International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1450

(10/2018)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Available – Security protocols (2)

Guidelines on hybrid authentication and key management mechanisms in the client-server model

Recommendation ITU-T X.1450

International Telecommunication Union

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES (1)	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols (1)	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES (2)	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1319
Smart grid security	X.1330–X.1339
Certified mail	X.1340–X.1349
Internet of things (IoT) security	X.1360–X.1369
Intelligent transportation system (ITS) security	X.1370–X.1389
Distributed ledger technology security	X.1400–X.1429
Distributed ledger technology security	X.1430–X.1449
Security protocols (2)	**X.1450–X.1459**
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1450

Guidelines on hybrid authentication and key management mechanisms in the client-server model

Summary

Client and server are often asymmetric regarding security credential management. Since in most cases there are many clients and a few servers, server credentials are distributed and managed with relatively low cost, but client credentials are apparently not. As most mobile services increasingly communicate security and privacy sensitive data, industry need to provide secure channel in client-server model using secure yet cost-effective methods addressing such asymmetric security requirements.

Passwords could be effective in terms of client credential management, and guidelines such as [ITU-T X.1151] are available for password-authenticated key exchange protocols. When client credentials are compromised, however, the adversary could impersonate not only clients but also service providers. Such server impersonation attacks could be mitigated by using public key techniques for server authentication with low credential management cost.

Recommendation ITU-T X.1450 provides guidelines for hybrid authentication and key exchange mechanisms in the client-server model. The underlying mechanism suggests the use of shared secrets and public key techniques for authentication and key exchange. This Recommendation covers service scenarios, and security threats and methods to mitigate such attacks.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1450	2018-10-14	17	11.1002/1000/13729

Keywords

Identity-based schemes, key exchange, password-based authentication, PKI.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

© ITU 2019

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

Recommendation ITU-T X.1450

Guidelines on hybrid authentication and key management mechanisms in the client-server model

1 Scope

This Recommendation describes hybrid authentication and key management mechanisms in the client-server model. It analyses the typical service scenarios, security threats and attack methods, and provides technical methods to mitigate these risks.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.509] Recommendation ITU-T X.509 (2016) | ISO/IEC 9594-8:2017, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks.*

[ITU-T X.1151] Recommendation ITU-T X.1151 (2007), *Guideline on secure password-based authentication protocol with key exchange.*

[ITU-T X.1158] Recommendation ITU-T X.1158 (2014), *Multi-factor authentication mechanisms using a mobile device.*

3 Terms and definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 authentication [b-ITU-T Q.1743]: A property by which the correct identity of an entity or party is established with a required assurance. The party being authenticated could be a user, subscriber, home environment or serving network.

3.1.2 authentication factor [b-ITU-T X.1154]: A type of credential; there are three types of authentication factors: ownership factor, knowledge factor and biometric factor.

3.1.3 authentication protocol [b-ITU-T X.1254]: A defined sequence of messages between an entity and a verifier that enables the verifier to perform authentication of an entity.

3.1.4 certification authority (CA) [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.

3.1.5 credential [b-ITU-T X.1252]: A set of data presented as evidence of a claimed identity and/or entitlements.

3.1.6 digital signature [b-ITU-T X.843]: A cryptographic transformation of a data unit that allows a recipient of the data unit to prove the origin and integrity of the data unit and protect the sender and

the recipient of the data unit against forgery by third parties, and the sender against manipulated forgery by the recipient.

3.1.7 hybrid authentication [b-ISO/IEC 25185-1]and [b-CCHK15]: A type of authentication using symmetric (or weak secret) and asymmetric authentication systems.

3.1.8 mobile device [ITU-T X.1158]: A small, hand-held computing device with a subscriber identity module (SIM) card, typically having a display screen with touch input and/or a miniature keyboard and is not heavy.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 hybrid authentication and key exchange (HAKE): A type of authenticated key agreement where a client and a server make use of hybrid authentication to negotiate and authenticate one or more shared secret keys.

3.2.2 identity-based cryptography: A type of public-key cryptography in which a publicly known string representing an individual or organization is used as a public key. The public string could be an email address, domain name, or an IP address (Definition based on [b-S84]).

3.2.3 identity-based signature scheme: A type of digital signature scheme in which a publicly known string representing an individual or organization is used as a public verification key to verify a signature (Definition based on [b-S84]).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CA	Certification Authority
DNS	Domain Name System
HAKE	Hybrid Authentication and Key Exchange
IBC	Identity-based Cryptosystem
IBE	Identity-based Encryption
IBS	Identity-based Signature
KE	Key Exchange
KGA	Key Generation Authority
MFA	Multi-Factor Authentication
MITM	Man-in-the-middle
OTP	One-Time Password
PA	Password-based Authentication
PAKE	Password-based Authentication and Key Exchange or Password-Authenticated Key Exchange
PKC	Public Key Cryptography
PKI	Public Key Infrastructure
RP	Relying Party

5 Conventions

None.

6 Overview

Client and server are often asymmetric regarding security credential management. Since in most cases there are many clients and a few servers, server credentials are distributed and managed with relatively low cost, but client credentials are apparently not. As most mobile services increasingly communicate security and privacy sensitive data, industry need to provide secure channel in the client-server model using secure yet cost-effective methods addressing such asymmetric security requirements. Passwords could be effective in terms of client credential management. Various password-based authentication and key exchange or password-authenticated key exchange (PAKE) protocols are standardized in [b-ITU-T X.1035], [b-IEEE P1363.2], [b-ISO/IEC 11770-4] and guidelines such as [ITU-T X.1151] are available for PAKE protocols.

A PAKE protocol makes use of a password, which is shared between a client and a server, as an authentication means. Server authentication is performed by giving a proof of possession of a password or its verifier[1]. After performing a PAKE, a client and a server authenticate each other and share a cryptographic key. In [ITU-T X.1151], a set of requirements for a secure password-based authentication (PA) and key exchange (KE) is presented. In addition, the Recommendation gives comparison of existing PAKE protocols.

A PAKE protocol can be extended to a protocol using multiple authentication factors (MFAs). It can make use of a combination of various authentication factors such as a password, a long random key such as a signing key and a decryption key, and a one-time password (OTP) and biometrics. In [ITU-T X.1158], a client authentication is described by using multiple authentication factors. However, server authentication is not clearly described in [ITU-T X.1158]. In other words, when authentication factors are compromised and revealed, the security against a server impersonation attack is not sufficiently considered.

When client credentials are compromised, however, the adversary could impersonate not only clients but also service providers in PAKE or MFA using shared secrets. Such server impersonation attacks could be mitigated by using public key techniques for server authentication with low credential management cost.

This Recommendation provides guidelines for hybrid authentication and key exchange (HAKE) in the client-server model. The underlying protocol suggests the use of shared secrets and public key techniques, i.e., identity-based cryptosystem (IBC) or public key cryptography (PKC) for authentication and key exchange. It will cover service scenarios, security threats and typical attack methods, and technical methods to mitigate such attacks. It significantly enhances key management mechanisms based only on shared secrets such as [b-ITU-T X.1035], [b-ITU-T X.1151], [b-ISO/IEC 11770-4], [b-IEEE P1363.2] by preventing server impersonation attacks.

7 Security threats

This clause describes major threats for HAKE protocols. It does not contain a complete set of potential threats.

---

[1] The verifier denotes the information computed from the password, which is used in the server to prove that a client knows the password [ITU-T X.1151]. It is different from the notion of 'verifier' which means an entity to verify and validate identity information [b-ITU-T X.1252].

7.1 Server impersonation

Server impersonation is an attack wherein an attacker is able to masquerade a server when a weak secret or its verifier is revealed. This attack allows the attacker to perform an action they would otherwise not be able to perform (e.g., gain access to an otherwise inaccessible asset). For some relative notions, i.e., spoofing and masquerading, refer to [ITU-T X.1158].

7.2 Other types of attacks

[ITU-T X.1151] and [ITU-T X.1158] provide details for other types of attacks such as key logging attacks, lost and stolen mobile devices, shoulder surfing, phishing, etc. The following are examples of some of the attacks:

– Password leakage

  1) Key logging attacks [ITU-T X.1158]: Key logging (more often referred to as keylogging or "keyloggers") is the action of recording (or logging) the key strokes on a keyboard, typically in a covert manner so that the person using the keyboard is unaware that their actions are being monitored and recoded. This is sometimes implemented through a software that is run on a mobile device.
  2) Lost or stolen mobile device [ITU-T X.1158]: Mobile devices are becoming a favourite target for theft since they contain credential information that may lead to financial gains. A malicious attacker may be able to steal a user's mobile device and attempt to log in to their various user accounts.
  3) Pharming [ITU-T X.1151]: Whereas phishing involves redirecting the website's traffic to another forged website, pharming attacks by compromising the domain name system (DNS) server. Specifically, a pharming attack replaces with fake addresses the correct IP addresses that correspond to a domain name in the DNS server; thus redirecting the user to a hacker's forged website when he/she is asked to enter the company's web address.
  4) Phishing [ITU-T X.1151]: The act of sending an e-mail to a user, falsely claiming to be an established legitimate enterprise in an attempt to deceive the user into disclosing private information that will be used for identity theft. The e-mail directs the user to a website where he/she is asked to update personal information such as passwords and credit card, social security, and bank account numbers, information that the legitimate organization already has. Note, however, that the website is bogus, set up only to steal the user's information.
  5) Shoulder surfing [ITU-T X.1158]: An attack wherein an attacker obtains all or part of a user's credentials by taking a brief look (typically, over the user's shoulder) at the information provided by the user during authentication.

– Dictionary attack [ITU-T X.1151]: An attack wherein an attacker collects a data base of commonly used words and passwords that can be encrypted using all possible salts and compares its database of encrypted terms against the encrypted passwords found in a password file on the system. If a match is found, the actual password is known, and access is gained. The dictionary attack can be grouped into two categories: online dictionary attack and offline dictionary attack.

  1) Online dictionary attack [ITU-T X.1151]: An attack wherein an attacker repeatedly attempts authentication with the server using guessed passwords until he or she succeeds. The online dictionary attack can be detected or prevented by counting the number of access failures.
  2) Offline dictionary attack [ITU-T X.1158]: An attack wherein secrets associated with credential generation are exposed using analytical methods outside of the authentication transaction. The attacker uses the captured packets to guess the password. Password cracking often relies upon brute force methods, such as the use of dictionary attacks.

With dictionary attacks, an attacker uses a program to iterate through all the words in a dictionary (or multiple dictionaries in different languages), computes the hash value for each word and checks the resultant hash value against a database. The use of rainbow tables is another password cracking method. Rainbow tables are pre-computed tables of clear text/hash value pairs. Rainbow tables are quicker than brute-force attacks because they use reduction functions to decrease the search space. Once generated or obtained, rainbow tables can be used repeatedly by an attacker.

– Man-in-the-middle (MITM) attack [ITU-T X.1151]: An attack wherein an attacker intercepts the public or cryptographic keys being exchanged by two entities and substitutes his/her own public key to impersonate the recipient. This successful attack results in the compromise of the cryptosystem or PAKE.
– Reply attack [ITU-T X.1158]: An attack wherein an attacker is able to replay previously captured messages (between a legitimate entity and a relying party (RP)) to authenticate as that entity to an RP.
– Denning-Sacco attack [ITU-T X.1151]: An attack wherein an attacker is able to find the shared password from the compromised session key of a previous session.
– Server-compromised attack [ITU-T X.1151]: An attack wherein an attacker compromising the password verification-related file from the server can impersonate the user without launching a dictionary attack on the password file or derive old session keys from such compromised password. It is related to client impersonation.

8 General model

8.1 Entities

The main entities in HAKE protocols consist of a client, a server, and a key generation authority (KGA) or a certification authority (CA).

8.1.1 Client and server

8.1.1.1 Client

Client denotes an entity to be authenticated by a server or a service provider through on-line services. A client is assumed to have a knowledge-based authentication factor, e.g., password or PIN. The client can use devices such as mobile devices or connected devices [ITU-T X.1158] to conduct a HAKE protocol. The client can provide security and privacy sensitive data to the server in order to receive the Internet application service from a server. The client has to authenticate the server by verifying authentication information, e.g., an identity-based signature (IBS) using a server's identity as a verification key.

8.1.1.2 Server

Server denotes an entity to be authenticated by a client. A server can provide Internet application services to a client after authenticating the client. The server has to provide the authentication information, for example, an identity-based signature generated from the private key corresponding to its identity. In addition, the server has to verify the authentication information transmitted from user's devices using a shared secret such as password or PIN.

8.1.2 Certification authority (CA) and key generation authority (KGA)

8.1.2.1 Certification authority (CA)

It denotes a typical authority for PKI [ITU-T X.509]. It is trusted by one or more users to create and assign public-key certificates. Optionally the certification authority may create the subjects' keys. CA

can exist as an independent and trusted authority. According to application services, CA and a server can be combined into a single entity.

8.1.2.2 Key generation authority (KGA)

Key generation authority denotes an entity trusted by one or more users to issue a private key corresponding to a public string to represent a server. The public string can be defined by a publicly known arbitrarily string representing a server that may be an individual or organization. For example, it could be a phone number, an e-mail address, domain name, service URL or an IP address. The private key is used for signing a message or decrypting an encrypted message. The public string is a kind of public key which can be used for verifying a signature or encrypting a message. KGA can exist as an independent and trusted authority such as CA of the public key infrastructure (PKI). According to application services, KGA and a server can be combined into a single entity.

8.2 Operational procedure of HAKE

In the HAKE protocol, authentication based on symmetric and asymmetric authentication systems is carried out. For symmetric authentication, a client and a server are assumed to share weak secret such as a password. It is assumed that the password is generated strongly according to password creation guidelines while it can be memorized by the client. For asymmetric authentication, a server is assumed to hold a private key and its corresponding public key for an IBC or PKI. Since the public key of an IBC can be a publicly known arbitrarily string as explained in clause 8.1.3, the public key of an IBC is simply defined by the identity of a server. One can access the public key.

Public protocol parameters include public parameters of PAKE and an IBC or PKI, such as the description of a mathematical group, pseudo-random number generators, and cryptographic one-way hash functions.

A client is assumed to use secure devices such as mobile devices or connected devices [ITU-T X.1158], to generate and transmit authentication information based on public protocol parameters, private password or the keys of an IBC or PKI. The client or the device held by the client is assumed to be connected to the server via a public network, e.g., the Internet, which may be vulnerable to various threats and attacks.

The HAKE protocol consists of two phases, enrolment, and authentication and session key establishment:

In the enrolment phase, a shared secret such as a password or password-related information is generated between a client and a server, and registered at the server. If necessary, a transformed form of the password can be stored using a salt and/or one-way hash function. In addition, a server's private key is generated. When using an IBC, KGA issues the private key for the server and a publicly known server's identity is defined as a server's public key. When using PKI, a CA issues a certificate for the server where its name and a server's public key are cryptographically bound via a CA's signature. In the above description, it is assumed that messages are transmitted via a secure channel.

In the authentication and key establishment phase, a client and a server exchange public protocol transcripts to generate a common key. The following steps describe how a HAKE protocol with password and an IBC or PKI generally works:

– The client enters a private weak secret such as password and a server's public identity.
– The client and the server exchange public transcripts defined for the HAKE protocol.
– The server authenticates the client using a shared secret such as a password or a hashed password, H(*pw*).
– The client authenticates the server using a shared secret such as a hashed password, H(*pw*) and the server's public identity or public key.

– The server and the client derive a common secret key that can be used as a cryptographic key for a session.

Enrolment, and authentication and key establishment are illustrated in Figure 1.

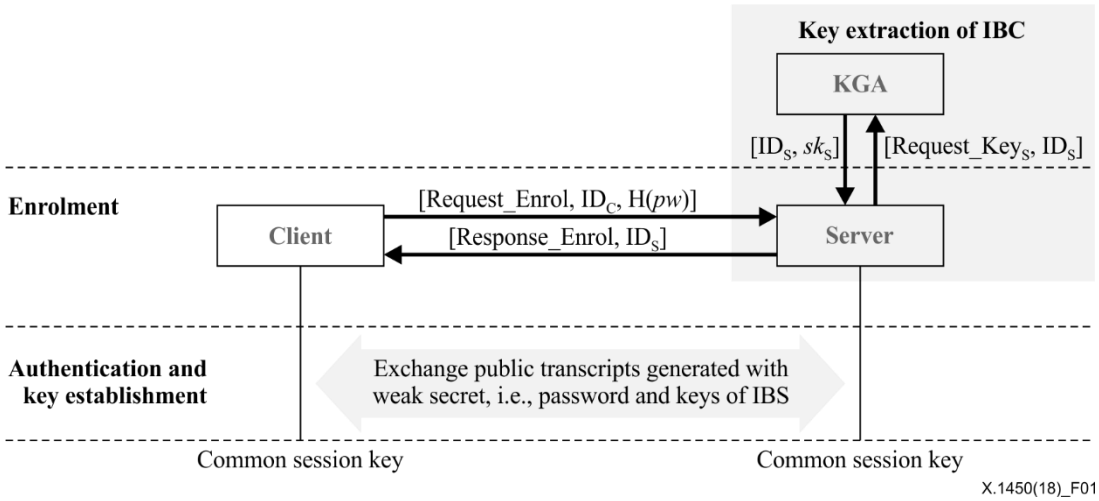

Figure 1 – General procedure of HAKE with an IBC

8.3 Requirements for HAKE

There are requirements that must apply to HAKE protocols to guarantee protection against the vulnerabilities described in clause 7.

– The protocol offers perfect forward secrecy ([ITU-T X.1151]).

– The protocol can guarantee the authenticity of the server ([ITU-T X.1151]).

– The protocol can guarantee the authenticity of the client ([ITU-T X.1151]).

– The protocol provides mutual authentication based on a pre-shared, human-memorable password, and an IBC or PKC.

– The protocol is required to be resistant to the threats described in clause 7.1.2.

– The protocol prevents any leakage of the information viewed during a successful run ([ITU-T X.1151]).

– The protocol supports a range of cryptographic algorithms including symmetric and asymmetric cryptographic algorithms, hash algorithms, and MAC algorithms.

– Client-initiated authentication information, i.e., password change must be supported.

– The protocol should be simple or easy to implement to promote widespread adoption and to minimize security flaw.

– The protocol requires minimal client configuration ([ITU-T X.1151]).

– The protocol requires minimum storage. A client is not required to hold a device to store a private key or password, which can be memorized.

– The identity of an entity in the multi-factor authentication mechanism is required to be established and managed by an enrolment procedure, which consists of four processes: application and initiation, identity proofing, identity verification, and record-keeping/ recording ([ITU-T X.1254]).

9 Protocols

9.1 Two types of HAKE

HAKE protocols can be classified into two types of protocols according to the structure of authentication as depicted in Figures 1 and 2. For convenience, they are called Type A and B. The figures show conceptual features in terms of authentication.

In Type A protocol, two-factor authentication [ITU-T X.1158] based on a knowledge factor such as a password and a possession factor such as a private key of an IBC or PKC is provided by a server. In contrast, in Type B protocol, a single factor authentication based on a possession factor such as a private key of an IBC or PKC is provided by a server.

9.1.1 Type A HAKE

In Type A, the protocol conducts mutual authentication using shared secrets such as passwords and additional authentication for preventing server impersonation attacks when shared secrets are compromised (ssee Figure 2).

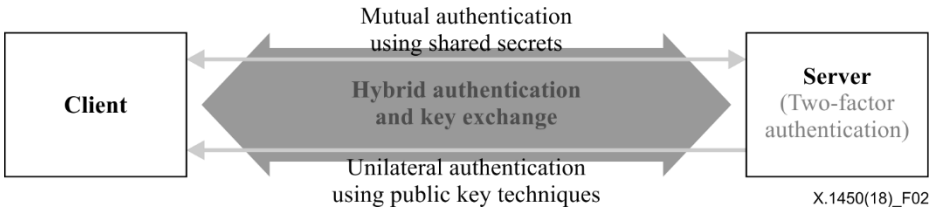

Figure 2 – Type A HAKE protocol

Type-A HAKE protocol can be constructed in a generic way by extending PAKE protocols with public key cryptography (PKC). For example, the PAKE protocol of [ITU-T X.1151] combined with an additional public key cryptosystem for server authentication. For clarification, instances of a Type-A protocol are presented in clauses 9.2 and 9.3.

9.1.2 Type B HAKE

In Type B, the protocol conducts a unilateral authentication for a client using a shared secret such as passwords and unilateral authentication for a server using public key techniques. In other words, a server authenticates a client using a shared secret such as passwords and a client authenticates a server using public key techniques (see Figure 3). A server does not use a shared secret for authentication therefore server impersonation attacks do not occur when shared secrets are compromised.

Figure 3 – Type B HAKE Protocol

9.2 HAKE protocols using shared secrets and identity-based cryptosystem

In the HAKE protocols, authentication is executed by a process of verifying that a client knows a weak secret (or its verifier) and a server knows the private key corresponding to the server's identity.

The following notations are used in the description of HAKE protocols:

– CT_C: ciphertext generated by a client.

– dk_S: private decryption key corresponding to the identity of a server.

– H: cryptographic hash function to give collision-resistance for input massages.
– ID_C: identity of a client.
– ID_S: identity of a serve.r
– m_S: messages generated by a server.
– m_C: messages generated by a client.
– *PP*: public parameters of a HAKE protocol. It includes PP_{PAKE} and PP_{IBS}.
– PP_{IBE}: public parameters of an IBE scheme.
– PP_{IBS}: public parameters of an IBS scheme.
– PP_{PAKE}: public parameters of a PAKE protocol.
– *pw*: password shared between a client and a server.
– sk_S: private signing key corresponding to the identity of a server.

9.2.1 Using shared secrets and an identity-based signature (IBS)

In the protocol, a two-party PAKE protocol and an IBS scheme are used as protocol components. Many IBS schemes can be constructed using various mathematical techniques. Refer to [b-S84], [b-BNN], [b-HKCJS15], [b-ISO/IEC 14888-3], and [b-ISO/IEC 29192-4] for more details.

The protocol consists of two phases, initialization and key establishment. A client and a server proceed as follows:

Initialization: Setup, extract, and enrolment are executed as follows:

– Setup: For a given security parameter λ, it generates public parameters, PP_{PAKE} for the given PAKE protocol, and a cryptographic hash function, H: $\{0,1\}^* \rightarrow \{0,1\}^L$. It runs the setup algorithm of the IBS scheme to generate (*msk*, PP_{IBS}) where *msk* is the master secret key and PP_{IBS} is public parameters for the IBS. The system public parameters, $PP = (PP_{PAKE}, PP_{IBS}, H)$ are made public. KGA keeps *msk* secret.
– Extract: For a given identity ID of a server, KGA runs the private key generation algorithm of the IBS, to output a private signing key, sk_{ID}. It is assumed that the private signing key is transmitted to a server with the identity ID via a secure channel.
– Enrolment: A client, *C* generates a password, *pw* according to a predefined password creation guideline [b-G17]. It is assumed that a secure channel is established between the client and the server, *S*. To register an application service, the client sends (Register-Enrol, ID_C, H(*pw*)) to the server via a secure channel.

Key establishment: a client, *C* and a server, *S* execute a run of a HAKE protocol to share a key for a session as follows (see also Figure 4):

– The client computes a hashed password, H(*pw*). Using it, the client performs its part in PAKE with the following modification: Whenever the client receives a pair of a message and a signature, (m_S, σ_S) from the server, the client verifies the signature, σ_S on m_S, that is, checks if the result of a signature verification, Vrfy(PP_{IBS}, ID_S, σ_S, m_S) is valid. If the signature is valid then the client performs its part of PAKE. The client then computes a common session key, *ssk*.
– The server performs its part in PAKE with the following modification: For each m_S to be sent to the client in PAKE, the server generates a signature, σ_S on m_S using Sign(PP_{IBS}, ID_S, sk_S, m_S) of the IBS, and then sends (m_S, σ_S). Finally, the server computes a common session key, *ssk*.

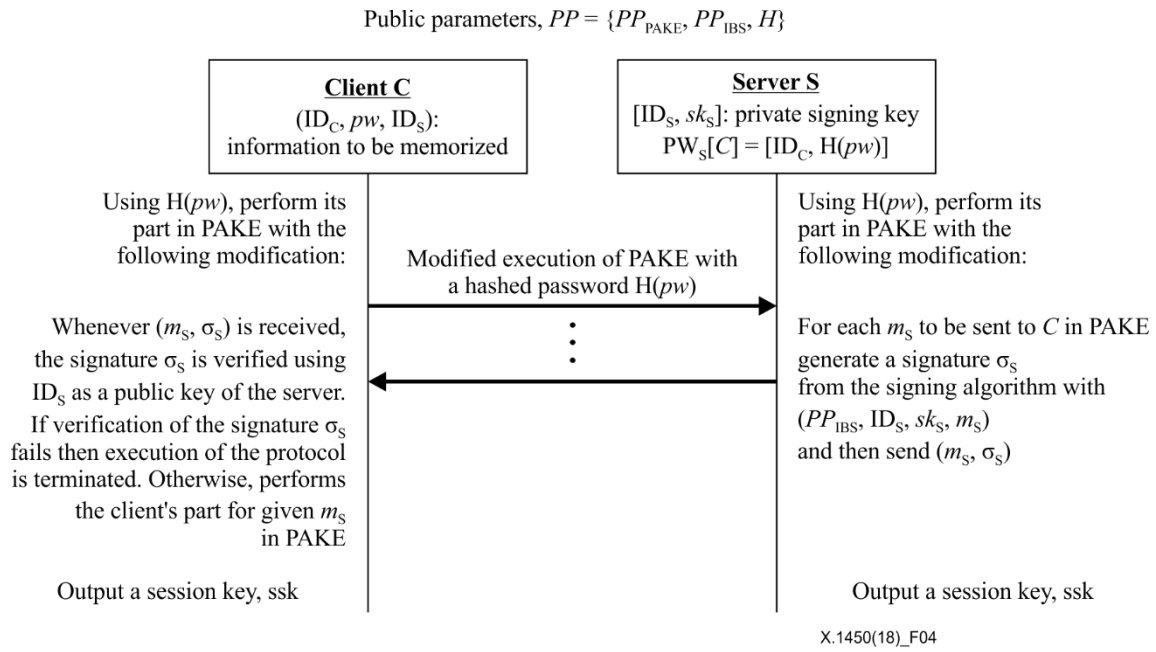

Figure 4 – Construction of HAKE using two-party PAKE and an IBS

Figure 5 is the result of the HAKE protocol from application to the PAKE protocol of [ITU-T X.1151], **[**b-ITU-T X.1035] and an IBS scheme. In the description, the following notation is used.

– 'A||B' is concatenation of two strings A and B

– G is a cyclic group of prime order q

– g and g_1 are random generators of G

– Z_q^* is a set of positive integers from 1 to q-1

– H and H_1 are cryptographic hash functions

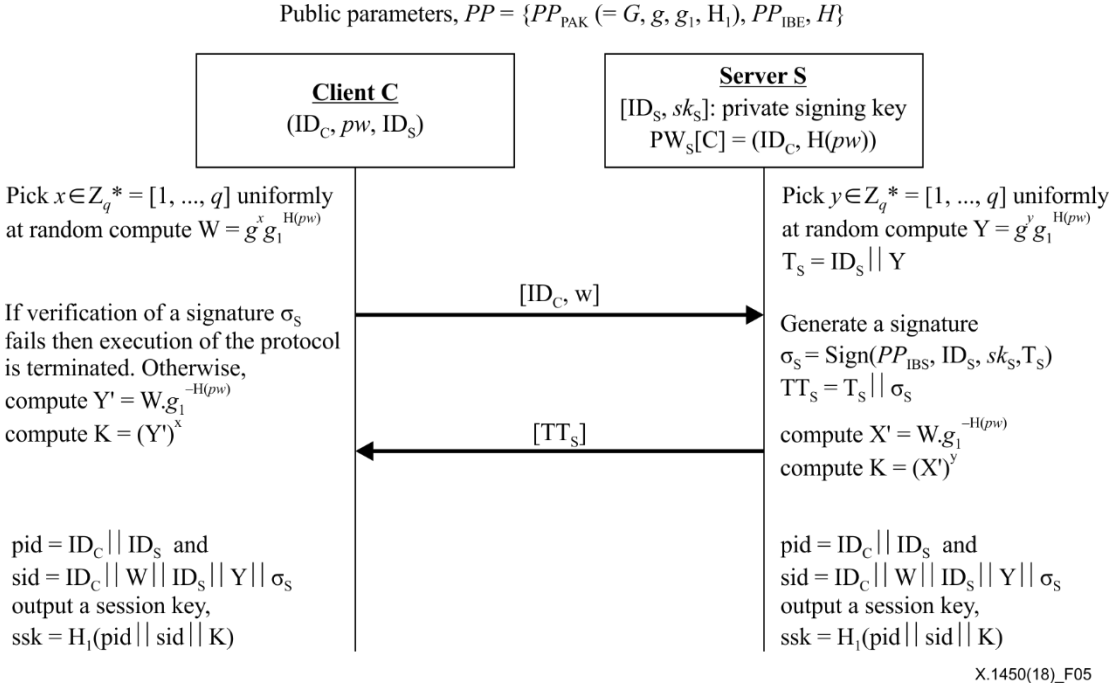

Figure 5 – HAKE based on PAK [b-ITU-T X.1035] and an IBS

Using different PAKE protocols similarly, various HAKE protocols can be constructed. There are many standardized PAKE protocols. For example, DH-EKE [b-IETF RFC 6124], SPEKE [b-ISO/IEC 11770-4], SRP [b-IETF RFC 2945], PAK [b-ITU-T X.1035], and AMP [b-ISO/IEC 11770-4]. For HAKE protocols based on the above PAKE protocols, security properties are analysed and compared. The result of the analysis and comparison is provided in Appendix I.

9.2.2 Using shared secrets and identity-based encryption (IBE)

In the protocol, a two-party PAKE protocol and an IBE scheme are used as protocol components. IBE schemes can be constructed using various mathematical techniques. Refer to [b-S84], [b-ISO/IEC 18033-5], and [b-BF01] for more details.

The protocol consists of two phases, initialization and key establishment. A client and a server proceed as follows.

Initialization: Setup, extract, and enrolment are executed as follows:

– Setup: For a given security parameter λ, it generates public parameters, PP_{PAKE} for the given PAKE protocol, and a cryptographic hash function, H: $\{0.1\}^{*} \rightarrow \{0.1\}^{L}$. It runs the setup algorithm of the IBE scheme to generate (msk, PP_{IBE}) where msk is the master secret key and PP_{IBE} is public parameters for the IBE. The system public parameters, $PP = (PP_{\text{PAKE}}, PP_{\text{IBE}}, \text{H})$ are made public. KGA keeps msk secret.

– Extract: For a given identity ID of a server, KGA runs the private key generation algorithm of the IBE, to output a private decryption key, dk_{ID}. It is assumed that the private decryption key is transmitted to a server with the identity ID via a secure channel.

– Enrolment: A client, C generates a password, pw according to a predefined password creation guideline [b-G17]. It is assumed that a secure channel is established between the client and the server, S. To register an application service, the client sends (Register-Enrol, ID_{C}, H(pw)) to the server via a secure channel.

Key Establishment: a client, C and a server, S execute a run of a HAKE protocol to share a key for a session as follows (see also Figure 6):

– The client computes a hashed password, H(pw). Using it, the client performs its part in PAKE with the following modification: For each m_{C} to be sent to the server in PAKE, the client encrypts m_{C} by using the IBE, and then sends a ciphertext, CT_{C}. The client finally computes a common session key, ssk by performing its part of PAKE.

– Whenever the server receives a ciphertext, CT_{C} from the client, the server decrypts the message m_{C} with its private decryption key dk_{S}. The server finally computes a common session key, ssk by performing its part of PAKE.

Similar to the HAKE based on an IBS in clause 8.2.1, various HAKE protocols can be constructed by using a PAKE and an IBE. For a simple HAKE design, it can be constructed by encrypting the password such as [b-CCHK15].

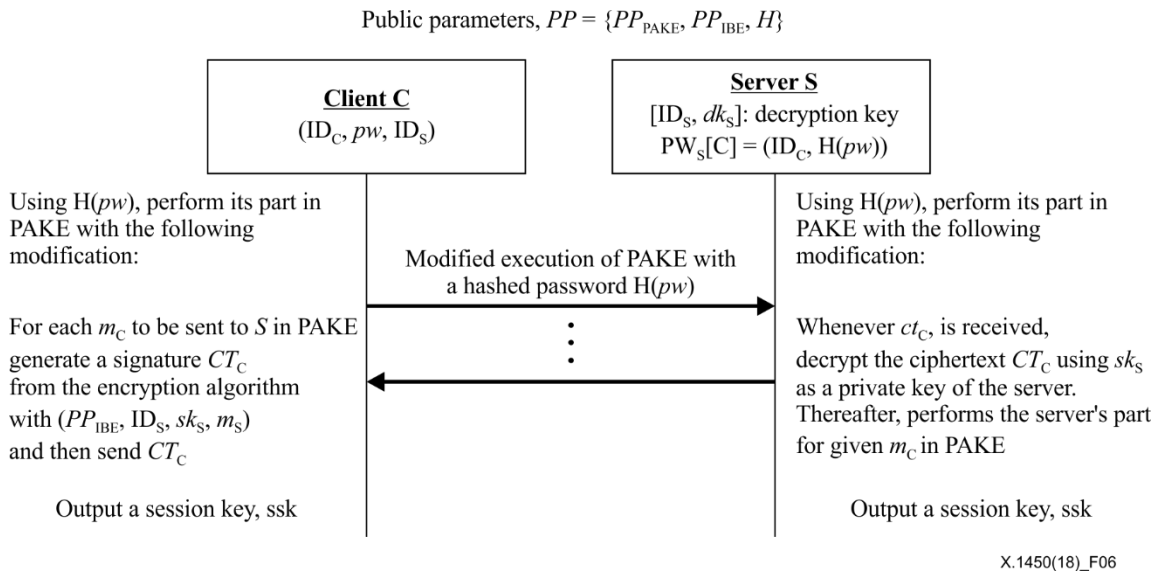

Figure 6 – Construction of HAKE using two-party PAKE and an IBE

9.3 HAKE protocols using shared secrets and public key infrastructure (PKI)

In the protocols, the authentication between a client and a server relies not only on the shared weak secrets, but also on the availability of a PKI. The server is issued a certificate by a CA, where its name and server public key are cryptographically bound by using a signature. Usually the public key in the certificate serves only one purpose, that is, either signature verification or message encryption. The following notations are used in the description of two succeeding schemes:

– *PW*: password shared between client and server
– $m_{1...}m_n$: messages exchanged between client and server by a PAKE protocol
– *REQ*: request message to authenticate the server
– PRK_S: private key of server
– PUK_S: public key of server
– $Cert_S$: server certificate
– N_C: nonce generated by client
– *sign*: signing algorithm
– σ_S: signature generated by server
– *encrypt*: public key encryption algorithm
– *CT_C*: ciphertext generated by client

9.3.1 Using shared secrets and digital signature schemes

A client and a server perform mutual authentication and key exchange by using the shared weak secret *PW*. Thereafter, the client initiates a request message *REQ* to authenticate the server. The server computes the hash value of all transaction messages ($m_{1,...,}m_n$) between the client and server during the password based authentication and key exchange process, as well as the message *REQ*, i.e., $H(m_1||...||m_n||REQ)$. The server signs this hash value by using its private key, PRK_S which is associated with the server public key PUK_S in the certificate $Cert_S$. The signature σ_S along with the server certificate $Cert_S$ is sent to the client. The client verifies the signature σ_S by using the server public key PUK_S in the certificate $Cert_S$ in order to validate the authenticity of the server. The process is depicted in Figure 7.

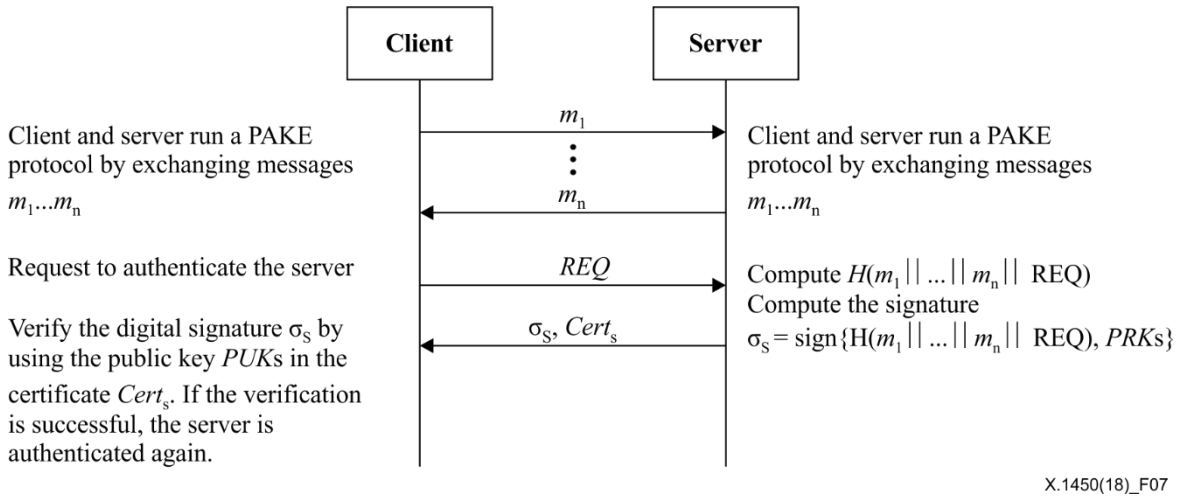

Figure 7 – Construction using weak secrets and a digital signature scheme

9.3.2 Using shared secrets and public-key encryption schemes

A client and a server authenticate each other based on the shared weak secret *PW*. Thereafter, the client initiates a request message *REQ* to authenticate the server, and the server responds to the client with its certificate $Cert_S$. The client hashes all transaction messages between the client and server during the password based authentication process, as well as the message *REQ*, i.e., $H(m_1||...||m_n||REQ)$. The client generates a nonce *Nc* and encrypts it along with the hash value by using the server public key PUK_S in the certificate. The client sends the encrypted message *CT_C* to the server. The server decrypts the message with its private key PRK_S, and acknowledges the client with the nonce *N*c. The client compares the received nonce with the sent one. If both are identical, then the client can assure the authenticity of the server as only the server can obtain the nonce by using its private key PRK_S. The process is depicted in Figure 8.

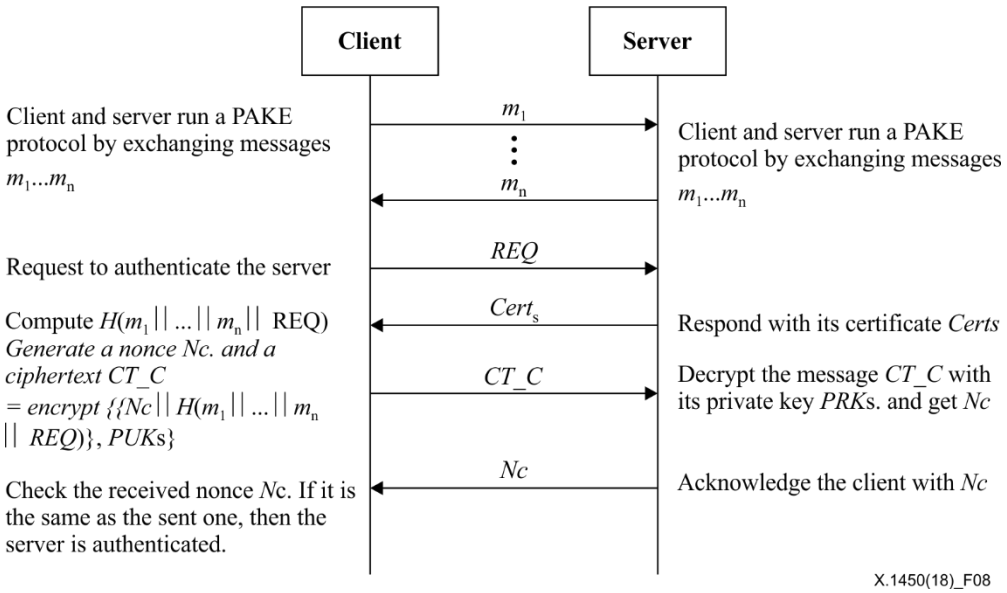

Figure 8 – Construction using weak secrets and a public-key encryption scheme

10 Extensions of HAKE

In order to mitigate server impersonation attacks, the HAKE protocols presented above can be extended with a slight modification to deal with the single-factor authentication using a symmetric key or biometrics such as fingerprint, face, and voice [b-ITU-T X.1087], and multi-factor authentication [b-ITU-T X.1158] using a combination of shared secrets such as a password and biometrics.

Similar to the protocols in clause 9, the resulting protocols perform authentication based on symmetric and asymmetric authentication systems. For symmetric authentication, a client and a server are assumed to share secrets, for example, biometrics such as fingerprint and face or information transformed from biometrics. For asymmetric authentication, a server is assumed to hold a private key and its corresponding public key for the IBC or PKI.

Appendix I

Comparison of HAKE protocols

(This appendix does not form an integral part of this Recommendation.)

Some examples of well-known two-party PAKE protocols, are DH-EKE [b-IETF RFC 6124], SPEKE [b-ISO/IEC 11770-4], SRP [b-IETF RFC 2945], PAK [b-ITU-T X.1035], and AMP [b-ISO/IEC 11770-4]. Similar to the example of clause 9.2.1, an HAKE protocol can be constructed straight forwardly using each of the above PAKE protocols as a component protocol, and an IBS scheme or PKI. For simplicity, the resulting HAKE protocols are denoted by "HAKE with DH-EKE", "HAKE with SPEKE", "HAKE with SRP", "HAKE with PAK", and "HAKE with AMP", respectively. Security properties of the HAKE protocols are then analysed and a comparison is provided in terms of the security properties.

A detailed comparison among the HAKE protocols is given in Table 1. In the table, "Y" means that the security property on the left-hand side is satisfied by the specific HAKE protocol, "-" that the security property on the left-hand side is not satisfied by the specific HAKE protocol.

Table 1 – Comparison of HAKE protocols using an IBC or PKI

	HAKE with DH-EKE [b-IETF RFC 6124]	**HAKE with SPEKE [b-ISO/IEC 11770-4]**	**HAKE with SRP [b-IETF RFC 2945]**	**HAKE with PAK [b-ITU-T X.1035]**	**HAKE with AMP [b-ISO/IEC 11770-4]**
Perfect forward secrecy	Y	Y	Y	Y	Y
Mutual authentication	Y	Y	Y	Y	Y
Authenticity of server	Y	Y	Y	Y	Y
Authenticity of client	Y	Y	Y	Y	Y
Resistance to Reply attack	Y	Y	Y	Y	Y
Resistance to MITM attack	Y	Y	Y	Y	Y
Resistance to on-line dictionary attack	Y	Y	Y	Y	Y
Resistance to off-line dictionary attack	Y	Y	Y	Y	Y
Resistance to server-compromised attack (Verified-based PAKE)	–	–	Y	–	Y
Resistance to server-compromised dictionary attack	–	–	–	–	Y
Resistance to server impersonation with PW leakage	Y	Y	Y	Y	Y

Bibliography

[b-ITUT-Q.1743] Recommendation ITU-T Q.1743 (2016), *IMT-Advanced references to Release 11 of LTE-Advanced evolved packet core network*.

[b-ITU-T X.843] Recommendation ITU-T X.843 (2000) | ISO/IEC 15945:2002, *Information technology – Security techniques – Specification of TTP services to support the application of digital signatures.*

[b-ITU-T X.1035] Recommendation ITU-T X.1035 (2007), *Password-authenticated key exchange (PAK) protocol*.

[b-ITU-T X.1087] Recommendation ITU-T X.1087 (2016), *Technical and operational countermeasures for telebiometric applications using mobile devices.*

[b-ITU-T X.1154] Recommendation ITU-T X.1154 (2015), *Guidelines on local linkable anonymous authentication for electronic services*.

[b-ITU-T X.1252] Recommendation ITU-T X.1252 (2010), *Baseline identity management terms and definitions.*

[b-ITU-T X.1254] Recommendation ITU-T X.1254 (2012), *Entity authentication assurance framework.*

[b-BF01] D. Boneh, and M. Franklin, (2001), *Identity-based encryption from the weil pairing, CRYPTO 2001, Springer-Verlag.*

[b-BNN] M. Bellare, C. Namprempre, and G. Neven (2009), *Security Proofs for Identity-Based Identification and Signature Schemes. Journal of Cryptology 22(1):1--61*.

[b-CCHK15] K. Y. Choi, J. Cho, J. Y. Hwang, T. Kwon (2015), *Constructing Efficient PAKE Protocols from Identity-Based KEM/DEM, WISA'15, Springer-Verlag.*

[b-G17] P. Grassi, SP 800-63B-3 (2017), *Digital Identity Guidelines, Authentication and Lifecycle Management, NIST (doi:10.6028/NIST.SP.800-63b).*

[b-HKCJS15] J.Y. Hwang, S.-H. Kim, D. Choi, S.-H. Jin, and B. Song (2015), *Robust Authenticated Key Exchange Using Passwords and Identity-Based Signatures, SSR'15, LNCS Vol. 9497, Springer-Verlag, pp.43-69.*

[b-ISO/IEC 11770-4] ISO/IEC 11770-4 (2006), *Information technology – Security techniques – Key management – Part 4: Mechanisms based on weak secrets.*

[b-ISO/IEC 14888-3] ISO/IEC 14888-3 (2016), *Information technology – Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms.*

[b-ISO/IEC 18033-5] ISO/IEC 18033-5 (2015), *Information technology – Security techniques – Encryption algorithms – Part 5: Identity-based ciphers.*

[b-ISO/IEC 25185-1] ISO/IEC 25185-1 (2016), *Identification cards – Integrated circuit card authentication protocols – Part 1: Protocol for Lightweight Authentication of Identity.*

[b-ISO/IEC 29192-4] ISO/IEC 29192-4 (2013), *Information technology – Security techniques – Lightweight cryptography – Part 4: Mechanisms using asymmetric techniques.*

[b-IEEE P1363.2] IEEE P1363.2 (2008), *Standard Specifications for Password-Based Public-Key Cryptographic Techniques.*

[b-IETF RFC 6124] IETF RFC 6124 (2015), *An EAP Authentication Method Based on the Encrypted Key Exchange (EKE) Protocol*.

[b-IETF RFC 2945] IETF RFC 2945 (2000), *The SRP Authentication and Key Exchange System.*

[b-S05] B. Schneier (2005), *Two-Factor Authentication: Too Little, Too Late in Inside Risks 178, Communications of the ACM, 48(4S).*

[b-S84] A. Shamir (1984), *Identity-Based Cryptosystems and Signature Schemes. Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science, 7:47--53.*

[b-SK99] S. Halevi and H. Krawczyk (1999), *Public-key cryptography and password protocols. ACM Trans. Information and System Security, 2(3), pp.230-268.*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2019