International Telecommunication Union

ITU-T Technical Report

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(26 SEPTEMBER 2014)

XSTR-PKIS
Current and new challenges for public-key infrastructure standardization

Summary

This Technical Report explores the issues and threats currently facing the deployment of public-key infrastructure (PKI), and the new challenges PKI will experience in areas such as wireless PKI (WPKI), cloud computing, smart grid, and machine-to-machine (M2M) in general.

Keywords

Public-key infrastructure (PKI).

Change Log

This Technical Report contains Version 1 of the ITU-T Technical Report on "*Current and new challenges for public-key infrastructure standardization within ITU-T*" approved at the ITU-T Study Group 17 meeting held in Geneva, 17-26 September 2014.

Editors:

Erik Andersen
Rapporteur for Question 11/17 on Generic technologies to support secure applications
ITU-T Study Group 17

Tel: +45 20 97 14 90
E-mail: era@x500.eu

CONTENTS

Page

Page

Page

List of Figures

Technical Report ITU-T XSTP-PKIS

Technical Report ITU-T
Current and new challenges for public-key infrastructure standardization

Summary

This Technical Report explores the issues and threats currently facing the deployment of public-key infrastructure (PKI), and the new challenges PKI will experience in areas such as wireless PKI (WPKI), cloud computing, smart grid, and machine-to-machine (M2M) in general.

1 Introduction

Public-key infrastructure (PKI), as defined by Recommendation ITU-T X.509, *Information Technology – Open Systems Interconnection – The Directory − Public-key and attribute certificate frameworks*, has primarily been deployed in areas such as e-government, e-banking, e-commerce, e-health, etc.

PKI is typically used together with the deployment of other specifications. Cryptographic key management is a crucial and complex activity for network security, especially in a network with many entities. Such networks can only be managed with the deployment of PKI. Transport layer security (TLS) is another example of an important IETF standard that is dependent on PKI. Such related specifications are described to establish an understanding of the PKI requirements and the security around PKI deployments in relation to these sibling specifications.

There are numerous threats to PKI and its sibling specifications. Many of these threats are identified and documented together with possible mitigations.

PKI also is seen as having a major role in new areas, such as in machine-to-machine (M2M) environments with special consideration to cloud computing, smart grid, e-health, etc., and within wireless PKI (WPKI), which is especially interesting for developing countries. These environments require special considerations with respect to the basic PKI standardization (b-ITU-T X.509) and procedures required for the secure deployment of PKI.

For those who are not familiar with PKI, see the Introduction for more information.

This Technical Report describes these issues in details and suggests relevant PKI standards activities to be initiated within ITU-T Study Group 17 in cooperation with other standardization organizations.

This introduction provides to a casual reader an overview of the content of this Technical Report and inspires experienced readers to dig into the details in the main clauses. It is recommended that everyone read the conclusions in clause 30.

Public-key infrastructure (PKI), as specified in [b-ITU-T X.509], is currently used in different areas to provide security for sensitive transactions and operations. It is primarily deployed in areas such as e-banking, e-government, e-commerce, etc., where human beings are involved in the establishment, maintenance and operation of PKI.

This Technical Report explores the issues and threats currently facing the deployment of PKI, and the new challenges PKI will experience in areas such as wireless PKI (WPKI), cloud computing, smart grid, and machine-to-machine (M2M) in general.

PKI deployment is part of a general security strategy. PKI is most often used together with or is part of the deployment of other general specifications, where transport layer security (TLS) and Internet protocol security (IPsec) are important examples. It is therefore useful to consider these other general specifications and to see how they relate to PKI and how they affect PKI development and deployment.

Security is a very broad issue, and this Technical Report does not intend to cover all possible aspects of security. There are issues around physical protection, careless or malicious employees, firewalls, virus protection programs, etc. Such issues are covered by more general specifications such as [b-ISO/IEC 27001] and [b-ISO/IEC 27002]. This Technical Report does not discuss these issues, but it is assumed that proper actions are taken in all relevant areas.

This Technical Report covers the following subjects:

a) Clauses 2 to 6 introduce some basic PKI concepts for those who do not have the necessary PKI background. Those clauses also describe the basic features of digital signatures.

b) Clause 7 introduces PKI deployment issues.

c) Clause 8 introduces the concept of a trust broker as a new entity type to be included in PKI. Such a trust broker will increase security by providing assistance to those entities relying on the credibility of PKI.

d) Clause 9 describes briefly specifications for digital signatures.

e) Clause 10 considers message authentication codes (MACs), as they are alternatives or supplementary to digital signatures. Message authentication codes are, for example, part of the transport layer security (TLS).

f) Clause 11 discusses the role of hardware security modules (HSMs). Such HSMs are important components in the practical deployment of PKI. HSMs are used to protect important PKI information and crucial PKI procedures.

g) Clause 12 considers key management, as key management is a very important security concept, which is strongly related to and in most cases is dependent on PKI.

h) Clause 13 is concerned with TLS. TLS is used in many environments where PKI is employed and there is a substantial interplay between TLS and PKI. The clause provides an overview of TLS. This clause discusses some threats specific to TLS and their preventions.

i) Clause 14 considers an alternative TLS specification to be used by mobile networks. It is interesting, as it is adapted to a resource constraint environment.

j) Clause 15 considers secure associations and discusses Internet key exchange (IKE) and Internet protocol security (IPsec).

k) Clause 16 discusses the concept of perfect forward secrecy (PFS). When evaluating a key management system, it is important for it to have perfect forward security property to limit the damage of an intrusion.

l) Clause 17 lists the identified threats to PKI and to information and communication technology (ICT) networks in general. It also gives some indication on how to mitigate such threats.

m) Clause 18 lists the current PKI and key management activities completed and/or under development by different standardization bodies.

n) Clause 19 considers the PKI challenges imposed by wireless PKI (WPKI). As the mobile phone becomes a prevailing communication device, there is a need to adapt PKI to the mobile environment. Developing counties have an interest supporting WPKI for low function devices.

o) Clause 20 considers the general PKI challenges imposed by a machine-to-machine (M2M) environment. PKI is relevant in M2M environments where human beings are not involved in the communication. Smart grid and aspects of cloud computing are examples of M2M environments.

p) Clause 21 considers the PKI challenges imposed by cloud computing.

q) Clause 22 considers the security issues for the smart grid.

r) Clause 23 considers the special issues concerning smart grid substations.

s) Clause 24 considers the security aspect for the electricity consumers' special networks.

t) Clause 25 considers security threats specific to the smart grid.

u) Clause 26 considers security threats to smart metering.

v) Clause 27 discusses possible extensions to the base [b-ITU-T X.509].

w) Clause 28 discusses a possible need for PKI profiling.

x) Clause 29 discusses a possible need for developing PKI management procedures.

y) Clause 30 is the Conclusion of this Technical Report, which summarizes general recommendations for future steps.

z) The Glossary: This is a list of all the acronyms and their meanings used in this Technical Report.

z1) The Bibliography: This is a list of all the specifications referenced by this Technical Report. These references may also be useful in the future work on PKI.

Many standards make use of PKI technologies; however, they are not covered in this Technical Report.

2 Basic PKI concepts

2.1 Object identifiers

Object identifiers are not particular to PKI or to any other subject for that matter. Object identifiers are widely used in many different areas.

Object identifiers are just strings of numbers. They are allocated in a hierarchical manner, and are allocated in such a way that two distinct objects should not have the same object identifier.

An object identifier can identify any physical or virtual object. It can identify a particular type of document, a particular information type, a specific hashing algorithm, etc.

2.2 Cryptography

Cryptography in the context of this Technical Report is about encryption and decryption of data to be transmitted (or stored). Encryption is performed by applying a mathematical algorithm to transform data so that it can only be read by the intended party after having been decrypted. Cryptographic technologies are the underlying techniques for PKI and related specifications. Cryptographic techniques provide for integrity, confidentiality, authenticity and to some degree for non-repudiation.

– Integrity implies that data from creation until consumption cannot be modified without detection.

– Confidentially implies prevention of disclosure of information to unauthorized entities.

– Authenticity implies that the origin of the information is known with a high degree of certainty.

– Non-repudiation implies that an issuer of information cannot successfully repudiate having done that.

Cryptographic systems belong to one of two categories:

– *Symmetric-key cryptography*; or

– *Public-key cryptography*.

2.2.1 Symmetric-key cryptography

In symmetric-key cryptography, the same secret key is used for both encryption and decryption of data. This implies that both parties in a communication have copies (share) of the same cryptographic key. The symmetric-key cryptography requires less processing than public-key cryptography described in the following clause. Symmetric-key cryptography is therefore the prevailing technique for ensuring confidentiality of data.

The symmetric-key cryptography is effective only when the two parties keep the shared key secret. If anyone else discovers the key in any way, it affects both confidentiality and authentication. A person with an unauthorized symmetric key not only can decrypt messages sent using that key, but can also encrypt false messages and send them on behalf of the legitimate parties.

For security reasons, different shared secret keys must be used for different communication channels. Otherwise, the compromise of the single shared key will put the whole network in jeopardy.

Management of shared secret keys, especially in a large network, can be a major issue. Various key management techniques have been devised to facilitate establishment of shared secret keys in a secure way as discussed in clauses 12 to 15.

There are different algorithms for symmetric cryptography. In this Technical Report, such an algorithm is called a *symmetric key algorithm*. Other sources also use the term secret key algorithm.

2.2.2 Public-key cryptography

Public-key cryptography is a technique that makes use of a mathematically related key-pair comprised of a *private key* and a *public key*. Data encrypted by one of the keys of the pair can only be decrypted by the other key.

The holder of a private key must keep it secured, while the corresponding public key may be distributed more freely.

– Data encrypted by the private key can be decrypted by anyone having the public key. This mode is used for digital signatures.

– Data encrypted by a public key can only be decrypted by the holder of the private key. This mode may, for example, be used for encryption of e-mails. It may also be used for encryption of a symmetric shared key transmitted to the holder of the private key (see clause 12.7).

Public-key cryptography is also called *asymmetric key cryptography* to distinguish it from symmetric key cryptography

2.3 Hashing algorithms

A hashing algorithm is the specification on how to take an arbitrary message (bit string) and produce a fixed length hash value called a *digest*. A good hash algorithm is designed to satisfy the following properties:

– It is a one-way algorithm, meaning it is infeasible to find a message that maps to a specific given digest.

– It is infeasible to modify a message without changing the digest.

– It is collision resistant, meaning it is infeasible to find any two distinct messages that map to the same digest.

Many different hashing algorithms of different quality have been defined. A hashing algorithm is assigned an object identifier for easy reference.

The US National Institute of Standards and Technology (NIST) has published what they call secure hashing algorithms. They are published in a series of NIST FIPS PUB 180 standards.

2.4 ITU-T X.509 digital signatures

ITU-T X.509 digital signatures are used for digital signing of data of types defined by [b-ITU-T X.509] itself and by other parts of the ITU-T X.500 series of Recommendations.

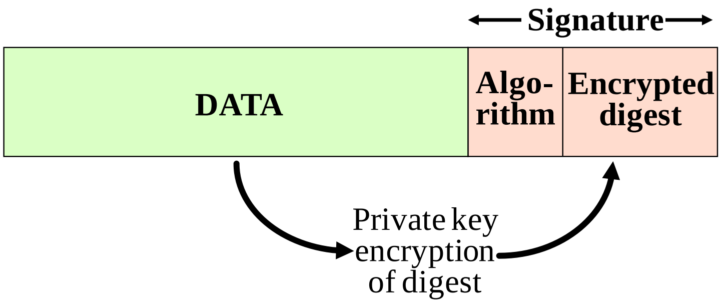


Figure 1 – ITU-T X.509 digital signature

To digitally sign some kind of data, an appropriate hashing algorithm computes a digest. The creator of the signature then uses its private key to encrypt the digest. A combination of a hashing algorithm and a public key algorithm is called a *signature algorithm*. A signature algorithm is also identified by an object identifier. The digital signature is comprised of the signature algorithm and the encrypted digest. Figure 1 illustrates how the signature is appended to the data to be transmitted (or to be stored).

The verifier of the signed data is assumed to be in the possession of the public key of the signer. The evaluator at the recipient side knows from the embedded signature algorithm the hashing and public key algorithms that have been used by the sending signer. The verifier can then decrypt the encrypted digest and thereby retrieve the digest as created by the signer. Next, the verifier then creates its own digest of the data received. If the two digests are identical, then, with high degree of certainty, the evaluator can assume:

a) The data has not been changed since the signer signed it.

b) The signer is in possession of the private key that corresponds to the public key held by the verifier providing proof of the identity of the signer.

c) The signer will have difficulties in reputing having ever signed the data.

This type of digital signatures may also be used in other areas, such as digital signing of e-mails, documents, and computer program code. However, the verifier may require some additional information to perform the evaluation. More information on digital signatures is provided in clause 9.

2.5 Certificates

[b-ITU-T X.509] defines two types of certificates:

– A public-key certificate is a digital document that ties an asymmetric key pair to a named entity. A public-key certificate is signed by a so-called *certification authority (CA)*.

– An attribute certificate is a digital document that assigns privileges to the holder of the attribute certificate. An attribute certificate is signed by a so-called *attribute authority* (AA).

This Technical Report currently only considers public-key certificates. Clause 4 gives some details on the structure and content of a public-key certificate.

3 PKI components

3.1 Certification authority (CA)

A certification authority (CA) is an entity that may issue and/or sign public-key certificates for other entities. A public-key certificate is a digital document that binds a public key to the identity of the subject of the public-key certificate. CA certifies that the information within the public-key certificate is reliable with some degree of certainty depending on the conditions under which the public-key certificate was issued.

The degree of checking by CA of the information provided in the public-key certificate and the level of trust in CA determine the level of confidence in the public-key certificate. A separate registration authority may perform the checking of the information before CA issues a public-key certificate.

CA may issue public-key certificates to other CAs. Such a public-key certificate is called a *CA-certificate*. This process may be repeated effectively creating a hierarchy of CAs (see Figures 8 and 9).

3.2 Registration authority

Some of the responsibilities of CA may be delegated to a registration authority, which may or may not be under the same management as CA. A registration authority is not described as a separate entity in [b-ITU-T X.509]. [b-IETF RFC 5280] is an important IETF specification, which includes the registration authority as a separate type of entity without much explanation. The term is also used in the general PKI literature.

A registration authority is responsible for validating the information to be included in a public-key certificate. As an example, a registration authority may check that the entity that is subject for the public-key certificate is entitled to use the suggested name and that the suggested name uniquely identifies the subject.

3.3 End-entities

An end-entity is an entity to which a public-key certificate may be issued and this is called an end-entity public-key certificate. By definition, an end-entity is not permitted to issue public-key certificates to other entities.

3.4 Relying party

A relying party is an entity that uses the information in a public-key certificate to make a decision as to the reliability of some information from the owner of the public-key certificate, e.g. a document, an e-mail, a financial transaction, etc.

A plain human user sitting at some kind of client system is not in a position to do all the validity checking of public key certificate. The user relies on the system they are using. Typically, this is a web browser, but it may also be an e-mail system or similar systems providing protection against potentially harmful attacks.

3.5 Certificate revocation list (CRL)

A relying party that relies on the content of a public-key certificate can perform some checking before possibly accepting the public-key certificate as being valid. One of the checks verifies that the public-key certificate has not been revoked.

CA is responsible for maintaining a list of public-key certificates that for some reason have been revoked before expiration date. Such a list is called a *certificate revocation list* (CRL). A public-key certificate stays on CRL until expiration time. In some environments, CRLs may be journalized for future legal actions.

When validating a public-key certificate, a relying party may retrieve the relevant CRL to check if the public-key certificate under validation is included in that list.

CA may delegate the issuing of CRLs to some other trusted organization.

3.6 Online certificate status protocol (OCSP)

CA may also establish a service based on the online certificate status protocol (OCSP) as defined by [b-IETF RFC 6960] by providing an OCSP responder. The operation of this responder may be delegated to a trusted entity by CA.

A relying party may use OCSP to retrieve the revocation status of one or more public-key certificates. This is an alternative to retrieving complete CRLs.

OCSP is part of the work performed by the IETF PKIX group (see clause 18.34). The lightweight OCSP profile for high-volume environments is defined in [b-IETF RFC 5019].

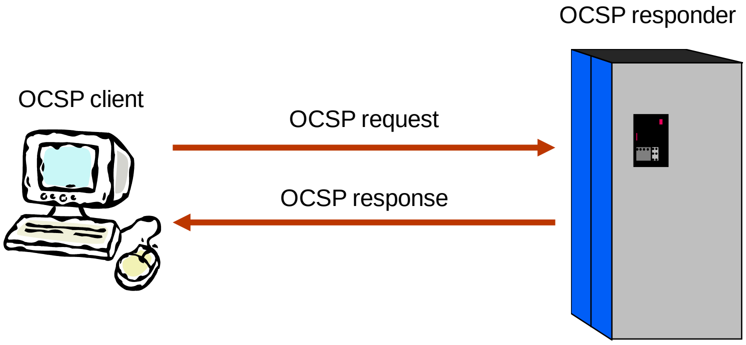


Figure 2 – Online certificate status protocol

OCSP is a request/response protocol for retrieving public-key certificate revocation status from an OSCP responder.

Figure 2 illustrates such retrieval. For example, the OCSP client may be a web browser that checks the validity of a public-key certificated provided by a web server.

The OCSP responder may be the CA that has issued public-key certificates or the CA may have delegated this responsibility to some other entity.

3.7 Trust anchor

The entire PKI concept is built on the assumption that somewhere there is a point of trust or a *trust anchor*. Such trust anchors may be provided by commercial or governmental organizations that are considered trustworthy or they may be established in other ways. As an example, an enterprise may establish its own trust anchor. A trust anchor is typically a CA with this special status.

A trust anchor is an entity that is trusted for the purpose of public-key certificate validation by a relying party. Information about a trust anchor (*trust anchor information*) is typically configured into the relying party in a so-called *trust anchor store*. Vendors of operating systems include and maintain a trust anchor store. This information is typically provided in the form of a CA-certificate. As the trust anchor may be at the top of a trust chain, its CA-certificate cannot be signed by another entity. Such a trust anchor therefore signs its own CA-certificate using its own private key. Such a CA-certificate is called a *self-signed certificate*.

3.8 Repository

A repository is a generic term used to denote any method for storing public-key certificates and CRLs so that they can be retrieved by entities. The repository may be provided by an [ITU-T X.500] or a lightweight Directory access protocol (LDAP) directory.

3.9 Key escrow

Key escrow is an arrangement in which the keys needed to decrypt encrypted data are held in an escrow so that, under certain circumstances, an authorized third party may gain access to those keys. These third parties may include businesses, who may want access to employees' private communications, or governments, who may wish to be able to view the contents of encrypted communications.

The technical problem is a largely structural one since access to protected information is provided *only* to the intended recipient and to some third party. The third party could be permitted access only under carefully controlled conditions as, for example, a court order.

[b-ETSI TS 101 456] and [b-ETSI TS 102 042] specify that private keys used for digital signatures shall not be stored in an escrow.

4 Public-key certificates

4.1 Introduction

A public-key certificate is issued and/or certified by a certification authority (CA). It is the responsibility of the issuing CA to ensure that the information in the public-key certificate is correct and that the entity, to which it is issued, has the corresponding private key.

4.2 Basic content

A public-key certificate (sometimes called a digital certificate, an (ID) certificate or just a certificate) is a digital document that holds information about an entity that has been assigned an asymmetric key pair. Its content is illustrated in Figure 3. In the following, there is a short description of the different fields of a public-key certificate:

– The version of the public-key certificate specification used for creating this public-key certificate - only version 3 is relevant for this Technical Report.

– The serial number that has to be different from the serial number of any other public-key certificate issued by CA. Some specifications require the sequence number to be assigned somewhat randomly to make it impossible to guess the next sequence number to be allocated by a particular CA.

– The signature algorithm (as an object identifier (OID)) used by CA to sign the certificate.

– The name of the *issuer* (the issuing CA, which could be a trust anchor). This name is a distinguished name as defined in [b-ITU-T X.501].

– The validity specifies the time period where the public-key certificate is valid. It is specified by the components, *notBefore* and *notAfter*. If a public-certificate is revoked before the end of the validity period, it will be put on a certificate revocation list (CRL) and it stays there until the expiration time. A long validity period could result in a long CRL.

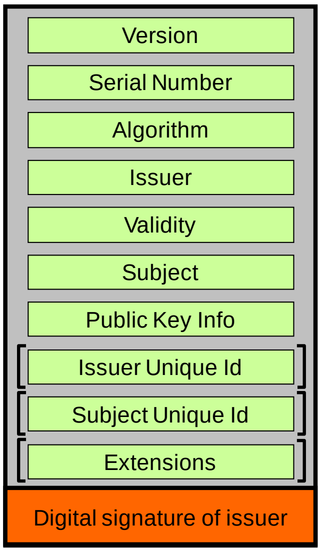


Figure 3 – Public-key certificate

- The name of the entity to which this public-key certificate is issued (the *subject*). This name is a distinguished name as defined in [b-ITU-T X.501]. However, this name may be empty, for an end-entity public key certificate provide that a name is instead given in the subject alternative name extension (see clause 4.3.2).
- The public-key specification.
- The Issuer Unique Id and Subject Unique Id fields are deprecated and typically are not be included.
- The Extensions field is discussed in clause 4.3.

A public-key certificate is digitally signed by the issuing CA.

4.3 Extensions

To provide for additional fields of a public-key certificate, an extension mechanism has been devised. Each extension provides more information and/or use restrictions on the public-key certificate.

An extension is identified by an object identifier. An extension may be labelled *critical* meaning that it shall be processed if supported by the relying party. Otherwise, the relying party rejects the public-key certificate. If the extension is labelled *non-critical*, it will be processed if supported. Otherwise, the extension is ignored.

General-purpose software with relying party support can handle all the extensions it supports and rejects the public-key certificate if does not support an extension labelled *critical*. However, a particular environment may require that certain extensions are present and some other extensions are not. A general-purpose program will not be able to check for such restrictions.

Both [b-ITU-T X.509] and [b-IETF RFC 5280] define several extensions. The following clauses describe examples of extensions.

4.3.1 Basic constraints extension

The basic constraints extension defined by [b-ITU-T X.509] specifies whether a public-key certificate is a CA-certificate or an end-entity public-key certificate. The private key associated with a CA-certificate may be used to sign other public-key certificates, while this is not the case for end-entity public-key certificates. If this extension is not included, it is assumed that the public-key certificate is an end-entity public-key certificate. However, this extension has been added at a later stage during the standardization process, so there may be other CA-certificates that do not include this extension.

Specification for the future use of PKI could mandate that CA-certificates have this extension and it could then be assumed that if the extension is not present, the public-key certificate is an end-entity public-key certificate.

4.3.2 Subject alternative name extension

The subject alternative name extension defined by [b-ITU-T X.509] allows name formats different from the directory distinguished name format to be included in a public-key certificate. As an example, a name in the Internet domain name format may be used. The extended validation public-key certificates defined by the CA/Browser Forum require such an extension to be present (see clause 18.2.2).

4.3.3 Name constraints extension

The name constraints extension defined by [b-ITU-T X.509] allows restriction to be put on names used in subsequent public-key certificates in a certification path (see clause 5.2 for a definition of certification path).

4.3.4 Key usage extension

The key usage extension defined by [b-ITU-T X.509] is used to specify the cryptographic operations that may be performed using the key-pair associated with the public-key certificate, for example, it could indicate that the key-pair could be used for the creation and evaluation of signatures but could not be used for encryption.

4.3.5 Extended key usage extension

The extended key usage extension defined by [b-ITU-T X.509] is used to apply restrictions on the uses of the key-pair associated with the public-key certificate. However, [b-ITU-T X.509] does not specify any values to be entered here. Such values are defined by [b-IETF RFC 5280]. As an example, the extension may specify that the public-key certificate in question be intended for use by a TLS server (see clause 13).

4.3.6 Subject directory attributes extension

The subject directory attributes extension defined by [b-ITU-T X.509] is used for holding one or more directory attributes providing information related to the owner (subject) of the public-key certificate. The format of directory attributes is defined in [b-ITU-T X.501] and a number of attribute types are defined in [b-ITU-T X.520]. Some ITU-T X.509 related attribute types are defined in [b-ITU-T X.509]. A directory attribute consists of an object identifier that identifies the type of attribute followed by one or more values.

4.3.7 Certificate policies extension

The certificate policies extension defined by [b-ITU-T X.509] is used for holding one or more object identifiers, where an object identifier identifies a certificate policy document as described in clause 4.4.

4.3.8 CRL distribution points extension

The CRL distribution points extension defined by [b-ITU-T X.509] is used for holding one or more addresses of points where revocation information may be found. An address may typically be a uniform resource locator (URL). By allowing different types of public-key certificates to have different CRL access points, it is possible to keep down the size of CRL.

4.4 Certificate policy (CP)

A certificate policy is some kind of a textual document describing the policies under which the public-key certificate has been issued.

A certificate policy focuses on public-key certificates and the issuing CA's responsibilities regarding these public-key certificates. It defines public-key certificate characteristics such as usage, enrolment and issuance procedures, as well as liability issues.

A certificate policy typically answers the question on the purposes of the certificate serves, and under which policies and procedures the public-key certificate has been issued. A certificate policy typically addresses the following issues:

- How are users authenticated during public-key certificate enrolment.
- Legal issues, such as liability that might arise if CA becomes compromised or is used for something other than its intended purpose.
- The intended purpose of the public-key certificate.
- Private key management requirements, such as storage on smart cards, hardware security module or other hardware devices.
- Whether the private key can be exported or archived.
- Requirements for users of the public-key certificates, including what users shall do if their private keys are lost or compromised.
- Requirements for public-key certificate enrolment and renewal.
- Minimum length for the public key and private key-pairs.

[b-IETF RFC 3647] and [b-ETSI TS 102 042] give guidance on how such a document may be drafted. Such a certificate policy document is typically not machine-readable and cannot be analysed by the relying party software. Only the identifying object identifier can be understood by software.

4.5 Certificate practice statement (CPS)

While the certificate policy focuses on a public-key certificate, CPS focuses on CA. CPS is a statement about the way CA issues public-key certificates.

CPS might include the following types of information:

- Positive identification of CA, including the CA name, server name, and the domain name system (DNS) address.
- Certificate policies that are implemented by CA and the public-key certificate types that are issued.
- Policies, procedures, and processes for issuing, renewing, and recovering the public-key certificates.
- Cryptographic algorithms, cryptographic service providers (CSPs), and the key length that is used for the CA-certificate.
- Physical, network, and procedural security for CA.
- The public-key certificate lifetime of every certificate issued by CA.
- Policies for revoking public-key certificates, including conditions for public-key certificate revocation, such as employee termination and misuse of security privileges.
- Policies for certificate revocation lists (CRLs), including where to locate CRL distribution points and how often CRLs are published.
- A policy for renewing the CA's own CA-certificate before it expires.

5 PKI overview

5.1 PKI components

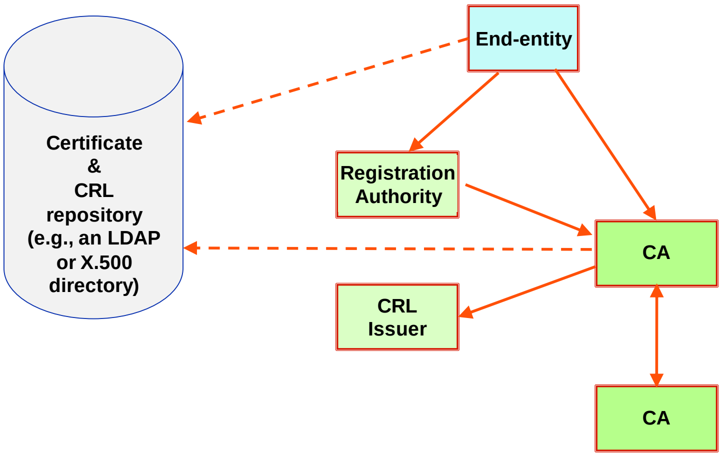

End-entity
Certificate & CRL repository (e.g., an LDAP or X.500 directory)
Registration Authority
CA
CRL Issuer
CA

Figure 4 – PKI components

Figure 4 illustrates the different types of PKI components. The figure does not explicitly represent a trust anchor component, but one can consider it as a CA with a special status.

The figure shows a registration authority as a separate entity. In this example, an end-entity is served by the registration authority, but it could also be a CA requiring a CA-certificate from a superior CA.

5.2 Certification path

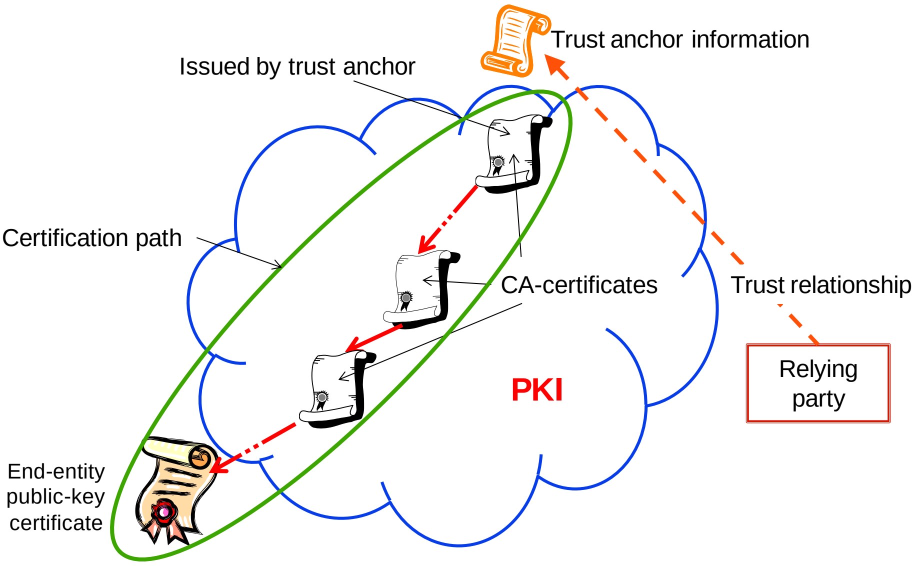


Figure 5 – Certification path

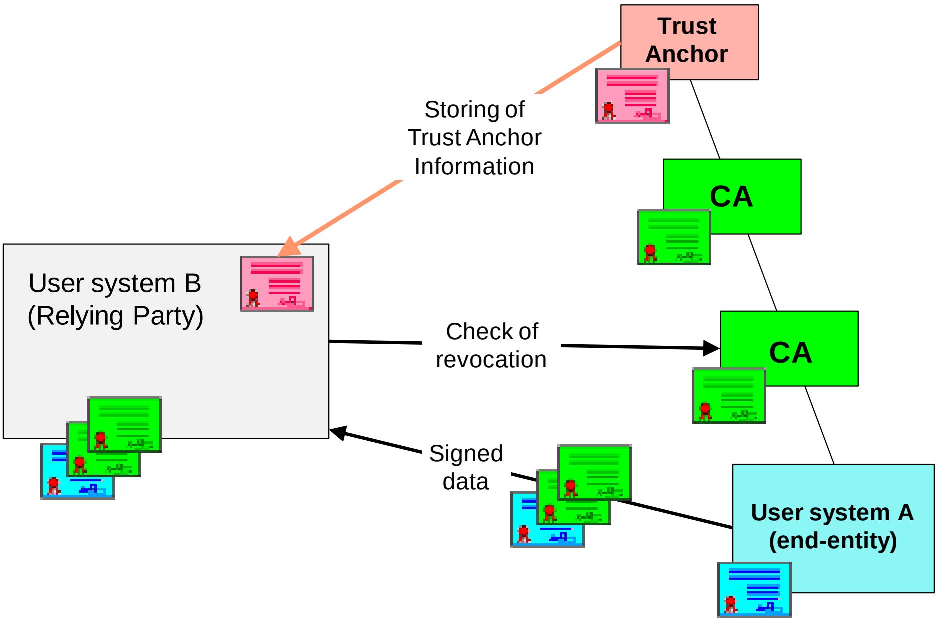
A relying party prefers to check the validity of an end-entity public-key certificate when evaluating information asserted to come from the owner of that public-key certificate (the subject). However, it is not sufficient just to check that end-entity public-key certificate, but also to check the chain of CA-certificates all the way from the trust anchor information down to the end-entity public-key certificate. There may be restrictions imposed by CA-certificates on subordinate public-key certificates (CA-certificate or end-entity public-key certificate, as appropriate). The validation is typically top-down. A certification path is therefore the list of public-key certificate starting with the CA-certificate issued by the trust anchor going down to the end-entity public-key certificate.

The trust anchor information is not included, as it is assumed that the trust anchor information is present in the trust anchor store of the relying party; the trust anchor information is checked for validity and restrictions. If the trust anchor is not available, validation is not possible and the public-key certificate will be rejected. The certification path concept is illustrated in Figure 5.

In the special case, the trust anchor can directly issue the end-entity public-key certificates without any intermediate CAs. This simplifies validation.

Some specifications, such as the transport security layer specification [b-IETF RFC 5246], use the term *certificate chain* in place of certification path, and may define such a certificate chain to be in the reverse order of a certification path as defined above.

5.3 Basic PKI operation



Trust Anchor
Storing of Trust Anchor Information
CA
User system B (Relying Party)
Check of revocation
CA
Signed data
User system A (end-entity)

Figure 6 – Example of PKI operation

Figure 6 shows an example on the interactions among the different PKI components. User system A has an end-entity public-key certificate signed by a CA, which in turn has a CA-certificate signed by another CA, which again has a CA-certificate signed by a trust anchor. The trust anchor has trust anchor information in the form of a self-signed CA-certificate.

When initiating a transaction toward system B, system A signs the transaction and it adds the complete set of public-certificates that comprises the certification path. Through other channels, system B receives the trust anchor information in the form of a self-signed CA-certificate. System B now has all the information necessary to start validating the digital signature on the received transaction.

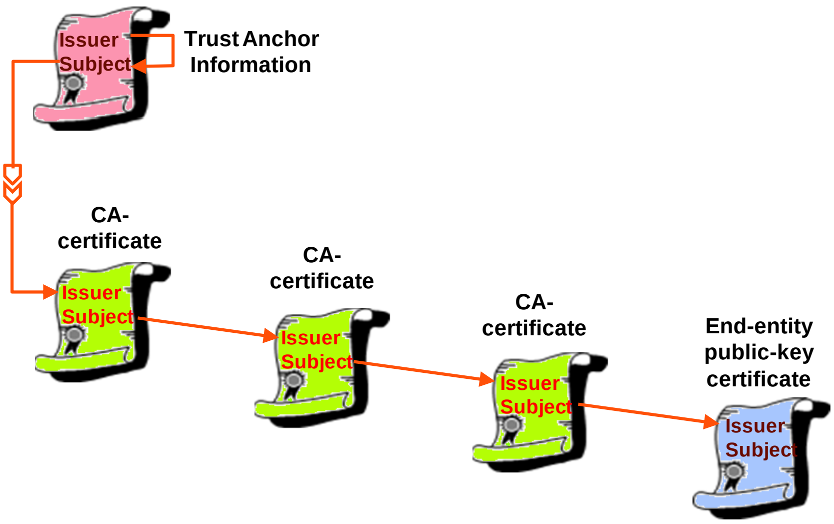


Figure 7 – Chain of public-key certificates

Figure 7 further illustrates a "chain" of public-key certificates. It illustrates that an end-entity public-key certificate has been issued by a CA, which again has a CA-certificate issued by another CA, etc. The top of the "chain" of CA-certificate is a CA-certificate that has been issued by the trust anchor. The figure also illustrates that the trust anchor information is not part of the "chain" of public-key certificates making up a certification path, and when the "chain" is made available at the relying party, the top (first) public-key certificate in the chain is "hooked-on" to the trust anchor information, that has been used for issuing that first CA-certificate.

In the simple and most efficient case, the trust anchor has directly issued the end-entity public-key certificates without any intervening CAs.

6 PKI configurations

6.1 Introduction

Establishment of a PKI is not a trivial matter. The logistics in establishing and maintaining a PKI requires substantial planning. Current specifications referring to PKI provide very little guidance in this area. However, successful PKI deployment is dependent on an efficient PKI management.

This clause discusses two different views of PKI structures: one view from a PKI structure perspective and another from an operational PKI perspective.

6.2 Physical PKI structure view

6.2.1 Hierarchical structure

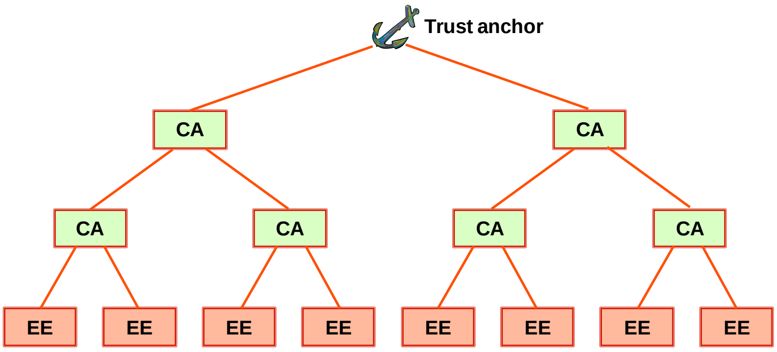


Figure 8 – Hierarchical PKI structure

Figure 8 shows a rather simple PKI structure. A trust anchor forms the root of a tree structure, followed by zero or more levels of intermediate CAs and with end-entities (EEs) at the leaves of the tree. This configuration makes it quite easy to establish a certification path.

6.2.2 Federated PKI structure

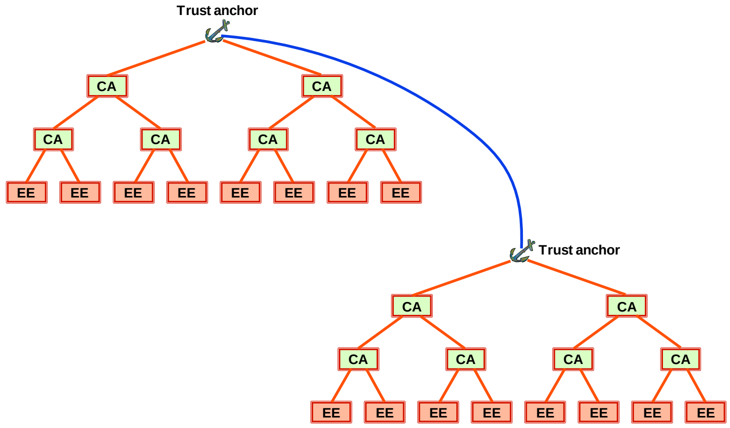


Figure 9 – Federated PKI structure

Figure 9 shows a more complex configuration where two (or more) tree structures are interconnected through cross-certification (see clause 6.4). In the figure, the trust anchors have issued CA-certificates to each other. They thereby allow for the establishment of a certification path that goes across the two otherwise disjoined PKI deployments. CAs lower in the trees could also establish cross certification to allow for alternative certification paths.

6.2.3 Mess PKI structure

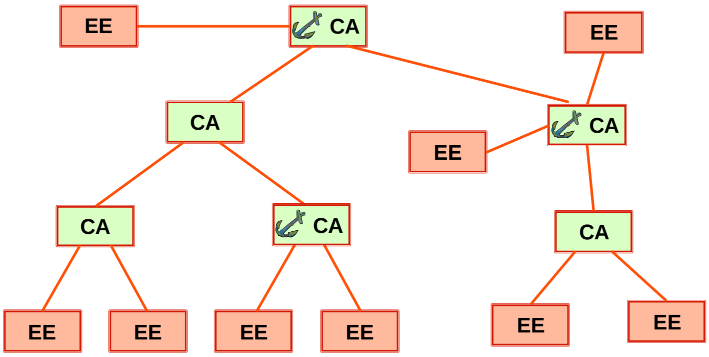


Figure 10 – Mess PKI structure

The situation may not be as simple as illustrated above. A relying party may not go all the way up to the trust anchor information, but may trust an intermediate CA without further checking. This may be a CA managed by the organization for the relying party.

Figure 10 shows such a complex example where there are several CAs in different places trusted by different relying parties, i.e. they function as trust anchors for such relying parties.

Many articles, specifications, etc., use the term root CA for a CA trusted by a relying party. This term is somewhat confusing, as it has the connotation of a structure as shown in Figure 8. For this reason, this term is not used by [b-ITU-T X.509] nor by this Technical Report.

6.3 Operational view

6.3.1 Closed PKI model

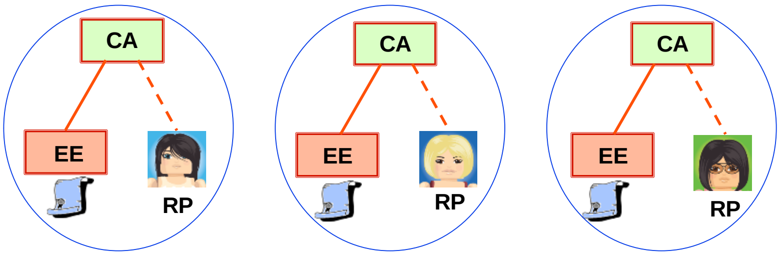


Figure 11 – Closed PKI model

This model reflects a closed community where there is some kind of relationship between a relying party and the CA(s) within that closed community. This implies that a relying party with a minimum of checking accepts public-key certificates by CAs within that community.

6.3.2 Open PKI model

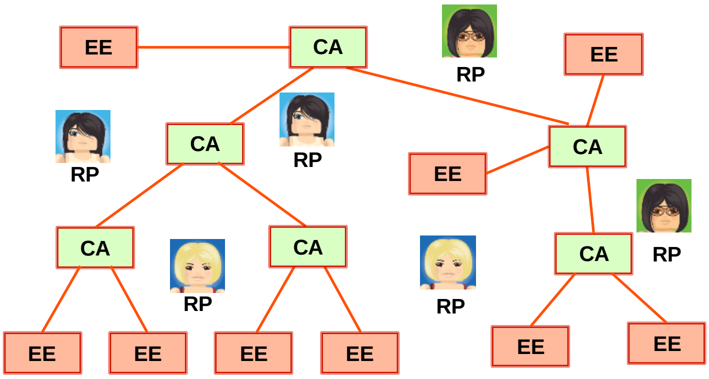


Figure 12 – Open PKI model

In an open model, a relying party may have no relationship with a CA that has issued a public-key certificate at hand. In this environment, a relying party needs to be more thorough in the checking of such a public-key certificate. It involves – among others – checking of the certificate policy document. As a certificate policy document is not machine readable, the software supporting a relying party is not able to analyse and make a decision on the content. This is left to a possible human user, which may not have the necessary background to make such an analysis. In such an environment, there is a need for some additional assistance, as discussed in clause 8.

6.4 Cross-certification

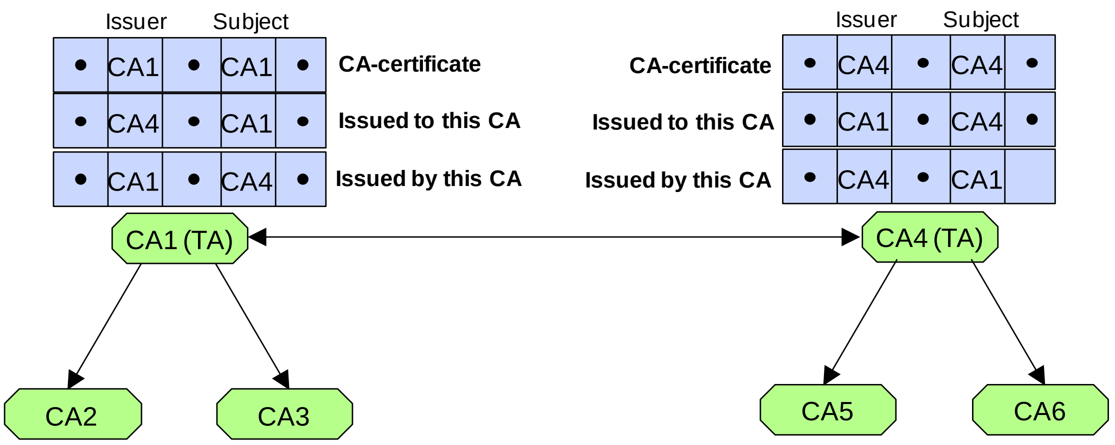


Figure 13 – Cross-certification

Cross-certification enables entities in one PKI domain to trust entities in another PKI domain. Cross-certification is established between two parts of a PKI that is otherwise separated by having two CAs in separate parts to issue CA-certificates to each other. This is illustrated in Figure 13, where CA1 has issued a CA-certificate to CA4 and CA4 has issued a CA-certificate to CA1. It is now possible to establish certification paths going from one PKI to the other.

In this example, the cross-certification is between two trust anchors (TA), but that does not need to be the case.

7 PKI deployment and procedures

7.1 Introduction

PKI is a powerful tool that can be used to provide secure authentication and authorization for security association and cryptographic key establishment. However, PKI can be difficult to deploy and operate. This is primarily because PKI standards (such as [b-ITU-T X.509]) only provide a high-level framework for public-key certificate usage and for implementing a PKI. They provide a mechanism for defining naming conventions, public-key certificate constraints, and certificate policies, but they do not specify how these should be used. These standards leave these details to the organizations implementing PKI, and working out these details is a rather complicated business.

This clause introduces some aspects to be considered when deploying a PKI.

7.2 Entropy

Entropy may be defined as a measure of the disorder or randomness in a closed system. In cryptography, entropy is a measure of the quality of a random number generator. Random number generators are used for key generation, nonce generation, etc. It is important that a random number generator create random numbers of pseudo-random nature, i.e., it should not be possible to predict where in the value range the next generated number falls. Any value should have equal probability within the value range. In addition, the value range should be sufficiently large.

7.3 Establishment of trust anchor information

In traditional web environments, trust anchor information is supplied by the browser vendor. The represented trust anchors are accepted as reliable trust anchors. The trust anchor information is typically supplied as a self-signed CA-certificate.

In other environments such as in an M2M environment where programmable entities are communicating, procedures are necessary for establishing access to trust anchor information.

It is not the intention here to specify procedures for establishing access to the necessary trust anchor information; it is sufficient to illustrate only the requirement for such procedures.

7.4 Establishment of CAs

Before a CA is established, its structure needs to be defined. The number of levels in the hierarchy of CAs affects the length of the certification path. The shortest certification path is obtained when the trust anchor directly issues the end-entity public-key certificates.

An excessive length of the certification path affects performance:

- More information is transmitted from the entity, whose public-key certificate is to be validated, to the relying party (see for example Figure 6).
- The validation process becomes more demanding when processing.
- The security becomes diluted for long certification paths.

The CA policies need to be established.

The specification of the content of CA-certificates needs to be defined.

The signature algorithm to be used for signing public-key certificates shall be selected.

Safety of the CAs' needs to be established such as physical safety, firewalls, personnel clearance, etc.

7.5 Key generation

How keys are generated and distributed need to be defined. Hardware modules for key generation and storage are one option for consideration.

7.6 Provisioning of public-key certificates

The content of a public-key certificate needs to be established:

- Version: The version shall be v3.
- Serial number: This shall be an integer. Consider using random serial numbers instead of sequential numbers. The same sequence number shall not be used twice for public-key certificates issued by a given CA. The range of the serial numbers should not be excessively large, but large enough for the CA to issue unique numbers during its lifetime.
- Signature: This shall be an object identifier for the selected algorithm as indicated in clause 7.4.
- Issuer: To reduce size and processing, the name of the issuer in a constraint environment should be as short as possible without losing global uniqueness. It has to be a directory distinguished name, but should it has as few naming components as possible.
- Validity: Two to three years is the typical validity time for a public-key certificate issued to a human being. It may be different for entities within some networks. There is literature indicating that 10 years may be appropriate in stable environments. The length of a CRL may be excessive for long validity periods. If, for example, the validity period is 10 years and the public-key certificate is revoked after one year, then the public-key certificate remains on the CRL for nine years.
- Subject: As for the issuer field, to reduce size and processing, the name of the subject in a constraint environment should be as short as possible without losing global uniqueness. It has to be a directory distinguished name, but should have as few naming components as

possible. An empty name could be considered if the subject alternative name extension is included (see clause 4.3.2).

- Subject key info: This field holds the signature algorithm and the public key. The algorithm and the key length should be selected in such a way that the public key-pair is assumed safe for the validity period.
- Issuer Unique ID, and Subject Unique ID are deprecated fields that are typically in no use anymore.
- Extensions: Consider the number and types of extensions carefully to keep down the size of a public-key certificate.

There are different types of public-key certificates with different characteristics:

- CA-certificates, including CA-certificates for trust anchors;
- end-entity public-key certificates installed by manufacturers of entities (to be used for some kind of bootstrapping);
- end-entity public-key certificates for non-human entities;
- end-entity public-key certificates for human users.

7.7 Validation procedures

Before starting to validate a public-key certificate, the relying party establishes the certification path to that public-key certificate. If it is not possible to establish a certification path, validation is not possible and the public-key certificate in question could be rejected.

When the action of a relying party is dependent on the content and validity of a public-key certificate, this public-key certificate has to be validated by checking the following criteria and properties:

– Whether the signatures on the public-key certificates of the certification path all are valid.

– Whether any of the public-key certificates in the certification path, including the trust anchor information, has expired.

– Whether any of the public-key certificates of the certification path has been revoked by checking against a certificate revocation list or by contacting an OCSP responder.

– Whether the issuing CA and the chain of CAs up to the trust anchor are trustworthy.

– Whether all possible constrains are honoured.

– A relying party may have local policies to be honoured, for example, presence or non-presence of extensions, whether the subject name has the right format, etc. Standard software such as browsers will not be able to make such checks based on local policy.

NIST has specified a PKI validation test suite in [b-NIST PKITS].

7.8 Revocation of certificates

A CA may revoke a public-key certificate that is no longer safe to use. There can be several reasons for that:

– The owner of the public-key certificate has reported that the private key has possibly been compromised.

– The CA suspects that a public-key certificate, which it has issued, is used for devious purposes.

– The signature algorithm used for signing the public-key certificate has been deprecated.

– The entity or person for which a public-key certificate has been issued is leaving the organization or has ceased to have responsibilities for the role for which the public-key certificate was issued.

8 Addition of new entity type in the PKI

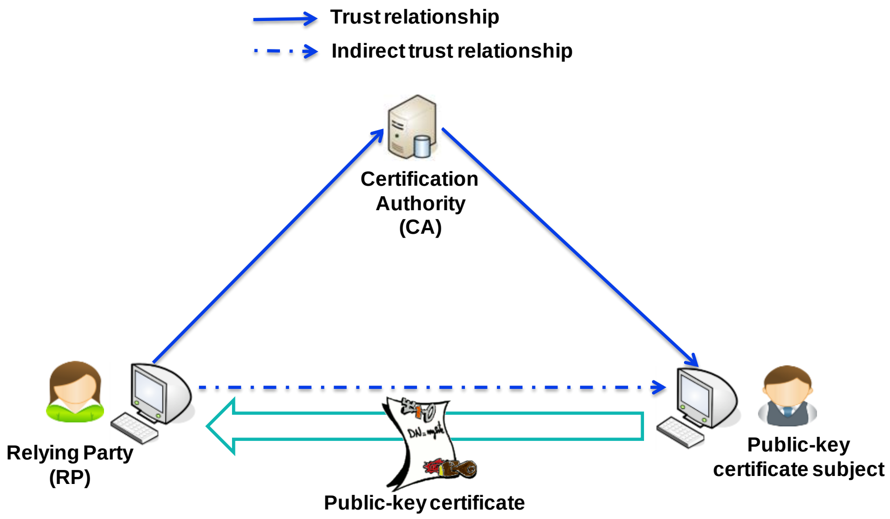

Trust relationship
Indirect trust relationship
Certification Authority (CA)
Relying Party (RP)
Public-key certificate subject
Public-key certificate

Figure 14 – Three-corner trust model

Figure 14 shows the traditional three-corner trust model that is applicable in particular for the closed model discussed in clause 6.3.1. In this trust model, the relying party (RP) has a trust relationship with the CA that issued the public-key certificate.

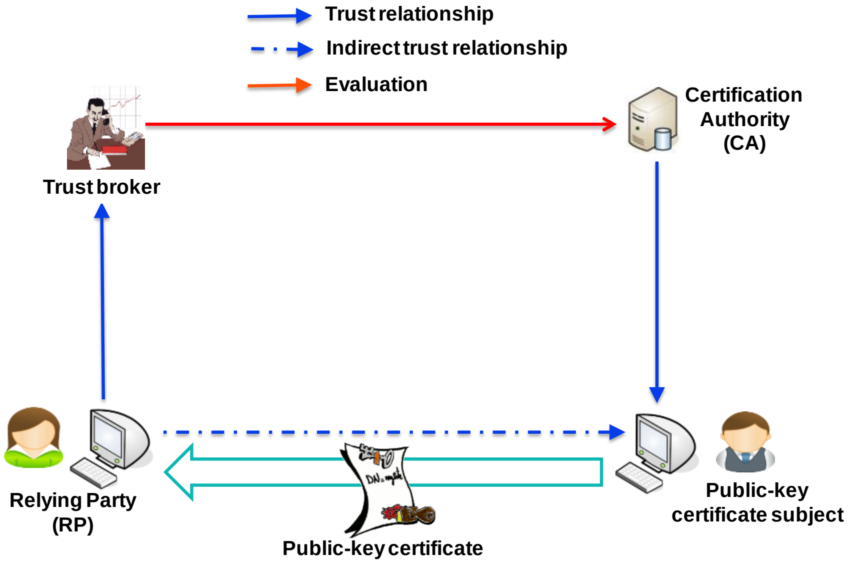

Trust relationship
Indirect trust relationship
Evaluation
Trust broker
Certification Authority (CA)
Relying Party (RP)
Public-key certificate subject
Public-key certificate

Figure 15 – Four-corner trust model

In the open four-corner trust model, as depicted in Figure 15, there is not such a shared relationship and the relying party has quite a challenge in checking the validity of a public-key certificate. In particular, a public-key certificate may have reference to one or more certificate policies referenced by the certificate policies extension (see clause 4.3) that specifies under which conditions the public-key certificate has been issued and the general use of the public-key certificate. The software supporting the relying party function is not able to evaluate the certificate policy and neither is the average user, and most software will not even allow a human user have a chance to do this checking. However, in the four-corner trust model, the relying party has a trust relationship with a so-called trust broker. This trust broker can then on behalf of several relying parties obtain the necessary validation information and could, if customized to a particular environment, do a more thorough checking; for example, check against the presence of mandatory, optional or forbidden public-key certificate extensions.

It is the plan to extend [b-ITU-T X.509] to include a specification for such a component type.

9 Digital signatures

9.1 Abstract syntax notation one (ASN.1) encoded digital signatures

The information provided in an ITU-T X.509 digital signature, as depicted in Figure 1, is not sufficient for a relying party to verify the signature. The relying party needs some additional information, such as the public-key certificates of the complete certification path.

In the ITU-T X.500 protocols, such additional information is provided by separate security parameters.

[b-IETF RFC 5652] specifies a more extensive digital signature structure, as illustrated in Figure 16. This cryptographic message syntax (CMS) format provides for multiple, parallel digital signatures by different signers meaning that more than one signer be committed to the signed content.

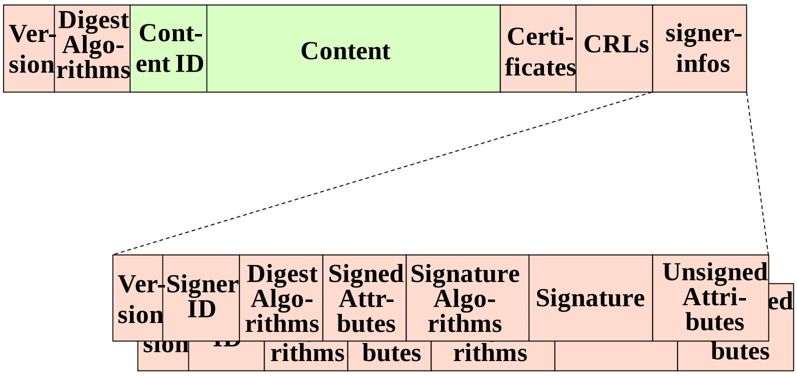


Figure 16 – CMS signed data

The structure may vary depending on the fields that are included and how they are used:

- The version field gives an indication on the kind of signed structure.
- The digest algorithms field gives the list of digest (hashing) algorithms used for the different signatures.
- The content is the data that is signed with the content ID indicating the type of data.
- The set of certificates includes a list of public-key certificates used for the evaluation of the different signatures. It may also include attribute certificates. If the set is not complete or absent, it is assumed that the certificates are available by other means.
- The CRLs field, if present, holds certificate revocation list information necessary for the evaluation of the public-key certificates. If the set is not complete or absent, it is assumed that CRLs are available by other means, for example, as an address in the public-key certificate CRL distribution points extension. Revocation information might also be available from an OCSP server.
- The signer-infos field holds a set of signer info subfields, one for each signer. The version field indicates the syntax of subfield.
  - The signer ID is an identification of the signer.
  - The digest and the signature algorithm are indicated.
  - Additional information is provided in attributes, such as signing time, digest of content, etc. Some of the attributes are part of the information that is protected by the signature, while other attributes are not protected as such.

10 Message authentication codes

10.1 Relevance

The concepts of *message authentication code (MAC)* and its derivate *hash-based message authentication code (HMAC)* are relevant in the context of this Technical Report, as they are both alternative ways to ensure integrity and authenticity. Both techniques are dependent on a shared secret key and are relevant in a key management context, for example, in TLS.

MAC and especially HMAC are used in lieu of electronic signatures, as they require less processing, which might be important in a restricted environment, especially within wireless PKI and smart grid.

MAC and HMAC do not provide non-repudiation.

MAC and HMAC are defined in [b-ISO/IEC 9797-1] and [b-ISO/IEC 9797-2].

10.2 Message authentication code (MAC)

A MAC is a cryptographic checksum of a piece of data using a *MAC algorithm*.

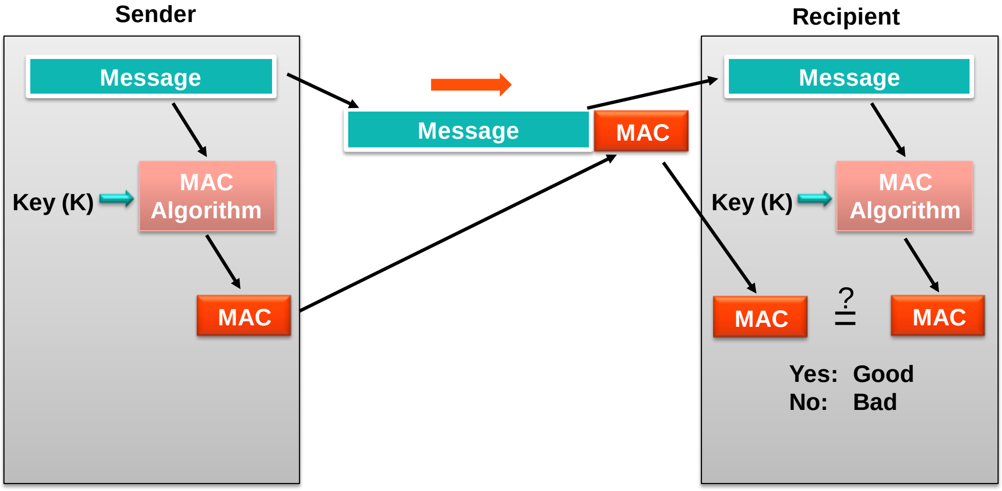


Figure 17 – Message authentication code

A bit string (message) to be sent from the sender to the recipient may be protected by a MAC.

A MAC differs from a digital signature because a MAC value is both generated and verified using the same secret MAC key. This implies that the sender and receiver of a message must agree on the same key before initiating communication.

MAC requires the use of the secret MAC key to generate a small fixed-size block of data, known as a *cryptographic checksum* or MAC. This MAC is then appended to the message as shown in Figure 17. A MAC algorithm is similar to a symmetric cryptographic algorithm, except that a MAC algorithm does not need to be reversible, that is, there is no need for decryption.

The recipient using the same MAC algorithm and the same MAC key can generate its own version of the MAC and compares it with the received MAC. If equal, it can be assumed that the message has not be changed during transmission and that it comes from the alleged sender, i.e., it provides for authentication and message integrity.

A MAC algorithm is a many-to-one function, since potentially many arbitrarily long messages can be condensed to the same MAC value. However, a good MAC algorithm is designed to satisfy the following properties:

- It is one-way algorithm, i.e., it is computationally infeasible to find a message that maps to a specific MAC.

– It is infeasible to modify a message without changing the MAC.

– It is collision resistant, i.e. it is computationally infeasible to find any two distinct messages that map to the same MAC.

10.3 Keyed hash-based message authentication code (HMAC)

Keyed hash-based message authentication code (HMAC) is a variant of MAC where a hashing algorithm is applied instead of a cryptographic algorithm. It requires less processing to generate HMAC compared to a MAC. HMAC involves a shared secret key, but the key is not used for encryption. It is combined in a rather simple way with the message for which a HMAC is to be created.

11 Hardware security module (HSM)

A hardware security module (HSM) is a secure crypto processor with the main purpose of managing cryptographic keys and offering accelerated cryptographic operations using such keys.

An HSM can perform a number of important security-related functions. It provides accelerated cryptographic operations such as encryption, digital signatures, hashing, MAC generation and backup of sensitive material in encrypted form.

[b-NIST FIPS 140-2] specifies the security requirements on a cryptographic module. It specifies four levels of security.

[b-PKCS#11] is a widely accepted standard for interfacing HSMs by specifying a C-language interface.

12 Key management

12.1 Importance of key management

If the same shared secret key were used on all communication channels, the compromise of that key would put the whole network in jeopardy. It is therefore essential that different shared secret keys be used on different communication channels.

A good practice is to use proven and widely used techniques rather than to develop specific techniques.

12.2 PKI relationship with key management

PKI is used in many different areas, but in most of those areas it is employed as part of the management of symmetric cryptographic keys in an environment requiring encryption of transferred (or stored) information. It is therefore useful to consider key management, including TLS, to analyse how different solutions are dependent on a PKI and what the consequent requirements are on PKI.

12.3 Concept of key management

Management of cryptographic keys is the foundation and the big challenge for security relying on cryptographic techniques.

Key management covers everything related to cryptographic keys except for the actual encryption and decryption: it covers the creation of keys, activation and deactivation of keys, transport of keys, storage of keys, destruction of keys, etc.

Key management is a major issue that has been the subject for much standardization activity by many different standardization groups.

12.4 Master/session scheme

Master/session scheme is a key management scheme in which a shared secret key, called the *master key*, in some secure, but unspecified way has been planted in two entities to communicate. One of the entities creates another shared key, called the *session key*, encrypts it using the master key and transmits it to the other entity. The session key is then subsequently used for encrypting and decrypting data to be exchanged.

The session key is renewed at certain time intervals and the same session key is never reused. This technique provides perfect forward security (see clause 15) if only the session key is compromised. If the master key is compromised, it takes some offline procedures to recover.

In general, this technique is only applicable in a very limited environment with only a rather small amount of communicating entities. The master keys have to be different for each connection; otherwise, a compromise of the master key will have serious consequences. If an entity has communication with several other entities, it needs a master key per connection.

12.5 Key distribution centre

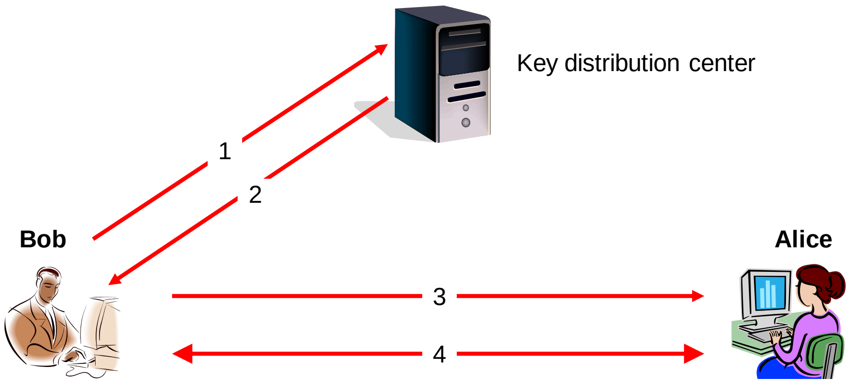


Figure 18 – Use of key distribution centre

The use of a key distribution centre (KDC) is shortly considered here, although it has no direct relevance to PKI. It might be seen as an obvious and easy solution to key management.

In this solution, a key distribution centre has established different shared secret keys with all the entities it serves. In this way, the key distribution centre can have a secure communication with all the entities. Using this secure communication, a shared secret key can be established for communication between two entities. This is illustrated in Figure 18 by using an example with the two human entities, Bob and Alice.

When Bob wants to establish an encrypted communication with Alice, he has to do the following steps:

1. Bobs sends a request to the key distribution centre indicating that he wants to communicate with Alice. This request is encrypted using the shared secret key between Bob and the key distribution centre.
2. The key distribution centre returns a shared secret key *K* to be used in the communication with Alice. It also returns the secret key *K* and the name of Bob encrypted with the shared key between Alice and the key distribution centre. All this is again encrypted with the shared key between Bob and the key distribution centre.
3. Bob sends to Alice the part that is encrypted by the key shared between Alice and the key distribution centre. Alice can now see that it came from Bob and she has the shared the secret key *K* to be used in the communication with Bob.
4. Bob and Alice may now engage in an encrypted communication using the shared key *K*.

This technique is mostly applicable in a limited and rather static environment. It does not scale very well, as the establishment and management of shared secret keys between the key distribution centre and all the entities it serves imply a major logistic and security problem in a dynamic and/or large network.

12.6 The Diffie-Hellman key agreement method

The Diffie-Hellman key agreement method is defined in [b-IETF RFC 2631]. It allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communication channel in a way that does not allow an eavesdropper to learn about the key. This key can then be used to encrypt subsequent communications using a symmetric key cipher. This is illustrated in Figure 19.

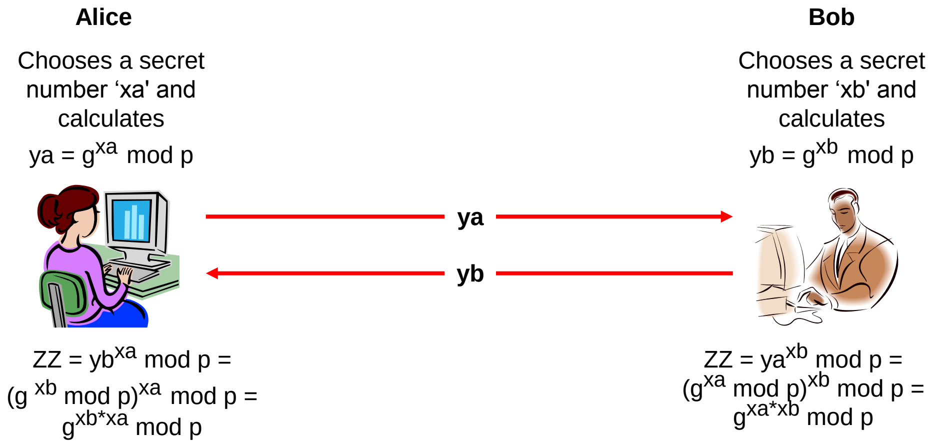

Alice
Chooses a secret number 'xa' and calculates
$ya = g^{xa} \bmod p$
Bob
Chooses a secret number 'xb' and calculates
$yb = g^{xb} \bmod p$
ya
yb
$ZZ = yb^{xa} \bmod p = (g^{xb} \bmod p)^{xa} \bmod p = g^{xb*xa} \bmod p$
$ZZ = ya^{xb} \bmod p = (g^{xa} \bmod p)^{xb} \bmod p = g^{xa*xb} \bmod p$

Figure 19 – Diffie–Hellman key agreement exchange

In Figure 19, Bob and Alice want to establish an encrypted connection between them and therefore need to establish a shared secret key. The technique requires Bob and Alice to have a prior agreement on two integers, a (large) prime number '*p*' and another integer '*g*' (generator). [b-IETF RFC 2631] specifies in details how these values should be selected.

The following steps are taken:

- Each of the two parties selects independently a secret integer. Alice selects '*xa*' and Bob selects '*xb*'.
- Each of the parties calculates a value as shown in the figure. Alice calculates a value '*ya*' and Bob calculates a value '*yb*'. Bob and Alice then exchange those values. The shared integers '*p*' and '*q*' have to be selected in a way that makes it infeasible to deduce the secret integers chosen by Bob and Alice, even when an attacker knows the public values '*p*' and '*q*' and can intercept the values '*ya*' and '*yb*'.
- Each of the participants combines the received values with its own secret value as also shown in the figure. This is done in such a way that both parties end up with the same value '*ZZ*', a common secret that can be used to generate a shared secret key.

The values '*xa*' may be considered the secret key and '*ya*' can be considered the public key for Alice, although these keys cannot be used for encryption and decryption. They are only used for key agreement. The associated key algorithm is called the *Diffie-Hellman key algorithm* and its object identifier and associated parameters are defined in [b-IETF RFC 3279], which also defines its associated parameter, including the values '*p*' and '*g*'.

The Diffie-Hellman key agreement algorithm is considered secure against eavesdroppers, as an eavesdropping attacker cannot deduce the shared key. However, it is subject to a man-in-the-middle attack, as shown in Figure 20.

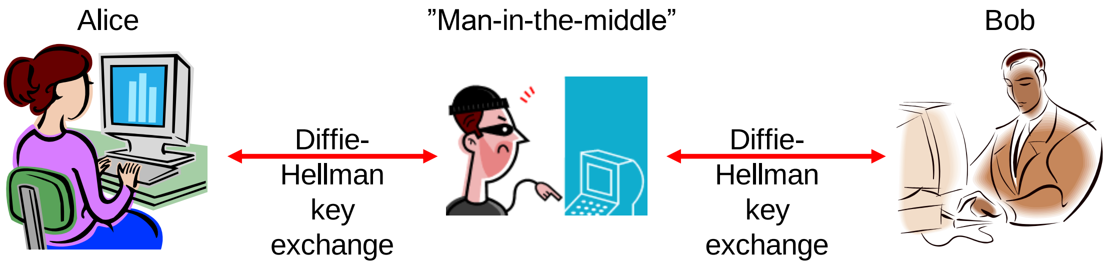


Figure 20 – Man-in-the-middle attack

The Diffie–Hellman agreement by itself does not provide authentication of the communicating parties and is thus vulnerable to a man-in-the-middle attack. An attacker in the middle may perform two Diffie-Hellman agreements, one with Alice and another one with Bob, effectively masquerading as Alice to Bob, and vice versa, allowing thus the attacker to decrypt (and read or store) then re-encrypt the messages passed between them. Mutual authentication of the communicating parties can prevent this type of attack.

[b-ITU-T X.1035] specifies a protocol where the two parties share a common secret password. This password is included in the key exchange procedure to ensure authentication. Whilst this protocol solves the man-in-the-middle problem, it seems to replace a key management problem with a secret password management problem.

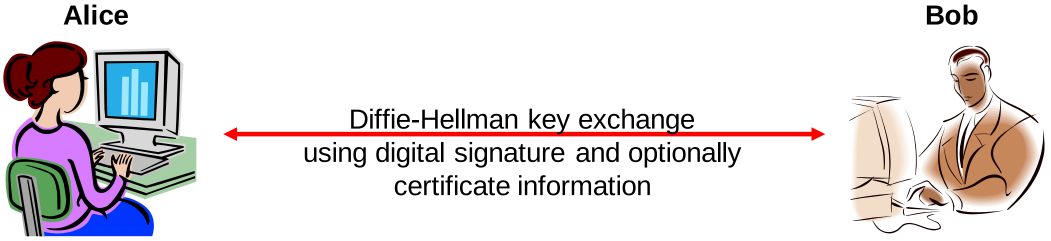

Alice
Bob
Diffie-Hellman key exchange
using digital signature and optionally
certificate information

Figure 21 – Diffie-Hellman with PKI deployment

Another method involves the use of asymmetric keys, which implies the deployment of PKI.

By digitally signing the Diffie-Hellman messages, Bob will detect that Alice did not sign the received messages, and Alice will detect that Bob did not sign the received messages. Bob and Alice will then abandon the communication.

12.7 RSA key exchanges

The Rivest-Shamir-Adleman (RSA) algorithms are widely used public key cryptography algorithms. For RSA key exchange, one party generates a secret symmetric key or premaster secret for TLS, then encrypts it using the public key of the remote partner and transmits it to that partner. Only the intended recipient is able to decrypt the secret key. An interceptor cannot decrypt and use the secret key or premaster secret.

This technique requires the presence of a PKI to ensure that the originator of the operation obtains in a secure way the identity of the recipient by obtaining the public-key in the public-key certificate issued to that recipient.

13 Transport layer security (TLS)

13.1 TLS overview

Many applications that require confidentiality and authentication use transport layer security (TLS). TLS is dependent on key management and key management is dependent on a functional PKI.

The primary goal of TLS and its predecessor, secure socket layer (SSL), is to provide privacy (confidentiality), data integrity and authenticity between two communicating entities. The latest version is TLS 1.2 as defined by [b-IETF RFC 5246], and it is the basis for this section.

The TLS (or SSL) is used by many applications. However, it is used to secure the communication between a web browser and a web server. The protocol supporting web communication is hypertext transfer protocol (HTTP). When used together with TLS/SSL, it is signalled as hypertext transfer protocol secure (HTTPS). As seen later, authentication of a web server requires that the web server has a public-key certificate issued for that purpose. Many organizations issue what is often called SSL certificates for use in web communication. TLS is used for many other applications where the public-key certificate requirements may be different. Examples of other uses are:

– The lightweight directory access protocol (LDAP) as defined by [b-IETF RFC 4511].

– The four protocols defined by [b-ITU-T X.519].

– The supervisory control and data acquisition (SCADA) protocols as defined by IEC TC 57.

TLS is asymmetric in the sense that it defines protocols between a server and a client. In some environments, this makes sense when a web browser, for example, accesses a website. In other environments, such as a machine-to-machine (M2M), the roles are not predetermined. The client and server roles have to be agreed on in some way.

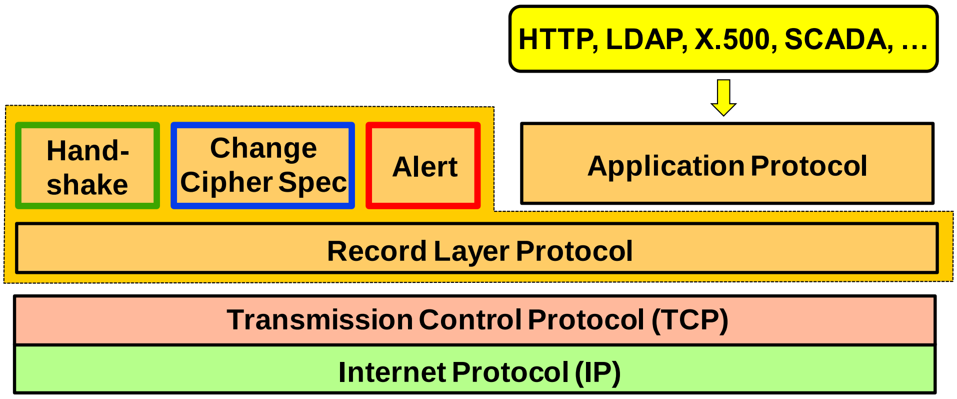


Figure 22 – TLS model

The TLS model is shown in Figure 22. It is composed of two layers. The TLS record layer protocol is placed on the top of the transport protocol, typically the transmission control protocol (TCP). The TLS handshake, change cipher specification and the alert protocols are placed on top of the record layer protocol. The handshake protocol is described in details later. The change cipher specification protocol is also described later but in simple terms only. The alert protocol defines a set of alert messages to be returned in exceptional conditions.

In addition and on the top of the record layer protocol are the application protocols that are users of the service provided by TLS. The TLS record layer protocol provides the TLS service as seen from the application protocols. The record layer provides security by encrypting transmitted messages. Integrity and authenticity of messages are provided by the use of HMAC (see clause 10.3). (In principle, also a MAC as defined in 10.2, could be used, but all cipher suites defined by TLS 1.2 specifies HMAC).

The TLS handshake protocol allows the server and client to authenticate each other, to negotiate encryption and hashing algorithms, and to negotiate cryptographic keys as needed for the encryption and handling of HMAC.

TLS supports three authentication modes: authentication of both parties, server authentication with an unauthenticated client, and total anonymity. Whenever the server is authenticated, the channel is secure against man-in-the-middle attacks, but completely anonymous sessions are inherently vulnerable to such attacks. Anonymous servers cannot authenticate clients. If the server is authenticated, it must provide a valid public-key certificate chain leading to an acceptable trust anchor. Similarly, authenticated clients must also supply a public-key certificate chain to the server. Each party is responsible for verifying that the other's public-key certificates are valid, which requires the deployment of a PKI with all its implications.

13.2 Transmission formats

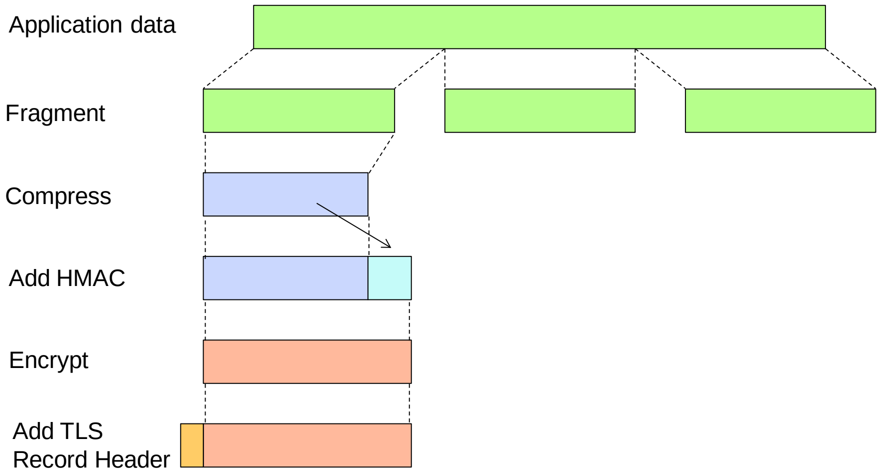


Figure 23 – Record layer protocol operation

Figure 23 illustrates the record layer protocol handling of data received from the layer above. The record layer protocol may fragment the data into smaller pieces. Each fragment may then be compressed. A HMAC is then added to the compressed fragment. Finally, the fragment with the HMAC is encrypted and a TLS header is added. This header is not encrypted.

The TLS header consists of:

- content type, which indicates whether the content is change cipher spec, handshake, alert or application data;
- protocol version (3.3);
- length of fragment in octets.

13.3 Key generation

The prime purpose for TLS is to produce a set of cryptographic keys in a secure way. For each direction, a MAC key, a bulk encryption key and for some symmetric key algorithms (authenticated encryption with associated data (AEAD) cipher), an initialization vector (IV) is generated.

The exchanges, as described in clause 13.8, enable the creation of a so-called shared *premaster secret*. A *master secret* is generated from the premaster secret and some exchanged random values. The key material is then generated by expanding the master secret by a repeated use of a specific hashing algorithm.

13.4 Concepts of connection and session

TLS connection and the TLS session are two important TLS concepts:

- Connection: A connection is a network transport that provides a suitable type of service. Such connections are transient, peer-to-peer relationships, associated with one particular session.
- Session: A TLS session is an association between a client and a server, created by the handshake protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

13.5 Cipher suites

A cipher suite consists of:

- a key exchange algorithm for generation of shared keys;
- a symmetric key algorithm for bulk encryption/decryption;
- a MAC algorithm for MAC handling.

All the cipher suites to expand the master secret into key material use the pseudorandom function (PRF) P_SHA-256.

A cipher suite is in TLS identified by a one-octet code.

The initial cipher suite is one that does not have a key exchange, a bulk cipher or a MAC specification. It is not allowed to negotiate that initial cipher suite.

Several cipher suites do not provide for authentication, but just have a pure anonymous Diffie-Hellman key agreement.

A specification that specifies the use of TLS will typically limit the allowed repertoire of cipher suites.

13.6 Compression methods

[b-IETF RFC 3749] specifies the compression methods for TLS. It is possible to specify a null_compression method (i.e., no compression will be applied then).

13.7 Extensions

TLS allows applications to specify extensions that add functionality to the TLS protocol. Both TLS clients and servers may set TLS extensions. The use of TLS extensions is backwards-compatible, that is, communication is possible between TLS clients that support TLS extensions and TLS servers that do not support TLS extensions, and vice versa.

The base TLS specification [b-IETF RFC 5246] provides a single TLS extension:

- Signature algorithms, which allow a client to inform the server the signature algorithms that the client supports.

[b-IETF RFC 5746] specifies an extension that counters against a renegotiation attack as described in clause 13.9.4. It does so by adding information that allows a server to distinguish between an initial handshake and a renegotiation handshake.

[b-IETF RFC 6066] defines more additional TLS extensions:

- server_name: Allows a client to provide the name of the server it is contacting.
- max_fragment_length: Allows a client and a server to negotiate the maximum fragment length to adapt to client memory and bandwidth constraints.

- client_certificate_url: Allows a client and a server to negotiate the use of client certificate URLs. A client may supply a URL where public-key certificates may be found in case the client is too constrained to hold this information.
- trusted_ca_keys: Allows a client to indicate to the server which trust anchor keys it possesses. Inclusion of this extension may prevent unnecessary handshaking.
- truncated_hmac: Allows a client and a server to negotiate the use of truncated message authentication codes (MACs) to conserve bandwidth.
- status_request: Allows a client and a server to negotiate if the server could send the client certificate status information (e.g., an OCSP response) during a TLS handshake. This functionality is sometimes referred to as OCSP stapling.

13.8 Handshake and change cipher specification protocols

13.8.1 Introduction

As many attacks on TLS are related to the sequence of exchanges, an understanding of these exchanges is therefore needed to understand the different types of attack.

The Handshake protocol is described without going into implementation details. As the change cipher specification exchange is interspersed with the Handshake exchanges, this exchange is also described here.

13.8.2 TSL handshake negotiation - 1

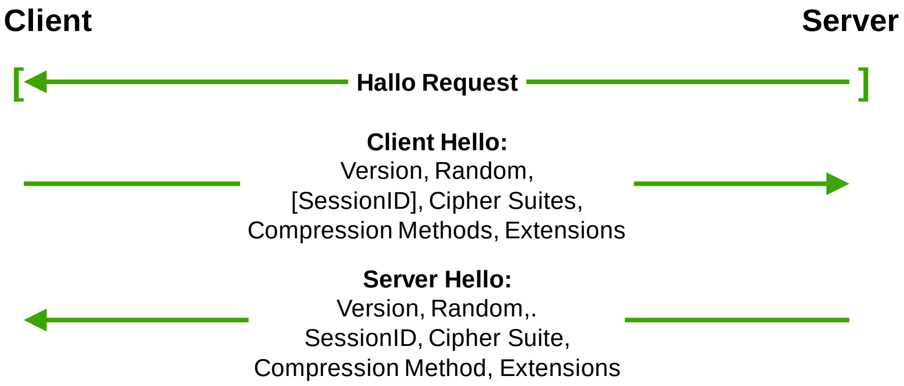


Figure 24 – TLS handshake negotiation - 1

Figure 24 illustrates the initial negotiation. When this part of the negotiation has completed, the two parties would have agreed on the cipher suite, the compression method and the extensions to be used.

The client starts the negotiation. However, the server may ask the client to start the negotiation by sending a Hello Request.

The Client Hello message sent by the client has the following content:

- The version parameter holds the TLS version the client requests to be used. The version number for TLS 1.2 is 3.3. As TLS is a successor of SSL, the version numbers are a continuation of the SSL version numbers. The last SSL version is 3.0. TLS version 1.0 therefore got version number 3.1, while TLS version 1.1 obtained version number 3.2.
- The random parameter holds the current time plus a random value changed for each occurrence. This parameter allows detection of replay and it is used when generating a key from the master secret (see 13.3).

- The SessionID parameter, if not empty, holds an identifier for a previous session, whose security parameters are to be reused. This mode is used to refresh the keys after a lapse of time.
- The cipher suite parameter holds a set of cipher suites that the client supports and is willing to use.
- The compressions parameter holds a set of compression methods that the client supports and is willing to use.
- The extensions parameter, if not empty, holds the set of extensions the client wants to use.

The Server Hello message returned by the server has the following content:

- The version parameter holds the TLS version the server wants to use. This version cannot be higher than the one suggested by the client.
- The random parameter holds the current time plus a random value independent of the client random value. This parameter allows detections of the replay and it is used when generating a key from the master secret (see clause 13.3).
- If the SessionID parameter is equal to the non-empty value of the corresponding client Hello value, then the old session will be resumed, which means that the server accepts renegotiation of the security parameters and both parties skip many of the following interactions and proceed to the Finish messages. Otherwise, a new complete handshake has to be performed.
- The cipher suites parameter holds the cipher suites that the server has selected from those proposed by the client.
- The compression parameter holds the compression methods that the server has selected from those proposed by the client.
- The extensions parameters, if not empty, holds the set of extensions that the server has selected from those proposed by the client.

13.8.3 TLS handshake negotiation - 2

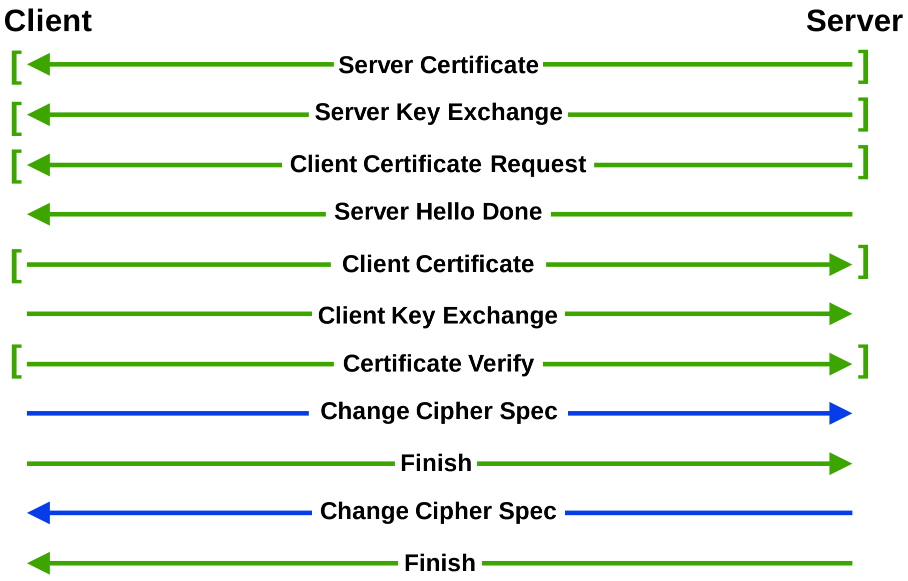


Figure 25 – TLS handshake negotiation - 2

Figure 25 shows the continued handshake sequences:

- The server will send the server's public-key certificate chain (a certification path in the reverse order) if the cipher suite does not specify an anonymous Diffie-Hellman key agreement.
- The server shall then send the Server Key Exchange message unless the exchanged server public-key certificate holds sufficient information in the public key information field to do key exchange. This message includes the prime integer '*p*' and the generator '*g*' for Diffie-Hellman key agreement as shown in Figure 19. It also includes the Diffie-Hellman public value as calculated by the server (for example, '*ya*' in Figure 19).
- If the server requires the authentication of the client, the server sends the Client Certificate Request message.
- The server sends the Server Hello Done Message to signal that the server has completed the Server Hello sequence.
- Having received the Server Hello Done message, the client sends a Client Certificate message, if so requested by the server. If no suitable public-key certificate is available, the client sends an empty message. Otherwise, the client includes a client certificate chain.
- The client always sends the Client Key Exchange message. If an RSA exchange is included in the negotiated cipher suite, a premaster secret generated by the client is transmitted. Otherwise, the client sends its calculated Diffie-Hellman public value.
- The client sends the Certificate Verify message when the client's public-key certificate specifies signing capabilities. In this case, a signature is created across all the previous Hello messages. This message provides explicit verification of the client's public-key certificate.
- The client issues the Change Cipher Spec message to indicate that subsequent messages sent from the client will be protected based on the established agreement.
- The Finish message is the first message protected under the new agreement. The content is a hash of the exchanged Hello messages.
- The server likewise sends a Change Cipher Spec and Finish messages.

After completion, application data transfer can start.

13.9 Specific TLS attacks

13.9.1 TLS downgrade attack

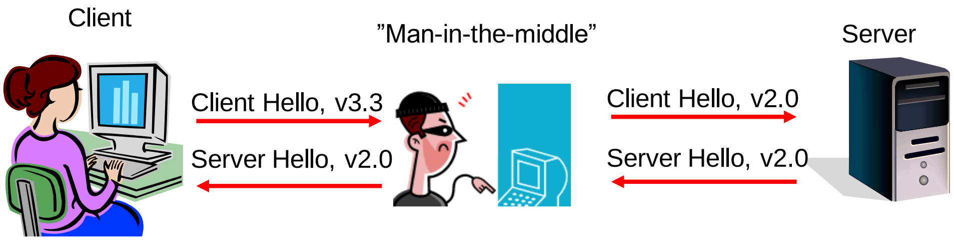


Figure 26 – TLS downgrade attack

The TLS downgrade attack is a type of man-in-the middle attack, where the communication is intercepted before encryption and authentication have been established. In Figure 26, Alice suggests using TLS version 1.2, but the man-in-middle changes this request to SSL version 2.0, which is much easier to attack. The server agrees with that proposal. Alice is made to believe that the server only supports SSL version 2.0 and may accept this proposal.

13.9.2 Prohibiting SSL version 2

[b-IETF RFC 6176] defines procedures a server shall observe to avoid the SSL version 2 to be negotiated. SSL version 2 is known to have deficiencies and should not be used. [b-IETF RFC 6176] describes the type of threats that have been identified against SSL version 2 and specifies that a compliant implementation shall not accept SSL version 2.

13.9.3 HTTP strict transport security (HSTS)

[b-IETF RFC 6797] defines HTTP strict transport security (HSTS). The primary benefit is that it makes it harder for attackers to conduct man-in-the-middle attacks by tricking browsers into using HTTP to transmit packets instead of HTTPS.

13.9.4 TLS renegotiation attack

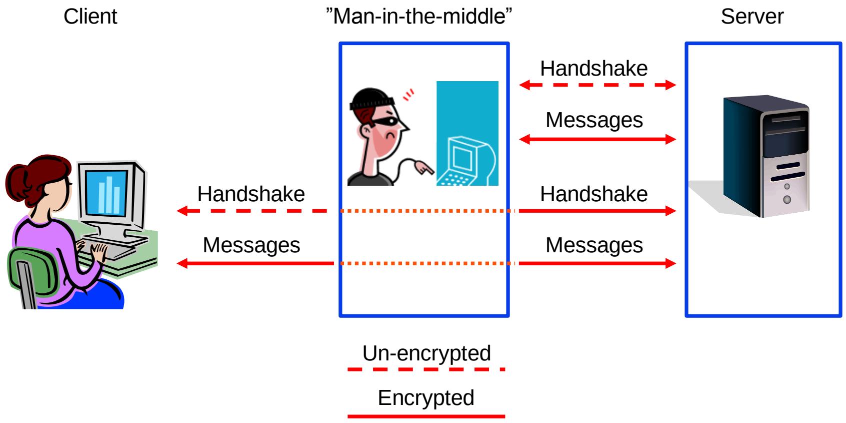


Figure 27 – TLS renegotiation attack

The flaw arises due to the manner TLS end-points negotiate a common cipher suite. As well as allowing the cipher suite to be negotiated at the beginning of a session. TLS allows renegotiation dynamically, at the instigation of either the client or the server, half way through a session. The problem is that the procedures for negotiation and renegotiation are identical. Therefore, one party (e.g. the victim client) may believe it is involved in an initial negotiation, while the other party (e.g. the server) may believe it is involved in a renegotiation. This can occur when a man-in-the-middle establishes an unauthenticated session with a server and then proxies another authenticated session between the client victim and the same server. In this situation, it is possible for the man-in-the-middle to trigger a renegotiation and for the last message received from the man-in-the-middle by the server prior to the renegotiation to be attached to the first message received from the victim client after the renegotiation. Depending upon the design of the web application, this vulnerability may be exploitable by the man-in-the-middle to execute a transaction that will be attributed to the victim client.

There is a TLS extension as defined by [b-IETF RFC 5746] that is intended to guard against this type of attack (see clause 13.7).

13.9.5 TLS renegotiation and denial of service attack

When a new TLS connection is being negotiated, the server will typically spend significantly more processing resources than the client will. Thus, if a client is requesting many new TLS connections per second, it may exhaust the processing capability of the server.

14 Wireless transport layer security

The former Wireless Application Protocol (WAP) Forum, now part of the Open Mobile Alliance (OMA), has issued a version of TLS for the mobile environment called wireless TLS (WTLS).

WTLS has been defined for a constraint mobile phone environment with limited storage, processing capability and bandwidth. As the same limitation may be present in other environments, the WTLS concepts may be applicable for other constraint environments.

It is not expected for WTLS to have the same history of hostile attacks as TLS. It may therefore need a thorough review to check its resistance to all known TLS attacks.

WTLS is similar to TLS with the following main differences:

– WTLS has datagram (connectionless) support and therefore supports sequence numbers to detect missing, duplicate or out of order messages. This allows the short message service (SMS) to be the underlying transport system.

– While TLS has a recommendation for the maximum time between renegotiation of secrets, WTLS may – in the protocol – specify the maximum number of messages to be exchanged before renegotiation shall occur.

– WTLS is optimized for low bandwidth environments.

– WTLS may support a very limited handshake utilizing some common secrets that have been placed in the parties some way ahead of the WTLS exchange.

– WTLS does not use fragmentation.

15 Secure associations

15.1 Purpose of secure associations

Secure associations and the Internet protocol security (IPsec) use of secure associations is an alternative to TLS. It is therefore useful to understand IPsec and to know how it may or may not fit into an overall PKI security architecture.

IPsec is also dependent on PKI; as a result, using IPSec does not necessarily imply avoiding PKI.

15.2 Secure association definition and overview

Secure association (SA) is a set of shared security aspects between two network entities to support a secure communication. A SA may include aspects such as cryptographic algorithms, traffic cryptographic key, and parameters for the network data to be passed over the connection. A SA is a simplex (one-way logical channel), which provides a secure data connection between two network entities.

Secure associations are established according to the Internet key exchange (IKE) protocol as specified by [b-IETF RFC 5996]. IPsec, as defined by [b-IETF RFC 4301], uses IKE to provide SAs.

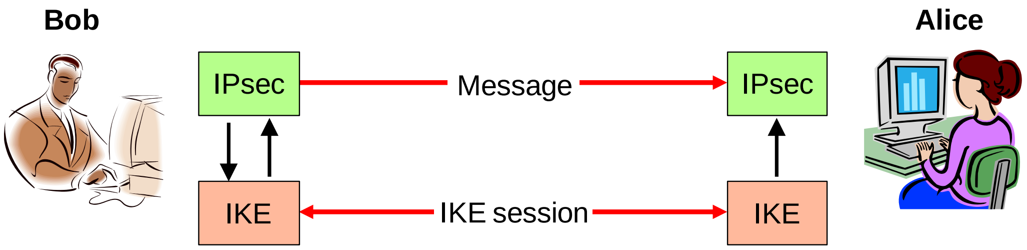


Figure 28 – IPsec and IKE relationship

Figure 28 illustrates the interaction between IPsec and IKE. Bob intends to send a message to Alice, but there is not an appropriate secure association available. Bob's IPsec component asks its IKE component to establish such a secure association with Alice's IKE. When this secure association is established, Bob sends the message.

15.3 Internet key exchange (IKE)

The Internet key exchange (IKE) protocol, as defined by [b-IETF RFC 5996], is an important management protocol, which is used with IPsec. It is a peer-to-peer key protocol. It consists of two phases:

a) In phase one, IKE creates an authenticated secure channel between the two IKE peers that are called the IKE security association. The Diffie-Hellman key agreement is always performed in this phase.

b) In phase two, IKE negotiates the IPsec security associations and generates the required key material for IPsec. The sender offers one or more transform sets that are used to specify an allowed combination of transforms with their respective settings. The sender also indicates the data flow to which the transform set is to be applied. The sender must offer at least one transform set. The receiver then sends back a single transform set, which indicates the mutually agreed-on transforms and algorithms for this particular IPsec session. A new Diffie-Hellman agreement can be done in phase two, or the keys can be derived from phase one shared secret.

15.3.1 IKE authentication techniques

IKE defines three different modes of authentication:

– With pre-shared keys, the same pre-shared key is established on each of the two IPsec entities, for example, using a key distribution centre as discussed in clause 12.5. IKE peers authenticate each other by computing and sending a keyed hash of data that includes the pre-shared key. If the receiving peer is able to create the same hash using its pre-shared key, then the other peer is authenticated. This mode does not require PKI, but it is impractical in a network with many entities.

– The RSA-encrypted nonces method assumes that asymmetric key pairs are assigned to the entities and that an entity has the public key of each entity with which it is communicating. Each party generates a pseudo-random number (a *nonce*) and encrypts it with the other party's public key. Authentication occurs when each party decrypts the other party's nonce with the local private key (and other publicly and privately available information) and then uses the decrypted nonce to compute a keyed hash. This method requires the establishment of a PKI.

– RSA is a public-key cryptosystem used by IPsec for authentication in IKE phase 1. Each entity is assigned a public-key certificate. The RSA signatures method uses a digital signature set-up in which each device digitally signs a set of data and sends it to the other party. This method requires the establishment of a PKI.

The IKEv2 protocol involves the exchange of messages in pairs.

15.3.2 First IKE phase

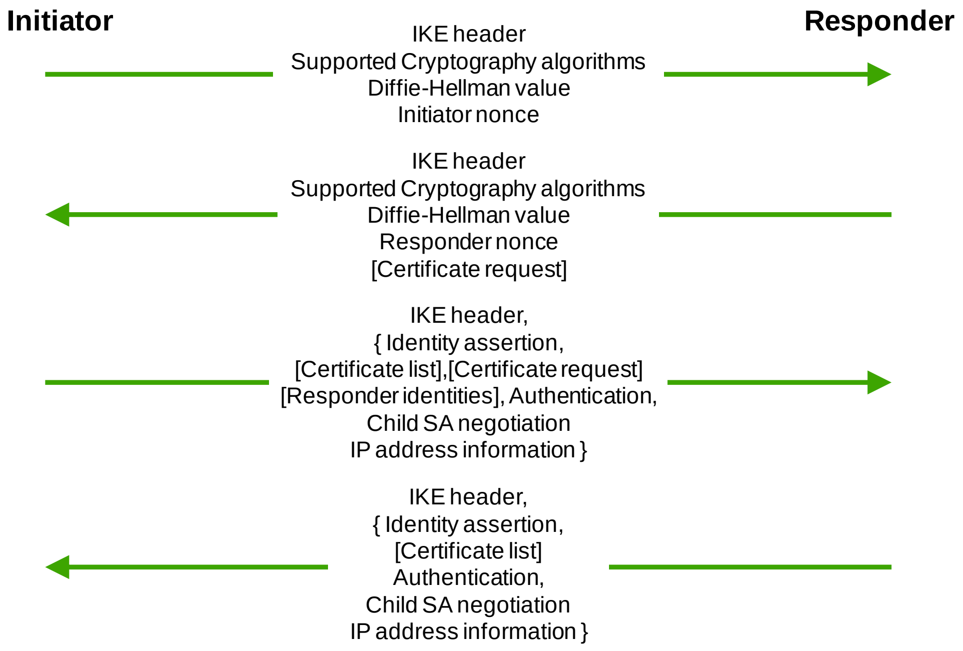


Figure 29 – First IKE phase

The first IKE phase is made of the first two pairs of exchanges, as shown in Figure 29.

The header information for each exchange holds SA identification for both the initiator and the responder, information about the purpose of the exchange, message sequence number, etc.

a) The first pair of exchanges establishes security parameter in the form of:

- cryptographic algorithms and other security parameters;
- Diffie-Hellman parameters, where IKE defines specific groups of Diffie-Hellman parameters (that is, prime and generator) that may be used for the phase 1 IKE exchange (two groups are defined by [b-IETF RFC 5996], while additional groups are defined in [b-IETF RFC 3526]); and
- nonce, where these nonces are used as input in cryptographic functions.

The result of this exchange is to set up a special SA called the IKE SA. This SA defines the parameters for a secure channel between the peers over which subsequent message exchanges take place. Thus, all subsequent IKE message exchanges are protected by encryption and message authentication.

b) The second pair of exchanges provides the following:

- the parties authenticate each other by each signing a block of data;
- optionally, a public-key certificate (or certificate chain) of the sender is submitted;
- set-up of the first IPsec SA.

15.3.3 Second IKE phase

In the second IKE phase, an IPsec child SA is established.

15.4 Internet protocol security (IPsec)

Internet protocol security (IPsec) is a protocol suite for securing Internet protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec is defined in [b-IETF RFC 4301].

IPsec is an end-to-end security scheme on the Internet protocol (IP) level. TLS provides authentication, integrity and confidentiality at the transport layer between two applications, while IPsec protects all communication between two end-systems on the IP network layer.

IPsec is part of IP version 6 and a supplement to IP version 4.

IPsec uses shared, symmetric cryptographic keys for authentication, integrity and confidentiality of data. The keys are generated according to the IKE specification, which in turn uses the Diffie-Hellman procedure for establishing the keys.

16 Perfect forward secrecy (PFS)

Perfect forward secrecy (PFS) refers to the notion that a single key will permit access to only data protected by a single key. For PFS to exist, the key used to protect transmission of data must not be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material must not be used to derive any more keys.

It is important for key management procedures to have the perfect forward secrecy.

17 Current threats to PKI and to ICT networks in general

17.1 Introduction

Clause 13.9 discusses TLS specific threats.

There is genuine belief that the security of a system is not about whether it is secure or not. It is rather a question about how secure it actually is. No system is absolutely secure. Most often, systems have flaws, either by design or by erroneous implementation. There may be also unanticipated threats. Therefore, absolute security cannot be guaranteed for any system. A system can only be considered secure to some extent. A reasonable policy is to assert that the cost of breaking into the system is higher than the value of getting into the system.

Whether motivated by international competition, corporate espionage, nation-state sponsored espionage, political ideology, organized crime, grudge against an employer or even idealism, malicious hacking continues to expand.

The purpose of this clause is to identify as many threats as possible to a PKI and related systems to ensure as far as possible that counter measures are being taken into account when deploying PKI within different areas. The list of threats should be extended as new threats are identified.

This clause will list threats to PKI as they have been identified through the years. It will also briefly describe the measures that may be applied to counteract these threats.

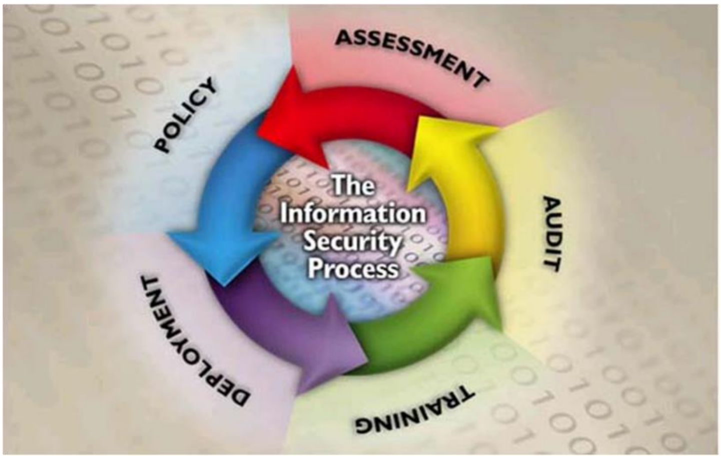


Figure 30 – General information security process – Continuous cycles

Traditionally, security is seen as an iterative process as illustrated in Figure 30. An analysis of possible threats is performed and security measures are taken. As new threats are experienced or identified, the process repeats itself. This is illustrated by the "wheel", as shown in Figure 30. However, in a PKI environment, it may not always be that simple. A PKI design may leave some cracks that cannot be mended by a patch but that require a redesign. The Danish NemID is such an example. NemID is a PKI primarily used for e-banking and e-government. NemID stores the private keys centrally. When a user wants to access an e-banking account, the bank server downloads a Java applet that communicates with the system holding the private key. The design is open for a man-in-the-middle attack. A complete redesign seems necessary to solve this problem.

17.2 Categories of attackers

There are categories of attackers who may have different motives for wanting to perform a hostile attack on a network:

- Countries: State-run, well-organized and well-financed attackers, for example, to gather classified or critical information from countries viewed as hostile or as having an economic, military or a political advantage.
- Hackers: A group of individuals who attack networks and systems seeking to exploit the vulnerabilities or other flaws in operating systems.
- Terrorists: Individuals or groups who operate domestically or internationally.
- Organized crime: Coordinated criminal activities including gambling, racketeering, narcotics trafficking, and many others, performed by well-organized and financed criminal organizations.
- Other types of crime: This can be a not very well organized or financed criminal community, which can consist of one or a few individuals.
- Competitors: Foreign and domestic enterprises engaged in the illegal gathering of information from their competitors.
- Discontented employees: Angry, dissatisfied individuals with the potential to impair the security.
- Careless or unskilled employees: Those people who, through lack of training, concern or attentiveness, pose a threat to the security.

17.3 Brute-force attack

The most basic form of attack on encrypted messages is the brute-force attack, in which every key that could possibly exist is used to decrypt the message until a readable message appears (also called exhaustive key search). The number of attempts trying to decrypt is exponentially proportional to the length of the key; the longer the key is, the more attempts are needed to break the encryption. For a reasonable key length, this type of attack is extremely time- and resource-consuming.

A long key is infeasible to crack using brute-force. However, if the encryption algorithm is known, it might be possible to devise techniques that require fewer trials than needed for brute-force.

As it is rather easy to protect against brute-force attacks, intruders will probably instead attempt to hack into the key management system.

17.4 Spoofing attack

A spoofing attack is a situation in which one entity (for example, person or program) successfully masquerades as being another entity by providing a false identity.

The IP spoofing attack, where a false requestor IP address is supplied, is a common attack. Firewalls using deep packet inspection may mitigate such attacks.

17.5 Common factor attack

This is a rather complex issue. For an RSA asymmetric key, the public key is generated by multiplying two large prime integers. It is extremely difficult out from this product alone to find the two prime integers that created the product. However, it can be shown that a key generator too often uses the same prime integer when creating keys, this prime integer is more easily detected. If the prime integers that go into a public key are found, it is straightforward also to deduce the private key.

The remedy is to ensure that the key generator in its selection of large prime integers uses a good prime random number generator for both factors.

17.6 Eavesdropping

Eavesdropping is a very general concept. It is the act of secretly listening to the private conversations of others without their consent. In ICT and in the context of this Technical Report, it is some kind of sniffing or spying on an ICT connection. There can be different reasons for eavesdropping, such as surveillance by intelligence services, spying for trade secrets, etc. Finally, it can be a preparation for a harmful attack.

Eavesdropping is especially easy to perform on non-encrypted or poorly encrypted connections (see, for example, clause 17.18).

Encryption is the leading technique for protection against eavesdropping.

17.7 Replay attack

A replay attack is a form of an ICT attack in which a valid data transmission is intercepted (eavesdropped) and then maliciously or fraudulently repeated. As an example, a password or even an encrypted password may be intercepted and later be replayed allowing the attacker to masquerade its identity. It can also be a replay of (encrypted) commands for the purpose of sabotage.

Inclusion of some random information, also called *nonce*, together with the valid information and then changing this nonce for each communication will allow detection of a replay attack. Use of time stamps is another or an additional way to detect replay.

The OCSP protocol exhibits a replay problem. A 'good' OCSP response may be intercepted and later replayed after a public-key has been revoked. The nonce OCSP extension as specified in section 4.4.1 of [b-IETF RFC 6960] allows reply attacks to be detected.

17.8 Insecure hashing algorithms

As discussed in clause 2.3, there are some requirements on a hashing algorithm. If these requirements are not met, there is a security risk.

There are different types of attacks related to insecure hash algorithms:

– *Preimage attack*: Given a hash value without knowing the corresponding message, it is possible to find a bit string having that hash value.

– *Second preimage attack*: Given a message, it is possible to find another message that has the same hash value.

– *Collision attack*: It is possible to find two different messages that result in the same hash value.

In a PKI environment, such attacks might be against a public-key certificate where an attacker may want to construct a faked public-key certificate from a genuine public-key certificate allowing identifiable information to remain the same (sequence number and issuer name). This fake public-key certificate appears to be signed by a genuine CA. If the fake public-key certificate is a CA-certificate, the attacker may establish a rogue CA (see clause 17.10) and can issue public-key certificates that appear to be genuine.

A collision attack is easier to perform than the other two types of attacks. The hashing algorithm MD5 has been shown to be vulnerable to collision attacks, but not to second preimage attacks. Some cryptographers anticipate the hashing algorithm SHA-1 to become vulnerable to collision attacks in a not too distant future.

If a CA signs public-key certificates as presented by the user and just adds the sequence number and issuer name, such public-key certificates have been shown to be subject to collision attacks if MD5 is used by the CA for signing. If, on the other hand, the CA creates the entire certificate, the attacker can only launch a second preimage attack.

If a CA does not use sequential sequence numbers in issued public-key certificates, a collision attack is more difficult to launch.

As explained in clause 4.3, [b-ITU-T X.509] specifies that an unrecognized extension labelled as *non-critical* shall be ignored. An attacker might define a fake non-critical extension with a binary content and then modify the bit content to obtain a collision. It appears that there would be less room for attacks if inclusion of public-key certificate extensions were tightly controlled.

17.9 Private key compromise

A private key is compromised if it is suspected that it has been revealed in some way to an outsider. In this case, the practice is to revoke the corresponding public-key certificate.

It is also a good security practice to securely store the private keys, but never to store the private key anywhere in plaintext form. The simplest storage mechanism is to encrypt the private key under a password and store the result on a disk. However, since some weak passwords are sometimes easily guessed, it is important to choose carefully a strong password.

In some environments where human beings are involved, the private key may be stored on portable hardware, such as a smart card. To protect the private key, it is typically password protected. However, weak passwords are often easily guessed, so care should be taken to accept only strong passwords.

In a machine-to-machine environment with no human intervention, it is not possible to store the private key on a removable device; the private key will have to be stored on some kind of tamper-resistant storage. The same may be the case for CAs in general.

17.10 Rogue CAs

A rogue CA is a CA that is trusted by the browser and by similar systems, but which has been established by an attacker. A public-key certificate issued by a rogue CA allows a system to impersonate any website on the Internet, including banking systems using the HTTPS protocol.

17.11 Hijacking of or hacking into a CA

Hijacking of or hacking into an otherwise genuine CA gives the same possibilities for issuing fraudulent public-key certificates as for a rogue CA.

It is crucial to take all possible measures to protect a CA from intruders. A good practice is to apply the traditional security precautions for protecting a computer site, for example, as described in [b-ISO/IEC 27001] and [b-ISO/IEC 27002].

Fraudulently issued public-key certificates should be revoked as soon the breach has been detected.

17.12 Bogus public-key certificates

A bogus or fraudulent public-key certificate is a public-key certificate that allows an entity to impersonate some other legal entity, for example, a website. A bogus public-key certificate may be issued by a rogue CA, by a hijacked CA, or be a modified public-key certificate. It could also be a public-key certificate that has been issued by a careless, but otherwise genuine CA. If the identity of a public-key certificate requestor is not thoroughly checked, the public-key certificate may be issued with a false name.

17.13 Denial-of-service attacks

A denial-of-service (DoS) attack or a distributed denial-of-service (DDoS) attack is an attempt to make a resource unavailable to its intended communication partners. There are many ways of performing a DoS attack. The more traditional types of DoS attacks are:

- consumption of computational resources, such as bandwidth, disk space, or processor time;
- disruption of configuration information, such as routing information;
- disruption of state information, such as unsolicited resetting of TCP sessions;
- disruption of physical network components;
- obstruction of the communication media between the intended users and the victim so that they can no longer communicate adequately.

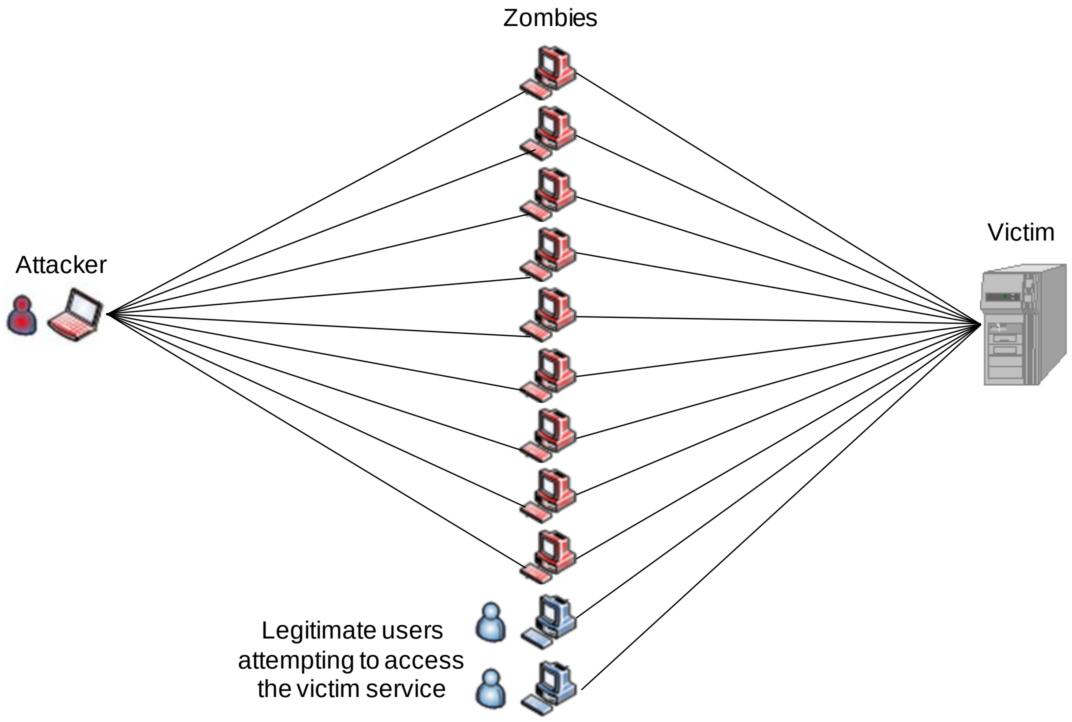


Figure 31 – DDoS attack

A DDoS attack is a type of DOS attack where multiple compromised systems, which are usually infected with a Trojan horse, are used to target a single victim. Victims of a DDoS attack consist of both the targeted system and all the systems used maliciously and controlled by the hacker in the distributed attack.

17.14 Redirection attacks

In a redirection attack, the users believe they are accessing a particular secure website, but in fact, they are being redirected to another website, possibly for phishing. In the context of PKI, a user may be redirected to a rogue website with a public-key certificate that appears to be valid.

17.15 Man-in-the-middle attack

A man-in-the middle (MITM) attack is one of the most common and dangerous means of attacks. It is a common term for different specialized attacks, and it is the basis for many different attack types. It is difficult to detect the consequences of the attack without prior knowledge of the type of the MITM attack and the measures taken to protect against such an attack. Intrusion detection does not detect man-in-the-middle attacks.

MITM is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. The attacker is able to intercept all messages between the two victims and inject new ones, which is done in a straightforward manner in many circumstances (for example, an attacker within the reception range of an unencrypted wireless fidelity (Wi-Fi) access point can insert himself as MITM).

Different types of MITM attacks are described in:

– Figure 20 and the associated text describe a MITM attack on a Diffie-Hellman agreement.

– Clause 13.9 describes a downgrade attack on a TLS communication.

In wireless networking, the MITM scheme is implemented in a number of ways. One way is to operate a rogue access point resembling a legitimate wireless hotspot. Often the real access point is jammed or blocked while the rogue, with the same service set identification (SSID), is in the clear with a strong signal. Another method is to break a client's connection and lure the client's hardware into reconnecting to the man-in-the middle. In this case, the man-in-the-middle has faked the access point MAC address.

17.16 Spear phishing e-mail

Spear phishing may be defined as “highly targeted phishing aimed at specific individuals or groups within an organization". Spear phishing makes use of information about a target to make attacks more specific and “personal” to the target. Spear-phishing e-mails, for instance, may refer to their targets by their specific name, rank, position or interests instead of using generic titles as in broader phishing campaigns.

There are examples of how phishing has been used to get malware onto a system with a somewhat careless user and then move on to another system to obtain critical information or inflict harm.

Protection is mostly a matter of good security practice, and it is not necessarily a PKI standardization issue.

17.17 HTTPS stripping attack

An HTTPS stripping attack is an attack where the attacker transparently transfers a secure HTTPS session to a plain HTTP session. It is also called an SSL stripping attack.

An HTTPS stripping attack assumes that an attacker can be between the victim and the intended server.

17.18 Security impact on new top level domains (TLDs)

It has been common for enterprises to configure DNS domains with suffixes that are not in the set of public top level domains (TLDs). The public delegation of these suffixes as new official TLDs will impose serious security risks.

Domains that are affected are: belkin, corp, domain, home, internal, intranet, invalid, lan, local, localdomain, localhost, private and wpad.

Any domain approved as a new global top level domain (gTLD) will have to be addressed by CAs. CAs will have to review the public-key certificates they have issued and advise customers that they have public-key certificates with a new official TLD. The customers will then have to register their domain. If the customer cannot or does not register the domain, then the CA could revoke the affected public-key certificates.

17.19 Weak encryption technique

Wired equivalent privacy (WEP): WEP is a security protocol based on 64- or 128-bit encryption for the wireless network. It was one of the first wireless encryption protocols. It is extremely unsecure, easily cracked. WEP, when it first came out, was the start of a great idea but since then it has been replaced by Wi-Fi protected access (WPA) that was again replaced by WPA2-P2K (WPA2-Personal) and WPA2-Enterprise that are more secure than WEP. Virtual private networks (VPNs) and other various protocols are alternatives to WEP.

17.20 Wireless router administration

Many wireless routers have a setting that allows administration of the router via a wireless connection. This means that it is possible to access all of the routers security settings and other features without having to be on a computer that is plugged into the router using an Ethernet cable. While this is convenient for administering the router remotely, it also provides for another point of entry for the hacker to get to the security settings and make them more hacker friendly. Many people never change the default factory administration password of their wireless router, which makes things even easier for the hacker. A good practice is to turn off the "allow admin via wireless" feature; in this manner, only a person with a physical connection to the network can attempt to administer the wireless router settings.

17.21 Thumb drive universal serial bus (USB) memory stick

Attacks on a network by using a USB memory stick may be as simple way to launch an attack. Malware can be inserted in a network computer by using a USB stick. It may also be an easy way to copy data that should not have been revealed.

18 Current PKI and key management activities

18.1 Introduction

Several standardization organizations are involved in PKI and key management specifications.

PKI, as defined by [b-ITU-T X.509], is typically deployed in an environment where browsers are used for accessing different services, such as e-banking, e-government, online shopping, etc. PKI may also be used for electronic signatures on e-mails, documents and computer codes. In these environments, the establishment, maintenance and operation involve human intervention, such as the use of passwords, pin-codes, code boxes, smart cards, one-time passwords over mobile phones, etc. The so-called two-factor authentication may be employed, and it may be based on what a user knows (e.g., password) together with what the user possesses (e.g., smart card).

Legal aspects often influence traditional PKI deployment by specifying how public-key certificates are constructed and what information has to be retained for possible court actions.

Industry groups or similar organizations have so far done the profiling of PKI for particular purposes, but standardization groups have also made specifications for the use of PKI for particular applications. The following clauses describe briefly the different PKI and/or key management activities.

18.2 Certification Authority Browser Forum

The Certification Authority Browser Forum (CA/Browser Forum) is a voluntary organization of certification authorities (CAs) and vendors of Internet browsers software and the like. The CA/Browser Forum defines rules (certificate policies) for TLS public-key certificates for web servers.

The CA/Browser Forum maintains two specifications for issuing public-key certificates.

18.2.1 Publicly trusted certificates

A publicly trusted certificate [b-CABF-PTC] is an end-entity public-key certificate or a CA-certificate, including a CA-certificate for a trust anchor. The intent is that all browsers and other relying party application software will incorporate these basic requirements where a self-signed CA-certificate is provided as trust anchor information.

A subscriber is assumed a person or a legal entity managing the end-entity to which the end-entity public-key certificate has been issued.

18.2.2 Guidelines for issuance and management of extended validation certificates

An extended validation (EV) certificate [b-CABF-EVC] is a public-key certificate issued according to a specific set of identity verification criteria. These criteria require extensive verification of the requesting entity's identity by the CA before a public-key certificate is issued. EV certificates are intended to be used in a web-based communication using the TLS protocol.

The primary objective for the specification is to allow secure identification of a web server that a user is accessing, and that this web server is controlled by a legal entity as identified by the EV certificate.

18.3 ETSI electronic signatures and infrastructure

The European Commission initiated some years ago an activity called the European Electronic Signature Standardization Initiative (EESSI) on the use of digital signatures mostly in the public area, but also within other areas with similar requirements, such as e-banking and e-commerce. The *Comité Européen de Normalisation* (CEN) and the European Telecommunications Standards Institute (ETSI) were mandated to carry out this work.

EESSI classifies public-key certificates and digital signatures according to the area of their deployment. Public-key certificates may be issued after some checking of the content. A PKI may be under more or less stringent management. Digital signatures may be used in different environments with different security requirements. Digital signatures can therefore be classified according to their use. In particular, qualified public-key certificates and qualified digital signatures are defined to be deployed in sensitive areas, such as significant economical transactions, transfer of sensitive personal information, etc.

The ETSI Electronic Signatures and Infrastructure (ESI) technical committee issued a large number of specifications to cover different aspects.

The activity covers signature creation and verification based on advanced digital signatures founded on CMS, as discussed in clause 9. It also covers extensible markup language (XML) advanced electronic signatures (XAdES), portable document format (PDF) advanced electronic signatures (PAdES), and associated signature container (ASiC).

18.4 IETF, the public-key infrastructure (X.509) (PKIX)

In the Internet Engineering Task Force (IETF), the public key infrastructure (X.509) (PKIX) working group (WG) developed Internet specifications to support ITU-T X.509-based public-key infrastructures. PKIX WG terminated its activity, but it still maintains an e-mail exploder.

While the CA/Browser Forum and ETSI ESI are primarily profiling [b-ITU-T X.509] for special purposes, PKIX was an additional standards making body.

18.5 ITU-T public-key infrastructure activities

18.5.1 ITU-T Study Group 17 activities

ITU-T Study Group 17 has the responsibility for [b-ITU-T X.509]. This is collaborative work with ISO/IEC JTC 1/SC 6, where the same specification is published as ISO/IEC 9594-8.

[b-ITU-T X.509] is the base specification for all PKI work and is the base for the descriptions given in clauses 2 to 5.

[b-ITU-T X.1164] specifies a different approach usage of PKI, where the user has its own CA, but there is no notion of a trust anchor. [b-ITU-T X.1164] assumes the use of HTTPS, which implies the deployment of TLS.

18.5.2 ITU-T Study Group 13 activities

ITU-T Study Group 13 has developed some security specifications for next-generation networks (NGNs). [b-ITU-T Y.2704] includes some PKI aspects.

NGN differentiates between trusted zones, trusted-but-vulnerable zones and untrusted zones. The trusted zone is the part of the network that is within the enterprise premises without any direct connection to the outside world. A trusted-but-vulnerable zone is part of the network within the enterprise premises that have outside connections. An untrusted zone is the part of the network that is outside the enterprise premises and control.

The philosophy is that trusted zones and trusted-but-vulnerable zones have different security requirements and that the threats to the trusted zones come through the trusted-but-vulnerable zones.

[b-ITU-T Y.2704] specifies the use of public-key certificates to provide security between network elements (M2M) and between the users and network elements.

NGN may optionally use TLS. [b-ITU-T Y.2704] has some specification on how TLS should be used, and it lists the different cipher suites that may be used.

[b-ITU-T Y.2704] also discusses at length key management and the use of IKE and IPsec. It suggests that the principle specified in [b-ITU-T X.1035] might be used to avoid middle-in-the-man attacks (see 12.6).

18.6 ISO/TC 68 (Financial services) PKI activities

ISO/TC 68 (Financial services) has developed [b-ISO 21188], which is a framework for the requirements to manage PKI through certificate policies and certification practice statements, and to enable the use of public-key certificates in the financial services industry.

[b-ISO 21188] provides an introduction on PKI and its uses in the banking environment. In addition, it introduces a certificate validation service provider, which may assist a relying party in public-key certificate validation, a function that is under consideration for standardization in [b-ITU-T X.509].

The major part of [b-ISO 21188] is a description of the procedures to be observed by the different partners managing a PKI.

18.7 ISO/TC 215 (Health informatics) PKI activities

ISO/TC 215 (Health informatics) has developed a three-part standard [b-ISO 17090].

[b-ISO 17090-1] defines the basic concepts underlying the use of public-key certificates in healthcare, and provides a scheme of interoperability requirements to establish a public-key certificate-enabled secure communication of health information. It also identifies the major stakeholders who are communicating health-related information, as well as the main security services required for health communication where public-key certificates may be required.

[b-ISO 17090-1] also introduces public key cryptography and the basic components needed to deploy public-key certificates in healthcare. It further introduces different types of digital certificates — identity certificates and associated attribute certificates for relying parties, self-signed CA- certificates, and CA hierarchies and bridging structures.

[b-ISO 17090-2] specifies the certificate profiles required to interchange healthcare information within a single organization, between different organizations and across jurisdictional boundaries. It details the use made of public-key certificates in the health industry and focuses, in particular, on specific healthcare issues relating to certificate profiles.

[b-ISO 17090-3] gives guidelines for certificate management issues involved in deploying public-key certificates in healthcare. It specifies a structure and minimum requirements for certificate policies, as well as a structure for associated certification practice statements.

[b-ISO 17090-3] also identifies the principles needed in a healthcare security policy for cross-border communication and defines the minimum levels of security required, concentrating on aspects unique to healthcare.

18.8 IEC/TC 57/WG 15 (smart grid) PKI activities

IEC Technical Committee 57, Working Group 15 defines key management for the smart grid environment. More on this committee's activity is provided in clause 22.5.

18.9 Open Mobile Alliance (OMA) - Wireless application protocol (WAP)

The Wireless Application Protocol (WAP) Forum has issued a large number of specifications with some of them are PKI related. The WAP Forum is now an affiliate of the Open Mobile Alliance (OMA).

OMA has issued several documents on OMA digital rights management (DRM), where at least one, the DRM specification, has PKI related specifications.

It specifies XML signature for XML-encoded messages and specifies use of OASIS XML signature.

The WAP specifications may be found in: http://technical.openmobilealliance.org/Technical/wapindex.aspx. Of interests are:

– WAP-211-WAPCert: *WAP Certificate and CRL Profiles Specification*

  This Technical Report makes specifications for public-key certificates to be used in WAP.

– WAP-217-WPKI: *Public Key Infrastructure Definition.*

– WAP-219-TLS: *TLS Profile and Tunneling Specification*.

18.10 ISO/IEC JTC 1/SC 27 activities

ISO/IEC JTC 1/SC 27 has developed a five-part standard [b-ISO/IEC 11770] on key management.

[b-ISO/IEC 11770-1] Framework: Defines a general model of key management that is independent of the use of any particular cryptographic algorithm. However, certain key distribution mechanisms can depend on particular algorithm properties, for example, properties of asymmetric algorithms.

[b-ISO/IEC 11770-2] Mechanisms using symmetric techniques: It specifies a series of 13 mechanisms for establishing shared secret keys using symmetric cryptography. These mechanisms address three different environments for the establishment of shared secret keys: point-to-point key establishment schemes, mechanisms using a key distribution centre (KDC), and techniques that use a key translation centre (KTC). It describes the content of messages that carry keying material.

[b-ISO/IEC 11770-3] Mechanisms using asymmetric techniques: It defines key management mechanisms based on asymmetric cryptographic techniques. It specifically addresses the use of asymmetric techniques to achieve the following goals:

a) Establish a shared secret key for a symmetric cryptographic technique between two entities *A* and *B* by key agreement. In a secret key agreement mechanism, the secret key is the result of a data exchange between the two entities *A* and *B*. Neither of them can predetermine the value of the shared secret key.

b) Establish a shared secret key for a symmetric cryptographic technique between two entities *A* and *B* by key transport. In a secret key transport mechanism, the secret key is chosen by one entity *A* and is transferred to another entity *B*, suitably protected by asymmetric techniques.

c) Make an entity's public key available to other entities by key transport. In a public key transport mechanism, the public key of entity *A* must be transferred to other entities in an authenticated way, but not requiring secrecy.

[b-ISO/IEC 11770-4] Mechanisms based on weak secrets: It defines key establishment mechanisms based on weak secrets, i.e., secrets that can be readily memorized by a human, and hence secrets that will be chosen from a relatively small set of possibilities. It specifies cryptographic techniques specifically designed to establish one or more secret keys based on a weak secret derived from a memorized password, while preventing offline brute-force attacks associated with the weak secret.

[b-ISO/IEC 11770-5] Group key management: It specifies key establishment mechanisms for multiple entities to provide procedures for handling cryptographic keying material used in symmetric or asymmetric cryptographic algorithms according to the security policy in force. It defines the symmetric key establishment mechanisms for multiple entities with a key distribution centre (KDC), and defines symmetric key establishment mechanisms based on general tree-based structure with both individual rekeying and batched rekeying. It also defines key establishment mechanisms based on key chain with both unlimited forward key chain and limited forward key chain. Both key establishment mechanisms can be combined by applications.

18.11 National Institute of Standards and Technology (NIST) activities

National Institute of Standards and Technology (NIST) has developed a three-part specification on "Recommendation for Key Management".

– [b-NIST 800-57-1] contains basic key management guidance by giving "best practices" associated with key management by:

  a) Defining the security services that may be provided and key types that may be employed in using cryptographic mechanisms.

  b) Providing background information regarding the cryptographic algorithms that use cryptographic keying material.

  c) Classifying the different types of keys and other cryptographic information according to their functions, specifying the protection that each type of information requires and identifying methods for providing this protection.

  d) Identifying the states in which a cryptographic key may exist during its lifetime.

  e) Identifying the multitude of functions involved in key management.

  f) Discussing a variety of key management issues related to the keying material.

– [b-NIST 800-57-2] identifies the structural and functional elements common to effective key management systems, and it identifies security planning requirements, general security policies and practices necessary to effective institutional key management, and finally it offers suggestions regarding how key management policies and procedures might be incorporated into security planning documentation.

– [b-NIST 800-57-3] gives a good discussion of some of the issues also discussed in this Technical Report.

NIST has also published some considerations on key management in a smart grid environment (see clause 22.4).

18.12 OASIS enterprise key management infrastructure (EKMI) TC

OASIS has a Technical Committee on enterprise key management infrastructure (EKMI).

The key management interoperability protocol [b-OASIS KMIP] is designed to be a comprehensive protocol for communication between enterprise key management systems and encryption systems. By using a consolidated protocol, organizations will be able to simplify key management and reduce operational costs significantly.

The key management interoperability protocol profiles [b-OASIS KMIP prof] provides guidance to developers and architects who wish to design systems and applications that conform to the key management interoperability protocol specification.

19 Wireless PKI (WPKI)

19.1 Mobile environment

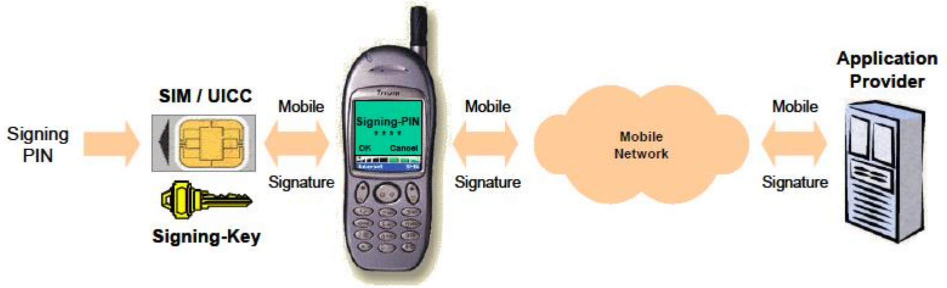


Figure 32 – Typical mobile implementation

In developing countries, e.g., in Africa, the mobile phone has become the primary device for data communication. Use of mobile devices requires new PKI approaches, e.g., new ways of providing trust. The special requirements for the establishment of a wireless PKI needs to be created.

– A wireless network has less bandwidth, more latency, and insecure connection and device problems such as a less powerful central processing unit (CPU), less memory size, restricted battery power, small display and input device.

– Mobile phones lack computing capabilities of PKI services such as key generation, digital signature generation and verification, certificate validation, and certificate revocation list (CRL) verification. They also lack memory size of storing public-key certificates and CRLs.

Newer phones (smart phones) may have more advanced capabilities, but they still have limited capabilities. Services should also be provided for those who cannot afford devices that are more expensive.

19.2 Current mobile PKI activities

19.2.1 ITU-T next-generation network activities

[b-ITU-T Y.2740] describes security issues related to the mobile payment system (MPS) as defined in [b-ITU-T Y.2741]:

- security goals for the mobile commerce and the mobile banking systems, based on four specified security levels;
- probable risks in mobile commerce and mobile banking systems, and specifies means for risk reduction;
- many PKI and TLS related terms, such as access control, authentication, non-repudiation, data confidentiality, data integrity, privacy, symmetric and asymmetric cryptography, etc., are used.

[b-ITU-T Y.2741] describes a scenario where a client, the mobile operator, an MPS operator ensuring security, a final institution accepting payment on behalf of a merchant.

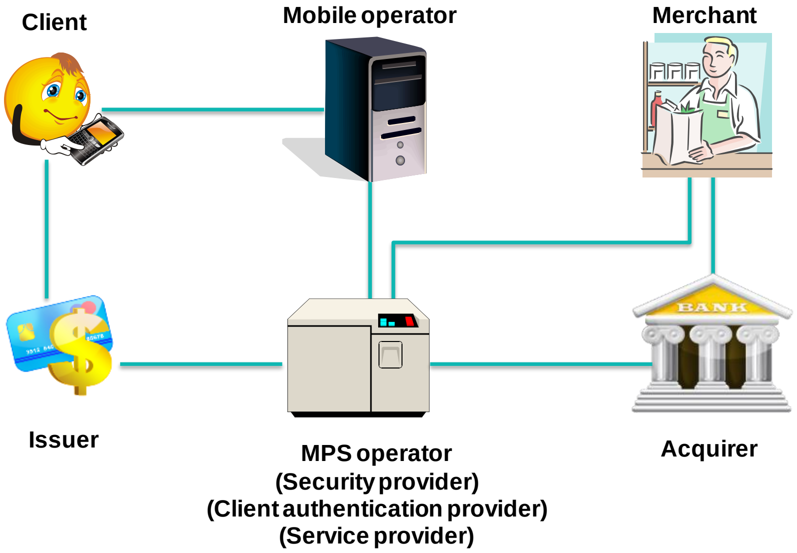


Figure 33 – Partners in mobile payment system as per [b-ITU-T Y.2741]

[b-ITU-T Y.2741] describes different scenarios:

a) Enrolment of client with the mobile payment system (MPS) operator, i.e., an interaction between the client and the MPS operator with the mobile operator as an intermediary.

b) Enrolment with the MPS operator of the type payment (bank card, PayPal, etc.)

c) A financial transaction initiated by the client is performed by going to the MPS operator with the mobile operator as an intermediary. The MPS operator authenticates the client, and the transaction is then forwarded to the acquirer and from there to the merchant. The result is returned along the same path.

d) A financial transaction initiated by the merchant going to the MPS operator with the mobile operator as an intermediary. The MPS operator then forwards the transaction to the client with the mobile operator as an intermediary. The transaction then follows the same procedure as in c).

From a security point of view, the concept seems a little shaky. A broken connection during the transaction could leave the partners in limbo. The encryption is dependent on the encryption capabilities established by the mobile operator. There is no end-to-end encryption, which requires some key management activities not included in the specification.

19.2.2 ETSI activities

The European Telecommunications Standards Institute (ETSI) has produced three specifications for digital signing using the mobile telephone: [b-ETSI TR 102 203], [b-ETSI TS 102 204] and [b-ETSI TS 102 207].

These specifications do not assume that the mobile phone is used for transaction processing, but is only used for signing transactions entered on another media. Finland has introduced a mobile service based on these specifications.

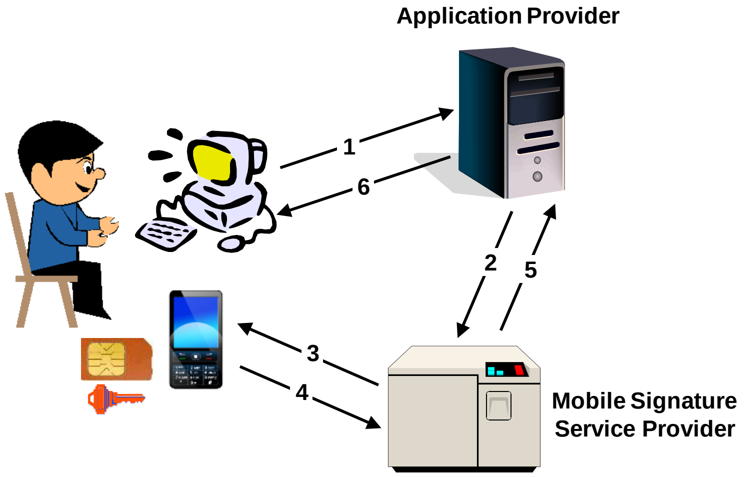


Figure 34 – Signing using mobile phone

In this environment, the private key used for signing is placed on the tamper-resistant subscriber identity module (SIM) card. The signing application may also be placed on the same SIM card. This is illustrated in Figure 34 where a transaction is entered through a workstation, for example, a personal computer (PC). The signing is then done through a number of steps as illustrated in the figure.

19.3 Relevance of WPKI

Future development within the mobile environment will assume that the mobile phone is also used for the actual transaction handling replacing PC.

A more generalized wireless PKI solution should enable secure mobile access to e-banking, e-government, e-commerce, e-health, etc.

19.4 Use of SIM and universal integrated circuit card (UICC) capabilities

The SIM card of a mobile phone securely stores the international mobile subscriber identity (IMSI). A SIM card has an integrated circuit card ID (ICCID). This is the identifier of the actual SIM card itself, i.e., an identifier for the SIM chip. It is possible to change the information contained on a SIM (including IMSI), but the identity of the SIM itself remains the same.

Symmetric-key cryptography (see clause 2.2.1) is used for authenticating the mobile device and for encryption of the communication. The same 128-bit key is stored on the SIM card and in the database of the mobile operator. The key is stored in such a way on the SIM card that it is supposed not to be revealed. This protection is apparently not always reliable.

UICC is a new generation SIM included in cell phones or laptops used in some high speed wireless 3G networks. UICC works with any 3G or 4G device. For example, subscribers are able to transfer easily their phonebook and preferences from one handset to another.

Technically, UICC works in all mobile telecom networks. It is a type of smart card technology. Smaller in size than a full card, it contains a computer, or microprocessor, and it has its own data storage and software. It is an evolution of the SIM used to identify subscribers in the global system for mobile communications (GSM).

UICC has an advantage over SIM; UICC can have multiple applications installed in it.

Another advancement is that UICC can communicate using IP, the same standard used in the Internet and the new generation of wireless networks.

Making WPKI dependent on the use of SIM or UICC cards implies that the mobile operators become part of a WPKI environment, as they are the issuers of such cards.

19.5 Key management issues

The native encryption on a mobile communication is not from end to end, but from the handset to the mobile operator. If the native mobile encryption is made as part of the overall security, the mobile operator becomes the partner in the security infrastructure. If independence of the mobile operator security capabilities is wanted, and/or end-to-end encryption is required, then key management independent of the mobile operator is required. As discussed in clause 14, the former WAP Forum has developed a version of TLS suited specifically for a mobile environment.

19.6 Threats specific to WPKI

The main specific threats to WPKI are that mobile devices might be handled more carelessly than PCs, as the general user may not be as careful in protecting the mobile phone in the same way as PC, e.g., by having a virus protection program. This provides a high risk for inclusion of malware when the mobile phone is not operating in a WPKI environment.

The native encryption on a mobile communication can be cracked and subsequently be eavesdropped.

20 PKI in M2M environment

20.1 Special M2M challenges

In a traditional PKI deployment, the operation of PKI requires some human decision taking and intervention. This could be showing up in person for secure identification. PKI relevant information (public-key certificate and private key) may be placed on a smart card for human handling, etc.

Embedded software is software running in entities or devices that are not normally considered PCs. It is software performing local control and/or communication functions with other similar entities for data collection, for controlling these entities or for being controlled. The information on the communication channels is mostly for control use.

PKI deployment in anM2M environment is quite different from traditional PKI deployment. It differs in the following ways:

- The operational procedures are conducted without human involvement requiring special procedures. Techniques like two-factor authentication cannot be utilized.
- Messages are typically required to be encrypted; this adds special requirements on key management.
- As encryption keys are held by the communication systems, rather than on some kind of smart card, tamper resistant storage is required.
- The number of systems may be too large for manual configuration requiring standardized establishment and maintenance procedures.

The confidentiality is typically provided by IPsec or TLS. Both techniques are dependent on underlying key management capabilities, which again is dependent on a functional PKI.

20.2 Encryption and key management issues

In many M2M networks, there is a requirement for authentication and confidentiality. When commands and information are flowing among systems, authentication of the sender (sending entity) is important. Getting false information and/or false commands could jeopardise the whole network. Confidentiality may be achieved by the use of symmetric cryptography and some kind of automatic symmetric key generation, for example, using TLS or IPsec/IKE.

20.3 Threats specific to M2M

20.3.1 Configuration problems

A M2M network may have many entities and corresponding communication channels. This lends to many possibilities for configuration errors.

The generation and distribution of public-key certificates, private keys, trust anchor information, etc., may be error prone.

Although general configuration errors may impose security threats, they are outside the scope of this Technical Report. However, PKI configuration issues are within the scope of this Technical Report.

20.3.2 Software problems

Errors are known to exist in software causing security risks. Software vendors distribute daily updates to patch security errors in their products. It is anticipated that M2M software be subject to software weaknesses. Procedures for keeping the software updated at all times are required. With possible millions of very different entities with different software vendors, this is not a trivial problem.

20.3.3 Denial-of-service (DoS) attacks

Clause 17.13 gives a general overview of DoS attacks. All these types of attacks have to be considered when analysing M2M security. This Technical Report discusses those types of DoS attacks that are specific to M2M:

- If a public-key certificate of an entity is deleted or replaced with an bogus one, this device can no longer communicate with other entities and is thereby put out of service.
- If a public-key certificate for an entity is revoked, this device is out of service.
- If the revocation service is out of order, it is not possible to validate public-key certificates and the whole network is out of service.
- Communication overhead may cause delays in the ICT network, which may result in faulty controls.
- Destruction of the private key will put an entity out of operation.
- If the trust anchor information in an entity were replaced with illegal information, such a device would accept a public-key certificate from a hostile entity and may get into communication with such an entity.

21 PKI and cloud computing

21.1 Introduction

There are two aspects of PKI with relation to cloud computing:

a) providing security to users accessing resources in the cloud;

b) PKI as a cloud software-as-a-service (SaaS).

21.2 PKI structures for cloud computing

The considerations given in this clause have some general applicability and are in particular relevant for cloud computing.

The following different PKI environments have been identified as interesting for cloud computing. A question for enterprises to decide is to what degree they elect to be engaged in PKI.

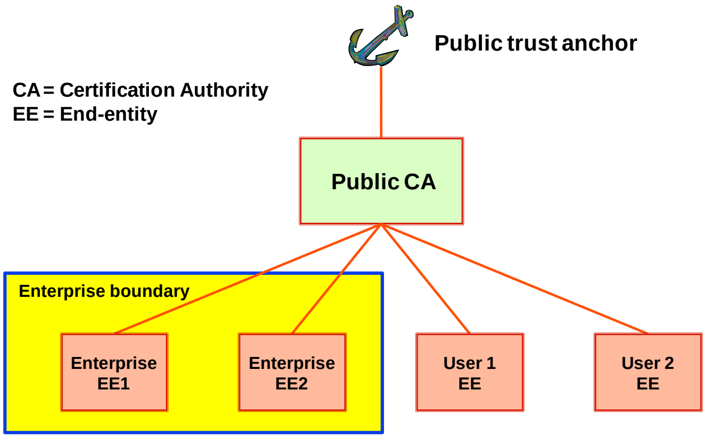


Figure 35 – Public trust anchor and public CA

Figure 35 illustrates a situation where an organization or enterprise only obtains public-key certificates for the end-entities from a public CA, that is, a CA that serves several organizations.

This configuration has the advantage that web browsers (acting as relying party) will accept the public-key certificates, as the relevant trust anchor information is included in the web browser trust anchor store.

This configuration has some disadvantages such as:

- There is a limited enterprise control of security properties, such as algorithms used, key strength and type of extensions, for example, key usage.
- The enterprise cannot issue public-key certificates on demand, but is dependent on procedures established by the public CA.
- There is a cost encountered for issuing the public-key certificates by a public CA.

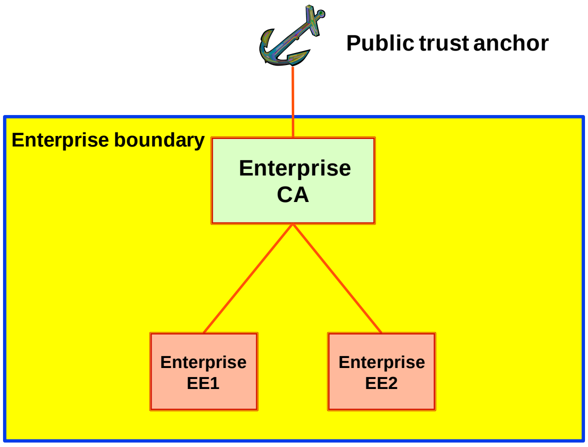


Figure 36 – Public trust anchor and enterprise CA

Figure 36 shows the situation where an enterprise decides to establish one or more CAs under its own management, but does not provide its own trust anchor.

This configuration has the following advantages:

– Web browsers (acting as relying party) will accept the public-key certificates, as the relevant trust anchor information is included in the web browser trust anchor store.

– Public-key certificates may be issued on demand and public-key certificates may be revoked instantly.

On the other hand, the configuration has the following disadvantages:

– Some security aspects are controlled by the trust anchor and therefore are outside the jurisdiction of the enterprise.

– There is still a cost associated with the services of the trust anchor.

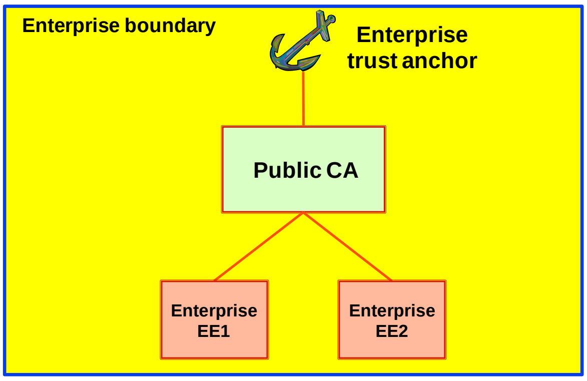


Figure 37 – Enterprise trust anchor and enterprise CA

Figure 37 illustrates the case where the enterprise also provides its own trust anchor.

This configuration has the following advantages:

– Public-key certificates may be issued on demand and public-key certificates may be revoked instantly. It is not necessary to wait for complex and long procedures.

– The enterprise has the full control of security parameters.

– The relying party within the same enterprise needs less validation of the end-end public-key certificates.

The configuration has the following disadvantages:

– Web browsers (acting as relying parties) may not accept the public-key certificates, as the relevant trust anchor information is not included in the web browser trust anchor store.

21.3 PKI-related accessing services from the cloud

A highly referenced presentation on RSA Conference 2011 considers PKI and cloud computing, [b-RCA NMS-301] indicates that TLS is used when accessing cloud services. A NIST document [b-NIST 800-144] also indicates that security assertion markup language (SAML), [b-ITU-T X.1141], may be used when accessing a cloud service. In this case, the messages are wrapped by the *simple object access protocol* (SOAP), which is based on extensible markup language (XML). SOAP messages are digitally signed. Accordingly, PKI is required depending on whether TLS or SAML is used for accessing cloud services.

No new ITU-T X.509 requirements for supporting cloud computing have been identified so far, but this requires some more investigation.

21.4 PKI as a service provided by cloud computing

21.4.1 Overview

An enterprise using the cloud PKI service wants to be able to see only its own PKI components. This results in a need to separate the customers of this service.

Three factors are considered when designing the A PKI solution for cloud computing: scalability, mobility and automation. A PKI solution for cloud computing needs to be able to add more CAs on demand, be relatively consistent in the required time to sign public-key certificates, and always be available. Hence, the PKI solution for cloud computing will support the CA operations being movable to another less strained server if the number of requested signatures increases beyond the limit of the hardware security module (HSM) or if the service fails without prior notice. To be able to move all CA operations to another server, all data regarding that CA need to be moved between databases and the private key has to be moved or be the same at the new location. However, no sufficiently secure procedure exists to move private keys between HSMs autonomously. Therefore, it is necessary that the same private keys be predefined in HSMs at all available locations of that CA. The ability to move the CA to another location and to bind private keys on demand provides scalability in the number of signatures the system can handle. The scalability of the number of CAs at one location is relative to the number of keys the hardware security module is able to store.

PKI as a cloud service addresses two dimensions; one dimension addresses reliability and availability aspects, and the other addresses security and trust. PKI as a system is entirely based on trust: who trusts whom and to what extent. A PKI also needs to be reliable and available when public-key certificates are to be issued and validated. These two areas are the basis for the greatest concern of organizations migrating to cloud services.

21.4.2 Possible solutions

Although at this stage no requirement has been identified yet that affects the base of [b-ITU-T X.509] specification, some clarifications might be desirable or needed. However, there seems to be a need for a specification that includes recommendations on how a PKI might be established as a service within the cloud computing environment.

22 Security issues for smart grid

22.1 Introduction

The world is trying to move away from fossil energy resources toward renewable electric energy by using natural resources such as solar energy, wind energy, tides, waves, biomass, etc. Use of electricity for heating, cars, etc., will put an additional load on the current electric grid. Some measures need to be taken:

– By having a tighter control of the grid, it is possible to increase its utilization.

– By taking steps to distribute the energy consumption more evenly over time, it is possible to get a better utilization of the available electricity supply and the transmission grid.

– By taking steps to reduce the consequences of varied electricity energy production by diverging more heavy use (charging of electric cars, heat pumps, etc.) to periods with ample supplies.

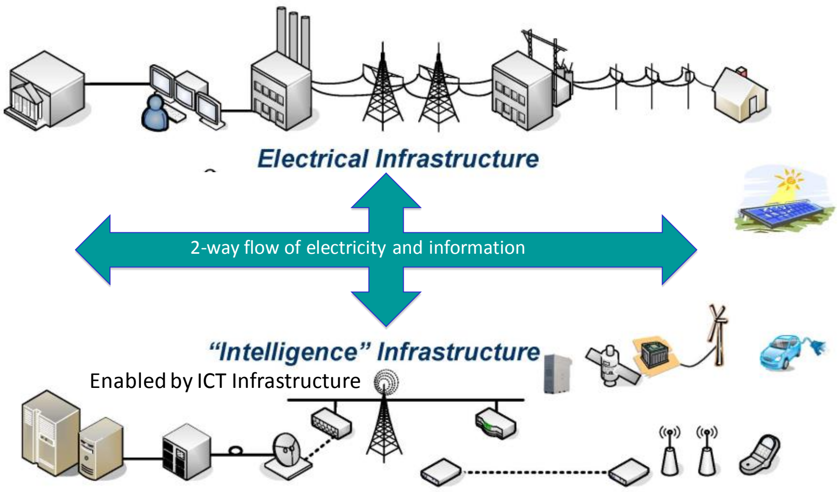

Electrical Infrastructure
2-way flow of electricity and information
"Intelligence" Infrastructure
Enabled by ICT Infrastructure

Figure 38 – Smart grid

All the above requires the establishment of a controlling ICT network parallel to the electric grid resulting in what is called *smart grid* as indicated in Figure 38.

The supporting ICT network interconnects a vast number of entities resulting in a corresponding vast number of communication connections, with a need for mutual authentication of partners and of encrypted communication.

The ICT network will be built on international standards to ensure interoperability. The following clauses do not discuss all such standards related to smart grid, but focus just on the security aspects and related smart grid security standards, where PKI plays a major role.

22.2 Smart grid environment

The smart grid is quite a particular environment where traditional PKI methods may not be applicable and where the concept of PKI needs to be re-evaluated. It may involve extensions and adjustments of [b-ITU-T X.509], as well as the development of some supplementary specifications.

- Communication is typically between entities (M2M communication), which means that procedures requiring human intervention are not applicable.
- Some entities, particularly residential meters and in-home entities, are small and relative low-cost units. Such entities may have limited processing and storage capabilities. Traditional communication protocols such as TLS and traditional PKI handling are difficult or expensive to incorporate. However, low cost embedded processors with cryptographic capabilities could be part of the solution.
- Some parts of the smart grid have very stringent response time requirements.
- Even in a small country there will be millions of entities involved, which make the logistics enormous due to:
  a) the initial installation of public-key certificates and private keys;
  b) the establishment of trust anchor information;
  c) the renewable of the above information;
  d) the protection of the above information.
- Revocation of public-key certificates needs be re-evaluated. Retrieving revocation lists or going to a certification (OCSP) server may create overhead for some low capacity entities leading to unacceptable response times.
- The certificate validation procedure needs to be optimized, possibly requiring very short certification paths.

- The threats to a smart grid PKI is different and more numerous than for most other environments and they may have devastating consequences.

22.3 Partners in smart rid

There are many diverse actors in smart grid that might have very different objectives; the potential actors are listed below. It is worth noting that it is not expected all the actors will get together to profile the use of PKI across all aspects of the smart grid. The following list may not be complete:

- smart meter manufacturers;
- solar energy inverter manufacturers;
- heat pump manufacturers;
- different types of manufacturers of electrical equipment;
- wind mill manufacturers;
- manufacturers of other types of renewable energy installations;
- manufacturers of electric vehicle load stations;
- transmission system operators (TSOs);
- distribution system operators (DSOs);
- balance responsible entities;
- combined heat and power plants;
- wind mill operators;
- traditional power plant operators.

22.4 NIST on smart grid – [b-NISTIR 7628-1] report

22.4.1 Introduction

The USA National Institute of Standards and Technology (NIST) published substantial material on smart grid security issues. Of particular interest is the [b-NISTIR 7628-1] report which comprises three volumes. The following is the NIST introduction of this report:

"Smart Grid technologies will introduce millions of new intelligent components to the electric grid that communicate in much more advanced ways (e.g., two-way communications, and wired and wireless communications) than in the past. This report is for individuals and organizations who will be addressing cyber security for Smart Grid systems. The privacy recommendations, the security requirements, and the supporting analyses that are included in this report may be used by strategists, designers, implementers, and operators of the Smart Grid, e.g., utilities, equipment manufacturers, regulators, as input to their risk assessment process and other tasks in the security lifecycle of a Smart Grid information system. This report focuses on specifying an analytical framework that may be useful to an organization. It is a baseline, and each organization must develop its own cyber security strategy for the Smart Grid. The information in this report serves as guidance to various organizations for assessing risk and selecting appropriate security requirements and privacy recommendations."

22.4.2 Volume 1 – High level requirements

[b-NISTIR 7628-1] is a 289-page document covering high-level requirements that could be a base document for further international standardization. It does not provide solutions, but it gives a comprehensive overview of the security issues.

[b-NISTIR 7628-1] introduces the following figure where the main components of smart grid are depicted.

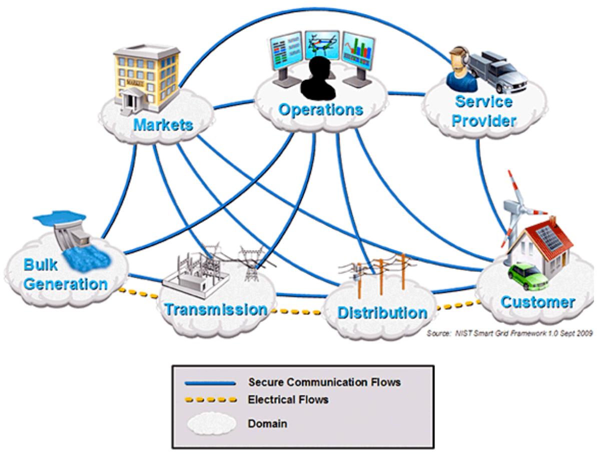


Figure 39 – Smart grid conceptual model

The report then goes on to subdivide each of the seven main actors into 49 types of actors with a short description of each actor type. The report identifies all the interactions among those actor types. It identifies 137 different types of communications. As an actor type may have many instantiations, it becomes quite a complex ICT network to support the smart grid. These 137 different types of communication are grouped into 22 logical interface categories. For each logical interface category, the report lists the technical high-level security requirements and a description of each requirement. There are in total 174 high-level security requirements, and each requirement has a short description of the required actions.

[b-NISTIR 7628-1] has a whole chapter on cryptography and key management. The report expresses concerns on processing and bandwidth constraints in the smart grid environment and on connectivity with PKI components (for example, with CAs and OCSP servers). It has some considerations for using shared secret keys using a key distribution centre (see clause 12.5), but misses to consider the possibility in favour of distributing public-key certificates.

The report has comprehensive sections on security issues encountered when establishing a PKI and a key management system, and it provides useful guidance to counter these issues.

22.4.3 Volume 2 – Privacy

[b-NISTIR 7628-2] is concerned with privacy issues. Volume 2 is mostly concerned with the privacy impact of smart metering. This volume is not in particular related to the deployment of PKI, although a PKI security solution can avoid or ensure that only through a major effort certain parties can get access to data that affects privacy. However, there are still privacy issues beyond such a protection.

Smart metering is the big issue. Smart metering allows the tariff to depend on different parameters, such as present load on the electric grid, availability of power from variable sources, such as wind power, etc. Smart metering allows probing of the electricity consumption down to a 15-minute interval. By doing this, the consumers are inspired to move consumptions to cheaper periods where there is ample supply. However, there are some implications.

There is the criminal attack caused by the disclosure of electricity use patterns. If the smart meter traffic can in some way be eavesdropped, this information may be used for criminal intent, for example for planning a burglary. This type of threat can be mitigated by proper security measures.

As data are available in a more granular form, not only is it possible to monitor the behaviour of the inhabitants of a dwelling but also to deduce information about the types of electric appliances used at different times. Data may reveal business activities and manufacturing procedures. It has been shown that by monitoring electricity usage every 15 minutes, it is possible to collect information about use pattern of electricity through the application of equipment electricity signatures.

22.4.4 Volume 3 – Analyses and references

[b-NISTIR 7628-3] provides useful information for the deployment of security for smart grid. The report considers different vulnerabilities, provides a bottom-up security analysis, and it analyses themes for cyber security research and development.

Volume 3 contains useful information for study, which could be a useful input for further smart grid security standardization.

22.5 Smart grid security standardization within IEC TC 57

IEC Technical Committee 57 develops and maintains International Standards for power systems control equipment, and systems including energy management systems (EMS) and supervisory control and data acquisition (SCADA).

Working Group 15 of TC 57, responsible for security, has developed the multi-part IEC TS 62351 standard. This series of standards has references to PKI, TLS and role-based access control (RBAC). Both TLS and RBAC require a backbone PKI.

22.5.1 [b-IEC TS 62351-1], Introduction to security issues

[b-IEC TS 62351-1] provides an introduction to the other parts of the IEC TS 62351 series, primarily to introduce the reader to various aspects of information security as applied to power system operations.

It has an important section on threats, but they are not directly related to the scope of this Technical Report, as they are mostly concerned with cyber security issues.

22.5.2 [b-IEC TS 62351-2], Glossary of terms

[b-IEC TS 62351-2] covers most of the terms used in the IEC TS 62351 series. It is not meant to be an exhaustive list.

22.5.3 [b-IEC TS 62351-3], Security for profiles that include TCP/IP

[b-IEC TS 62351-3] specifies how to provide confidentiality, tamper detection, and message level authentication for protocols that make use of TCP/IP as the underlying protocol stack.

Connections in this environment are almost permanent requiring renegotiation of symmetric cryptographic keys at intervals.

The part specifies:

- support of TLS 1.2, but for backward compatibility, it also supports TLS 1.0 and TLS 1.1;
- mutual authentication of entities;
- mandated use of MAC;
- support of both Diffie-Hellman agreement and RSA key exchange;
- a maximum public-key certificate size;
- a connection shall not be established based on a revoked or expired public-key certificate:
- a revoked public-key certificate shall cause an existing connection to be terminated;
- an expired public-key certificate shall not cause an existing connection to be terminated;
- support for multiple CAs as illustrated below in Figure 40.

Referring specifications will have to specify:

- the number of CAs to be supported;
- the mandatory TLS cipher suites to be supported;
- maximum time and/or number of messages between renegotiation of keys;

- maximum public-key certificate size.

[b-IEC TS 62351-3] does not specify:

- the content of a public-key certificate;
- possible naming structure;
- requirements on accuracy of information;
- recommended validity period;
- use and/or non-use of public-key certificate extensions;
- use or non-use of TLS extensions.

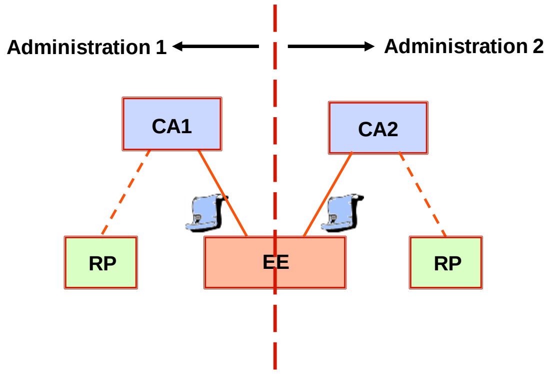


Figure 40 – Support of multiple CAs

Support of multiple CAs implies that the end-entity is part of different administrative domains. This allows each domain to establish its own enterprise PKI independent of PKIs of other domains. It has the advantages indicated in the text associated with Figure 37.

22.5.4 [b-IEC TS 62351-4], Security for profiles that include manufacturing message specification (MMS)

[b-IEC TS 62351-4] is concerned with security for protocols running on top of a protocol stack defined by the manufacturing message specification (MMS). MMS is running the top of an open systems interconnection (OSI) stack. This OSI stack may be a pure OSI stack or it may be the upper four OSI layers running on top of TCP/IP (or TLS) using the function as described by [b-IETF RFC 1006] or [b-IETF RFC 2126]. Only this mapping on top of TCP or TLS seems to be feasible.

TLS may or may not be used as stated in [b-IEC TS 62351-3]. As the association control service element (ACSE) is included in the protocol stack, the ACSE security feature may be utilized by providing public-key certificates in the ACSE header.

22.5.5 [b-IEC TS 62351-5], Security for IEC 60870-5 and derivatives (i.e., DNP 3)

[b-IEC TS 62351-5] specifies messages, procedures and algorithms to secure the operations of all protocols based on or derived from [b-IEC 60870-5].

This standard avoids the use of PKI for more simple applications. It uses the master/session key scheme described in clause 12.4.

Both session key generation/renewable and transmission of critical operations are protected by a challenge-response protocol.

There is no procedure defined for renewal of key update when the key is compromised. Lack of scalability of this procedure is a concern.

22.5.6 [b-IEC TS 62351-6], Security for IEC 61850 peer-to-peer profiles (e.g., GOOSE)

[b-IEC TS 62351-6] specifies messages, procedures, and algorithms to secure the operations of all protocols based on or derived from [b-IEC 61850].

22.5.7 [b-IEC TS 62351-7], Security through network and system management

[b-IEC TS 62351-7] defines network and system management (NSM) data object models that are specific to power system operations. These NSM data objects are used to monitor the health of networks and entities, to detect possible security intrusions, and to manage the performance and reliability of the information infrastructure.

22.5.8 [b-IEC TS 62351-8], role-based access control

[b-IEC TS 62351-8] defines role-based access control (RBAC). RBAC can be applied in a human-initiated communication or in a M2M (entity-to-entity) environment.

The RBAC specification distinguishes between accessing users (called subjects), roles and access rights (view, read, file, etc.). A subject may have different roles. A role may have different access rights.

The specification defines a number of roles and rights that may be extended for particular types of equipment and environments.

The specifications for mapping between roles and rights are generated into the object being the target for access.

The assignment of roles to a subject may be stored in a public-key certificate or in an attribute certificate.

The assignment may be provided in a public-key certificate if the role assignment is long lived. The name of the subject is then supplied in the subject field of the public-key certificate, while the role assignments are given by an attribute in the subject directory attribute extension (see clause 4.3.6).

The assignment may be provided in an attribute certificate if the role assignment is short lived.

22.5.9 IEC TS 62351-9, key management

IEC TC 57/WG 15 has not issued any document so far on key management.

23 Smart grid and substations

23.1 Substation description

A substation is a part of an electrical generation, transmission, and distribution system. A substation is a location where electricity transmission/distribution voltage-levels are changed and where switching is carried out, as opposed to a power station, where electricity is generated. A substation, therefore, is a yard or building in which there are transformers, switchgears, bus bar systems, and protection systems.

Generally speaking, substations are unmanned, relying on automated operation and on remote supervision and control.

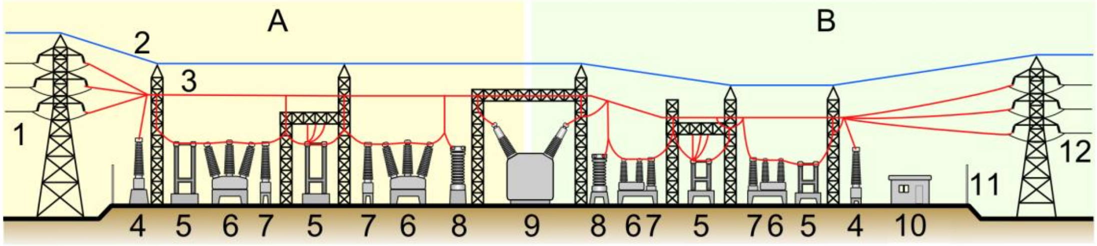


Figure 41 – Electrical view of a substation

Figure 41 shows an example of a substation. A substation comprises different electrical units that have to interwork in a manner outside the scope of this Technical Report. The interworking is accomplished by communication among the so-called intelligent electronic devices (IEDs) each serving a substation unit.

In the figure, '*A*' is the primary power line side, while '*B'* is the secondary power line side. '1' are the primary power lines; '2' is the ground wire; '3' are the overhead lines; '4' is a transformer for the measurement of electric voltage, '5' is a disconnect switch, '6' is a circuit breaker; '7' is a current transformer; '8' is a lightning arrester; '9' is the main transformer; '10' is the control building; '11' is the security defence, and '12' are the secondary power lines.

IED is a microprocessor-based controller for a power system unit, for example, circuit breaker, a protective relay, etc. It receives digitalized data from sensors and power equipment and it issues commands to maintain the desired status of the power grid. IED is the interface to the controlling ICT network.

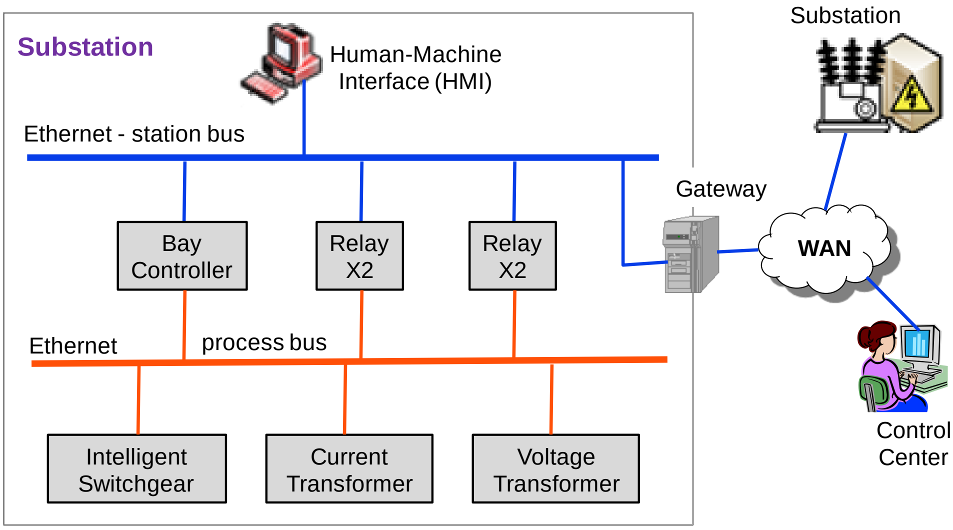
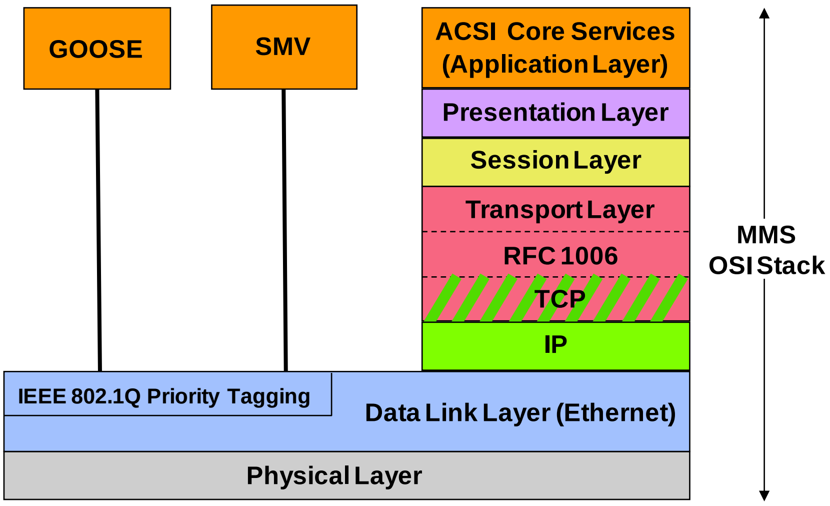


Figure 42 – ICT view of substation

Figure 42 shows an ICT view of a substation. The communication specifications for smart grid substations are given in the multi-part standard [b-IEC 61850], Communication networks and systems in substations.

23.2 Substation communications architecture



ACSI	=	Abstract Communications Service Interface
GOOSE	=	Generic Object Oriented Substation Event
SMV	=	Sampled Measured Values
MMS	=	Manufacturing Message Specification
OSI	=	Open Systems Interconnection

Figure 43 – IEC 61850 protocol stacks

[b-IEC 61850] is a multi-part standard that specifies the different protocols for communication among IEDs in substations. [b-IEC 61850] considers the different communications requirements.

Figure 43 illustrates some parts of the IEC 61850 protocol stack for substation communication; this is briefly described as follows:

The abstract communications service interface (ACSI) is an abstract interface for the different services as defined by [b-IEC 61850-7-2]. It utilizes the service as provided by the underlying protocol stacks. For its core services, it uses the protocol stack provided by the manufacturing message specification (MMS), as defined by [b-ISO/IEC 9506-1] and [b-ISO/IEC 9506-2]. MMS depends on the open systems interconnection (OSI) protocol stack. As for most other OSI applications, the OSI protocol stack is mapping onto TCP/IP as specified in [b-IETF RFC 1006] (or in a somewhat extended [b-IETF RFC 2126]). For time critical services, ACSI is mapped directly to the data link layer (Ethernet).

[b-IETF RFC 1006] considers TCP as being part of a network layer protocol, although it is a transport layer protocol.

[b-IEC 62351-4] covers the security aspects of ACSI based on MMS (see clause 22.5.4).

The generic object oriented substation event (GOOSE) protocol is used in situations where there are very rigorous response requirements for substation controls. It is a protocol for the fast transmission of substation events, such as commands, alarms, indications, and messages. Some types of commands require a response within less than 4-10 ms. This time-frame includes time for generating a message, transmit it to another entity which has to process the message and ensure that the message is not a fake message intended to cause damage.

23.3 Use of Ethernet

As seen from Figure 43, GOOSE data are directly embedded into Ethernet data packets bypassing all other protocol layers. A single GOOSE message sent by an IED can be received and used by several receivers (multicast or broadcast).

GOOSE uses virtual local access network (VLAN) and priority tagging as per [b-IEEE 802.1Q] to allow a separate virtual network within the same physical network and to set the appropriate message priority levels.

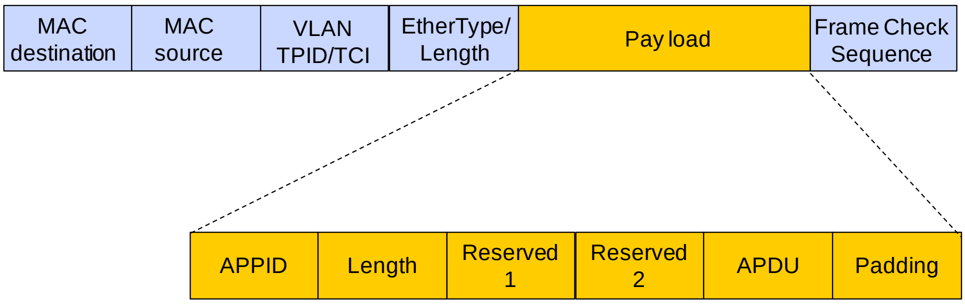


Figure 44 – Ethernet frame structure

Figure 44 depicts the traditional Ethernet frame structure with destination and source media access control (MAC) addresses.

The VLAN TPIP/TCI field provides information for the IEEE 802.1Q functionality.

The EtherType/Length field is an EtherType field if the value is '0600'H or greater. Otherwise, it is a length field for the payload. The field indicates that it is an EtherType field, where the EtherType is used to indicate what protocol is encapsulated in the payload field. The general structure is as shown in the lower part of Figure 44.

The exact meaning of the different fields is not significant for the purpose of this Technical Report except for the discussion in clause 23.4.

23.4 Substation security aspects

Security for GOOSE and SMV is specified in [b-IEC 62351-6]. Currently, the specification is not clear and contains some mistakes, but from the text, it looks like security is provided in the form of a simple digital signature by generating a hash over the extension and then encrypting this hash using the private key. However, the specification calls it a message authentication code (MAC) (see clause 10).

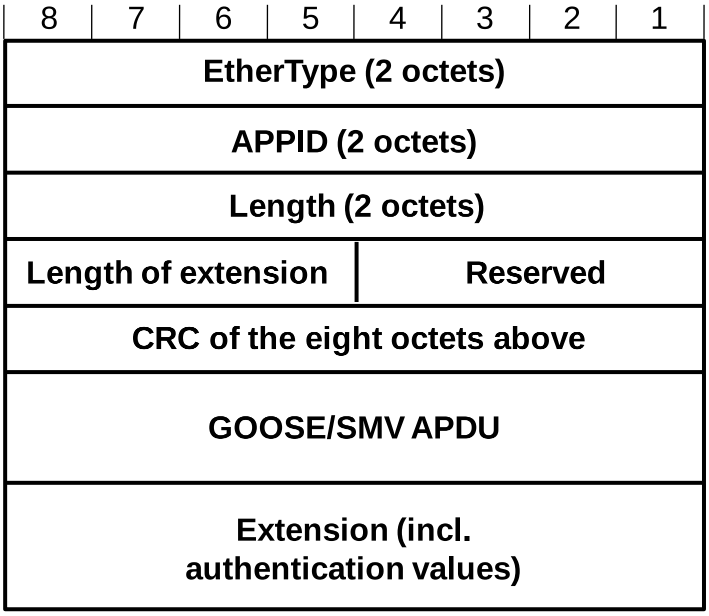


Figure 45 – GOOSE and SMV Ethernet types

Figure 45 shows the part of an Ethernet structure specifically for GOOSE and SMV application-protocol-data-units (APDUs). A GOOSE APDU is identified by an EtherType '88B8'H and SMV APDU is identified by an EtherType 88BA'H.

The two reserved fields shown in Figure 44 are used for a length field that shows the lengths of a GOOSE/SMV specific security extension and for a cyclic redundancy check (CRC).

24 Smart grid consumers

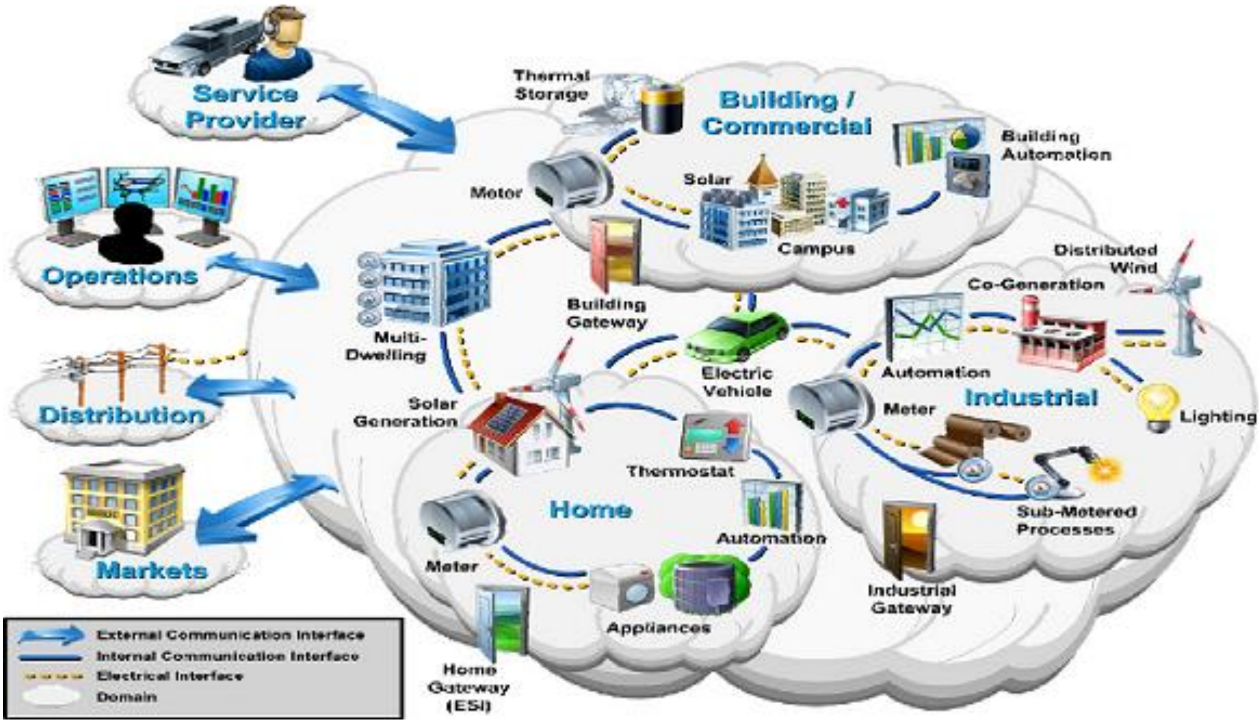


Figure 46 – Smart grid consumers

Figure 46 shows an overview of the networks for different smart grid consumers.

In the following clauses, only home-related issues are covered in some details. This is closely related or overlaps with the so-called home area networks (HAN) considering home automation in general.

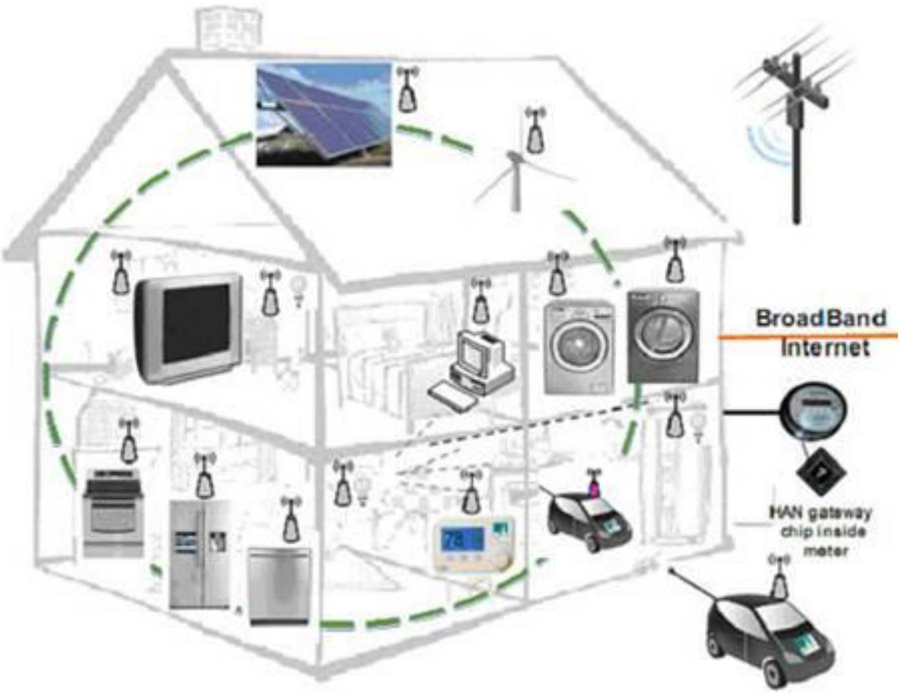


Figure 47 – Home area network example

Figure 47 shows an example of a home area network where a wireless LAN is used for the communication among entities within the home. Other techniques may be used, such as a wired Ethernet, and point-to-point connection.

A home area network is an obvious entry point for malware that might propagate to more critical parts of the smart grid ICT infrastructure. In particular, a wireless LAN is in particular vulnerable, as no physical contact is necessary for an intruder.

25 Threats specific to smart grid

A smart grid environment has more access points and multiple entities, which all open the door to more potential cybersecurity breaches than in any other environment.

25.1 Physical threats

Smart meters, photovoltaic inverters, etc., are placed in consumer areas to which there is very easy access. Such devices are difficult to protect physically and can easily be made the entry point for all kinds of malware. It is necessary to separate the network serving the consumer units as far as possible from the network serving the more critical units.

25.2 Configuration problems

There are many devices and their communication channels, and thus, there are many possibilities for configuration errors.

The generation and distribution of public-key certificates, private keys, trust anchor information, etc., may be error prone.

Although general configuration errors may impose security threats, they are outside the scope of this Technical Report. However, PKI configuration issues are within the scope of that Technical Report.

25.3 Software problems

Security weaknesses are known to exist in software. Software vendors daily distribute updates to patch security problems in their products. This issue also applies to smart grid software. Procedures for keeping the smart grid software updated at all times are required. With millions of very different entities with different software vendors, this is not a trivial problem. However, this issue is currently outside the scope of this Technical Report.

25.4 Denial-of-service (DoS) attacks

Clause 17.13 gives a general overview of DoS attacks. All these types of attacks have to be considered when analysing smart grid security. The following points consider those types of DoS attacks that are specific to smart grid:

– If a public-key certificate of an entity is deleted or replaced with a bogus public-key certificate, this device can no longer communicate with the other entities and is thereby put out of service.

– If a public-key certificate for an entity is revoked, this device is out of service.

– If the revocation service is out of order, it is not possible to validate public-key certificates and the whole network is out of service.

– Communication overhead may cause delays in the ICT network, which may result in faulty control of the electric grid.

– Destruction of the private key will put an entity out of operation.

– If the trust anchor information in an entity were replaced by unauthorized information, such a device would accept a public-key certificate from a hostile entity and may get into communication with such an entity.

26 Smart metering and smart meter hacking

A **smart meter** is usually an electrical meter that records the consumption of electric energy in intervals of an hour or less and communicates that information, at least on a daily basis, back to the utility company for monitoring and billing purposes.

Smart meters are placed in consumer premises and therefore are easy targets for attacks.

– The consumer may want to manipulate the meter operation to save expenses.

– Outsiders may want to intercept the traffic to monitor the habits of the inhabitants for malicious purposes.

– A hijacked smart meter may be the malware entry point into the smart grid ICT network.

Clause 22.4.3 describes some privacy concerns with smart metering.

27 Possible updates to [b-ITU-T X.509]

27.1 White listing

An update to [b-IEC TS 62351-3] specifies a requirement for whitelisting. White listing can mean different things:

a) proof of existence of a particular public-key certificate, for example, using an OCSP extension; or

b) a positive list of public-key certificates that a particular entity accepts. Public-key certificates not on the list shall not be accepted by that entity.

Proof of existence might also be an interesting subject, but the requirement has to be analysed more carefully.

27.2 Fast validity checking

Currently, the use of OCSP is the quickest way to check the status of a public-key certificate but, in some cases, even this technique maybe time consuming (see clause 23.2).

The OCSP stapling, as described in clause 13.7, may be a solution when TLS is used. However, highly time-critical protocols are not based on TLS.

A possible solution might be some kind of subscription service. An entity could subscribe on revocation warnings for particular public-key certificates. Such a service could possibly be combined with the requirement specified in clause 27.1 b). This might involve, in some way, the trust broker referenced in clause 27.4.

27.3 Application specific public-key certificate formats

For fast processing of public-key certificates, there may be a requirement for having public-key certificate formats optimized for a particular purpose. In addition, specific encoding rules could be applied to specific formats, as suggested below.

```
publicKeyCertificate CHOICE {
  old             TSBCertificate,
  fastPKCert [0] FastPKCert,
  miniMobil  [1] MiniMobile,
  etc.
  ... }
```

Using basic encoding rules (BER)/distinguished encoding rules (DER) encoding would not change the encoding of current public-key certificates. A public-key certificate used for a smart grid substation may specify a more efficient encoding rule than BER. Such a format could be:

```
FastPKCert ::= SECUENCE {
  serialNumber          CertificateSerialNumber,
  issuer                dnsName,
  validity              Validity,  -- only using UTC time
  subject               MACaddress,
  subjectPublicKeyInfo  SubjectPublicKeyInfo,
  trustBroker           MACaddress,
  extensions            Extensions,
  ... }
```

27.4 Inclusion of trust broker

As discussed in clause 8, in certain environments, inclusion of a trust broker will improve security. This concept is already under development as an addition to [b-ITU-T X.509]. However, this concept could be even further explored, for example, for a smart grid substation.

In the gateway of a substation (see Figure 42), a trust broker could be included to have complete status of all the public-key certificates, possibly by a subscription service from a CA.

27.5 Additional public-key certificate extensions

Additional public-key certificate extensions may be required. For example, an extension that provides the address of a trust broker.

28 Requirement for PKI profiling

Such a document should cover all issues necessary to establish a PKI. This could include the following areas.

28.1 Recommendation on public-key certificate content

A public-key certificate has a number of fields to be filled with information. Guidelines for how such fields should be filled for different environments should be part of such a profiling document, which should concentrate on areas not covered by other organizations, like The CA Browser Forum (see clause 18.2).

In constrained environments, it is particularly important to have strict specification for the public-key certificate content.

28.2 Public-key certificate extensions

The number of extensions to be included should be kept down to a minimum.

28.3 Random number generation for nonce, cryptographic keys, etc.

There are attacks identified on PKI systems using poor random number generators. The random number generator shall therefore have a high level of entropy (see clause 7.2). In addition, when creating RSA key pairs it is important to change both of the large prime integers that go into the keys (see clause 17.5). Guidance on these issues should be included in the proposed specification.

28.4 Hash collision avoidance

As hash collision is a security issue (see clause 17.8), guidance on the selection of hashing algorithms should be provided.

28.5 Validating procedures

Validation of a public-key certificate can be a time consuming exercise. To make validation efficient, it is required to have procedures and guidance on how to structure PKI.

28.6 Revocation issues

Revocation of public-key certificates is a major issue. In some environments, it is important to guard against malicious attempt to invalidate otherwise genuine public-key certificates. This could set a series of critical entities out of operation.

28.7 Machine readable certificate policies

Currently, certificate policy documents are documents readable only by humans assuming that the human user reads such a document before accepting a public-key certificate. However, in some environments there is no human user involved. In any case, it could be useful to have machine-readable certificate policy documents. A template for such a machine-readable certificate policy should be developed.

29 Requirement for PKI management procedure specification

29.1 Introduction

ITU-T Study Group 17 is progressing draft Recommendation ITU-T X.pki-em: Public-key infrastructure: Establishment and maintenance. It has been recognized that such procedures are necessary for the success of large scale deployment of PKIs.

The following clauses elaborate some ideas on the possible content of such a Recommendation.

29.2 Boot strapping and maintenance of PKI information

In an M2M environment, the establishment and maintenance of a PKI cannot rely on human participation, which is the case for typically web-based PKIs.

The following outlines a possible procedure, but other techniques may also be developed.

29.2.1 Two-PKI approach

For security and maintenance reasons, two different PKIs should be established.

A management PKI supports the establishment and the maintenance of an operational PKI, while the operational PKI supports the normal operation of the M2M network.

The management PKI can remain stable over a longer period. It has to be established mainly using manual procedures, while the operational PKI may be more dynamic and be maintained using automated procedures.

29.2.2 Management PKI establishment

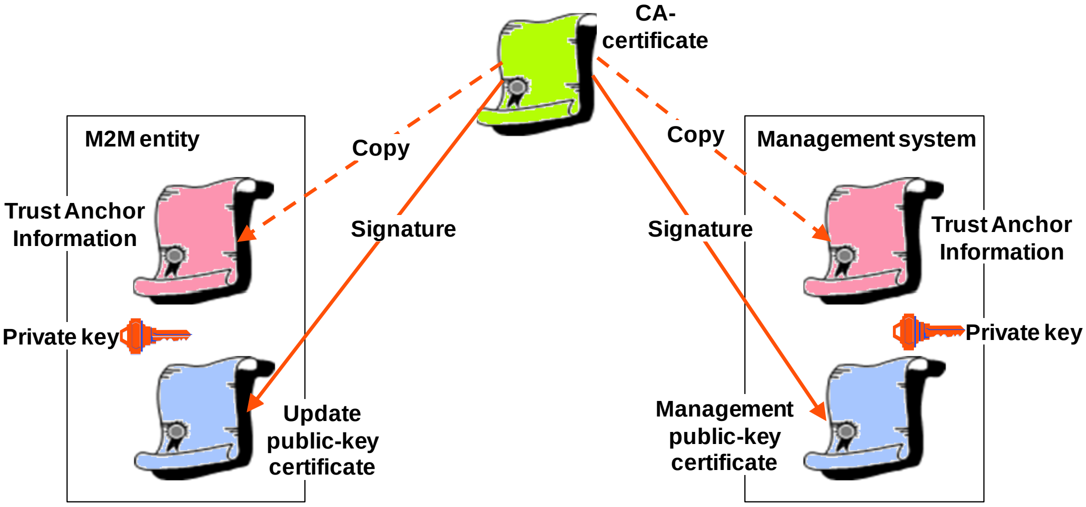


Figure 48 – Management PKI

A management PKI could include:

- A specific public-key certificate for each entity under a particular management domain together with a corresponding private key. Such a public-key certificate is referred to below as an update public-key certificate.
- A specific public-key certificate for the management system together with a corresponding private key. This public-key certificate is referred to below as a management public-key certificate.
- A specific CA dedicated to sign update public-key certificates and the management public-key certificate. Its CA-certificate may be a self-signed CA-certificate.
- The CA-certificate is placed in all the entities and in the management system as trust anchor information.

When an M2M entity is made ready for installation, as part of this pre-installation procedure, the update public-key certificate, the corresponding private key and the trust anchor information are placed in secure storage (for example, a hardware security module as discussed in clause 11). This load shall be performed locally using strict security measures, as illustrated in Figure 48. Using an asymmetric cryptography, an M2M entity and the management system can now securely authenticate each other.

If it becomes necessary to replace the management PKI information in one or more M2M entities, this could be done by local procedures.

It is now possible to establish a secure channel between the management system and the M2M entity, for example, by establishing a TLS communication or by encrypting the information using the public key of the sender.

The update public-key certificate is very essential for establishing and maintaining the operational PKI.

As the management PKI is only used occasionally and not during a normal operation, it is less vulnerable to attacks.

29.2.3 Operational PKI establishment

The secure channel established by the use of the management PKI may now be used for the transfer of operational PKI information to the M2M entities both initially and for maintenance purposes.

The public-key certificates used during normal operations should be signed by a CA different from the CA used in the management PKI. The public-key certificate and the corresponding private key should be generated by the CA (or management system) to mitigate collision attacks (see clause 17.8).

29.3 Central maintenance of information

A network with many entities controlled by embedded software requires automated procedures to keep the supporting PKI updated. This is in particular relevant to M2M environments, including smart grid. The issue is in particular important for smart grid, as millions of entities need to be PKI protected.

The following types of information need to be distributed:

- private keys (or means for local generation);
- end-entity public-key certificates for the entities;
- trust anchor information;
- cross certifications information; and
- possible intermediate CA information, if appropriate.

It will be necessary to establish some kind of database or ITU-T X.500/LDAP directory to maintain, where each network entity is represented as an entry holding the relevant information. Such information could be:

a) End-entity public-key certificate information in a format for quick access:
   - the entity public-key certificate itself;
   - sequence number;
   - issuer;
   - validity period; and
   - extensions, where each extension is possibly represented by a directory attribute.

b) Trust anchor information in the form of a CA-certificate and possible intermediate CA-certificates:
   - the CA-certificate itself;
   - sequence number;
   - issuer, which shall be the same as the owner of the entry;
   - validity period; and
   - extensions, where each extension is possibly represented by a directory attribute and where the basic constraints extension shall be present.

c) Cross certification information that may represented by the `crossCertificatePair` attribute type defined by [b-ITU-T X.509].

The following tasks have to be undertaken:

a) Exact definition of the information to be maintained by a directory system or a similar database system. The specification could specify generation of the necessary directory schema information (attribute types, object classes, matching rules) with the understanding that such definition could be translated to database information types.

b) A database structure that allows modelling of the complete network (this may already exist in some way).

c) A database structure that can keep track of the PKI structure (what entities have public-key certificates issued by what CAs).

d) Procedures on how to detect when public-key certificates are close to expiring and therefore to be replaced.

e) Procedures for how to recover from compromised private key.

f) Procedures for when a public-key certificate has been revoked for any other reason.

f) Procedures for how to recover from a compromised CA.

It should be analysed how much of above needs to be standardized for the purpose of interworking. Only the requirements may have to be listed, but not necessarily have they are fulfilled. Some aspects, however, may need standardization.

30 Conclusion

There are many standardization organizations that refer to ITU-T X.509 public-key certificates and other PKI concepts. Many, or most, of these specifications make only rudimentary PKI specifications not necessarily sufficient for the interworking of implementations. Very little is provided on how to establish a secure PKI and how to guard against cyber-attacks.

This Technical Report has identified requirements for additional work in the following areas:

a) It is the plan to extend [b-ITU-T X.509] to include a specification for a trust broker component type. Such a trust broker may require an extension that provides the address of the trust broker.

b) It is to be investigated if there is a need to provide more details on PKI on top of [b-ITU-T X.1164]

c) Development of extensions to the public key infrastructure (PKI) of [b-ITU-T X.509] to support environments with resource and/or time constraints, such as wireless PKI (WPKI) and smart grid, with the purpose to optimize PKI in the mobile environment, especially for developing countries.

d) A more generalized wireless PKI solution should enable secure mobile access to e-banking, e-government, e-commerce, e-health, etc.

e) It is to be investigated if cloud computing will bring new requirements to [b-ITU-T X.509], e.g., to consider the need for guidelines that specify how PKI might be established as a service within the cloud computing environment.

f) Development of an ITU-T Recommendation that provides guidance and makes recommendations for the deployment of PKI, especially within environments outside traditional web environments like M2M.

g) Standardize fast validity checking procedures of public-key certificates e.g., some kind of subscription procedure, possibly in combination with a trust broker especially in a smart grid environment. In that context, consider standardization of requirements and solutions for optimized formats of public key certificate formats, and/or specific encoding rules which could be applied to specific formats, e.g., in smart grid environments).

h) Development of an ITU-T Recommendation that establishes specifications to establish and maintain PKI for very large networks, like the smart grid.

i) Consideration of standards for PKI profiling such as guidelines for how public-key certificate fields should be filled for different environments (including constrained environments) (in cooperation with the CA Browser Forum).

j) Develop guidelines that could be available to specify the generation of random prime numbers, and provide guidance on the selection of hashing algorithms.

k) Develop guidelines for PKI architectures, which supports efficient procedures for the validation of public-key certificates.

l) Consider developing machine-readable certificate policy documents by using, for example, the approach of a template.

m) Address the problem of boot strapping and maintenance of PKI information as this is of importance in M2M environments, for example, the two PKI approaches should be studied where a management PKI and an operational PKI are available and are used jointly. Explore the standardization need for central maintenance of information.

There is an enormous amount of key management activities by almost every standardization organization. It is therefore not recommended that ITU-T Study Group 17 undertake such work. However, there might be a need to classify all these specifications and to identify how they relate to each other.

Glossary

3G	Third Generation
4G	Fourth Generation
AA	A*ttribute Authority*
ACSE	Association Control Service Element
ACSI	Abstract Communications Service Interface
AEAD	Authenticated Encryption with Associated Data
APDU	Application-Protocol-Data-Unit
ASiC	Associated Signature Container
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rule
CA	Certification Authority
CEN	Comité Européen de Normalisation (European Committee for Standardization)
CMS	Cryptographic Message Syntax
CP	Certificate Policy
CPS	Certificate Practice Statement
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CSP	Cryptographic Service Providers
DDoS	Distributed Denial-of-Service
DER	Distinguished Encoding Rule
DNP	Distributed Network Protocol
DNS	Domain Name System
DoS	Denial-of-Service
DRM	Digital Rights Management
DSO	Distribution System Operator
EE	End-Entity
EESSI	European Electronic Signature Standardization Initiative
EKMI	Enterprise Key Management Infrastructure
EMS	Energy Management System
ESI	Electronic Signatures and Infrastructure
ETSI	European Telecommunications Standards Institute
EV	Extended Validation
GOOSE	Generic Object Oriented Substation Event
GSM	Global System for Mobile Communications

gTLD	Global Top Level Domain
HAN	Home Area Network
HMAC	*Hash-based Message Authentication Code*
HMI	*Human-Machine Interface*
HSM	Hardware Security Module
HSTS	HTTP Strict Transport Security
HTTP	HyperText Transfer Protocol
HTTPS	Secure HyperText Transfer Protocol
ICCID	Integrated Circuit Card ID
ICT	Information and Communication Technology
ID	Identifier, Identity
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPSec	Internet Protocol Security
IV	Initialization Vector
KDC	Key Distribution Centre
KMIP	Key Management Interoperability Protocol
KTC	Key Translation Centre
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
M2M	Machine-to-Machine
MAC	*Message Authentication Code*
MAC	*Media Access Control*
MITM	Man-in-the-Middle
MMS	Manufacturing Message Specification
MPS	Mobile Payment System
NGN	Next-Generation Network
NIST	National Institute of Standards and Technology
NSM	Network and System Management
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
OID	Object Identifier

OMA	Open Mobile Alliance
OSI	Open Systems Interconnection
PAdES	PDF Advanced Electronic Signatures
PC	Personal Computer
PDF	Portable Document Format
PFS	Perfect Forward Secrecy
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure X.509
PRF	Pseudo Random Function
RBAC	Role-Based Access Control
RP	Relying Party
RSA	Rivest-Shamir-Adleman
SA	Security Association
SaaS	Software-as-a-Service
SAML	Security Assertion Markup Language
SCADA	Supervisory Control And Data Acquisition
SIM	Subscriber Identity Module
SMS	Short Message Service
SMV	Sampled Measured Values
SOAP	*Simple Object Access Protocol*
SSID	*Service Set IDentification*
SSL	Secure Socket Layer
TA	Trust Anchor
TCI	Tag Control Information
TCP	Transmission Control Protocol
TLD	Top Level Domain
TLS	Transport Layer Security
TPID	Tag Protocol IDentifier
TSO	Transmission System Operator
UICC	Universal Integrated Circuit Card
URL	Uniform Resource Locator
USB	Universal Serial Bus
UTC	Coordinated Universal Time
VLAN	Virtual Local Access Network
VPN	Virtual Private Network
WAN	Wide Area Network

WAP	Wireless Application Protocol
WEP	Wired Equivalent Privacy
WG	Working Group
Wi-Fi	Wireless Fidelity
WPA	Wi-Fi Protected Access
WPKI	Wireless PKI
WTLS	Wireless TLS
XAdES	XML Advanced Electronic Signatures
XML	eXtensible Markup Language

Bibliography

[b-ITU-T X.500] The series of ITU-T X.500 Recommendations | ISO/IEC 9594-all parts, *Information technology − Open Systems Interconnection – The Directory.*

[b-ITU-T X.501] Recommendation ITU-T X.501 (2012) | ISO/IEC 9594-2:2014, *Information technology – Open Systems Interconnection – The Directory: Models.*

[b-ITU-T X.509] Recommendation ITU-T X.509 (2012) | ISO/IEC 9594-8:2014, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks.*

[b-ITU-T X.519] Recommendation ITU-T X.519 (2012) | ISO/IEC 9594-5:2014, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications.*

[b-ITU-T X.520] Recommendation ITU-T X.520 (2012) | ISO/IEC 9594-6:2014, *Information technology – Open Systems Interconnection – The Directory: Selected attribute types.*

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

*[*b-ITU-T X.1141*]* *Recommendation ITU-T X.1141 (2006), Security Assertion Markup Language (SAML 2.0).*

[b-ITU-T X.1164] Recommendation ITU-T X.1164 (2012), *Use of service providers' user authentication infrastructure to implement public key infrastructure for peer-to-peer networks.*

[b-ITU-T Y.2704] Recommendation ITU-T Y.2704 (2010), *Security mechanisms and procedures for NGN.*

[b-ITU-T Y.2740] Recommendation ITU-T Y.2740 (2011), *Security requirements for mobile remote financial transactions in next generation networks.*

[b-ITU-T Y.2741] Recommendation ITU-T Y.2741 (2011), *Architecture of secure mobile financial transactions in next generation networks.*

[b-CABF-PTC] CA/Browser Forum (2013), *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates*, v.1.1.3.

[b-CABF-EVC] CA/Browser Forum (2014), *Guidelines for the Issuance and Management of Extended Validation Certificates*.

[b-ETSI TS 101 456] ETSI TS 101 456 V1.4.3 (2007), *Electronic Signatures and Infrastructures (ESI); Policy requirements for certification authorities issuing qualified certificates.*

[b-ETSI TS 102 042] ETSI TS 102 042 V2.1.1 (2009), *Electronic Signatures and Infrastructures (ESI); Policy requirements for certification authorities issuing public key certificates.*

[b-ETSI TR 102 203] ETSI TR 102 203 V1.1.1 (2003), *Mobile Commerce (M-COMM); Mobile Signatures; Business and Functional Requirements.*

[b-ETSI TS 102 204] ETSI TS 102 204 V1.1.4 (2003), *Mobile Commerce (M-COMM); Mobile Signature Service; Web Service Interface.*

[b-ETSI TS 102 207] ETSI TS 102 207 V.1.1.3 (2003), *Mobile Commerce (M-COMM); Mobile Signature Service; Specifications for Roaming in Mobile Signature Services.*

[b-ETSI TS 133 220] ETSI TS 133 220 V7.6.0 (2006), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Generic Authentication Architecture (GAA); Generic bootstrapping architecture*. (3GPP TS 33.220 version 7.6.0 Release 7).

[b-IEC 60870-5] IEC 60870-5-SER Ed. 1.0 (2013), *Telecontrol equipment and systems - Part 5: Transmission protocols - ALL PARTS.*

[b-IEC 61850] IEC 61850-SER Ed. 1.0 (2013), *Communication networks and systems in substations - ALL PARTS.*

[b-IEC 61850-7-2] IEC 61850-7-2 Ed. 2.0 (2010), *Communication networks and systems for power utility automation - Part 7-2: Basic information and communication structure - Abstract communication service interface (ACSI).*

[b-IEC 61850-8-1] IEC 61850-8-1 Ed. 2.0 (2011), Communication networks and systems for power utility automation - Part 8-1: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3.

[b-IEC 61850-9-2] IEC 61850-9-2 Ed. 2.0 (2011), Communication networks and systems for power utility automation - Part 9-2: Specific communication service mapping (SCSM) - Sampled values over ISO/IEC 8802-3.

[b-IEC TS 62351-1] IEC TS 62351-1 Ed. 1.0 (2007), *Power systems management and associated information exchange - Data and communications security - Part 1: Communication network and system security - Introduction to security issues.*

[b-IEC TS 62351-2] IEC TS 62351-2 Ed. 1.0 (2008), *Power systems management and associated information exchange - Data and communications security - Part 2: Glossary of terms.*

[b-IEC TS 62351-3] IEC TS 62351-3 Ed. 1.0 (2007), *Power systems management and associated information exchange - Data and communications security - Part 3: Communication network and system security - Profiles including TCP/IP*.

[b-IEC TS 62351-4] IEC TS 62351-4 Ed. 1.0 (2007), *Power systems management and associated information exchange - Data and communications security - Part 4: Profiles including MMS.*

[b-IEC TS 62351-5] IEC TS 62351-5 Ed. 2.0 (2009), *Power systems management and associated information exchange - Data and communications security - Part 5: Security for IEC 60870-5 and derivatives.*

[b-IEC TS 62351-6] IEC TS 62351-6 Ed. 1.0 (2007), *Power systems management and associated information exchange - Data and communications security - Part 6: Security for IEC 61850.*

[b-IEC TS 62351-7] IEC TS 62351-7 Ed. 1.0 (2010), *Power systems management and associated information exchange - Data and communications security - Part 7: Network and system management (NSM) data object models.*

[b-IEC TS 62351-8] IEC TS 62351-8 Ed. 1.0 (2011), *Power systems management and associated information exchange - Data and communications security - Part 8: Role-based access control.*

[b-IEEE 802.1Q] 802.1Q-2011 - IEEE Standard for Local and metropolitan area networks--Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks [b-IETF RFC 1006] IETF RFC 1006 (1987), *ISO Transport Service on top of the TCP Version: 3.*

[b-IETF RFC 2126] IETF RFC 2126 (1997), *ISO Transport Service on top of TCP (ITOT).*

[b-IETF RFC 2631] IETF RFC 2631 (1999), *Diffie-Hellman Key Agreement Method.*

[b-IETF RFC 3279] IETF RFC 3279 (2002), *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.*

[b-IETF RFC 3447] IETF RFC 3447 (2003), *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1.*

[b-IETF RFC 3526] IETF RFC 3526 (2003), *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE).*

[b-IETF RFC 3647] IETF RFC 3647 (2003), *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework.*

[b-IETF RFC 3749] IETF RFC 3749 (2004), *Transport Layer Security Protocol Compression Methods.*

[b-IETF RFC 4301] IETF RFC 4301 (2005), *Security Architecture for the Internet Protocol.*

[b-IETF RFC 4302] IETF RFC 4302 (2005), *IP Authentication Header*.

[b-IETF RFC 4303] IETF RFC 4303 (2005), *IP Encapsulating Security Payload (ESP).*

[b-IETF RFC 4511] IETF RFC 4511 (2006), Lightweight Directory Access Protocol (LDAP): The Protocol.

[b-IETF RFC 4835] IETF RFC 4835 (2007), *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH).*

[b-IETF RFC 5019] IETF RFC 5019 (2007), *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments.*

[b-IETF RFC 5246] IETF RFC 5246 (2008), *The Transport Layer Security (TLS) Protocol Version 1.2.*

[b-IETF RFC 5280] IETF RFC 5280 (2008), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

[b-IETF RFC 5652] IETF RFC 5652 (2009), *Cryptographic Message Syntax (CMS).*

[b-IETF RFC 5746] IETF RFC 5746 (2010), *Transport Layer Security (TLS) Renegotiation Indication Extension.*

[b-IETF RFC 5751] IETF RFC 5751 (2010), *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification*.

[b-IETF RFC 5996] IETF RFC 5996 (2010), *Internet Key Exchange Porotcol Version 2 (IKEv2).*

[b-IETF RFC 6066] IETF RFC 6066 (2011), *Transport Layer Security (TLS) Extensions: Extension Definitions.*

[b-IETF RFC 6151] IETF RFC 6151 (2011), *Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms.*

[b-IETF RFC 6176] IETF RFC 6176 (2011), *Prohibiting Secure Sockets Layer (SSL) Version 2.0.*

[b-IETF RFC 6265] IETF RFC 6265 (2011), *HTTP State Management Mechanism.*

[b-IETF RFC 6797] IETF RFC 6797 (2012), *HTTP Strict Transport Security (HSTS).*

[b-IETF RFC 6960] IETF RFC 6960 (2013), *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP.*

[b-ISO 9506-1] ISO 9506-1:2003, *Industrial automation systems -- Manufacturing Message Specification -- Part 1: Service definition.*

[b-ISO 9506-2] ISO 9506-2:2003, *Industrial automation systems -- Manufacturing Message Specification -- Part 2: Protocol specification.*

[b-ISO/IEC 9797-1] ISO/IEC 9797-1:2011, *Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher.*

[b-ISO/IEC 9797-2] ISO/IEC 9797-2:2011, *Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-function.*

[b-ISO/IEC 11770-1] ISO/IEC 11770-1:2010, *Information technology – Security techniques – Key management – Part 1: Framework.*

[b-ISO/IEC 11770-2] ISO/IEC 11770-2:2008, *Information technology – Security techniques – Key management – Part 2: Mechanisms using symmetric techniques.*

[b-ISO/IEC 11770-3] ISO/IEC 11770-3:2008, *Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques.*

[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 11770-5] ISO/IEC 11770-5:2011, *Information technology – Security techniques – Key management – Part 5: Group key management.*

[b-ISO 17090-1] ISO 17090-1:2013, *Health informatics – Public key infrastructure – Part 1: Overview of digital certificate services.*

[b-ISO 17090-2] ISO 17090-2:2008, *Health informatics – Public key infrastructure – Part 2: Certificate profile.*

[b-ISO 17090-3] ISO 17090-3:2008, *Health informatics – Public key infrastructure – Part 3: Policy management of certification authority.*

[b-ISO 21188] ISO 21188:2006, *Public key infrastructure for financial services -- Practices and policy framework.*

[b-ISO/IEC 27001] ISO/IEC 27001:2013, *Information technology – Security techniques – Information security management systems – Requirements.*

[b-ISO/IEC 27002] ISO/IEC 27002:2013, *Information technology – Security techniques – Code of practice for information security controls.*

Information technology -- Security techniques -- Code of practice for information security controls

[b-NIST FIPS 140-2] FIPS PUB 140-2 (2001), *Security Requirements for Cryptographic Modules.*

[b-NIST 800-57-1] NIST 800-57-1 (2012), *Recommendation for Key Management – Part 1: General (Revision 3).*

[b-NIST 800-57-2] NIST 800-57-2 (2005), *Recommendation for Key Management – Part 2: Best Practices for Key Management Organization.*

[b-NIST 800-57-3] NIST 800-57-3 (2009), *Recommendation for Key Management , Part 3: Application-Specific Key Management Guidance.*

[b-NIST 800-144] NIST 800-144 (2011), *Guidelines on Security and Privacy in Public Cloud Computing*.

[b-NISTIR 7628-1] NISTIR 7628 (2010), *Guidelines for Smart Grid Cyber Security: Vol. 1, Smart Grid Cyber Security Strategy, Architecture, and High-Level Requirements*.

http://csrc.nist.gov/publications/nistir/ir7628/nistir-7628_vol1.pdf

[b-NISTIR 7628-2] NISTIR 7628 (2010), *Guidelines for Smart Grid Cyber Security: Vol. 2, Privacy and the Smart Grid*.

[b-NISTIR 7628-3] NISTIR 7628 (2010), *Guidelines for Smart Grid Cyber Security: Vol. 3, Supportive Analyses and References*.

[b-NIST PKITS] PKITS Version 1.01 (2011), *Public Key Interoperability Test Suite (PKITS), Certification Path Validation*.

[b-OASIS KMIP] OASIS standard (2013), *Key Management Interoperability Protocol Specification Version 1.1*.

[b-OASIS KMIP prof] OASIS standard (2013),*Key Management Interoperability Protocol Profiles Version 1.1*.

[b-PKCS#11] PKCS #11: *Cryptographic Token Interface Standard*.

[b-RCA NMS-301] *PKI Reborn in the Cloud*.

[b-W3C XAdES] W3C XAdES (2003), *XML Advanced Electronic Signatures (XAdES)*.

[b-W3C XKMS] W3C XKMS (2005), *XML Key Management Specification (XKMS 2.0)*.

[b-WAP WTLS] *Wireless Application Protocol WAP-261-WTLS-20010406-a, Wireless Transport Layer Security*, Version 06-Apr-2001.