Elliptic Curve Cryptography in X.509-based PKI

Sibylle Hick[1,2], Luigi Lo Iacono[3]

[1]Institute for Data Communications Systems (DCS), University of Siegen
Hoelderlinstraße 3
D-57076 Siegen (Germany)

[2]secunet Security Networks AG
Kronprinzenstrasse 30
D-45128 Essen (Germany)
sibylle.hick@secunet.com

[3]NEC Laboratories Europe
IT Research Division
Rathausallee 10
D-53757 Sankt Augustin (Germany)
lo_iacono@it.neclab.eu

Abstract: In recent years a new category of digital signature algorithms based on Elliptic Curve Cryptography (ECC) has taken place besides well known schemes as RSA or DSA. So far it is, however, still not obvious how ECC-based signature schemes can be integrated in X.509-based Public Key Infrastructures (PKI).This paper briefly introduces cryptographic basics of signature schemes based on elliptic curves and points out the necessary cryptography parameters that are important in this context. Afterwards the structure and the encoding of X.509 certificates and Certificate Revocation Lists (CRL) are discussed regarding the integration of ECC public keys and ECC signatures respectively. The paper closes with exemplary implementations of ECC-based security systems.

1. Introduction

The use of Elliptic Curve Cryptography (ECC) is connected to various advantages compared to the previously issued asymmetric cryptographic methods regarding security, applied key length and computational time, offering a wide variety of possibilities in different security applications. The exploitation of ECC-based signature schemes in the scope of limited resources such as smartcards for example makes explicitly use of these properties and is therefore one particular research and development interest.

These benefits of ECC are also recognized in the scope of standardisation and are actively taken into account by various standardisation bodies. In January 1999 the ANSI X9.62 Standard [ANS99] was introduced, specifying the Elliptic Curve Digital Signature Algorithm (ECDSA). In February 2000 the NIST FIPS 186-2 [FIP00] standard followed ANSI X9.62 whereas the FIPS 186-2 additionally introduced a list of recommended elliptic curves. In August of the same year IEEE released the P1363 [IEEE01] standard which includes specifications of ECC schemes that can be used for signing, encryption and key exchange. In 2002 ECC schemes were standardised in ISO/IEC 15946 [ISO02a], [ISO02b], [ISO02c]. The first three parts in this standards family enclose

basic information and specifications, signature schemes such as ECDSA, ECGDSA (German variant) and ECKCDSA (South Korean variant) as well as key exchange schemes. The fourth part [ISO05] was introduced in 2005 and extended the standards family with signatures giving message recovery. ECC is also in the process of being added to the standards that are established by the IETF. Here especially RFC 3279 [IET02a] and RFC 3280 [IET02b] can be named. They contain data structures for signatures and key agreement. Besides the standardisation within national and international standardisation bodies industry-driven standards have also been established. The standards SEC 1 [SEC00a] and SEC 2 [SEC00b] for instance are mainly influenced by Certicom Corp.

Although ECC is recognized as a very reliable method it is still not obvious how ECC can be integrated in X.509-based Public Key Infrastructures (PKI) which is possibly also a reason for the missing adoption of these schemes by certification service providers. There has been an attempt to specify the use of ECC within the X.509 framework by the SECG X.509 Working Group [SEC99]. However, the developed draft has not been advanced since 1999 and is therefore still uncompleted work. This paper provides an overview of the necessary steps for the integration of ECC into X.509. In section 2 the fundamentals of ECC-based signature schemes are introduced in order to outline how this technique can be integrated in public key infrastructures. Section 3 identifies the necessary enhancements and alignments to X.509 certificates as well as certificate revocation lists. ECC is more and more used in industrial and in governmental projects which will be given in extracts and exemplary in section 4. Finally section 5 summarises the overview provided by this paper.

2. Elliptic Curve Cryptography

In order to understand what kind of information needs to be managed in the X.509 data structures for ECC, a brief introduction to ECC is given. The focus is set in favour of an easy understanding. A more formal and comprehensive guide can e.g. be found in [HMV04].

The concept of using elliptic curves in cryptography was independently proposed in 1985 by Victor Miller [Mi86] and Neal Koblitz [Ko87]. Elliptic curves used in cryptography are defined by the following simplified affine Weierstrass equation:

$$E: y^2 = x^3 + ax + b \qquad (1)$$

They can be represented over different types of numbers. For cryptographic purposes the representation over finite fields $\boldsymbol{F}_p$ (where p is prime) and $\boldsymbol{F}_{2^n}$ is used commonly. The coefficients a and b identify an elliptic curve unambiguously. The solutions of (1) are given by the points P_i (x_i, y_i) which are part of the elliptic curve. If two points of an elliptic curve are added the result is again a point that resides on this curve. Taking this point addition together with the point of infinity as well as some further properties an Abelian group is formed. Given a primitive element (also known as generator) in form of a point all points P_i of the elliptic curve can be computed by adding the primitive

element successively (cyclic group). Moreover, a scalar multiplication can be defined by applying the point addition n times. This operation can be done very easily over integers, the inverse operation, however (again over integers) is computational infeasible and would require a logarithmic operation. Therefore ECC is based on the Discrete Logarithm Problem (DLP).

Not every elliptic curve can be applied for cryptographic purposes. A curve has to fulfil various properties in order to be recognized as suitable for cryptography. The process to determine elliptic curves with good cryptographic properties is complex and rather time-consuming. Thus, domain parameters of elliptic curves known to have good cryptographic properties have been standardised by distinct standards and specifications (see section 1). Unfortunately, different names have been assigned by the numerous standards leading to a set of identifiers for a particular curve. Table 1 shows an exemplary abstract of standardised elliptic curves sorted by key length:

Table 1. Selected standardised elliptic curves

Key Length	Finite Field	Object ID (OID)	ANSI X9.62	NIST	SEC 2
160 Bit	F_p	1.3.132.0.30	-	-	secp160r2
163 Bit	F_{2^n}	1.3.132.0.15	-	B-163	sect163r2
191 Bit	F_{2^n}	1.2.840.10045.3.0.5	c2tnb191v1	-	-
191 Bit	F_{2^n}	1.2.840.10045.3.0.6	c2tnb191v2	-	-
191 Bit	F_{2^n}	1.2.840.10045.3.0.7	c2tnb191v3	-	-
192 Bit	F_p	1.2.840.10045.3.1.1	prime192v1	P-192	secp192r1
192 Bit	F_p	1.2.840.10045.3.1.2	prime192v2	-	-
192 Bit	F_p	1.2.840.10045.3.1.3	prime192v3	-	-
256 Bit	F_p	1.2.840.10045.3.1.7	prime256v1	P-256	secp256r1
283 Bit	F_{2^n}	1.3.132.0.17	-	B-283	sect283r1

The elliptic curve with the OID 1.2.840.10045.3.1.1 is standardised in ANSI X9.62, NIST and SEC2 and is therefore a good example to show the domain parameters in more detail. While the parameters *a, b,* and *p* have already been introduced *G* describes the base point that forms a cyclic group with order *q* (where *q* is prime). The cofactor h is sometimes needed in connection with key exchange.

Table 2. ECC domain parameters for prime192v1, P-192 and secp192r1

Name	Description	Value (hexadecimal)
a	coefficient	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFC
b	coefficient	64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1
p	prime	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFF
q	order, prime	FFFFFFFF FFFFFFFF FFFFFFFF 99DEF836 146BC9B1 B4D22831
G_x	base point x	188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012
G_y	base point y	07192B95FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811
h	cofactor	01

In the scope of ECC a private key is a random number d (a scalar) together with its system parameters while a public key is a point P_i which is calculated by the scalar multiplication of d and the domain-specific base point G. A special characteristic in ECC is the possibility to represent the public key in compressed form, herby only the x-coordinate is given completely while for the y-coordinate only the sign information is provided.

In general the form of an ECC signature does not differ from the form of other signatures which are generated using a DLP-based signature scheme. Since these signature schemes are randomized signature schemes, the signature is composed of the number pair r and s with $0 < r < q$ and $0 < s < q$ where r is a random number and s is the actual signature.

3. ECC in X.509v3 Certificates and CRL

The most common applied standard for public key certificates and aligned aspects such as Certificate Revocation Lists (CRL) and certificate chain validation schemes is the ITU-T recommendation X.509 [ITU01]. Most of the X.509 specifications like the structure of a certificate and a CRL can also be found in its internet interpretation RFC 3280 [IET02b].

In the context of this paper, X.509v2 CRL differs from X.509v3 certificates by not containing public keys but only a signature. Thus, in the following the descriptions will focus on certificates only whereas the application of ECC to CRL is equivalent to the explanations provided in section 3.2 regarding the structure and coding of ECC signatures in X.509v3 certificates. The data structures in the X.509 recommendation are specified using the data definition language ASN.1 [ITU02a] which enables in a platform and programming language independent way to specify data structures. The basic structure of an X.509v3 certificate is described by the following ASN.1 statements:

```
Certificate  ::=  SEQUENCE
{
  tbsCertificate      TBSCertificate,
  signatureAlgorithm  AlgorithmIdentifier,
  signatureValue      BIT STRING
}
```

A certificate is a sequence of octets which consists of a part to be signed (`TBSCertificate`), information about the signature algorithm that is used to sign the certificate (`AlgorithmIdentifier`) and finally the signature (`signatureValue`) of the issuer. The main part in a certificate is denoted as `TBSCertificate` and works as a container for the information that should be trusted. The public key together with information about the subject and the issuer as well as additional information is contained in the `TBSCertificate` structure:

```
TBSCertificate  ::=  SEQUENCE
{
  version         [0]   EXPLICIT Version DEFAULT v1,
  serialNumber          CertificateSerialNumber,
```

```
  signature              AlgorithmIdentifier,
  issuer                 Name,
  validity               Validity,
  subject                Name,
  subjectPublicKeyInfo   SubjectPublicKeyInfo,
  issuerUniqueID  [1]    IMPLICIT UniqueIdentifier
                         OPTIONAL,
  subjectUniqueID [2]    IMPLICIT UniqueIdentifier
                         OPTIONAL,
  extensions      [3]    EXPLICIT Extensions OPTIONAL
}
```

The fields and structures which are influenced by the integration of ECC in a X.509v3 certificate are the `signatureValue` in the `Certificate` structure and its corresponding `AlgorithmIdentifier` in the Certificate and `TBSCertificate` structure as well as the subject's public key in the `TBSCertificate` structure (`subjectPublicKeyInfo`). The following subsections focus on theses elements and provide additional and detailed information.

3.1 ECC Public Keys in X.509v3 Certificates

To insert an ECC public key into an X.509v3 certificate the `SubjectPublicKeyInfo` part has to be reviewed in more detail:

```
SubjectPublicKeyInfo ::= SEQUENCE
{
  algorithm         AlgorithmIdentifier
  subjectPublicKey  BIT STRING
}
```

A public key defined by the `SubjectPublicKeyInfo` structure is composed of two entries. First, the algorithm which is used with the enclosed public key is specified and second the public key as a string of bits is holt in the `subjectPublicKey` element. As already mentioned in section 2 an ECC public key is a point on an elliptic curve and is given by its coordinates. This public key can be encoded in compressed or uncompressed form. In the compressed case the y-coordinate is only represented with one octet which holds the sign information while in the uncompressed case both coordinates are given in total. [ANS99] defines in section 4.3.6 (and [IET02a] in section 2.3.5) the rules how these two representations are encoded in an `ECPoint` structure. It is built as an octet string. In both cases the result of the `ECPoint` is written in the `BIT STRING` therefore finally it consists of the concatenation of these two values.

The first element in the `SubjectPublicKeyInfo` structure specifies the cryptographic algorithm the public key is related to and is slightly more complex than the `subjectPublicKey`, since it is a data structure:

```
AlgorithmIdentifier ::= SEQUENCE
{
  algorithm   OBJECT IDENTIFIER,
```

```
  parameters  ANY DEFINED BY algorithm OPTIONAL
}
```

The specification of the algorithm expects an Object Identifier (OID) which identifies the underlying information object (here the cryptographic algorithm) unambiguously. In this way an unique Object Identifier Tree (OIT) is formed in which every information object with its unique OID is represented as a node. Starting from the root of the OIT the edges are numbered one by one. The edges starting from the root are reserved for the ITU-T (0), the ISO (1) and the union between ITU-T and ISO (2). With this method an object can be identified by following the numbered edges and can be written as a numbered path with dots. The ECDSA algorithm in X9.62 standardised as prime192v1 e.g. was unambiguously assigned to the OID 1.2.840.10045.3.1.1. Following the ISO part of the tree the number (2) connects the members in ISO and the next edge numbered (840) stands hence for United States of America. With (10045) the ANSI X9.62 Standard [ANS99] is reached. (3) describes the curves and is connected to the node prime by the following (1). The last number (1) in this chain concretes the prime curve version by specifying that prime192v1 is reached. Besides this shortening form of writing an object identifier the following label can also be used: {iso(1) member-body(2) us(840) ansi-x962(10045) curves(3) prime(1) prime192v1(1)}. All registered object identifiers can be found for instance in the OID-database provided at http://asn1.elibel.tm.fr/en/oid/. This representation in doted form is completed together with additional and optional parameters that are bounded to the algorithm.

If the OID in the `AlgorithmIdentifier` part is e.g. set to `ecPublicKey` (OID: 1.2.840.10045.2.1) than the algorithm-specific parameters must be filled with additional information on the underlying domain parameters. In general, `ecPublicKey` is used for ECDSA and ECDH keys. While the ECDSA algorithm is applied in the scope of digital signatures ECDH is used in the environment of key agreement. Specifications on how to represent parameters of an ECC public key are available in RFC 3279 [IET02a]:

```
EcpkParameters ::= CHOICE
{
  ecParameters  ECParameters,
  namedCurve    OBJECT IDENTIFIER,
  implicitlyCA  NULL
}
```

This choice defines three distinct ways to specify the respective elliptic curve domain parameters according to the underlying algorithm and exact one of those has to be chosen.

The elliptic curve parameters can be represented and shown in detail by choosing the field `ECParameters` [IET02a]:

```
ECParameters ::= SEQUENCE
{
  version   ECPVer,
  fieldID   FieldID,
  curve     Curve,
  base      ECPoint,
```

```
    order      INTEGER,
    cofactor   INTEGER OPTIONAL
  }
```

The `ECParameters` hold all information that is needed to specify the elliptic curve domain parameters which have already been introduced in chapter 2, such as the finite field, the curve that is specified by the two coefficients *a* and *b* and an optional seed. Furthermore the base point *G* is defined in an `OCTET STRING` representation and the order of *G* as well as the optional cofactor can be given by an integer value. The first four fields are data structures by themselves while the two last fields are described by simple data types.

Another possible solution can be retrieved by selecting the `namedCurve` field that works as a reference to a specific curve. This is again reached by an object identifier. Table 1 in section 2 shows selected standardised elliptic curves as well as their assigned OID.

The third possibility in the above stated choice can be applied if the additional parameters are inherited from a higher instance by using `implicitlyCA`. This could e.g. be realized over the issuer of the certificate and therefore this field is set to `NULL`.

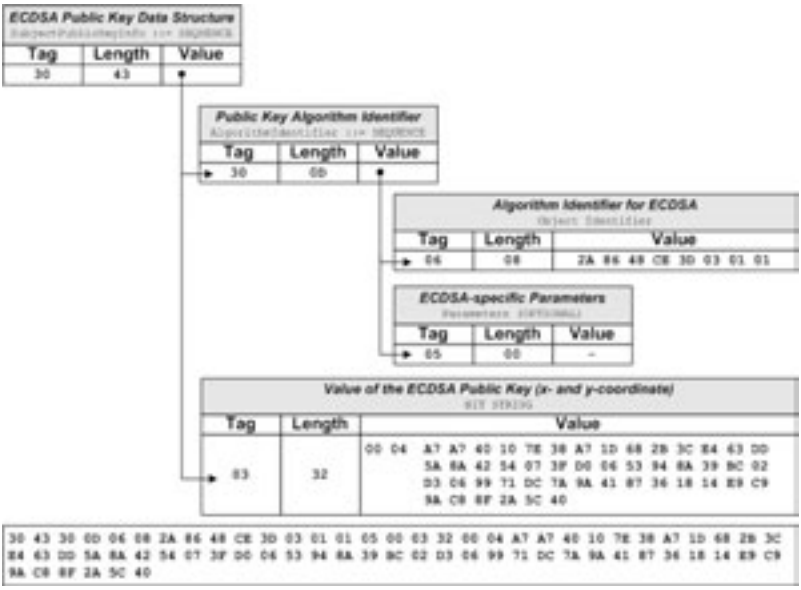

Figure 1: DER-encoded ECDSA public key

To illustrate how all these aspects fit together, the following example shows the structure and encoding of an ECDSA public key (see Figure 1) where the algorithm parameters are inherited by a higher CA in the hierarchical PKI. The encoding of X.509 related data structures are carried out using the Distinguished Encoding Rules (DER) specified in [ITU02b]. DER consists of three fields. The tag characterizes the data type while the length field states how many octets for the value will follow. The value field itself holds

the main information but can also be nested in further data structures. The DER-encoded octets are expressed as hexadecimal values.

Figure 1 shows a DER-encoded ECDSA public key in connection with the ASN.1 structured components and the final raw encoding. The encoding of the OID 1.2.840.10045.3.1.1 into a sequence of octets (here `2A 86 48 CE 3D 03 01 01`) has to be done according to section 8.19 in [ITU02b]. Further information about the encoding can also be found in [Go92].

Another important detail for encoding can be observed in the value part of the `BIT STRING` in Figure 1. The first byte in the value part claims how many bits in the last octet are unused (here zero) while the second byte states that the coordinates of the public key are given in uncompressed form. As mentioned before in section 2 this is a special feature in ECC and an advantage since it offers the possibility to generate certificates with less bytes. The ECC public key can be given in compressed form [ANS99] so that fewer octets are needed to specify the public key value.

3.2 ECC Signatures in X.509v3 Certificates and CRL

Besides including an ECC public key into a certificate the integration of ECC signatures in X.509v3 certificates has to be taken into account, which in this case holds for X.509v2 CRL in addition. As introduced before, two aspects have to be considered here: (1) the specification of the signature algorithm which can be found in the first layer of the certificate structure as well as in the `TBSCertificate` part and (2) the structure of the signature itself represented as a `BIT STRING` is of main interest.

The specification of the signature algorithm is equivalent to the specification of the ECC public key algorithm. Here again the data structure `AlgorithmIdenfier` as already introduced in chapter 3.1 is used to specify the algorithm which was used by the issuer to sign the certificate. So far the signature algorithm ECDSA together with the hash function SHA-1 is entered in RFC 3279 [IET02a] and ANSI X9.62 [ANS99]. In the case of ECDSA-with-sha1 its OID (1.2.840.10045.4.1) is assigned to the algorithm field of the `AlgorithmIdentifier` structure. [IET02a] states in chapter 2.2.3 that in the case of ECDSA-with-sha1 the parameters field must be left out and that the ECDSA signature is represented through the two values *r* and *s*. The ASN.1 representation in RFC 3279 consists of the following structure that holds the two values in a sequence of integer values:

```
Ecdsa-Sig-Value  ::=  SEQUENCE
{
  r  INTEGER,
  s  INTEGER
}
```

Figure 2 gives an overview of an ECDSA signature integrated in an X.509 certificate as specified in [IET02b] while the emphasis is laid on the `AlgorithmIdentifier` and the signature part.

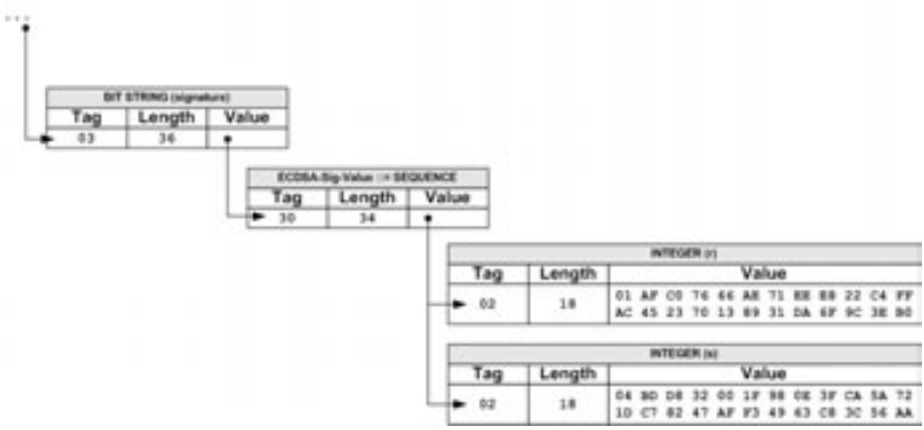

Figure 2: ASN.1 structure of an ECDSA signature

4. Exemplary Implementation

One recent governmental-driven example, where ECDSA is integrated, was set up by the Council of the European Union. In December 2004 the council passed a decree [CR04] to expand the passports and travel documents in the European Union with security features and biometrics. This project is often related to as the ePass. In Germany the BSI (Bundesamt für Sicherheit in der Informationstechnik, http://www.bsi.bund.de) has released a technical guideline [BSI06] that specifies requirements for the implementation. This document includes among other algorithms also specifications for the use of ECDSA with SHA 1, ECDSA with SHA 224, and ECDSA with SHA 256. The algorithms are applied in the scope of terminal authentication, so that only terminals with permission to read sensitive passport data are allowed to do so.

An industry-driven example is the usage of an ECC-based PKI in the energy industry environment. In the scope of the German liberalised energy market the project SELMA (Secure Electronic Measurement dAta exchange) [LRZ06] prosecutes the goal to secure transmit monetary measuring data over open communication channels between the system's stakeholders. The research and development project was funded from 2001 to 2005 by the German Federal Ministry of Economics and Employment (BMWA) and a lot of experiences were gathered while deploying it in a large-scale field-trial. Since the requirements to security were high and the communication over open networks had to be taken into account ECC was considered to satisfy the guidelines. Furthermore the property of shorter signature length turned out to be an important advantage. Additional information about the project SELMA can be found at http://www.selma.eu.

5. Conclusion

ECC, although mathematically more complex compared to well known and established cryptographic algorithms such as RSA and DSA, has revealed to be very efficient and advantageous in different system environments. The benefits of this cryptographic technology are well recognized, which can e.g. be seen by the adoption into national and international standards. Nevertheless, ECC can only be adjuvant if it is integrated into accompanying technologies such as X.509-based PKI. So far only a few ECC schemes are applied in this context and therefore it is not totally clear how these mechanisms can be correctly encoded. Notably first attempts are the ePass and SELMA projects. This paper explained the necessary enhancements to the X.509 data structures. It was shown that the integration if ECC is quite complex since different data structures needed to be accommodated.

References

[ANS99] ANSI X9.62: Public Key Cryptography for the Financial Services Industry: *The Elliptic Curve Digital Signature Algorithm (ECDSA)*, American Bankers Association, 1999.

[BSI06] Bundesamt für Sicherheit in der Informationstechnik: *Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC), Version 1.01*., 2006. Available online at: http://www.bsi.bund.de/fachthem/epass.

[CR04] Council Regulation (EC) No 2252/2004 of 13 December 2004 on standards for security features and biometrics in passports and travel documents issued by Member States. OJ L385, pp. 1-6, December 2004.

[FIP00] FIPS 186-2: *Digital Signature Standard*, Federal Information Processing Standards Publication 186-2, January 2000. Available online at: http://csrc.nist.gov/publications/fips/index.html.

[Go92] Gora, W.: *ASN.1: Abstract Syntax Notation One*, DATACOM, 1992.

[HMV04] Hankerson, D., Menezes, A. J., Vanstone, S.: *Guide to Elliptic Curve Cryptography*, Springer, 2004.

[IEEE01] IEEE P1363: *Standard Specifications for Public-Key Cryptography*, Institute of Electrical and Electronics Engineers, 2001.

[IET02a] IETF RFC 3279: *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, IETF, April 2002. Available online at: http://www.ietf.org/rfc/rfc3279.txt.

[IET02b] IETF RFC 3280: *Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile*, IETF, April 2002. Available online at: http://www.ietf.org/rfc/rfc3280.txt.

[ISO02a] ISO/IEC 15946-1: *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General*, International Organization for Standardization, Geneva, 2002.

[ISO02b] ISO/IEC 15946-2: *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital Signatures*, International Organization for Standardization, Geneva, 2002.

[ISO02c] ISO/IEC 15946-3: *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 3: Key establishment*, International Organization for Standardization, Geneva, 2002.

[ISO05] ISO/IEC FDIS 15946-4: *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 4: Digital signatures giving message recovery*; International Organization for Standardization, Geneva, 2005.

[ITU01] ITU-T X.509 RECOMMENDATION: *Information Technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*, International Telecommunication Union, May 2001. Also published as ISO/IEC 9594-8.

[ITU02a] ITU-T X.680 RECOMMENDATION: *Information Technology – Abstract Syntax Notation One (ASN.1): Specification of Basic Notation*, International Telecommunication Union, July 2002. Also published as ISO/IEC 8824-1.

[ITU02b] ITU-T X.690 RECOMMENDATION: *Information Technology – ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, International Telecommunication Union, July 2002. Also published as ISO/IEC 8825-1.

[Ko87] Koblitz, N.: *Elliptic curve cryptosystems*, Math. Comp. 48, pp. 203-209, 1987.

[LRZ06] Lo Iacono, L., Ruland, C., Zisky, N.: *Secure transfer of measurement data in open systems*, In (Richter, D., Editor), Computer Standards & Interfaces, Vol. 28:3, pp. 311-326, Elsevier, January 2006.

[Mi86] Miller, V.: *Uses of elliptic curves modulo large primes*, Advances in Cryptology – Crypto'85, Springer, pp. 417-426, 1986.

[SEC00a] SECG WORKING GROUP: *Standards for Efficient Cryptography – SEC 1: Elliptic Curve Cryptography*, SECG, September 2000. Available online at: http://www.secg.org.

[SEC00b] SECG WORKING GROUP: *Standards for Efficient Cryptography – SEC 2: Recommended Elliptic Curve Domain Parameters*, SECG, September 2000. Available online at: http://www.secg.org.

[SEC99] SECG X.509 Working Group: *ECC in X.509*, SECG, August 1999. Available online at: http://www.secg.org.