INTERNATIONAL STANDARD

**ISO/IEC 15946-1**

Third edition
2016-07-01

# Information technology — Security techniques — Cryptographic techniques based on elliptic curves —

## Part 1: General

*Technologies de l'information — Techniques de sécurité — Techniques cryptographiques basées sur les courbes elliptiques —*

*Partie 1: Généralités*



Reference number
ISO/IEC 15946-1:2016(E)



ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

 

# Contents

Page

# Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword — Supplementary information.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *Security techniques*.

This third edition cancels and replaces the second edition (ISO/IEC 15946-1:2008 with ISO/IEC 15946-1/Cor 1:2009), which has been technically revised.

ISO/IEC 15946 consists of the following parts, under the general title *Information technology — Security techniques — Cryptographic techniques based on elliptic curves*:

— *Part 1: General*

— *Part 5: Elliptic curve generation*



 

# Introduction

Cryptosystems based on elliptic curves defined over finite fields provide an interesting alternative to the RSA cryptosystem and to finite field discrete log based cryptosystems. The concept of an elliptic curve based public-key cryptosystem is simple.

— Every elliptic curve over a finite field is endowed with an addition operation "+" under which it forms a finite abelian group.

— The group law on elliptic curves extends in a natural way to a "discrete exponentiation" on the point group of the elliptic curve.

— Based on the discrete exponentiation on an elliptic curve, one can easily derive elliptic curve analogues of the well-known public-key schemes of the Diffie–Hellman and ElGamal type.

The security of such a public-key cryptosystem depends on the difficulty of determining discrete logarithms in the group of points of an elliptic curve. This problem is, with current knowledge, much harder for a given parameter size than the factorisation of integers or the computation of discrete logarithms in a finite field. Indeed, since Miller and Koblitz independently suggested the use of elliptic curves for public-key cryptographic systems in 1985, the elliptic curve discrete logarithm problem has only been shown to be solvable in certain specific, and easily recognisable, cases. There has been no substantial progress in finding a method for solving the elliptic curve discrete logarithm problem on arbitrary elliptic curves. Thus, it is possible for elliptic curve based public-key systems to use much shorter parameters than the RSA system or the classical discrete logarithm based systems that make use of the multiplicative group of some finite field. This yields significantly shorter digital signatures and system parameters and the integers to be handled by a cryptosystem are much smaller.

This part of ISO/IEC 15946 describes the mathematical background and general techniques necessary for implementing the elliptic curve cryptography mechanisms defined in ISO/IEC 15946-5, ISO/IEC 9796-3, ISO/IEC 11770-3, ISO/IEC 14888-3, ISO/IEC 18033-2 and other ISO/IEC standards.

It is the purpose of this part of ISO/IEC 15946 to meet the increasing interest in elliptic curve based public-key technology and to describe the components that are necessary to implement secure elliptic curve cryptosystems such as key-exchange, key-transport and digital signatures.

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this part of ISO/IEC 15946 may involve the use of patents.

The ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured the ISO and IEC that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC. Information may be obtained from:

Certicom Corp. Address: 4701 Tahoe Blvd., Building A, Mississauga, ON L4W0B5, Canada

Matsushita Electric Industrial Co., Ltd. Address: 1006, Kadoma, Kadoma City, Osaka, 571-8501, Japan

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and/or IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

# Information technology — Security techniques — Cryptographic techniques based on elliptic curves —

# Part 1: General

## 1 Scope

This part of ISO/IEC 15946 describes the mathematical background and general techniques necessary for implementing the elliptic curve cryptography mechanisms defined in ISO/IEC 15946-5, ISO/IEC 9796-3, ISO/IEC 11770-3, ISO/IEC 14888-3, ISO/IEC 18033-2 and other ISO/IEC standards.

This part of ISO/IEC 15946 does not specify the implementation of the techniques it defines. For example, it does not specify the basis representation to be used when the elliptic curve is defined over a finite field of characteristic two. Thus, interoperability of products complying with this part of ISO/IEC 15946 will not be guaranteed.

## 2 Normative references

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

ISO/IEC 15946-5, *Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 5: Elliptic curve generation*

## 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

**3.1**
**abelian group**
group $(S, *)$ such that $a*b = b*a$ for every $a$ and $b$ in $S$

**3.2**
**cubic curve**
set of solutions, made up of pairs of elements of a specified field known as points, to a cubic equation of special form

**3.3**
**elliptic curve**
cubic curve $E$ without a singular point

Note 1 to entry: The set of points $E$ together with an appropriately defined operation (see 6.2) forms an abelian group. The field that includes all coefficients of the equation describing $E$ is called the definition field of $E$. In this part of ISO/IEC 15946, only finite fields $F$ are dealt with as the definition field. When it is necessary to describe the definition field $F$ of $E$ explicitly, the curve is denoted as $E/F$.

Note 2 to entry: The form of a cubic curve equation used to define an elliptic curve varies depending on the field. The general form of an appropriate cubic equation for all possible finite fields is defined in 6.1.

Note 3 to entry: A definition of a cubic curve is given in Reference [15].

**3.4**
**field**
set of elements $S$ and a pair of operations $(+,*)$ defined on $S$ such that: (i) $a*(b + c) = a*b + a*c$ for every $a$, $b$ and $c$ in $S$, (ii) $S$ together with + forms an abelian group (with identity element 0), and (iii) $S$ excluding 0 together with $*$ forms an abelian group

**3.5**
**finite field**
field containing a finite number of elements

Note 1 to entry: For any positive integer $m$ and a prime $p$, there exists a finite field containing exactly $p^m$ elements. This field is unique up to isomorphism and is denoted by $F(p^m)$, where $p$ is called the characteristic of $F(p^m)$.

**3.6**
**group**
set of elements $S$ and an operation $*$ defined on the set of elements such that (i) $a*(b*c) = (a*b)*c$ for every $a$, $b$ and $c$ in $S$, (ii) there exists an identity element $e$ in $S$ such that $a*e = e*a = a$ for every $a$ in $S$, and (iii) for every $a$ in $S$ there exists an inverse element $a^{-1}$ in $S$ such that $a*a^{-1} = a^{-1}*a = e$

**3.7**
**cryptographic bilinear map**
map satisfying the non-degeneracy, bilinearity, and computability conditions

Note 1 to entry: Definitions of non-degeneracy, bilinearity and computability are provided in 6.4.

**3.8**
**singular point**
point at which a given mathematical object is not defined

## 4 Symbols

| | |
|---|---|
| $B$ | smallest integer such that $n$ divides $q^B$-1 |
| $d$ | private key of a user ($d$ is a random integer in the set [2, $n$-2]) |
| $E$ | elliptic curve, given by an equation of the form $Y^2 = X^3 + aX + b$ over the field $F(p^m)$ for $P > 3$, by an equation of the form $Y^2 + XY = X^3 + aX^2 + b$ over the field $F(2^m)$, or by an equation of the form $Y^2 = X^3 + aX^2 + b$ over the field $F(3^m)$, together with an extra point $O_E$ referred to as the point at infinity; the curve is denoted by $E/F(p^m)$, $E/F(2^m)$, or $E/F(3^m)$, respectively |
| $E(F(q))$ | set of $F(q)$-valued points of $E$ together with $O_E$ |
| $\#E(F(q))$ | order (or cardinality) of $E(F(q))$ |
| $E[n]$ | $n$-torsion group of $E$, that is $\{Q \in E \mid nQ = O_E\}$ |
| $e_n$ | cryptographic bilinear map |
| $\|F\|$ | number of elements in $F$ |
| $F(q)$ | finite field consisting of exactly $q$ elements; this includes the cases of $F(p)$, $F(2^m)$, and $F(p^m)$ |
| $F(q)^*$ | $F(q)\backslash\{0_F\}$ |
| $G$ | base point on $E$ with prime order $n$ |
| $<G>$ | group generated by $G$ with prime cardinality $n$ |
| $h$ | cofactor of $E(F(q))$ |

| | |
|---|---|
| $kQ$ | $k$th multiple of some point $Q$ of $E$, *i.e.* $kQ = Q + ...+ Q$ ($k$ summands) if $k > 0$, $kQ = (-k)(-Q)$, if $k < 0$, and $kQ = O_E$ if $k = 0$ |
| $\mu_n$ | cyclic group of order $n$ comprised of the $n$th roots of unity in the algebraic closure of $F(q)$ |
| $n$ | prime divisor of $\#E(F(q))$ |
| $O_E$ | elliptic curve point at infinity |
| $p$ | prime number |
| $P$ | public key of a user ($P$ is an elliptic curve point in $<G>$) |
| $q$ | prime power $p^m$ for some prime $p$ and some integer $m \geq 1$ |
| $Q$ | point on $E$ with coordinates $(x_Q, y_Q)$ |
| $Q_1+Q_2$ | elliptic curve sum of two points $Q_1$ and $Q_2$ |
| $x_Q$ | $x$-coordinate of $Q \neq O_E$ |
| $y_Q$ | $y$-coordinate of $Q \neq O_E$ |
| $[0, k]$ | set of integers from 0 to $k$ inclusive |
| $0_F$ | identity element of $F(q)$ for addition |
| $1_F$ | identity element of $F(q)$ for multiplication |

# 5 Conventions for fields

## 5.1 Finite prime fields $F(p)$

For any prime $p$, there exists a finite field consisting of exactly $p$ elements. This field is uniquely determined up to isomorphism and in this part of ISO/IEC 15946 it is referred to as the finite prime field $F(p)$.

The elements of a finite prime field $F(p)$ may be identified with the set $[0, p - 1]$ of all non-negative integers less than $p$. $F(p)$ is endowed with two operations called addition and multiplication such that the following conditions hold:

— $F(p)$ is an abelian group with respect to the addition operation "+".

For $a, b \in F(p)$ the sum $a + b$ is given as $a + b := r$, where $r \in F(p)$ is the remainder obtained when the integer sum $a + b$ is divided by $p$.

— $F(p)\backslash\{0\}$ denoted as $F(p)^*$ is an abelian group with respect to the multiplication operation "×".

For $a, b \in F(p)$ the product $a \times b$ is given as $a \times b := r$, where $r \in F(p)$ is the remainder obtained when the integer product $a \times b$ is divided by $p$. When it does not cause confusion, × is omitted and the notation $ab$ is used or the notation $a \cdot b$ is used.

## 5.2 Finite fields $F(p^m)$

For any positive integer $m$ and prime $p$, there exists a finite field of exactly $p^m$ elements. This field is unique up to isomorphism and in this part of ISO/IEC 15946 it is referred to as the finite field $F(p^m)$.

NOTE 1 $F(p^m)$ is the general definition including $F(p)$ for $m = 1$ and $F(2^m)$ for $p = 2$.

NOTE 2 If $p = 2$, then field elements may be identified with bit strings of length $m$ and the sum of two field elements is the bit-wise XOR of the two bit strings.

The finite field $F(p^m)$ may be identified with the set of $p$-ary strings of length $m$ in the following way. Every finite field $F(p^m)$ contains at least one basis $\{\xi_1, \xi_2, ..., \xi_m\}$ over $F(p)$ such that every element $\alpha \in F(p^m)$ has a unique representation of the form $\alpha = a_1\xi_1 + a_2\xi_2 + ... + a_m\xi_m$ , with $a_i \in F(p)$ for $i = 1, 2,\cdots, m$. The element $\alpha$ can then be identified with the $p$-ary string $(a_1, a_2,\cdots, a_m)$. The choice of basis is beyond the scope of this part of ISO/IEC 15946. $F(p^m)$ is endowed with two operations called addition and multiplication such that the following conditions hold:

— $F(p^m)$ is an abelian group with respect to the addition operation "+".

For $\alpha = (a_1, a_2,\cdots, a_m)$ and $\beta = (b_1, b_2,\cdots, b_m)$, the sum $\alpha + \beta$ is given by $\alpha + \beta := \gamma = (c_1, c_2,\cdots, c_m)$, where $c_i = a_i + b_i$ is the sum in $F(p)$. The identity element for addition is $0_F = (0, ..., 0)$.

— $F(p^m)\backslash\{0\}$, denoted by $F(p^m)^*$, is an abelian group with respect to the multiplication operation "×".

For $\alpha = (a_1, a_2,\cdots, a_m)$ and $\beta = (b_1, b_2,\cdots, b_m)$ the product $\alpha \times \beta$ is given by a $p$-ary string $\alpha \times \beta := \gamma = (c_1, c_2,\cdots, c_m)$, where $c_i = \sum_{1 \le j,k \le m} a_j\, b_k d_{i,j,k}$ for $\xi_j\xi_k = d_{1,j,k}\xi_1 + d_{2,j,k}\,\xi_2 + ... + d_{m,j,k}\xi_m$ $(1 \le j,\ k \le m)$. When it does not cause confusion, × is omitted and the notation $ab$ is used. The basis can be chosen in such a way that the identity element for multiplication is $1_F = (1, 0, ..., 0)$.

NOTE 3 The choice of basis is described in Reference [4].

# 6 Conventions for elliptic curves

## 6.1 Definitions of elliptic curves

### 6.1.1 Elliptic curves over $F(p^m)$

Let $F(p^m)$ be a finite field with a prime $P > 3$ and a positive integer $m$. In this part of ISO/IEC 15946, it is assumed that $E$ is described by a "short (affine) Weierstrass equation", that is an equation of type

$$Y^2 = X^3 + aX + b \qquad \text{with } a, b \in F(p^m)$$

such that $4a^3 + 27b^2 \neq 0_F$ holds in $F(p^m)$.

NOTE The above curve with $4a^3 + 27b^2 = 0_F$ is called a singular curve, which is not an elliptic curve.

The set of $F(p^m)$-valued points of $E$ is given by Formula (1):

$$E(F(p^m)) = \{Q = (x_Q, y_Q) \in F(p^m) \times F(p^m) | y_Q{}^2 = x_Q{}^3 + ax_Q + b\} \cup \{O_E\} \qquad (1)$$

where $O_E$ is an extra point referred to as the point at infinity of $E$.

### 6.1.2 Elliptic curves over $F(2^m)$

Let $F(2^m)$, for some $m \ge 1$, be a finite field. In this part of ISO/IEC 15946, it is assumed that $E$ is described by an equation of the type

$$Y^2 + XY = X^3 + aX^2 + b \qquad \text{with } a, b \in F(2^m)$$

such that $b \neq 0_F$ holds in $F(2^m)$.

For cryptographic use, $m$ shall be a prime to prevent certain kinds of attacks on the cryptosystem.

NOTE The above curve with $b = 0_F$ is called a singular curve, which is not an elliptic curve.



 

The set of $F(2^m)$-valued points of $E$ is given by Formula (2):

$$E(F(2^m)) = \{Q = (x_Q, y_Q) \in F(2^m) \times F(2^m) | y_Q^2 + x_Q y_Q = x_Q^3 + a x_Q^2 + b\} \cup \{O_E\} \quad (2)$$

where $O_E$ is an extra point referred to as the point at infinity of $E$.

### 6.1.3 Elliptic curves over $F(3^m)$

Let $F(3^m)$ be a finite field with a positive integer $m$. In this part of ISO/IEC 15946, it is assumed that $E$ is described by an equation of the type

$$Y^2 = X^3 + aX^2 + b \qquad \text{with } a, b \in F(3^m)$$

such that $a, b \neq 0_F$ holds in $F(3^m)$.

NOTE The above curve with $a$ or $b = 0_F$ is called a singular curve, which is not an elliptic curve.

The set of $F(3^m)$-valued points of $E$ is given by Formula (3):

$$E(F(3^m)) = \{Q = (x_Q, y_Q) \in F(3^m) \times F(3^m) | y_Q^2 = x_Q^3 + a x_Q^2 + b\} \cup \{O_E\} \quad (3)$$

where $O_E$ is an extra point referred to as the point at infinity of $E$.

## 6.2 Group law on elliptic curves

Elliptic curves are endowed with the addition operation +: $E \times E \rightarrow E$, defining for each pair $(Q_1, Q_2)$ of points on $E$ a third point $Q_1 + Q_2$. With respect to this addition, $E$ is an abelian group with identity element $O_E$. The $k$th multiple of $Q$ is given as $kQ$, where $kQ = Q + \ldots + Q$ ($k$ summands) if $k > 0$, $kQ = (-k)(-Q)$ if $k < 0$, and $kQ = O_E$ if $k = 0$. The smallest positive $k$ with $kQ = O_E$ is called the order of $Q$.

NOTE Formulae of the group law and $Q$ are given in B.3, B.4, and B.5.

## 6.3 Generation of elliptic curves

In order to use an elliptic curve for a cryptosystem, it is necessary to generate an appropriate elliptic curve. ISO/IEC 15946-5 shall be referred to for methods of generation of elliptic curves.

## 6.4 Cryptographic bilinear map

A cryptographic bilinear map $e_n$ is used in some cryptographic applications such as signature schemes or key agreement schemes. A cryptographic bilinear map $e_n$ is realized by restricting the domain of the Weil or Tate pairings as follows.

$$e_n: <G_1> \times <G_2> \rightarrow \mu_n$$

where the cryptographic bilinear map $e_n$ satisfies the following properties:

— bilinearity: $e_n(aG_1, bG_2) = e(G_1, G_2)^{ab}$ ($\forall a,b \in [0, n\text{-}1]$);

— non-degeneracy: $e_n(G_1, G_2) \neq 1$;

— computability: There exists an efficient algorithm to compute $e_n$.

NOTE 1 The relation between the cryptographic bilinear map and the Weil or Tate pairing is given in B.7.

NOTE 2 Formulae for the Weil and Tate pairings are given in C.6.

NOTE 3 There are two types of pairings:

— the case $G_1 = G_2$;

— the case $G_1 \neq G_2$.

# 7 Conversion functions

## 7.1 Octet string/bit string conversion: OS2BSP and BS2OSP

Primitives OS2BSP and BS2OSP to convert between octet strings and bit strings are defined as follows:

— The function OS2BSP($x$) takes as input an octet string $x$, interprets it as a bit string $y$ and outputs the bit string $y$. Set the first bit of the bit string to the most significant (leftmost) bit of the first octet, the second bit to the next most significant bit of the first octet, continue in the same way, and finally set the last bit to the least significant (rightmost) bit of the last octet.

— The function BS2OSP($y$) takes as input a bit string $y$, whose length is a multiple of 8, and outputs the unique octet string $x$ such that $y$ = OS2BSP($x$).

## 7.2 Bit string/integer conversion: BS2IP and I2BSP

Primitives BS2IP and I2BSP to convert between bit strings and integers are defined as follows:

— The function BS2IP($x$) maps a bit string $x$ to an integer value $x'$, as follows:

If $x = \langle x_{l-1}, \ldots, x_0 \rangle$, where $x_0, \ldots, x_{l-1}$ are bits, then the value $x'$ is defined as $x' = \sum_{0 \leq i < l,\, xi = \text{'1'}} 2^i$.

— The function I2BSP($m$, $l$ ) takes as input two non-negative integers, $m$ and $l$, and outputs the unique bit string $x$ of length $l$, such that BS2IP($x$) = $m$, if such an $x$ exists. Otherwise, the function outputs an error message.

The length in bits of a non-negative integer $m$ is the number of bits in its binary representation, i.e. $[\log_2(m + 1)]$. As a notational convenience, Oct($m$) is defined as Oct($m$) = I2BSP($m$, 8).

NOTE I2BSP($m$, $l$) fails if, and only if, the length of $m$ in bits is greater than $l$.

## 7.3 Octet string/string conversion: OS2IP and I2OSP

Primitives OS2IP and I2OSP to convert between octet strings and integers are defined as follows:

— The function OS2IP($x$) takes as input an octet string $x$, and outputs the integer BS2IP[OS2BSP($x$)].

— The function I2OSP($m$, $l$) takes as input two non-negative integers, $m$ and $l$, and outputs the unique octet string $x$ of length $l$ in octets, such that OS2IP($x$) = $m$, if such an $x$ exists. Otherwise, the function outputs an error message.

The length in octets of a non-negative integer $m$ is the number of digits in its representation base 256, i.e. $[\log_{256}(m + 1)]$.

NOTE 1 I2OSP($m$, $l$) fails if, and only if, the length of $m$ in octets is greater than $l$.

NOTE 2 An octet $x$ is often written in its hexadecimal format of length 2; when OS2IP($x$) < 16, "0", representing the bit string 0000, is prepended. For example, an integer 15 is written as 0f in its hexadecimal format.

NOTE 3 The length in octets of a non-negative integer $m$ is denoted by $L(m)$.



 

## 7.4 Finite field element/integer conversion: FE2IP$_F$

The primitive FE2IP$_F$ to convert elements of $F$ to integer values is defined as follows:

— The function FE2IP$_F$ maps an element $a \in F$ to an integer value $a'$, as follows:

If an element $a$ of $F$ is identified with an $m$-tuple $(a_1, \ldots, a_m)$, where the cardinality of $F$ is $q = p^m$ and $a_i \in [0, p - 1]$ for $1 \le i \le m$, then the value $a'$ is defined as $a' = \sum_{1 \le i \le m} a_i\, p^{i-1}$.

## 7.5 Octet string/finite field element conversion: OS2FEP$_F$ and FE2OSP$_F$

Primitives OS2FEP$_F$ and FE2OSP$_F$ to convert between octet strings and elements of an explicitly given finite field $F$ are defined as follows:

— The function FE2OSP$_F$ ($a$) takes as input an element $a$ of the field $F$ and outputs the octet string I2OSP($a'$, $l$), where $a'$ = FE2IP$_F$ ($a$) and $l = L(|F|-1)$. Thus, the output of FE2OSP$_F$ ($a$) is always an octet string of length exactly $[\log_{256}|F|]$.

NOTE 1 $L(x)$ represents the length in octets of integer $x$ or octet string $x$ (non-negative integer).

— The function OS2FEP$_F$ ($x$) takes as input an octet string $x$, and outputs the (unique) field element $a \in F$, such that FE2OSP$_F$ ($a$) = $x$, if such an $a$ exists, and otherwise fails.

NOTE 2 OS2FEP$_F$ ($x$) fails if and only if either $x$ does not have length exactly $[\log_{256}|F|]$, or OS2IP($x$) $\ge |F|$.

## 7.6 Elliptic curve point/octet string conversion: EC2OSP$_E$ and OS2ECP$_E$

### 7.6.1 Compressed elliptic curve points

Let $E$ be an elliptic curve over an explicitly given finite field $F$, where $F$ has characteristic $p$. A point $P \ne O_E$ can be represented in either `compressed`, `uncompressed`, or `hybrid` form. If $P = (x, y)$, then $(x, y)$ is the uncompressed form of $P$. The compressed form of $P$ is the pair $(x, \tilde{y})$, where $\tilde{y} \in \{0, 1\}$ and is determined as follows:

— if $p \ne 2$ and $y = 0_F$, then $\tilde{y} = 0$;

— if $p \ne 2$ and $y \ne 0_F$, then $\tilde{y} = [(y'/p^f) \bmod p] \bmod 2$, where $y'$ = FE2IP$_F$($y$), and where $f$ is the largest non-negative integer such that $p^f | y'$;

NOTE 1 If $p \ne 2$ and $y = (y_1, \ldots, y_m) \ne 0_F$, this is equivalent to letting $j$ be the smallest index with $y_j \ne 0$ and defining $\tilde{y} = y_j \bmod 2$.

— if $p = 2$ and $x = 0_F$, then $\tilde{y} = 0$;

— if $p = 2$ and $x \ne 0_F$, then $\tilde{y} = [z'/2^f] \bmod 2$, where $z = y/x$, where $z'$ = FE2IP$_F$($z$), and where $f$ is the largest non-negative integer such that $2^f$ divides FE2IP$_F$($1_F$).

NOTE 2 If $p = 2$ and $x \ne 0$, this is equivalent to letting $y/x = (z_1, \ldots, z_m)$ and defining $\tilde{y} = z_1$.

The hybrid form of $P = (x, y)$ is the triple $(x, \tilde{y}, y)$, where $\tilde{y}$ is as in the previous paragraph.

### 7.6.2 Point decompression algorithms

There exist efficient procedures for point decompression, i.e. computing $y$ from $(x, \tilde{y})$. These are briefly described here.

— If $p \ne 2$, then let $(x, \tilde{y})$ be the compressed form of $(x, y)$ where the point $(x, y)$ satisfies the Weierstrass equation $y^2 = f(x)$ defined in 6.1.1 or 6.1.3. If $f(x) = 0_F$, then there is only one possible choice for $y$, namely, $y = 0_F$. Otherwise, if $f(x) \ne 0_F$, then there are two possible choices of $y$, which differ only in sign, and the correct choice is determined by $\tilde{y}$. There are well-known algorithms for computing square roots in finite fields, and so the two choices of $y$ are easily computed.

— If $p = 2$, then let $(x, \tilde{y})$ be the compressed form of $(x, y)$ where the point $(x, y)$ satisfies the equation $y^2 + xy = x^3 + ax^2 + b$. If $x = 0_F$, then the equation is $y^2 = b$, from which $y$ is uniquely determined and easily computed. Otherwise, if $x \neq 0_F$, then setting $z = y/x$, the equation is $z^2 + z = g(x)$, where $g(x) = x + a + bx^{-2}$. The value of $y$ is uniquely determined by, and easily computed from, the values $z$ and $x$, and so it suffices to compute $z$. To compute $z$, observe that for a fixed $x$, if $z$ is one solution to the equation $z^2 + z = g(x)$, then there is exactly one other solution, namely $z + 1_F$. It is easy to compute these two candidate values of $z$, and the correct choice of $z$ is easily seen to be determined by $\tilde{y}$.

### 7.6.3 Conversion functions

Let $E$ be an elliptic curve over an explicitly given finite field $F$.

Primitives $\text{EC2OSP}_E$ and $\text{OS2ECP}_E$ for converting between points on an elliptic curve $E$ and octet strings are defined as follows:

— The function $\text{EC2OSP}_E$ ($P$, fmt) takes as input a point $P$ on $E$ and a format specifier fmt, which is one of the symbolic values `compressed`, `uncompressed`, or `hybrid`. The output is an octet string, EP, computed as follows:

  — if $P = O_E$, then EP = Oct(0);

  — if $P = (x, y) \neq O_E$, with compressed form $(x, \tilde{y})$, then EP = $H||X||Y$, where

    — $H$ is a single octet of the form $\text{Oct}[4U + C(2 + \tilde{y})]$, where

    — $U = 1$ if fmt is either `uncompressed` or `hybrid`, and otherwise, $U = 0$,

    — $C = 1$ if fmt is either `compressed` or `hybrid`, and otherwise, $C = 0$;

    — $X$ is the octet string $\text{FE2OSP}_F$ ($x$);

    — $Y$ is the octet string $\text{FE2OSP}_F$ ($y$) if fmt is either `uncompressed` or `hybrid`, and otherwise $Y$ is the null octet string.

— The function $\text{OS2ECP}_E$ (EP) takes as input an octet string EP. If there exists a point $P$ on the curve $E$ and a format specifier fmt, such that $\text{EC2OSP}_E$ ($P$, fmt) = EP, then the function outputs $P$ (in uncompressed form), and otherwise, the function fails. Note that the point $P$, if it exists, is uniquely defined, and so the function $\text{OS2ECP}_E$(EP) is well defined.

NOTE If the format specifier fmt is uncompressed, then both $x$ and $y$ are used; and the value $\tilde{y}$ need not be computed.

## 7.7 Integer/elliptic curve conversion: I2ECP

Let $E$ be an elliptic curve over an explicitly given finite field $F$. Primitive I2ECP to convert from integers to elliptic curve points is defined as follows:

a) The function I2ECP($x$) takes as input an integer $x$.

b) Convert the integer $x$ to an octet string $X = \text{I2OSP}[x, L(|F|-1)]$.

c) If there exists a point $P$ on the curve $E$ such that $\text{EC2OSP}_E(P,$ `compressed`$) = 03||X$, then the function outputs $P$, and otherwise, the function fails.

NOTE 1 The output of point $P$, if it exists, is uniquely defined.

NOTE 2 The function I2ECP will fail on input $x$ if there does not exist a point $P$ on the curve $E$ such that $\text{EC2OSP}_E$ ($P$, `compressed`) = $03||X$.

NOTE 3 The range of the I2ECP is approximately half of $E(F)$. That is, the I2ECP always outputs elliptic curve points $P = (x, y)$ with compressed form $(x, 1)$. It will not output either the point at infinity or an elliptic curve point $P = (x, y)$ with compressed form $(x, 0)$.



 

NOTE 4 Some applications based on elliptic curves may need a function which maps octet strings to elliptic curve points. The function I2ECP is used as a component together with OS2IP or a hash function.

# 8 Elliptic curve domain parameters and public key

## 8.1 Elliptic curve domain parameters over $F(q)$

Elliptic curve parameters over $F(q)$ [including the special cases $F(p)$ and $F(2^m)$] shall consist of the following:

If $m > 1$, there should be an agreement on the choice of the basis between the communicating parties.

— the field size $q = p^m$ which defines the underlying finite field $F(q)$, where $p$ shall be a prime number, and an indication of the basis used to represent the elements of the field in case $m > 1$;

— if $q = p^m$ with $P > 3$, two field elements $a$ and $b$ in $F(q)$ which define the equation of the elliptic curve

$E: y^2 = x^3 + ax + b$;

— if $q = 2^m$, two field elements $a$ and $b$ in $F(2^m)$ which define the equation of the elliptic curve

$E: y^2 + xy = x^3 + ax^2 + b$;

— if $q = 3^m$, two field elements $a$ and $b$ in $F(3^m)$ which define the equation of the elliptic curve

$E: y^2 = x^3 + ax^2 + b$;

— two field elements $x_G$ and $y_G$ in $F(q)$ which define a point $G = (x_G, y_G)$ of prime order on $E$;

— the order $n$ of the point $G$;

— the cofactor $h = \#E(F(q))/n$ (when required by the underlying scheme).

NOTE The computation of $\#E(F(q))$ is described in Reference [4].

## 8.2 Elliptic curve key generation

Given a valid set of elliptic curve domain parameters, a private key and corresponding public key may be generated as follows:

a) Select a random or pseudorandom integer $d$ in the set [2, $n$-2]. The integer $d$ shall be protected from unauthorised disclosure and be unpredictable.

b) Compute the point $P = (x_P, y_P) = dG$.

c) The key pair is $(P, d)$, where $P$ will be used as the public key and $d$ is the private key.

In some applications, the public key may be $eG$, where $de = 1 \bmod n$.

# Annex A
(informative)

# Background information on finite fields

## A.1 General

Annex A presents the information on finite fields that is necessary for the elliptic curve based public key schemes.

## A.2 Bit strings

A bit is either zero "0" or one "1". A bit string $x$ is a finite sequence $\langle x_{l-1}, \ldots, x_0 \rangle$ of bits $x_0, \ldots, x_{l-1}$. The length of a bit string $x$ is the number $l$ of bits in the string $x$. Given a non-negative integer $n$, $\{0, 1\}^n$ denotes the set of bit strings of length $n$. $\{0, 1\}^* = \cup_{n \geq 0} \{0, 1\}^n$ denotes the set of bit strings, including the null string (whose length is 0).

## A.3 Octet strings

An octet is a bit string of length 8. An octet string is a finite sequence of octets. The length of an octet string is the number of octets in the string. $\{0, 1\}^{8*}$ denotes the set of octet strings, including the null string (whose length is 0). An octet is often written in its hexadecimal format, using the range between 00 and FF.

## A.4 Characteristic of a finite field $F(p^m)$

The characteristic of a field is the smallest positive integer $c$ such that $c$ additions of $1_F$ give the zero element. If no such $c$ exists, the characteristic is 0. For any prime $p$, the characteristic of the field $F(p^m)$ is $p$.

## A.5 Inverting elements of a finite field $F(p^m)$

Let $a$ be an element of $F(p^m)$. Then there exists a unique element $b \in F(p^m)$ such that $a \cdot b = b \cdot a = 1_F$, and $b$ is called the multiplicative inverse of $a$, denoted by $a^{-1}$. If $a = \gamma^i$, then $a^{-1}$ can be computed as $a^{-1} = \gamma^{q-1-i}$.

NOTE If $m = 1$, $a^{-1}$ is given as $x$ in the equation $ax + py = 1$, which can be solved using the extended Euclidean algorithm.

## A.6 Squares and non-squares in a finite field $F(p^m)$

An element $a \in F(p^m)$ is called a square in $F(p^m)$, if there exists an element $b \in F(p^m)$, such that $a = b^2$. Whether or not $a \in F(p^m)$ is a square can be determined by making use of Formula (A.1):

$$a \text{ is a square in } F(p^m) \Leftrightarrow a^{(q-1)/2} = 1_F \qquad \text{(A.1)}$$

## A.7 Finding square-roots in $F(p^m)$

There are various methods for finding square roots in $F(p^m)$. That is, given $a \in F(p^m)$, where $a$ is a square, find $b \in F(p^m)$, such that $a = b^2$.

NOTE If $q = 3 \pmod 4$, then the square-root can be computed as $b = a^{(q+1)/4}$. The other cases are described in References [4] and [5].

# Annex B
(informative)

# Background information on elliptic curves

## B.1 General

Annex B presents the information on elliptic curves that is necessary for the elliptic curve based public key schemes.

## B.2 Properties of elliptic curves

An elliptic curve $E$ over $F(q)$ is endowed with a binary operation "+": $E \times E \rightarrow E$ assigning to any two points $Q_1$, $Q_2$ on $E$ a third point $Q_1 + Q_2$ on $E$. The elliptic curve $E$ is an abelian group with respect to "+".

The number of points of $E$ (including $O_E$) is called the order (or cardinality) of $E$ and is denoted by $\#E(F(q))$. The order satisfies the following theorem of Hasse:

$$q + 1 - 2\sqrt{q} \leq \#E(F(q)) \leq q + 1 + 2\sqrt{q}$$

The integer $t$ defined by $t = q + 1 - \#E(F(q))$ is called the *trace*. Hasse's theorem gives a bound on the trace. B.6 gives sufficient conditions that for a given $t$ in $[-2\sqrt{q}, 2\sqrt{q}]$, there is an elliptic curve $E$ over $F(q)$ with trace $t$.

**Anomalous and supersingular curves**

An elliptic curve $E$ defined over $F(q)$ with trace $t$ divisible by $p$ is called supersingular. An elliptic curve $E$ defined over $F(q)$ with $\#E(F(q)) = q$ is called anomalous. Supersingular curves are subject to the Frey-Rück[7] and Menezes-Okamoto-Vanstone[10] algorithms. Cryptsystems using anomalous curves are vulnerable to attacks using the Araki-Satoh[12], Semaev[13] and Smart[14] algorithms.

## B.3 Group law for elliptic curves $E$ over $F(q)$ with $P > 3$

### B.3.1 Overview of coordinates

An elliptic curve is generally defined in terms of affine coordinates. Therefore, the base point or a user public key is given in affine coordinates. The major drawback of affine coordinates is that they make heavy use of divisions in $F(q)$ for both addition and doubling. In most implementations of finite field arithmetic, field division is a very "expensive" operation and in such situations it can be prudent to avoid divisions as much as possible. This can be achieved by using other coordinates for the elliptic curve points such as projective, Jacobian, and modified Jacobian coordinates. All of the coordinate systems given for an elliptic curve are compatible.

### B.3.2 Group law in affine coordinates

Let $F(q)$ be a finite field with $P > 3$. Let $E$ be an elliptic curve over $F(q)$ given by the "short Weierstrass equation":

(Aff) $y^2 = x^3 + ax + b$ with $a, b \in F(q)$

where the inequality $4a^3 + 27b^2 \neq 0_F$ holds in $F(q)$.

NOTE More precisely, (Aff) is called the affine Weierstrass equation.

In affine coordinates, the group law on an elliptic curve given by (Aff) reads as follows:

— the point at infinity is the identity element $O_E$ with respect to "+";

— let $R = (x, y)$ be a point on $E$, such that $R \neq O_E$, then $-R = (x, -y)$;

— let $R_1 = (x_1, y_1)$ and $R_2 = (x_2, y_2)$ be two distinct points on $E$, such that $R_1 \neq \pm R_2$ and $R_1, R_2 \neq O_E$; the sum is the point $R_3 = (x_3, y_3)$ where:

$x_3 = r^2 - x_1 - x_2$;

$y_3 = r\,(x_1 - x_3\,) - y_1$;

with $r = (y_2 - y_1) / (x_2 - x_1)$;

— let $R = (x, y)$ be a point on $E$, such that $R \neq O_E$ and $y \neq 0_F$; its doubling is the point $2R = (x_3, y_3)$, where:

$x_3 = r^2 - 2x$;

$y_3 = r\,(x - x_3) - y$;

with $r = (3x^2 + a) / (2y)$.

In the case of $R = (x, 0_F)$, its doubling is $2R = O_E$.

### B.3.3 Group law in projective coordinates

NOTE 1 Using projective coordinates will result in more multiplications during the calculation of the group laws but no inversions have to be computed.[6]

NOTE 2 When using elliptic curves for cryptosystems, usually a transformation into affine coordinates has to be done at the end of the scalar multiplication. When converting projective into affine coordinates, 1 division is necessary.

The two-dimensional projective space over $F(q)$, $\Pi_{\text{proj}}(F(q))$, is given by equivalence classes of triplets $(X, Y, Z) \in F(q) \times F(q) \times F(q) \setminus \{(0_F, 0_F, 0_F)\}$, where two triplets $(X, Y, Z), (X', Y', Z') \in F(q) \times F(q) \times F(q) \setminus \{(0_F, 0_F, 0_F)\}$ are said to be equivalent, if there exists $\lambda \in F(q)^*$, such that $(X', Y', Z') = (\lambda X, \lambda Y, \lambda Z)$. The projective analogue of the short affine Weierstrass equation (Aff) is defined over $\Pi_{\text{proj}}(F(q))$ and given by the homogeneous cubic formula

(Proj) $Y^2Z = X^3 + aXZ^2 + bZ^3$ with $a, b \in F(q)$.

NOTE 3 The set of all triplets equivalent to $(X, Y, Z)$ is denoted by $(X, Y, Z)/\sim$.

The elliptic curve given in projective coordinates consists of all points $R = (X, Y, Z)$ of $F(q) \times F(q) \times F(q) \setminus \{(0_F, 0_F, 0_F)\}$, such that the triple $(X, Y, Z)$ is a solution of the equation (Proj), where by an abuse of notation $(X, Y, Z)$ is identified with the equivalence class $(X, Y, Z)/\sim$ containing $(X, Y, Z)$. There is a

relation between the points $Q$ of $E$ when the curve is given in affine coordinates and the points $R$ in projective coordinates. Indeed, the following conditions hold:

— if $Q = (X_Q, Y_Q)$ is an affine point of $E$, then $R = (X_Q, Y_Q, 1_F)$ is the corresponding point in projective coordinates;

— if $R = (X, Y, Z)$ (with $Z \neq 0_F$) is a solution of (Proj), then $Q = (X/Z, Y/Z)$ is the corresponding affine point of $E$;

— there is only one solution of (Proj) with $Z = 0$, namely the point $(0_F, 1_F, 0_F)$; this point corresponds to $O_E$.

In projective coordinates, the group law on an elliptic curve given by (Proj) reads as follows.

— The point $(0_F, 1_F, 0_F)$ is the identity element $O_E$ with respect to "+".

— Let $R = (X, Y, Z) \neq (0_F, 1_F, 0_F)$ be a point on $E$ given in projective coordinates; then $-R = (X, -Y, Z)$.

— Let $R_1 = (X_1, Y_1, Z_1)$ and $R_2 = (X_2, Y_2, Z_2)$ be two distinct points on $E$, such that $R_1 \neq R_2$ and $R_1, R_2 \neq (0_F, 1_F, 0_F)$ and denote the sum by $R_3 = (X_3, Y_3, Z_3)$. The coordinates $X_3$, $Y_3$ and $Z_3$ can be computed using the following formulae:

$X_3 = -su$;

$Y_3 = t\,(u + s^2X_1Z_2) - s^3Y_1Z_2$;

$Z_3 = s^3Z_1Z_2$;

with $s = X_2Z_1 - X_1Z_2$, $t = Y_2Z_1 - Y_1Z_2$, and $u = s^2\,(X_1Z_2 + X_2Z_1) - t^2Z_1Z_2$.

— Let $R = (X, Y, Z) \neq (0_F, 1_F, 0_F)$ be a point on $E$ and denote its doubling by $2R = (X_3, Y_3, Z_3)$. The coordinates $X_3$, $Y_3$ and $Z_3$ can be computed using the following formulae:

$X_3 = -su$;

$Y_3 = t\,(u + s^2X) - s^3Y$;

$Z_3 = s^3Z$;

with $t = 3X^2 + aZ^2$, $s = 2YZ$ and $u = 2s^2X - t^2Z$.

### B.3.4 Group law in Jacobian coordinates

NOTE 1 Using Jacobian coordinates will result in more multiplications during the calculation but no inversions have to be computed.[6]

The two-dimensional space over $F(q)$, $\Pi_{\text{jac}}(F(q))$, is given by equivalence classes of triplets $(X, Y, Z) \in F(q) \times F(q) \times F(q) \setminus \{(0_F, 0_F, 0_F)\}$, where two triplets $(X, Y, Z), (X', Y', Z') \in F(q) \times F(q) \times F(q) \setminus \{(0_F, 0_F, 0_F)\}$ are said to be equivalent if there exists $\lambda \in F(q)^*$, such that $(X', Y', Z') = (\lambda^2X, \lambda^3Y, \lambda Z)$. The Jacobian analogue of the short affine Weierstrass equation (Aff) is defined over $\Pi_{\text{jac}}(F(q))$ and given by the cubic formula

(Jac) $Y^2 = X^3 + aXZ^4 + bZ^6$ with $a, b \in F(q)$.

NOTE 2 The set of all triplets equivalent to $(X, Y, Z)$ is denoted by $(X, Y, Z)/\sim$.

The elliptic curve given in Jacobian coordinates consists of all points $R = (X, Y, Z)$ of $F(q) \times F(q) \times F(q) \setminus \{(0_F, 0_F, 0_F)\}$, such that the triple $(X, Y, Z)$ is a solution of the equation (Jac), where by an abuse of notation $(X, Y, Z)$ is identified with the equivalence class $(X, Y, Z)/\sim$ containing $(X, Y, Z)$. There is a relation



 

between the points $Q$ of $E$ when the curve is given in affine coordinates and the points $R$ of the Jacobian coordinates. Indeed, the following conditions hold:

— if $Q = (X_Q, Y_Q)$ is an affine point of $E$, then $R = (X_Q, Y_Q, 1_F)$ is the corresponding point in Jacobian coordinates;

— if $R = (X, Y, Z)$ (with $Z \neq 0_F$) is a solution of (Jac), then $Q = (X/Z^2, Y/Z^3)$ is the corresponding affine point of $E$;

— there is only one solution of (Jac) with $Z = 0_F$, namely the point $(1_F, 1_F, 0_F)$; this point corresponds to $O_E$.

In Jacobian coordinates, the group law on an elliptic curve given by (Jac) reads as follows:

— The point $(1_F, 1_F, 0_F)$ is the identity element $O_E$ with respect to "+".

— Let $R = (X, Y, Z) \neq (1_F, 1_F, 0_F)$ be a point on $E$ given in Jacobian coordinates; then $-R = (X, -Y, Z)$.

— Let $R_1 = (X_1, Y_1, Z_1)$ and $R_2 = (X_2, Y_2, Z_2)$ be two distinct points on $E$, such that $R_1 \neq R_2$ and $R_1, R_2 \neq (1_F, 1_F, 0_F)$ and denote the sum by $R_3 = (X_3, Y_3, Z_3)$. The coordinates $X_3$, $Y_3$ and $Z_3$ can be computed using the following formulae:

$$X_3 = -h^3 - 2u_1h^2 + r^2;$$

$$Y_3 = -s_1h^3 + r(u_1h^2 - X_3);$$

$$Z_3 = Z_1Z_2h;$$

with $u_1 = X_1Z_2^2$, $u_2 = X_2Z_1^2$, $s_1 = Y_1Z_2^3$, $s_2 = Y_2Z_1^3$, $h = u_2 - u_1$, and $r = s_2 - s_1$.

— Let $R = (X, Y, Z) \neq (1_F, 1_F, 0_F)$ be a point on $E$ and denote its doubling by $2R = (X_3, Y_3, Z_3)$. The coordinates $X_3$, $Y_3$ and $Z_3$ can be computed using the following formulae:

$$X_3 = t;$$

$$Y_3 = -8Y^4 + m(s - t);$$

$$Z_3 = 2YZ;$$

with $s = 4XY^2$, $m = 3X^2 + aZ^4$ and $t = -2s + m^2$.

### B.3.5 Group law in modified Jacobian coordinates

Under the same cubic equation (Jac), the group law in the modified Jacobian is given by representing the Jacobian coordinates as a quadruple $(X, Y, Z, aZ^4)$, which provides the fastest possible doublings over $E(F(q))$.

In the modified Jacobian coordinates, the group law on an elliptic curve given by (Jac) reads as follows.

— Let $R_1 = (X_1, Y_1, Z_1, aZ_1^4)$ and $R_2 = (X_2, Y_2, Z_2, aZ_2^4)$ be two distinct points on $E$ such that $R_1 \neq R_2$ and $R_1, R_2 \neq (1_F, 1_F, 0_F, 0_F)$ and denote the sum by $R_3 = (X_3, Y_3, Z_3, aZ_3^4)$. The coordinates $X_3$, $Y_3$ and $Z_3$ can be computed using the following formulae:

$$X_3 = -h^3 - 2u_1h^2 + r^2;$$

$$Y_3 = -s_1h^3 + r(u_1h^2 - X_3);$$

$Z_3 = Z_1 Z_2 h$;

$aZ_3^4 = aZ_3^4$;

with $u_1 = X_1 Z_2^2$, $u_2 = X_2 Z_1^2$, $s_1 = Y_1 Z_2^3$, $s_2 = Y_2 Z_1^3$, $h = u_2 - u_1$, and $r = s_2 - s_1$.

— Let $R = (X, Y, Z, aZ^4) \neq (1_F, 1_F, 0_F, 0_F)$ be a point on $E$ and denote its doubling by $2R = (X_3, Y_3, Z_3, aZ_3^4)$. The coordinates $X_3$, $Y_3$ and $Z_3$ can be computed using the following formulae:

$X_3 = t$;

$Y_3 = m(s - t) - u$;

$Z_3 = 2YZ$;

$aZ_3^4 = 2u\,(aZ^4)$;

with $s = 4XY^2$, $u = 8Y^4$, $m = 3X^2 + (a\,Z^4)$, and $t = -2s + m^2$.

### B.3.6 Mixed coordinates

There are computational advantages and disadvantages to representing an elliptic curve point in affine, projective, Jacobian or modified Jacobian coordinates in Reference [6]. There is no coordinate system which gives both fast additions and fast doublings. It is possible to mix different coordinates, i.e. to add two points where one is given in some coordinate system, and the other point is in some other coordinate system. The coordinate system of the result can be chosen in some coordinate system. Since there are four different kinds of coordinate systems, this gives a large number of possibilities. Mixed coordinates give the best combination of coordinate systems for doublings or additions to minimize the time for elliptic curve exponentiation. Mixed coordinates run most efficiently in the pre-computation algorithm, which is described in C.3.2.

## B.4 Group law for elliptic curves over $F(2^m)$

### B.4.1 Group law in affine coordinates

Let $F(2^m)$, for some $m \geq 1$, be a finite field. Let $E$ be an elliptic curve over $F(2^m)$ given by Formula (B.1):

(Aff) $y^2 + xy = x^3 + ax^2 + b$

with $a, b \in F(2^m)$, such that $b \neq 0_F$.

In affine coordinates, the group law on an elliptic curve given by (Aff) reads as follows:

— The point at infinity is the identity element $O_E$ with respect to "+".

— Let $R = (x, y) \neq O_E$ be a point on $E$ given affine notation. Then $-R = (x, x + y)$.

— Let $R_1 = (x_1, y_1)$ and $R_2 = (x_2, y_2)$ be two distinct points on $E$, such that $R_1 \neq \pm R_2$ and $R_1, R_2 \neq O_E$. The sum is the point $R_3 = (x_3, y_3)$, where:

$x_3 = r^2 + r + x_1 + x_2 + a$;



 

$y_3 = r(x_1 + x_3) + x_3 + y_1$;

with $r = (y_2 + y_1)/(x_2 + x_1)$.

— Let $R = (x, y)$ be a point on $E$, such that $R \neq O_E$ and $x \neq 0_F$. Its doubling is the point $2R = (x_3, y_3)$, where:

$x_3 = r^2 + r + a$;

$y_3 = r(x + x_3) + x_3 + y$;

with $r = x + (y/x)$. In the case of $R = (0_F, y)$, its doubling is $2R = O_E$.

As with the group law in the affine description of an elliptic curve over $F(p^m)$, the group law given above makes heavy use of divisions in $F(2^m)$, when the scalar multiplication is computed. However, the projective description of the elliptic curve group law can be used, which makes only one division at the end of the scalar multiplication. Both descriptions of elliptic curves are compatible.

### B.4.2 Group law in projective coordinates

NOTE 1 Using projective coordinates will result in more multiplications during the calculation but no inversions have to be computed.

The two-dimensional projective space over $F(2^m)$, $\Pi_{\text{proj}}(F(2^m))$, is given by the equivalence classes of triplets $(X, Y, Z) \in F(2^m) \times F(2^m) \times F(2^m) \setminus \{(0_F, 0_F, 0_F)\}$, where two triplets $(X, Y, Z)$, $(X', Y', Z') \in F(2^m) \times F(2^m) \times F(2^m) \setminus \{(0_F, 0_F, 0_F)\}$ are said to be equivalent if there exists $\lambda \in F(2^m)^*$, such that $(X', Y', Z') = (\lambda X, \lambda Y, \lambda Z)$. The projective analogue of the affine equation (Aff) is defined over $\Pi_{\text{proj}}(F(2^m))$, and given by the homogeneous cubic formula

(Proj) $Y^2Z + XYZ = X^3 + aX^2Z + bZ^3$ with $a, b \in F(2^m)$.

NOTE 2 The set of all triplets equivalent to $(X, Y, Z)$ is denoted by $(X, Y, Z)/\sim$.

The elliptic curve given in projective coordinates consists of all points $R = (X, Y, Z)$ of $F(2^m) \times F(2^m) \times F(2^m) \setminus \{(0_F, 0_F, 0_F)\}$ such that the triple $(X, Y, Z)$ is a solution of the equation (Proj), where by an abuse of notation $(X, Y, Z)$ is identified with the equivalence class $(X, Y, Z)/\sim$ containing $(X, Y, Z)$. Clearly, there should be a 1:1 relation between the points $Q$ of $E$ when the curve is given in affine coordinates and the points $R$ of the projective coordinates. Indeed, the following conditions hold:

— if $Q = (x_Q, y_Q)$ is an affine point of $E$, then $R = (x_Q, y_Q, 1_F)$ is the corresponding point in projective coordinates;

— if $R = (X, Y, Z)$ (with $Z \neq 0_F$) is a solution of (Proj), then $Q = (X/Z, Y/Z)$ is the corresponding affine point of $E$;

— there is only one solution of (Proj) with $Z = 0_F$, namely the point $(0_F, 1_F, 0_F)$; this point corresponds to $O_E$.

In projective coordinates, the group law on an elliptic curve given by (Proj) reads as follows.

— The point $(0_F, 1_F, 0_F)$ is the identity element $O_E$ with respect to "+".

— Let $R = (X, Y, Z) \neq (0_F, 1_F, 0_F)$ be a point on $E$ given in projective coordinates. Then $-R = (X, X + Y, Z)$.

— Let $R_1 = (X_1, Y_1, Z_1)$ and $R_2 = (X_2, Y_2, Z_2)$ be two distinct points on $E$, such that $R_1 \neq R_2$ and $R_1, R_2 \neq (0_F, 1_F, 0_F)$ and denote the sum by $R_3 = (X_3, Y_3, Z_3)$. The coordinates $X_3$, $Y_3$ and $Z_3$ can be computed using the following formulae:

$X_3 = su$;



 

$Y_3 = t(u + s^2X_1Z_2) + s^3Y_1Z_2 + su$;

$Z_3 = s^3Z_1Z_2$;

with $s = X_2Z_1 + X_1Z_2$, $t = Y_2Z_1 + Y_1Z_2$, and $u = (t^2 + ts + as^2)\, Z_1Z_2 + s^3$.

— Let $R = (X, Y, Z) \neq (0_F, 1_F, 0_F)$ be a point on $E$ and denote its doubling by $2R = (X_3, Y_3, Z_3)$. The coordinates $X_3$, $Y_3$, and $Z_3$ can be computed using the following formulae:

$X_3 = st$;

$Y_3 = X^4s + t(s + YZ + X^2)$;

$Z_3 = s^3$;

with $s = XZ$ and $t = bZ^4 + X^4$.

## B.5 Group law for elliptic curves over $F(3^m)$

### B.5.1 Group law in affine coordinates

Let $F(3^m)$, for some $m \geq 1$, be a finite field. Let $E$ be an elliptic curve over $F(3^m)$ given by Formula (B.2):

(Aff) $y^2 = x^3 + ax^2 + b$ (B.2)

with $a, b \in F(3^m)$, such that $a, b \neq 0_F$.

In affine coordinates, the group law on an elliptic curve given by (Aff) reads as follows.

— The point at infinity is the identity element $O_E$ with respect to "+".

— Let $R = (x, y) \neq O_E$ be a point on $E$ given affine notation. Then $-R = (x, -y)$.

— Let $R_1 = (x_1, y_1)$ and $R_2 = (x_2, y_2)$ be two distinct points on $E$, such that $R_1 \neq \pm R_2$ and $R_1, R_2 \neq O_E$. The sum is the point $R_3 = (x_3, y_3)$, where:

$x_3 = r^2 - a - x_1 - x_2$;

$y_3 = r(x_1 - x_3) - y_1$;

with $r = (y_2 - y_1)/(x_2 - x_1)$.

— Let $R = (x, y)$ be a point on $E$, such that $R \neq O_E$ and $y \neq 0_F$. Its doubling is the point $2R = (x_3, y_3)$, where:

$x_3 = r^2 - a + x$;

$y_3 = r(x - x_3) - y$;

with $r = ax/y$.

In the case of $R = (x, 0_F)$, its doubling is $2R = O_E$.



 

As with the group law in the affine description of an elliptic curve over $F(p^m)$, the group law given above makes heavy use of divisions in $F(3^m)$, when the scalar multiplication is computed. However, the projective description of the elliptic curve group law can be used, which makes only one division at the end of scalar multiplication. Both descriptions of elliptic curves are compatible.

### B.5.2 Group law in projective coordinates

NOTE 1 Using the projective description will result in more multiplications during the calculation but no inversions have to be computed.

The two-dimensional projective space over $F(3^m)$, $\Pi_{\text{proj}}(F(3^m))$, is given by equivalence classes of triplets $(X, Y, Z) \in F(3^m) \times F(3^m) \times F(3^m) \setminus \{(0_F, 0_F, 0_F)\}$, where two triplets $(X, Y, Z), (X', Y', Z') \in F(3^m) \times F(3^m) \times F(3^m) \setminus \{(0_F, 0_F, 0_F)\}$ are said to be equivalent if there exists $\lambda \in F(3^m)^*$, such that $(X, Y, Z) = (\lambda X, \lambda Y, \lambda Z)$. The projective analogue of the affine equation (Aff) is defined over $\Pi_{\text{proj}}(F(3^m))$, and given by the homogeneous cubic formula

(Proj) $Y^2Z = X^3 + aX^2Z + bZ^3$ with $a, b \in F(3^m)$.

NOTE 2 The set of all triplets equivalent to $(X, Y, Z)$ is denoted by $(X, Y, Z)/\sim$.

The elliptic curve given in projective coordinates consists of all points $R = (X, Y, Z)$ of $F(3^m) \times F(3^m) \times F(3^m) \setminus \{(0_F, 0_F, 0_F)\}$, such that the triple $(X, Y, Z)$ is a solution of the equation (Proj), where by an abuse of notation $(X, Y, Z)$ is identified with the equivalence class $(X, Y, Z)/\sim$ containing $(X, Y, Z)$. Clearly, there should be a 1:1 relation between the points $Q$ of $E$ when the curve is given in affine coordinates and the points $R$ of the projective coordinates. Indeed, the following conditions hold:

— if $Q = (x_Q, y_Q)$ is an affine point of $E$, then $R = (x_Q, y_Q, 1_F)$ is the corresponding point in projective coordinates;

— if $R = (X, Y, Z)$ (with $Z \neq 0_F$) is a solution of (Proj); then $Q = (X/Z, Y/Z)$ is the corresponding affine point of $E$;

— there is only one solution of (Proj) with $Z = 0_F$, namely the point $(0_F, 1_F, 0_F)$; this point corresponds to $O_E$.

In projective coordinates, the group law on an elliptic curve given by (Proj) reads as follows.

— The point $(0_F, 1_F, 0_F)$ is the identity element $O_E$ with respect to "+".

— Let $R = (X, Y, Z) \neq (0_F, 1_F, 0_F)$ be a point on $E$ given in projective coordinates. Then $-R = (X, -Y, Z)$.

— Let $R_1 = (X_1, Y_1, Z_1)$ and $R_2 = (X_2, Y_2, Z_2)$ be two distinct points on $E$, such that $R_1 \neq \pm R_2$ and $R_1, R_2 \neq (0_F, 1_F, 0_F)$, and denote the sum by $R_3 = (X_3, Y_3, Z_3)$. The coordinates $X_3$, $Y_3$ and $Z_3$ can be computed using the following formulae:

$X_3 = st^2Z_1Z_2 - s^3u;$

$Y_3 = t(s^2X_1Z_2 - t^2Z_1Z_2 + s^2u) - s^3Y_1Z_2;$

$Z_3 = s^3Z_1Z_2;$

with $s = X_2Z_1 - X_1Z_2$, $t = Y_2Z_1 - Y_1Z_2$, and $u = aZ_1Z_2 + X_1Z_2 + X_2Z_1$.

— Let $R = (X, Y, Z) \neq (0_F, 1_F, 0_F)$ be a point on $E$ and denote its doubling by $2R = (X_3, Y_3, Z_3)$. The coordinates $X_3$, $Y_3$, and $Z_3$ can be computed using the following formulae:

$X_3 = tY;$



 

$Y_3 = s(XY^2 - t) - Y^4$;

$Z_3 = Y^3Z$;

with $s = aX$ and $t = s^2Z - aY^2Z + XY^2$.

## B.6 Existence conditions for an elliptic curve $E$

### B.6.1 Order of an elliptic curve $E$ defined over $F(p)$

The trace of $E$ over $F(p)$ is bounded in $[-2\sqrt{p}, 2\sqrt{p}]$ by Hasse's theorem. Waterhouse's theorem[16] states that for a given $t$ in $[-2\sqrt{p}, 2\sqrt{p}]$, there exists an elliptic curve $E$ over $F(p)$ with trace $t$.

> "Every integer $n$ in the interval given by Hasse's theorem is the order of some elliptic curve defined over $F(p)$."[16]

### B.6.2 Order of an elliptic curve $E$ defined over $F(2^m)$

The trace of $E$ over $F(2^m)$ is bounded in $[-2\sqrt{2^m}, 2\sqrt{2^m}]$ by Hasse's theorem. The conditions that for a given $t$ in $[-2\sqrt{2^m}, 2\sqrt{2^m}]$ there is an elliptic curve $E$ over $F(2^m)$ with trace $t$ is given by Waterhouse's theorem:

> Let $t$ be an integer where $|t| \leq 2\sqrt{2^m}$. Then there exists an elliptic curve defined over $F(2^m)$ of order $2^m + 1 - t$ if, and only if, one of the following conditions hold:
>
> — $t$ is odd;
>
> — $t = 0$;
>
> — $m$ is odd and $t^2 = 2^{m+1}$;
>
> — $m$ is even and $t^2 = 2^{m+2}$ or $t^2 = 2^m$.

### B.6.3 Order of an elliptic curve $E$ defined over $F(p^m)$ with $P \geq 3$

The trace of $E$ over $F(p^m)$ is bounded in $[-2\sqrt{p^m}, 2\sqrt{p^m}]$ by Hasse's theorem. The conditions that for a given $t$ in $[-2\sqrt{p^m}, 2\sqrt{p^m}]$ there is an elliptic curve $E$ over $F(p^m)$ with trace $t$ is given by Waterhouse's theorem:

Let $t$ be an integer where $|t| \leq 2\sqrt{p^m}$. Then there exists an elliptic curve defined over $F(p^m)$ of order $p^m+1 - t$ if and only if one of the following conditions hold:

— $t$ is not divisible by $p$;

— $m$ is odd and one of the following holds:

  — $t = 0$;

  — $t^2 = 3^{m+1}$ and $P = 3$;

— $m$ is even and one of the following holds:

  — $t^2 = 4p^m$;

  — $t^2 = p^m$ and $p - 1$ is not divisible by 3;

  — $t = 0$ and $p - 1$ is not divisible by 4.



 

## B.7 Pairings

### B.7.1 Overview of pairings

Let $E$ be an elliptic curve over $F(q)$ where $q = p^m$, and let $n$ be relatively prime to the characteristic $p$ of $F(q)$. The $n$-torsion group is generated by two points when $n$ is relatively prime to $p$. $E(F(q))$ includes an $n$-torsion point $G_1$ because by definition, $n$ is a prime divisor of $\#E(F(q))$ (see Clause 4). Note that this fact does not imply $E(F(q)) \supset E[n]$. The Weil and Tate pairings are non-degenerate, bilinear maps defined over an elliptic curve $E$ to $\mu_n$. The Weil pairing is defined over the $n$-torsion group $E[n]$, and thus requires $E(F(q^B))$ such that $E(F(q^B)) \supset E[n]$. On the other hand, the Tate pairing can work if only $E(F(q^B)) \ni G_1$ and $F(q^B) \supset \mu_n$. Therefore, the computation of the Tate pairings is more efficient than that of the Weil pairing.

### B.7.2 Definitions of Weil and Tate pairings

Let $E/F$ be an elliptic curve, $n$ be a prime divisor of $\#E(F(q))$, and $E[n]$ be the $n$-torsion group, where $n$ is relatively prime to $q$. Then $E[n]$ contains two points $G_1$ and $G_2$ such that $E[n] = \langle G_1 \rangle \times \langle G_2 \rangle$ . Let $B$ be the smallest integer such that $q^B - 1$ is divisible by $n$. Then $E[n] \subseteq E[F(q^B)]$.

The Weil pairing is

$$e_n: E[n] \times E[n] \rightarrow \mu_n,$$

and the Tate pairing is

$$E(F(q^B))[n] \times E(F(q^B)) \,/\, nE(F(q^B)) \rightarrow \mu_n.$$

NOTE The detailed information on Weil and Tate pairings is described in Reference [15].

### B.7.3 Cryptographic bilinear map

A cryptographic bilinear map $e_n$ is realized by restricting the domain of the Weil or Tate pairings, which satisfy the conditions of non-degeneracy, bilinearity, and computability. In cryptographic applications, the cryptographic bilinear maps $e_n$ are described in two methods:

— $e_n: \langle G_1 \rangle \times \langle G_2 \rangle \rightarrow \mu_n$;

— $e_n: \langle G_1 \rangle \times \langle G_1 \rangle \rightarrow \mu_n$;

where $\langle G_1 \rangle$ and $\langle G_2 \rangle$ are cyclic groups of order $n$ and $\mu_n$ is the cyclic group of the $n$th roots of unity.

# Annex C
(informative)

# Background information on elliptic curve cryptosystems

## C.1 General

Annex C gives some algorithms on elliptic curve cryptosystems that are necessary for the secure elliptic curve based public key schemes described in ISO/IEC 15946-5.

## C.2 Definition of cryptographic problems

### C.2.1 Elliptic curve discrete logarithm problem (ECDLP)

For an elliptic curve $E/F(q)$, the base point $G \in E(F(q))$ with order $n$, and a point $P \in E(F(q))$, the elliptic curve discrete logarithm problem (with respect to the base point $G$) is to find the integer $x \in [0, n\text{-}1]$ such that $P = xG$ if such an $x$ exists.

The security of elliptic curve cryptosystems is based on the believed hardness of the elliptic curve discrete logarithm problem.

### C.2.2 Elliptic curve computational Diffie Hellman problem (ECDHP)

For an elliptic curve $E/F(q)$, the base point $G \in E(F(q))$ with order $n$, and points $aG, bG \in E(F(q))$, the computational elliptic curve Diffie Hellman problem is to compute $abG$.

The security of some elliptic curve cryptosystems is based on the believed hardness of the computational elliptic curve Diffie Hellman problem.

### C.2.3 Elliptic curve decisional Diffie Hellman problem (ECDDHP)

For an elliptic curve $E/F(q)$, the base point $G \in E(F(q))$ with order $n$, and points $aG, bG, Y \in E(F(q))$, the decisional elliptic curve Diffie-Hellman problem is to decide whether $Y = abG$ or not.

The security of some elliptic curve cryptosystems is based on the believed hardness of the decisional elliptic curve Diffie-Hellman problem.

### C.2.4 Bilinear Diffie-Hellman (BDH) problem

The bilinear Diffie-Hellman problems are described in two ways according to the corresponding cryptographic bilinear maps.

— For two groups $<G_1>$ and $<G_2>$ with order $n$, a cryptographic bilinear map $e_n$: $<G_1> \times <G_2> \rightarrow \mu_n$, $aG_1$, $bG_1 \in <G_1>$ , and $aG_2, cG_2 \in <G_2>$ , the bilinear Diffie-Hellman problem is to compute $e_n(G_1, G_2)^{abc}$.

— For a group $<G_1>$ with order $n$, a cryptographic bilinear map $e_n$: $<G_1> \times <G_1> \rightarrow \mu_n$, and $aG_1$, $bG_1$ $cG_1 \in <G_1>$ , the bilinear Diffie-Hellman problem is to compute $e_n(G_1, G_1)^{abc}$.

### C.2.5 Gap Diffie-Hellman (GDH) problem

The Gap Diffie-Hellman problem is the Computational Diffie-Hellman problem given access to an oracle that can solve the Decisional Diffie-Hellman problem.

## C.3 Algorithms to determine discrete logarithms on elliptic curves

### C.3.1 Security of ECDLP

The security of ECDLP depends on the selection of elliptic curves $E/F(q)$ and the size $n$ of order of the base point $G$. In C.3.1 an overview of algorithms to solve ECDLP is given. The elliptic curve $E/F(q)$ should be chosen to meet the defined security objectives taking into account the following algorithms to solve ECDLP. The size of $n$ should be set to meet the defined security objectives taking into account the baby-step-giant-step algorithm and various variants of the Pollard $\rho$ algorithm.

### C.3.2 Overview of algorithms

The following techniques are available to determine discrete logarithms on an elliptic curve.

— The Pohlig-Silver-Hellman algorithm. This is a "divide-and-conquer" method which reduces the discrete logarithm problem for an elliptic curve $E$ defined over $F(q)$ to the discrete logarithm in the cyclic subgroups of prime order dividing $\#E\ (F(q))$.

— The baby-step-giant-step algorithm and various variants of the (parallel) Pollard-ρ algorithm.

— The algorithm of Frey-Rück[7] and the Menezes-Okamoto-Vanstone algorithm[10] which both transform the discrete logarithm problem in a cyclic subgroup of $E$ with prime order $n$ to the smallest extension field $F(q^B)$ of $F(q)$, such that $n$ divides $(q^B - 1)$. The Frey-Rück algorithm runs under weaker conditions than the Menezes-Okamoto-Vanstone algorithm.

— The algorithm of Araki-Satoh[12], Smart[13] and Semaev[14] solve the discrete logarithm problem for an elliptic curve $E$ defined over $F(p^m)$ in the case $\#E(F(p^m)) = p^m$.

Unlike the situation of the discrete logarithm in the multiplicative group of some finite field, there is no known "index-calculus" available in the case of elliptic curves.

NOTE 1 The Pohlig-Silver-Hellman and baby-step-giant-step algorithms work generally on all kinds of elliptic curves while the Frey-Rück, the Menezes-Okamoto-Vanstone, Araki-Satoh, Smart, and Semaev algorithms work only on curves with special properties.

NOTE 2 The security of $n$-bit $G$ against the baby-step-giant-step algorithm and various variants of the Pollard ρ algorithm correspond to $2^{n/2}$.

### C.3.3 MOV condition

Let $n$ be as defined in the set of elliptic curve domain parameters, where $n$ is a prime divisor of $\#E(F(q))$. A value $B$ is given as the smallest integer such that $n$ divides $q^B - 1$. As mentioned above, Frey-Rück and Menezes-Okamoto-Vanstone algorithms reduce the discrete logarithm problem in an elliptic curve over $F(q)$ to the discrete logarithm in the finite field $F(q^B)$. By using the attack, the difficulty of the discrete logarithm problem in an elliptic curve $E/F(q)$ is related to the discrete logarithm problem in the finite field $F(q^B)$. The *MOV condition* describes the degree of $B$ that ensures that the security level of the discrete logarithm problem in the elliptic curve case is equivalent to the discrete logarithm problem in the finite field case. For some applications based on the Weil and Tate pairing, a reasonably small value of $B$ such as 6 is preferable.

## C.4 Scalar multiplication algorithms of elliptic curve points

### C.4.1 Basic algorithm

The computation of multiples of an elliptic curve point is called the scalar multiplication of an elliptic curve point. The scalar multiplication of an elliptic curve point is easily done using the "double-and-add" algorithm. Let $k$ be an arbitrary $l$-bit positive integer and let $k = k_{l-1}\ 2^{l-1} + ..... + k_1 2 + k_0$ be the binary representation of $k$, where $k_{l-1} = 1$.

To compute $Q = kG$, proceed as follows:

a) Set $Q: = G$.

b) For $i = l - 2$ down to $i = 0$, do:

1) $Q:= 2Q$.

2) If $k_i = 1$ then $Q:= Q + G$.

Hence, for a randomly chosen $k$ it may be expected that the process of computing $kG$ will entail ($l$-1) elliptic-curve doublings plus about $l/2$ elliptic-curve additions.

The scalar multiplication of an elliptic curve point may also be done using the "addition-subtraction" algorithm based on the non-adjacent form representation (NAF). Let $k$ be an arbitrary $l$-bit positive integer, and let $k = k_l 2^l + k_{l-1}\, 2^{l-1} + ..... + k_1 2 + k_0$ be a signed-binary representation of $k$, where $k_i = 0, +1, -1$ and no two values $k_i$ and $k_{i+1}$ are both non-zero.

NOTE The NAF representation of $k$ is uniquely determined[4]. The length of NAF representation of $k$ becomes $l$ or $l$ +1.

To determine $Q = kG$, proceed as follows:

a) Set $Q: = O_E$.

b) For $i = l$ down to $i = 0$, do:

1) Set $Q:= 2Q$.

2) If $k_i = 1$ then set $Q:= Q + G$.

3) If $k_i = -1$ then set $Q: = Q - G$.

For a randomly chosen $k$ it may be expected that the process of evaluating $kG$ will entail at most $l$ elliptic-curve doublings and about $l/3$ elliptic-curve additions.

### C.4.2 Algorithm with pre-computed table

The scalar multiplication of an elliptic curve point is easily done using the well-known "window" algorithm. The algorithm consists of two parts: precomputation and main loop. In the precomputation stage, the points $G_i = iG$ are computed for odd $i \in [1, 2^w\text{-}1]$ for some $w > 0$, where $w$ determines the size of the pre-computed table. In the main loop stage, $kG$ is computed by using the pre-computed points.

Let $k$ be an arbitrary positive integer and let $k = k_{l-1}\, 2^{l-1} + ..... + k_1 2 + k_0$ be the binary representation of $k$, where $k_{l-1} = 1$. To compute $Q = kG$, proceed as follows:

— Precomputation:

a) $G_1: = G$, $G_2: = 2G$.

b) For $i = 1$ to $2^{w-1} - 1$, do: $G_{2i+1}: = G_{2i-1} + G_2$.

— Main loop:

c) $j: = l - 1$, $Q: = G$.

d) While $j \geq 0$, do:

1) If $k_j = 0$, then $Q:=2Q$ and $j:= j - 1$.

2) Else, $h:= \sum_{j \geq i \geq t} k_i 2^{i-t}$, $Q:= 2^{j-t+1}Q + G_h$ for the least integer $t$ such that $j - t + 1 \leq w$ and $k_t = 1$, and $j:= t - 1$.



 