользователю B. Защита достигается путем применения однонаправленной функции (*f*1) на рисунке 6, где временная отметка и/или случайное число (если используется) используется для минимизации воспроизведения и для маскировки пароля.

   Защита пароля A имеет форму:

   $Protected1 = f1\ (t1^A, q1^A, A, passw^A)$.

   Информация, передаваемая B, имеет форму:

   $Authenticator1 = t1^A, q1^A, A, Protected1$.

2) Пользователь B проверяет идентификационную информацию, предложенную пользователем A, путем генерации (с использованием выделенного имени и дополнительно временной отметки и/или случайного числа, предоставленного пользователем A, вместе с локальной копией пароля пользователя A) локальной защищенной копии пароля A (формы Protected1). B сравнивает на предмет эквивалентности заявленную идентификационную информацию и локально сгенерированное значение.

3) Пользователь B подтверждает или отрицает пользователю A проверку защищенной идентификационной информации.

Данная процедура может быть модифицирована для предоставления большей защиты, используя *f*1 и *f*2. Основные отличия заключаются в следующем:

1) Пользователь A посылает свою дополнительно защищенную идентификационную информацию (Authenticator2) пользователю B. Дополнительная защита достигается путем применения в дальнейшем однонаправленной функции *f*2, как показано на рисунке 6. Дополнительная защита имеет форму:

   $Protected2 = f2\ (t2^A, q2^A, Protected1)$.

   Информация, передаваемая B, имеет форму:

   $Authenticator2 = t1^A, t2^A, q1^A, q2^A, A, Protected2$.

Для сравнения пользователь B генерирует локальное значение дополнительно защищенного пароля пользователя A и сравнивает его на предмет эквивалентности со значением Protected2.

2) Пользователь B подтверждает или отрицает пользователю A проверку защищенной идентификационной информации.

ПРИМЕЧАНИЕ. – Процедуры, определенные в данных пунктах, определены в терминах пользователей A и B. Применительно к Справочнику (определенному в Рек. МСЭ-Т X.511 | ИСО/МЭК 9594-3 и Рек. МСЭ-Т X.518 | ИСО/МЭК 9594-4), в качестве A может выступать DUA, связанный с DSA, выступающим в качестве B; альтернативно, в качестве A может выступать DSA, связанный с другим DSA, выступающим в качестве B.

18.1.3 Тип атрибута пароль пользователя

Тип атрибута пароль пользователя содержит пароль объекта. Значением атрибута для пароля пользователя является строка, определенная объектом.

```
userPassword ATTRIBUTE  ::=  {
    WITH SYNTAX                 OCTET STRING (SIZE (0..ub-user-password))
    EQUALITY MATCHING RULE      octetStringMatch
    ID                          id-at-userPassword }
```

18.2 Строгая аутентификация

Процедуры, описанные в данном подпункте, предназначены для использования при аутентификации между DUA и DSA, а также между парами DSA. Процедуры используют структуру сертификатов открытых ключей, определенную в данной Спецификации. Более того, процедуры используют сам Справочник как хранилище информации об открытых ключах, требуемой для выполнения аутентификации. Включение соответствующих параметров в протоколы Справочника определено непосредственно в спецификациях протоколов. Процедуры, определенные здесь для строгой аутентификации, также могут использоваться приложениями, отличными от Справочника, которые также используют подобное хранилище. При использовании Справочником данных процедур, термин 'пользователь' в данных процедурах может относиться либо к DUA, либо к DSA.

Подход к строгой аутентификации, принятый в данной спецификации Справочника, использует свойства семейства криптографических систем, известных как криптосистемы открытых ключей (PKCS). Данные криптосистемы, также описываемые как асимметричные, включают пару ключей, один частный и один открытый, в отличие от одного ключа в традиционных криптографических системах. В Приложении E представлено кратное введение в эти криптосистемы и их свойства, делающие их полезными при аутентификации. Чтобы PKCS могла использоваться в данной структуре аутентификации в настоящее время, она должна обладать свойством, что оба ключа в паре ключей могут использоваться для шифрования, причем частный ключ должен использоваться для дешифрования, если для шифрования использовался открытый ключ, и открытый ключ должен использоваться для дешифрования, если для шифрования использовался частный ключ. Другими словами, $X_p \bullet X_s = X_s \bullet X_p$, где X_p/X_s являются функциями шифрования/дешифрования с использованием открытого/частного ключа пользователя X.

ПРИМЕЧАНИЕ. – Альтернативные типы PKCS, т. е. не требующие свойства перестановочности, которые могут поддерживаться без значительных модификаций данной спецификации Справочника, в будущем могут быть расширены.

Данная структура аутентификации не требует использования определенной криптосистемы. Предполагается, что данная структура должна быть применима к любой подходящей криптосистеме открытых ключей, и должна таким образом поддерживать изменения в используемых методах, которые появятся в результате будущих достижений в криптографии, математических методах или вычислительных возможностях. Тем не менее, два пользователя, которые желают пройти аутентификацию, для правильного выполнения аутентификации должны поддерживать один и тот же криптографический алгоритм. Таким образом, в контексте совокупности связанных приложений, выбор единого алгоритма должен способствовать максимизации сообщества пользователей, которым доступна конфиденциальная аутентификация и связь.

Аутентификация предполагает, что каждый пользователь обладает уникальным выделенным именем. Распределение выделенных имен является обязанностью органов по присвоению имен. Каждый пользователь должен таким образом доверять, что органы по присвоению имен не выдают повторяющиеся выделенные имена.

Каждый пользователь определяется своим обладанием частным ключом. Другой пользователь имеет возможность определить, обладает ли его партнер по соединению частным ключом, и может использовать это для подтверждения того, что партнер по соединению действительно является пользователем. Подлинность данного подтверждения зависит от того, остается ли частный ключ конфиденциальным для пользователя.

Чтобы пользователь мог определить, что партнер по соединению обладает частным ключом другого пользователя, он должен сам обладать открытым ключом того пользователя. Если получение значения данного открытого ключа из записи пользователя в Справочнике является простым, то проверка его правильности является более сомнительной. Существует много возможных способов осуществления этого: в подпункте 18.2.1 описан процесс, в котором открытый клюв пользователя может быть проверен путем обращения к Справочнику. Данный процесс может действовать, только если в Справочнике существует непрерывная цепочка доверенных точек между пользователями, запрашивающими аутентификацию. Такая цепочка может быть создана путем определения общей точки доверия. Данная общая точка доверия должна быть привязана к каждому пользователю при помощи непрерывной цепочки доверенных точек.

18.2.1 Получение сертификатов открытых ключей из справочника

Сертификаты хранятся в записях справочника как атрибуты типов **UserCertificate**, **CACertificate** и **CrossCertificatePair**. Данные типы атрибутов являются известными для Справочника. Данными атрибутами можно оперировать, используя те же операции протокола, как и для других атрибутов. Определение данных типов содержится в п. 3.3, а спецификация данных типов атрибутов определена в п. 11.2.

В общем случае, прежде чем пользователи смогут обоюдно аутентифицировать друг друга, Справочник должен предоставить полные тракты сертификации и обратной сертификации. Тем не менее, на практике объем информации, который должен быть получен из Справочника, может быть сокращен для каждого определенного случая аутентификации за счет того, что:

a) если два пользователя, желающие аутентифицировать друг друга, обслуживаются одни и тем же органом по сертификации, то тракт сертификации становится тривиальным, и пользователи могут непосредственно раскрывать сертификаты друг друга;

b) если CA пользователей расположены в иерархическом порядке, пользователь может хранить открытые ключи, сертификаты и обратные сертификаты всех органов по сертификации между пользователем и корнем DIT. Как правило, это предполагает знание пользователем открытых ключей и сертификатов только трех или четырех органов по сертификации. Пользователь затем должен только получить тракты сертификации от общей точки доверия;

c) если пользователь часто соединяется с пользователями, сертифицированными определенным другим CA, то данный пользователь может изучить тракт сертификации к данному CA и обратный тракт сертификации от данного CA, после чего ему необходимо только получить сам сертификат другого пользователя из Справочника;

d) органы по сертификации могут перекрестно сертифицировать один другого на основе двустороннего соглашения. В результате этого сокращается тракт сертификации;

e) если два пользователя уже взаимодействовали ранее и изучили сертификаты друг друга, они могут аутентифицировать друг друга без какого-либо обращения к Справочнику.

В любом случае, изучив сертификаты друг друга из тракта сертификации, пользователи должны проверять подлинность полученных сертификатов.

18.2.1.1 Пример

На рисунке 8 показан гипотетический пример фрагмента DIT, где CA формируют иерархию. Помимо информации, показанной в CA, мы допускаем, что каждый пользователь знает открытый ключ своего органа по сертификации и свой собственный открытый и частный ключи.

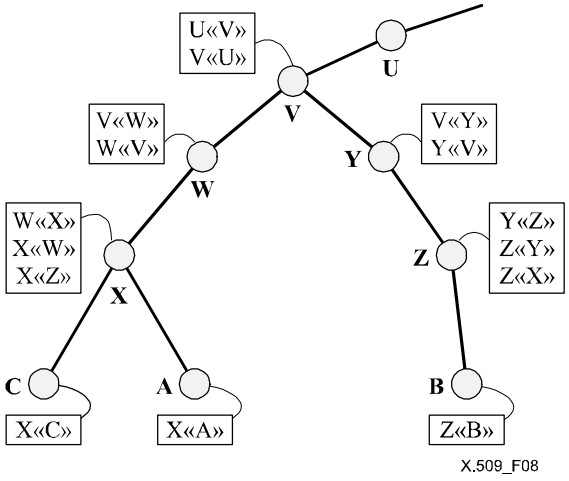


Рисунок 8 – Иерархия CA – Гипотетический пример

Если CA пользователей расположены иерархически, что пользователь A может получить следующие сертификаты из Справочника для создания тракта сертификации к пользователю B:

X<<W>>, W<<V>>, V<<Y>>, Y<<Z>>, Z<<B>>.

Если пользователь A получил эти сертификаты, он может последовательно раскрыть тракт сертификации, чтобы получить содержимое сертификата пользователя B, включая Bp:

Bp = Xp • X<<W>> W<<V>> V<<Y>> Y<<Z>> Z<<B>>.

В общем случае, пользователь A также может получить следующие сертификаты из Справочника для создания обратного тракта сертификации от пользователя B к пользователю A:

Z<<Y>>, Y<<V>>, V<<W>>, W<<X>>, X<<A>>.

Если пользователь B получил эти сертификаты от пользователя A, он может последовательно раскрыть обратный тракт сертификации, чтобы получить содержимое сертификата пользователя A, включая Ap:

Ap = Zp • Z<<Y>> Y<<V>> V<<W>> W<<X>> X<<A>>.

Применяя оптимизацию, рассмотренную в п. 18.2.1:

a) возьмем, например, пользователей A и C: им обоим известен Xp, поэтому пользователь A может непосредственно получить сертификат пользователя C. Раскрывание тракта сертификации сокращается до:

   Cp = Xp • X<<C>>

   и раскрывание обратного тракта сертификации сокращается до:

   Ap = Xp • X<<A>>;

b) если предположить, что пользователь A таким образом знает W<<X>>, Wp, V<<W>>, Vp, U<<V>>, Up и т. д., то сокращается объем информации, которую пользователь A должен получить из Справочника для формирования тракта сертификации, до:

   V<<Y>>, Y<<Z>>, Z<<B>>;

   и информации, которую пользователь A должен получить из Справочника для формирования обратного тракта сертификации, до:

   Z<<Y>>, Y<<V>>;

c) если предположить, что пользователь A часто взаимодействует с пользователями, сертифицированными Z, он может изучить (в дополнение к открытым ключам, изученным в пункте b) выше) V<<Y>>, Y<<V>>, Y<<Z>> и Z<<Y>>. Чтобы взаимодействовать с пользователем B, ему достаточно получить из Справочника только Z<<B>>;

d) если предположить, что пользователи, сертифицированные X и Z, часто взаимодействуют, то в записи справочника для X должно храниться X<<Z>> и наоборот (это показано на рисунке 8). Если пользователь A желает пройти аутентификацию к пользователю B, то пользователю A достаточно получить только:

   X<<Z>>, Z<<B>>,

   чтобы сформировать тракт сертификации, и:

   Z<<X>>,

   чтобы сформировать обратный тракт сертификации;

e) если предположить, что пользователи A и C взаимодействовали ранее и изучили сертификаты друг друга, они могут непосредственно использовать открытые ключи друг друга, т. е.

   Cp = Xp • X<<C>>

   и

   Ap = Xp • X<<A>>.

В более общем случае органы по сертификации не связаны иерархически. Предположим, что пользователь D, сертифицированный органом U, как показано на рисунке 9, желает пройти аутентификацию к пользователю E, сертифицированному органом W. Запись Справочника пользователя D должна содержать сертификат U<<D>>, а запись пользователя E должна содержать сертификат W<<E>>.

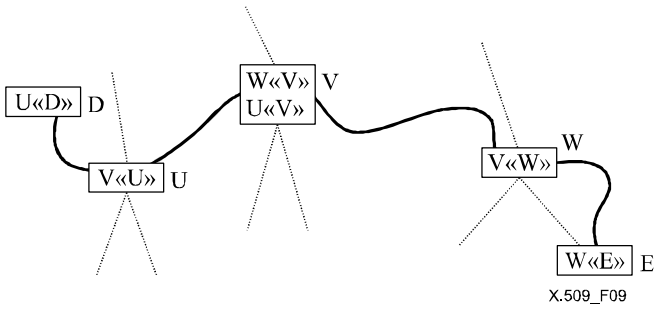


Рисунок 9 – Неиерархический тракт сертификации – Пример

Пусть V будет CA, с которым такие CA, как U и W ранее обменивались открытыми ключами доверенным способом. В результате этого были сгенерированы и сохранены в Справочнике сертификаты U<<V>>, V<<U>>, W<<V>> и V<<W>>. Предположим, что сертификаты U<<V>> и W<<V>> хранятся в записи V, V<<U>> хранится в записи U, а V<<W>> хранится в записи W.

Пользователь D должен найти тракт сертификации к E. Могут использоваться различные стратегии. Одной из стратегий может быть рассмотрение пользователей и CA как узлов, а сертификатов как дуг ориентированного графа. В данных терминах D должен выполнить поиск в графе для нахождения тракта от U к E, одним из которых будет U<<V>>, V<<W>>, W<<E>>. Если такой тракт обнаружен, то также необходимо создать обратный тракт W<<V>>, V<<U>>, U<<D>>.

18.2.2 Процедуры строгой аутентификации

Основной подход к аутентификации был определен выше, а именно подтверждение идентификационной информации путем демонстрирования обладания частным ключом. Тем не менее, для использования данного подхода возможно много процедур аутентификации. В общем случае, определение подходящих процедур является задачей определенного приложения, с учетом требований политики его безопасности. В данном пункте описаны три определенные процедуры аутентификации, которые могут быть полезными для целого ряда приложений.

ПРИМЕЧАНИЕ. – В данной спецификации Справочника процедуры не определяются в подробностях, требуемых для реализации. Тем не менее, могут быть предусмотрены дополнительные стандарты, обеспечивающие необходимые подробности, либо в общем виде, либо в применении к определенным приложениям.

Данные три процедуры включают различное число обменов информацией аутентификации, и, следовательно, обеспечивают различные типы гарантий своим участникам. В частности:

a) односторонняя аутентификация, описанная в п. 18.2.2.1, включает однократную передачу информации от пользователя (A), предназначенную для другого пользователя (B), и устанавливает следующее:
   - идентификационную информацию пользователя A, и что маркер аутентификации был на самом деле сгенерирован A;
   - идентификационную информацию пользователя B, а и что маркер аутентификации был на самом деле был предназначен для отправки B;
   - целостность и "новизну" (свойство, означающее, что информация не повторяется два или более раз) передаваемого маркера аутентификации.

     Последние свойства могут также быть установлены для произвольных дополнительных данных, сопровождающих передачу;

b) двусторонняя аутентификация, описанная в п. 18.2.2.2, включает дополнительно ответ пользователя B пользователю A. Дополнительно она устанавливает следующее:
   - что маркер аутентификации, сгенерированный при ответе, был на самом деле сгенерирован B и был предназначен для отправки A;
   - целостность и новизну маркера аутентификации, отправленного при ответе;
   - (необязательно) взаимную секретность части маркеров;

c) трехсторонняя аутентификация, описанная в п. 18.2.2.3, включает дополнительно дальнейшую передачу от A к B. Она устанавливает те же свойства, как и двусторонняя аутентификация, но выполняет это без необходимости проверке временной отметки соединения.

Во всех случаях, когда имеет место Строгая аутентификация, пользователь A должен получить открытый ключ пользователя A и возвратить тракт сертификации от B к A еще до какого-либо обмена информацией. Это может включать доступ к Справочнику, как описано в п. 18.2. Ниже при описании процедур любой подобный доступ более не упоминается.

Проверка временных отметок, как упоминается в следующих пунктах, применяется, только если в локальной среде используются синхронизированные часы или если часы логически синхронизируются на основании двусторонних соглашений. В любом случае, рекомендуется использование Универсального глобального времени.

Для каждой из трех процедур аутентификации, описанных ниже, предполагается, что сторона A проверила подлинность всех сертификатов в тракте сертификации.

18.2.2.1 Односторонняя аутентификация

Выполняются следующие этапы, как показано на рисунке 10:

1) A генерирует r^A, неповторяющееся число, используемое для обнаружения атак воспроизведения и для предотвращения фальсификации.
2) A отправляет следующее сообщение B:

$$B\,A,\ A\{t^A,\ r^A,\ B\},$$

где t^A представляет собой временную отметку. t^A состоит из одной или двух дат: времени генерации маркера (необязательно) и срока истечения его действия. Альтернативно, если аутентификация источника данных "sgnData" обеспечивается цифровой подписью, сообщение примет вид:

$$B\,A,\ A\{t^A,\ r^A,\ B,\ sgnData\}.$$

В случаях, когда передается информация, которая впоследствии будет использоваться как частный ключ (данная информация в дальнейшем именуется encData"), сообщение примет вид:

$$B\,A,\ A\{t^A,\ r^A,\ B,\ sgnData,\ Bp[encData]\}.$$

Использование "encData" в качестве частного ключа подразумевает, что он должен тщательно выбираться, например, быть сильным ключом для любой используемой криптосистемы, как указано в поле маркера "sgnData".

3) B выполняет следующие действия:

   a) получает Ap у BA, проверяя, что период действия сертификата A не истек;

   b) проверяет подпись и, таким образом, целостность подписанной информации;

   c) проверяет, что именно B является предполагаемым получателем;

   d) проверяет, что временная отметка является "текущей";

   e) дополнительно, проверяет, что r^A не был воспроизведен. Это может быть достигнуто, например, включением в r^A последовательной части, проверяемой на уникальность значения локальной реализацией.

      r^A является действительным до истечения срока его действия, указанного t^A. r^A всегда сопровождается последовательной частью, которая указывает, что A не должен повторять маркер в течение временного интервала t^A, и поэтому не требуется проверка самого значения r^A.

      В любом случае для стороны B является целесообразным хранить последовательную часть вместе с временной отметкой t^A в чистом виде и вместе с хэшированной частью маркера в течение временного интервала t^A.

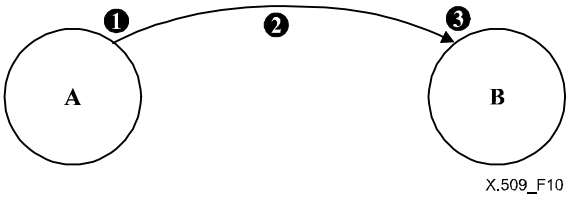


Рисунок 10 – Односторонняя аутентификация

18.2.2.2 Двусторонняя аутентификация

Выполняются следующие этапы, как показано на рисунке 11:

1) как для п. 18.2.2.1;

2) как для п. 18.2.2.1;

3) как для п. 18.2.2.1;

4) B генерирует r^B, неповторяющееся число, используемое для той (тех) же цели(ей) как r^A;

5) B отправляет следующий маркер аутентификации A:

$$B\{t^B,\ r^B,\ A,\ r^A\},$$

где t^B представляет собой временную отметку, определенную таким же образом, как t^A.

Альтернативно, если аутентификация источника данных "sgnData" обеспечивается цифровой подписью:

$$B\{t^B,\ r^B,\ A,\ r^A,\ sgnData\}.$$

В случаях, когда передается информация, которая впоследствии будет использоваться как частный ключ (данная информация в дальнейшем именуется encData"):

$$B\{t^B,\ r^B,\ A,\ r^A,\ sgnData,\ Ap[encData]\}.$$

Использование "encData" в качестве частного ключа подразумевает, что он должен тщательно выбираться, например, быть сильным ключом для любой используемой криптосистемы, как указано в поле маркера "sgnData".

6) A выполняет следующие действия:
   a) проверяет подпись и, таким образом, целостность подписанной информации;
   b) проверяет, что именно A является предполагаемым получателем;
   c) проверяет, что временная отметка t^B является "текущей";
   d) дополнительно, проверяет, что r^B не был воспроизведен (см. п 18.2.2.1, этап 3), d)).

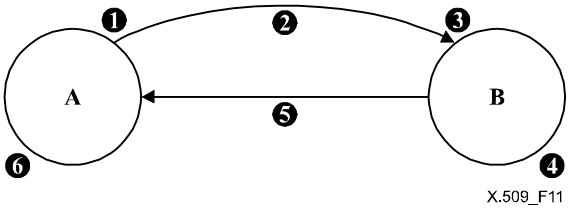


Рисунок 11 – Двусторонняя аутентификация

18.2.2.3 Трехсторонняя аутентификация

Выполняются следующие этапы, как показано на рисунке 12:

1) как для п. 18.2.2.2;
2) как для п. 18.2.2.2. Временная отметка t^A может быть равна нулю;
3) как для п. 18.2.2.2, за исключением отсутствия необходимости проверки временной отметки;
4) как для п. 18.2.2.2;
5) как для п. 18.2.2.2. Временная отметка t^B может быть равна нулю;
6) как для п. 18.2.2.2, за исключением отсутствия необходимости проверки временной отметки;
7) A проверяет, что полученный r^A является идентичным отправленному r^A;
8) A отправляет следующий маркер аутентификации B:

$$A\{r^B, B\};$$

9) B выполняет следующие действия:
   a) проверяет подпись и, таким образом, целостность подписанной информации;
   b) проверяет, что полученный r^B является идентичным отправленному B r^B.

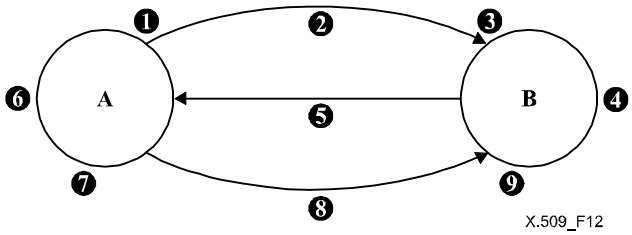


Рисунок 12 – Трехсторонняя аутентификация

19 Управление доступом

Справочник существует в среде, где различные административные органы управляют доступом к своей части DIB. Определение схемы управления доступом в контексте Справочника включает методы для:

- определения информации по управлению доступом (ACI);
- внедрения прав доступа, определенных данной информацией по управлению доступом;
- поддержки информации по управлению доступом.

Внедрение прав доступа применяется к управлению доступом к:

- информации Справочника, относящейся к именам;
- информации о пользователях Справочника;
- операционной информации Справочника, включая информацию по управлению доступом.

Административные органы могут использовать любые стандартизованные схемы управления доступом полностью или частично при реализации своих политик безопасности, а также могут свободно определять собственные схемы на свое усмотрение.

Схема базового управления доступом (BAC), определенная в Рек. МСЭ-Т X.501 | ИСО/МЭК 9594-2, представляет собой схему, основанную на списке управления доступом, которая позволяет администрациям Справочника налагать ограничения на уровень аутентификации, выполняемой для привязывания к Справочнику. Структура сертификатов открытых ключей, определенная в данной Спецификации, используется для предоставления схемы строгой аутентификации, используемой для данного привязывания.

Схема управления доступом на основе правил (RBAC), определенная в Рек. МСЭ-Т X.501 | ИСО/МЭК 9594-2, использует структуру сертификатов атрибутов, определенную в данной Спецификации, для хранения атрибутов допуска при принятии решений об управлении доступом. RBAC также может быть использована совместно с BAC.

20 Защита операций Справочника

Структура сертификатов открытых ключей, определенная в данной Спецификации, используется во всех протокола Справочника, определенных в данной серии Рекомендаций, для дополнительной защиты операций, включая запросы, ответы и ошибки. Защита целостности обеспечивается посредством цифровой подписи отправителем и проверки данной подписи получателем при использовании соответствующего сертификата открытого ключа отправителя. Защита секретности обеспечивается посредством использования шифрования открытого ключа, когда содержимое шифруется при помощи открытого ключа, полученного из сертификата открытого ключа намеченного получателя, и дешифруется получателем при использовании его соответствующего частного ключа.

Определенные механизмы и синтаксис для запрашивания и включения элементов защиты в обмены протоколами определяются в пределах каждого протокола Справочника в данной серии Спецификаций.

Приложение A

Структуры открытого ключа и сертификатов атрибутов

(Данное Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

ASN.1 , ,
ASN.1: **AuthenticationFramework**, **CertificateExtensions** **AttributeCertificateDefinitions**.

-- A.1 Модуль структуры аутентификации

```
AuthenticationFramework {joint-iso-itu-t ds(5) module(1) authenticationFramework(7) 5}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --

-- , , ASN.1,
-- , ,
-- .
-- , ,
-- .

IMPORTS
    id-at, id-nf, id-oc, informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl,
    certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}

    Name, ATTRIBUTE, OBJECT-CLASS, NAME-FORM, top
        FROM InformationFramework informationFramework

    ub-user-password, ub-content
        FROM UpperBounds upperBounds

    UniqueIdentifier, octetStringMatch, DirectoryString{}, commonName
        FROM SelectedAttributeTypes selectedAttributeTypes

    certificateExactMatch, certificatePairExactMatch, certificateListExactMatch, KeyUsage, GeneralNames,

       CertificatePoliciesSyntax, algorithmIdentifierMatch, CertPolicyId
        FROM CertificateExtensions certificateExtensions ;

-- определение сертификата открытого ключа --

Certificate                 ::=  SIGNED { SEQUENCE {
    version                 [0]  Version DEFAULT v1,
    serialNumber                 CertificateSerialNumber,
    signature                    AlgorithmIdentifier,
    issuer                       Name,
    validity                     Validity,
    subject                      Name,
    subjectPublicKeyInfo         SubjectPublicKeyInfo,
    issuerUniqueIdentifier  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                                 -- при наличии, версия должна быть v2 или v3
    subjectUniqueIdentifier [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                                 -- при наличии, версия должна быть v2 или v3
    extensions              [3]  Extensions OPTIONAL
                                 -- При наличии, версия должна быть v3 -- } }

Version                     ::=  INTEGER { v1(0), v2(1), v3(2) }

CertificateSerialNumber     ::=  INTEGER

AlgorithmIdentifier             ::=  SEQUENCE {
    algorithm               ALGORITHM.&id ({SupportedAlgorithms}),
    parameters              ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }

-- Определение следующей совокупности информационных объектов передается, возможно, в стандартизированные
-- профили или в свидетельства о соответствии протокольной реализации. Эта совокупность необходима для определения
-- табличного ограничения, накладываемого на компонент параметров AlgorithmIdentifier.

SupportedAlgorithms         ALGORITHM  ::=  { ... }
```

```
Validity                    ::=   SEQUENCE {
    notBefore       Time,
    notAfter        Time }

SubjectPublicKeyInfo        ::=   SEQUENCE {
    algorithm               AlgorithmIdentifier,
    subjectPublicKey        BIT STRING }

Time  ::=  CHOICE {
    utcTime                 UTCTime,
    generalizedTime         GeneralizedTime }

Extensions ::= SEQUENCE OF Extension
-- Для расширений, где имеет значение порядок индивидуальных расширений в SEQUENCE, спецификация
-- данных индивидуальных расширений должна включать правила значимости порядка в них.

Extension ::= SEQUENCE {
    extnId          EXTENSION.&id ({ExtensionSet}),
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING
                    -- содержит кодирование с использованием значения типа &ExtnType
                    -- для объекта расширения, определенного extnId -- }

ExtensionSet EXTENSION      ::=   { ... }

EXTENSION ::= CLASS {
    &id         OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX              &ExtnType
    IDENTIFIED BY       &id }

-- другие структуры сертификатов PKI

Certificates                ::=         SEQUENCE {
    userCertificate                     Certificate,
    certificationPath                   ForwardCertificationPath OPTIONAL}

ForwardCertificationPath    ::=         SEQUENCE OF CrossCertificates

CrossCertificates           ::=         SET OF Certificate

CertificationPath           ::=         SEQUENCE {
    userCertificate                     Certificate,
    theCACertificates                   SEQUENCE OF CertificatePair OPTIONAL}

CertificatePair         ::=         SEQUENCE {
    forward             [0]         Certificate OPTIONAL,
    reverse             [1]         Certificate OPTIONAL
                                    -- должен присутствовать по меньшей мере один из пары -- }
    (WITH COMPONENTS { ..., forward PRESENT} |
    WITH COMPONENTS { ..., reverse PRESENT})

-- список аннулированных сертификатов (CRL)

CertificateList         ::=         SIGNED { SEQUENCE {
    version                         Version OPTIONAL,
                                -- при наличии, версия должна быть v2
    signature                       AlgorithmIdentifier,
    issuer                          Name,
    thisUpdate                      Time,
    nextUpdate                      Time OPTIONAL,
    revokedCertificates                 SEQUENCE OF SEQUENCE {
        serialNumber                        CertificateSerialNumber,
        revocationDate                      Time,
        crlEntryExtensions                  Extensions OPTIONAL } OPTIONAL,
    crlExtensions       [0]             Extensions OPTIONAL }}

-- информационные классы объектов --

ALGORITHM  ::=  TYPE-IDENTIFIER

-- параметризированные типы --
HASH {ToBeHashed}       ::=         SEQUENCE {
    algorithmIdentifier             AlgorithmIdentifier,
```

```
    hashValue                       BIT STRING ( CONSTRAINED BY {
        -- должен быть результат применения процедуры хэширования к октетам, кодированным с
        -- использованием DER --

        -- значения --ToBeHashed } ) }
ENCRYPTED-HASH { ToBeSigned }       ::=     BIT STRING ( CONSTRAINED BY {
    -- должен быть результат применения процедуры хэширования к октетам, кодированным с
    -- использованием DER значения (см. п. 6.1) -- ToBeSigned -- и затем применения процедуры шифрования к
    -- данным октетам -- })

ENCRYPTED { ToBeEnciphered }        ::=     BIT STRING ( CONSTRAINED BY {
    -- должен быть результат применения процедуры шифрования к октетам, кодированным с
    -- использованием BER значения -- ToBeEnciphered})


SIGNATURE { ToBeSigned }            ::=     SEQUENCE {
    algorithmIdentifier             AlgorithmIdentifier,
    encrypted                       ENCRYPTED-HASH { ToBeSigned }}


SIGNED { ToBeSigned }               ::=     SEQUENCE {
    toBeSigned                      ToBeSigned,
    COMPONENTS OF                   SIGNATURE { ToBeSigned }}

-- классы объектов PKI --

pkiUser  OBJECT-CLASS       ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {userCertificate}
    ID              id-oc-pkiUser }


pkiCA OBJECT-CLASS          ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {cACertificate |
                    certificateRevocationList |
                    authorityRevocationList |
                    crossCertificatePair }
    ID              id-oc-pkiCA }


cRLDistributionPoint        OBJECT-CLASS        ::= {
    SUBCLASS OF             { top }
    KIND                    structural
    MUST CONTAIN            { commonName }
    MAY CONTAIN             { certificateRevocationList |
                            authorityRevocationList |
                            deltaRevocationList }
    ID                      id-oc-cRLDistributionPoint }


cRLDistPtNameForm  NAME-FORM    ::= {
    NAMES                   cRLDistributionPoint
    WITH ATTRIBUTES         { commonName}
    ID                      id-nf-cRLDistPtNameForm }


deltaCRL    OBJECT-CLASS        ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {deltaRevocationList}
    ID                  id-oc-deltaCRL }


cpCps   OBJECT-CLASS        ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {certificatePolicy |
```

```
                        certificationPracticeStmt}
    ID                  id-oc-cpCps }

pkiCertPath     OBJECT-CLASS    ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         { pkiPath }
    ID                  id-oc-pkiCertPath }
```

-- атрибуты справочника PKI --

```
userCertificate             ATTRIBUTE   ::=     {
    WITH SYNTAX                         Certificate
    EQUALITY MATCHING RULE              certificateExactMatch
    ID                                  id-at-userCertificate}

cACertificate               ATTRIBUTE   ::=     {
    WITH SYNTAX                         Certificate
    EQUALITY MATCHING RULE              certificateExactMatch
    ID                                  id-at-cAcertificate }

crossCertificatePair        ATTRIBUTE   ::=     {
    WITH SYNTAX                         CertificatePair
    EQUALITY MATCHING RULE              certificatePairExactMatch
    ID                                  id-at-crossCertificatePair }

certificateRevocationList   ATTRIBUTE   ::=     {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-certificateRevocationList }

authorityRevocationList     ATTRIBUTE   ::=     {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-authorityRevocationList }

deltaRevocationList ATTRIBUTE ::= {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-deltaRevocationList }

supportedAlgorithms  ATTRIBUTE ::= {
    WITH SYNTAX                         SupportedAlgorithm
    EQUALITY MATCHING RULE              algorithmIdentifierMatch
    ID                                  id-at-supportedAlgorithms }

SupportedAlgorithm ::= SEQUENCE {
    algorithmIdentifier                         AlgorithmIdentifier,
    intendedUsage                       [0]     KeyUsage OPTIONAL,
    intendedCertificatePolicies         [1]     CertificatePoliciesSyntax OPTIONAL }

certificationPracticeStmt ATTRIBUTE   ::=     {
    WITH SYNTAX         InfoSyntax
    ID                  id-at-certificationPracticeStmt }

InfoSyntax    ::=   CHOICE {
    content         DirectoryString {ub-content},
    pointer         SEQUENCE {
```

```
        name                GeneralNames,
        hash                HASH { HashedPolicyInfo } OPTIONAL } }

POLICY  ::=   TYPE-IDENTIFIER

HashedPolicyInfo      ::=    POLICY.&Type( {Policies} )

Policies POLICY ::= {...} -- Определяется использующими документ --


certificatePolicy   ATTRIBUTE   ::=   {
    WITH SYNTAX       PolicySyntax
    ID                id-at-certificatePolicy }


PolicySyntax ::=   SEQUENCE {
    policyIdentifier      PolicyID,
    policySyntax          InfoSyntax
    }


PolicyID        ::=   CertPolicyId


pkiPath         ATTRIBUTE   ::= {
    WITH SYNTAX       PkiPath
    ID                id-at-pkiPath }


PkiPath         ::=   SEQUENCE OF Certificate


userPassword      ATTRIBUTE   ::=   {
    WITH SYNTAX                     OCTET STRING (SIZE (0..ub-user-password))
    EQUALITY MATCHING RULE          octetStringMatch
    ID                              id-at-userPassword }

-- присвоения идентификаторов объектов --
-- классы объектов --

id-oc-cRLDistributionPoint          OBJECT IDENTIFIER ::=          {id-oc 19}
id-oc-pkiUser                       OBJECT IDENTIFIER ::=          {id-oc 21}
id-oc-pkiCA                         OBJECT IDENTIFIER ::=          {id-oc 22}
id-oc-deltaCRL                      OBJECT IDENTIFIER ::=          {id-oc 23}
id-oc-cpCps                         OBJECT IDENTIFIER ::=          {id-oc 30}
id-oc-pkiCertPath                   OBJECT IDENTIFIER ::=          {id-oc 31}

-- формы имен --

id-nf-cRLDistPtNameForm             OBJECT IDENTIFIER ::=          {id-nf 14}

-- атрибуты справочника --

id-at-userPassword                  OBJECT IDENTIFIER ::=          {id-at 35}
id-at-userCertificate               OBJECT IDENTIFIER ::=          {id-at 36}
id-at-cAcertificate                 OBJECT IDENTIFIER ::=          {id-at 37}
id-at-authorityRevocationList       OBJECT IDENTIFIER ::=          {id-at 38}
id-at-certificateRevocationList     OBJECT IDENTIFIER ::=          {id-at 39}
id-at-crossCertificatePair          OBJECT IDENTIFIER ::=          {id-at 40}
id-at-supportedAlgorithms           OBJECT IDENTIFIER ::=          {id-at 52}
id-at-deltaRevocationList           OBJECT IDENTIFIER ::=          {id-at 53}
id-at-certificationPracticeStmt     OBJECT IDENTIFIER ::=          {id-at 68}
id-at-certificatePolicy             OBJECT IDENTIFIER ::=          {id-at 69}
id-at-pkiPath                       OBJECT IDENTIFIER ::=          {id-at 70}

END
```

-- A.2 Модуль расширения сертификатов

```
CertificateExtensions {joint-iso-itu-t ds(5) module(1) certificateExtensions(26) 5}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
        usefulDefinitions(0) 5}

    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute, MATCHING-RULE
        FROM InformationFramework informationFramework

    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, Time, PolicyID
        FROM AuthenticationFramework authenticationFramework

    DirectoryString {}
        FROM SelectedAttributeTypes selectedAttributeTypes

    ub-name
        FROM UpperBounds upperBounds

    ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;
-- Если явно не отмечено иное, порядок компонентов
-- структуры SEQUENCE OF не имеет значения в данной Спецификации.

-- расширения сертификатов открытых ключей и CRL --


authorityKeyIdentifier EXTENSION ::= {
    SYNTAX              AuthorityKeyIdentifier
    IDENTIFIED BY       id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier               OPTIONAL,
    authorityCertIssuer         [1] GeneralNames                OPTIONAL,
    authorityCertSerialNumber   [2] CertificateSerialNumber     OPTIONAL }
    ( WITH COMPONENTS           {..., authorityCertIssuer PRESENT,
                    authorityCertSerialNumber PRESENT} |
    WITH COMPONENTS             {..., authorityCertIssuer ABSENT,
                    authorityCertSerialNumber ABSENT} )

KeyIdentifier ::= OCTET STRING


subjectKeyIdentifier EXTENSION ::= {
    SYNTAX              SubjectKeyIdentifier
    IDENTIFIED BY       id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier


keyUsage EXTENSION ::= {
        SYNTAX              KeyUsage
        IDENTIFIED BY       id-ce-keyUsage }

KeyUsage ::= BIT STRING {
    digitalSignature        (0),
    contentCommitment       (1),
    keyEncipherment         (2),
    dataEncipherment        (3),
    keyAgreement            (4),
    keyCertSign             (5),
    cRLSign                 (6),
    encipherOnly            (7),
    decipherOnly            (8) }
```

```
extKeyUsage EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF KeyPurposeId
    IDENTIFIED BY       id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER


privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX              PrivateKeyUsagePeriod
    IDENTIFIED BY       id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore       [0]   GeneralizedTime OPTIONAL,
    notAfter        [1]   GeneralizedTime OPTIONAL }
    ( WITH COMPONENTS {..., notBefore PRESENT} |
    WITH COMPONENTS     {..., notAfter PRESENT} )


certificatePolicies EXTENSION ::= {
    SYNTAX              CertificatePoliciesSyntax
    IDENTIFIED BY       id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier      CertPolicyId,
    policyQualifiers      SEQUENCE SIZE (1..MAX) OF
                              PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId           CERT-POLICY-QUALIFIER.&id
                                ({SupportedPolicyQualifiers}),
    qualifier                   CERT-POLICY-QUALIFIER.&Qualifier
                                ({SupportedPolicyQualifiers}{@policyQualifierId})
                                OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }

anyPolicy      OBJECT IDENTIFIER    ::=   { 2 5 29 32 0 }

CERT-POLICY-QUALIFIER ::= CLASS {
    &id               OBJECT IDENTIFIER UNIQUE,
    &Qualifier        OPTIONAL }
WITH SYNTAX {
    POLICY-QUALIFIER-ID   &id
    [QUALIFIER-TYPE   &Qualifier] }


policyMappings EXTENSION ::= {
    SYNTAX              PolicyMappingsSyntax
    IDENTIFIED BY       id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy          CertPolicyId,
    subjectDomainPolicy         CertPolicyId }


subjectAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0]   INSTANCE OF OTHER-NAME,
    rfc822Name                  [1]   IA5String,
    dNSName                     [2]   IA5String,
    x400Address                 [3]   ORAddress,
    directoryName               [4]   Name,
    ediPartyName                [5]   EDIPartyName,
    uniformResourceIdentifier   [6]   IA5String,
    iPAddress                   [7]   OCTET STRING,
    registeredID                [8]   OBJECT IDENTIFIER }
```

```
OTHER-NAME ::= TYPE-IDENTIFIER

EDIPartyName ::= SEQUENCE {
    nameAssigner        [0]   DirectoryString {ub-name} OPTIONAL,
    partyName           [1]   DirectoryString {ub-name} }


issuerAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-issuerAltName }


subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX              AttributesSyntax
    IDENTIFIED BY       id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute


basicConstraints EXTENSION ::= {
    SYNTAX              BasicConstraintsSyntax
    IDENTIFIED BY       id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
    cA                          BOOLEAN DEFAULT FALSE,
    pathLenConstraint           INTEGER (0..MAX) OPTIONAL }


nameConstraints EXTENSION ::= {
    SYNTAX          NameConstraintsSyntax
    IDENTIFIED BY       id-ce-nameConstraint }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees       [0]   GeneralSubtrees OPTIONAL,
    excludedSubtrees        [1]   GeneralSubtrees OPTIONAL,
    requiredNameForms       [2]   NameForms OPTIONAL  }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base                    GeneralName,
    minimum  [0]   BaseDistance DEFAULT 0,
    maximum  [1]   BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)


NameForms  ::= SEQUENCE {
   basicNameForms    [0]   BasicNameForms OPTIONAL,
   otherNameForms    [1]   SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL }
(ALL EXCEPT ({-- отсутствует, т. е. должен присутствовать по меньшей мере один компонент -- }))


BasicNameForms  ::= BIT STRING {
    rfc822Name              (0),
    dNSName                 (1),
    x400Address             (2),
    directoryName           (3),
    ediPartyName            (4),
    uniformResourceIdentifier    (5),
    iPAddress               (6),
    registeredID            (7) }  (SIZE (1..MAX))


policyConstraints EXTENSION ::= {
    SYNTAX              PolicyConstraintsSyntax
    IDENTIFIED BY       id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
    requireExplicitPolicy     [0] SkipCerts OPTIONAL,
    inhibitPolicyMapping      [1] SkipCerts OPTIONAL }
```

```
SkipCerts ::= INTEGER (0..MAX)

cRLNumber EXTENSION ::= {
    SYNTAX              CRLNumber
    IDENTIFIED BY       id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)

reasonCode EXTENSION ::= {
    SYNTAX              CRLReason
    IDENTIFIED BY       id-ce-reasonCode }

CRLReason ::= ENUMERATED {
    unspecified                 (0),
    keyCompromise               (1),
    cACompromise                (2),
    affiliationChanged          (3),
    superseded                  (4),
    cessationOfOperation        (5),
    certificateHold             (6),
    removeFromCRL               (8),
    privilegeWithdrawn          (9),
    aaCompromise                (10) }

holdInstructionCode EXTENSION ::= {
    SYNTAX              HoldInstruction
    IDENTIFIED BY       id-ce-instructionCode }

HoldInstruction ::= OBJECT IDENTIFIER

invalidityDate EXTENSION ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-invalidityDate }

crlScope EXTENSION ::= {
    SYNTAX                  CRLScopeSyntax
    IDENTIFIED BY           id-ce-cRLScope }

CRLScopeSyntax ::=          SEQUENCE SIZE (1..MAX) OF PerAuthorityScope

PerAuthorityScope ::= SEQUENCE {
    authorityName               [0]     GeneralName OPTIONAL,
    distributionPoint           [1]     DistributionPointName OPTIONAL,
    onlyContains                [2]     OnlyCertificateTypes OPTIONAL,
    onlySomeReasons             [4]     ReasonFlags OPTIONAL,
    serialNumberRange           [5]     NumberRange OPTIONAL,
    subjectKeyIdRange           [6]     NumberRange OPTIONAL,
    nameSubtrees                [7]     GeneralNames OPTIONAL,
    baseRevocationInfo          [9]     BaseRevocationInfo OPTIONAL
    }

OnlyCertificateTypes ::= BIT STRING {
    user            (0),
    authority       (1),
    attribute       (2) }

NumberRange ::= SEQUENCE {
    startingNumber      [0]     INTEGER OPTIONAL,
    endingNumber        [1]     INTEGER OPTIONAL,
    modulus                     INTEGER OPTIONAL }

BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier         [0]     CRLStreamIdentifier    OPTIONAL,
    cRLNumber                   [1]     CRLNumber,
    baseThisUpdate              [2]     GeneralizedTime }

statusReferrals EXTENSION  ::= {
    SYNTAX              StatusReferrals
    IDENTIFIED BY       id-ce-statusReferrals }
```

```
StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral

StatusReferral  ::=  CHOICE {
    cRLReferral          [0]        CRLReferral,
    otherReferral        [1]        INSTANCE OF OTHER-REFERRAL}

CRLReferral ::= SEQUENCE  {
    issuer               [0]        GeneralName OPTIONAL,
    location             [1]        GeneralName OPTIONAL,
    deltaRefInfo         [2]        DeltaRefInfo OPTIONAL,
    cRLScope                        CRLScopeSyntax,
    lastUpdate           [3]        GeneralizedTime OPTIONAL,
    lastChangedCRL       [4]        GeneralizedTime OPTIONAL}

DeltaRefInfo  ::=  SEQUENCE {
    deltaLocation        GeneralName,
    lastDelta            GeneralizedTime OPTIONAL }

OTHER-REFERRAL ::= TYPE-IDENTIFIER


cRLStreamIdentifier EXTENSION      ::= {
    SYNTAX               CRLStreamIdentifier
    IDENTIFIED BY        id-ce-cRLStreamIdentifier }


CRLStreamIdentifier  ::=  INTEGER (0..MAX)


orderedList EXTENSION      ::= {
    SYNTAX               OrderedListSyntax
    IDENTIFIED BY        id-ce-orderedList }

OrderedListSyntax  ::= ENUMERATED {
ascSerialNum       (0),
ascRevDate         (1) }


deltaInfo EXTENSION  ::=   {
    SYNTAX               DeltaInformation
    IDENTIFIED BY        id-ce-deltaInfo }

DeltaInformation   ::=   SEQUENCE {
    deltaLocation        GeneralName,
    nextDelta            GeneralizedTime OPTIONAL }


cRLDistributionPoints EXTENSION ::= {
    SYNTAX               CRLDistPointsSyntax
    IDENTIFIED BY        id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
    distributionPoint    [0]        DistributionPointName OPTIONAL,
    reasons              [1]        ReasonFlags OPTIONAL,
    cRLIssuer            [2]        GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
    fullName                        [0]        GeneralNames,
    nameRelativeToCRLIssuer         [1]        RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
    unused                  (0),
    keyCompromise           (1),
    cACompromise            (2),
    affiliationChanged      (3),
    superseded              (4),
    cessationOfOperation    (5),
    certificateHold         (6),
    privilegeWithdrawn      (7),
    aACompromise            (8) }
```

```
issuingDistributionPoint EXTENSION  ::= {
    SYNTAX IssuingDistPointSyntax
    IDENTIFIED BY id-ce-issuingDistributionPoint }


IssuingDistPointSyntax ::= SEQUENCE {
    -- Если onlyContainsUserPublicKeyCerts и onlyContainsCACerts имеют значение FALSE,
    -- CRL охватывает оба типа сертификатов
    distributionPoint                [0] DistributionPointName OPTIONAL,
    onlyContainsUserPublicKeyCerts   [1] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts              [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons                  [3] ReasonFlags OPTIONAL,
    indirectCRL                      [4] BOOLEAN DEFAULT FALSE }

certificateIssuer EXTENSION ::= {
    SYNTAX          GeneralNames
    IDENTIFIED BY   id-ce-certificateIssuer }


deltaCRLIndicator EXTENSION ::= {
    SYNTAX          BaseCRLNumber
    IDENTIFIED BY   id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber

toBeRevoked  EXTENSION  ::= {
    SYNTAX   ToBeRevokedSyntax
    IDENTIFIED BYid-ce-toBeRevoked }

ToBeRevokedSyntax::=  SEQUENCE SIZE(1..MAX) OF ToBeRevokedGroup

ToBeRevokedGroup      ::=     SEQUENCE {
    certificateIssuer     [0]     GeneralName OPTIONAL,
    reasonInfo            [1]     ReasonInfo     OPTIONAL,
    revocationTime                GeneralizedTime,
    certificateGroup              CertificateGroup }

ReasonInfo        ::=     SEQUENCE {
    reasonCode                    CRLReason,
    holdInstructionCode   HoldInstruction OPTIONAL }

CertificateGroup   ::=    CHOICE {
    serialNumbers         [0]     CertificateSerialNumbers,
    serialNumberRange     [1]     CertificateGroupNumberRange,
    nameSubtree           [2]     GeneralName }

CertificateGroupNumberRange ::= SEQUENCE {
    startingNumber        [0]     INTEGER,
    endingNumber          [1]     INTEGER }

CertificateSerialNumbers ::=      SEQUENCE SIZE(1..MAX) OF CertificateSerialNumber

revokedGroups  EXTENSION  ::= {
    SYNTAX   RevokedGroupsSyntax
    IDENTIFIED BYid-ce-RevokedGroups }

RevokedGroupsSyntax ::=           SEQUENCE SIZE (1..MAX) OF RevokedGroup

RevokedGroup ::=SEQUENCE {
    certificateIssuer             [0]     GeneralName OPTIONAL,
    reasonInfo                    [1]     ReasonInfo OPTIONAL,
    invalidityDate                [2]     GeneralizedTime OPTIONAL,
    revokedcertificateGroup       [3]     RevokedCertificateGroup }

RevokedCertificateGroup ::=       CHOICE {
    serialNumberRange     NumberRange,
    nameSubtree           GeneralName }

expiredCertsOnCRL EXTENSION  ::= {
    SYNTAXExpiredCertsOnCRL
    IDENTIFIED BYid-ce-expiredCertsOnCRL }
```

```
ExpiredCertsOnCRL       ::= GeneralizedTime

baseUpdateTime EXTENSION    ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-baseUpdateTime }


freshestCRL EXTENSION    ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-freshestCRL }


aAissuingDistributionPoint EXTENSION ::= {
    SYNTAX AAIssuingDistPointSyntax
    IDENTIFIED BY id-ce-aAissuingDistributionPoint }

AAIssuingDistPointSyntax ::= SEQUENCE {
    distributionPoint               [ 0 ] DistributionPointName OPTIONAL,
    onlySomeReasons                 [ 1 ] ReasonFlags OPTIONAL,
    indirectCRL                     [ 2 ] BOOLEAN DEFAULT FALSE,
    containsUserAttributeCerts      [ 3 ] BOOLEAN DEFAULT TRUE,
    containsAACerts                 [ 4 ] BOOLEAN DEFAULT TRUE,
    containsSOAPublicKeyCerts       [ 5 ] BOOLEAN DEFAULT TRUE }


inhibitAnyPolicy   EXTENSION   ::= {
    SYNTAX          SkipCerts
IDENTIFIED BY       id-ce-inhibitAnyPolicy }
```

-- правила соответствия PKI --

```
certificateExactMatch MATCHING-RULE ::= {
    SYNTAX          CertificateExactAssertion
    ID              id-mr-certificateExactMatch }

CertificateExactAssertion ::= SEQUENCE {
    serialNumber        CertificateSerialNumber,
    issuer              Name }


certificateMatch MATCHING-RULE ::= {
    SYNTAX          CertificateAssertion
    ID              id-mr-certificateMatch }

CertificateAssertion ::= SEQUENCE {
    serialNumber            [0] CertificateSerialNumber     OPTIONAL,
    issuer                  [1] Name                        OPTIONAL,
    subjectKeyIdentifier    [2] SubjectKeyIdentifier        OPTIONAL,
    authorityKeyIdentifier  [3] AuthorityKeyIdentifier      OPTIONAL,
    certificateValid        [4] Time                        OPTIONAL,
    privateKeyValid         [5] GeneralizedTime             OPTIONAL,
    subjectPublicKeyAlgID   [6] OBJECT IDENTIFIER           OPTIONAL,
    keyUsage                [7] KeyUsage                    OPTIONAL,
    subjectAltName          [8] AltNameType                 OPTIONAL,
    policy                  [9] CertPolicySet               OPTIONAL,
    pathToName              [10] Name                       OPTIONAL,
    subject                 [11]  Name                      OPTIONAL,
    nameConstraints         [12]  NameConstraintsSyntax     OPTIONAL }

AltNameType ::= CHOICE {
    builtinNameForm     ENUMERATED {
            rfc822Name                  (1),
            dNSName                     (2),
            x400Address                 (3),
            directoryName               (4),
            ediPartyName                (5),
            uniformResourceIdentifier   (6),
            iPAddress                   (7),
            registeredId                (8) },
    otherNameForm       OBJECT IDENTIFIER }

CertPolicySet ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId
```

```
certificatePairExactMatch MATCHING-RULE   ::=   {
    SYNTAX          CertificatePairExactAssertion
    ID              id-mr-certificatePairExactMatch }

CertificatePairExactAssertion ::= SEQUENCE {
    issuedToThisCAAssertion         [0] CertificateExactAssertion OPTIONAL,
    issuedByThisCAAssertion         [1] CertificateExactAssertion OPTIONAL }
    ( WITH COMPONENTS               {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS               {..., issuedByThisCAAssertion PRESENT} )


certificatePairMatch MATCHING-RULE ::={
    SYNTAX          CertificatePairAssertion
    ID              id-mr-certificatePairMatch }

CertificatePairAssertion ::= SEQUENCE {
    issuedToThisCAAssertion    [0] CertificateAssertion OPTIONAL,
    issuedByThisCAAssertion    [1] CertificateAssertion OPTIONAL }
    ( WITH COMPONENTS   {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS   {..., issuedByThisCAAssertion PRESENT} )


certificateListExactMatch MATCHING-RULE   ::=   {
    SYNTAX          CertificateListExactAssertion
    ID              id-mr-certificateListExactMatch }

CertificateListExactAssertion ::= SEQUENCE {
    issuer              Name,
    thisUpdate          Time,
    distributionPoint   DistributionPointName OPTIONAL }


certificateListMatch MATCHING-RULE ::= {
    SYNTAX          CertificateListAssertion
    ID              id-mr-certificateListMatch }

CertificateListAssertion ::= SEQUENCE {
    issuer                          Name                OPTIONAL,
    minCRLNumber            [0]     CRLNumber           OPTIONAL,
    maxCRLNumber            [1]     CRLNumber           OPTIONAL,
    reasonFlags                     ReasonFlags         OPTIONAL,
    dateAndTime                     Time                OPTIONAL,
    distributionPoint       [2]     DistributionPointName OPTIONAL,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier OPTIONAL}


algorithmIdentifierMatch MATCHING-RULE    ::=   {
    SYNTAX          AlgorithmIdentifier
    ID              id-mr-algorithmIdentifierMatch }


policyMatch MATCHING-RULE   ::= {
    SYNTAX          PolicyID
    ID              id-mr-policyMatch }


pkiPathMatch MATCHING-RULE  ::= {
SYNTAX      PkiPathMatchSyntax
ID          id-mr-pkiPathMatch }

PkiPathMatchSyntax      ::= SEQUENCE {
firstIssuer     Name,
lastSubject     Name }


enhancedCertificateMatch MATCHING-RULE  ::= {
    SYNTAX  EnhancedCertificateAssertion
    ID      id-mr-enhancedCertificateMatch }

EnhancedCertificateAssertion  ::= SEQUENCE {
    serialNumber            [0]  CertificateSerialNumber    OPTIONAL,
    issuer                  [1]  Name                       OPTIONAL,
```

```
    subjectKeyIdentifier      [2]  SubjectKeyIdentifier       OPTIONAL,
    authorityKeyIdentifier    [3]  AuthorityKeyIdentifier     OPTIONAL,
    certificateValid          [4]  Time                       OPTIONAL,
    privateKeyValid           [5]  GeneralizedTime            OPTIONAL,
    subjectPublicKeyAlgID     [6]  OBJECT IDENTIFIER          OPTIONAL,
    keyUsage                  [7]  KeyUsage                   OPTIONAL,
    subjectAltName            [8]  AltName                    OPTIONAL,
    policy                    [9]  CertPolicySet              OPTIONAL,
    pathToName                [10] GeneralNames               OPTIONAL,
    subject                   [11] Name                       OPTIONAL,
    nameConstraints           [12] NameConstraintsSyntax      OPTIONAL
}

(ALL EXCEPT ({-- отсутствует, т. е. должен присутствовать по меньшей мере один компонент -- }))

AltName  ::= SEQUENCE {
        altnameType         AltNameType,
        altNameValue        GeneralName OPTIONAL }

-- присвоения идентификаторов объектов --

id-ce-subjectDirectoryAttributes        OBJECT IDENTIFIER  ::=  {id-ce 9}
id-ce-subjectKeyIdentifier              OBJECT IDENTIFIER  ::=  {id-ce 14}
id-ce-keyUsage                          OBJECT IDENTIFIER  ::=  {id-ce 15}
id-ce-privateKeyUsagePeriod             OBJECT IDENTIFIER  ::=  {id-ce 16}
id-ce-subjectAltName                    OBJECT IDENTIFIER  ::=  {id-ce 17}
id-ce-issuerAltName                     OBJECT IDENTIFIER  ::=  {id-ce 18}
id-ce-basicConstraints                  OBJECT IDENTIFIER  ::=  {id-ce 19}
id-ce-cRLNumber                         OBJECT IDENTIFIER  ::=  {id-ce 20}
id-ce-reasonCode                        OBJECT IDENTIFIER  ::=  {id-ce 21}
id-ce-instructionCode                   OBJECT IDENTIFIER  ::=  {id-ce 23}
id-ce-invalidityDate                    OBJECT IDENTIFIER  ::=  {id-ce 24}
id-ce-deltaCRLIndicator                 OBJECT IDENTIFIER  ::=  {id-ce 27}
id-ce-issuingDistributionPoint          OBJECT IDENTIFIER  ::=  {id-ce 28}
id-ce-certificateIssuer                 OBJECT IDENTIFIER  ::=  {id-ce 29}
id-ce-nameConstraint                    OBJECT IDENTIFIER  ::=  {id-ce 30 1}

id-ce-cRLDistributionPoints             OBJECT IDENTIFIER  ::=  {id-ce 31}
id-ce-certificatePolicies               OBJECT IDENTIFIER  ::=  {id-ce 32}
id-ce-policyMappings                    OBJECT IDENTIFIER  ::=  {id-ce 33}
-- исключено                            OBJECT IDENTIFIER  ::=  {id-ce 34}
id-ce-authorityKeyIdentifier            OBJECT IDENTIFIER  ::=  {id-ce 35}
id-ce-policyConstraints                 OBJECT IDENTIFIER  ::=  {id-ce 36}
id-ce-extKeyUsage                       OBJECT IDENTIFIER  ::=  {id-ce 37}
id-ce-cRLStreamIdentifier               OBJECT IDENTIFIER  ::=  {id-ce 40}
id-ce-cRLScope                          OBJECT IDENTIFIER  ::=  {id-ce 44}
id-ce-statusReferrals                   OBJECT IDENTIFIER  ::=  {id-ce 45}
id-ce-freshestCRL                       OBJECT IDENTIFIER  ::=  {id-ce 46}
id-ce-orderedList                       OBJECT IDENTIFIER  ::=  {id-ce 47}
id-ce-baseUpdateTime                    OBJECT IDENTIFIER  ::=  {id-ce 51}
id-ce-deltaInfo                         OBJECT IDENTIFIER  ::=  {id-ce 53}
id-ce-inhibitAnyPolicy                  OBJECT IDENTIFIER  ::=  {id-ce 54}
id-ce-toBeRevoked                       OBJECT IDENTIFIER  ::=  {id-ce 58}
id-ce-RevokedGroups                     OBJECT IDENTIFIER  ::=  {id-ce 59}
id-ce-expiredCertsOnCRL                 OBJECT IDENTIFIER  ::=  {id-ce 60}
id-ce-aAissuingDistributionPoint        OBJECT IDENTIFIER  ::=  {id-ce 63}

-- правило соответствия OID --

id-mr-certificateExactMatch             OBJECT IDENTIFIER  ::=  {id-mr 34}
id-mr-certificateMatch                  OBJECT IDENTIFIER  ::=  {id-mr 35}
id-mr-certificatePairExactMatch         OBJECT IDENTIFIER  ::=  {id-mr 36}
id-mr-certificatePairMatch              OBJECT IDENTIFIER  ::=  {id-mr 37}
id-mr-certificateListExactMatch         OBJECT IDENTIFIER  ::=  {id-mr 38}
id-mr-certificateListMatch              OBJECT IDENTIFIER  ::=  {id-mr 39}
id-mr-algorithmIdentifierMatch          OBJECT IDENTIFIER  ::=  {id-mr 40}
id-mr-policyMatch                       OBJECT IDENTIFIER  ::=  {id-mr 60}
```

```
id-mr-pkiPathMatch                OBJECT IDENTIFIER   ::=   {id-mr 62}
id-mr-enhancedCertificateMatch    OBJECT IDENTIFIER   ::=   {id-mr 65}

-- Следующие OBJECT IDENTIFIERS не используются в данной Спецификации:
-- {id-ce 2}, {id-ce 3}, {id-ce 4}, {id-ce 5}, {id-ce 6}, {id-ce 7},
-- {id-ce 8}, {id-ce 10}, {id-ce 11}, {id-ce 12}, {id-ce 13},
-- {id-ce 22}, {id-ce 25}, {id-ce 26}, {id-ce 30}

END

-- A.3      Модуль структуры сертификатов атрибутов

AttributeCertificateDefinitions {joint-iso-itu-t ds(5) module(1) attributeCertificateDefinitions(32) 5}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds, id-oc, certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
        usefulDefinitions(0) 5}


    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute,
        MATCHING-RULE, AttributeType, OBJECT-CLASS, top
        FROM InformationFramework informationFramework

    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, SIGNED{}, InfoSyntax, PolicySyntax, Extensions, Certificate
        FROM AuthenticationFramework authenticationFramework

    DirectoryString{}, TimeSpecification, UniqueIdentifier
        FROM SelectedAttributeTypes selectedAttributeTypes

    GeneralName, GeneralNames, NameConstraintsSyntax, certificateListExactMatch
       FROM CertificateExtensions certificateExtensions

    ub-name
        FROM UpperBounds upperBounds

    UserNotice
        FROM PKIX1Implicit93 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5)
pkix(7) id-mod(0) id-pkix1-implicit-93(4)}

    ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;
-- Если явно не отмечено иное, порядок компонентов структуры SEQUENCE OF не имеет значения в данной Спецификации.

-- структуры сертификатов атрибутов --


AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE
    {
    version                 AttCertVersion, -- версия v2
    holder                  Holder,
    issuer                  AttCertIssuer,
    signature               AlgorithmIdentifier,
    serialNumber            CertificateSerialNumber,
    attrCertValidityPeriod  AttCertValidityPeriod,
    attributes              SEQUENCE OF Attribute,
    issuerUniqueID          UniqueIdentifier    OPTIONAL,
    extensions              Extensions          OPTIONAL
    }


AttCertVersion      ::= INTEGER { v2(1) }
```

```
Holder    ::=   SEQUENCE
          {
          baseCertificateID       [0] IssuerSerial              OPTIONAL,
               -- выдавший орган и порядковый номер сертификата открытого ключа держателя
          entityName              [1] GeneralNames              OPTIONAL,
               -- имя объекта или роли
          objectDigestInfo        [2] ObjectDigestInfo          OPTIONAL
               -- используется для непосредственной аутентификации держателя, например, выполнимый
-- должен присутствовать по меньшей мере один из baseCertificateID, entityName или objectDigestInfo --}


ObjectDigestInfo    ::= SEQUENCE {
    digestedObjectType  ENUMERATED {
        publicKey            (0),
        publicKeyCert        (1),
        otherObjectTypes     (2) },
    otherObjectTypeID       OBJECT IDENTIFIER  OPTIONAL,
    digestAlgorithm         AlgorithmIdentifier,
    objectDigest            BIT STRING }

AttCertIssuer ::=   [0]     SEQUENCE {
    issuerName                    GeneralNames  OPTIONAL,
    baseCertificateID     [0]     IssuerSerial  OPTIONAL,
    objectDigestInfo      [1]     ObjectDigestInfo  OPTIONAL }


-- Должен присутствовать по меньшей мере один компонент
    ( WITH COMPONENTS { ..., issuerName  PRESENT } |
      WITH COMPONENTS { ..., baseCertificateID  PRESENT } |
      WITH COMPONENTS { ..., objectDigestInfo PRESENT } )


IssuerSerial  ::=  SEQUENCE {
    issuer          GeneralNames,
    serial          CertificateSerialNumber,
    issuerUID       UniqueIdentifier OPTIONAL }

AttCertValidityPeriod  ::= SEQUENCE {
    notBeforeTime       GeneralizedTime,
    notAfterTime        GeneralizedTime }

AttributeCertificationPath  ::=  SEQUENCE {
    attributeCertificate          AttributeCertificate,
    acPath                        SEQUENCE OF ACPathData OPTIONAL }

ACPathData  ::=  SEQUENCE {
    certificate           [0]  Certificate  OPTIONAL,
    attributeCertificate  [1]  AttributeCertificate  OPTIONAL }


PrivilegePolicy     ::=   OBJECT IDENTIFIER

-- атрибуты привилегий --


role    ATTRIBUTE::= {
        WITH SYNTAX         RoleSyntax
        ID                  id-at-role }


xmlPrivilegeInfo   ATTRIBUTE  ::= {
    WITH SYNTAX UTF8String --contains XML-encoded privilege information
    ID              id-at-xMLPrivilegeInfo }


RoleSyntax    ::=   SEQUENCE {
    roleAuthority       [0]       GeneralNames OPTIONAL,
    roleName            [1]       GeneralName }

-- классы объектов PMI --
```

```
pmiUser OBJECT-CLASS ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {attributeCertificateAttribute}
    ID               id-oc-pmiUser
    }

pmiAA OBJECT-CLASS ::= {
-- AA PMI
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {aACertificate |
                      attributeCertificateRevocationList |
                      attributeAuthorityRevocationList}
    ID               id-oc-pmiAA
    }

pmiSOA OBJECT-CLASS ::= {  -- источник органа PMI
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {attributeCertificateRevocationList |
                      attributeAuthorityRevocationList |
                      attributeDescriptorCertificate}
    ID               id-oc-pmiSOA
    }

attCertCRLDistributionPt   OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    KIND                 auxiliary
    MAY CONTAIN          { attributeCertificateRevocationList |
                           attributeAuthorityRevocationList }
    ID                   id-oc-attCertCRLDistributionPts
    }

pmiDelegationPath        OBJECT-CLASS   ::= {
    SUBCLASS OF          {top}
    KIND                 auxiliary
    MAY CONTAIN          { delegationPath }
    ID                   id-oc-pmiDelegationPath }

privilegePolicy          OBJECT-CLASS   ::= {
    SUBCLASS OF          {top}
    KIND                 auxiliary
    MAY CONTAIN          {privPolicy }
    ID                   id-oc-privilegePolicy }

protectedPrivilegePolicy OBJECT-CLASS   ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {protPrivPolicy }
    ID               id-oc-protectedPrivilegePolicy }

-- атрибуты справочника PMI --

attributeCertificateAttribute      ATTRIBUTE  ::= {
    WITH SYNTAX                    AttributeCertificate
    EQUALITY MATCHING RULE         attributeCertificateExactMatch
    ID                             id-at-attributeCertificate }

aACertificate                      ATTRIBUTE   ::=  {
    WITH SYNTAX                    AttributeCertificate
    EQUALITY MATCHING RULE         attributeCertificateExactMatch
    ID                             id-at-aACertificate }
```

```
attributeDescriptorCertificate      ATTRIBUTE  ::= {
    WITH SYNTAX                     AttributeCertificate
    EQUALITY MATCHING RULE          attributeCertificateExactMatch
    ID                              id-at-attributeDescriptorCertificate }

attributeCertificateRevocationList  ATTRIBUTE ::= {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-attributeCertificateRevocationList}

attributeAuthorityRevocationList    ATTRIBUTE  ::= {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-attributeAuthorityRevocationList }

delegationPath          ATTRIBUTE  ::= {
    WITH SYNTAX         AttCertPath
    ID                  id-at-delegationPath }

AttCertPath  ::=  SEQUENCE OF AttributeCertificate

privPolicy              ATTRIBUTE  ::= {
    WITH SYNTAX         PolicySyntax
    ID                  id-at-privPolicy }

protPrivPolicy      ATTRIBUTE  ::= {
    WITH SYNTAX                     AttributeCertificate
    EQUALITY MATCHING RULE          attributeCertificateExactMatch
    ID                              id-at-protPrivPolicy }

xmlPrivPolicyATTRIBUTE  ::= {
    WITH SYNTAX UTF8String -- содержит информацию о политике привилегий, закодированную в XML
    ID          id-at-xMLPprotPrivPolicy }

-- расширения сертификатов атрибутов и правила соответствия --

attributeCertificateExactMatch MATCHING-RULE  ::= {
    SYNTAX          AttributeCertificateExactAssertion
    ID              id-mr-attributeCertificateExactMatch }

AttributeCertificateExactAssertion   ::=  SEQUENCE {
    serialNumber        CertificateSerialNumber,
    issuer              AttCertIssuer
    }

attributeCertificateMatch  MATCHING-RULE  ::=  {
    SYNTAX          AttributeCertificateAssertion
    ID              id-mr-attributeCertificateMatch  }

AttributeCertificateAssertion  ::=  SEQUENCE  {
    holder              [0]  CHOICE {
        baseCertificateID    [0]  IssuerSerial,
        holderName           [1]  GeneralNames} OPTIONAL,
    issuer              [1]  GeneralNames OPTIONAL,
    attCertValidity     [2]  GeneralizedTime OPTIONAL,
    attType             [3]  SET OF AttributeType OPTIONAL}
-- Должен присутствовать по меньшей мере один компонент последовательности

holderIssuerMatch MATCHING-RULE  ::= {
    SYNTAX          HolderIssuerAssertion
    ID              id-mr-holderIssuerMatch }

HolderIssuerAssertion ::=  SEQUENCE {
    holder          [0]     Holder    OPTIONAL,
    issuer          [1]     AttCertIssuer OPTIONAL
    }
```

```
delegationPathMatch MATCHING-RULE  ::= {
    SYNTAX          DelMatchSyntax
    ID              id-mr-delegationPathMatch }

DelMatchSyntax   ::=  SEQUENCE {
    firstIssuer     AttCertIssuer,
    lastHolder      Holder }

sOAIdentifier EXTENSION  ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-sOAIdentifier }

sOAIdentifierMatch MATCHING-RULE ::= {
    SYNTAX              NULL
    ID                  id-mr-sOAIdentifierMatch }


authorityAttributeIdentifier EXTENSION  ::=
    {
    SYNTAX              AuthorityAttributeIdentifierSyntax
    IDENTIFIED BY       { id-ce-authorityAttributeIdentifier } }


AuthorityAttributeIdentifierSyntax    ::=  SEQUENCE SIZE (1..MAX) OF AuthAttId


AuthAttId     ::=  IssuerSerial

authAttIdMatch MATCHING-RULE  ::= {
    SYNTAX          AuthorityAttributeIdentifierSyntax
    ID              id-mr-authAttIdMatch }


roleSpecCertIdentifier EXTENSION ::=
    {
    SYNTAX              RoleSpecCertIdentifierSyntax
    IDENTIFIED BY       { id-ce-roleSpecCertIdentifier } }

RoleSpecCertIdentifierSyntax  ::= SEQUENCE SIZE (1..MAX) OF RoleSpecCertIdentifier


RoleSpecCertIdentifier  ::= SEQUENCE {
    roleName                [0]     GeneralName,
    roleCertIssuer          [1]     GeneralName,
    roleCertSerialNumber    [2]     CertificateSerialNumber     OPTIONAL,
    roleCertLocator         [3]     GeneralNames                OPTIONAL }


roleSpecCertIdMatch MATCHING-RULE  ::= {
    SYNTAX          RoleSpecCertIdentifierSyntax
    ID              id-mr-roleSpecCertIdMatch }


basicAttConstraints EXTENSION ::=
    {
        SYNTAX              BasicAttConstraintsSyntax
        IDENTIFIED BY       { id-ce-basicAttConstraints }
    }

BasicAttConstraintsSyntax ::= SEQUENCE
    {
        authority           BOOLEAN DEFAULT FALSE,
        pathLenConstraint   INTEGER (0..MAX) OPTIONAL
    }


basicAttConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          BasicAttConstraintsSyntax
    ID              id-mr-basicAttConstraintsMatch }
```

```
delegatedNameConstraints EXTENSION ::= {
    SYNTAX              NameConstraintsSyntax
    IDENTIFIED BY       id-ce-delegatedNameConstraints }

delegatedNameConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          NameConstraintsSyntax
    ID              id-mr-delegatedNameConstraintsMatch}

timeSpecification EXTENSION  ::=  {
    SYNTAX              TimeSpecification
    IDENTIFIED BY       id-ce-timeSpecification }

timeSpecificationMatch MATCHING-RULE  ::= {
    SYNTAX          TimeSpecification
    ID              id-mr-timeSpecMatch }

acceptableCertPolicies EXTENSION ::= {
    SYNTAX              AcceptableCertPoliciesSyntax
    IDENTIFIED BY       id-ce-acceptableCertPolicies }

AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER

acceptableCertPoliciesMatch MATCHING-RULE  ::= {
    SYNTAX          AcceptableCertPoliciesSyntax
    ID              id-mr-acceptableCertPoliciesMatch }

attributeDescriptor EXTENSION  ::= {
    SYNTAX              AttributeDescriptorSyntax
    IDENTIFIED BY       {id-ce-attributeDescriptor } }

AttributeDescriptorSyntax  ::= SEQUENCE {
    identifier                  AttributeIdentifier,
    attributeSyntax             OCTET STRING (SIZE(1..MAX)),
    name                [0]     AttributeName   OPTIONAL,
    description         [1]     AttributeDescription    OPTIONAL,
    dominationRule              PrivilegePolicyIdentifier}

AttributeIdentifier  ::= ATTRIBUTE.&id({AttributeIDs})

AttributeIDs ATTRIBUTE  ::= {...}

AttributeName  ::= UTF8String(SIZE(1..MAX))

AttributeDescription  ::= UTF8String(SIZE(1..MAX))

PrivilegePolicyIdentifier     ::=   SEQUENCE {
    privilegePolicy             PrivilegePolicy,
    privPolSyntax               InfoSyntax }

attDescriptor MATCHING-RULE  ::= {
    SYNTAX          AttributeDescriptorSyntax
    ID              id-mr-attDescriptorMatch }

userNotice  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF UserNotice
    IDENTIFIED BY       id-ce-userNotice }

targetingInformation  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF Targets
    IDENTIFIED BY       id-ce-targetInformation }
```

```
Targets  ::=  SEQUENCE SIZE (1..MAX) OF Target

Target::=CHOICE {
    targetName          [0]     GeneralName,
    targetGroup         [1]     GeneralName,
    targetCert          [2]     TargetCert }

TargetCert    ::=  SEQUENCE {
    targetCertificate   IssuerSerial,
    targetName          GeneralName OPTIONAL,
    certDigestInfo      ObjectDigestInfo OPTIONAL }

noRevAvail EXTENSION ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-noRevAvail }

acceptablePrivilegePolicies EXTENSION ::= {
    SYNTAX              AcceptablePrivilegePoliciesSyntax
    IDENTIFIED BY       id-ce-acceptablePrivilegePolicies }

AcceptablePrivilegePoliciesSyntax  ::=  SEQUENCE SIZE (1..MAX) OF PrivilegePolicy

indirectIssuer EXTENSION ::= {
    SYNTAX          BOOLEAN
    IDENTIFIED BY       id-ce-indirectIssuer }

indirectIssuerMatch MATCHING-RULE  ::= {
    SYNTAX  BOOLEAN
    ID  id-mr-indirectIssuerMatch }

noAssertion EXTENSION ::= {
    SYNTAX NULL
    IDENTIFIED BY id-ce-noAssertion }

issuedOnBehalfOf EXTENSION ::= {
    SYNTAX GeneralName
    IDENTIFIED BY id-ce-issuedOnBehalfOf }
```

-- присвоения идентификаторов объектов --

-- классы объектов --

```
id-oc-pmiUser                       OBJECT IDENTIFIER ::=   {id-oc 24}
id-oc-pmiAA                         OBJECT IDENTIFIER ::=   {id-oc 25}
id-oc-pmiSOA                        OBJECT IDENTIFIER ::=   {id-oc 26}
id-oc-attCertCRLDistributionPts     OBJECT IDENTIFIER ::=   {id-oc 27}
id-oc-privilegePolicy               OBJECT IDENTIFIER ::=   {id-oc 32}
id-oc-pmiDelegationPath             OBJECT IDENTIFIER ::=   {id-oc 33}
id-oc-protectedPrivilegePolicy      OBJECT IDENTIFIER ::=   {id-oc 34}
```

-- атрибуты справочника --

```
id-at-attributeCertificate              OBJECT IDENTIFIER ::=   {id-at 58}
id-at-attributeCertificateRevocationList OBJECT IDENTIFIER ::=  {id-at 59}
id-at-aACertificate                     OBJECT IDENTIFIER ::=   {id-at 61}
id-at-attributeDescriptorCertificate    OBJECT IDENTIFIER ::=   {id-at 62}
id-at-attributeAuthorityRevocationList  OBJECT IDENTIFIER ::=   {id-at 63}
id-at-privPolicy                        OBJECT IDENTIFIER ::=   {id-at 71}
id-at-role                              OBJECT IDENTIFIER ::=   {id-at 72}
id-at-delegationPath                    OBJECT IDENTIFIER ::=   {id-at 73}
id-at-protPrivPolicy                    OBJECT IDENTIFIER ::=   {id-at 74}
```

```
id-at-xMLPrivilegeInfo                  OBJECT IDENTIFIER ::=   {id-at 75}
id-at-xMLPprotPrivPolicy                OBJECT IDENTIFIER ::=   {id-at 76}

-- расширения сертификатов атрибутов --

id-ce-authorityAttributeIdentifier      OBJECT IDENTIFIER ::=   {id-ce 38}
id-ce-roleSpecCertIdentifier            OBJECT IDENTIFIER ::=   {id-ce 39}
id-ce-basicAttConstraints               OBJECT IDENTIFIER ::=   {id-ce 41}
id-ce-delegatedNameConstraints          OBJECT IDENTIFIER ::=   {id-ce 42}
id-ce-timeSpecification                 OBJECT IDENTIFIER ::=   {id-ce 43}
id-ce-attributeDescriptor               OBJECT IDENTIFIER ::=   {id-ce 48}
id-ce-userNotice                        OBJECT IDENTIFIER ::=   {id-ce 49}
id-ce-sOAIdentifier                     OBJECT IDENTIFIER ::=   {id-ce 50}
id-ce-acceptableCertPolicies            OBJECT IDENTIFIER ::=   {id-ce 52}
id-ce-targetInformation                 OBJECT IDENTIFIER ::=   {id-ce 55}
id-ce-noRevAvail                        OBJECT IDENTIFIER ::=   {id-ce 56}
id-ce-acceptablePrivilegePolicies       OBJECT IDENTIFIER ::=   {id-ce 57}
id-ce-indirectIssuer                    OBJECT IDENTIFIER ::=   {id-ce 61}
id-ce-noAssertion                       OBJECT IDENTIFIER ::=   {id-ce 62}
id-ce-issuedOnBehalfOf                  OBJECT IDENTIFIER ::=   {id-ce 64}

-- Правила соответствия PMI --

id-mr-attributeCertificateMatch         OBJECT IDENTIFIER ::=   {id-mr 42}
id-mr-attributeCertificateExactMatch    OBJECT IDENTIFIER ::=   {id-mr 45}
id-mr-holderIssuerMatch                 OBJECT IDENTIFIER ::=   {id-mr 46}
id-mr-authAttIdMatch                    OBJECT IDENTIFIER ::=   {id-mr 53}
id-mr-roleSpecCertIdMatch               OBJECT IDENTIFIER ::=   {id-mr 54}
id-mr-basicAttConstraintsMatch          OBJECT IDENTIFIER ::=   {id-mr 55}
id-mr-delegatedNameConstraintsMatch     OBJECT IDENTIFIER ::=   {id-mr 56}
id-mr-timeSpecMatch                     OBJECT IDENTIFIER ::=   {id-mr 57}
id-mr-attDescriptorMatch                OBJECT IDENTIFIER ::=   {id-mr 58}
id-mr-acceptableCertPoliciesMatch       OBJECT IDENTIFIER ::=   {id-mr 59}
id-mr-delegationPathMatch               OBJECT IDENTIFIER ::=   {id-mr 61}
id-mr-sOAIdentifierMatch                OBJECT IDENTIFIER ::=   {id-mr 66}
id-mr-indirectIssuerMatch               OBJECT IDENTIFIER ::=   {id-mr 67}

END
```

Приложение B

Генерация CRL и правила обработки

(Данное Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

B.1 Введение

Зависимая сторона (пользователь сертификата) должен иметь возможность проверять статус аннулирования сертификата, чтобы определить, доверять ли данному сертификату. Списки аннулированных сертификатов (CRL) являются одним из механизмов получения зависимыми сторонами информации об аннулировании. Другие механизмы также могут использоваться, но находятся вне области применения данной Спецификации.

В данном Приложении рассматривается использование CRL для проверки зависимыми сторонами статуса аннулирования сертификата. Различные органы могут иметь разные политики в отношении выдачи списков аннулирования. Например, в некоторых случаях орган, выдающий сертификаты, может авторизовать другой орган для выдачи списков аннулированных сертификатов для сертификатов, которые он выдает. Некоторые органы могут сочетать аннулирование сертификатов оконечных объектов и CA-сертификатов в один список, в то время как другие органы могут разделять их на отдельные списки. Некоторые органы могут разбивать совокупность сертификатов на фрагменты CRL, и некоторые органы могут выдавать дельта обновления для списков аннулирования между выдачами регулярных CRL. В результате зависимые стороны должны иметь возможность определить границы CRL, который они ищут, для того чтобы обеспечить наличие у них полного набора информации об аннулировании, охватывающего границы рассматриваемого сертификата для рассматриваемых причин аннулирования, задавая политику, согласно которой они работают. В данном Приложении предоставляется механизм для определения зависимыми сторонами границ искомых CRL.

Данное Приложение разработано для проверки статуса аннулирования сертификатов открытых ключей с использованием CRL, полных и целых CRL оконечных объектов (EPRL), а также списков аннулирования органов по сертификации (CARL). Тем не менее, данное описание также может применяться проверке статуса аннулирования сертификатов атрибутов с использованием списков аннулированных сертификатов атрибутов (ACRL) и списков аннулирования органов атрибутов (AARL). Для целей данного Приложения, ACRL может рассматриваться на месте CRL, EPRL может быть полным и целым ACRL оконечного объекта, а AARL на месте CARL. Аналогично, атрибуты справочника, определенные в п. B.4, должны отображаться в атрибуты для AARL и ACRL, и поля, определяющие типа сертификатов в расширении выдающая точка распределения, могут отображаться в поля, применимые к PMI.

B.1.1 Типы CRL

Для зависимых сторон могут быть доступны CRL одного или нескольких следующих типов, на основе аспектов аннулирования политики органа, выдающего сертификаты:

- полный и целый CRL;
- полный и целый CRL оконечного объекта (EPRL);
- полный и целый список аннулирования органов по сертификации (CARL);
- CRL, EPRL или CARL точки распределения;
- непрямой CRL, EPRL или CARL (ICRL);
- дельта CRL, EPRL или CARL;
- непрямой dCRL, EPRL или CARL.

Полный и целый CRL представляет собой список всех аннулированных сертификатов оконечных объектов и CA-сертификатов, выданный органом по какой-либо причине и по всем причинам.

Полный и целый EPRL представляет собой список всех аннулированных сертификатов оконечных объектов, выданный органом по какой-либо причине и по всем причинам.

Полный и целый CARL представляет собой список CA-сертификатов, выданный органом по какой-либо причине и по всем причинам.

CRL, EPRL или CARL точки распределения охватывает все или подмножество сертификатов, выданных органом. Подмножество может быть основано на различных критериях.

Непрямой CRL, EPRL или CARL (ICRL) представляет собой CRL, содержащий список аннулированных сертификатов, в котором некоторые или все сертификаты не были выданы органом, подписывающим и выдающим CRL.

Дельта CRL, EPRL или CARL представляет собой CRL, содержащий только изменения в CRL, являющийся полным для заданных границ на момент CRL, адресуемого в dCRL. Отметим, что адресуемый CRL может быть полным для заданных границ, а также dCRL, используемым для локального создания CRL, полного для заданных границ.

Все вышеперечисленные типы CRL (исключая dCRL) являются типами CRL, полными для заданных границ. dCRL должен использоваться в сочетании с соответствующим CRL, полным для тех же границ для формирования полной картины статуса аннулирования сертификатов.

Непрямой dCRL, EPRL или CARL представляет собой CRL, содержащий только изменения в множестве из одного или нескольких CRL, полных для заданных границ, в которых некоторые или все сертификаты могут не быть выданы органом, подписывающим и выдающим данный CRL.

В пределах данного Приложения, а также данной Спецификации, "границы CRL" определяются двумя независимыми измерениями. Одним из измерений является множество сертификатов, охватываемых CRL. Другим измерением является множество кодов причин, охватываемых CRL. Границы CRL могут определяться одним или несколькими следующими способами:

– расширением выдающая точка распределения (IDP) в CRL; или

– другими способами, вне области применения данной Спецификации.

B.1.2 Обработка CRL

Если зависимая сторона использует CRL в качестве механизма определения, является ли сертификат аннулированным, она должна использовать подходящий CRL для данного сертификата. В данном Приложении описана процедура получения и обработки соответствующего CRL путем прохождения нескольких определенных шагов. Реализация, функционально эквивалентная внешнему поведению, вызываемому данной процедурой, также должна считаться совместимой с данным приложением и соответствующей Спецификацией. Алгоритм, использованный определенной реализацией для получения правильного результата (т. е. статуса аннулирования сертификата) из заданных входных данных (самого сертификата и данных из локальной политики), не стандартизируется. Например, несмотря на то, что данная процедура описана как последовательность шагов, которые должны обрабатываться по порядку, реализация может использовать CRL, находящиеся в ее локальном кэше, а не искать CRL каждый раз при обработке сертификата, если данные CRL являются полными для границ сертификата и не нарушают параметров сертификата или политики.

Следующие общие шаги описаны ниже с п. B.2 по п. B.5:

1) определить параметры для CRL;
2) определить требуемые CRL;
3) получить CRL;
4) обработать CRL.

Шаг 1) определяет параметры из сертификата и других мест, которые будут использоваться при определении требуемых типов CRL.

Шаг 2) применяет значения параметров к выполнению определения.

Шаг 3) определяет атрибуты справочника, из которых можно восстановить типы CRL.

Шаг 4) описывает обработку соответствующих CRL.

B.2 Определить параметры для CRL

Информация, расположенная в самом сертификате, а также информация из политики, согласно которой действует зависимая сторона, предоставляет параметры для определения пригодности CRL соискателя. Для определения подходящих типов CRL требуется следующая информация:

– тип сертификата (т. е. оконечного объекта или CA);

– критическая точка распределения CRL;

– критический наиболее новый CRL;

– рассматриваемые коды причин.

Тип сертификата может определяться из расширения основных ограничений в сертификате. Если расширение присутствует, оно указывает, является ли сертификат CA-сертификатом или сертификатом оконечного объекта. Если расширение отсутствует, то типом сертификата считается сертификат оконечного объекта. Данная информация требуется для определения того, может ли CRL, EPRL или CARL использоваться для проверки сертификата на аннулирование.

Если сертификат содержит критическое расширение точки распределения CRL, то система обработки сертификатов зависимой стороны должна понять это расширение и получить и использовать CRL, на который указывает расширение точки распределения CRL для рассматриваемых кодов причин, чтобы определить статус аннулирования сертификата. Доверия полному CRL, например, было бы недостаточно.

Если сертификат содержит критическое расширение наиболее нового CRL, то зависимая сторона не может использовать сертификат без предварительного поиска и проверки наиболее нового CRL.

Рассматриваемые коды причин определяются политикой и, как правило, предоставляются приложением. Рекомендуется, чтобы они включали все коды причин. Данная информация требуется для определения CRL, являющихся достаточными в условиях кодов причин.

Отметим, что политика может также определять, должна ли зависимая сторона проверять на статус аннулирования dCRL, когда расширение **freshestCRL** помечено как некритическое или отсутствует в сертификате. Будучи исключенной из данного шага, обработка данных дополнительных dCRL описана в шаге 4).

B.3 Определить требуемые CRL

Значения параметров, описанных в п. B.2, определяют критерии, на основании которых определяются типы CRL, требуемые для проверки статуса аннулирования заданного сертификата. Определение типов CRL может быть выполнено на основе следующих наборов критериев, как описано ниже в пп. B.3.1–B.3.4.

- сертификат оконечного объекта с заявленной критической DP CRL;
- сертификат оконечного объекта без заявленной критической DP CRL;
- CA-сертификат с заявленной критической DP CRL;
- CA-сертификат без заявленной критической DP CRL.

Управление оставшимися параметрами (критическое расширение наиболее нового CRL и множество рассматриваемых кодов причин) производится в каждом из подпунктов.

Отметим, что в каждом случае несколько типов CRL могут удовлетворять требованиям. Если существует выбор между типами CRL, зависимая сторона может выбрать для использования любой из соответствующих типов.

B.3.1 Оконечный объект с критической DP CRL

Если сертификат является сертификатом оконечного объекта и в сертификате присутствует расширение **cRLDistributionPoints**, помеченное как критическое, должны быть получены следующие CRL:

- CRL от одной из названных CRL точек распределения, охватывающий один или несколько рассматриваемых кодов причин;
- если все рассматриваемые коды причин не охватываются данным CRL, то статус аннулирования по оставшимся кодам причин может быть выполнен любым сочетанием следующих CRL:
  - дополнительными CRL точки распределения;
  - дополнительными полными CRL;
  - дополнительными полными EPRL.

Если расширение наиболее нового CRL также присутствует в сертификате и помечено как критическое, то должны быть также получены один или несколько CRL из одной или нескольких названных в данном расширении точек распределения, обеспечивая проверку наиболее новой информации об аннулировании для всех рассматриваемых кодов причин.

B.3.2 Оконечный объект без критической DP CRL

Если сертификат является сертификатом оконечного объекта и расширение **cRLDistributionPoints** отсутствует в сертификате или присутствует, но помечено как некритическое, то статус аннулирования по рассматриваемым кодам причин может быть выполнен любым сочетанием следующих CRL:

- CRL точек распределения (при наличии);
- полными CRL;
- полными EPRL.

Если расширение наиболее нового CRL также присутствует в сертификате и помечено как критическое, то должны быть также получены один или несколько CRL из одной или нескольких названных в данном расширении точек распределения, обеспечивая проверку наиболее новой информации об аннулировании для всех рассматриваемых кодов причин.

B.3.3 CA с критической DP CRL

Если сертификат является CA-сертификатом и в сертификате присутствует расширение **cRLDistributionPoints**, помеченное как критическое, должны быть получены следующие CRL/CARL:

- CRL или CARL от одной из названных точек распределения, охватывающий один или несколько рассматриваемых кодов причин;
- если все рассматриваемые коды причин не охватываются данным CRL/CARL, то статус аннулирования по оставшимся кодам причин может быть выполнен любым сочетанием следующих CRL/CARL:
  - дополнительными CRL/CARL точки распределения;
  - дополнительными полными CRL;
  - дополнительными полными CARL.

Если расширение наиболее нового CRL также присутствует в сертификате и помечено как критическое, то должны быть также получены один или несколько CRL/CARL из одной или нескольких названных в данном расширении точек распределения, обеспечивая проверку наиболее новой информации об аннулировании для всех рассматриваемых кодов причин.

B.3.4 CA без критической DP CRL

Если сертификат является CA-сертификатом и расширение **cRLDistributionPoints** отсутствует в сертификате или присутствует, но помечено как некритическое, то статус аннулирования по рассматриваемым кодам причин может быть выполнен любым сочетанием следующих CRL:

- CRL/CARL точек распределения (при наличии);
- полными CRL;
- полными CARL.

Если расширение наиболее нового CRL также присутствует в сертификате и помечено как критическое, то должны быть также получены один или несколько CRL/CARL из одной или нескольких названных в данном расширении точек распределения, обеспечивая проверку наиболее новой информации об аннулировании для всех рассматриваемых кодов причин.

B.4 Получить CRL

Если зависимая сторона восстанавливает соответствующий CRL из Справочника, то данные CRL получаются из DP CRL или записи справочника органа, выдающего сертификаты, путем восстановления соответствующих атрибутов, т. к. одного или нескольких из следующих атрибутов:

- список аннулированных сертификатов;
- список аннулирования органов;
- дельта список аннулирования.

B.5 Обработать CRL

После рассмотрения параметров, описанных в п. B.2, определения соответствующих типов CRL, как описано в п. B.3, и восстановления соответствующего набора CRL, как описано в п. B.4, зависимая сторона готова обработать CRL. Множество CRL будет содержать по меньшей мере один основной CRL и может также содержать один или несколько dCRL. Для каждого обрабатываемого CRL зависимая сторона должна обеспечивать, что CRL является правильным относительно своих границ. Зависимая сторона уже определила, что CRL является подходящим для границ рассматриваемого сертификата, путем выполнения B.2 и B.3, описанных выше. Более того, проверки подлинности должны выполняться на CRL и должны проверяться для определения того, что сертификат не был аннулирован. Данные проверки описаны ниже в пп. B.5.1–B.5.4.

B.5.1 Проверить подлинность границ основного CRL

Как описано в п. B.3, может быть несколько типов CRL, которые могут быть использованы как основной CRL для проверки статуса аннулирования сертификата. В зависимости от политики выдающего органа в отношении выдачи CRL, для зависимой стороны может быть доступен один или несколько из следующих основных типов CRL:

- полный CRL для всех объектов;
- полный EPRL;
- полный CARL;
- основанный на точке распределения CRL/EPRL/CARL.

В подпунктах B.5.1.1–B.5.1.4 предоставлен набор условий, которые должны выполняться, чтобы зависимая сторона использовала CRL каждого типа как основной CRL для проверки статуса аннулирования сертификата для рассматриваемых кодов причин.

Непрямые основные CRL рассмотрены в каждом из подпунктов.

B.5.1.1 Полный CRL

Чтобы определить, что CRL является полным CRL для сертификатов оконечных объектов и CA-сертификатов, за которые выдающий орган несет ответственность, для всех рассматриваемых кодов причин, должно выполняться следующее:

- должно отсутствовать расширение индикатора дельта CRL; и
- может присутствовать расширение выдающей точки распределения; и
- либо расширение выдающей точки распределения не должно содержать поля точки распределения, либо одно из имен в поле точки распределения должно соответствовать полю **issuer** в CRL; и

- расширение выдающей точки распределения должно либо не содержать ни одного из следующих полей, либо, если содержит какое-либо из следующих полей, ни одно из присутствующих полей не должно быть установлено в TRUE: containsUserPublicKeyCerts, containsCACerts, containsUserAttributeCerts, containsAACerts, и/или containsSOAPublicKeyCerts; и
- если в расширении выдающей точки распределения присутствует поле **reasonCodes**, поле кода причин должно включать все рассматриваемые причины к приложению; и
- расширение выдающей точки распределения может содержать, а может не содержать поле **indirectCRL** (значит, данное поле не должно проверяться).

B.5.1.2 Полный EPRL

Чтобы определить, что CRL является полным EPRL для рассматриваемых кодов причин, должны выполняться все следующие условия:

- должно отсутствовать расширение индикатора дельта CRL; и
- должно присутствовать расширение выдающей точки распределения; и
- либо расширение выдающей точки распределения не должно содержать поля точки распределения, либо одно из имен в поле точки распределения должно соответствовать полю **issuer** в CRL; и
- расширение выдающей точки распределения должно содержать поле **containsUserPublicKeyCerts**. Данное поле должно быть установлено в **TRUE**; и
- если в расширении выдающей точки распределения присутствует поле **reasonCodes**, поле кода причин должно включать все рассматриваемые причины к приложению; и
- расширение выдающей точки распределения может содержать, а может не содержать поле **indirectCRL** (значит, данное поле не должно проверяться); и

Данный CRL может использоваться, только если зависимая сторона определила, что сертификатом субъекта является сертификат оконечного объекта. Таким образом, если сертификат субъекта содержит расширение **basicConstraints**, его значение должно быть **cA=FALSE**.

B.5.1.3 Полный CARL

Чтобы определить, что CRL является полным CARL для рассматриваемых кодов причин, должны выполняться все следующие условия:

- должно отсутствовать расширение индикатора дельта CRL; и
- должно присутствовать распределение выдающей точки; и
- либо расширение выдающей точки распределения не должно содержать поля точки распределения, либо одно из имен в поле точки распределения должно соответствовать полю **issuer** в CRL; и
- выдающая точка распределения должна содержать поле **containsCACerts**. Данное поле должно быть установлено в **TRUE**; и
- если в расширении выдающей точки распределения присутствует поле **reasonCodes**, поле кода причин должно включать все рассматриваемые причины к приложению; и
- выдающая точка распределения может содержать, а может не содержать поле **indirectCRL** (значит, данное поле не должно проверяться); и

Данный CARL может использоваться, только если сертификатом субъекта является CA-сертификат. Таким образом, сертификат субъекта должен содержать расширение **basicConstraints** со значением **cA=TRUE**.

B.5.1.4 Основанный на точке распределения CRL/EPRL/CARL

Чтобы определить, что CRL является одним из CRL, указанных в расширении точки распределения CRL или расширении Наиболее новый CRL в сертификате, должны выполняться все следующие условия:

- либо поле точки распределения в расширении выдающей точки распределения CRL должно отсутствовать (только когда не рассматривается критическая DP CRL), либо одно из имен в поле точки распределения в расширении точки распределения CRL или расширении наиболее нового CRL должно соответствовать одному из имен в поле тоски распределения в расширении выдающей точки распределения в CRL. Альтернативно, одно из имен в поле **cRLIssuer** DP CRL сертификата или расширении наиболее нового CRL может соответствовать одному из имен в DP IDP; и
- расширение выдающей точки распределения должно либо не содержать ни одного из следующих полей, либо, если содержит какое-либо из следующих полей, ни одно из присутствующих полей не должно быть установлено в TRUE: containsUserPublicKeyCerts, containsCACerts, containsUserAttributeCerts, containsAACerts, и/или containsSOAPublicKeyCerts, или поле, подходящее для типа сертификата должно быть установлено в TRUE (тип поля для каждого типа сертификата см. в таблице B.1); и

– если в расширении точки распределения CRL или в расширении наиболее нового CRL сертификата присутствует поле кода причин, данное поле должно либо отсутствовать в расширении выдающей точки распределения CRL, либо содержать по меньшей мере один из кодов причин, заявленных в расширении точки распределения CRL сертификата; и

– если поле **cRLIssuer** отсутствует в расширении точки распределения CRL сертификата CRL должен быть подписан тем же CA, который подписал сертификат; и

– если поле **cRLIssuer** присутствует в относительном расширении (расширении точки распределения CRL или наиболее нового CRL) сертификата, CRL должен быть подписан органом, выдающим CRL, определенным в расширении точки распределения CRL или расширении наиболее нового CRL сертификата и CRL должен содержать поле **indirectCRL** в расширении выдающей точки распределения.

ПРИМЕЧАНИЕ. – При проверке наличия причин и поля cRLIssuer, проверка будет успешной, только если поле присутствует в том же расширении DistributionPoint DP CRL или наиболее нового CRL, для которого существует соответствие имен в поле точки распределения расширения IDP в соответствующем CRL.

Таблица B.1 – Тип сертификата и поле выдающей точки распределения

Тип сертификата	Поле выдающей точки распределения
Оконечный объект (открытый ключ)	containsUserPublicKeyCerts
CA	containsCACerts
Оконечный объект (атрибут)	containsUserAttributeCerts
AA	containsAACerts
SOA	containsSOAPublicKeyCerts

B.5.2 Проверить подлинность границ дельта CRL

Зависимая стороне может также проверять dCRL, либо по причине того, что этого требует критическое расширение в сертификате или CRL, либо если политика, согласно которой действует зависимая сторона, требует проверки dCRL.

Зависимая сторона может всегда быть уверена, что имеет соответствующую информацию о CRL для сертификата, если выполняются все следующие условия:

– основной CRL, используемый зависимой стороной, является подходящим для сертификата (в отношении границ); и

– дельта CRL, используемый зависимой стороной, является подходящим для сертификата (в отношении границ); и

– основной CRL был выдан одновременно или позже основного CRL, к которому обращается dCRL.

Чтобы определить, что dCRL является подходящим для сертификата, должны выполняться все следующие условия:

– должно присутствовать расширение индикатора дельта CRL; и

– dCRL должен быть выдан после основного CRL. Одним из способов обеспечения этого является проверка, что номер CRL в расширении **crlNumber** dCRL превышает номер CRL в расширении **crlNumber** основного CRL, используемого зависимой стороной, и поля **cRLStreamIdentifier** в основном CRL и совпадают. Данный подход может потребовать дополнительную логику для обертывания чисел. Другим способом является сравнение полей **thisUpdate** в основном CRL и dCRL зависимой стороны; и

– основной CRL, используемый зависимой стороной, должен быть тем, для которого выдан dCRL, или более поздним. Одним из способов обеспечения этого является проверка, что номер CRL в расширении **deltaCRLIndicator** dCRL меньше или равен номеру CRL в расширении **crlNumber** основного CRL, используемого зависимой стороной, и поля **cRLStreamIdentifier** в основном CRL и совпадают. Данный подход может потребовать дополнительную логику для обертывания чисел. Другим способом является сравнение поля **thisUpdate** в основном CRL зависимой стороны и dCRL и расширения **baseUpdateTime** в dCRL зависимой стороны; и

ПРИМЕЧАНИЕ. – Зависимая сторона может всегда создать основной CRL путем применения dCRL в основному CRL, если оба указанных выше правила выполняются при использовании проверок **crlNumber** и **cRLStreamIdentifier**. В таком случае, расширение **crlNumber** нового основного CRL и поле **thisUpdate** берутся из dCRL. Зависимая сторона не знает поля **nextUpdate** нового основного CRL и не должна его знать для целей связывания его с другим dCRL.

– если dCRL содержит расширение выдающей точки распределения, то границы выдающей точки распределения должны быть совместимы с сертификатом, как описано выше в п. B.5.1.4; и

– если dCRL не содержит ни одного из следующих расширений: **streamIdentifier** и **issuingDistributionPoint**, он должен использоваться только совместно с полным и целым основным CRL.

B.5.3 Проверка подлинности и текущие проверки по основному CRL

Чтобы проверить, что основной CRL является правильным и не был изменен с момента выдачи, должны выполняться все следующие условия:

- зависимая сторона должна иметь возможность получить открытый ключ выдавшего органа, определенного в CRL, используя средства аутентификации; и
- подпись на основном CRL должна быть проверена с использованием данного аутентифицированного открытого ключа; и
- если присутствует поле **nextUpdate**, текущее время должно быть раньше, чем поле **nextUpdate**; и
- имя выдавшего органа в CRL должно соответствовать имени выдавшего органа в сертификате, проверяемом на аннулирование, если CRL не был получен из DP CRL в сертификате и расширение DP CRL не содержит компонента выдавшего органа CRL. В таком случае, одно из имен в компоненте выдавшего органа CRL в расширении DP CRL должно соответствовать имени выдавшего органа в CRL.

B.5.4 Проверка подлинности и проверки по дельта CRL

Чтобы проверить, что dCRL является правильным и не был изменен с момента выдачи, должны выполняться все следующие условия:

- зависимая сторона должна иметь возможность получить открытый ключ выдавшего органа, определенного в CRL, используя средства аутентификации; и
- подпись на dCRL должна быть проверена с использованием данного аутентифицированного открытого ключа; и
- если присутствует поле **nextUpdate**, текущее время должно быть раньше, чем поле **nextUpdate**; и
- имя выдавшего органа в dCRL должно соответствовать имени выдавшего органа в сертификате, проверяемом на аннулирование, если дельта CRL не был получен из DP CRL в сертификате и расширение DP CRL не содержит компонента выдавшего органа CRL. В таком случае, одно из имен в компоненте выдавшего органа CRL в расширении DP CRL должно соответствовать имени выдавшего органа в CRL.

Приложение C

Примеры выдачи CRL

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Для выдачи CRL существует две модели, включающие использование dCRL для заданного набора сертификатов.

В первой модели каждый dCRL обращается к самому последнему CRL, являющемуся полным для заданных границ. Несколько dCRL могут быть выданы для тех же границ до выдачи нового CRL, являющегося полным для заданных границ. Новый CRL, являющийся полным для заданных границ, используется в качестве основы для следующей последовательности dCRL и является CRL, к которому обращается соответствующее расширение в dCRL. Когда выдается новый CRL, являющийся полным для заданных границ, также выдается окончательный dCRL для предыдущего CRL, являющегося полным для заданных границ.

Вторая модель, несмотря на то, что очень похожа на первую, отличается тем, что CRL, к которому обращается dCRL, не обязательно является полным для заданных границ (т. е. адресуемый CRLможет быть выдан как только dCRL). Если адресуемый CRL является полным для заданных границ, он не обязательно должен быть самым последним, полным для данных границ.

Система, использующая сертификаты, которая обрабатывает dCRL, также должна иметь CRL, являющийся полным для заданных границ и по меньшей мере таким же современным, как и CRL, к которому обращается dCRL. Данный CRL, являющийся полным для заданных границ, может быть как выдан ответственным органом, так и создан локально системой, использующей сертификаты. Отметим, что в некоторых ситуациях может происходить дублирование информации в dCRL и CRL, являющемся полным для заданных границ, если, например, система, использующая сертификаты, имеет CRL, выданный после CRL, к которому обращается dCRL.

В следующей таблице приведено три примера использования dCRL. Пример 1 представляет собой традиционную схему, описанную выше как первая модель. Примеры 2 и 3 представляют собой варианты описанной выше второй модели.

В примере 2, орган выдает CRL, являющиеся полными для заданных границ, через день, а dCRL обращаются к предпоследнему, полному для границ CRL. Данная схема может быть полезной в средах, где существует необходимость в уменьшении числа пользователей, получающих доступ к хранилищу в одно и то же время для поиска CRL, являющегося полным для заданных границ. В примере 2, пользователи, имеющие самый последний CRL, являющийся полным для заданных границ, а также пользователи, имеющие предпоследний полный для границ CRL, могут использовать один и тот же dCRL. Обе совокупности пользователей имеют полную информацию об аннулировании сертификатов для данных заданных границ на момент выдачи используемого dCRL.

В примере 3, CRL, являющиеся полными для заданных границ, выдаются раз в неделю, как в примере 1, но каждый dCRL обращается к базе информации об аннулировании за 7 дней до данного dCRL.

Пример использования непрямых CRL здесь не предоставляется, но является расширенным множеством данных примеров.

Это только примеры, и также возможны другие варианты, в зависимости от локальной политики. К некоторым факторам, которые могут приниматься во внимание при создании такой политики, относятся: количество пользователей и частота доступа к CRL, тиражирование CRL, балансировка нагрузки на системы справочника, содержащие CRL, рабочие характеристики, требования задержки и т. д.

День	Пример 1 – Дельта обращается к самому последнему CRL, являющемуся полным для заданных границ		Пример 2 – Дельта обращается к предпоследнему CRL, являющемуся полным для заданных границ		Пример 3 – Дельта обращается к информации об аннулировании, выданной 7 дней назад	
	CRL, полный для заданных границ	**Дельта-CRL**	**CRL, полный для заданных границ**	**Дельта-CRL**	**CRL, полный для заданных границ**	**Дельта-CRL**
8	thisUpdate=день 8 nextUpdate=день 15 crlNumber=8	thisUpdate=день 8 nextUpdate=день 9 crlNumber=8 BaseCRLNumber=1	thisUpdate=день 8 nextUpdate=день 10 crlNumber=8	thisUpdate=день 8 nextUpdate=день 9 crlNumber=8 BaseCRLNumber=6	thisUpdate=день 8 nextUpdate=день 15 cRLNumber=8	thisUpdate=день 8 nextUpdate=день 9 cRLNumber=8 BaseCRLNumber=1
9	не выдан	thisUpdate=день 9 nextUpdate=день 10 crlNumber=9 BaseCRLNumber=8	не выдан	thisUpdate=день 9 nextUpdate=день 10 crlNumber=9 BaseCRLNumber=6	не выдан	thisUpdate=день 9 nextUpdate=день 10 cRLNumber=9 BaseCRLNumber=2
10	не выдан	thisUpdate=день 10 nextUpdate=день 11 crlNumber=10 BaseCRLNumber=8	thisUpdate=день 10 nextUpdate=день 12 crlNumber=10	thisUpdate=день 10 nextUpdate=день 11 crlNumber=10 BaseCRLNumber=8	не выдан	thisUpdate=день 10 nextUpdate=день 11 cRLNumber=10 BaseCRLNumber=3
11–14	Диаграммы продолжаются как для предыдущего дня					
15	thisUpdate=день 15 nextUpdate=день 22 crlNumber=15	thisUpdate=день 15 nextUpdate=день 16 crlNumber=15 BaseCRLNumber=8	не выдан	thisUpdate=день 15 nextUpdate=день 16 crlNumber=15 BaseCRLNumber=12	thisUpdate=день 15 nextUpdate=день 22 cRLNumber=15	thisUpdate=день 15 nextUpdate=день 16 cRLNumber=15 BaseCRLNumber=8
16	не выдан	thisUpdate=день 16 nextUpdate=день 17 crlNumber=16 BaseCRLNumber=15	thisUpdate=день 16 nextUpdate=день 18 crlNumber=16	thisUpdate=день 16 nextUpdate=день 17 crlNumber=16 BaseCRLNumber=14	не выдан	thisUpdate=день 16 nextUpdate=день 17 cRLNumber=16 BaseCRLNumber=9

Приложение D

Примеры политик привилегий и определения атрибутов привилегий

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

D.1 Введение

Политика привилегий точно определяет, для управления привилегиями, когда верификатор привилегий должен заключить, что представленный набор привилегий является достаточным для предоставления доступа (к запрашиваемому объекту, ресурсу, приложению и т. д.) заявителю привилегий. Формальная спецификация политики привилегий может помочь верификатору привилегий в автоматической оценке привилегий заявителя в соответствии с чувствительностью запрашиваемого ресурса, так как она включает правила для определения прохождения/неудачи запроса заявителя привилегий при задании привилегии и чувствительности ресурса.

В связи с тем, что существуют требования к обеспечению целостности политики привилегий, используемой в данных определениях, идентификатор политики привилегий в форме идентификатора объекта, а также HASH полной политики привилегий могут передаваться в подписанных объектах, храниться в записях справочника и т. д. Тем не менее, в данной Спецификации не стандартизируется определенный синтаксис, который должен использоваться для определения экземпляра политики привилегий.

D.2 Выборочные синтаксисы

Политика привилегий может быть определена с использование любого синтаксиса, включая простой текст. Чтобы помочь определяющим политики привилегий в понимании различных возможностей для определений, в данном Приложении приводятся два примерных синтаксиса, которые могут использоваться для данной цели. Необходимо подчеркнуть, что это только примеры, и реализация управления привилегиями посредством использования сертификатов атрибутов или расширения **subjectDirectoryAttributes** сертификатов открытых ключей НЕ должны обязательно поддерживать данные или любые другие определенные синтаксисы.

D.2.1 Первый пример

Следующий синтаксис ASN.1 представляет собой пример комплексного и гибкого инструмента для определения политики привилегий.

```
PrivilegePolicySyntax       ::= SEQUENCE {
    version         Version,
    ppe             PrivPolicyExpression }

PrivPolicyExpression        ::= CHOICE {
    ppPredicate     [0] PrivPolicyPredicate,
    and             [1] SET SIZE (2..MAX) OF PrivPolicyExpression,
    or              [2] SET SIZE (2..MAX) OF PrivPolicyExpression,
    not             [3] PrivPolicyExpression,
    orderedPPE      [4] SEQUENCE OF PrivPolicyExpression }
    -- Отметим: "sequence" определяет временной порядок, в котором должна
    -- рассматриваться привилегия

PrivPolicyPredicate         ::= CHOICE {
    present                 [0] PrivilegeIdentifier,
    equality                [1] PrivilegeComparison, -- однозначное/имеющее значением множество priv.
    greaterOrEqual          [2] PrivilegeComparison, -- однозначное priv.
    lessOrEqual             [3] PrivilegeComparison, -- однозначное priv.
    subordinate             [4] PrivilegeComparison, -- однозначное priv.
    substrings              [5] SEQUENCE {      -- однозначное priv.
        type                        PrivilegeType,
        initial                 [0] PrivilegeValue OPTIONAL,
        any                     [1] SEQUENCE OF PrivilegeValue,
        final                   [2] PrivilegeValue OPTIONAL },
    subsetOf                [6] PrivilegeComparison, -- имеющее значением множество priv.
    supersetOf              [7] PrivilegeComparison, -- имеющее значением множество priv.
    nonNullSetInter         [8] PrivilegeComparison, -- имеющее значением множество priv.
    approxMatch             [9] PrivilegeComparison,
    -- однозначное/имеющее значением множество priv. (приближение, определяемое приложением)
    extensibleMatch         [10] SEQUENCE {
        matchingRule            OBJECT IDENTIFIER,
        inputs                  PrivilegeComparison } }


PrivilegeComparison   ::= CHOICE {
    explicit        [0] Privilege,
```

```
    -- значение(я) внешней привилегии, определяемой
    -- Privilege.privilegeId сравнивается(ются) со значением(ями),
    -- явно представленным(и) в Privilege.privilegeValueSet
    byReference     [1] PrivilegeIdPair }
    -- значение(я) внешней привилегии, определяемой
    -- PrivilegeIdPair.firstPrivilege сравнивается(ются) со значением(ями)
    -- второй внешней привилегии, определяемой
    -- PrivilegeIdPair.secondPrivilege

Privilege           ::= SEQUENCE {
    type            PRIVILEGE.&id ({SupportedPrivileges}),
    values          SET SIZE (0..MAX) OF
                    PRIVILEGE.&Type ({SupportedPrivileges} {@type})
    }

SupportedPrivileges  PRIVILEGE ::=     { ... }
PRIVILEGE ::=     ATTRIBUTE
-- Привилегия аналогична атрибуту

PrivilegeIdPair     ::=  SEQUENCE {
    firstPrivilege      PrivilegeIdentifier,
    secondPrivilege     PrivilegeIdentifier }

PrivilegeIdentifier ::=  CHOICE {
    privilegeType       [0] PRIVILEGE.&id ({SupportedPrivileges}),
    xmlTag              [1] OCTET STRING,
    edifactField        [2] OCTET STRING }
-- PrivilegeIdentifier расширяет понятие AttributeType на другие
-- (например, помеченные) среды, такие как XML и EDIFACT

Version             ::= INTEGER { v1(0) }
```

Конкретный пример может помочь в прояснении создания и использования вышеописанной структуры **PrivilegePolicy**.

Рассмотрим привилегию санкционировать повышение заработной платы. Для простоты, предположим, что политика, которую нужно внедрить, утверждает, что только старшие менеджеры и выше могут санкционировать повышения, и что санкционирование может быть дано только для должности, ниже чем ваша (например, директор может санкционировать повышение для старшего менеджера, но не для вице-президента). Для данного примера предположим, что возможно шесть должностей ("технический персонал" = 0, "менеджер" = 1, "старший менеджер" = 2, "директор" = 3, "вице-президент" = 4, "президент" = 5).

Далее предположим, что тип атрибута ("привилегия"), определяющий должность в сертификате атрибута, является ID OBJECT *OID-C* и что тип атрибута ("чувствительность"), определяющий должность в записи базы данных, поле зарплата которой должна быть изменена, является ID OBJECT *OID-D* (это должно быть, конечно, заменено реальными идентификаторами объектов в фактической реализации). Следующее булево выражение обозначает желательную политику "санкционирования зарплат" (кодифицирование ее в выражении **PrivilegePolicy** является относительно простой задачей):

И (НЕ (lessOrEqual (значение, соответствующее OID-C, значение, соответствующее OID-D))
subsetOf (значение, соответствующее OID-C, { 2, 3, 4, 5 })).

Данное кодирование политики гласит, что должность санкционирующего должна быть больше чем (выражается как "НЕ меньше или равно") должность санкционируемого И должность санкционирующего должна быть одной из {старший менеджер, ..., президент}, чтобы данное булево выражение приняло значение TRUE. Первое сравнение привилегий происходит "по обращению", сравнивая значения, соответствующие типу атрибута "должность" для обоих участвующих объектов. Второе сравнение привилегий является "явным", здесь значение, соответствующее "должности" привилегии санкционирующего, сравнивается с явно-включенным списком значений. Поэтому верификатору привилегий в данной ситуации необходима структура для кодифицирования данной политики вместе с двумя атрибутами, одним, связанным с санкционирующим и одним, связанным с санкционируемым. атрибут санкционирующего (который должен содержаться в сертификате атрибута) может иметь значение {*OID-C* 3}, а атрибут санкционируемого (который может содержаться в записи базы данных) может иметь значение {*OID-D* 3}. Сравнивая значение атрибута, соответствующее типу атрибута санкционирующего (в данном примере, 3), со значением атрибута, соответствующее типу атрибута санкционируемого (в данном примере, также 3), возвращает значение FALSE для выражения "NOT lessOrEqual", и поэтому первому директору отказывается в возможности санкционировать повышение зарплаты второму директору. С другой стороны, если атрибут санкционируемого был бы *OID-D* 1}, директору была бы предоставлена возможность санкционировать повышение менеджеру.

Не является сложным представить полезные дополнения к приведенному выше выражению. Например, может быть добавлен третий компонент '**and**', гласящий, что переменная среды "currentTime", прочитанная из локальных часов и затем закодированная как тип атрибута ID OBJECT *OID-E*, должна быть в пределах определенного промежутка, явно определенного в выражении как атрибут типа ID OBJECT *OID-F*. Таким образом, обновления зарплат могут разрешаться, только если запрос удовлетворяет вышеуказанным условиям и имеет место в рабочее время.

D.2.2 Второй пример

Политика безопасности в своей простейшей форме является набором критериев для предоставления услуг безопасности. В отношении управления доступом, политика безопасности является подмножеством политики безопасности системы более высокого уровня, которая определяет средства внедрения политик управления доступом между инициаторами и целями. Механизмы управления доступом необходимы для разрешения соединения, если разрешает определенная политика, а также для запрета соединения, если определенная политика не дает явного разрешения.

Политика безопасности является основой для решений, принимаемых механизмами управления доступом. Информация политики безопасности, определенная для каждой области, передается через Файл информации о политике безопасности (SPIF).

SPIF представляет собой подписанный объект для защиты от несанкционированного доступа. SPIF содержит информацию, используемую для толкования параметров управления доступом, содержащихся в метке безопасности и атрибуте допуска. Идентификатор политики безопасности, встречающийся в атрибуте допуска, должен быть связан с определенным синтаксисом реализации и семантикой, как определяется политикой безопасности. Данный синтаксис реализации, связанный с определенной политикой безопасности, содержится в SPIF.

SPIF передает эквивалентности между авторизациями и чувствительностями по областям политики безопасности, как определено политиками безопасности, предоставляет печатаемое представление меток безопасности, а также отображает отображаемые строки для уровней безопасности и категорий для представления оконечным пользователям при выборе атрибутов безопасности объекта данных. Отображения эквивалентностей выражаются таким образом, что метка, сгенерированная согласно одной области политик безопасности, может быть правильно истолкована приложением, действующим в другой области политик безопасности. SPIF также отображает атрибут допуска в поля меток безопасности сообщений и метки представлений, которые должны отображаться пользователю. Данное отображение, если прошло успешно, проверяет, что намеченный получатель имеет правильные авторизации для получения объекта данных.

SPIF содержит последовательность следующего:

- **versionInformation** – указывает версию синтаксиса ASN.1.
- **updateInformation** – указывает версию синтаксиса и семантику спецификации SPIF.
- **securityPolicyIdData** – указывает политику безопасности, к которой применяется SPIF.
- **privilegeId** – указывает OID, определяющий синтаксис, включенный в категорию безопасности атрибута допуска.
- **rbacId** – идентификатор объекта, определяющий синтаксис категории безопасности, использующейся совместно со SPIF.
- **securityClassifications** – отображает классификацию метки безопасности в классификацию в атрибуте допуска, а также обеспечивает отображения эквивалентностей.
- **securityCategoryTagSets** – отображает категории безопасности метки безопасности в категории безопасности в атрибуте допуска, а также обеспечивает отображения эквивалентностей.
- **equivalentPolicies** – объединяет все эквивалентные политики в SPIF.
- **defaultSecurityPolicyIdData** – определяет политику безопасности, которая будет применяться при получении данных без метки безопасности.
- **extensions** – обеспечивает механизм включения дополнительных возможностей по мере определения новых требований в будущем.

Файл информации о политике безопасности определяется в следующем синтаксисе:

```
SecurityPolicyInformationFile ::= SIGNED {SPIF}
SPIF ::=  SEQUENCE    {
    versionInformation              VersionInformationData DEFAULT v1,
    updateInformation               UpdateInformationData,
    securityPolicyIdData                    ObjectIdData,
    privilegeId                             OBJECT IDENTIFIER,
    rbacId                                  OBJECT IDENTIFIER,
    securityClassifications         [0]     SEQUENCE OF SecurityClassification OPTIONAL,
    securityCategories              [1]     SEQUENCE OF SecurityCategory  OPTIONAL,
    equivalentPolicies              [2]     SEQUENCE OF EquivalentPolicy  OPTIONAL,
    defaultSecurityPolicyIdData     [3]     ObjectIdData OPTIONAL,
    extensions                      [4]     Extensions OPTIONAL }

VersionInformationData ::= INTEGER { v1(0) }


UpdateInformationData  ::=  SEQUENCE    {
    sPIFVersionNumber               INTEGER,
    creationDate                    GeneralizedTime,
    originatorDistinguishedName     Name,
    keyIdentifier                   OCTET STRING OPTIONAL }
```

```
ObjectIdData ::=  SEQUENCE  {
    objectId            OBJECT IDENTIFIER,
    objectIdName        DirectoryString {ubObjectIdNameLength} }

SecurityClassification  ::=  SEQUENCE  {
    labelAndCertValue                       INTEGER,
    classificationName                      DirectoryString {ubClassificationNameLength},
    equivalentClassifications   [0]         SEQUENCE OF EquivalentClassification OPTIONAL,
    hierarchyValue                          INTEGER,
    markingData                 [1]         SEQUENCE OF MarkingData OPTIONAL,
    requiredCategory            [2]         SEQUENCE OF OptionalCategoryGroup OPTIONAL,
    obsolete                                BOOLEAN DEFAULT FALSE           }

EquivalentClassification ::= SEQUENCE {
    securityPolicyId            OBJECT IDENTIFIER,
    labelAndCertValue           INTEGER,
    applied                     INTEGER {
                                    encrypt   (0),
                                    decrypt   (1),
                                    both      (2) } }

MarkingData ::= SEQUENCE {
    markingPhrase       DirectoryString {ubMarkingPhraseLength} OPTIONAL,
    markingCodes        SEQUENCE OF MarkingCode OPTIONAL }

MarkingCode ::= INTEGER {
    pageTop                 (1),
    pageBottom              (2),
    pageTopBottom           (3),
    documentEnd             (4),
    noNameDisplay           (5),
    noMarkingDisplay        (6),
    unused                  (7),
    documentStart           (8),
    suppressClassName       (9)}

OptionalCategoryGroup ::= SEQUENCE {
    operation               INTEGER {
                                onlyOne         (1),
                                oneOrMore       (2),
                                all             (3)},
    categoryGroup           SEQUENCE OF OptionalCategoryData }

OptionalCategoryData ::= SEQUENCE {
    optCatDataId            OC-DATA.&id({CatData}),
    categorydata            OC-DATA.&Type({CatData}{@optCatDataId }) }

OC-DATA ::= TYPE-IDENTIFIER

CatData OC-DATA ::= { ... }

EquivalentPolicy  ::= SEQUENCE {
    securityPolicyId        OBJECT IDENTIFIER,
    securityPolicyName      DirectoryString {ubObjectIDNameLength}
    OPTIONAL}

Extensions ::= SEQUENCE OF Extension

Extension ::= SEQUENCE {
    extensionId             EXTENSION.&objId ({ExtensionSet}),
    critical                BOOLEAN DEFAULT FALSE,
    extensionValue              OCTET STRING  }
```

Отметим, что пример SPIF представляет собой развертывающийся синтаксис, и полное определение и описание каждого элемента находится в Рек. МСЭ-Т X.841 | ИСО/МЭК 15816

D.3 Пример атрибута привилегии

Следующий пример атрибута, передающего определенную привилегию, предоставляется исключительно в целях иллюстрации. Фактическая спецификация данного синтаксиса и соответствующий атрибут содержатся в пункте 19.5 в Рек. МСЭ-Т X.501 | ИСО/МЭК 9594-2. Данный определенный атрибут передает допуск, который может быть связан с поименованным объектом, включая DUA для целей соединения с DSA.

Атрибут допуска связывает допуск с поименованным объектом, включая DUA.

```
clearance  ATTRIBUTE ::= {
    WITH SYNTAX                 Clearance
    ID                          id-at-clearance }

Clearance  ::=  SEQUENCE {
    policyId                    OBJECT IDENTIFIER,
    classList                   ClassList DEFAULT {unclassified},
    securityCategories          SET SIZE (1MAX) OF SecurityCategory  OPTIONAL}

ClassList  ::=  BIT STRING {
    unmarked            (0),
    unclassified        (1),
    restricted          (2),
    confidential        (3),
    secret              (4),
    topSecret           (5) }
```

Индивидуальные компоненты описаны в фактических спецификациях данной привилегии в адресуемом документе.

Приложение E

Введение в криптографию открытых ключей[3)]

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В традиционных криптографических системах ключ, используемый для шифрования информации источником секретного сообщения, совпадает с ключом, используемым законным получателем для дешифрования сообщения.

В криптосистемах с открытыми ключами (PKCS), тем не менее, ключи используются парами, один из которых используется для шифрования, а другой – для дешифрования. Каждая пара ключей связана с определенным пользователем X. Один из ключей, известный как открытый ключ (Xp), является общеизвестным и может использоваться любым пользователем для шифрования данных. Только пользователь X, обладающий дополняющим частным ключом (Xs), может дешифровать данные. (Это представляется следующей условной записью: D = Xs[Xp[D]].) Не существует вычислительных средств для получения частного ключа при знании открытого. Таким образом, любой пользователь может передать часть информации, которую может раскрыть только пользователь X, путем шифрования ее при помощи Xp. Расширение данного метода состоит в том, что два пользователя могут устанавливать секретное соединение друг с другом, каждый используя для шифрования данных открытый ключ другого, как показано на рисунке E.1.

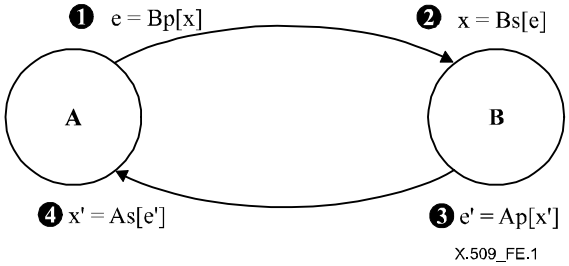


Рисунок E.1 – Использование PKCS для обмена секретной информацией

Пользователь A имеет открытый ключ Ap и частный ключ As, а пользователь B имеет другой набор ключей, Bp и Bs. A и B оба знают открытые ключи друг друга, но не знают частного ключа другой стороны. Поэтому A и B могут обмениваться друг с другом секретной информацией, используя следующие шаги (показанные на рисунке E.1).

1) Пользователь A желает отправить пользователю B некоторую секретную информацию x. Таким образом, пользователь A шифрует x при помощи ключа шифрования пользователя пользователь и отправляет пользователю B зашифрованную информацию e. Это можно представить следующим образом:

$$e = Bp[x].$$

2) Теперь пользователь B может дешифровать данное шифрование e для получения информации x, используя секретный ключ дешифрования Bs. Отметим, что B является единственным обладателем Bs, и в связи с тем, что данный ключ никогда не может быть ни раскрыт, ни передан, получить информацию x не представляется возможным ни для какой другой стороны. Обладание Bs определяет подлинность пользователя B. Операцию дешифрования можно представить следующим образом:

$$x = Bs[e], \text{ или } x = Bs[Bp[x]].$$

3) Теперь пользователь B может аналогично отправить некоторую секретную информацию x' пользователю A при помощи ключа шифрования A, Ap:

$$e' = Ap[x'].$$

4) Пользователь A получает x' путем дешифрования e':

$$x' = As[e'], \text{ или } x' = As[Ap[x']].$$

---

3) Более подробную информацию см.:

ДИФФИ (B.) и ХЭЛЛМАН (M.E.): Новые направления в криптографии, *IEEE Операции по теории информации*, IT-22, № 6, ноябрь 1976 г.

При помощи данного способа, пользователи A и A обменялись секретной информацией x и x'. Данная информация не может быть получена никем, кроме A и B, при условии, что их частные ключи не раскрыты.

Такой обмен, помимо передачи секретной информации между пользователями, может участвовать в проверке их идентификационной информации. В частности, пользователи A и B определяются обладанием секретных ключей дешифрования, As и Bs, соответственно. Пользователь A может определить, обладает ли B секретным ключом дешифрования, путем возвращения части своей информации x в сообщении x' от пользователя B. Это указывает пользователю A на то, что данное соединение имеет место с обладателем Bs. Пользователь B может аналогично проверить идентификационную информацию пользователя A.

Некоторые PKCS обладают свойством, что шаги дешифрования и шифрования могут быть обратными, как в D = Xp[Xs[D]]. Это дает возможность части информации, которая могла быть создана только пользователем Ч, быть прочитанной любым пользователем (обладающим Xp). Таким образом, это может быть использовано при сертификации источника информации и является основной для цифровых подписей. Только PKCS, обладающие данным свойством (перестановочности), подходят для использования в данной структуре аутентификации. Один из подобных алгоритмов описан в Приложении D.

Приложение F

Эталонное определение идентификаторов объектов алгоритмов

(Данное Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В данном Приложении определяются идентификаторы объектов, присвоенные алгоритмам аутентификации и шифрования, при отсутствии формального реестра. Оно предназначено для использования такого реестра, как только он станет доступным. Определения представлены в форме модуля ASN.1, "**AlgorithmObjectIdentifiers**".

```
AlgorithmObjectIdentifiers {joint-iso-itu-t ds(5) module(1) algorithmObjectIdentifiers(8) 5}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --

--              ,                          ,                                                   ASN.1,
--                                                    ,                                      ,
--                                                             .
--                     ,                                                   ,
--                                                      .

IMPORTS
    algorithm, authenticationFramework
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}
    ALGORITHM
        FROM AuthenticationFramework authenticationFramework       ;

-- категории идентификатора объекта --

encryptionAlgorithm    OBJECT IDENTIFIER    ::=    {algorithm 1}
hashAlgorithm          OBJECT IDENTIFIER    ::=    {algorithm 2}
signatureAlgorithm     OBJECT IDENTIFIER    ::=    {algorithm 3}

-- синонимы --

id-ea   OBJECT IDENTIFIER ::=   encryptionAlgorithm
id-ha   OBJECT IDENTIFIER ::=   hashAlgorithm
id-sa   OBJECT IDENTIFIER ::=   signatureAlgorithm

-- алгоритмы --

rsa ALGORITHM    ::=   {
    KeySize
    IDENTIFIED BY id-ea-rsa }

KeySize        ::=   INTEGER

-- следующие присвоения идентификаторов объектов обращают значения, присвоенные
--  исключенным функциям

id-ea-rsa                  OBJECT IDENTIFIER    ::=         {id-ea 1}
id-ha-sqMod-n              OBJECT IDENTIFIER    ::=         {id-ha 1}
id-sa-sqMod-nWithRSA       OBJECT IDENTIFIER    ::=         {id-sa 1}

END
```

Приложение G

Примеры использования ограничений тракта сертификации

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

G.1 Пример 1: Использование основных ограничений

Предположим, что Корпорация Widget хочет перекрестно сертифицировать центральный CA Группы корпораций Acme, но только хочет, чтобы сообщество Widget использовало сертификаты оконечных объектов, выданных данным CA, а не сертификаты, выданные другими CA, сертифицированными данным CA.

Корпорация Widget могла бы удовлетворить данное требование путем выдачи сертификата для центрального CA Acme, включая следующее значение поля расширения:

Значение поля основных ограничений:

{ cA TRUE, pathLenConstraint 0 }

G.2 Пример 2: Использование ограничений отображения политик и политик

Предположим, что между правительствами Канады и США требуется следующий сценарий перекрестной сертификации:

a) CA правительства Канады хочет сертифицировать использование подписей правительства США в отношении политики Канады, называемой *Can/US-Trade*;

b) правительство США имеет политику, называемую *US/Can-Trade*, которую правительство Канады готово считать эквивалентной своей политике *Can/US-Trade*;

c) правительство Канады хочет применить меры безопасности, которые требуют, чтобы все сертификаты США явно утверждали поддержку политик, и запрещают отображение в другие политики в пределах области США.

CA правительства Канады мог бы выдать сертификат для CA правительства США со следующими значениями полей расширений:

Значение поля политик сертификата:

{{ policyIdentifier *-- идентификатор объекта для Can/US-Trade --* **}}**

Значение поля отображения политик:

{{ issuerDomainPolicy *-- идентификатор объекта для Can/US-Trade --* **,**

subjectDomainPolicy *-- идентификатор объекта для US/Can-Trade --* **}}**

Значение поля ограничений политик:

{{ policySet { *-- идентификатор объекта для Can/US-Trade --* **}, requireExplicitPolicy (0),**

inhibitPolicyMapping (0)}}

G.3 Использование расширения ограничений имен

G.3.1 Примеры формата сертификатов с расширением ограничений имен

CA могут налагать различные ограничения на имена субъектов (в поле **subject** и расширении **subjectAltName**) сертификатов, которые они выдают, и последующих сертификатов в тракте сертификации, путем включения расширения ограничения имен в их CA-сертификаты. В данном пункте описаны примеры формата сертификатов с расширением ограничений имен.

Для упрощения примеров, требуемые формы имен (**requiredNameForms**) расширения ограничений имен в данных примерах указывают только имя rfc822 (**rfc822Name**) и DN (**directoryName**).

G.3.1.1 Примеры *permittedSubtrees*

(1-1) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, должно быть эквивалентно или подчинено Acme Inc. в США (т. е. {C=US, O=Acme Inc}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base**(**directoryName**) {C=US, O=Acme Inc}}}	(пусто)	(пусто)

(1-2) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, должно быть эквивалентно или непосредственно подчинено Acme Inc. в США (т. е. {C=US, O=Acme Inc}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base**(**directoryName**) {C=US, O=Acme Inc}, **maximum 1**}}	(пусто)	(пусто)

(1-3) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, должно быть подчинено Acme Inc. в США (т. е. {C=US, O=Acme Inc}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base**(**directoryName**) {C=US, O=Acme Inc}, **minimum 1**}}	(пусто)	(пусто)

(1-4) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, должно быть эквивалентно или подчинено Acme Inc. в США (т. е. {C=US, O=Acme Inc}) или эквивалентно или подчинено Acme Ltd. в U.K. (т. е. {C=UK, O=Acme Ltd}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**directoryName**) {C=UK, O=Acme Ltd}}}	(пусто)	(пусто)

G.3.1.2 Примеры *excludedSubtrees*

(2-1) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, не должно быть ни эквивалентно, ни подчинено Acme Corp. в Канаде (т. е. {C= CA, O= Acme Corp}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
(пусто)	{{**base**(**directoryName**) {C=CA, O=Acme Corp}}}	(пусто)

(2-2) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, не должно быть подчинено каждому непосредственному подчиненному Acme Corp. в Канаде (т. е. {C= CA, O= Acme Corp}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
(пусто)	{{**base**(**directoryName**) {C=CA, O=Acme Corp}, **minimum 2**}}	(пусто)

(2-3) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, не должно быть эквивалентно Acme Corp. в Канаде (т. е. {C= CA, O= Acme Corp}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
(пусто)	{{**base**(**directoryName**) {C=CA, O=Acme Corp}, **maximum 0**}}	(пусто)

(2-4) Если CA-сертификат содержит следующее расширение Ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, не должно быть ни эквивалентно, ни подчинено Asia Acme в Японии (т. е. { C=JP, O=Asia Acme}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
(пусто)	{{**base**(**directoryName**) {C=CA, O=Acme Corp}}, {**base**(**directoryName**) {C=JP, O=Asia Acme}}}	(пусто)

G.3.1.3 Примеры permittedSubtrees и excludedSubtrees

(3-1) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, должно быть эквивалентно или подчинено Acme Inc. в США (т. е. { C=US, O=Acme Inc}), за исключением подразделения организации R&D в Acme Inc. и подчиненных организации R&D.

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base**(**directoryName**) {C=US, O=Acme Inc}}}	{{**base**(**directoryName**) {C=US, O=Acme Inc, OU=R&D}}}	(пусто)

(3-2) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, должно быть равно единице или непосредственно подчинено Acme Inc. в США (т. е. {C=US, O=Acme Inc}), за исключением подразделения организации Снабжения (т. е. {C=US, O=Acme Inc, OU=Purchasing}).

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base**(**directoryName**) {C=US, O=Acme Inc}, **minimum 1**, **maximum 1**}}	{{**base**(**directoryName**) {C=US, O=Acme Inc, OU=Purchasing}}}	(пусто)

G.3.1.4 Примеры permittedSubtrees и excludedSubtrees с requiredNameForms

(4-1) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации по меньшей мере одно из имен субъекта (в поле **subject** или расширении **subjectAltName**) сертификата должно быть в форме имен DN. Тем не менее, каждое имя субъекта не ограничивается какими-либо пространствами имен.

Расширение **nameConstraints**			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		имя rfc822	DN
(пусто)	(пусто)	ВЫКЛЮЧЕН	ВКЛЮЧЕН

(4-2) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации по меньшей мере одно из имен субъекта (в поле **subject** или расширении **subjectAltName**) сертификата должно быть в форме имен DN. Более того, каждое имя субъекта в форме имен DN должно удовлетворять пространствам имен, ограниченным **permittedSubtrees** и **excludedSubtrees**.

Расширение **nameConstraints**			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		имя rfc822	DN
{{**base**(**directoryName**) {C=JP, O=Asia Acme}}}	{{**base**(**directoryName**) {C=JP, O=Asia Acme, OU=Marketing}}}	ВЫКЛЮЧЕН	ВКЛЮЧЕН

(4-3) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, должно удовлетворять пространствам имен, ограниченным **permittedSubtrees** и **excludedSubtrees**.

Расширение **nameConstraints**			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		имя rfc822	DN
{{**base**(**directoryName**) {C=JP, O=Asia Acme}}}	{{**base**(**directoryName**) {C=JP, O=Asia Acme, OU=Marketing}}}	ВЫКЛЮЧЕН	ВЫКЛЮЧЕН

ПРИМЕЧАНИЕ. – Приведенный выше пример CA-сертификата совместим со следующим CA-сертификатом с расширением ограничения имен без элемента **requiredNameForms**.

Расширение **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base**(**directoryName**) {C=JP, O=Asia Acme}}}	{{**base**(**directoryName**) {C=JP, O=Asia Acme, OU=Marketing}}}	(пусто)

(4-4) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, если существует, должно удовлетворять пространствам имен, ограниченным **permittedSubtrees** и **excludedSubtrees**. Более того, должен присутствовать по меньшей мере один **subjectAltName** в форме имен **rfc822Name**, хотя его имя не ограничивается какими-либо пространствами имен.

Расширение **nameConstraints**			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		имя rfc822	DN
{{**base**(**directoryName**) {C=JP, O=Asia Acme}}}	{{**base**(**directoryName**) {C=JP, O=Asia Acme, OU=Marketing}}}	ВКЛЮЧЕН	ВЫКЛЮЧЕН

(4-5) Если CA-сертификат содержит следующее расширение ограничений имен, для всех последующих сертификатов в тракте сертификации по меньшей мере одно из имен субъекта (в поле **subject** или расширении **subjectAltName**) сертификата должно быть в форме имен DN или в форме имен rfc822. Каждое имя субъекта в форме имен DN, если существует, должно удовлетворять пространствам имен, ограниченным **permittedSubtrees** и **excludedSubtrees**. Каждое имя субъекта в форме имен **rfc822Name** не ограничивается какими-либо пространствами имен.

Расширение **nameConstraints**			
permittedSubtrees	**ExcludedSubtrees**	**requiredNameForms**	
		имя rfc822	DN
{{**base**(**directoryName**) {C=JP, O=Asia Acme}}}	{{**base**(**directoryName**) {C=JP, O=Asia Acme, OU=Marketing}}}	ВКЛЮЧЕН	ВКЛЮЧЕН

G.3.2 Примеры управления сертификатами с расширением ограничений имен

В данном пункте описаны примеры проверки подлинности имен субъектов (в поле **subject** или расширении **subjectAltName**) при обработке сертификата с переменными состояния обработки тракта, а именно *разрешенные-поддеревья, исключенные-поддеревья* и *требуемые-формы-имен*.

Для упрощения примеров, переменные состояния обработки тракта *требуемые-формы-имен* в данных примерах указывают только имя rfc822 (**rfc822Name**), DN (**directoryName**) и URI (**uniformResourceIdentifier**).

G.3.2.1 Ограничения пространств имен посредством *разрешенных-поддеревьев* в форме имен DN

В данном случае каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, встречающееся в рассматриваемом сертификате, должно удовлетворять ограничению путем переменной состояния обработки тракта *разрешенные-поддеревья.*

(1-1) Присутствует одно разрешенное поддерево для DN и DN требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}}	ОТСУТСТВУЕТ	ВЫКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName** (**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName** (**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName** (**rfc822Name**) = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName** (**directoryName**) = {C=US, O=Acme Inc, OU=Accounting}

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Ltd*,OU=Purchasing}
2	**subject** = {} **subjectAltName** (**rfc822Name**) = manager@purchasing.acme.com ПРИМЕЧАНИЕ. – *DN отсутствует.*
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName** (**directoryName**) = {C=US, O=*Acme Ltd*, OU=Purchasing}
4	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName** (**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName** (**directoryName**) = {C=US, O=*Acme Ltd*, OU=Accounting}

(1-2) Присутствуют два разрешенных поддерева для DN и DN требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**directoryName**) {C=US, O=Acme Ltd}}}	ОТСУТСТВУЕТ	ВЫКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing}
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU= Accounting}

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme International*, OU=Accounting}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com ПРИМЕЧАНИЕ. – *DN отсутствует.*
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme International*, OU=Accounting}
4	**subject** = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
5	**subject** = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Corp*, OU=Accounting}
6	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com

(1-3) Присутствует одно разрешенное поддерево для DN и *требуемые-формы-имен* являются пустыми.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}}	ОТСУТСТВУЕТ	Пустые		

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Accounting}
6	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Ltd*, OU=Accounting}
3	**subject** = {C=US, O=*Acme Ltd*, OU=Accounting} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
4	**subject** = {C=US, O=*Acme Ltd*, OU=Accounting} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Ltd*, OU=Purchasing}

G.3.2.2 Ограничения пространств имен посредством *исключенных-поддеревьев* в форме имен DN

В данном случае каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN, встречающееся в рассматриваемом сертификате, должно удовлетворять ограничению путем переменной состояния обработки тракта *исключенные-поддеревья.*

(2-1) Присутствует одно исключенное поддерево для DN и DN требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
ОТСУТСТВУЕТ	{{**base**(**directoryName**) {C=US, O=Acme Ltd}}}	ВЫКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Accounting}

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com ПРИМЕЧАНИЕ. – *DN отсутствует.*
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Ltd*, OU=Accounting}

(2-2) Присутствуют два исключенных поддерева для DN и DN требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
ОТСУТСТВУЕТ	{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**directoryName**) {C=US, O=Acme Ltd}}}	ВЫКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme International, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme International, OU=Purchasing}
3	**subject** = {C=US, O=Acme International, OU=Purchasing} **subjectAltName**(**rfc822Name**) = purchasing@acme-international.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme International, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme N.Y, OU=Purchasing} **subjectAltName**(**rfc822Name**) = purchasing@acme-international.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
3	**subject** = {} **subjectAltName**(**rfc822Name**) = purchasing@acme-international.com ПРИМЕЧАНИЕ. – *DN отсутствует*.
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme International, OU=Accounting}
5	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme International, OU=Purchasing} **subjectAltName**(**rfc822Name**) = purchasing@acme-international.com

(2-3) Присутствует одно исключенное поддерево для DN и *требуемые-формы-имен* являются пустыми.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
ОТСУТСТВУЕТ	{{**base**(**directoryName**) {C=US, O=Acme Inc}}}	пустые		

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing}
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}
6	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}

G.3.2.3 Ограничения пространств имен посредством только требуемых-форм-имен

(3-1) DN требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
ОТСУТСТВУЕТ	ОТСУТСТВУЕТ	ВЫКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=JP, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=JP, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
5	**subject** = {C=JP, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}

Примеры неприемлемых сертификатов

1	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com ПРИМЕЧАНИЕ. – *DN отсутствует.*
2	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com ПРИМЕЧАНИЕ. – *DN отсутствует.*

(3-2) DN или **rfc822Name** требуются в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
ОТСУТСТВУЕТ	ОТСУТСТВУЕТ	ВКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=JP, O=Acme Inc, OU=Purchasing}
3	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
5	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com **subjectAltName**(**rfc822Name**) = purchasing@acme-ltd.com

Примеры неприемлемых сертификатов

1	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com ПРИМЕЧАНИЕ. – *DN и rfc822 отсутствуют.*
2	**subject** = {} **subjectAltName**(**dNSName**) = www.acme-ltd.com ПРИМЕЧАНИЕ. – *DN и rfc822 отсутствуют.*

G.3.2.4 Ограничения пространств имен посредством *разрешенных-поддеревьев* в нескольких формах имен

В данном случае каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN или форме имен rfc822, встречающееся в рассматриваемом сертификате, должно удовлетворять ограничению путем переменной состояния обработки тракта *разрешенные-поддеревья.*

(4-1) Присутствуют одно разрешенное поддерево для DN и другое разрешенное поддерево для **rfc822Name**. Более того, DN требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	ОТСУТСТВУЕТ	ВЫКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Accounting}
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com ПРИМЕЧАНИЕ. – *DN отсутствует.*
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-ltd.com*
6	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com ПРИМЕЧАНИЕ. – *DN отсутствует.*

(4-2) Присутствуют одно разрешенное поддерево для DN и другое разрешенное поддерево для **rfc822Name**. Более того, по меньшей мере один из DN или **rfc822Name** требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	ОТСУТСТВУЕТ	ВКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = { C=US, O=Acme Inc, OU=Accounting}
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-ltd.com*
6	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com ПРИМЕЧАНИЕ. – *DN и rfc822 отсутствуют*.

(4-3) Присутствуют одно разрешенное поддерево для DN и другое разрешенное поддерево для **rfc822Name**. В *требуемых-формах-имен* не требуется никаких форм имен.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	ОТСУТСТВУЕТ	пустые		

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*

G.3.2.5 Ограничения пространств имен посредством *исключенных-поддеревьев* в нескольких формах имен

В данном случае каждое имя субъекта (в поле **subject** или расширении **subjectAltName**) в форме имен DN или форме имен rfc822, встречающееся в рассматриваемом сертификате, должно удовлетворять ограничению путем переменной состояния обработки тракта *исключенные-поддеревья.*

(5-1) Присутствуют одно исключенное поддерево для DN и другое исключенное поддерево для **rfc822Name**. Более того, DN требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
ОТСУТСТВУЕТ	{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	ВЫКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}
4	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com ПРИМЕЧАНИЕ. – *DN отсутствует.*
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
6	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Inc*, OU=Accounting}
7	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com ПРИМЕЧАНИЕ. – *DN отсутствует.*

(5-2) Присутствуют одно исключенное поддерево для DN и другое исключенное поддерево для **rfc822Name**. Более того, по меньшей мере один из DN или **rfc822Name** требуется в *требуемых-формах-имен*.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
ОТСУТСТВУЕТ	{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	ВКЛЮЧЕН	ВКЛЮЧЕН	ВЫКЛЮЧЕН

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.org
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
6	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Inc*, OU=Accounting}
7	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com ПРИМЕЧАНИЕ. – *DN и rfc822 отсутствуют*.

(5-3) Присутствуют одно исключенное поддерево для DN и другое исключенное поддерево для **rfc822Name**. В *требуемых-формах-имен* не требуется никаких форм имен.

Переменные состояния обработки тракта				
разрешенные-поддеревья	*исключенные-поддеревья*	*требуемые-формы-имен*		
		rfc822	DN	URI
ОТСУТСТВУЕТ	{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	Пустые		

Примеры приемлемых сертификатов

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com

Примеры неприемлемых сертификатов

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
6	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Inc*, OU=Accounting}

Приложение H

Руководство по определению политик, для которых тракт сертификации является действительным

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Целью данного Приложения является предоставление руководства для работающих в PKI приложений относительно управления связанной с политикой сертификатов обработкой проверки подлинности тракта сертификации. Управление PKI связанной с политикой сертификатов обработкой посредством содержимого сертификатов описано в пункте Процедура обработки тракта сертификации в данной Спецификации.

В данном Приложении рассматривается инициализация двух связанных с политикой входных значений в процедуру обработки тракта: *начального-набора-политик* и *начальной-явной-политики*. Дополнительно, входные значения для процедуры *начальный-запрет-отображения-политик* и *начальное-запретить-любую-политику*, которые также могут быть инициализированы пользователем, влияют на обработку информации, связанной с политикой, в течение обработки тракта; тем не менее, они находятся вне области применения данного Приложения. Установка *начального-запрета-отображения-политик* в значение **TRUE** предотвращает использование отображения политик при успешных проверках подлинности трактов. Установка *начального-запретить-любую-политику* в значение **TRUE** предотвращает то, что особый OID для **anyPolicy**, при наличии в сертификате, будет приемлемым соответствием для OID определенной политики.

Термин "пользователь" в данном Приложении может использоваться для обозначения "пользователя-человека" или работающего в PKI "приложения".

Можно предвидеть следующие сценарии:

1) Пользователь требует, чтобы тракт сертификации был действителен для одной из рассматриваемых пользователем политик.
2) Пользователь требует, чтобы тракт сертификации был действителен по меньшей мере для одной политики, но для пользователя не имеет значения, что это за политика. Данный сценарий должен (может) использоваться, когда пользователь намерен выполнить дополнительную обработку политики, используя другую контекстуальную информацию и информационное содержимое, чтобы определить, является ли одна из политик, для которой тракт сертификации является действительным, приемлемой для пользователя для определенной операции.
3) Пользователь не имеет связанных с политикой требований к тракту сертификации. Другими словами, пользователь желает принять тракт сертификации, не являющийся действительным ни для одной из политик, но иначе действительный.
4) Пользователь желает, чтобы тракт сертификации был действителен для одной из рассматриваемых пользователем политик, но при невозможности этого, хочет иметь возможность пересмотреть тракты, не являющиеся действительными для рассматриваемых пользователем политик. Данный сценарий должен (может) использоваться, когда пользователь в общем случае требует, чтобы тракт сертификации был действителен для приемлемой для пользователя политики, но, основываясь на другой контекстуальной информации и информационного содержимого, пользователь может пожелать отклонить неудачу политики.

В следующих пунктах описано, как пользователю следует получить желаемую информацию из совместимого устройства проверки подлинности тракта.

H.1 Тракт сертификации, действительный для требуемой политики, определяемой пользователем

В данном сценарии, пользователь требует, чтобы тракт сертификации был действителен для одной из рассматриваемых пользователем политик. Чтобы получить желаемую информацию, пользователь должен установить входные значения для связанной с обработкой политики проверки подлинности тракта сертификации следующим образом:

начальный-набор-политик = {набор рассматриваемых пользователем политик}

начальная-явная-политика = **TRUE**.

Если проверка подлинности тракта является успешной, то тракт сертификации является действительным по меньшей мере для одной из политик, рассматриваемых пользователем. Тракт сертификации является действительным для политик, перечисленных в итоговой переменной *набор-политик-ограниченных-пользователем.*

Согласно данному сценарию, приложения не должны использовать тракт сертификации, если он отклонен устройством проверки подлинности тракта по причинам, связанным с неудачами политик сертификатов[4)].

---

4) Неудача проверки подлинности тракта является неудачей, связанной с политикой сертификатов, если неудача вызвана связанными(и) с политикой сертификатов расширением(ями) или связанными(и) с политикой сертификатов переменной(ыми) состояния. К связанным с политикой сертификатов расширениям относятся: **certificatePolicies**, **policyMappings**, **policyConstraints** и **inhibitAnyPolicy**. К связанным с политикой сертификатов переменным состояния относятся: *набор-политик-ограниченных-органами, индикатор-явной-политики, индикатор-запрета-отображения-политик* и *индикатор-запретить-любую-политику*.

H.2 Тракт сертификации, действительный для любой требуемой политики

В данном сценарии, пользователь требует, чтобы тракт сертификации был действителен по меньшей мере для одной политики, но для пользователя не имеет значения, что это за политика. Чтобы получить желаемую информацию, пользователь должен установить входные значения для связанной с обработкой политики проверки подлинности тракта сертификации следующим образом:

начальный-набор-политик = {**anyPolicy**}

начальная-явная-политика = **TRUE**.

Если проверка подлинности тракта является успешной, то тракт сертификации является действительным по меньшей мере для одной из политик. Тракт сертификации является действительным для политик, перечисленных в итоговой переменной *набор-политик-ограниченных-пользователем*.

Согласно данному сценарию, приложения не должны использовать тракт сертификации, если он отклонен устройством проверки подлинности тракта по причинам, связанным с неудачей политик сертификатов.

H.3 Тракт сертификации, действительный независимо от политики

В данном сценарии, пользователь не имеет связанных с политикой требований к тракту сертификации. Чтобы получить желаемую информацию, пользователь должен установить входные значения для связанной с обработкой политики проверки подлинности тракта сертификации следующим образом:

начальный-набор-политик = {**anyPolicy**}

начальная-явная-политика = **FALSE**.

Если проверка подлинности тракта является успешной, то тракт сертификации является действительным для политик, перечисленных в итоговой переменной *набор-политик-ограниченных-пользователем*.

Согласно данному сценарию, приложения не должны использовать тракт сертификации, если он отклонен устройством проверки подлинности тракта по причинам, связанным с неудачей политик сертификатов.

Необходимо отметить, что в данном сценарии тракт сертификации может иметь неудачу, связанную с политикой. Например, если инфраструктура (т. е. сертификат CA в тракте сертификации) устанавливает *индикатор-явной-политики.* В данном случае, если тракт не является действительным ни для одной политики, т. е. *набор-политик-ограниченных-пользователем* является пустым, то совместимое устройство проверки подлинности тракта возвратит неудачу. Приложения должны отклонять тракт сертификации по причине неудачи данного типа.

H.4 Тракт сертификации, действительный для желательной, но не требуемой политики, определяемой пользователем

В данном сценарии, пользователь желает, чтобы тракт сертификации был действителен для одной из рассматриваемых пользователем политик, но не хочет отклонять тракты, не являющиеся действительными ни для одной рассматриваемых пользователем политик. Чтобы получить желаемую информацию, пользователь должен установить входные значения для связанной с обработкой политики проверки подлинности тракта сертификации следующим образом:

начальный-набор-политик = {набор рассматриваемых пользователем политик}

начальная-явная-политика = **FALSE**.

Если проверка подлинности тракта является успешной, то тракт сертификации является действительным для политик, перечисленных в итоговой переменной *набор-политик-ограниченных-пользователем. Набор-политик-ограниченных-пользователем* является подмножеством *начального-набора-политик*. Отметим, что в данном случае *набор-политик-ограниченных-пользователем* может быть **NULL**, если не установлен *индикатор-явной-политики*. Приложение должно проверить возвращенный *набор-политик-ограниченных-пользователем* для определения того, является ли тракт приемлемым для пользователя.

Приложения должны отклонять тракт сертификации из-за неудачи, связанной в политикой, вызванной инфраструктурой в данном сценарии (т. е. когда *набор-политик-ограниченных-пользователем* является пустым и не установлен *индикатор-явной-политики*).

Необходимо отметить, что в данном сценарии тракт сертификации может иметь неудачу, связанную с политикой. Например, если инфраструктура (т. е. сертификат CA в тракте сертификации) устанавливает *индикатор-явной-политики.* В данном случае, если тракт не является действительным ни для одной политики, т. е. *набор-политик-ограниченных-пользователем* является пустым, то совместимое устройство проверки подлинности тракта возвратит неудачу. Приложения должны отклонять тракт сертификации по причине неудачи данного типа.

Другим примером является сочетание входных данных пользователя и инфраструктуры, вызывающей неудачу, связанную с политикой. Это происходит, когда сертификат CA в тракте сертификации устанавливает *индикатор-явной-политики, набор-политик-ограниченных-органами* является непустым*, а набор-политик-ограниченных-пользователем* является пустым. Совместимое устройство проверки подлинности тракта возвратит неудачу. В данных условиях если единственной причиной возвращения неудачи устройством проверки подлинности тракта является то, что *набор-политик-ограниченных-пользователем* является пустым, то приложения могут принять решение отклонить данную неудачу и принять тракт сертификации. Ограничения, наложенные органом, по-прежнему соблюдаются в силу того, *набор-политик-ограниченных-органами* что не является пустым. Принятие данного тракта приложением эквивалентно тому, что приложение повторно представляет тракт на рассмотрение устройству проверки подлинности со значениями *начального-набора-политик*, равного **anyPolicy**, и *начальной-явной-политики*, равной **FALSE**, и проверяет возвращенный *набор-политик-ограниченных-пользователем* для определения того, является ли тракт приемлемым.

Приложение I

Вопросы расширений сертификата использования ключа

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Сочетание бита contentCommitment в расширении сертификата keyUsage с другими битами keyUsage может иметь последствия безопасности в зависимости от среды безопасности, в которой должен использоваться сертификат. Если средой субъекта можно полностью управлять и доверять, то определенных последствий безопасности нет. Например, в случаях, когда субъект полностью уверен в том, какие именно данные подписываются, или полностью уверен в характеристиках безопасности используемого протокола аутентификации. Если средой субъекта нельзя полностью управлять или доверять, то возможно непреднамеренное подписание обязательств. К примерам относятся использование плохо сформированных обменов аутентификации и использование неисправного компонента программного обеспечения. Если субъект использует недоверенные среды, то данные последствия безопасности могут быть ограничены путем применения следующих мер:

- не сочетать настройку использования ключа contentCommitment в сертификате с какой-либо другой настройкой использования и использовать соответствующий частный ключ только в данном сертификате;
- ограничить использование частного ключа, связанного с сертификатами с установленным битом использования ключа contentCommitment, до сред, считающихся должным образом управляемыми и надежными.

Приложение J

Алфавитный указатель определений единиц информации

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В данном Приложении представлен алфавитный указатель к определениям форматов сертификатов и CRL, расширений сертификатов, классов объектов, форм имен, типов атрибутов и правил соответствия, определенных в данной спецификации Справочника.

Единица информации	Пункт
Форматы сертификатов и CRL	
Список аннулированных сертификатов	7.3
Формат сертификата атрибута	12.1
Формат сертификата открытого ключа	7
Расширения сертификатов, CRL и записей CRL	
Расширение альтернативного имени выдавшего органа	8.3.2.2
Расширение альтернативного имени субъекта	8.3.2.1
Расширение аннулированной группы сертификатов	8.5.2.11
Расширение атрибутов справочника субъекта	8.3.2.3
Расширение быть аннулированным	8.5.2.10
Расширение выдано от имени	15.5.2.6
Расширение выдающей точки распределения	8.6.2.2
Расширение границ CRL	8.5.2.5
Расширение даты недействительности	8.5.2.4
Расширение дескриптора атрибута	15.3.2.2
Расширение запретить любую политику	8.4.2.4
Расширение идентификатора SOA	15.3.2.1
Расширение идентификатора атрибута органа	15.5.2.4
Расширение идентификатора ключа органа	8.2.2.1
Расширение идентификатора ключа субъекта	8.2.2.2
Расширение идентификатора потока CRL	8.5.2.7
Расширение идентификатора сертификата спецификации роли	15.4.2.1
Расширение индикатора дельта CRL	8.6.2.4
Расширение информации дельта	8.5.2.9
Расширение использования ключа	8.2.2.3
Расширение истекших сертификатов в CRL	8.5.2.12
Расширение кода инструкции удержания	8.5.2.3
Расширение кода причины	8.5.2.2
Расширение наиболее нового CRL	8.6.2.6
Расширение направления информации	15.1.2.2
Расширение непрямого выдающего органа	15.1.2.5
Расширение нет заявления	15.1.2.6
Расширение нет информации об аннулировании	15.2.2.2
Расширение номера CRL	8.5.2.1
Расширение ограничений делегированных имен	15.5.2.2
Расширение ограничений имен	8.4.2.2
Расширение ограничений политик	8.4.2.3
Расширение органа, выдающего сертификаты	8.6.2.3
Расширение основного обновления	8.6.2.5
Расширение основных ограничений	8.4.2.1
Расширение основных ограничений атрибутов	15.5.2.1

Единица информации	Пункт
Расширение отображений политик	8.2.2.7
Расширение передачи статуса	8.5.2.6
Расширение периода использования частного ключа	8.2.2.5
Расширение политик сертификатов	8.2.2.6
Расширение приемлемых политик привилегий	15.1.2.4
Расширение приемлемых политик сертификатов	15.5.2.3
Расширение расширенного использования ключа	8.2.2.4
Расширение спецификации времени	15.1.2.1
Расширение точек распределения CRL	8.6.2.1
Расширение уведомления пользователя	15.1.2.3
Расширение упорядоченного списка	8.5.2.8
Классы объектов и формы имен	
Класс объектов AA PMI	17.1.2
Класс объектов CA PKI	11.1.2
Класс объектов SOA PMI	17.1.3
Класс объектов дельта CRL	11.1.4
Класс объектов защищенная политика привилегий	17.1.7
Класс объектов и форма имени точек распределения CRL	11.1.3
Класс объектов политика привилегий	17.1.6
Класс объектов политика сертификатов и CPS	11.1.5
Класс объектов пользователь PKI	11.1.1
Класс объектов пользователь PMI	17.1.1
Класс объектов точки распределения CRL сертификатов атрибутов	17.1.4
Класс объектов тракт сертификатов PKI	11.1.6
Тракт делегирования PMI	17.1.5
Атрибуты Справочника	
Атрибут дельта списка аннулирования	11.2.6
Атрибут защищенной политики привилегий	17.2.8
Атрибут защищенной политики привилегий XML	17.2.9
Атрибут информации о привилегиях XML	14.5
Атрибут пары перекрестных сертификатов	11.2.3
Атрибут поддерживаемых алгоритмов	11.2.7
Атрибут политики привилегий	17.2.7
Атрибут политики сертификатов	11.2.9
Атрибут сертификата AA	17.2.2
Атрибут сертификата CA	11.2.2
Атрибут сертификата атрибутов	17.2.1
Атрибут сертификата дескриптора атрибута	17.2.3
Атрибут сертификата пользователя	11.2.1
Атрибут списка аннулирования органов	11.2.5
Атрибут списка аннулирования сертификатов атрибутов	17.2.4
Атрибут списка аннулированных сертификатов	11.2.4
Атрибут списка аннулированных сертификатов AA	17.2.5
Атрибут тракта PKI	11.2.10
Атрибут тракта делегирования	17.2.6
Атрибут утверждения выполнения сертификации	11.2.8
Правила соответствия	
Расширенное соответствие сертификатов	11.3.10
Соответствие ID сертификатов спецификации ролей	15.4.2.1.1
Соответствие выдавших органов держателя	17.3.3

Единица информации	Пункт
Соответствие дескрипторов атрибутов	15.3.2.2.1
Соответствие идентификаторов AA	15.5.2.4.1
Соответствие идентификаторов SOA	15.3.2.1.1
Соответствие идентификаторов алгоритмов	11.3.7
Соответствие непрямых выдавших органов	15.1.2.5
Соответствие ограничений делегированных имен	15.5.2.2.1
Соответствие основных ограничений атрибутов	15.5.2.1.1
Соответствие пар сертификатов	11.3.4
Соответствие политик	11.3.8
Соответствие приемлемых политик сертификатов	15.5.2.3.1
Соответствие сертификатов	11.3.2
Соответствие сертификатов атрибутов	17.3.2
Соответствие спецификаций времени	15.1.2.1.1
Соответствие списков сертификатов	11.3.6
Соответствие трактов PKI	11.3.9
Соответствие трактов делегирования	17.3.4
Точное соответствие пар сертификатов	11.3.3
Точное соответствие сертификатов	11.3.1
Точное соответствие сертификатов атрибутов	17.3.1
Точное соответствие списков сертификатов	11.3.5

Приложение K

Поправки и исправления

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В настоящее издание данной спецификации Справочника включены следующие проекты поправок, по которым было проведено голосование и которые были утверждены ИСО/МЭК:

– Поправка 4 для расширений Сертификатов открытых ключей и атрибутов.

Настоящее издание данной спецификации Справочника включает следующие технические исправления, которые устраняют дефекты, указанные в следующих сообщениях о дефектах относительно 4-го издания данной Спецификации:

– техническое исправление 1 (охватывает сообщения о дефектах 272, 273, 274, 275, 276, 277, 278 и 279);

– техническое исправление 2 (охватывает сообщения о дефектах 284, 285 и 286); и

– техническое исправление 3 (охватывает сообщения о дефектах 281, 282, 289, 291, 296, 298, 299, 300, 301, 304 и 305).

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия A	Организация работы МСЭ-Т
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	Управление электросвязью, включая СУЭ и техническое обслуживание сетей
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	**Сети передачи данных, взаимосвязь открытых систем и безопасность**
Серия Y	Глобальная информационная структура, аспекты межсетевого протокола и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи

Отпечатано в Швейцарии
Женева, 2007 г.