



INTERNATIONAL STANDARD ISO/IEC 9594-8:2014
TECHNICAL CORRIGENDUM 2

Published 2016-10-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Open Systems Interconnection — The Directory —

Part 8: Public-key and attribute certificate frameworks

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — L'annuaire —

Partie 8: Cadre général des certificats de clé publique et d'attribut

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to ISO/IEC 9594-8:2014 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*. The identical text is published as Rec. ITU-T X.509 (2012)/ Cor.2 (04/2016).

INTERNATIONAL STANDARD
ITU-T RECOMMENDATIONInformation technology – Open Systems Interconnection –
The Directory: Public-key and attribute certificate frameworks

Technical Corrigendum 2

*(Covering resolution to defect reports 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419 and 420)***1) Correction of the defects reported in defect report 406***In clause 8.5.3.1 and Annex A, update the ASN.1 for **CRLReason** as follows:*

```

CRLReason ::= ENUMERATED {
  unspecified          (0),
  keyCompromise        (1),
  cACompromise         (2),
  affiliationChanged    (3),
  superseded           (4),
  cessationOfOperation (5),
  certificateHold       (6),
  removeFromCRL         (8),
  privilegeWithdrawn    (9),
  aACompromise         (10),
  ... weakAlgorithmOrKey (11) }

```

Add a new bullet point:

- **weakAlgorithm** indicates that the certificate was revoked due to a weak cryptographic algorithm and/or key (e.g., due to short key length or unsafe key generation).

*In clause 8.6.2.1 and Annex A, update the ASN.1 for **ReasonFlags** as follows:*

```

ReasonFlags ::= BIT STRING {
  unused          (0),
  keyCompromise   (1),
  cACompromise    (2),
  affiliationChanged (3),
  superseded      (4),
  cessationOfOperation (5),
  certificateHold   (6),
  privilegeWithdrawn (7),
  aACompromise     (8),
  weakAlgorithmOrKey (9) } (SIZE(0..9, ..., 10))

```

2) Correction of the defects reported in defect report 407*Add the following abbreviations to clause 4:*

DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
RSA	Rivest-Shamir-Adelman

*Delete the old DSA abbreviation**Update 6.1 as shown***6.1 Digital signatures***Replace this clause with:*

This subclause describes digital signatures in general. Sections 2 and 3 of this Directory Specification discuss the use of digital signatures within PKI and PMI specifically. This subclause is not intended to specify a specification for digital signatures in general, but to specify the means by which instances of the PKI and PMI specific data types are signed.

There are different types of digital signatures, such as RSA digital signatures, DSA digital signatures and ECDSA digital signatures.

NOTE 1 – It is not within the scope of this Specification to describe these different techniques in details, but Annex F gives a short introduction with references to more detailed specifications.

Figure 1 illustrates how a signer of instances of PKI/PMI data types (public-key certificates, attribute certificates, revocation lists, etc.) creates a digital signature and then adds that to the data before transmission. It also illustrates how the recipient of the signed data (the validator) validates the digital signature.

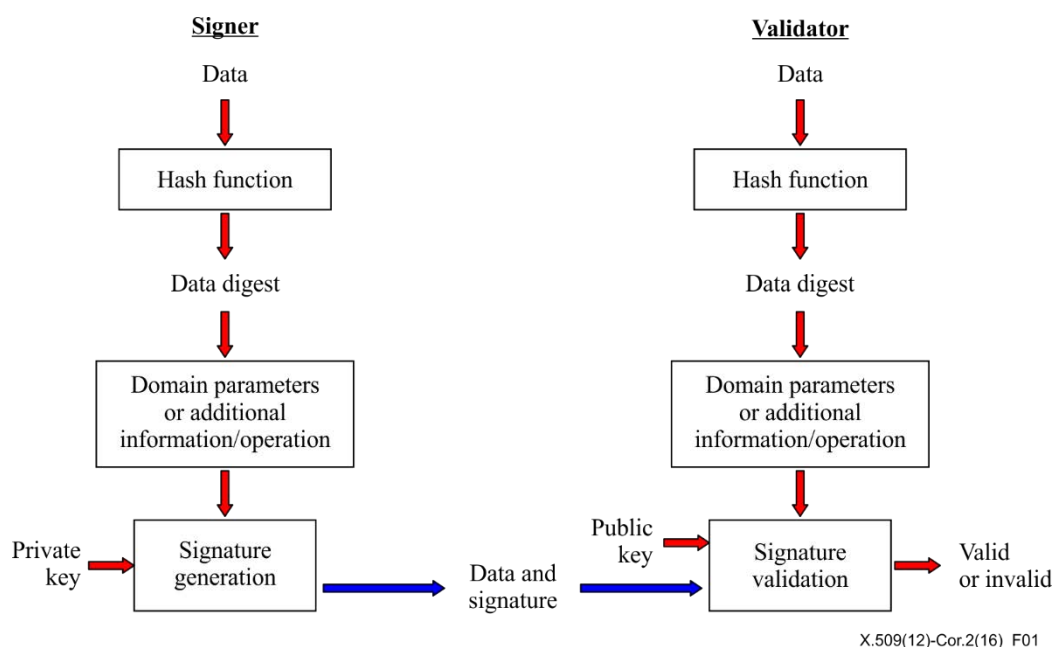


Figure 1 – Digital signature generation and validation

The digital signature signer goes through the following procedure:

- 1) The signer creates a hash digest over the PKI/PMI data using a secure hashing algorithm (see Annex F).
- 2) The hash digest is then supplemented with additional information in preparation for generating of the digital signature for improved security and for padding the hash digest to a length required by the asymmetric cryptographic function. For the RSA algorithms that can be adding some information to the hash digest and in some cases, to perform yet another hashing operation. For the DSA and ECDSA signature algorithms additional domain parameters are added.
- 3) The result from item 2) together with the private key of the signer and the use of a specific algorithm result in a bit string that together with an identity of the used algorithm constitute the digital signature.
- 4) The signature is appended to data to be signed.

Having received the data, the recipient (validator) goes through a similar procedure:

- 1) The validator goes through the same procedure as in steps 1) and 2) above, and if the received data is unmodified, the result will be the same as for the signer. If not, the next step will fail.
- 2) The result from item 1) together with the public key of the signer, the bit string of the signature and the use of a corresponding algorithm, the digital signature is evaluated as either valid or invalid.

If the digital signature proves valid, the validator has ensured that the data has not been modified and that the signer is in the position of the private key that corresponds to the public key used by the validator, i.e., the digital signature provides insurance of data integrity and authentication of the signer.

If the digital signature proves invalid, either the data has been modified or the signing private key does not corresponds to the public key used by the validator.

NOTE 2 – Data to be stored in a database or a directory may also be appended a digital signature, which then can evaluated at the retrieval of the data.