INTERNATIONAL STANDARD

ISO/IEC 29192-4

First edition
2013-06-01

Information technology — Security techniques — Lightweight cryptography

Part 4: Mechanisms using asymmetric techniques

Téchnologies de l'information — Techniques de sécurité — Cryptographie pour environnements contraints

Partie 4: Mécanismes basés sur les techniques asymétriques

Reference number
ISO/IEC 29192-4:2013(E)

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

Published in Switzerland

Contents

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.

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

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

ISO/IEC 29192-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

ISO/IEC 29192 consists of the following parts, under the general title *Information technology — Security techniques — Lightweight cryptography*:

— *Part 1: General*

— *Part 2: Block ciphers*

— *Part 3: Stream ciphers*

— *Part 4: Mechanisms using asymmetric techniques*

Further parts may follow.

Introduction

This part of ISO/IEC 29192 specifies three lightweight mechanisms based on asymmetric cryptography. The three mechanisms have different functionality, different supporting infrastructures, and different performance profiles.

— cryptoGPS is a lightweight asymmetric identification scheme; in the cryptographic literature such schemes are generally described as interactive proofs of knowledge. While there are many types of such scheme, the computational costs for the prover when using cryptoGPS are relatively low. This is particularly the case since cryptoGPS is well-suited to an implementation strategy using what is often referred to as "coupons". These are, essentially, the results given by a modest off-line pre-computation, with coupons being used by the prover at each invocation of the cryptoGPS scheme. The resultant scheme, with the role of the prover being taken by a computationally restricted device such as an RFID tag, offers very useful performance trade-offs.

— ALIKE is an asymmetric mechanism for authentication and key exchange. Based on a variant of RSA, ALIKE offers a unilateral authentication and an additional functionality, i.e. secure key establishment. ALIKE offers implementation advantages when compared to conventional asymmetric solutions such as RSA.

— The third mechanism is an identity-based signature scheme. Hence a trusted third party is involved in the computation of distinct signature keys. This scheme offers implementation advantages over many other schemes in the cryptographic literature.

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

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

The holders of these patent rights have assured ISO and the IEC that they are willing to negotiate licences either free of charge or 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:

France Telecom
38-40, rue du Général Leclerc, F-92794 Issy Les Moulineaux CEDEX 9, France

Gemalto SA
6, rue de La Verrerie, 92917 Meudon CEDEX, France

Agency for Science, Technology and Research
Agency for Science, Technology and Research c/o Exploit Technologies Pte Ltd,
30 Biopolis Street, #09-02 Matrix, Singapore 138671

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 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 — Lightweight cryptography

Part 4: Mechanisms using asymmetric techniques

1 Scope

This part of ISO/IEC 29192 specifies three lightweight mechanisms using asymmetric techniques:

- a unilateral authentication mechanism based on discrete logarithms on elliptic curves;
- an authenticated lightweight key exchange (ALIKE) mechanism for unilateral authentication and establishment of a session key;
- an identity-based signature mechanism.

2 Normative references

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

ISO/IEC 15946-1, *Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 1: General*

ISO/IEC 29192-1, *Information technology — Security techniques — Lightweight cryptography — Part 1: General*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 29192-1 and the following apply.

3.1
asymmetric cryptographic technique
cryptographic technique that uses two related operations: a public operation defined by a public data item, and a private operation defined by a private data item

Note 1 to entry: The two operations have the property that, given the public operation, it is computationally infeasible to derive the private operation.

[SOURCE: ISO/IEC 9798-5:2009, definition 2.3]

3.2
asymmetric pair
two related data items where the private data item defines a private operation and the public data item defines a public operation

[SOURCE: ISO/IEC 9798-5:2009, definition 2.5]

3.3
challenge

procedure parameter used in conjunction with secret parameters to produce a response

[SOURCE: ISO/IEC 9798-5:2009, definition 2.6]

3.4
claimant

entity whose identity can be authenticated, including the functions and the private data necessary to engage in authentication exchanges on behalf of a principal

[SOURCE: ISO/IEC 9798-5:2009, definition 2.7]

3.5
claimant parameter

public data item, number or bit string, specific to a given claimant within the domain

[SOURCE: ISO/IEC 9798-5:2009, definition 2.9]

3.6
collision-resistant hash-function

hash-function satisfying the following property: it is computationally infeasible to find any two distinct inputs which map to the same output

Note 1 to entry: computational feasibility depends on the specific security requirements and environment.

[SOURCE: ISO/IEC 10118-1:2000, definition 3.2]

3.7
coupon

pair of pre-computed numbers to be used only once

Note 1 to entry: One of the numbers shall be kept secret, and the other shall remain secret until the time of use.

[SOURCE: ISO/IEC 9798-5:2009, definition 2.8, modified]

3.8
domain

collection of entities operating under a single security policy

Note 1 to entry: For instance, public key certificates created either by a single certification authority, or by a collection of certification authorities using the same security policy.

[SOURCE: ISO/IEC 9798-5:2009, definition 2.11]

3.9
domain parameter

public key, or function, agreed and used by all entities within the domain

[SOURCE: ISO/IEC 9798-5:2009, definition 2.12]

3.10
entity authentication

corroboration that an entity is the one claimed

[SOURCE: ISO/IEC 9798-1:2010, definition 3.14]

3.11
exchange multiplicity parameter
number of exchanges of information involved in one instance of an authentication mechanism

[SOURCE: ISO/IEC 9798-5:2009, definition 2.15]

3.12
hash-function
function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:

— it is computationally infeasible to find for a given output, an input which maps to this output;

— it is computationally infeasible to find for a given input, a second input which maps to the same output

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment.

[SOURCE: ISO/IEC 10118-1:2000, definition 3.5]

3.13
master secret key
secret data item

Note 1 to entry: Master secret key should only be used by the trusted server in accordance with the process of generation of signer private data.

3.14
private key
private data item of an asymmetric pair

Note 1 to entry: Private key shall be kept secret and should only be used by a claimant in accordance with an appropriate response formula, thereby establishing its identity.

[SOURCE: ISO/IEC 9798-5:2009, definition 2.21]

3.15
procedure parameter
transient public data item used in an instance of an authentication mechanism, e.g. a witness, challenge or response

[SOURCE: ISO/IEC 9798-5:2009, definition 2.22]

3.16
public key
public data item of an asymmetric pair, that can be made public and shall be used by every verifier for establishing the claimant's identity

[SOURCE: ISO/IEC 9798-5:2009, definition 2.23]

3.17
random number
time variant parameter whose value is unpredictable

[SOURCE: ISO/IEC 9798-1:2010, definition 3.29]

3.18
response
procedure parameter produced by the claimant, and processed by the verifier for checking the identity of the claimant

[SOURCE: ISO/IEC 9798-5:2009, definition 2.25]

3.19
secret parameter
number or bit string that does not appear in the public domain and is only used by a claimant

Note 1 to entry: For instance, a private key.

[SOURCE: ISO/IEC 9798-5:2009, definition 2.26]

3.20
sign
signature generation process that takes a message and a signing key of a signer to produce a signature

3.21
signer
entity with a unique bit string as an identity, including the functions and the private data necessary to engage in generation of a signature

3.22
signing key
secret data item given by the trusted server

Note 1 to entry: Signing key should only be used by a signer in accordance with the process of generation of a signature.

3.23
token
message consisting of data fields relevant to a particular communication and which contains information that has been produced using a cryptographic technique

[SOURCE: ISO/IEC 9798-5:2009, definition 2.27]

3.24
unilateral authentication
entity authentication which provides one entity with assurance of the other's identity but not vice versa

[SOURCE: ISO/IEC 9798-1:2010, definition 3.39]

3.25
verifier
entity including the functions necessary for engaging in authentication exchanges on behalf of an entity requiring an entity authentication or for engaging in verifying a signature of a given message and signer

[SOURCE: ISO/IEC 9798-5:2009, modified — Added the signature verification case]

3.26
verify
verification process that takes a message, a signature and an identity of a signer to output `accept` meaning the given signature is generated by the signer with the corresponding signing key, or `reject` otherwise

3.27
witness
procedure parameter that provides evidence of the claimant's identity to the verifier

[SOURCE: ISO/IEC 9798-5:2009, definition 2.31]

4 Symbols and abbreviated terms

For the purposes of this part of ISO/IEC 29192, the following symbols and abbreviated terms apply.

$|A|$ bit size of the number A if A is a non-negative integer (i.e. the unique integer i so that $2^{i-1} \le A < 2^{i}$ if $A > 0$, or 0 if $A = 0$, e.g. $|65\,537| = |2^{16} + 1| = 17$), or bit length of the bit string A if A is a bit string

NOTE To represent a number A as a string of α bits with $\alpha > |A|$, $\alpha - |A|$ bits set to 0 are appended to the left of the $|A|$ bits.

$\lfloor A \rfloor$ the greatest integer that is less than or equal to the real number A

$A[i]$ the i^{th}-bit of the number A, where $A[1]$ is the right-most bit and $A[|A|]$ is the left-most bit

$B \| C$ bit string resulting from the concatenation of data items B and C in the order specified. In cases where the result of concatenating two or more data items is input to a cryptographic algorithm as part of an authentication mechanism, this result shall be composed so that it can be uniquely resolved into its constituent data strings, i.e. so that there is no possibility of ambiguity in interpretation. This latter property could be achieved in a variety of different ways, depending on the application. For example, it could be guaranteed by

(a) fixing the length of each of the substrings throughout the domain of use of the mechanism, or

(b) encoding the sequence of concatenated strings using a method that guarantees unique decoding, e.g. using the distinguished encoding rules defined in ISO/IEC 8825-1 [18]

D response (procedure parameter)

d challenge (procedure parameter)

E elliptic curve (domain parameter)

E_K block cipher encryption function with key K

e public exponent (domain parameter)

$f_0(u, x)$ $f_0(u, x) = 0 \| x \| \ldots \| 0 \| x \| 0 \| x^*$ where x^* represents the most significant bits of x (potentially no bits) required so that the length of $0 \| x \| \ldots \| 0 \| x \| 0 \| x^*$ is equal to u

$f_1(u, x)$ $f_1(u, x) = 1 \| x \| \ldots \| 1 \| x \| 1 \| x^*$ where x^* represents the most significant bits of x (potentially no bits) required so that the length of $1 \| x \| \ldots \| 1 \| x \| 1 \| x^*$ is equal to u

h hash-function

$|h|$ bit length of the hash-code produced by the hash-function h

HE padding function based on the block cipher E_K (domain parameter)

ID binary string that represents the identity or identification information

L bit length of the padding-code produced by the function HE (domain parameter)

m message

N composite modulus (domain parameter)

n order of the base point P (domain parameter)

$[n]P$ multiplication operation that takes a positive integer n and a point P on the curve E as input and produces as output another point Q on the curve E, where $Q = [n]P = P + P + \ldots + P$ is the sum of n occurrences of P. The operation satisfies $[0]P = 0_E$ (the point at infinity), and $[-n]P = [n](-P)$

P base point over the elliptic curve E (domain parameter)

$p_1, p_2 \ldots$ prime factors of the modulus in ascending order, i.e. $p_1 < p_2 < \ldots$ (secret parameters)

Q, Q_i private key (secret parameter)

q field size (domain parameter)

r fresh random number or fresh string of random bits (secret parameter)

T public point (domain parameter)

t master secret key (secret parameter)

u bit length of the key K in the block cipher encryption function E_K (domain parameter)

v bit length of a block-message in the block cipher encryption E_K (domain parameter)

W witness (procedure parameter)

w security parameter (domain parameter)

'$X_1X_2 \ldots$' number whose hexadecimal representation is $X_1X_2 \ldots$, where each X_i is equal to one of 0-9 and A-F

α modulus size in bits, i.e. $2^{\alpha-1} \leq$ modulus $< 2^{\alpha}$, also denoted $|\text{modulus}|$ (domain parameter)

δ length of fresh strings of random bits for representing challenges (domain parameter)

ρ length of fresh strings of random bits for representing random numbers (domain parameter)

$\{a, b, c, \ldots\}$ set containing the elements a, b, c, …

5 Unilateral authentication mechanism based on discrete logarithms on elliptic curves

5.1 General

This mechanism, cryptoGPS – also called GPS in the earlier cryptographic literature –, is due to Girault, Poupard, and Stern [6]. The revised name is now used so as to avoid confusion with the physical location service GPS. cryptoGPS is a zero-knowledge identification scheme that provides unilateral entity authentication. Several variants of cryptoGPS are specified in ISO/IEC 9798-5 [21] and the version most suitable to constrained devices, along with some optimisations, is presented below.

5.2 Security requirements for the environment

The cryptoGPS mechanism enables a verifier to check that a claimant knows the elliptic curve discrete logarithm of a claimed public point with respect to a base point. A general framework for cryptographic techniques based on elliptic curves is given in ISO/IEC 15946-1.

NOTE 1 This mechanism implements the elliptic curve variant [5] of the cryptoGPS [6] scheme due to Girault, Poupard and Stern. It allows use of the so-called LHW (Low Hamming Weight) variant [4] particularly suitable for environments where the resources of the claimant are very low.

Within a given domain, the following requirements shall be satisfied.

a) Domain parameters that govern the operation of the mechanism shall be selected. The selected parameters shall be made available in a reliable manner to all entities within the domain.

b) Every claimant shall be equipped with the same elliptic curve E and a set of parameters, namely the field size q, a base point P over E, and n the order of point P. The curve and the set of parameters are either domain parameters or claimant parameters.

c) Each point P used as the base for elliptic curve discrete logarithms shall be such that, for any arbitrary point J of the curve, finding an integer k in $[0, n - 1]$ (if one exists) such that $J = [k]P$ is computationally infeasible, where feasibility is defined by the context of use of the mechanism.

d) Every claimant shall be equipped with a private key.

e) Every verifier shall obtain an authentic copy of the public key corresponding to the claimant's private key.

NOTE 2 The exact means by which the verifier obtains a trusted copy of the public point specific to the claimant is beyond the scope of this part of ISO/IEC 29192. This may, for example, be achieved by the use of public-key certificates or by some other environment-dependent means.

f) Every verifier shall have the means to produce fresh strings of random bits. When coupons are not used, every claimant shall also have the means to produce fresh strings of random bits.

g) If the mechanism makes use of a hash-function, then all entities within the domain shall agree on a hash-function, e.g. one of the functions specified in ISO/IEC 10118-3 [23].

5.3 Key production

For claimant A, a fresh string shall be uniformly selected at random from the set $\{2, 3, \ldots, n - 2\}$. The string represents the private key, denoted Q.

The number $\sigma = |n|$ gives the number of bits to be used to represent private keys.

Denoted G(A), the public point for claimant A is set equal to either

a) The inverse of the multiplication of the base point P by the number Q.

$$G(A) = (x_G, y_G) = -[Q]P$$

NOTE 1 This version is the most suitable for constrained devices.

b) Or the multiplication of the base point P by the number Q.

$$G(A) = (x_G, y_G) = [Q]P$$

The challenges are selected from a set of integers S of cardinality Δ, where $2^{\delta - 1} < \Delta \leq 2^{\delta}$. The length in bits of the greatest possible challenge is denoted by β. A value of δ from 8 to 40 is appropriate for most applications. Unless otherwise specified, the value of δ is set equal to 40. It is a domain parameter.

NOTE 2 The total number of possible challenges should be limited to 2^{40}. If this recommendation is not followed, then special care should be taken to prevent the verifier using the claimant as a signing oracle.

NOTE 3 When the set of challenges is the interval $[0, \Delta - 1]$, then $\beta = \delta$.

NOTE 4 A challenge is said to be LHW (Low Hamming Weight) if there are at least $\sigma - 1$ zero bits between any two consecutive one bits in its binary representation.

NOTE 5 The definition of the public point $G(A)$ differs slightly from that defined in ISO/IEC 9798-5 [21]. This change allows more compact and efficient implementations of the resultant on-tag computation because the response formula is now an addition which is easier and more compact to implement than an integer subtraction.

5.4 Unilateral authentication mechanism

The bracketed letters in Figure 1 correspond to the steps of the mechanism, including the exchanges of information, described in detail below. The claimant is denoted by *A*. The verifier is denoted by *B*.

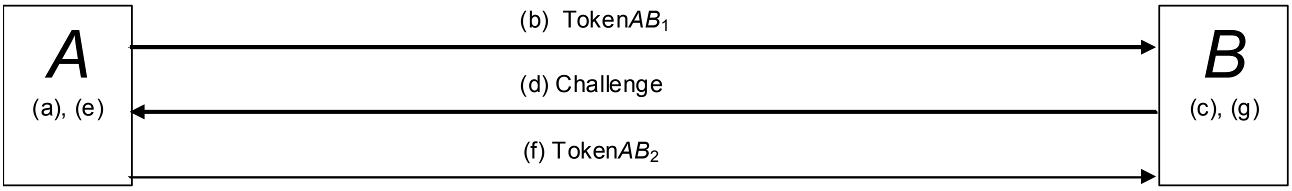

Figure 1 — Mechanism using a discrete logarithm with respect to elliptic curves

The claimant shall store a number δ, a base P, and a private key Q (as a string of σ bits). Unless otherwise specified, $\delta = 40$.

In the case of coupon use, in addition to a number δ and a private key Q, the claimant need only store a set of coupons and is not required to have a means to produce fresh strings of random bits. To be used only once, each coupon consists of a ρ-bit string (that need not be stored if it can be reproduced by a pseudo-random function, e.g. one of the functions specified in ISO/IEC 18031 [25]) and a witness.

In addition to a number δ and a number σ, the verifier shall be provided with a trusted copy of a public point $G(A)$ and a trusted copy of the curve E, the base point P and the parameters q and n.

For each application of the mechanism, the following procedure shall be performed. The verifier *B* shall only accept the claimant *A* as valid if the procedure completes successfully.

a) For each authentication,

   1) either a coupon (r, W) is used.

   2) or a fresh string of ρ bits shall be uniformly selected at random. It shall be kept secret.

$$\rho = \sigma + \beta + 80$$

NOTE 1 If the fresh string of ρ bits is selected at random, then the probability that the leftmost 80 bits are all equal is negligible.

Denoted r, the number represented by the fresh string shall be converted into a witness, denoted W.

Witness formula: $W = \mathrm{EC2OSP}_E([r]P, \mathrm{fmt})$

with $\mathrm{EC2OSP}_E$ the function to convert a point on elliptic curve E to octet strings defined in ISO/IEC 15946-1 and fmt a format specifier, which is one of the symbolic values `compressed`, `uncompressed`, or `hybrid`.

NOTE 2 Under certain implementation circumstances some might prefer to use the witness formula $W = \mathrm{EC2OSP}_E([r \bmod n]P, \mathrm{fmt})$.

b) *A* sends TokenAB_1 to *B*. TokenAB_1 can be either witness W or a hash-code of W and *Text*, one of the following four hash variants.

The four hash variants are $h(W \,||\, Text)$, $h(W \,||\, \mathrm{h}(Text))$, $h(h(W) \,||\, Text)$, and $h(h(W) \,||\, h(Text))$, where h is a hash-function and *Text* is an optional text field (it may be empty). If the text field is non-empty, then B shall have the means to recover the value of *Text*; this may require that A sends all or part of the text field with the token. How the text field is made available for use in applications is outside the scope of this part of ISO/IEC 29192. Annex A of ISO/IEC 9798-1 [20] gives information on the use of text fields. The hash variant is a domain parameter.

c) On receipt of TokenAB_1, a fresh string shall be uniformly selected at random from the set S.

d) B sends the fresh string as a challenge to A. The fresh string represents a number denoted d.

NOTE 3 If an LHW challenge is used, it can be transmitted in a compressed form to A who must have the means to retrieve the original challenge before step e)1).

e) On receipt of the challenge, the following computational steps are performed.

1) If the challenge is not an element of S, then the procedure fails.

2) A response D shall be computed from the random number r and the private key Q.

Response formula is either:

i) $D = r + d \times \mathrm{Q}$ if $G(A) = -[\mathrm{Q}]P$

NOTE 4 If the challenge received is an LHW challenge, the computation of D is reduced to a serial addition of r with a concatenation of copies of Q, separated by zero bits.

or:

ii) $D = r - d \times \mathrm{Q}$ if $G(A) = [\mathrm{Q}]P$

f) A sends TokenAB_2 to B. TokenAB_2 is the response D computed from step e)2).

g) On receipt of TokenAB_2, the following computational steps are performed.

1) If the response D is not a string of ρ bits and/or if the leftmost 80 bits of D are all equal, then the procedure fails.

2) Denoted W^*, a witness shall be computed.

Verification formula: $W^* = \mathrm{EC2OSP}_E([d]G(A) + [D]P, \mathrm{fmt})$

NOTE 5 Under certain implementation circumstances some might prefer to use the verification formula $W = \mathrm{EC2OSP}_E([d]G(A) + [D \bmod n]P, \mathrm{fmt})$.

3) If either witness W^* or a hash-code of W^* and Text (one of four hash variants) is identical to TokenAB_1 received in step b), then the procedure is successful. Otherwise the procedure fails.

NOTE 6 Other information may be sent with any exchange of the procedure. B may use such information to help compute the value of the optional Text field. For example, A may send information such as certificates with TokenAB_1.

6 Unilateral authenticated key exchange mechanism based on encryption

6.1 General

This mechanism, ALIKE, has been designed for contact-less transactions which are subject to very strong time limitations. In this protocol, a verifier (e.g. a reader or a terminal) authenticates a prover (e.g. a contact-less card) relative to a certification authority. Additionally, the prover and the verifier establish a session key for secure messaging. The originality of ALIKE is that it allows the use of low-cost readers (without Secure Access Module) while achieving strong time limitations. ALIKE is based on a public-key encryption scheme called RSA for paranoids [15] – a variant of RSA [13] – that enjoys very fast decryption. In ALIKE, the decryption is done by the prover (e.g. a contact-less card) where a cryptographic coprocessor is commonly available.

NOTE ALIKE stands for Authenticated Lightweight Key Exchange. The previous name for ALIKE was SPAKE [3].The security proofs of ALIKE are available in [3].

6.2 Security requirements for the environment

The ALIKE mechanism enables a verifier to authenticate a claimant relative to a certification authority and to establish a session-key for secure messaging.

Within a given domain, the following requirements shall be satisfied.

a) Domain parameters that govern the operation of the mechanism shall be selected. The selected parameters shall be made available in a reliable manner to all entities within the domain.

b) Every claimant shall be equipped with distinct prime factors so that knowledge of their product, i.e. the modulus (a claimant parameter), shall not feasibly enable any entity to deduce them, where feasibility is defined by the context of use of the mechanism.

c) All entities within the domain shall agree on a block cipher E_K, e.g. one of the algorithms specified in ISO/IEC 29192-2 [27] or in ISO/IEC 18033-3[26]. The key-size is denoted by u and the block-size is denoted by v. u and v shall be equal or greater than 128 bits and u shall be equal or greater than v. The maximum entropy of the secret key K of the block-cipher E_K is set to v - 1. The secret key K of length u is computed from a secret key x of length v - 1 using the functions $f_0(u, x)$ and $f_1(u, x)$.

NOTE By virtue of the definition of $f_0(u, x)$ and $f_1(u, x)$, the first bit of the block cipher key is set either to 0 or 1. This guarantees the independence of the two different uses of the block cipher in the protocol.

d) Every claimant and every verifier shall have the means to produce random numbers.

6.3 Key production

A number, denoted α, fixes the bit length of the modulus N, i.e. $2^{\alpha-1}$ < modulus < 2^{α}, in accordance with the context of use of the mechanism. It is a domain parameter. The bit length of the modulus shall be chosen such that the complexity estimation of the fastest factorization algorithms [8], [16] – whose running time depends on the size of the modulus N – is greater than the required level of security.

A non-negative integer, denoted w, shall be chosen such that $w > 2.v$. w is a security parameter and it is a domain parameter. w is also the bit length of p_1 and shall be chosen such that the complexity estimation of the fastest known algorithm [7] – whose running time depends on the size of $|p_1|$ – is greater than the required level of security.

Claimant A shall keep secret the two distinct large prime factors, denoted p_1 and p_2, of the modulus N. The prime factors p_1 and p_2 shall be chosen such that the modulus N is unbalanced with $|p_1| << |p_2|$.

a) Generate two primes p_1 and p_2 such that

   1) $|p_1| = w$ and $\gcd(e, p_1 - 1) = 1$ where e is the public exponent. e shall be chosen large enough to avoid the Coppersmith attack [2] and compliant with the Shamir inferior bound [15]. The value $e = 11$ has some practical advantages.

   2) $|p_2| = \alpha - w$.

   3) $|p_1 \text{ x } p_2| = \alpha$.

b) Compute $N = p_1 \text{ x } p_2$ and $t = e^{-1} \bmod (p_1 - 1)$. The public key is (N, e) and the private key is (p_1, t). The public key is certified by a certification authority.

Claimant A shall be equipped with private key $S_A = (p_1, t)$ and public key $P_A = (N, e)$ corresponding to the modulus N, see [15].

Claimant A shall be equipped with a certificate of the public key P_A, denoted σ.

NOTE The exact means by which the claimant obtains a trusted copy of his public key is beyond the scope of this part of ISO/IEC 29192. This may, for example, be achieved by the use of public-key certificates or by some other environment-dependent means.

6.4 Unilateral authentication exchange

The bracketed letters in Figure 2 correspond to the steps of the mechanism, including the exchanges of information, described in detail below. The claimant is denoted *A*. The verifier is denoted *B*.

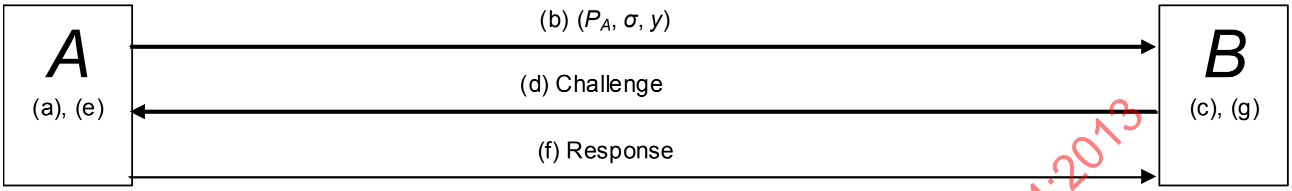

Figure 2 — ALIKE

The following procedure shall be performed. The verifier *B* shall only accept the claimant *A* as valid if the procedure completes successfully.

a) A fresh number k of length $v - 1$ shall be uniformly selected at random by the claimant. The block cipher is used to calculate a commitment y: $y = E_{f\_0(u,k)}(0)$

b) *A* sends (P_A, σ, y) to the verifier *B*.

c) A fresh number r of length $v - 1$ shall be uniformly selected at random by the verifier. The padding value $pad = E_{f\_1(u,r)}(0)$ is derived and the message ($r \| pad$) is encrypted with P_A. The result $d = (r \| pad)^{e} \bmod N$ is the challenge.

d) *B* sends the challenge d to A.

e) On receipt of the challenge, the following computational steps are performed.

   1) If the size of the challenge is not equal to $|N|$, then the procedure fails.

   2) *A* uses the private key S_A to recover the plaintext, $d^{t} \bmod p_1$, and verify the consistency of the padding in the plaintext:

      i) If the padding is not correct, then the procedure fails.

      ii) The padding is removed to recover r.

   3) A response D shall be computed by encrypting the number k using the block cipher, i.e. $D = E_{f\_0(u,\, r)}(0 \| k)$

f) *A* sends the response *D* to *B.*

g) On receipt of the response *D*, the verification process is performed.

   1) *B* verifies that σ is a trusted copy of the public key of the claimant *A.*

      iii) If the verification fails, then the procedure fails.

      iv) Else *B* recovers k' from the response *D*.

   2) If $E_{f\_0(u,\, k')}(0) = y$, then the procedure is successful. Otherwise the procedure fails.

6.5 Session-key derivation

Optionally, a session-key could be established between the claimant *A* and the verifier *B* for secure messaging using the common secret (*r*, *k*).

7 Identity-based signature mechanism

7.1 General

An identity based cryptosystem, is an asymmetric cryptographic technology that allows a public key to be calculated from an identity and a set of public mathematical parameters and that allows for the corresponding private key to be calculated from an identity, a set of public mathematical parameters, and a domain-wide secret value. A user public key can be calculated by anyone who has the necessary public parameters; while a cryptographic secret is needed to calculate a user private key, and the calculation can only be performed by a trusted server that has this secret.

An Identity Based Signature (IBS) scheme under this framework is a signature scheme where the signature verification can be done without needing the verifier and the signer to interact with each other, either directly or through a proxy such as a directory or certificate server, before verifying the signatures. Other systems may require a connection to a server for each verification operation.

The scheme [9] described below is motivated by Bellare, Namprepre and Neven's identity-based identification scheme [1], which is actually based on Schnorr's identification scheme [14], but is further optimized. By using the technique specified in Annex B, the Identity Based Signature Mechanism stated in this section can become a very efficient mechanism for lightweight devices.

NOTE The security proofs of the scheme described below are available in [9].

7.2 Security requirements for the environment

The IBS mechanism enables a verifier to check that a signer uses his signing key to produce a signature for a given message.

Within a given domain, the following requirements shall be satisfied.

a) Domain parameters that govern the operation of the mechanism shall be selected. The selected parameters shall be made available in a reliable manner to all entities within the domain.

b) Domain parameters shall include an elliptic curve *E* and a set of parameters, namely the base point *P* over *E*, and *n* the order of point *P*.

c) Each point *P* used as the base for elliptic curve discrete logarithms shall be such that, for any arbitrary point *J* of the curve, finding a number *k* in [0, $n - 1$] (if one exists), so that $J = [k]P$ is computationally infeasible, where feasibility is defined by the context of use of the mechanism.

d) Every signer and the trusted server shall have the means to produce random numbers.

e) All entities within the domain shall agree on a collision-resistant hash-function, e.g. one of the functions specified in ISO/IEC 10118-3 [23].

f) The trusted server shall uniformly select a fresh number *t* at random, which shall be non-zero and less than *n*, and compute the public point *T* which is set equal to the multiplication of the base point *P* by number *t*: $T = [t]P$

g) The trusted server shall keep *t* as the master secret key and publish the curve *E*, the point *T*, the base point *P*, and the number *n* as the parameters.

7.3 Key production

Signer *A* shall ask the trusted server to generate his signing key. For each application of the mechanism, the following procedure shall be performed by the trusted sever.

The trusted server shall uniformly select a fresh number *r* at random, which shall be non-zero and less than *n* and compute the point $R = (x_R, y_R)$ which is set equal to the multiplication of the base point *P* by number *r*.

$$R = [r]P$$

a) A number *s* shall be computed from the random number *r* and the master secret key of the trusted server *t*.

$$s = r + h(x_R \,||\, ID) \text{ x } t \bmod n$$

where *h* is a collision-resistant hash-function and *ID* is a binary string that represents the identity or identification information of the signer *A*.

b) The signing key for signer *A* shall be {*R, s*}.

NOTE A correctly generated private key shall fulfill the following formula $[s]P = R + [h(x_R \,||\, ID)]T$

7.4 Sign

To sign an arbitrary length message *m* with the signing key of the signer *A*, the following procedure shall be performed.

a) The signer shall uniformly select a fresh number *y* at random, which shall be non-zero and less than *n*.

b) Compute the point $Y = (x_Y, y_Y)$ which is set equal to the multiplication of the base point *P* by number *y*.

$$Y = [y]P$$

c) A number *z* shall be computed from the private key *R* and *s*.

$$z = y + h(x_Y \,||\, x_R \,||\, m) \text{ x } s \bmod n$$

d) The signature of signer *A* and the message *m* shall be {*Y, R, z*}.

7.5 Verify

To verify a signature {*Y, R, z*} of the signer *A* with identity *ID* for the message *m*, the verifier shall compute

$$c = h(x_Y \,||\, x_R \,||\, m)$$

and check if the following equality holds:

$$[z]P = Y + [c]R + [c \text{ x } h(x_R \,||\, ID)]T$$

The verifier shall output `accept` if it holds, or `reject` otherwise.

Annex A
(normative)

Object identifiers

```
LightweightCryptography-4{
    iso(1) standard(0) lightweight-cryptography(29192)
    part4(4) asn1-module(0) algorithm-object-identifiers(0)}
    DEFINITIONS ::= BEGIN
EXPORTS ALL;

OID ::= OBJECT IDENTIFIER -- alias
-- Synonyms
is29192-4 OID ::= {iso(1) standard(0) lightweight-cryptography(29192) part4(4)}
mechanism OID ::= {is29192-4 mechanisms(1)}
-- Lightweight cryptographic mechanisms
lw-discrete-logarithms-ecc-CryptoGPS OID ::= {mechanism
    lw-discrete-logarithms-ecc-CryptoGPS(1)}
lw-authenticated-key-exchange-ALIKE OID ::= {mechanism
    lw-authenticated-key-exchange-ALIKE(2)}
lw-identity-based-signature-IBS OID ::= {mechanism
    lw-identity-based-signature-IBS(3)}

END -- LightweightCryptography-4
```

Annex B
(normative)

Memory-Computation Trade-Off Technique

The following technique (due to Liu et al. [9]) can be used to simplify the computation of an exponentiation or a scalar multiplication. Such operations are used in many ISO/IEC specified cryptographic mechanisms, in particular those based on discrete logarithm. For example, some of the signature mechanisms specified in ISO/IEC 14888-3 [24] and ISO/IEC 9796-3 [19], and the entity-authentication mechanisms in ISO/IEC 9798-5 [21] can be implemented in this way. By using this technique, the Identity Based Signature Mechanism stated in the section 7 can become a very efficient mechanism for lightweight devices.

The trade-off of this computation-facilitating technique is an increase in memory requirement.

Let E be an elliptic curve, P be the base point over E, and n be the order of point P.

For each number i from the set $[0, |n|-1]$, the points Y_i are set equal to the scalar multiplication of the base point P by 2^i, i.e.

$$Y_\mathrm{i} = [2^i]P$$

To compute $Y = [y]P$ for a non-zero number y which is less than n, the following procedure shall be performed:

a) Set $Y = [0]P$

b) For $i = 1$ to $|n|$ compute:

$$\text{If } y[i] = 1\text{, then } Y = Y + Y_{\mathrm{i}-1}$$

c) Output Y

Annex C
(informative)

Numerical examples

C.1 cryptoGPS mechanism

C.1.1 Key production

The elliptic E curve for this example is curve P-192 defined in FIPS PUB 186-3 [17].

$$E\colon Y^2 = X^3 - 3X + b \text{ over } F_q$$

q = `FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFE  FFFFFFFF  FFFFFFFF`

b = `64210519  E59C80E7  0FA7E9AB  72243049  FEB8DEEC  C146B9B1`

Base point P over E.

P = (x_P, y_P)
= `(188DA80E  B03090F6  7CBF20EB  43A18800  F4FF0AFD  82FF1012,`
`07192B95  FFC8DA78  631011ED  6B24CDD5  73F977A1  1E794811)`

n is the order of point P.

n = `FFFFFFFF  FFFFFFFF  FFFFFFFF  99DEF836  146BC9B1  B4D22831`

For hash-function h, this example makes use of SHA-256, i.e. the fourth hash-function specified in ISO/IEC 10118-3 [23].

The bit length is $\delta = 40$ for the challenges and $\sigma = |n| = 192$ bits.

C.1.2 Authentication exchange

Private key

Q = `4F1DF03A  A32DCA02  652E83E7  E5FF5259  D61F5563  B3A0FA10`

Public point

In the first variant, the public point $G(A)$ is equal to the inverse of the multiplication of the base point P by the number Q.

$G(A)$ = $-[Q]P$
= (x_G, y_G)
= `(D753BF14  9529BC23  B1850A37  57C4D34A  0D686A95  C3B03855,`
`1656B8CB  2896BFD4  BC8F94A8  F3708741  B954CC44  4FC3951A)`

In the second variant, the public point $G(A)$ is equal to the multiplication of the base point P by the number Q.

```
G(A) = [Q]G
     = (xG, yG)
     = (D753BF14  9529BC23  B1850A37  57C4D34A  0D686A95  C3B03855,
        E9A94734  D769402B  43706B57  0C8F78BD  46AB33BB  B03C6AE5)
```

Step a

r is a fresh string of random bits of length $\rho = \sigma + \beta + 80 = 192 + 40 + 80 = 312$ bits.
W is the witness such that $W = \text{EC2OSP}_E([r]P,$ `uncompressed`$)$.

```
r =        05E8B1  E1121B08  FB9A0F58  FC1E932F  9CEFE94D  629BC223  40B5F04B
         554DCD2B  C812A76D  98F8BA3E

[r]P =   (DAD48D02  4B83E223  4C0F5FFF  B51C15B7  1D52CF92  B35358CF,
          FFE42756  843D0DF8  F3166971  E8AF6E22  6FD381B0  A816720F)

W =            04  DAD48D02  4B83E223  4C0F5FFF  B51C15B7  1D52CF92  B35358CF
         FFE42756  843D0DF8  F3166971  E8AF6E22  6FD381B0  A816720F
```

Step b

TokenAB_1 is equal to $h(W \,||\, Text)$ where text field $Text$ is empty. Hence TokenAB_1 = $h(W)$.

```
h(W) =   0EB01E5E  32CA889D  099C8F6E  4CC3CB08  A3CD6008  C2849B43  0E07BCC7
         B5241843
```

Step c, d

Verifier's challenge:

```
d =            2D  F0F5B4F2
```

Step e, f

In the first variant, the response to challenge, TokenAB_2, is $D = r + d$ x Q.

```
D =         5E8B1  E1121B08  FB9A0F67  2ED9CE48  044BD618  3242087C  ADDDA392
         F2CA1F36  FDD94248  E8485D5E
```

In the second variant, the response to challenge, TokenAB_2, is $D = r - d$ x Q.

```
D =         5E8B1  E1121B08  FB9A0F4A  C9635817  3593FC82  92F57BC9  D38E3D03
         B7D17B20  924C0C92  49A9171E
```

Step g

Verification:

```
W* =           04  DAD48D02  4B83E223  4C0F5FFF  B51C15B7  1D52CF92  B35358CF
         FFE42756  843D0DF8  F3166971  E8AF6E22  6FD381B0  A816720F

h(W*) =  0EB01E5E  32CA889D  099C8F6E  4CC3CB08  A3CD6008  C2849B43  0E07BCC7
         B5241843
```

Authentication is valid.

C.2 ALIKE mechanism

The parameters of ALIKE are chosen for 80 bit-security.

The bit size for the modulus N is 1280 bits and for p_1 is 352 bits.

The block cipher E_K is chosen, for instance, from ISO/IEC 29192-2[27] or ISO/IEC 18033-3[26]. In this example, it is AES with a key-size u of 128 bits.

NOTE Since AES has a block size of 128 bits, u and v are equal in this example.

C.2.1 Key production

$e =$

```
0000000B
```

$p_1 =$

```
DD30D446   E32767CF   E14885E7   44D077D0   89F82A87   37F53C4D   36AA9463
7C250E7D   A516CA16   15C3B394   2B1CA791
```

$p_2 =$

```
B544FE3B   FB7D54D3   FA19B2E6   275CD79E   B09CC643   44C03C6C   268F3624
5989FECC   F44EC445   72A1F3C6   CD245A4D   4D17FDEC   0BF550D3   39C14EE8
4893CF1A   1E9BAF91   341AC6A9   E8B337B1   6B13B3A0   DF31E1A5   E5D63E70
0B93030D   BDAF9D6B   AFDBD696   6C1F09A0   95FA383C   32272D88   77A3F8FD
```

$N = p_1 \times p_2$

$=$

```
9C9F22B8   C7999ED9   54E7F600   63D134AB   6AF4BA29   046C2048   C7C0BC70
07686209   092D5B0B   BE6E2D88   2E76E9B2   D2A43371   29490102   2401CCE7
A0143B96   13B1727B   BC704892   F22B9EE6   A0C1F377   03229588   2EAC4879
3D88C4B3   800F5021   BAC0884C   A05EA932   38FD8D35   50F227C6   8DB51EFE
A8051C08   8D475FC4   9A563C02   9616FDD0   650C5B66   ED2E1EFD   84732F70
F6F1A24A   D5F88B5D   19864A5D   75F9124D
```

$t =$

```
C9151E11   E5C6BB77   29E4D6D2   3E8EF88F   091026A9   78B0655D   7783CCB7
8821B015   21B7A071   2B0F0058   27315283
```

C.2.2 Authentication exchange

Step a, b

k is a fresh string of random bits of length $u - 1$ used to calculate the commitment y.

$k =$

```
6C64D272   0B770A23   D5700C0B   EBC63E5E
```

$y = E_{f\_0(k)}(0)$

$=$

```
E85D2E05   D4C6592B   E571EE71   9BA636E7
```

Step c, d

r is a fresh string of random bits of length $u - 1$.

$r =$

```
6E5707FA   1F9171C1   D802C92C   605A3FD1
```

$1 \mid\mid r =$

```
EE5707FA   1F9171C1   D802C92C   605A3FD1
```

$pad =$ $HE(r)$ where $HE(r)$ is equal to the $L = 128$ left-most bits of $E_{f\_1(r)}(0)$.

$pad =$

```
B8C940AE   B22FDB93   7A1FE295   1584A26C
```

Verifier's challenge:

$d =$ $(\, r \,||\, pad \,)^{e} \bmod N$

```
  =      18240256  E10CFD25  725AD87B  7EBAFB43  81988968  B7D35E4F  6D75A201
         6480DFA6  B5E4E78A  EDE764E7  49CB5880  4BFA2A81  088ECFB3  3903AA0F
         31E3CE42  C653CA28  4F418EED  F76D6914  D6B40C9B  205A00E5  6C8008AC
         13FFD2F1  CA57FB8A  B6B57001  A5E3B04D  BBE14BB5  D5200511  20F744E4
         9B87B87E  7F411F3D  4657E4AF  A26E6D0B  F4414095  816D90CD  06CF6EE5
         6C244F17  F30CDB58  C6226D80  AEDC70F4
```

Step e, f

Response to challenge:

$D =$ $E_{f\_0(r)}(0 \,||\, k)$

```
  =      01203402  350C0611  F34C71BF  59F9CC3E
```

Step g

Verification.

C.2.3 Session-key derivation

Optionally, a session-key s_k could be established.

$s_k =$ $r \oplus k$

```
  =      0233D588  14E67BE2  0D72C527  8B9C018F
```

C.3 Identity-based signature mechanism

C.3.1 Key production

The elliptic curve for this example is

$$E\colon Y^2 = X^3 + aX + b \text{ over } F_q$$

with

```
a =      FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  7FFFFFFC

b =      1C97BEFC  54BD7A8B  65ACF89F  81D4D4AD  C565FA45

q =      FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  7FFFFFFF
```

Base point P over E.

$P =$ (x_P, y_P)

```
  =      (4A96B568  8EF57328  46646989  68C38BB9  13CBFC82,
          23A62855  3168947D  59DCC912  04235137  7AC5FB32)
```

n is the order of point P.

```
n =              1  00000000  00000000  0001F4C8  F927AED3  CA752257
```

t is the master secret key.

t = D21DF3A7 5787F180 5F00792F 9D8C317C 23FDF91B

T is the public point.

T = (x_T, y_T)

= (1B2F7E1F 831DF943 F82CFBE2 FF753A4C 9DF8040A,
1FFE799A 563024AF 86652027 CEA9A60A 00E1FB73)

ID is the identification information of the signer. $|\mathrm{ID}|$ = 8 bits.

ID = 01

r is a fresh number less than n.

r = 8A29A77B 8826FC67 2ABEA882 FEAEE9C3 6E1A78C2

R = $[r]P$
= (x_R, y_R) with $|x_R| = |y_R|$ = 160 bits
= (1040E9BF 14546E1B 38FC74B5 31228C69 AF0BAED3,
8DC50619 E3B28AEC B8296F17 51466289 D32053F6)

For hash-function h, this example makes use of SHA-1, i.e. the third hash-function specified in ISO/IEC 10118-3 [23].

s = $r + h(x_R \,||\, ID)$ x t mod n
= 49952E7E 4289DFA8 CE6ADB2F 55BA9C70 D89AA3C7

C.3.2 Sign

C.3.2.1 Example 1

m a message of 160 bits:

m = 00000000 00000000 00000000 00000A73 199606B1

Step a

y is a fresh number less than n.

y = 00000000 00000000 00000000 00000000 00000007

Step b

Y = $[y]P$
= (x_Y, y_Y) with $|x_Y| = |y_Y|$ = 160 bits

x_Y = 7A7F99D5 6472F619 577C4E8C 9B3A35E9 61472188

Step c, d

z = $y + h(x_Y \,||\, x_R \,||\, m)$ x s mod n
= 92D28A45 FFDE887E C8D297A2 7FA02CB5 7DF2CBAF

C.3.2.2 Example 2

Message:

m = 00000000 00000000 00000000 00000A79 19B70693

Step a

y is a fresh number less than n.

y = 00000000 00000000 00000000 00000000 00000010

Step b

$Y = [y]P$
$= (x_Y, y_Y)$

x_Y = B32F7DFA 2A82B99B 5CAC2772 AA6661BE 5F315034

Step c

$z = y + h(x_Y \,||\, x_R \,||\, m) \text{ x } s \bmod n$
= BB7A0E5A 805F67A6 CF00FF5A 0BF8B782 0803751E

C.3.3 Verify

C.3.3.1 Example 1

$c = h(x_Y \,||\, x_R \,||\, m)$
= 043969EF 9D9C6429 495139BD 8B37E086 FAA78FFB

$[z]P = (x_{[z]P}, y_{[z]P})$ with

$x_{[z]P}$ = 88913D78 4FD959FF 91E14157 D44799FA 674B2717

C.3.3.2 Example 2

$c = h(x_Y \,||\, x_R \,||\, m)$
= F8228399 413773F9 EB23CCA0 DFD1D416 D50941B7

$[z]P = (x_{[z]P}, y_{[z]P})$ with

$x_{[z]P}$ = E0B100B3 1CA6F0E7 251275A7 8B5F0BFB C6207A29