ITUPublications

International Telecommunication Union
Standardization Sector

ITU-T Technical Report

(03/2023)

XSTR.ibc-cd

Guidelines for identity-based cryptosystems used for cross-domain secure communications

Technical Report ITU-T XSTR.ibc-cd

Guidelines for identity-based cryptosystems used for cross-domain secure communications

Summary

Secure communications take place not only within an operator's network but also across operators' networks. Public-key cryptosystem has become the foundation for secure communications since it was invented. An identity-based public-key cryptography (ID-PKC) system has the advantage over the PKI-based public-key cryptography (PKI-PKC) system as ID-PKC removes the need for certificate management. However, current bootstrap schemes for ID-PKC rely on the availability of PKI. The multi-CA trust issue in the PKI-PKC system is transmitted to the ID-PKC system.

In this Technical Report, the secure bootstrap of an ID-PKC without relying on PKI is studied. The weaknesses of current identity-based cryptography systems for cross-domain secure communications are identified and potential solutions to overcome these weaknesses are introduced. Further, the evaluation of these solutions and a way forward to standardization is given.

Keywords

Bootstrap, cross-domain, identity-based public-key cryptography (ID-PKC), key generation centre (KGC), PKI-based public-key cryptography (PKI-PKC).

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Table of Contents

Technical Report ITU-T XSTR.ibc-cd

Guidelines for identity-based cryptosystems used for cross-domain secure communications

1 Scope

This Technical Report provides guidelines for identity-based cryptosystems used for cross-domain secure communications:

- It surveys and assesses the PKI-based bootstrap mechanism for an ID-PKC system;
- It specifies potential solutions to distribute genuine public parameters and identity revocation list without PKI; and
- It evaluates the potential solutions and put the way forward to standardization

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

This Technical Report uses the following terms defined elsewhere:

3.1.1 blockchain [b-ITU-T F.751.0]: A type of distributed ledger that is composed of digitally recorded data arranged as a successively growing chain of blocks with each block cryptographically linked and hardened against tampering and revision.

3.1.2 certification authority (CA) [b-ITU-T X.509]: An authority trusted by one or more entities to create and digital sign public-key certificates. Optionally the certification authority may create the subjects' keys.

3.1.3 cross-certificate [b-ITU-T X.509]: A certification authority (CA) certificate where the issuer and the subject are different CAs. CAs issue cross-certificates to other CAs as a mechanism to authorize the subject CA's existence.

3.1.4 distributed ledger [b-ITU-T F.751.0]: A type of ledger that is shared, replicated, and synchronized in a distributed and decentralized manner.

3.1.5 distributed ledger technology (DLT) [b-ISO/TC 307]: Technology that enables the operation and use of distributed ledgers.

3.1.6 distributed ledger technology system [b-ISO/TC 307]: A system that implements a distributed ledger.

3.1.7 ledger [b-ITU-T X.1400]: Information store that keeps final and definitive (immutable) records of transactions.

3.1.8 private key [b-ITU-T X.509]: (in a public-key cryptosystem) That key of an entity's key pair which is known only by that entity.

3.1.9 public key [b-ITU-T X.509]: That key of an entity's key pair which is publicly known.

3.1.10 public-key infrastructure (PKI) [b-ITU-T X.509]: The infrastructure able to support the management of public keys able to support authentication, encryption, integrity or non-repudiation services.

3.2 Terms defined in this Technical Report

This Technical Report defines the following terms:

3.2.1 identity-based public-key cryptography (ID-PKC) system: A system where publicly known identity information, such as phone numbers and e-mail addresses, serves as the public key.

3.2.2 identity management server (IMS): A server whose function is to manage user identity, including ensuring the uniqueness of user identity in the management domain, maintaining identity status (valid, revoked), and publishing an identity revocation list. The communication channel between the user and IMS should be authentic but not necessarily confidential.

3.2.3 identity revocation list (IRL): A set of the revoked identities.

NOTE – Adapted from [b-ITU-T X.1365]

3.2.4 key generator centre (KGC): A centre composed of the private-key generator, public parameter server and identity management server.

3.2.5 private-key generator (PKG): This generates a user's private key based on the securely stored master secret of the ID-PKC system and the user's identity. The private key is distributed to the user via a secure channel, which provides confidentiality and integrity protection.

NOTE – Adapted from [b-IETF RFC 5408]

3.2.6 public parameter server (PPS): A server whose function is to provide public parameters of the ID-PKC system and policy information that describes the operation of a PKG to users.

NOTE – Adapted from [b-IETF RFC 5408]

3.2.7 permissioned distributed ledger technology system: A distributed ledger technology (DLT) system in which permissions are required to maintain and operate a node.

NOTE – Adapted from [b-ITU-T X.1400].

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

CA	Certification Authority
DLT	Distributed Ledger Technology
ID-PKC	Identity-based Public-Key Cryptography
IMS	Identity Management Server
IRL	Identity Revocation List
KGC	Key Generation Centre
PKC	Public-Key Cryptography
PKI	Public-Key Infrastructure
PKI-PKC	PKI-based Public-Key Cryptography
PKG	Private-Key Generator
PPS	Public Parameter Server

RA	Registration Authority
TLS	Transport Layer Security
TTP	Trusted Third Party

5 Conventions

None.

6 Overview

Secure communications take place not only within an operator's network but also across operators' networks as network operators run their own network independently. Public-key cryptography is an essential method used for secure communications. It is easy for an operator to perform secure communication inside its own network as the trust of public keys can be well managed within a network. However, it is difficult to realize secure across-domain communications as there are many trust roots for public keys in this context.

The public-key cryptosystem has become the foundation for secure communications since it's invention [b-Diffie]. It can provide fundamental security primitives including confidentiality, authenticity and non-repudiation. A conventional public-key cryptosystem relies on public-key infrastructure (PKI) to assure the authenticity of the public keys. The public key and identity of a user are cryptographically bound by using the digital signature, and they constitute the main components of the certificate for the user. Thus, the conventional public-key cryptosystem is also called a PKI-based public-key cryptography (PKI-PKC) system. PKI is responsible for the certificate management, i.e., certificate creation, certificate distribution and certificate revocation. The complexity of the PKI has become a great impediment to the widespread usage of the PKI-PKC system. Certification authority (CA) is the key component in PKI. Usually, each operator runs its own PKI, meaning that the trust of multi-CA becomes a key issue for secure across-domain communications. Although several schemes, such as bridge CA, are available, they are rarely applied in practice due to their complexity.

As an alternative approach of the public-key cryptosystem, an identity-based public-key cryptography (ID-PKC) system [b-Shamir] and [b-Boneh] has been proposed to eliminate the need for the certificate, as the publicly known identity information, such as phone numbers and email addresses, are used as the public keys in the system. The operation of ID-PKC system relies on a trusted third party called key generation centre (KGC). Usually, each domain has its own KGC. The KGC generates a master public key and a master private key; the master public key can be published, and the master private key is kept secret. The KGC derives the private key for a user by using the user's identity, a set of public parameters (including the master public key) and a master private key. It is obvious that the private key shall be delivered to the corresponding user securely. The public parameters of the ID-PKC system are not required to be transferred securely, but they are required to be delivered without any change as the integrity of public parameters is vital to the correct use of an ID-PKC system. The most common usage of ID-PKC lies in securing e-mail communications within an enterprise [b-Voltage], as initialization of an ID-PKC system within a domain is relatively easier than cross-domain. Users within a domain can obtain its private key and the public parameters of the ID-PKC system securely, such as via offline methods. In the [b-IETF RFC 5408] scheme, the secure delivery of a user's private key and the public parameters of an ID-PKC system is achieved using TLS. Although this scheme can work for both intra- and cross-domain use cases, it requires the certificates to establish a secure channel between the user and KGC. This implies that the bootstrap of an ID-PKC system relies on PKI meaning that the multi-CA trust issue in the PKI-PKC system is transmitted to the ID-PKC system.

There is a need to study how to realize secure cross-domain communications based on ID-PKC without relying on PKI. The secure bootstrap of an ID-PKC system within a domain can be done without using TLS as this can be realized by using offline methods, such as by delivering USIM cards or TransFlash cards containing the user's private key and the public parameters of the ID-PKC to a dedicated user. Once the bootstrap of an ID-PKC within a domain is completed, a user can acquire its private key and public parameters associated with the domain's KGC. For secure cross-domain communication, a user has to acquire the genuine public parameters of another domain's KGC, as the communication partner is located in a different domain. As an alternative solution to the PKI-based scheme, some decentralized technologies, such as blockchain, can provide an immutable storage for the public parameters of KGCs and a user could retrieve genuine public parameters for any KGC from the blockchain without relying on TLS.

Revocation is required in an ID-PKC system to prevent the continued use of an identity or credential that is no longer valid or has a security breach such as discontinued service or compromised private key. If an identity is revoked, the identity shall be set in the revoked status. The revoked identities form the identity revocation list (IRL) which requires an authentic channel for delivery to a user. TLS is used to establish a secure channel between a user and KGC to deliver IRL in the [b-ITU-T X.1365] scheme.

The identity revoke information provided by KGC may face single point of failure attacks, thus the inquiry efficiency needs to be enhanced. Analogous to the delivery of public parameters of KGCs, the publication and retrieval of the IRL can be achieved using blockchain-based technologies without resorting to PKI.

7 Introduction to PKI-based bootstrap mechanism for an ID-PKC system

7.1 PKI-based public-key cryptosystem

7.1.1 Public-key certificate

In a PKI-based public-key cryptosystem, public and private key pairs are generated randomly without any association to a user's identity. The authenticity of the public key has to be assured in the PKI-PKC system otherwise it is at risk of man-in-the-middle attacks. To assure the integrity of the public key and bind it with a user's identity, certificates (usually X.509 certificates) are deployed in the PKI-PKC system. Certificates are signed by a trusted third party (TTP) called a certificate authority (CA) to prove their authenticity. It is usually assumed that the public key of the TTP is widely available. A user can verify the signature of a certificate by using the public key of the TTP so that they can confirm the validity of the certificate. Public/private-key pairs can be generated either by the user themselves or by the CA. In the case of key pair generation by the user, an authentic channel is needed to deliver the self-signed certificate of CA to the user whereas, in the case of key pair generation by the CA, a secure channel is required to deliver the private key and self-signed certificate of the CA to the user.

7.1.2 Public-key infrastructure (PKI)

The X.509 certificate management relies on the existence of a PKI which has a centralized architecture. The main purpose of a PKI is to manage public-key certificates and to make them widely available for a community of users in an application using asymmetric cryptography. The functions of a PKI primarily include the creation, revocation, storage and archival of public-key certificates. The main components of a PKI are shown in Figure 1.

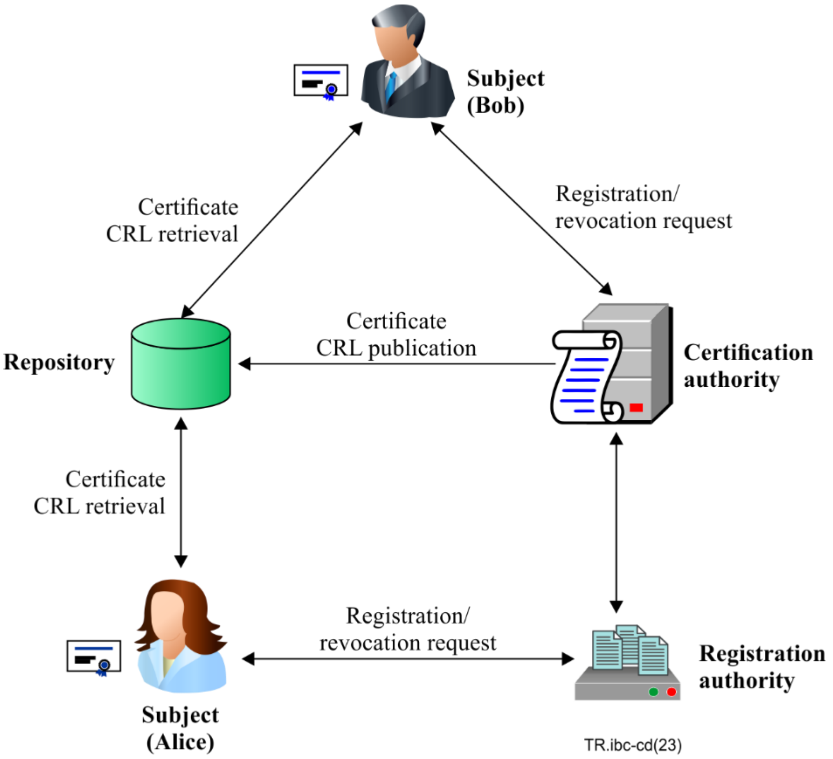

Figure 1 – Components of a PKI

A PKI consists of the following components:

- The **certificate authority (CA)** is the issuer of certificates, and the certificate revocation lists (CRLs).
- The **registration authority (RA)** is an optional component used to undertake some administrative functions from the CA, mainly associated with the subject registration process.
- The **repository** is the directory to store X.509 certificates and CRLs, and to make them publicly available.
- **Subjects** are certificate holders.

7.1.3 PKI interoperability

No global PKI is available to manage the certificates of all mobile network operators (MNOs) in the world. Usually, each MNO has its own PKI for certificate management within its domain. For inter-domain communications, PKI interoperability becomes an issue that needs to be addressed. There are three typical schemes for PKI interoperability, i.e., the cross-certification model, bridge CA model and certificate trust list model.

7.1.3.1 The cross-certification model

To make secure communication between two MNOs possible, the trust relationship between them has to be established. For this, two root CAs of MNOs issue certificates to each other, i.e., mutual cross-certification, as shown in Figure 2.

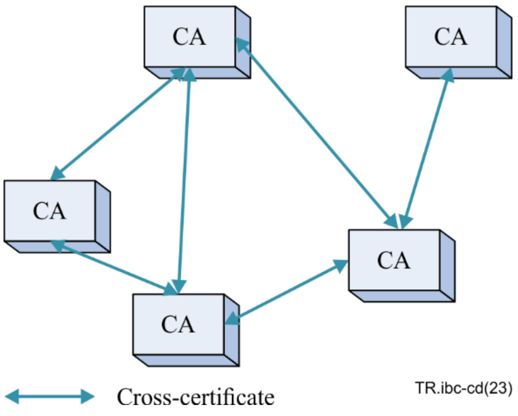

Figure 2 – Cross-certificate model

Based on the cross-certificate, a user can verify a certificate issued by an unknown CA back to a local trusted CA. If every pair of CAs mutually certifies, the number of cross-certificates required is $n*(n-1)/2$, where n is the number of CAs. Thus, the cross-certificate approach is suited for a small group CAs to cross certify, but not well suited for a large group of CAs to interoperate with each other.

7.1.3.2 Bridge CA model

In this model, a new CA called a bridge CA, which is trusted by other CAs, is introduced. It establishes the trust relationship with each CA in a way that the bridge CA cross-certifies with each CA, as shown in Figure 3.

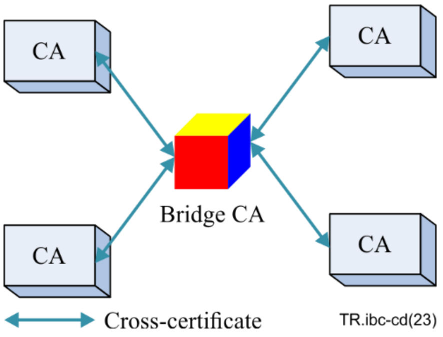

Figure 3 – Bridge CA model

Based on the certificate of the bridge CA and the cross-certificate, a user can verify a certificate issued by an unknown CA back to a local trusted CA. This scheme is well suited for a large group of CAs to interoperate with each other, as it only requires one pair of cross-certifications for each CA. The bridge CA becomes the single point of failure as it takes full responsibility for establishing trust among CAs. Thus, due to great liability, it is rare for entities to act as bridge CA. In practice, only a few bridge CAs are deployed, such as the Federal Bridge Certification Authority (FBCA) [b-fbca].

7.1.3.3 Certificate trust list model

A list of trusted root CAs' certificates forms the certificate trust list. This model is mainly used in web browser applications. The trusted root CAs are pre-installed in a browser. A root CA can either directly issue end entity certificate or issue the certificate to an intermediate CA (ICA), which in turn issues end entity certificates, as shown in Figure 4.

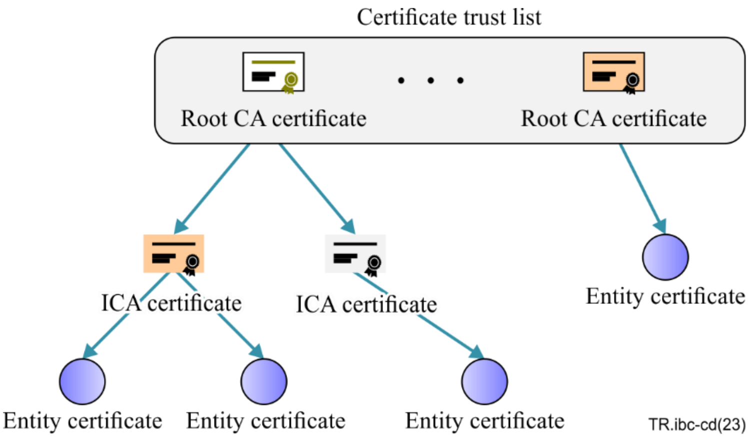

Figure 4 – Certificate trust list model

Based on the certificate trust list, a user can verify a certificate whose trust anchor (root CA) is in the certificate trust list. This model provides flexibility in the sense that a user can manually add a CA that they believe to be trusted to the certificate trust list in the browser. However, it is not easy to achieve such flexibility in mobile communication systems, especially in IoT communications where IoT devices are usually not managed manually.

7.2 Identity-based public-key cryptosystem

Certificate management, including the creation, revocation, storage and archival of public-key certificates, makes the PKI-PKC system rather complex. Moreover, PKI interoperability is a crucial issue that has not yet been well addressed. In contrast, in an ID-PKC system, a public key is derived from the system public parameters and a user's public identity information, such as phone number and email address. This eliminates the need for the certificate to bind the user's identity and its public key. The private key is generated by the KGC which is composed of a public parameter server (PPS) and a private-key generator (PKG).

- **Private-key generator (PKG):** a PKG generates a user's private key based on the secure stored master secret of an ID-PKC system and user's identity. The private key is distributed to a user via a secure channel which provides confidentiality and integrity protection. As such, the private key is only known to the user with the associated identity.
- **Public parameter server (PPS):** PPS provides the public parameters of an ID-PKC system and policy information that describes the operation of a PKG to users. The communication channel between a user and PPS shall be authentic as the integrity of the public parameters and policy information is vital to the normal operation of an ID-PKC system. The communication channel between a user and PPS is not necessarily confidentiality protected as the public parameters and policy information are public information that can be fetched by anyone.

7.3 Initialization of ID-PKC system based on PKI

To utilize an ID-PKC system, a user has to retrieve their private key from the PKG and their public parameters from the PPS. A method for the initialization of an ID-PKC system based on PKI was proposed in [b-IETF RFC 5408], as shown in Figure 5.

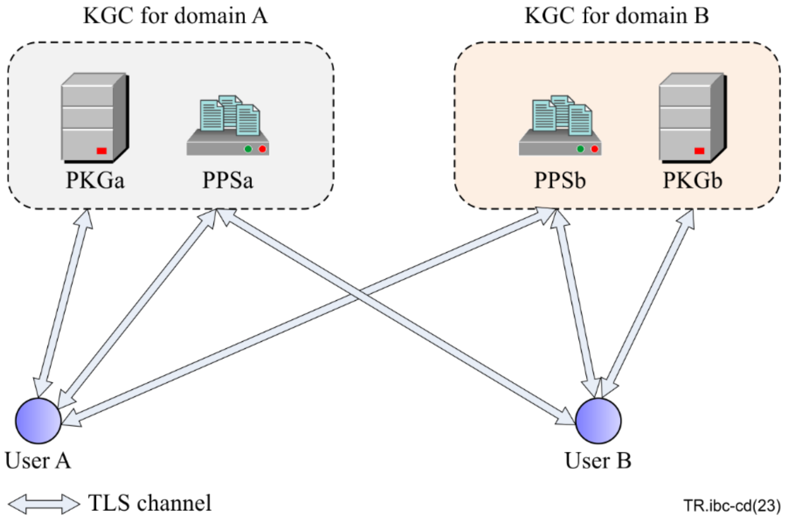

Figure 5 – PKI-based initialization of the ID-PKC system

Within a domain, a user fetches their private key and the public parameters of the system via HTTPS which runs on the top of TLS secure channel. To establish the TLS channel, the user needs to know the server certificate of a domain and can verify the authenticity of the server certificate. This requires a PKI to be available for this, meaning that the complexity of the certificate management remains in the ID-PKC system. Thus, the PKI-based initialization approach is not in line with the original design goal of the ID-PKC system that removes the need for certificate management.

To verify a signature during inter-domain communication, a receiver in one domain has to acquire the public parameters of the system in the other domain to which the sender belongs. To encrypt a message for inter-domain communication, a sender in one domain has to obtain the public parameters of the system in the other domain to which the receiver belongs. A TLS secure channel is used to protect the confidentiality and integrity of the public parameters of the system. This means that the ID-PKC system has to deal with the PKI interoperability issue. It is worth noting that the approach in [b-IETF RFC 5408] using TLS channel surpasses the requirement of delivery of public parameters of the ID-PKC system as only the integrity of the public parameters of the system is required to be protected.

A security methodology for the use of identity-based cryptography technology in support of IoT services over telecommunications networks is proposed in [b-ITU-T X.1365] where the delivery of public parameters and the identity revocation list (IRL) relies on PKI. The issue of trust between multi-CA arises when it is used in cross-domain communication. Clause 8 will focus on the distribution of public parameters and IRL without PKI.

8 Authentic distribution of public parameters and identity revocation list

8.1 Brief introduction to DLT system

A DLT system can create and maintain an immutable ledger to record transactions among untrusted or semi-trusted participants using cryptographic algorithms and consensus mechanisms. DLT systems can be classified into two categories depending on the method a participant is granted access to the system i.e., permissionless and permissioned DLT system [b-NISTIR 8202].

In a permissionless DLT system e.g., [b-Bitcoin] and [b-Ethereum] anyone can access the system without authorization where participants are untrusted. In contrast, in a permissioned DLT system, e.g., [b-Hyperledger], participants can access the system only after they have been proved to have the right to enter the system, where participants do not fully trust each other (semi-trusted).

NOTE – In this report, the term "distributed ledger" is used instead of the term "blockchain". Both terms in this report have the same meaning.

8.2 Identity management server (IMS)

Identity management plays a key role in an ID-PKC system as identity serves as the public key. This function is missing in the system specified in [b-IETF RFC 5408]. Besides PKG and PPS, IMS is added to the KGC to take responsibility for managing user identity, including ensuring the uniqueness of user identity in the management domain, maintaining identity status (valid, revoked), and publishing the IRL. A user identity may be revoked due to various reasons, such as key compromise or change of affiliation.

The communication channel between the user and IMS should be authentic but not necessarily confidential as the IRL is public information that anyone can obtain.

8.3 DLT based approach for authentic distribution of public parameters and IRL

8.3.1 System architecture

The permissioned DLT system is selected as the basis for authentic distribution of public parameters and IRL, since only stakeholders (e.g., mobile operators that have business relationships) are involved in the running of the DLT system. Multiple pre-selected nodes are designated as accounting nodes in the permissioned DLT system. The generation of each block is jointly determined by all accounting nodes using the consensus mechanism. Other access nodes can read the information on the ledger, but do not touch the accounting process. The permissioned DLT system exploits permissioned distributed ledger and distributed consensus technology to form a distributed immutable database. As long as the information is released on the permissioned distributed ledger, it is authentic and immutable.

A number of KGCs which are composed of PKG, PPS and IMS, together with users, form a permissioned DLT system so that only authorized PPS and IMS can operate in the permissioned DLT system. After the consensus process, ID-PKC system public parameters of a domain are written on the permissioned distributed ledger by a PPS, which at a minimujm include the domain name, distributed ledger name, status of system public parameter, hash algorithm for hiding user identities, public parameter server name and identity management server name. The detailed specification of system public parameters is described in Annex A. After a consensus process, an IMS writes the IRL of its domain on the permissioned DLT system, which at a minimum contains the domain name, distributed ledger name, revocation identity set and identity management server name. The detailed specification of a revocation identity list refers to Annex B. The user terminal cannot write data on the permissioned distributed ledger but can only read data from the permissioned distributed ledger. The architecture of a permissioned DLT system for authentic distribution of public parameters and IRL is illustrated in Figure 6.

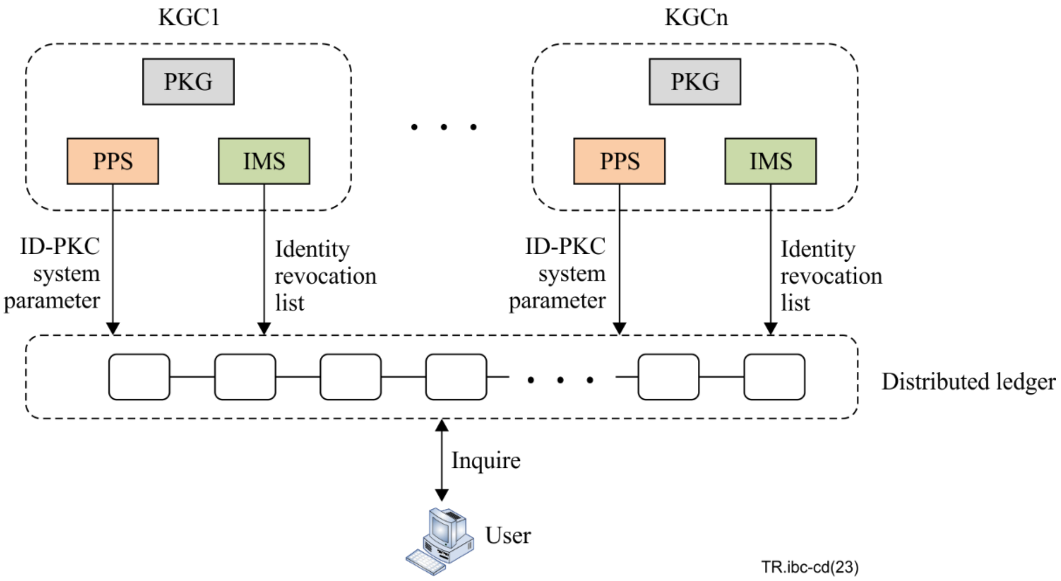

Figure 6 – System architecture

8.3.2 Management of the system public parameters

8.3.2.1 Process to write the system public parameters on the permissioned distributed ledger

The detailed steps to write ID-PKC system public parameters on the permissioned distributed ledger are as follows:

1) PPS of a domain generates ID-PKC system public parameters and marks their status as valid.

2) One or more accounting nodes of PPS in one domain together with the accounting nodes of other domains in the permissioned distributed ledger apply the consensus mechanism to write the ID-PKC system public parameters into the permissioned distributed ledger.

8.3.2.2 Process to update the system public parameters on the permissioned distributed ledger

The information of ID-PKC system parameters may need to be updated. There are various reasons for this, such as the cryptographic algorithm used in the system has to be changed or the master key may have been compromised. Since the message on the distributed ledger cannot be deleted, it is necessary to generate an ID-PKC system parameter that is the same as the original ID-PKC system parameter with the exception that its status is marked as invalid, and to put it on the permissioned distributed ledger. After that, the PPS generates an ID-PKC system parameter whose content has been updated, marks its status as valid and writes it on the permissioned distributed ledger, so as to complete the updating of ID-PKC system public parameter information. The concrete steps are as follows:

1) The PPS generates an ID-PKC system public parameter with the same content as the ID-PKC system parameter on the permissioned distributed ledger (except for the `id-pkcParamStatus` field, which is specified in Annex A), and marks its status as invalid.

2) The PPS writes the newly generated ID-PKC system public parameter into the permissioned distributed ledger by using the consensus mechanism with one or more accounting nodes of the domain and accounting nodes of other domains in the distributed ledger.

3) The PPS generates an ID-PKC system public parameter whose information content has been updated and marks its status as valid.

4) One or more accounting nodes of the PPS in the domain and accounting nodes of other domains apply the consensus mechanism to write the updated ID-PKC system public parameter into the permissioned distributed ledger.

8.3.2.3 Inquiring system public parameters on the permissioned distributed ledger

The detailed steps are as follows:

1) To obtain the ID-PKC system public parameter, the user first initiates a query to the permissioned distributed ledger according to the `domainName` field and / or `ppsName` field, and `distributedLedgerName` field, which is specified in Annex A.

2) The query starts from the latest block on the permissioned distributed ledger. If the `domainName` and/or `ppsName` to be queried are not found on the ledger, the query is terminated and the error message (i.e., *the ID-PKC system parameter does not exist*) is returned. If it exists, the obtained ID-PKC system public parameter whose corresponding domain serial is the largest is checked: if its status is invalid, it returns the error message (*ID-PKC has system public parameter but the status is invalid*); if the status of ID-PKC system public parameter is valid, the ID-PKC system public parameter that the user wants is returned.

8.3.3 Management of identity revocation list

8.3.3.1 Process to write the identity revocation list on the permissioned distributed ledger

The detailed steps to write the IRL on the permissioned distributed ledger are as follows:

1) The IMS of a domain generates the IRL and marks its status as valid. If the revoked identity needs to be anonymous, the hash value of the revoked identity is set to `identity field,` which is calculated by using the hash algorithm indicated in the system public parameters, referring to Annex B.

2) One or more accounting nodes of the IMS in one domain together with the accounting nodes of other domains in the permissioned distributed ledger apply the consensus mechanism to write the IRL into the distributed ledger.

8.3.3.2 Inquiring the identity revocation list on the permissioned distributed ledger

The detailed steps are as follows:

1) The user applies the identity to be checked to inquire the permissioned distributed ledger. If it is found, the identity has been revoked and a message that the *identity has been revoked* is returned to the user. If it is not found, the second step is carried out, as the identity on the permissioned distributed ledger may be anonymous.

2) The user exploits the hash function indicated in the ID-PKC system public parameter to calculate the identity to be queried and applies the calculation result to inquire the ledger. If there is the same value on the permissioned distributed ledger, the identity has been revoked, and a message *the identity has been revoked* is returned to the user; if the same value is not found on the permissioned distributed ledger, the identity can be determined to be valid, and a message that *the identity is valid* is returned to the user. In this way, the validity of the anonymous identity is checked on the permissioned distributed ledger.

9 Evaluation and way forward

In this report, a distributed ledger-based scheme is proposed to authentically distribute public parameters and the IRL in the ID-PKC system.

With the proposed ID-PKC system, a user can acquire the public parameters and IRL intra- or inter-domain without using PKI. Therefore, the proposed scheme has a great advantage over the existing solution specified in [b-IETF RFC 5408] which works relying on PKI, as the complex certificate management is eliminated in the proposed ID-PKC system. In this way, the proposed ID-PKC system can achieve the original design objective of an ID-PKC system.

The detailed comparison between the proposed ID-PKC system and the [b-IETF RFC 5408] scheme is illustrated in Table 1.

Table 1 – Comparison between the proposed ID-PKC system and the [b-IETF RFC 5408] scheme

Features	Proposed system	Scheme in [b-IETF RFC 5408]
Relying on the PKI	No	Yes
Certificate management	Does not need	Need
TLS channel	Does not need	Need
Identity revocation list	Support	Does not support
Retrieve the public system parameters of another domain	Relatively easy	Relative difficult
Single point of failure	No	Yes

The proposed ID-PKC system can operate not only intra- but also inter-domain. This means that various organizations (or companies) may be involved to run the proposed ID-PKC system. Thus, there is a strong need to standardize the proposed system to realize cross-domain secure communications.

Annex A

ID-PKC system public parameters in ASN.1

(This annex forms an integral part of this Technical Report.)

This annex includes all the ASN.1 type, value and information object definitions for ID-PKC system public parameter and constitute a formal ASN.1 module.

```
ID-PKCSysParams ::= SEQUENCE {
      version              INTEGER {V1(1)},
      domainName                OCTET STRING,
      ppsName              OCTET STRING OPTIONAL,
     imsName               OCTET STRING OPTIONAL,
      domainSerial              INTEGER,
      validity             ValidityPeriod,
      id-pkcPublicParameters    ID-PKCPublicParameters,
      id-pkcIdentityType        OBJECT IDENTIFIER,
      distributedLedgerName     OCTET STRING,
     hashAlgorithm              AlgorithmIdentifier
     id-pkcParamStatus          ID-PKCParamStatus,
      id-pkcParamExtensions[0]  IMPLICIT ID-PKCParamExtensions OPTIONAL,
}
```

- `version`: it is the version number of the ID-PKC system public parameter.
- `domainName`: it is the name of the domain where the KGC is located. It can be the name defined according to URI or URL for KGC addressing, or the users defined name in their own way.
- `ppsName`: as an option, it is the name of the PPS. It can be the name defined by using URI or URL for PPS addressing, or the user's name defined in their own way.
- `imsName`: as an option, it is the name of the IMS. It can be the name defined by using URI or URL for IMS addressing, or the user's name defined in their own way.
- `domainSerial`: this field is an integer representing a unique set of ID-PKC system parameters that can be used on the `domainName`. It is thus increased by one for each updating the system public parameter.
- `validity`: this field defines the lifetime of the ID-PKC system parameter and is defined as the following:

```
 ValidityPeriod ::= SEQUENCE {
   notBefore  GeneralizedTime,
   notAfter  GeneralizedTime
 }
```

- `id-pkcPublicParameters`: this is a structure that contains the public parameters corresponding to the ID-PKC algorithms supported by the KGC. The structure is defined as follows:

```
 ID-PKCPublicParameters ::= SEQUENCE (1..MAX) OF ID-PKCPublicParameter
 ID-PKCPublicParameter ::= SEQUENCE {
   id-pkcAlgorithm            OBJECT IDENTIFIER,
   publicParameterData        OCTET STRING

 }
```

- `id-pkcAlgorithm`: this describes the ID-PKC algorithms supported by the KGC of a system.

- publicParameterData: this is a DER encoded structure that contains actual cryptographic parameters. Its specific structure depends on the algorithm.
- id-pkcIdentityType: this is used to define the identity of the type using identity in a domain. How this field is used depends on the application.
- distributedLedgerName: ID-PKC system parameters are published on the distributed ledger. This field is used to indicate the name of the distributed ledger.
- hashAlgorithm: this field indicates the hash algorithm used to hide the user identity and to anonymize the identity in the IRL. It is defined as follows:

```
hashAlgorithm::= AlgorithmIdentifier,
  AlgorithmIdentifier::=CHOICE {
       SHA-256,
       SHA-3,
}
```

- id-pkcParamStatus: this is used to indicate the status of ID-PKC system public parameters. There are two statuses: valid or invalid. Its definition is as follows:

```
ID-PKCParamStatus ::= ENUMERATED {
  valid (0),
  invalid (1)
}
```

- id-pkcParamExtensions: this is a set of extensions that can be used to define other parameters that may be required for a specific implementation. This structure is defined as follows:

```
ID-PKCParamExtensions ::= SEQUENCE OF ID-PKCParamExtension
ID-PKCParamExtension ::= SEQUENCE {
  id-pkcParamExtensionOID     OBJECT IDENTIFIER,
  id-pkcParamExtensionValue   OCTET STRING
}
```

Annex B

Identity revocation list in ASN.1

(This annex forms an integral part of this Technical Report.)

This annex includes all the ASN.1 type, value and information object definitions for IRL and constitute a formal ASN.1 module.

```
IdentityRevocationList::= SEQUENCE {
      version                    INTEGER { v1(1) },
     issuer               Name,
     irlNumber                  INTEGER OPTIONAL,
     deltaList                  BOOLEAN OPTIONAL,
      domainName                      OCTET STRING,
      domainSerial                    INTEGER,
     imsName                    OCTET STRING OPTIONAL,
      thisUpdate                      Time,
      nextUpdate                      Time OPTIONAL,
     distributedLedgerName           OCTET STRING
      revokedIdentities               SEQUENCE OF SEQUENCE {
         anonymity                    Anonymity,
           identity                        ID-PKCIdentityInfo,
           revokeReason                    RevokeReason
         revocationDate                    Time,
         irlEntryExtensions           Extensions OPTIONAL
         }
      }
```

- version: this is the version number of the IRL.
- issuer: this is used to distinguish the publisher of the IRL.
- irlNumber: this is the publisher number of the current IRL. It starts at 0. For each full IRL update, the number is increased by 1. It is optional.
- deltaList: this indicates whether the current IRL is an incremental IRL. This list contains only identity information that has been revoked since the full IRL of the irlNumber index was published.
- domainName: this is the name of the domain of the KGC that generates the IRL. It can be the name defined by URI or URL for KGC addressing, or the users defined name in their own way.
- domainSerial: this field is an integer representing the collection of unique IRLs that can be used on domainName.
- imsName: this is the name of IMS. It can be a name defined by URI or URL for IMS addressing, or the users defined name in their own way.
- thisUpdate: this indicates when this IRL was generated.
- nextUpdate: this indicates the generation time of the next IRL, it is optional.
- distributedLedgerName: the user identity revocation list is published on the distributed ledger. This field is used to indicate the name of the distributed ledger.
- revokedIdentities: this is used to indicate the revocation identity set, including the following fields: anonymity, identity, revokeReason, revocationDate, irlEntryeXtensions. These fields are described as follows:

- anonymity: it is used to indicate whether the revocation identity needs to be anonymous. Its description is as follows:

```
anonymity::= Anonymity
  Anonymity::=CHOICE{
        YES,
   NO
  }
```

- identity: it is used to describe the revocation identity, which is described as follows:

```
identity::= ID-PKCIdentityInfo
   ID-PKCIdentityInfo::=CHOICE{
         Hash (RovokedIdentity),
         RovokedIdentity
   }
```

If anonymity is YES, the ID-PKCIdentityInfo field corresponds to the hash value of the RovokedIdentity; otherwise, the ID-PKCIdentityInfo field corresponds to the RovokedIdentity itself.

- revocationDate: it indicates when this identity is revoked.

- revokeReason: it is used to describe the reason for identity revocation. Its description is as follows:

```
revokeReason ::=RevokeReason
   RevokeReason ::= ENUMERATED {
unspecified (0),
keyCompromise     (1),
kgcCompromise     (2),
affiliationChanged (3),
superseded (4),
cessationOfOperation (5)
  }
```

- irlEntryExtensions: this defines possible revocation identity extensions.

Bibliography

[b-ITU-T F.751.0] Recommendation ITU-T F.751.0 (2020), *Requirements for distributed ledger systems*.

[b-ITU-T X.509] Recommendation ITU-T X.509 (2019) | ISO/IEC 9594-8:2020, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

[b-ITU-T X.1365] Recommendation ITU-T X.1365 (2020), *Security methodology for the use of identity-based cryptography in support of Internet of things (IoT) services over telecommunication networks*.

[b-ITU-T X.1400] Recommendation ITU-T X.1400 (2020), *Terms and definitions For distributed ledger technology*.

[b-ISO/TC 307] ISO/TC 307: *Blockchain and distributed ledger technologies*.

[b-NISTIR 8202] NISTIR 8202:2018/10, *Blockchain technology overview*.

[b-IETF RFC 5408] IETF RFC 5408 (2009), *Identity-based encryption architecture and supporting data structures*.

[b-Bitcoin] Bitcoin Project (2023), *Open Source P2P Money*. https://bitcoin.org

[b-Boneh] Boneh, D. and Franklin, M. (2001), *Identity-Based Encryption from the Weil Pairing*, Advances in Cryptology – *CRYPTO*, pp. 213–229.

[b-Diffie] Diffie, W. and Hellman, M.E. (1976), *New Directions in Cryptography*, IEEE Transactions on Information Theory, Vol. 22, No. 6, pp. 644–654.

[b-Ethereum] ethereum.org (2023) *Welcome to Ethereum*. https://ethereum.org

[b-fbca] Computer Security Resource Center (2002) *Federal Bridge Certification Authority*. https://csrc.nist.rip/pki/fbca

[b-Hyperledger] Hyperledger (2022), *Hyperledger Fabric*. https://www.hyperledger.org/use/fabric

[b-Shamir] Shamir, A. (1984), *Identity-based Cryptosystems and Signature Schemes*, CRYPTO 1984 – Advances in Cryptology, pp. 47–53.

[b-Voltage] Voltage Secure Mail, *Achieving End-to-End Email Security Without Impacting the User Experience*. https://www.microfocus.com/en-us/cyberres/data-privacy-protection/secure-mail