ISO/IEC JTC 1/WG 7
Working Group on Sensor Networks

Document Number:	N056
Date:	2010-07-05
Replace:	
Document Type:	Liaison Organization Contribution
Document Title:	Liaison Statement from JTC 1/SC 27/WG 2 to JTC 1/WG 7 on the ISO/IEC 2nd WD 29192-4
Document Source:	JTC 1/SC 27/WG 2
Document Status:	For consideration at the 2nd WG 7 meeting in US.
Action ID:	FYI
Due Date:	
No. of Pages:	23

ISO/IEC JTC 1/WG 7 Convenor:
Dr. Yongjin Kim, Modacom Co., Ltd (Email: cap@modacom.co.kr)
ISO/IEC JTC 1/WG 7 Secretariat:
Ms. Jooran Lee, Korean Standards Association (Email: jooran@kisi.or.kr)

ISO/IEC JTC 1/SC 27 N8759

REPLACES: N8226

ISO/IEC JTC 1/SC 27

Information technology - Security techniques

Secretariat: DIN, Germany

DOC TYPE:	text for working draft
TITLE:	Text for ISO/IEC 2nd WD 29192-4 -- Information technology -- Security techniques -- Lightweight cryptography - Part 4: Mechanisms using asymmetric techniques
SOURCE:	Project Editor (Matt Robshaw), Project Co-Editor (Jean-Francois Misarsky)
DATE :	2010-06-15
PROJECT:	**29192-4 (1.27.82.04)**
STATUS:	In accordance with resolutions 1 and 6 (contained in SC 27 N8789) of the 40th SC 27/WG 2 meeting held in Melaka (Malaysia) 19th – 23rd April 2010 , this document is being circulated to National Bodies and liaison organizations for STUDY AND COMMENT. The National Bodies and liaison organizations of SC 27 are requested to send their comments / contributions on the hereby attached document directly to the SC 27/WG 2 Secretariat sc27wg2-secretary@ipa.go.jp as soon as possible but no later than **2010-09-05**.
PLEASE NOTE:	For comments please use THE SC 27 TEMPLATE separately attached to this document.
ACTION:	**COM**
DUE DATE:	**2010-09-05**
DISTRIBUTION:	P-, O- and L-Members W. Fumy, SC 27 Chairman M. De Soete, SC 27 Vice Chair E. J. Humphreys, K. Naemura, M. Bañón, M.-C. Kang, K. Rannenberg, WG-Conveners
MEDIUM:	**http://isotc.iso.org/livelink/livelink/open/jtc1sc27**
NO. OF PAGES:	1 + 21

Secretariat ISO/IEC JTC 1/SC 27 -
DIN Deutsches Institut für Normung e. V., Burggrafenstr. 6, 10772 Berlin, Germany
Telephone: + 49 30 2601-2652; Facsimile:+ 49 30 2601-1723; E-Mail: krystyna.passia@din.de;
HTTP://www.jtc1sc27.din.de/en

ISO/IEC JTC 1/SC 27 N 8759

Date: 2010-06-15

ISO/IEC WD 29192-4

ISO/IEC JTC 1/SC 27/WG 2

Secretariat: DIN

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

Document type: International Standard
Document subtype:
Document stage: (20) Preparatory
Document language: E

C:\%Data\%SC27WG2@DocPrep\SC27N8759 2ndWD 29192-4 Tatsuta 20100617.doc STD Version 2.1c2

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.

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.

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.

ISO/IEC 29192-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *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 different 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 attractive. 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 around a variant of RSA, ALIKE offers a single-party 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 contribution offers what is termed an identity-based signature scheme. Here the infrastructure model involves a trusted third party in the computation of distinct signature keys. The functionality provided by the scheme is the possibility to sign arbitrary messages. This scheme offers implementation advantages over many other schemes in the cryptographic literature.

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:

— Mechanism based on discrete logarithms on elliptic curves and providing unilateral authentication.

— Authenticated Lightweight Key Exchange (ALIKE) mechanism for unilateral authentication and establishment of a session key.

— Identity-Based Signature scheme

2 Normative references

There are no normative references for this part of ISO/IEC 29192.

3 Terms and definitions

Fort the purposes of this document, the following terms and definitions 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 (the two operations have the property that, given the public operation, it is computationally infeasible to derive the private operation)

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

3.3
challenge
procedure parameter used in conjunction with secret parameters to produce a response

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

3.5
claimant parameter
public data item, number or bit string, specific to a given claimant within the domain

3.6
coupon
coupon is pre-computed numbers to be used only once; one shall be kept secret and the second shall remain secret until its use by an entity

3.7
domain
collection of entities operating under a single security policy, e.g., public key certificates created either by a single certification authority, or by a collection of certification authorities using the same security policy

3.8
domain parameter
public key, or function, agreed and used by all entities within the domain

3.9
entity authentication
corroboration that an entity is the one claimed

[ISO/IEC 9798-1:1997]

3.10
exchange multiplicity parameter
number of exchanges of information involved in one instance of an authentication mechanism

3.11
hash-function
function that maps strings of bits to fixed-length strings of bits, satisfying the following two properties:
— for a given output, it is computationally infeasible to find an input that maps to this output;
— it is computationally infeasible to find two distinct inputs that map to the same output

[ISO/IEC 10118-1:2000]

3.12
master secret key
data item that shall be kept secret and should only be used by the trusted server in accordance with the process of generation of signer private data

3.13
private key
private data item of an asymmetric pair, that shall be kept secret and should only be used by a claimant in accordance with an appropriate response formula, thereby establishing its identity

3.14
procedure parameter
transient public data item used in an instance of an authentication mechanism, e.g. a witness, challenge or response

3.15
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

3.16
random number
time variant parameter whose value is unpredictable

[ISO/IEC 9798-1:1997]

3.17
response
procedure parameter produced by the claimant, and processed by the verifier for checking the identity of the claimant

3.18
secret parameter
number or bit string that does not appear in the public domain and is only used by a claimant, e.g., a private key

3.19
sign
signature generation process that takes a message and a signing key of a signer to produce a signature

3.20
signer
entity with an unique bit string as an identity, including the functions and the private data necessary to engage in generation of a signature

3.21
signing key
data item given by the trusted server that shall be kept secret and should only be used by a signer in accordance with the process of generation of a signature

3.22
token
message consisting of data fields relevant to a particular communication and which contains information that has been produced using a cryptographic technique

3.23
unilateral authentication
entity authentication that provides one entity with assurance of the other's identity but not vice versa

[ISO/IEC 9798-1:1997]

3.24
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

3.25
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.

3.26
witness
procedure parameter that provides evidence of the claimant's identity to the verifier

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 $2^{i-1} \leq A < 2^i$ 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 The binary representation of a number A as a string of $|A|$ bits is straightforward. 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 [11]

d challenge (procedure parameter)

D response (procedure parameter)

e public exponent (domain parameter)

E_K block cipher encryption function with key K

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)

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

N composite modulus (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 $[-n]P = [n](-P)$

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

Q, Q_i private key (secret parameter)

r fresh random number or fresh string of random bits (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., $2^{\alpha-1} \leq \text{modulus} < 2^{\alpha}$, also denoted $\lceil \text{modulus} \rceil$ (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, …*} set containing the elements *a*, *b*, *c, …*

5 Mechanism based on discrete logarithms with respect to elliptic curve

5.1 General

This mechanism, cryptoGPS – also called GPS in the earlier cryptographic literature –, is due to Girault, Poupard, and Stern [5]. The revised name is now used so as to avoid confusion with the physical location service GPS and cryptoGPS is a zero-knowledge identification scheme that provides unilateral entity authentication. Several variants of GPS are outlined in ISO/IEC 9798-5 [13] 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 [15].

NOTE 1 This mechanism implements the elliptic curve variant [4] of the GPS [5] scheme due to Girault, Poupard and Stern. It allows use of the so-called LHW (Low Hamming Weight) variant [3] 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 an 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), so 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 coupon strategy is not used, every claimant shall have also 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 [14].

5.3 Key production

For claimant *A*, a fresh string shall be uniformly selected at random from the set {2, 3, …, *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 the opposite of 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 $2^{\delta-1} < \Delta \leqq 2^{\delta}$. 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 1 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 2 When the set of challenges is the interval [0, *Δ* – 1], then: *β* = *δ*.

NOTE 3 A challenge is said to be LHW (Low Hamming Weight) if there are at least *σ* – 1 zero bits between any two consecutive one bits in its binary representation.

NOTE 4 The definition of the public point *G(A)* differs slightly from that defined in ISO/IEC 9798-5 [13]. 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 numbers 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*.

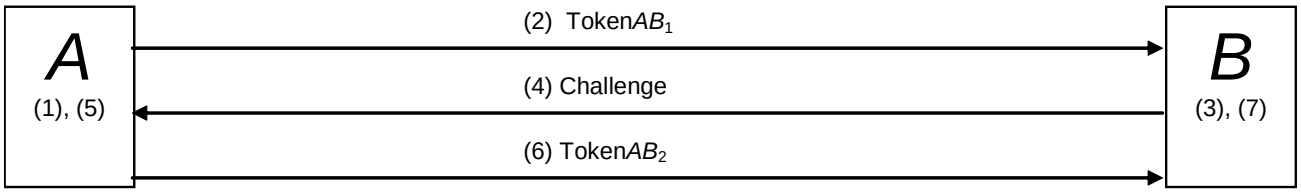

Figure 1 — Mechanism using a discrete logarithm with respect to elliptic curves

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

In the case of a coupon strategy, in addition to a number *δ* and a private key *Q*, the claimant shall only store a set of coupons. 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 [16]) 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, 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: $W = P2OS([r]P)$

NOTE 2 *P2OS* is the function used to convert a point to an octet string.

NOTE 3 Under certain implementation circumstances some might prefer to use the witness formula $W = P2OS([r \bmod n]P)$.

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, to *B*.

The four hash variants are $h(W \| Text)$, $h(W \| h(Text))$, $h(h(W) \| Text)$, 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 [12] 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 4 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 5a

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: $D = r + d \times Q$

NOTE 5 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.

f) *A* sends TokenAB_2 to *B*. TokenAB_2 is the response *D* computed from step 5)b).

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: $W^* = P2OS([d]G(A) + [D]P)$

NOTE 6 Under certain implementation circumstances some might prefer to use the verification formula $W = P2OS([d]G(A) + [D \bmod n]P)$.

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 (2), then the procedure is successful. Otherwise the procedure fails.

NOTE 7 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 Mechanism based on encryption scheme

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 [9] – a variant of RSA [8] – 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 [2].

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*, e.g. one of the algorithms specified in ISO/IEC 29192-2 [17]. 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.

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

e) All entities within the domain shall agree on padding function *HE*. The padding function depends on the block-cipher E_K and produces a padding-code of bit-length *l*. The value *L* shall be such that $L < v + 1$.

6.3 Key production

A number, denoted α, fixes the bit length of the modulus *n*, i.e. $2^{\alpha-1} <$ modulus $< 2^{\alpha}$, in accordance with the context of use of the mechanism. It is a domain parameter.

NOTE 1 The bit length of the modulus shall be chosen such that the complexity estimation of the fastest factorization algorithms [7], [10] – 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 $w > u + v$. *w* is a security parameter and it is a domain parameter.

NOTE 2 *w* is also the bit length of p_1 and shall be chosen such that the complexity estimation of the fastest known algorithm [6] – 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 a prime p_1 with $|p_1| = w$ such that

$$\gcd(e, p_1 - 1) = 1 \text{ with } e \text{ the public exponent}$$

NOTE 3 e shall be chosen high enough to avoid the Coppersmith [1] attack and compliant with the Shamir inferior bound [9]. The value $e = 11$ has some practical advantages.

b) Then generate a prime p_2 such that $|p_2| = \alpha - w$ and compute $N = p_1 \times p_2$.

c) Compute $t = e^{-1} \bmod (p_1 - 1)$. 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 and a public key P_A corresponding to the modulus n, see [9].

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

NOTE 4 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 numbers 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.

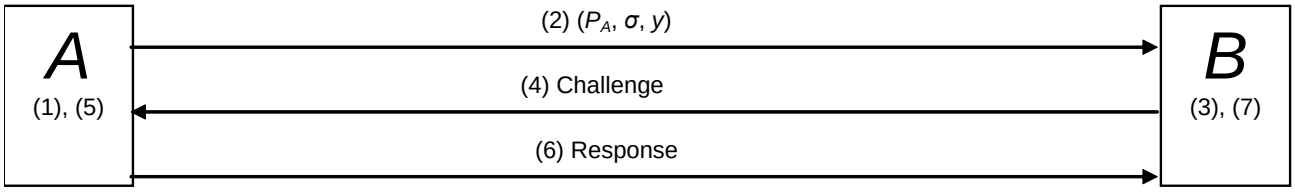

Figure 2 — ALIKE

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 shall be uniformly selected at random of length $u - 1$. The block-cipher is used to calculate a commitment y: $y = E_{0||k}(0)$

NOTE The first bit of the block-cipher key is set to 0. This permits to guarantee the independency from the block-cipher used to construct the function HE in which the first bit of the key should be set to 1.

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

c) A fresh number r shall be uniformly selected at random of length $u - 1$. The padding function HE is applied to r to get the padding value $pad = HE(r)$. $HE(r)$ is equal to the L left-most bits of $E_{1||k}(0)$.The message (r || pad) is encrypted with P_A. The result is the challenge.

d) B sends the challenge 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 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 with the block-cipher : $D = E_{0||r}(k)$

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 $E_{0||k'}(0) = y$, 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 by computing $r \oplus k$.

7 Mechanism based on the identity

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 for 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 described below is most suitable to constrained devices, as the signing stage only requires very light computation.

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 *q* 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 hash-function, e.g. one of the functions specified in ISO/IEC 10118-3 [14].

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

g) For each number i from the set [0,|q|-1], the trusted server shall compute the public points Y_i which are set equal to the multiplication of number -2^i by the base point P: $Y_i = [-2^i]P$

h) The trusted server shall keep t as the master secret key and publish the curve E, the points T, $\{Y_0,\ldots,Y_{|q|-1}\}$, the base point P, and the number q as the parameters.

7.3 Key production

Signer A shall ask the trust 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 q and compute the point R which is set equal to the multiplication of number r by the base point P:

$$R = [r]P$$

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

$$s = r + h(R \,||\, ID) \times t \bmod q$$

where h is a 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 $[s]P = R + [h(R \,||\, ID)]T$

7.4 Sign

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

a) A fresh number y shall be uniformly selected at random, non-zero and less than q.

b) Set $Y = [0]P$.

c) For $i = 1$ to $|q|$ compute:

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

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

$$z = y + h(Y \,||\, R \,||\, m) \times s \bmod q$$

e) 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(*Y* || *R* || *m*) and check if the following equality is hold.

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

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

Annex A
(normative)

Object identifiers

```
LightweightCryptographicMechanisms{
    iso(1) standard(0) light-crypto-mechanisms(29192)
    part(4) asn1-module(0) light-crypto-mechanisms (0)}
    DEFINITIONS ::= BEGIN
EXPORTS ALL;

OID ::= OBJECT IDENTIFIER -- alias
-- Synonyms
is29192-4 OID ::= {iso(1) standard(0 light-crypto-mechanisms(29192) part(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 -- LightweightCryptographicMechanisms
```

Annex B
(informative)

Test vectors

Bibliography

[1] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. Journal of Cryptology, 10:233--260, 1997.

[2] J-S. Coron, A. Gouget, P. Paillier and K. Villegas. SPAKE: a Single-party Public-key Authenticated Key Exchange Protocol for Contact-less Applications. In Proceedings of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010, January 2010

[3] M. Girault and D. Lefranc. Public key authentication with one (online) single addition. In CHES'04, pages 413-427, 2004

[4] M. Girault, L. Juniot, and M.J.B. Robshaw. The feasibility of on-the-tag public key cryptography. In RFIDSEC 2007, 11-13 july 2007

[5] M. Girault, G. Poupard, and J. Stern. On the fly authentication and signature schemes based on groups of unknown order. J. Cryptology, 19(4):463-487, 2006

[6] H. W. Jr. Lenstra. Factoring Integers with Elliptic Curves. Ann. Math. 126, 649-673, 1987.

[7] A. K. Lenstra and H. W. Lenstra, Jr. The development of the number field sieve. Lecture Notes in Math. (1993) 1554. Springer Verlag.

[8] R.L Rivest, A. Shamir and L. Adleman. A method for obtaining digital signature and public-key cryptosystems. In technical report LCS!TM82, MIT Laboratory for Computer Science, Cambridge, Massachusetts, 4th April 1977

[9] A. Shamir. RSA for paranoids. In Cryptobytes, the technical newsletter from RSA Laboratories, Volume 1, Number 3 – Autumn 1995

[10] European Network of Excellence ECRYPT, Yearly Report on Algorithms and Keysizes (2007-2008), available on http://www.ecrypt.eu.org/ecrypt1/documents/D.SPA.28-1.1.pdf

[11] ISO/IEC 8825-1:2002, *Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

[12] ISO/IEC 9798-1:1997, *Information technology — Security techniques — Entity authentication — Part 1: General*

[13] ISO/IEC 9798-5:2009, *Information technology — Security techniques — Entity authentication — Part 5: Mechanisms using zero-knowledge techniques*

[14] ISO/IEC 10118-3:2004, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

[15] ISO/IEC 15946-1:2008, *Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 1: General*

[16] ISO/IEC 18031:2005, *Information technology — Security techniques — Random bit generation*

[17] ISO/IEC 29192-2, *Information technology — Security techniques — Lightweight Cryptography — Part 2: Block ciphers*[1)]

1) To be published.