INTERNATIONAL STANDARD

ISO 20828

First edition
2006-07-01

Road vehicles — Security certificate management

Véhicules routiers — Gestion des certificats de sécurité

Reference number
ISO 20828:2006(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 20828 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

Introduction

Often data transmitted within road vehicles, between road vehicles or from and to road vehicles have to be protected to guarantee their confidentiality and integrity. Cryptography provides excellent means for this kind of protection. Depending on the protection requirements, different schemes may be used. In some situations it is sufficient to lock a data link involving a specific device, and to unlock it only if a second device has sent the correct key in response to an arbitrary seed. The corresponding security access service is specified in various International Standards and is widely used today.

ISO 15764 defines an extended security scheme. It does not just restrict the access to data, but protects the data when transmitted over the data link. Protection is provided against masquerade, replay, eavesdropping, manipulation and repudiation. Before starting the secured data transmission, the data link must be established as a secured link. ISO 15764 provides two methods for this:

a) Both devices participating in the data transmission have a pre-established secret cryptographic key. This key is used to establish the secured link and excludes all third parties not having access to it from participating in the secured link. This method is based on symmetric keys and is applicable to devices with a limited processing power and memory.

b) The secured link may be established between arbitrary devices, if these devices have a private key and a security certificate for the corresponding public key. This method involves asymmetric cryptography requiring a higher amount of processing power and memory at the devices.

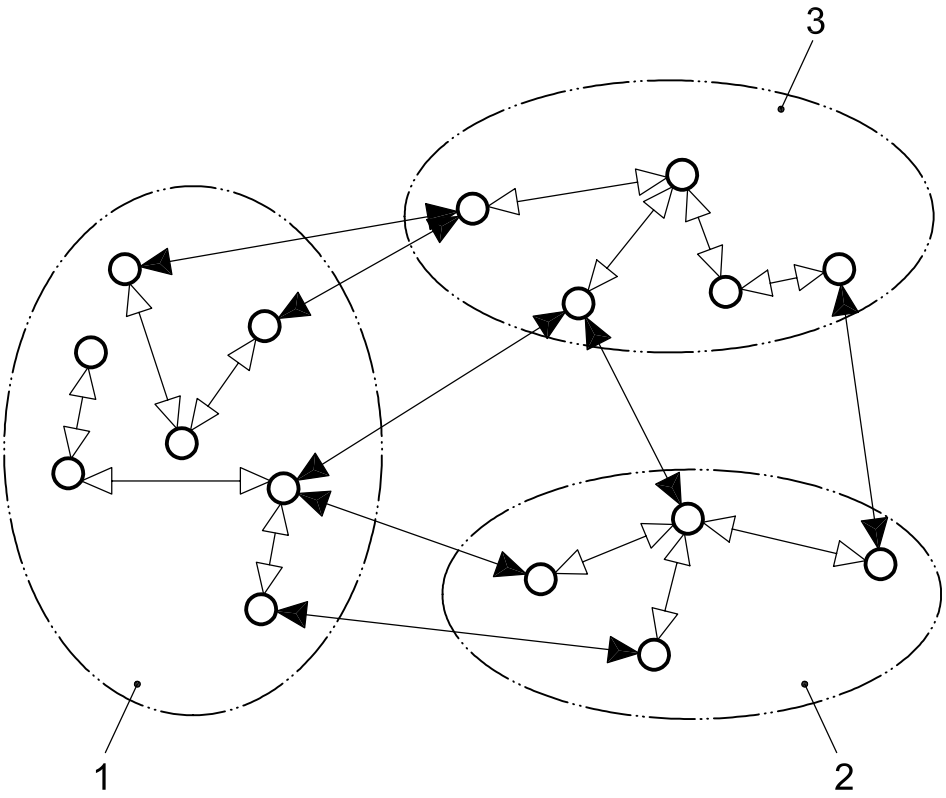
Public keys are cryptographic keys that are publicly available and are linked to a private key, which is kept secret by the device owning it. There are two ways of using a public/private key pair:

a) The device owning the private key may add an electronic signature to data it sends out. This signature is specific for the data sent out and may only be generated with the private key. Both a different data string to be signed and a different private key would lead to a different signature. Any other device possessing the corresponding public key is able to verify the signature and therefore to confirm that the data string originates from the device owning the private key and has not been altered after being sent out.

b) Any device possessing the public key may use it to encrypt data before sending it to the device owning the private key. As the data can only be decrypted with the aid of the private key, no other device is able to correctly interpret the data sent out.

But how does the user of the public key know that it uses the correct one? A malicious third party could send its own public key, pretending it is from a trusted device, and could hope to get access to the secured data transmissions. For each domain of secured data transmissions, there must be an authority (or several of them) deciding which devices can be trusted. This is called Certification Authority. For the trusted devices, it issues security certificates, confirming that the public key is from that device (meaning that the device owns the corresponding private key). The electronic signature of the Certification Authority is attached to the certificate, rendering it unforgeable. As part of the procedure to set up a secured link, the devices involved verify the certificates of each other.

With the second method specified in ISO 15764, a secured link can be established between devices using the public key of the Certification Authority of each other. But in many cases there are different security domains with different authorities responsible to establish trusted devices, and secured links must be established between devices of different domains, not knowing the public keys of the Certification Authorities of the other domain. This International Standard specifies how trust between devices from different security domains is established based on security certificates. In this sense it extends the application range of ISO 15764.



Key

1 security domain 1

2 security domain 2

3 security domain 3

internal secured links covered by ISO 15764

external secured links covered by ISO 20828

Figure 1 — How ISO 20828 extends the application range of ISO 15764

The focus of this International Standard is on the management of certificates. Various security domains based on certificates have already been defined in various contexts. The task of a security certificate management for road vehicles is to give a framework in which such security domains can interact in the sense that secured links can be established from one domain to the other. For instance, there may be specific security domains for different car manufacturers, for public authorities in charge of tachographs or other legislated vehicle components, for telematics service providers, authorized dealers and workshops, emergency task forces and fleet operators. The framework should cover all of them.

When defining this security framework, the following specific requirements of the road vehicle environment have been considered:

— There should be no need for an overall infrastructure to be shared by all security systems. For instance, it can't be expected that shared databases are installed to which the devices involved have access.

— It should be possible to easily integrate existing security systems in the various domains without major modifications.

— The additional security framework should not affect the security of each domain.

— Devices with different security levels are considered. Breaking the security of a device with little protection should not affect the security of other devices.

— It should be possible to use the framework even for devices with limited resources. This means that the provisions requested from the framework should be easy to handle.

The special situation of mobile devices with limited and non-permanent access to communication facilities are considered.

Road vehicles — Security certificate management

1 Scope

This International Standard establishes a uniform practice for the issuing and management of security certificates for use in Public Key Infrastructure applications. Assuming that all entities, intending to set up a secure data exchange to other entities based on private and public keys, are able to provide their own certificate, the certificate management scheme guarantees that the entities will get all additional information needed to establish trust to other entities, from a single source in a simple and unified format. The certificate management is flexible with respect to the relations between Certification Authorities, not requesting any hierarchical structure. It does not prescribe centralized directories or the like, being accessible by all entities involved. With these properties, the management scheme is optimized for applications in the automotive domain.

This International Standard details the role and responsibilities of the Certification Authority relating to certificate issuing and distribution. It specifies how to handle certificate validity and certificate policies. This is the prerequisite for each entity to make sure it can actually trust another entity when intending to exchange data of a specific kind with it.

This International Standard prescribes a Certificate format, which is a special implementation of the well-known X.509 certificate according to ISO/IEC 9594-8. It specifies the structure and use of every certificate component such that it complies with the certificate management established.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 3779, *Road vehicles — Vehicle identification number (VIN) — Content and structure*

ISO 3780, *Road vehicles — World manufacturer identifier (WMI) code*

ISO/IEC 8824-1, *Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation*

ISO/IEC 8824-2, *Information technology — Abstract Syntax Notation One (ASN.1) — Part 2: Information object specification*

ISO/IEC 8824-3, *Information technology — Abstract Syntax Notation One (ASN.1) — Part 3: Constraint specification*

ISO/IEC 9594-2, *Information technology — Open Systems Interconnection — Part 2:The Directory: Models*

ISO/IEC 9594-8, *Information technology — Open Systems Interconnection — Part 8: The Directory: Public-key and attribute certificate frameworks*

ISO/IEC 15408-3, *Information technology — Security techniques — Evaluation criteria for IT security — Part 3: Security assurance requirements*

ISO 15764, *Road vehicles — Extended data link security*

IETF RFC 3279, *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, R. Housley, W. Polk, W. Ford, D. Solo, April 2002*

IETF RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, W. Polk, R. Housley, L. Bassham, April 2002*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 9594-8, in ISO 15764 and the following apply.

3.1
certificate
public-key certificate as defined in ISO/IEC 9594-8, including further information as specified in this International Standard

3.2
certificate validity
assignment of one of the two states “valid” or “invalid” to a certificate by its issuer, which only guarantees that the certificate can be used to establish trust between end entities if it is valid

3.3
Certification Authority List
CAL
list maintained by a CA for one of its public keys, the corresponding private key being used to sign certificates, containing information on other CA having issued CA-certificates with this public key being the public key of the subject, and information on these CA-certificates

3.4
certification path
ordered sequence of different CAs, together with their public keys and CA-certificates issued by them and signed with the corresponding private key, in which each public key of the subject in one of these CA-certificates is the public key of the next CA in the sequence

3.5
Certification Path Information (CPI)
information maintained by a CA for one of its public keys, the corresponding private key being used to sign certificates including information on all certification paths starting at that CA with a CA-certificate being signed by that private key, as well as validity information on the CA-certificates in the certification paths and on the certificates issued for end entities by one of the CA in the certification paths and being signed with the private key corresponding to its public key

3.6
confirmation of trust
information accessible without restrictions and allowing an entity to verify that it can trust another entity

3.7
end entity
entity involved in the establishment of a secure data exchange and not installed at a CA

3.8
entity
technical equipment, protected against access by third parties, that is capable to exchange data on a communication link to which third parties may get access

EXAMPLE 1 A vehicle has a number of ECU (Electronic Control Unit) connected by an internal communication network. Through a gateway, this communication network is connected to a mobile external communication link. The vehicle manufacturer may protect the internal communication network against access by third parties. Then data security on the external communication link may be maintained by the gateway, leaving the data exchanged on the internal network without further protection. In this case, the whole vehicle, represented by the gateway, may be considered as one entity.

EXAMPLE 2 In the vehicle as described, there may be an ECU that needs to exchange sensitive data with an external device using the internal communication network, gateway and external communication link. For these data, the protection level on the internal communication network may be considered as too low. The ECU will then be considered as a separate entity maintaining data security on a higher level than would be if it would rely on the gateway.

3.9
initial trust
trust of an entity in another entity, which is based on direct knowledge about this other entity and not on information received from third parties

3.10
issuer (of a certificate)
entity identified in a certificate that has signed the certificate

NOTE According to this International Standard, the issuer of a certificate is always a CA.

3.11
path certificate
the public key of a CA, together with additional information according to this International Standard, rendered unforgeable by signing it with the private key of another CA, confirming the existence of certification paths to the public key holder

3.12
subject (of a certificate)
the entity identified in a certificate being the holder of the public key

4 Symbols and abbreviated terms

For the purpose of this International Standard the following abbreviations apply:

CA Certification Authority

CAL Certification Authority List

CPI Certification Path Information

DER Distinguished Encoding Rules

ECU Electronic Control Unit

VIN Vehicle Identification Number

WMI World Manufacturer Identifier

5 Certificate Management Principles

5.1 Establishment of trust

5.1.1 Security-Related classes

Entities intending to exchange sensitive data shall exchange these data only after having established trust between them. This International Standard specifies how this trust is established.

Each entity may take appropriate measures to protect sensitive data it handles, and will trust in these measures. As soon as sensitive data are exchanged between entities, this is not sufficient; it is necessary that the other entity gets involved in the protection and guarantees that it takes appropriate security measures to protect the data. The establishment of trust means that the trusting entity is convinced that the other entity takes appropriate security measures on sensitive data.

To specify the trust needed, data to be exchanged between two entities shall be classified according to the following four security-related classes:

Class 0: No protection

The data exchanged are not sensitive, such that security threats may be excluded. Such data may be exchanged without any trust being established.

Class 1: Confidentiality-related protection

There is potential misuse of sensitive data by third parties. The sender of the data is responsible for maintaining confidentiality. The sender shall:

— Only send such data to receivers being entitled to have access to them.

— Only send such data to receivers keeping the confidentiality. This includes the property that they don't forward data marked as being prohibited to forward, and that they only forward such data to receivers subject to the same confidentiality rules.

— When sending the data, protect them such that no non-authorized third party can guess the content of the data or their meaning.

To act according to these rules, the sender shall establish trust in the receiver before sending the data. An entity that is not trusted is not entitled to have access to class 1 data. The trust must extend to the fact that this entity keeps class 1 data confidential.

Class 2: Integrity-related protection

False data could potentially cause adverse effects at the receiver of the sensitive data, when using them. The receiver of the data is responsible for verifying data integrity. The receiver shall:

— Only use such data when received from a sender entitled to send them.

— Only use such data when they originate from the sender or from an entity subject to the same integrity rules as the sender (including integrity protection on the way from that entity to the sender).

— Use such data only after having verified that the integrity is maintained from the sender to the receiver. Integrity means that the data are sent from the entity that claimed to be their sender, have been generated under appropriate circumstances for the intended use and were not manipulated on their way to the receiver.

To act according to these rules, the receiver shall establish trust in the sender before accepting his data. An entity that is not trusted is not entitled to send class 2 data. The trust must extend to the fact that this entity only sends out class 2 data that originate from the entity or another trusted entity.

Class 3: Confidentiality- and integrity-related protection

Combination of class 1 and 2. Sender and receiver of the data shall apply class 1 and class 2 rules. Both entities shall be able to trust each other.

5.1.2 Extension of trust

For the exchange of data of class 1, 2 and 3, trust shall be established between the two entities. For this, the method specified in this International Standard assumes that trust has the following fundamental properties:

— It is oriented: If entity A trusts entity B, then this doesn't mean automatically that entity B trusts entity A. Therefore, the method to establish trust shall be applied in the direction trust is needed according to the data classification.

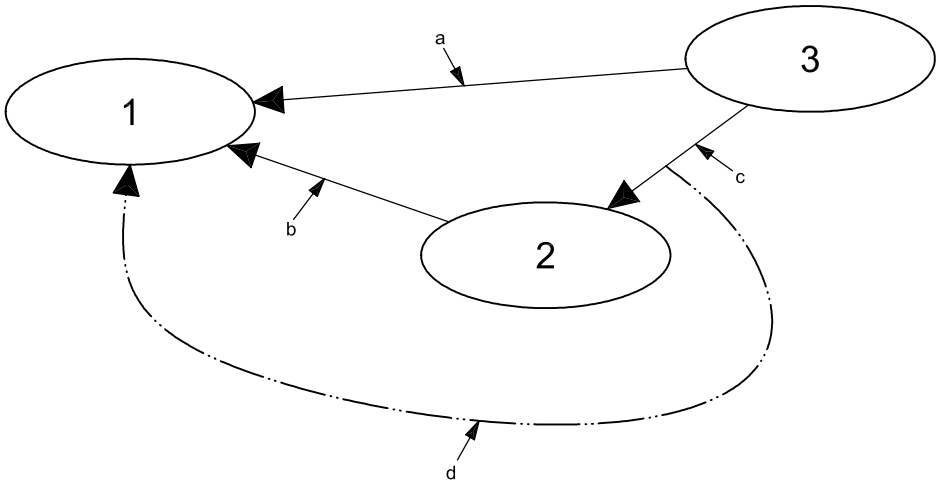
— It is transitive: If entity A trusts entity B and entity B trusts entity C, then it follows that entity A can trust entity C.

The second property allows extending existing trust between entities, to other entities. This extension only works under the following conditions:

— As a starting point, there is some trust that has not been established by such an extension process. This trust shall be called initial trust.

— The confirmation about the trust being established from entity B to entity C is available to any entity A intending to extend its trust in entity B to entity C. Entity A shall be able to verify this confirmation.

An entity A shall only extend trust it has in entity B to entity C if it gets the confirmation about B trusting C and after having successfully verified this confirmation. This situation is shown in Figure 2.



Key

1 entity A
2 entity B
3 entity C

a Extension of trust.

b Established trust to be extended.

c Established trust.

d Confirmation of established trust needed for the extension of trust.

Figure 2 — Confirmation needed for the extension of trust

In the extension of trust, the information flow needed is always in the direction opposite to the direction of trust. If trust of entity A in B and of B in C is to be extended, the information shall be transmitted from entity B to A.

If two entities need to exchange class 1, 2 or 3 data and could not establish initial trust in the direction needed, they shall establish trust before exchanging the data, with a step-by-step extension of already established (initial) trust. This extension involves a sequence of intermediate entities, each of them having established initial trust with the next one in the sequence.

In case mutual trust is requested, the intermediate entities to establish trust need not be the same for both directions.

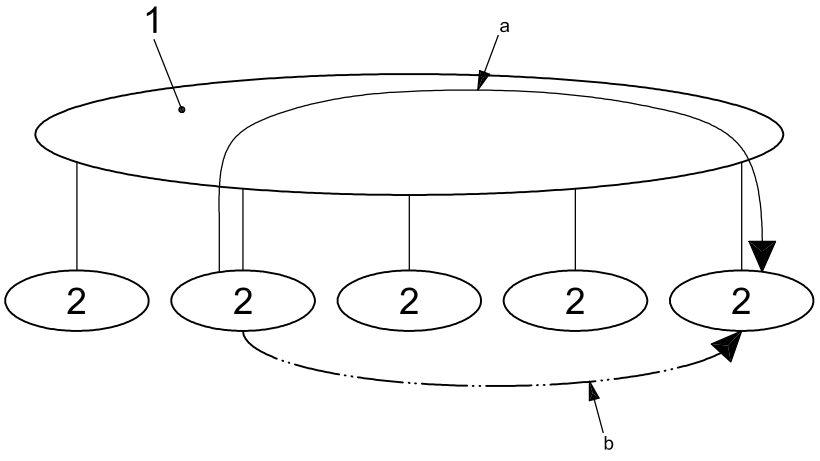
The extension process may start at an arbitrary point in the sequence of entities. No restriction is posed on the timely sequence of the extension process.

5.1.3 Trust Establishment Principles

All intermediate entities used to establish trust shall be of a special type, called Certification Authority (CA). Their only task shall be to support the establishment of trust and they shall not be involved in the exchange of sensitive data, except those related to their task. To be distinguished from CA, the entities establishing trust between each other to exchange class 1, 2 or 3 data shall be called end entities.

The number of CA to be set up is expected to be small as compared to the end entities, and each CA is part of the trust-establishing sequence for a large number of end entities. The CA may be installed in a favourable environment and high security standards may be applied, whereas the end entities in some cases are operated under adverse conditions related to security. The advantage of the concept, to establish trust via CA and not via end entities, is that breaking the security of an end entity only affects the trust to this one end entity. The CA, being involved in the established trust between many different entities, may be much better protected.

Between end entities, this International Standard considers only the establishment of trust based on an extension of trust via CA, as shown in Figure 3. Although not considered, the establishment of initial trust between end entities, based on methods that are outside the scope of this International Standard, is not excluded.



Key

1 Certification Authorities

2 End Entity

a Path to establish trust.

b Trust to be established.

Figure 3 — Roles of Certification Authorities (CAs) and end entities in the establishment of trust

As initial trust between end entities is excluded, it can only be established from an end entity to a CA, from a CA to another CA, as well as from a CA to an end entity. Therefore, for the method of establishing trust to work, each end entity that needs to trust a different end entity shall establish initial trust to at least one CA, and for each end entity that has to be trusted by a different end entity at least one initial trust from a CA to it shall be established. Initial trust between end entities and CA shall always be established such that it is mutual. A CA, having established mutual trust with an end entity, is said to be the CA of that end entity and the end entity is said to be an end entity of that CA.

NOTE 1 Typically, a CA is the CA of many end entities.

NOTE 2 Between CAs the trust established is not automatically assumed to be mutual. Trust to a CA may be established from several other CAs.

EXAMPLE A vehicle manufacturer may install a CA for his vehicles. The end entities of that CA will then be a central ECU in each vehicle that controls the data exchange with exterior devices. The CA of the vehicle manufacturer will trust the vehicles and the vehicles will trust the CA of the vehicle manufacturer. A company operating a repair shop network may install a CA for these repair shops. The end entities of this CA will be some tool at each repair shop to be connected to vehicles. The repair shop CA will trust the tools and the tools will trust the repair shop CA. Now the vehicle manufacturer CA may install initial trust to the repair shop CA, thus accepting that the repair shops of that CA may repair its vehicles. Assuming that a tool of a repair shop is connected to a vehicle of the vehicle manufacturer and sends some calibration data to that vehicle, being classified by the vehicle as security related class 2 (meaning that integrity related protection is needed). To accept these data, the vehicle needs to trust the tool. Knowing that it can trust its CA, the CA trusts the repair shop CA and the repair shop CA trusts the tool, it will accept the calibration data. But calibration data from a different tool, where the vehicle manufacturer CA has not established trust to the corresponding repair shop CA, will be refused.

5.2 Certificates

This clause gives a brief introduction on how to establish initial trust and how to verify that an entity has initial trust in a second entity. The principles listed here are well known in cryptography and are the subject of many standards. More details are given in ISO/IEC 9594-8 and IETF RFC 3280.

Entities to which trust must be established shall have a specific property on which this trust can be based. For the trust to be verified, they shall be able to prove that they have this property. Moreover, other entities not trusted must be excluded from having the same property. Initial trust shall relate to this specific property, thus excluding other entities from being erroneously trusted.

This property shall be a private key owned by the entity. Each entity shall have a different private key. The entity owning it shall keep it secret. The proof of owning the private key is performed with the corresponding public key. As the private key can't practically be derived from the public key, this public key can be distributed to other entities without the risk that they could get the private key and therefore would inherit the property that is the foundation of trust.

NOTE 1 An entity having access to the public key can verify that the other entity owns the corresponding private key, for instance by performing a strong authentication procedure as described in ISO/IEC 9594-8.

Let us assume trust has been established between an entity A and an entity B. This means that entity A has the public key of entity B, has verified that entity B owns the corresponding private key and has found that entity B has all other properties needed to trust it. Let us further assume that entity B was able to establish initial trust to entity C. In other words, it has the public key of entity C, has verified that entity C owns the corresponding private key, and trusts entity C and therefore its public key. If entity A wants to extend the trust it has in entity B to entity C, the confirmation of trust needed from entity B must link the public key of entity B (which it trusts already) to the public key of entity C (to which it would like to establish trust). This confirmation is called a certificate. The certified entity, in this case entity C, is called the subject of the certificate. The entity generating the certificate, in this case entity B, is called the issuer of the certificate.

The certificate is a set of data that may be exchanged between entities without any restriction. Therefore, entity A may receive the certificate issued by entity B for entity C and shall use it to verify the initial trust of entity B in entity C.

The certificate establishes the link between the public key of entity B and C, and by this serves as a confirmation of the initial trust of entity B in entity C, in the following manner: It contains as one data element the public key of entity C. It includes as a second data element the signature, generated by entity B using its private key, and extending (among other data elements) on the public key of entity C. Each certificate shall include only one public key of its subject.

NOTE 2 A (digital) signature is a code that is generated for arbitrary data — the data to be signed — with the aid of a private key, and that is usually added to these data. It may be verified based on the signature itself, the data to be signed and a public key. The verification is successful if the data to be signed have not been altered and if the public key is the one linked to the private key used for the signature. In all other cases, there is an extremely high probability for the verification to fail. Therefore, the signature renders the data to be signed unforgeable and proves that it originates from the entity owning the private key linked to the public key used to verify it.

To validate the certificate, entity A verifies the signature using the public key of entity B. If the signature is correct, entity A knows only entity B could have generated it, as only entity B has the corresponding private key linked to the public key used to verify it. Moreover, it knows that the public key of entity C contained in the certificate is correct, as if this would not be the case, the signature would fail to verify.

Successful validation of a certificate, using the public key of the issuer (to verify the signature contained in the certificate), shall extend the trust from the issuer to the subject of the certificate. The protection of data of class 1, 2 or 3 shall be based on the public key contained in the certificate and on the private key linked to it.

The specification of how to use the private and public keys to protect class 1, 2 and 3 data is outside the scope of this International Standard. Nevertheless, the following indicative examples may enlighten the relation between security-related class, trust to be established and public and private key:

— Class 1: If confidentiality-related protection is requested according to class 1, then as mentioned in 5.1 the trust shall be established from the sender of the data to the receiver. After having established trust in the corresponding direction, the sender will know and trust the public key of the receiver, contained in the receiver's certificate. He may use this key to encrypt the data to be protected. Even when sending the data on an open communication channel, no one will be able to interpret the encrypted data, except the receiver who exclusively has access to the corresponding private key and may use it to decrypt the data.

— Class 2: If integrity-related protection is requested according to class 2, then as mentioned in section 5.1 the trust shall be established from the receiver of the data to the sender. After having established trust in the corresponding direction, the receiver will know and trust the public key of the sender, contained in the sender's certificate. The sender may use the corresponding private key to sign the data to be protected. When receiving the data, the receiver may verify the signature using the public key contained in the sender's certificate to make sure the data was sent by the trusted subject of this certificate and have not been altered on their way from there.

— Class 3: If both confidentiality and integrity related protection is requested according to class 3, then as mentioned in section 5.1 mutual trust between the sender and the receiver of the data has to be established. After having established this trust, both the sender and the receiver will know and trust the public key of each other. The sender may use his private key to sign the data and the public key of the receiver to encrypt it. The receiver will use his private key to decrypt the data and the public key of the sender to verify the signature.

The procedures described above may, in some situations, not be optimized with respect to the processing effort needed at the entities involved, and the amount of data to be transferred on the communication link. More elaborate procedures may be found for instance in ISO 15764.

5.3 Certification authorities

It is the task of the CA to support the end entities in establishing trust between them. For this the CA issues certificates. Only a CA shall be allowed to issue certificates.

Each CA shall issue certificates for each of its end entities. Moreover, each CA is allowed to issue CA-certificates for any other CA it trusts. A CA may issue different certificates with different content for the same subject, if appropriate.

NOTE 1 It is possible to set up CA not having end entities and only issuing certificates for other CA. In some cases this facilitates the management of certificates.

By issuing the certificate, the CA shall confirm that the subject of the certificate:

a) owns the private key corresponding to the public key listed in the certificate;

b) has taken appropriate measures to keep this private key secret. If the certificate states a security level (see 5.5), then the private key is required to be kept secret according to this security level;

c) has taken appropriate measures to handle data of class 1, 2 or 3 according to the requirements outlined in this International Standard, if this subject is an end entity;

d) has taken appropriate measure to issue certificates according to the requirements outlined in this International Standard, if this subject is a CA.

NOTE 2 The methods used by the issuer to evaluate the subject of the certificate and to get evidence that what is confirmed actually holds true are outside the scope of this International Standard.

As establishing trust from an end entity to a second end entity goes via some CA, the end entity shall have at least one CA it trusts. Each end entity shall trust its own CA (which is the CA that issued a certificate for it). As the end entity is not allowed to issue certificates, this trust has no certificate as corroboration.

NOTE 3 This is no disadvantage, as no other entity will use the initial trust of the end entity in its CA for the extension of its own trust.

For an end entity, its CA shall serve as the starting point to establish trust. Therefore, it must know the public keys of its CA. In addition to the points mentioned, a CA, when issuing an end entity certificate, confirms that the subject has the public keys of the issuer and has taken appropriate measures such that it only uses these public keys to establish trust.

NOTE 4 If such measures were not taken, a malicious third party could implement its own public key at the end entity and could sign with the corresponding private key a certificate, the subject of it being a fake end entity. This fake end entity could then send the certificate to the affected end entity and based on it would be trusted.

When a CA has issued an end entity certificate, it shall send the certificate to that end entity, such that the end entity can provide it to other end entities when setting up a secured data link.

Any end entity shall keep its own certificates (which are the certificates its CA issued for it) after receiving them from its CA, to the end of the time interval of initial validity (see 5.4.1), and provide them to other end entities on request.

NOTE 5 In case both end entities have the same CA, this is sufficient to establish trust between them.

If the two end entities have different CAs, then they need additional information about the trust between CAs, to establish trust between each other. The end entities do not have to keep this information permanently. Their CAs shall keep it and shall provide it on request (see 5.6.2).

End entities are allowed to keep information needed to establish trust. But before using it, they should make sure it is still valid (see 5.4).

The CA may assign the task to a different entity of keeping information about the trust between CA and of distributing it to the end entities. Still, it is essential that the information originate from the CA and the end entities are able to get all information they need, to establish trust to a second end entity (except the relevant end entity certificate) from one source.

NOTE 6 To get information from its own CA while setting up a data link to a second end entity, it is necessary that the end entity can switch between different communication partners. For some end entities, especially mobile ones, this might be complicated or even impossible. In such a case, it is recommended that they request the information needed from their CA via the second end entity and that the second end entity provides the information to the first one. This means that a secured link can only be set up if the entity that needs to trust a second entity either is able to contact its CA (or a third party providing the information instead of the CA) during the set-up procedure, or the second end entity is able to establish the contact and to forward the information needed.

This International Standard excludes many options in the relation between end entities and CA that are common elsewhere. According to this International Standard, an end entity may not have several CAs, it may not have several trust anchors, and the trust anchor must be its CA. The reason for this is that all other options would lead to additional ambiguities and therefore require more complicated specifications.

For instance, if an end entity has several CAs, what happens if one of them loses trust in the end entity? Some other entities might have established trust via this CA and therefore also cease to trust the end entity, whereas others have established trust via a different CA and still trust it. This would need to be the foundation of a special relation between CAs having issued a certificate for the same end entity.

Also, if an end entity has a trust anchor distinct from its CA, a loss of trust of the CA in the trust anchor would have to cause the loss of trust in the end entity (because the end entity establishes trust via the trust anchor and with a trust anchor not being trustworthy could accept fake data from other end entities or send confidential data to end entities not to be trusted). This would mean that the CA would, as an additional task, have to monitor the trust anchor.

5.4 Certificate validity

5.4.1 Time interval of initial validity

Each certificate may have two states, valid or invalid. At any given time, it has exactly one of these states. The CA issuing a certificate shall be responsible for assigning its validity state. Whenever the certificate is valid, the CA having issued it guarantees that the public key of the certificate subject can be trusted.

Each certificate includes a time interval of initial validity consisting of a start date and an expiry date. The certificate shall be invalid before the start date and after the expiry date. The start date of the time interval shall not be before the date the corresponding certificate is issued.

NOTE 1 If several CAs issue a certificate for the same entity and the same public key, it is recommended that they agree on a common expiry date of the time interval of initial validity.

Each CA shall fix an envisaged validity time interval for its own key pair used to sign certificates. It shall chose the time intervals of initial validity for the certificates it signs with the private key such that it does not exceed the envisaged validity time interval of the key pair.

Instead of issuing a certificate with a time interval of initial validity exceeding the envisaged validity time interval of the CA's key pair, the CA may generate a new key pair with an expiry date after the envisaged expiry date of the certificate, and use its private key to generate the certificate. It might not be practical to issue more than one certificate for the same public key of an end entity, as these certificates have to be forwarded to their subject (see 5.3). To avoid this, the envisaged validity time interval of the CA's key pair shall exceed the intended time interval of initial validity of the end entity certificate.

Together with its public key, a CA shall forward the information on the envisaged validity time interval to all other CA intending to issue a CA-certificate for it. The time interval of initial validity for these CA-certificates shall not exceed the envisaged validity interval. For this to be excluded, the transfer of the information on the envisaged validity time interval needs integrity related protection.

If a CA issues a new certificate for the same end entity and the same public key, the time interval of initial validity stated in this new certificate shall not exceed the time interval of initial validity stated in the old one.

NOTE 2 A new certificate for the same end entity and public key may for instance be issued to change the certificate policy (see 5.5) or in case a new private/public key pair is implemented at the issuer of the initial certificate. In the latter case, there is no need to evaluate the subject of the certificate again.

As described in 5.3, the trust of an end entity in its CA has no certificate as corroboration. When receiving the public keys of its CA to be used to establish trust, the end entity shall also receive the time intervals of initial validity for these public keys. The end entity shall not establish trust based on it outside this time interval.

NOTE 3 This time interval may be distinct from the time interval of initial validity in the certificate issued by the CA for the end entity.

To make sure an entity establishing trust doesn't use invalid certificates or, in case of an end entity, an invalid public key of its CA, this entity needs an indication of the actual time, to be compared to the start and expiry date. This indication is critical and shall be included in the scope of the protection measures for the entity.

NOTE 4 The way date and time is made available at the entity is outside the scope of this International Standard.

5.4.2 Revocation and re-establishment

Within the time interval of initial validity, the issuer shall revoke a certificate, if one of the properties of the subject confirmed by the certificate (see 5.3) is no longer valid. Revocation renders the certificate invalid. Only the issuer is entitled to revoke a certificate. All other CAs shall inform the issuer on all observations that might justify the revocation of the certificate. In addition to the points mentioned in 5.3, a CA issuing certificates shall guarantee that it takes all appropriate measures to investigate on possible reasons for revocation of these certificates.

NOTE 1 As in the case of a CA-certificate being revoked, there may be other CA-certificates for the same subject and the same public key, CAs should exchange information on revocation to allow other CAs to revoke their CA-certificate as well.

Revocation affects the public key of the subject of the certificate. As the subject may have more than one valid public key, the issuer shall determine if other public keys are affected as well and if the corresponding certificates have to be revoked.

To be able to identify the certificate to be revoked, the CA shall maintain a list of the certificates it issued, at least including the serial number (see 7.2), the time interval of initial validity, an identifier for the subject and for the public key of the subject as well as the validity state.

If the issuing CA finds that the reasons for the revocation are no longer justified, it may re-establish the revoked certificate. A re-established certificate changes its state to valid. Only the issuer is entitled to re-establish a certificate.

For the exchange of revocation and re-establishment information between CAs, this International Standard applies a push-principle. A CA revoking or re-establishing a certificate, or receiving revocation or re-establishment information from another CA, shall forward the corresponding information immediately to all other CAs that might be affected. Details on the procedures are specified in 5.6.1.

For the distribution of information on certificate validity from CA to end entities, a pull-principle is applied. Each CA will provide this information to its end entities on request. The distribution of the revocation and re-establishment information among CA guarantees that the validity information provided to end-entities is up-to-date and complete.

Certificate validity information is part of the information on trust between CAs and, as mentioned in 5.3, the CAs may assign its distribution to a different entity.

An end entity shall not use certificates known to be invalid for the establishment of trust.

To use a certificate to establish trust, the validity state of which is unknown, involves a certain risk. The entity using it shall decide, depending on the given circumstances, if this risk can be taken. If not, the validity state shall be requested from the CA before establishing trust. Rules on when an end entity must check certificate validity before using a certificate may be included in the certificate policy for that end entity (see 5.5).

For the use of the public keys of the CA, stored at its end entities and serving as starting point to establish trust, there is no revocation procedure. Instead, the CA shall implement a procedure to block any of these public keys as soon as possible after a reason has been found not to trust their use. The end entity shall not use a public key of its CA that has been blocked. The CA and end entity shall take appropriate measures to protect the implementation of the blocking procedures.

NOTE 2 The procedure to block a key at an end entity is outside the scope of this International Standard. One way to implement it would be that the CA sends a blocking message to the end entity and would sign it using a private key with the corresponding public key being installed at the end entity, and being different from the public key to be blocked.

It is possible to implement a procedure to de-block public keys at end entities for the same reasons as the re-establishment, with analogous procedures as those used to block the key. Again, it is necessary to implement protection measures for this procedure.

5.4.3 Key renewal

For some entities, it is possible to implement new public/private key pairs during their lifetimes. This allows an entity to contribute to the establishment of trust after its public key became invalid. Each CA shall be able to execute such a key renewal.

NOTE 1 This International Standard does not specify where the key pair is generated and how the keys are forwarded to the appropriate entities. In any case, it is the responsibility of the CA issuing a certificate for the new public key to confirm that all properties according to 5.3 are maintained.

In case of a new key pair of an end entity, it is the responsibility of its CA to issue a new certificate for the new public key, if appropriate, and to send it to the subject of the certificate (see 5.3). This completes the renewal procedure.

In case of a new key pair of a CA, this CA shall specify the envisaged validity time interval according to 5.4.1 and assign a key identifier.

NOTE 2 A CA may use different key pairs towards its end entities than towards other CA. This means it possesses one key pair the public key of which is implemented at its end entities and used by them to establish trust [and the private key is used to sign path certificates to be sent to the end entities (see 5.6.2)], and a different key pair the private key of which is used to sign the certificates (and the public key is contained in the CA-certificates issued for that CA).

A CA having implemented a new key pair, the public key of which is intended to be used at the CA's end entities for the establishment of trust, shall implement the new public key at its end entities, together with the appropriate time interval of initial validity.

5.5 Certificate policies

5.5.1 Structure of certificate policies

So far, trust has been considered as a property only depending on entities; an entity trusts another entity or it does not. But trust may depend on the data to be exchanged. For some data to be exchanged, trust may be there, for other data it may not. A certificate policy is a certificate attribute expressing restrictions of the trust with respect to the data to be exchanged.

ISO/IEC 9594-8 provides a general framework for the implementation and use of certificate policies. This International Standard defines certificate policies being compliant to the framework, but with more structure and handling rules. The aim is to allow automated processing of policies and to automatically derive policies for certification paths (see 5.6).

The certificate policy shall be included in the part of the certificate containing the data to be signed. The trust of an end entity to its CA shall not be restricted by a certificate policy.

A CA issuing a certificate shall include in this certificate the appropriate certificate policy. This certificate policy applies to all data to be exchanged between end entities based on trust established with the help of the certificate.

To note the certificate policy, predefined data categories shall be used. For data of class 1, 2 or 3 to be exchanged, the CA shall investigate in which of the predefined data categories they are contained. The trust to be established in order to exchange class 1, 2 and 3 data (see 5.1.1) shall only be deemed as being established if the data is compliant to the certificate policy of all certificates used to establish the trust.

NOTE 1 With these settings, it is clear for the situation of two end entities with the same CA, which data they may exchange; all class 0 data and those class 1, 2 or 3 data that are compliant to the certificate of the end entity to which trust must be established, depending on which class the data are (see 5.1.1). For end entities with different CA, the trust to be established involves CA-certificates. In many cases, the set of certificates, that may be used to establish trust, is not unique. The rules on what data may be exchanged in such a situation are listed in 5.6.

There are two ways to introduce predefined data categories:

a) Standardized data categories. These are specified in this International Standard (see 5.5.2).

b) User defined data categories. These are specified by a CA issuing certificates according to this International Standard.

It is recommended that a registration procedure be established for user defined data categories such that categories may be used by different CA and processed by end entities of different CAs. The establishment of such a registration procedure is outside the scope of this International Standard.

Each certificate policy consists of two lists of data categories that may include an arbitrary number of entries: the in-list and the ex-list. Data of class 1, 2 or 3 to be exchanged shall be compliant to the certificate policy, if and only if they are compliant to the in-list and to the ex-list. This compliance is defined as follows:

— In-list: Data are compliant to this list, if and only if they are contained in all of the data categories listed in it. If the in-list list contains no data categories, then all data are compliant.

— Ex-list: Data are compliant to this list, if and only if they are contained in none of the data categories listed in it. If the ex-list list contains no data categories, then all data are compliant.

No data category shall be contained in both the in-list and the ex-list.

NOTE 2 If a data category would be contained in both the in-list and the ex-list, then no data to be exchanged would be compliant to the certificate policy, therefore rendering the certificate useless.

The in-list may contain, instead of simple data categories, unions of data categories. If D_1 to D_n are simple data categories, then their union is denoted by $\cup(D_1, ..., D_n)$. Data are contained in it if and only if they are contained in at least one of the data categories D_1 to D_n.

The ex-list may contain, instead of simple data categories, intersections of data categories. If D_1 to D_n are simple data categories, then their intersection is denoted by $\cap(D_1, ..., D_n)$. Data are contained in it if and only if they are contained in all of the data categories D_1 to D_n.

NOTE 3 The in-list may be considered as the intersection of the data categories contained in it. The ex-list may be considered as the union of the data categories contained in it. Data to be exchanged may be considered as elements of data categories. The usual theorems of set theory apply. If the in-list is the set denoted by I and the ex-list the set denoted by E, then data to be exchanged are compliant to the certificate policy, if they are contained in I\E (\ being the exclusion mark).

A data category D_1 may be contained in a second data category D_2. Then data contained in D_1 shall automatically be contained in D_2. For the standardized data categories, the relevant information on which data category is contained in which other data category is included in this International Standard (see 5.5.2). For the user-defined data categories, the CA specifying it shall list the data categories it is contained in and the data categories that are contained in it. If a data category D_1 is contained in a data category D_2 and D_2 is contained in D_3, then D_1 shall be contained in D_3 even if this is not explicitly listed. Moreover, D_i is contained in $\cup(D_1, ..., D_n)$ for all i between 1 and n, and $\cap(D_1, ..., D_n)$ is contained in D_i for all i between 1 and n.

As new user-defined data categories may be added by the CA and included in certificates at any time, it may happen that end entities are not aware if the data to be exchanged are contained in these data categories, and that the corresponding information is not available to them when the data should be exchanged. Nevertheless, the data may be considered as compliant to the certificate under the following conditions:

— For unknown data categories in the in-list, if the data are contained in a data category that is contained in the unknown data category.

— For unknown data categories in the ex-list, if the data are not contained in a data category the unknown data category is contained in.

5.5.2 Standard data categories

Table 1 lists the standard data categories.

NOTE Care should be taken to include the standard data categories in the appropriate list, as some of them only make sense in the in-lists and some of them only in the ex-list. For instance, the “Detection of physical attack” data category would make no sense in the ex-list.

Table 1 — Standard data categories

Name	ASN.1 value [a]	Description	Contained in
Security class 1	{securityClass, 1}	This data category is specifically for the exchange of data of class 1 according to 5.1.	
Security class 2	{securityClass, 2}	This data category is specifically for the exchange of data of class 2 according to 5.1.	
Security class 3	{securityClass, 3}	This data category is specifically for the exchange of data of class 3 according to 5.1.	Security class 1, Security class 2
Evaluation Assurance Level <*n*> [b]	{eal, <*n*>}[b]	Data complying with this data category are only exchanged between entities being a target of evaluation and having achieved the evaluation assurance level n (n an integer number between 1 and 7) or higher than n according to ISO/IEC 15408-3.	Evaluation Assurance Level <m> with m < n
Path verification level 1	{pathVerification, 1}	Data complying with this data category are only sent (in case of class 1 or 3 data) to an end entity being trusted according to a certification path, or used by the receiver (in case of class 2 or 3 data) if received from an end entity being trusted according to a certification path, if the verification of this certification path being valid is not more than three years old at the time the data is sent or used.	
Path verification level 2	{pathVerification, 2}	Data complying with this data category are only sent (in case of class 1 or 3 data) to an end entity being trusted according to a certification path, or used by the receiver (in case of class 2 or 3 data) if received from an end entity being trusted according to a certification path, if the verification of this certification path being valid is not more than 30 days old at the time the data is sent or used.	Path verification level 1
Path verification level 3	{pathVerification, 3}	Data complying with this data category are only sent (in case of class 1 or 3 data) to an end entity being trusted according to a certification path, or used by the receiver (in case of class 2 or 3 data) if received from an end entity being trusted according to a certification path, if the verification of this certification path being valid is not more than 1 day old at the time the data is sent or used.	Path verification level 1 and 2
Path verification level 4	{pathVerification, 4}	Data complying with this data category are only sent (in case of class 1 or 3 data) to an end entity being trusted according to a certification path, or used by the receiver (in case of class 2 or 3 data) if received from an end entity being trusted according to a certification path, if the verification of this certification path being valid is not more than 30 minutes old at the time the data is sent or used.	Path verification level 1 to 3
Path verification level 5	{pathVerification, 5}	Data complying with this data category are only sent (in case of class 1 or 3 data) to an end entity being trusted according to a certification path, or used by the receiver (in case of class 2 or 3 data) if received from an end entity being trusted according to a certification path, if the verification of this certification path being valid is not more than 1 minute old at the time the data is sent or used.	Path verification level 1 to 4
Entity inspection 1	{entityInspection, 1}	Data complying with this data category are only exchanged between entities that are subject to inspection by authorised personnel at least once every 3 years.	
Entity inspection 2	{entityInspection, 2}	Data complying with this data category are only exchanged between entities that are subject to inspection by authorized personnel at least once per year.	Entity inspection 1
Executable code protection	exCodeProt	Data complying with this data category are only exchanged between entities that allow data only to be used as executable code at their processors if they are declared as being executable code, have a registered originator being liable for the correctness of the code, and after having verified the authenticity and integrity of the data.	Security class 2
Resistance to physical attack	resistPhysAttack	Data complying with this data category are only exchanged between entities protecting all their components storing or processing class 1, 2 or 3 data such that physical attacks to these components may not endanger the purpose of the security class assigned to the data affected by the attack.	

Table 1 (*continued*)

Name	ASN.1 value [a]	Description	Contained in
Detection of physical attack	detectPhysAttack	Data complying with this data category are only exchanged between entities protecting all their components storing or processing class 1, 2 or 3 data such that physical attacks endangering these data can be detected after they occurred.	
Reporting of physical attack	reportPhysAttack	Data complying with this data category are only exchanged between entities protecting all their components storing or processing class 1, 2 or 3 data such that physical attacks endangering these data are automatically reported to authorised personnel.	Detection of physical attack
Limited access	limitedAccess	Data complying with this data category are only exchanged between entities protecting all their components storing or processing class 1, 2 or 3 data such that only authorised personnel has access to these components after authentication.	
Privacy related data	privacy	Data complying with this data category are subject to a defined data privacy policy. Each device exchanging such data guarantees that these are only transferred on communication networks with public access in encrypted form, are only used for the originally intended purpose and are erased or rendered anonymous immediately after use. In case of enforcement they are only kept, if they potentially give evidence to the violation of some regulation.	Security class 1
Personal data	personalData	Data complying with this data category contain information that allows identifying human beings together with some of their properties or behaviour.	
Vehicle location data	vehicleLocation	Data complying with this data category contain information indicating the location of an identifiable vehicle at a specified time.	
In-vehicle data	inVehicle	The exchange of data complying with this data category is limited to the vehicle in which they are generated.	
Software	software	Data complying with this data category include executable code.	Vehicle repair data
Calibration data	calibration	Data complying with this data category, when implemented at an ECU, lead to a specific behaviour of this ECU related to its interaction with other devices.	Vehicle repair data
Vehicle repair data	vehicleRepair	Data complying with this data category are used to repair vehicles.	
User access data	userAccess	Data complying with this data category contain information that allows accessing a device with the rights of a device user.	
Supervisor access data	superAccess	Data complying with this data category contain information that allows accessing a device with the rights of a device supervisor, entitled to make all possible modifications to the device.	User access data
Vehicle access data 1	{vehicleAccess, 1}	Data complying with this data category contain information that allows accessing the passenger area of a vehicle.	
Vehicle access data 2	{vehicleAccess, 2}	Data complying with this data category contain information that allows accessing a vehicle to drive it.	
Vehicle safety data	vehicleSafety	Data complying with this data category are tested not to have adverse effects on the dynamic behaviour of a vehicle when being used within that vehicle.	
...	...	...	

[a] See 7.14.

[b] <*n*> is to be replaced by an integer number between 1 and 7.

5.6 Certificate Paths

5.6.1 Certification path information distribution

The information on the certification path must be available at the CA at which the path starts. For this purpose, each CA shall maintain for each of its key pairs, the private key of which is used to sign certificates, a Certification Authority List (CAL) and Certification Path Information (CPI). The CAL is a list of all CAs that issued a CA-certificate containing the public key of the key pair as the public key of the subject. It shall include information to identify the issuing CA of the CA-certificate, contact information for the data exchange with that CA as well as the initial time interval of validity and an identifier for the public key contained in the CA-certificate. The information on a CA-certificate shall be kept in the CAL until this CA-certificate reaches its expiry date (see 5.4.1).

A CA shall also be included in the CAL in case the CA-certificate it issued for the CAL holder has not reached the start date of validity. It will stay in the CAL if the CA-certificate has been revoked (see 5.4.2).

Each CA shall inform all other CAs being a subject of one of its CA-certificates on the changes of its contact information. The other CAs shall keep the contact information up-to-date and shall include it in each of their CAL as long as the corresponding CA-certificate is kept in the CAL.

The CPI includes a list of CA-certificates, validity information for each of these CA-certificates as well as a list of invalid end entity certificates issued by a subject of these CA-certificates and signed with the private key corresponding to the public key contained in the CA-certificate. Each CA-certificate in the list must either be signed with the private key for which the CPI is maintained, or have another CA-certificate in the list and be signed with the private key corresponding to the public key contained in this other CA-certificate.

NOTE 1 As a further condition, the CPI shall contain no loops in its certification paths. This condition is automatically fulfilled if the CPI is generated according to the procedures specified in this clause.

The CA-certificates shall be maintained in the CPI until they have expired. With the deletion of a CA-certificate, all other CA-certificates, not fulfilling the criteria for inclusion in the CPI any more, shall be deleted as well.

The procedure to issue a CA-certificate shall consist of the following steps:

a) The intended issuing CA A shall request key information from the intended subject CA B.

b) CA B shall send CA A its unique identifier (see 7.8 and 7.9), all its public keys that have not yet expired according to the envisaged validity time interval, together with the key identifier (see 7.10 and 7.11) and the information on the envisaged validity time interval (see 5.4.1) for each of them.

c) CA A shall verify the information received, select the appropriate public key, make sure that the conditions listed in 5.3 are fulfilled, and issue the CA-certificate including the public key and key identifier received. It shall inform CA B about the CA-certificate being issued, the key identifier of the public key included in it, and its time interval of initial validity. Furthermore, it shall forward its contact information.

d) If CA A is already on the CAL of CA B maintained for the public key identified in the information received according to step c, then CA B shall add the information on the time interval of initial validity to the corresponding list entry. If CA A was not yet on the CAL, then CA B shall put it there, together with the information on the time interval of initial validity and the contact information received, and shall send CA A its complete CPI maintained for the public key identified.

e) CA A, when receiving this information, shall:

   1) add the CA-certificate it issued for CA B to its appropriate CPI (being the one maintained for the public key corresponding to the private key used to sign the CA-certificate);

   2) add the CPI received from CA B to its own CPI mentioned in e)1);

3) update its CPI mentioned in e)1) such that it does not contain CA-certificates or end entity certificate validity information more than once and does not contain CA-certificates with CA A as the subject or validity information on end entity certificates issued by CA A;

4) forward the additional CA-certificates and validity information resulting from e)1), e)2) and e)3), together with the key identifier of the public key for which the updated CPI is maintained, to all CA in the CAL maintained for that public key except CA B.

f) Each CA C receiving information according to step e)4) above shall:

1) identify all of its own CA-certificates issued for CA A and containing the public key identified in the information received according to step e)4) as the subject public key. It shall add the CA-certificates and validity information received from CA A to the CPI maintained for these public keys;

2) update all CPI affected by f)1) in an analogous way as described in e)3), now deleting CA-certificates and validity information related to CA C and any other CA being on a certification path from CA C to CA A;

3) forward the additional CA-certificates and validity information resulting from f)1) and f)2), together with the key identifier of the public key for which the updated CPI is maintained, to all CA in the CAL, maintained for that public key, except CA A, CA B and those CAs being on a certification path from CA C to CA A. No information shall be forwarded if the CAL contains no other CA than CA A, CA B or CAs that are on a certification path from CA C to CA A.

Each CA, receiving information according to step f)3) above, shall go through the complete step f). The procedure goes on as long as there is information to be forwarded. A summary of the procedure is given in Figure 4.

NOTE 2 In the procedure as described, some CA may receive information several times, once for each certification path from them to CA A. After having received the information the first time, the other information can be discarded.

All information exchanged according to steps b) to f) needs integrity related protection. The sender of the information must make sure that the appropriate CA receives the information.

NOTE 3 The measures to protect the information exchanged according to steps b) to f) are outside the scope of this International Standard.

If a CA implements a new key pair to be used to sign certificates, it shall inform all CA being in one of its CAL and on request send its unique identifier, the public key together with its identifier and the envisaged validity time interval to them. Each of these CA may then issue a CA-certificate for the new public key according to step c) above, and the procedure for issuing a CA-certificate will apply including steps c) to f).

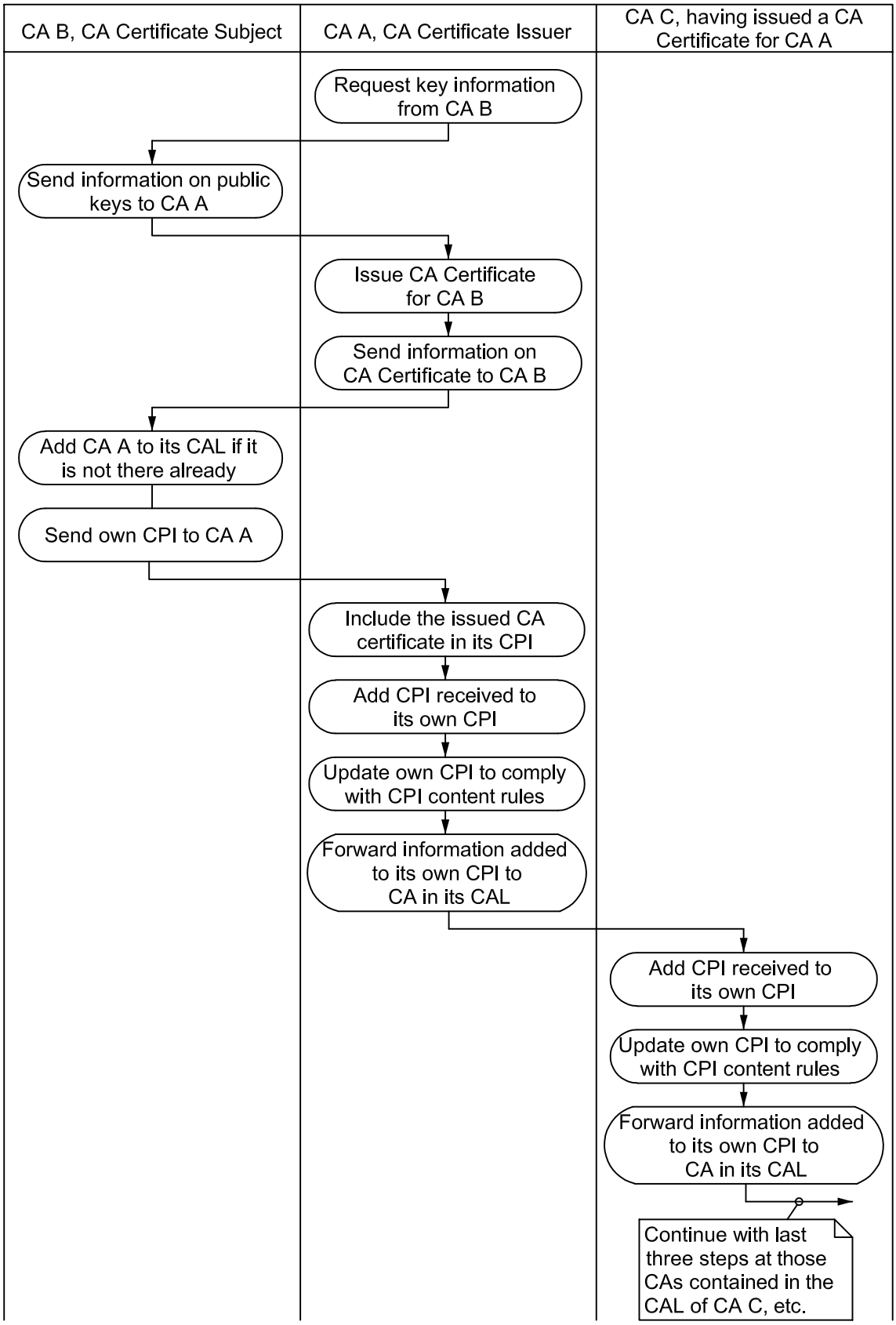


Figure 4 — Summary of the procedure for Certification Path Information (CPI) distribution

The information on revocation and re-establishment of any certificate shall be distributed among the CA, as soon as this information is generated. The procedure is as follows: The CA revoking or re-establishing the certificate forwards the corresponding information, including the unique identifiers of the issuing CA and of the subject as well as the key identifier of the public key corresponding to the private key used to sign it, to all CAs on its CAL maintained for the mentioned public key. Each CA receiving the information shall:

a) check if it has received the information already. If so, it will not further process it;

b) identify all of its own CA-certificates containing the public key identified in the information received as the subject public key and note the information in the CPI maintained for the public key corresponding to the private key used to sign that CA-certificate;

c) if the CA has noted the information, forward it to all CA on its CAL, excluding the CA having generated it and all other CA being on a certification path to that CA.

5.6.2 Path certificate

The CPI each CA gains through the procedures outlined in 5.6.1 shall be the basis for the information on established trust the CA forwards on request to its end entities.

If, as noted in 5.3, the CA assigns the task of distributing information about trust between CA to a different entity, the information shall be available at this entity and forwarded from there on request. The subsequent description is based on the assumption that the CA provides the information, but may be easily adapted to the other entity providing it.

The request will name the unique identifier of the end entity, for which trust shall be established, the unique identifier of its CA, the key identifier of the public key corresponding to the private key used to sign the end entity certificate, and the serial number of the end entity certificate. These items are extracted from the certificate received from the end entity to be trusted. As an option, the request may include a list of data categories of the data to be exchanged.

NOTE 1 The methods to exchange information between a CA and its end entities are outside the scope of this International Standard. It is recommended that integrity-related protection be applied to the request (see 5.1.1).

The CA shall analyse if its certification path information contains certification paths to the CA indicated in the request. If this is not the case, it shall send a response indicating that the CA can't be trusted.

The procedure on how to find out the certification paths to a CA out of the certification path information is not fixed in this International Standard. The procedure may be applied immediately after receiving information on established trust for all CA contained therein, or at the time of the request. In any case, the CA shall maintain the certificates received, from which the information is derived.

If there is at least one certification path, then the CA shall check if the certificate of the end entity indicated has been revoked and not re-established. In this case it shall send a response indicating that the end entity can’t be trusted.

If this is not the case, then the CA shall determine the validity of all certification paths to the CA indicated in the request. A certification path is not valid if one of its certificates has been revoked and not been re-established, or if the intersection of the time intervals of initial validity of all its certificates is zero (meaning that one of the certificates expires before another one starts its validity). If none of the certification paths investigated is valid, then the CA shall send a response to the end entity indicating that the CA listed in the request can’t be trusted.

If the certification path is valid, the validity starts at the latest start date of validity and ends at the earliest expiry date of the certificates involved.

As the next step, the CA shall determine the certificate policy of the valid certification paths. The in-list of this certificate policy consists of the data categories that are contained at least in one of the in-lists of the certificates included in the certification path. Data categories appearing in several in-lists shall only be listed once. Data categories that contain another data category listed in an in-list shall be deleted. The ex-list consists of the data categories listed at least in one of the ex-lists of the certificates included in the certification

path. Data categories appearing in several ex-lists shall only be listed once. Data categories that are contained in other data categories of an ex-list shall be deleted.

The certificate policy of a certification path shall be empty if all data contained in every data category listed in the in-list of the certification path is contained at least in one of the data categories listed in the ex-list of the certification path. The following special conditions are sufficient, but not necessary, for the certificate policy of the certification path to be empty:

a) All possible data are contained in a data category of the ex-list. The ex-list covers all data.

b) The in-list consists of data categories that are mutually exclusive, meaning that no data can be contained in all of them.

c) A data category listed in the in-list is equal to or contained in a data category listed in the ex-list.

If the certificate policy of all certification paths is empty, then the CA shall send a response to the end entity indicating that the CA listed in the request can't be trusted. If the certificate policy is not empty for all certification paths, but the request listed categories of data to be exchanged and all these data categories are not compliant to the certificate policy of any of the valid certification paths, then the CA shall send a response to the end entity indicating that the CA listed in the request can't be trusted with respect to the data categories requested.

In all other cases, the CA shall send a path certificate to the requesting end entity. The path certificate is a certificate issued by the CA of the end-entity having sent the request for information on established trust, containing as the subject the CA named in the request with the public key included in the request. Instead of one time interval of initial validity and one certificate policy, it includes such a time interval and policy for each valid, non-empty certification path from the issuer to the subject.

An end entity A, having received the certificate of the end entity B, with which it intends to exchange sensitive data, and the path certificate of the CA of B, shall establish the trust in B to exchange the data, if the data complies with the certificate policy of the certificate of B and to at least one of the certificate policies of a certification path included in the path certificate.

NOTE 2 If the data to be exchanged complies with the certificate policy of one certification path, but does not comply with the certificate policy of another one with an earlier expiry date, then the path certificate does not justify any trust for the exchange of data of the given type after that earlier expiry date.

The definition of the certificate policy of a certification path and of the use of path certificates implies that if CA A trusts CA B with respect to a certain data category, and CA B also trusts CA C with respect to the same data category, then CA A automatically trusts CA C with respect to that data category, even if it issues a CA-certificate for C excluding it. To exclude the trust in C for that data category, it is necessary for CA A to exclude it in the CA-certificate for CA B or to cause CA B to exclude it in its CA-certificate for CA C.

6 Certificate structure

End entity certificates and CA-certificates shall be according to ISO/IEC 9594-8 (so-called X509 certificates). They shall be version v3. The certificates shall be encoded using the Distinguished Encoding Rules (DER) specified in ISO/IEC 8825-1. The certificates shall include components and extensions according to 7.1 to 7.14. They may include further extensions, but these shall be marked as non-critical (see ISO/IEC 9594-8).

If a CA has issued a certificate for an end entity, that is not compliant to the specification in this clause, and if that end entity needs to exchange data of class 2 or 3 with an end entity having a different CA, then the CA shall issue a second certificate for its end entity having essentially the same content than the first one, but complying to the specification in this clause. If the end entity exchanges data of class 0 or 1, then it doesn't need to provide its certificate, and if the other end entity involved in the data exchange has the same CA, then it is expected to be able to use the certificate not complying with the specification in this clause.

Path certificates, being sent by the CA of an end entity to that end entity to confirm established trust to a second CA (see 5.6.2), have the same structure as normal certificates according to Clause 7, with the following exceptions:

— The issuer component, validity component and subject component (see 7.4 to 7.6) shall be absent.

— The certificate policy extension (see 7.13) shall be absent.

— The path certificates shall include a path information extension as defined in 7.15. The path information generalizes the validity and certificate policy information.

A complete specification of path certificates is given in Annex A.

7 Certificate components and extensions

7.1 General

The following clauses specify components and extensions of the certificate according to ISO/IEC 9594-8 and of the path certificate according to this International Standard. The specification uses the ASN.1 notation according to ISO/IEC 8824-1, ISO/IEC 8824-2 and ISO/IEC 8824-3. All ASN.1 constructs are noted in bold. A summary of the ASN.1 constructs is given in Annex A, where all values for the **OBJECT IDENTIFIER** data types used in this clause are listed.

NOTE The identifiers used for the ASN.1 constructs have no direct relevance and should not be interpreted, even though they are chosen such that they correspond more or less to the use of the constructs. In that sense, the character sequence **scm** is used in many identifiers to express the specific use of the identifiers in the framework of this International Standard on Security Certificate Management.

7.2 Certificate version

The **version** component of the **Certificate** data type according to ISO/IEC 9594-8 shall have fixed value 2, indicating that the certificate version is v3 (see Clause 6).

7.3 Certificate serial number

The **serialNumber** component of the **Certificate** data type shall be as defined in ISO/IEC 9594-8. The **serialNumber** shall be unique for each certificate issued by a given CA (i.e. the **issuerUniqueIdentifier** and the **serialNumber** shall identify a unique instance of **Certificate**).

If a CA sends path certificates for the same subject to several of its end entities (see 5.6.2), then the same serial number may be used (and in fact the same path certificate sent) as long as the path information for all certification paths remains valid. As soon as some path information changes (certification paths lose their validity, the start date or expiry date of some certification path have changed, the certificate policy for a path changes or new certification paths are added), a path certificate with a new serial number shall be issued and sent to the requesting end entity.

7.4 Certificate signature algorithm identifier

The **signature** component of the **Certificate** data type shall be as defined in ISO/IEC 9594-8. The content of this component shall comply with IETF RFC 3279. It is strongly recommended to use the Secure Hash Algorithm 1 (SHA-1) as the one way hash function and the Elliptic Curve Digital Signature Algorithm (ECDSA).

7.5 Certificate issuer

For all certificate issuers, the **issuer** component of the **Certificate** data type according to ISO/IEC 9594-8 shall consist of the same object identifier pointing to this International Standard.

NOTE 1 To identify the issuers providing certificates according to this International Standard, the **issuerUniqueIdentifier** data type is used.

NOTE 2 Certificates that are issued according to this International Standard may be identified by their property of containing the object identifier pointing to this International Standard in the **issuer** component.

An information object **scmName** according to ISO/IEC 8824-2 shall be added to the **SupportedAttributes** information object set in ISO/IEC 9594-2, defined as follows:

scmName ATTRIBUTE ::= {
ID **scm-standard**
WITH SYNTAX **NULL**
}

The **issuer** component of the **Certificate** data type according to ISO/IEC 9594-8 shall contain only one **RelativeDistinguishedName** with one **AttributeTypeAndValue** containing type and value fields from the **scmName** information object (see ISO/IEC 9594-2).

7.6 Certificate validity

The **validity** component of the **Certificate** data type shall be as defined in ISO/IEC 9594-8. It shall fix the time interval of initial validity of the certificate according to 5.4.1. The settings of section 4.1.2.5 of IETF RFC 3280 apply.

7.7 Certificate subject

For all certificate subjects, the **subject** component of the **Certificate** data type according to ISO/IEC 9594-8 shall consist of the same object identifier pointing to this International Standard.

NOTE To identify the subjects having certificates according to this International Standard, the **subjectUniqueIdentifier** data type is used.

The **subject** component of the **Certificate** data type according to ISO/IEC 9594-8 shall contain only one **RelativeDistinguishedName** with one **AttributeTypeAndValue** containing type and value fields from the **scmName** information object (see ISO/IEC 9594-2).

7.8 Certificate subject public key

The **subjectPublicKeyInfo** component of the **Certificate** data type shall be as defined in ISO/IEC 9594-8. The content of this component shall comply with IETF RFC 3279. It is strongly recommended to use the Secure Hash Algorithm 1 (SHA-1) as the one-way hash function and the Elliptic Curve Digital Signature Algorithm (ECDSA).

7.9 Certificate issuer unique identifier

The **issuerUniqueIdentifier** component of the **Certificate** data type shall always be present. It shall be used to uniquely identify the issuer of the certificate (see 5.2) among all entities according to this International Standard. It shall be a bit string as defined in ISO/IEC 9594-8. It shall have a size of 32 bits. The bit assignment of the **issuerUniqueIdentifier** is shown in Table 2.

Table 2 — Bit assignment of the unique identifier of a CA

Bit number	Use
1	Indication if the entity is a CA. Value shall be 1.
2 - 8	Binary integer number n ($0 \leqslant n \leqslant 127$) as a pointer to a specific register of CA.
9 - 31	According to the rules of the register of CA indicated in bits 2 to 8.
32	Reserved for future use. Value shall be 0.

7.10 Certificate subject unique identifier

The **subjectUniqueIdentifier** component of the **Certificate** data type shall always be present. It shall be used to uniquely identify the subject of the certificate (see 5.2) among all entities according to this International Standard. It shall be a bit string as defined in ISO/IEC 9594-8. It shall have a size of 32 bits. The bit assignment is as shown in Table 2 for CA-certificates and as shown in Table 3 for end entity certificates.

Table 3 — Bit assignment of the unique identifier of an end entity

Bit number	Use
1	Indication if the entity is a CA. Value shall be 0.
2 - 31	Assigned by the CA issuing the certificate.
32	Reserved for future use. Value shall be 0.

7.11 CA key identifier extension

The **authorityKeyIdentifier** extension according to ISO/IEC 9594-8 shall always be present. It shall identify the key pair of the CA being the issuer of the certificate, used to sign the certificate (see 5.2 and 5.6.1). It shall be assigned by the certificate issuer and shall be unique among its key pairs. The issuer shall always use the same key identifier for the same key pair. The **AuthorityKeyIdentifier** data type shall always include the **keyIdentifier** component and no other components. The **KeyIdentifier** data type, being an octet string, shall have size 2.

7.12 Certificate subject key identifier extension

The **subjectKeyIdentifier** extension according to ISO/IEC 9594-8 shall always be present for CA-certificates and certificates for end entities being able to renew their keys (see 5.4.3). It shall identify the key pair with the public key being included in the certificate as the subject public key (see 5.2 and 7.7). In case of an end entity certificate, its value shall be assigned by the CA issuing the certificate in such a way that every key pair of the subject certified by the CA has a unique identifier. In case of a CA-certificate, the key identifier assigned by the subject of the certificate shall be used (see 5.6.1). The **SubjectKeyIdentifier** data type, being an octet string, shall have size 2.

7.13 Extended key usage extension

The **extKeyUsage** extension according to ISO/IEC 9594-8 shall always be present. This extension shall include only one **KeyPurposeId** data type indicating the version number of this International Standard. The current version is 1. The extension shall be marked as non-critical.

7.14 Certificate policies extension

The certificate may include a **certificatePolicies** extension according to ISO/IEC 9594-8. It shall list the certificate policies according to 5.5. If it is not present, then the certificate shall support all certificate policies (corresponding to an empty in-list and ex-list).

If the certificate policies extension is present, then it shall contain a **policyIdentifier** component with fixed value **{iso standard 20828}**, indicating that the certificate policy is listed according to this International Standard. The **policyQualifiers** component shall contain no other **PolicyQualifierInfo** data types than the in-list and the ex-list.

To define the in-list and the ex-list, the following information objects are specified and added to the **SupportedPolicyQualifiers** information object set:

```
inList CERT-POLICY-QUALIFIER ::= {
POLICY-QUALIFIER-ID     scm-inList
QUALIFIER-TYPE          ScmPolicy
}

exList CERT-POLICY-QUALIFIER ::= {
POLICY-QUALIFIER-ID     scm-exList
QUALIFIER-TYPE          ScmPolicy
}
```

To define values of the **PolicyQualifierInfo** data type, only these two information objects shall be used. Both information objects may or may not be present. They are present at most once. At least one of them must be present.

The **ScmPolicy** data type shall be as follows:

```
ScmPolicy ::= SEQUENCE SIZE (1..MAX) OF DataCategory

DataCategory ::= CHOICE {
combinedCategory        SEQUENCE OF SimpleCategory,     --Union of data categories in case of
-- the in-list, intersection of data categories in case of the ex-list - see 5.5.1--
simpleCategory          SimpleCategory
}

SimpleCategory ::= CHOICE {
standardCategory        StandardCategory,
userDefined             UserDefinedCategory
}

StandardCategory ::= CHOICE {
singleStandard          SingleStandardCategory,
parametrizedStandard    ParametrizedStandardCategory,
 }

SingleStandardCategory ::= ENUMERATED {
exCodeProt (0),
resistPhysAttack (1),
detectPhysAttack (2),
reportPhysAttack (3),
limitedAccess (4),
privacy (5),
personalData (6),
vehicleLocation (7),
inVehicle (8),
software (9),
calibration (10),
vehicleRepair (11),
userAccess (12),
superAccess (13),
vehicleSafety (14)
}

ParametrizedStandardCategory ::= SEQUENCE {
name                    ParametrizedCategoryName,
number                  INTEGER
}
```

```
ParametrizedCategoryName ::= ENUMERATED {
securityClass (0),
eal (1),
pathVerification (2),
entityInspection (3),
vehicleAccess (4)
}

UserDefinedCategory ::= CHOICE {
simpleUserCategory          OCTET STRING,
categoryContained           CategoryContained,
categoryContaining          CategoryContaining
}

CategoryContained ::= SEQUENCE {
containedCategory           OCTET STRING,
standardCategories          SEQUENCE OF DataCategory (WITH COMPONENTS
                            {UserDefinedCategory ABSENT})
 }
--Used for categories in the ex-list. The containedCategory is in the intersection of the
-- standardCategories--

CategoryContaining ::= SEQUENCE {
containingCategory          OCTET STRING,
standardCategories          SEQUENCE OF DataCategory (WITH COMPONENTS
                            {UserDefinedCategory ABSENT})
 }
--Used for categories in the in-list. The containingCategory is in the union of the standardCategories--
```

A description of the **SingleStandardCategory** and **ParametrizedCategoryName** items is given in Table 1.

7.15 Vehicle identification number extension

If the certificate is an end entity certificate issued by a vehicle manufacturer for a vehicle, it may include a **vin** extension. This extension contains the Vehicle Identification Number (VIN) of the vehicle according to ISO 3779 and ISO 3780. It is added to the **ExtensionSet** information object set in ISO 9594-8 and defined as follows:

```
vin EXTENSION ::= {
SYNTAX                      Vin
IDENTIFIED BY               scm-vin
}

Vin ::= PrintableString (SIZE (17) ^ (1| 2| 3| 4| 5| 6| 7| 8| 9| 0| A| B| C| D| E| F| G| H| J| K| L| M| N| P| R| S| T| U| V|
W| X| Y| Z))
```

If a certificate for a vehicle includes the VIN, then it shall be guaranteed that the device storing the private key, corresponding to the subject public key in the certificate, can't be detached from the vehicle with this VIN.

NOTE A vehicle manufacturer issuing a certificate for a vehicle, as a whole, does not exclude issuing separate certificates for specific equipment in the vehicle either by the vehicle manufacturer or by another CA, for instance by the equipment manufacturer or a public authority.

7.16 Path information extension

Path certificates must include the **pathInformation** extension. This extension contains the validity and certificate policy of the certification paths between the issuer and the subject. It is added to the **ExtensionSet** information object set in ISO 9594-8 and defined as follows:

```
pathInformation EXTENSION ::= {
SYNTAX                  PathInformation
IDENTIFIED BY           scm-pathInformation
}

PathInformation ::= SEQUENCE SIZE (1..MAX) OF CertificationPath
--For each certification path from the issuer to the subject one CertificationPath item has to be included--

CertificationPath ::= SEQUENCE {
validity                Validity,
inList                  ScmPolicy OPTIONAL,
exList                  ScmPolicy OPTIONAL
}
```

The **Validity** data type is as defined in 7.5. The **ScmPolicy** data type is as defined in 7.13.

Annex A
(normative)

Security Certificate Management ASN.1 module definition

This annex collects all of the ASN.1 type, value, information object class and information object definitions contained in this International Standard in the form of the ASN.1 module **SecurityCertificateManagement**. With the exception of the **ScmPathCertificate** type, it only repeats those ASN.1 constructs contained in Clause 7, where all explanation on their use is given. The purpose of this annex is to facilitate the implementation of the exchange of security certificates and security path certificates.

```
SecurityCertificateManagement {iso standard 20828} DEFINITIONS ::=
BEGIN

IMPORTS

SIGNED, Certificate, Version, CertificateSerialNumber, AlgorithmIdentifier, Validity,
SubjectPublicKeyInfo, UniqueIdentifier, Extensions, EXTENSION, authorityKeyIdentifier,
subjectKeyIdentifier, extKeyUsage, CERT-POLICY-QUALIFIER
FROM AuthenticationFramework {joint-iso-itu-t-ds(5) module(1) authenticationFramework(7) 4}

ATTRIBUTE
FROM InformationFramework {joint-iso-itu-t ds(5) module(1) informationFramework(1) 4};

ScmCertificate ::= Certificate (WITH COMPONENTS {... , version (v3), issuer (CONSTRAINED BY {
--There shall be only one AttributeTypeAndDistinguishedValue with type and value taken from the
--information object-- ATTRIBUTE:scmName}), subject (CONSTRAINED BY {--There shall be only one
--AttributeTypeAndDistinguishedValue with type and value taken from the information object
ATTRIBUTE:scmName}), issuerUniqueIdentifier (SIZE (32)) PRESENT, subjectUniqueIdentifier (SIZE
(32)) PRESENT, extensions (CONSTRAINED BY {--The following extensions must be present
EXTENSION:authorityKeyIdentifier, EXTENSION:subjectKeyIdentifier, EXTENSION:extKeyUsage,
--The following extension shall not be present-- EXTENSION:pathInformation}) PRESENT})

ScmPathCertificate ::= SIGNED {SEQUENCE {
version                 [0]  Version (v1),
serialNumber                 CertificateSerialNumber,
signature                    AlgorithmIdentifier,
subjectPublicKeyInfo         SubjectPublicKeyInfo,
issuerUniqueIdentifier  [1]  IMPLICIT UniqueIdentifier (SIZE (32)),
subjectUniqueIdentifier [2]  IMPLICIT UniqueIdentifier (SIZE (32)),
extensions              [3]  Extensions (CONSTRAINED BY {--The following extensions must be
                             --present:-- EXTENSION:authorityKeyIdentifier,
                             EXTENSION:subjectKeyIdentifier, EXTENSION:extKeyUsage,
                             EXTENSION:pathInformation})
}}

scmName ATTRIBUTE ::= {
ID                      scm-standard
WITH SYNTAX             NULL
}

inList CERT-POLICY-QUALIFIER ::= {
POLICY-QUALIFIER-ID     scm-inList
QUALIFIER-TYPE          ScmPolicy
}
```

```
exList CERT-POLICY-QUALIFIER ::= {
POLICY-QUALIFIER-ID         scm-exList
QUALIFIER-TYPE              ScmPolicy
}

ScmPolicy ::= SEQUENCE SIZE (1..MAX) OF DataCategory

DataCategory ::= SEQUENCE SIZE (1..MAX) OF SimpleCategory      --If more than one component, union
--of data categories in case of the in-list, intersection of data categories in case of the ex-list - see 5.5.1--

SimpleCategory ::= CHOICE {
standardCategory            StandardCategory,
userDefined                 UserDefinedCategory
}

StandardCategory ::= CHOICE {
singleStandard              SingleStandardCategory,
parametrizedStandard        ParametrizedStandardCategory
 }

SingleStandardCategory ::= ENUMERATED {
exCodeProt (0),
resistPhysAttack (1),
detectPhysAttack (2),
reportPhysAttack (3),
limitedAccess (4),
privacy (5),
personalData (6),
vehicleLocation (7),
inVehicle (8),
software (9),
calibration (10),
vehicleRepair (11),
userAccess (12),
superAccess (13),
vehicleSafety (14)
}

ParametrizedStandardCategory ::= SEQUENCE {
name                        ParametrizedCategoryName,
number                      INTEGER
}

ParametrizedCategoryName ::= ENUMERATED {
securityClass (0),
eal (1),
pathVerification (2),
entityInspection (3),
vehicleAccess (4)
}

UserDefinedCategory ::= CHOICE {
simpleUserCategory          OCTET STRING,
categoryContained           CategoryContained,
categoryContaining          CategoryContaining
}

CategoryContained ::= SEQUENCE {
containedCategory           OCTET STRING,
standardCategories          SEQUENCE OF DataCategory
```

```
 }
--Used for categories in the ex-list. The containedCategory is in the intersection of the standardCategories.
--If a DataCategory type contains several SimpleCategory types, then the category is the union of the
--categories corresponding to the SimpleCategory types--

CategoryContaining ::= SEQUENCE {
containingCategory          OCTET STRING,
standardCategories          SEQUENCE OF SimpleCategory
 }
--Used for categories in the in-list. The containingCategory is in the union of the standardCategories.--
--If a DataCategory type contains several SimpleCategory types, then the category is the intersection of the
--categories corresponding to the SimpleCategory types--

vin EXTENSION ::= {
SYNTAX                      Vin
IDENTIFIED BY               scm-vin
}

Vin ::= PrintableString (SIZE (17) ^ (1| 2| 3| 4| 5| 6| 7| 8| 9| 0| A| B| C| D| E| F| G| H| J| K| L| M| N| P| R| S| T| U| V|
W| X| Y| Z))

pathInformation EXTENSION ::= {
SYNTAX                      PathInformation
IDENTIFIED BY               scm-pathInformation
}

PathInformation ::= SEQUENCE SIZE (1..MAX) OF CertificationPath
--For each certification path from the issuer to the subject one CertificationPath item has to be included--

CertificationPath ::= SEQUENCE {
validity                    Validity,
inList                      ScmPolicy OPTIONAL,
exList                      ScmPolicy OPTIONAL
}

scm-standard OBJECT IDENTIFIER ::= {iso standard 20828}

scm-inList OBJECT IDENTIFIER ::= {scm-standard 1}

scm-exList OBJECT IDENTIFIER ::= {scm-standard 2}

scm-vin OBJECT IDENTIFIER ::= {scm-standard 3}

scm-pathInformation OBJECT IDENTIFIER ::= {scm-standard 4}

END
```

Annex B
(informative)

Certificate examples

This annex describes examples of certificate management. The description here is for informative purpose only, and it does not represent any implementation or specification.

B.1 Trusted Package Download

B.1.1 Security requirements

This application is used to forward a software package for updating an ECU in a car. The security requirement in this example is to protect the ECU from a malicious package for illegal remodelling. For that the package integrity should be guaranteed to ensure that:

— the package was developed and customized by authorized personnel, and

— the package has not been altered after development and customizing.

This security requirement is “Class2: Integrity-related protection” because a package itself is not a confidential data, but the modification of it may cause a serious problem.

B.1.2 Certificates

The certificates used in this example are as follows:

(1) Car maker CA -> Package vendor:

The “certificate subject” component of this certificate identifies the package vendor to indicate that this certificate contains a public key of this package vendor. It is used for a package vendor which develops a package to be distributed.

The “certificate issuer” component of this certificate identifies the car maker CA to indicate that this certificate is signed by a private key of this car maker CA. This certificate is for an end entity.

The “certificate policies” extension of this certificate contains the data category “Software” in the in-list to indicate that this certificate allows distribution of software.

(2) Car maker CA -> Workshop company CA:

The “certificate subject” component of this certificate identifies the workshop company CA to indicate that this certificate contains a public key of this workshop company CA. It is used for a workshop company with many branches that are entitled to customize software packages for ECU in cars.

The “certificate issuer” component of this certificate identifies the car maker CA to indicate that this certificate is signed by a private key of this car maker CA.

This certificate is a CA-certificate.

The “certificate policies” extension of this certificate contains the data category “Vehicle repair data” in the in-list to indicate that this certificate allows distribution of vehicle repair data, including software.

(3) Workshop company CA -> Workshop branch:

The “certificate subject” component of this certificate identifies the workshop branch to indicate that this certificate contains a public key of this workshop branch. It is used for a workshop branch customizing software packages for ECU in cars.

The “certificate issuer” component of this certificate identifies the workshop company CA to indicate that this certificate is signed by a private key of this workshop company CA.

This certificate is for an end entity.

This certificate has no “certificate policy” extension, indicating that the certificate allows the exchange of all types of data.

(4) Workshop company CA -> Car maker CA:

The “certificate subject” component of this certificate identifies the car maker CA to indicate that this certificate contains a public key of this car maker CA. It is used to express trust in the car maker to get software he authorized.

The “certificate issuer” component of this certificate identifies the workshop company CA to indicate that this certificate is signed by a private key of this workshop company CA.

This certificate is a CA-certificate.

The “certificate policies” extension of this certificate contains the data category “Software” in the in-list to indicate that this certificate allows distribution of software.

The structure of the certificates is shown in Figure B.1.

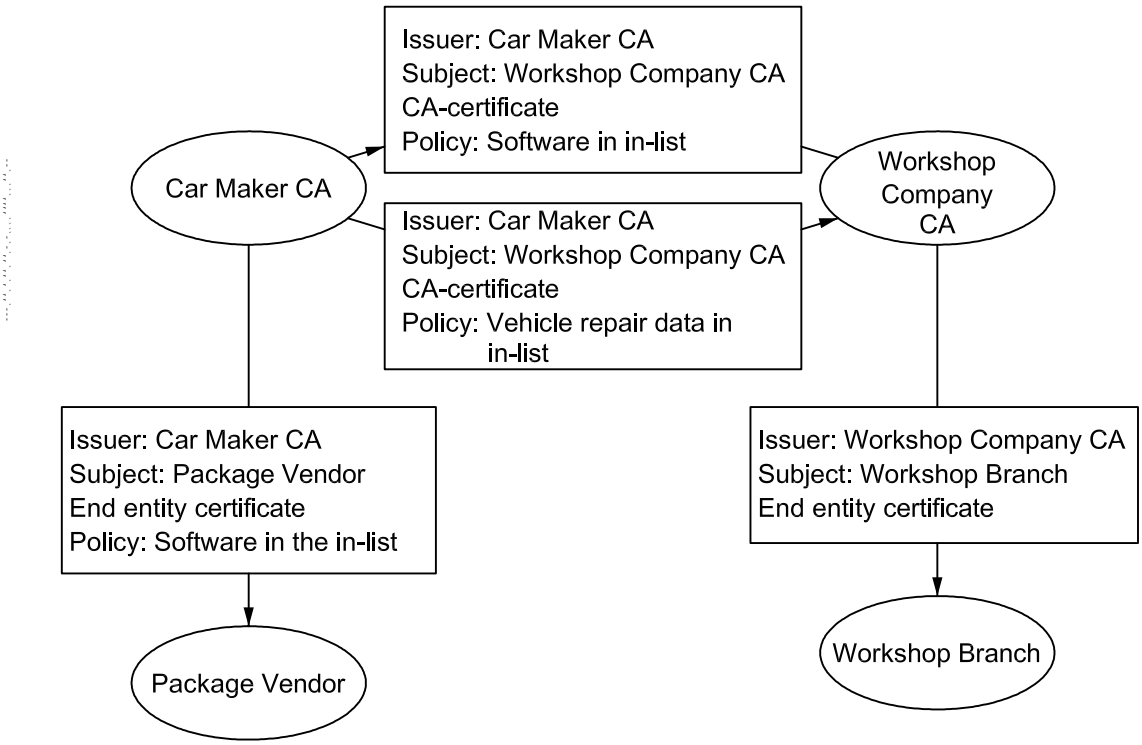


Figure B.1 — Structure of certificates for trusted package download

B.1.3 Information flow

a) An employee of a workshop branch intends to install a new software package in an ECU of a car. He sends a request for the package to the package vendor, indicating the type of ECU and the car make.

b) The package vendor forwards the software package requested to the server at the workshop, together with his certificate issued by the car maker CA responsible for the indicated car make. The software package is signed, using the private key of the package vendor corresponding to the public key in the certificate.

c) The server at the workshop verifies the signature of the software package, using the public key in the certificate received, and finds that the package was not altered since the package vendor signed it.

d) The server at the workshop sends a request for a path certificate to the workshop company CA, indicating the identifier of the package vendor and of the car maker CA having issued the package vendor certificate, the identifier of the key used to sign the certificate received from the package vendor and the serial number of this certificate. The request also lists the “Software” data category to indicate that based on the trust to be established software is intended to be exchanged.

e) The workshop company CA sends a path certificate to the server at the workshop, having as the subject the car maker CA and the public key identified in the request. This path certificate is based on the CA-certificate, the workshop company CA issued for the car maker CA and therefore has “Software” in the in-list of its certificate policy.

f) The server at the workshop has the public key of the workshop company CA stored in its security module and used as its trust anchor. It uses this public key to verify the path certificate. After successful verification, it knows that the certificate of the package vendor has not been revoked (as else the workshop company CA would not have sent the path certificate) and therefore takes the public key in the path certificate to verify the certificate of the package vendor. This verification being successful as well, the server knows that it can trust the software received from the package vendor. The employee will then get a confirmation that the software package may be downloaded to the car.

g) The employee will request from the car implementation information for the software package, for instance using a scan tool connected to the car. In the request, he will indicate the ECU for which the software package is intended.

h) The car will respond with all information needed to customize the software package according to the specific car. The response will include a non-repeating number.

i) Now the employee customizes, via some client to the server at the workshop, the software package, choosing appropriate software parameters for the target car. He forwards the customized software package to the car via an appropriate communication link. To the software package, the random number received from the car is appended and the package with the appended non-repeating number is signed using the private key of the server at the workshop. Together with the package, the certificate of the workshop branch is forwarded to the car, including the public key corresponding to the private key used to sign the customized package.

j) The car, having received the package and the certificate, uses the public key in the certificate to verify the signature at the customized software package. Successful verification means that the software package has not been altered after being sent out from the workshop. As the next step, the car compares the non-repeating number with the number sent out in step h). Coincidence means that the software package has been customized according to the information the car forwarded, and is not the copy of a software package customized for a different car earlier on.

k) Then the car sends a request for a path certificate to the car maker CA, similar to the request in step d), but based on the certificate of the workshop branch received. The path certificate received in response will contain the “Vehicle repair data” data category in the in-list of the certificate policy.

l) The car has the public key of the car maker CA stored in it at the time it is shipped. It uses this trust anchor to verify the path certificate. Then it uses the public key extracted from the path certificate to verify the workshop branch certificate. The package received complies with the “Software” data category. As this category is contained in the “Vehicle repair data” category included in the path certificate (see Table 1 in 5.5.2), the software module can be accepted if all verifications are successful.

As soon as one of the verifications fails, the whole procedure is stopped.

The processing steps and the information flow are shown in Figure B.2.

NOTE 1 If the message sequence according to ISO 15764 is used for the data exchange between the car and the workshop, then the random number mentioned in step h) is automatically included.

NOTE 2 Security is only maintained if the employee at the workshop customizes the software package according to the relevant rules. The information the workshop sends to the car may include an audit trail, indicating for instance who customized the software package.

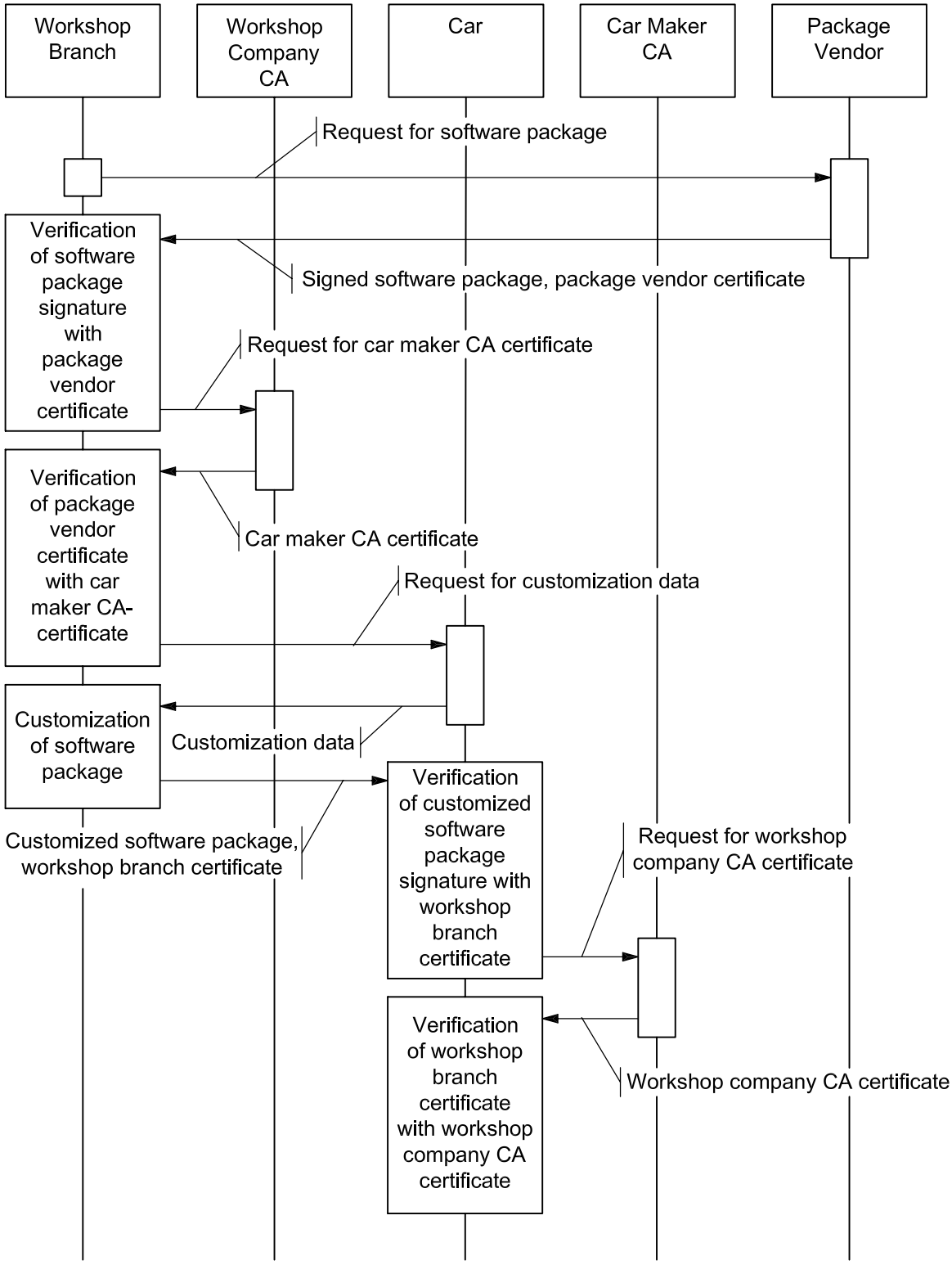


Figure B.2 — Sequence of processing steps and information flow for trusted package download

B.2 Remote maintenance

B.2.1 Security requirements

This application is used to perform remote maintenance from a remote server to a car via air communication. Remote maintenance includes retrieving diagnostic data from the car, performing an operation to the car, etc. The security requirement in this example is to protect from tampering with the information exchanged and from unauthorized access to the car. Privacy-related information such as the car owner and the car location shall be protected. This communication should be "secured" to ensure that:

— the data exchanged is not altered, and

— unauthorized persons cannot access the data exchanged.

This security requirement is "Class3: Confidentiality and Integrity related protection" because both unauthorized access and altering may cause serious problems.

B.2.2 Certificates

Certificates used in this example are as follows:

a) Maintenance centre CA -> Tokyo centre:

The "certificate subject" component of this certificate identifies the Tokyo centre to indicate that this certificate contains a public key of the Tokyo centre. It is used for a specific centre which performs remote maintenance.

The "certificate issuer" component of this certificate identifies the maintenance centre CA to indicate that this certificate is signed by a private key of the maintenance centre CA.

The certificate is issued for an end entity.

b) Car maker CA -> Car:

The "certificate subject" component of this certificate identifies the car to indicate that this certificate contains a public key of this car. It is used for the car under remote maintenance.

The "certificate issuer" component of this certificate identifies the car maker CA to indicate that this certificate is signed by a private key of this car maker CA.

The certificate is for an end entity.

c) Car maker CA -> Maintenance centre CA:

The "certificate subject" component of this certificate identifies the maintenance centre CA to indicate that this certificate contains a public key of this maintenance centre CA. It is used for a maintenance centre.

The "certificate issuer" component of this certificate identifies the car Maker CA to indicate that this certificate is signed by a private key of this car maker CA.

This certificate is a CA-certificate.

d) Maintenance centre CA -> Car maker CA:

The "certificate subject" component of this certificate identifies the car maker CA to indicate that this certificate contains a public key of this car maker CA. It is used for a car maker.

The "certificate issuer" component of this certificate identifies the maintenance centre CA to indicate that this certificate is signed by a private key of this maintenance centre CA.

This certificate is a CA-certificate.

NOTE The two certificates "Car maker CA -> Maintenance centre CA" and "Maintenance centre CA -> Car maker CA" are so-called cross certificates.

The structure of the certificates is shown in Figure B.3.

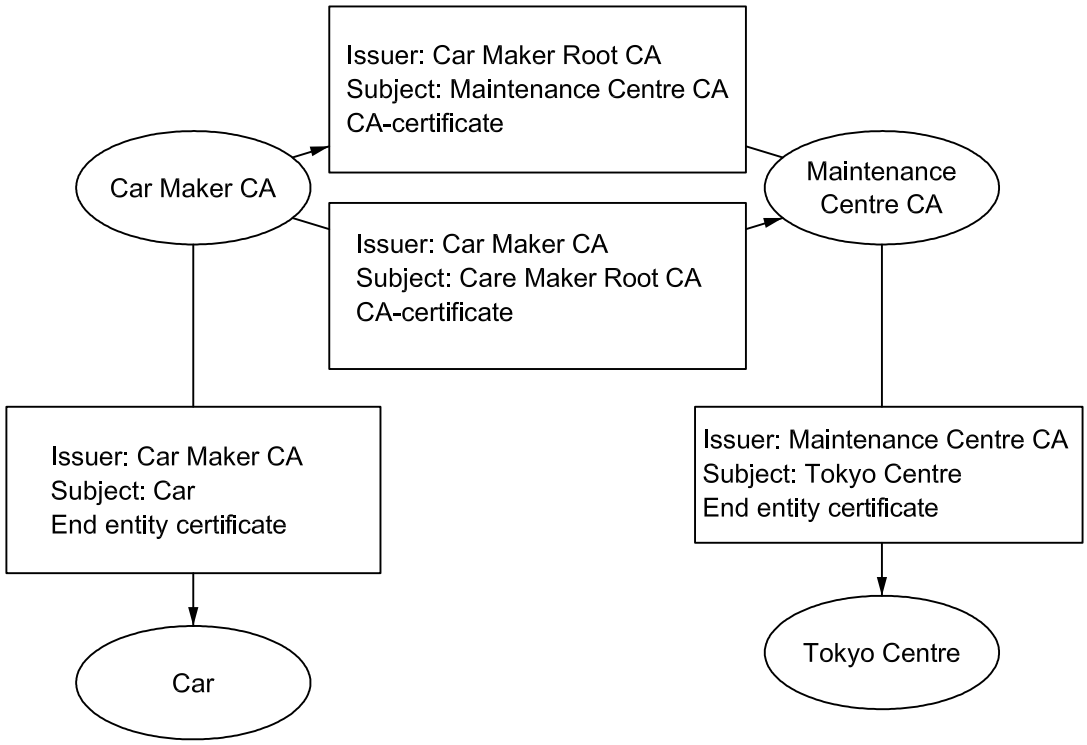


Figure B.3 — Structure of certificates for remote maintenance

B.2.3 Information flow

a) At the beginning of the communication, mutual authentication between a car and its remote maintenance centre (Tokyo centre) is performed. It includes the exchange of the corresponding certificates.

b) The exchange of the maintenance data may be protected applying encryption at the sender of the data with the public key extracted from the certificate received, and decryption at the receiver with the corresponding private key. Data needing integrity related protection may be signed by the sender with his private key and the signature may be verified by the receiver, using the public key extracted from the certificate received.

c) As an additional option, the car and the centre may use their public and private keys to establish a session key that they keep secret and use to protect the data exchanged. One possible implementation of this option is specified in ISO 15764.

d) In any case, both entities must make sure that they can trust the public key they received. For this they must verify the certificate from which they extract the key, as described in B.2.4 and B.2.5, respectively.

B.2.4 Certificate verification at the car

Verification of the certificate at the car is performed as follows:

a) The initial set of information at the car is:

   1) Trust anchor: Car maker CA public key.

   2) Target certificate: Tokyo centre certificate.

b) The car requests a path certificate for the information contained in the target certificate from the car maker CA.

c) The car maker CA checks its CPI and identifies the certification path to the maintenance centre CA consisting of the CA-certificate it issued for this CA. Based on its CPI it confirms that the CA-certificate and the target certificate are both valid. Then the car maker CA sends the requested path certificate to the car.

d) The car verifies the signature of the path certificate using the trust anchor. After successful verification, it extracts the public key contained in the path certificate and uses it to verify the signature of the target certificate.

e) If all above verification succeeds, the target certificate is verified. Else the car stops the data exchange.

B.2.5 Certificate verification at the centre

Verification of the certificate at the Tokyo centre is performed as follows:

a) The initial set of information at the centre is:

   1) Trust anchor: Maintenance centre CA public key.

   2) Target certificate: Car certificate.

b) The centre requests a path certificate for the information contained in the target certificate from the maintenance centre CA.

c) The maintenance centre CA checks its CPI and identifies the certification path to the car maker CA consisting of the CA-certificate it issued for this CA. Based on its CPI, it confirms that the CA-certificate and the target certificate are both valid. Then the maintenance centre CA sends the requested path certificate to the Tokyo centre.

d) The Tokyo centre verifies the signature of the path certificate using the trust anchor. After successful verification, it extracts the public key contained in the path certificate and uses it to verify the signature of the target certificate.

e) If all of the above verification succeeds, the target certificate is verified, or else the centre stops the data exchange.

ICS 43.020

Price based on 38 pages