FSS-ID A Fast Safe, Identity Based, Aggregate Signature Scheme

Sami Harari

Laboratoire SNC, ISITV, Université du Sud Toulon Var BP 56, 83162 La Valette du Var cedex harari@univ-tln.fr

June 22,2009

Introduction 1

In this paper a signature scheme is presented. It is analogous to DSA/DSS but relies on the classical RSA problem for its cryptographic strength.

- It is a variant of the Guillou Quisquater scheme.

Introduction 1

In this paper a signature scheme is presented. It is analogous to DSA/DSS but relies on the classical RSA problem for its cryptographic strength.

- It is a variant of the Guillou Quisquater scheme.
- It use less exponentiations.

Introduction 1

In this paper a signature scheme is presented. It is analogous to DSA/DSS but relies on the classical RSA problem for its cryptographic strength.

- It is a variant of the Guillou Quisquater scheme.
- It use less exponentiations.
- It is well adapted to aggregation.

Introduction 1

In this paper a signature scheme is presented. It is analogous to DSA/DSS but relies on the classical RSA problem for its cryptographic strength.

- It is a variant of the Guillou Quisquater scheme.
- It use less exponentiations.
- It is well adapted to aggregation.
- It does not rely on bilinear mappings.

Introduction 2

- The exponentiation can be precomputed.
- The other part of the signature, which is message dependent is obtained by two modular multiplications and the computation of a hash.
- Another essential characteristic is that the verifying key can be chosen arbitrarily and therefore can be chosen as an ID string or an e-mail address.

Motivation

The need to elaborate new signature schemes **that can be aggregated**, fast and safe is motivated by many applications and constraints linked to the condition of use. The most evident ones are smart cards, contactless or mobile applications using RFID, electronic commerce protocols and voting protocols.

Motivation 2

To increase trust on signature, multi signature schemes have been developped. In order to distribute trust and not to rely on a single TTP multisignature is an advantage. This new presented signature scheme can be transformed into a multi signature scheme as will be shown.

Implementation Requirements

Let n be a RSA integer, that is a product of two large safe primes. All integers needed for implementation are less than n. Let k and k' be two RSA keys, that is two integers satisfying $k.k' = 1 \ mod \ \phi(n)$ The verifying quantity denoted by Id, is an integer, characteristic of the signer, which is unforgeable.
It is a string of characters to which is associated an integer less than n, through the use of a hash function for example. As an identifier one can take, depending on applications, an e mail address or a concatenation of an e mail address with the number of an Id card
These quantities are rendered unforgeable by a preliminary control and a very large diffusion to a certain number of public repositories and eventually to a trusted third party.

Implementation Requirements 2

The charateristic quantity of the signer with identification string Id, which is the secret key is the following:

$$s_i = Id^{(\phi(n)-1).k'}.$$

It will be kept secret. The inverse of the identity string $Id\ mod\ n$ can also be computed with other algorithms instead of the exponential.
In order to compute it, one must know the RSA secret key, the converse being false.

Implementation Requirements 3

The *public quantities* that help for checking the signature are the following.

$$(n, k, Id, h())$$

The integers n and k of the RSA and Id characteristic of the signer. A hash function $h()$ is also required. This function yields bit strings of length less than $log_2 n$.

Signing a Document

Let M a document to be signed, of arbitrary volume. The signature creation has three steps.

- Choose a random integer r and compute the first part of the signature $x = r^k \ mod \ n$.

Signing a Document

Let M a document to be signed, of arbitrary volume. The signature creation has three steps.

- Choose a random integer r and compute the first part of the signature $x = r^k \ mod \ n$.
- Compute a hash of the message M concatenated with the preceding quantity:
  $e = h(M||x)$

Signing a Document

Let M a document to be signed, of arbitrary volume. The signature creation has three steps.

- Choose a random integer r and compute the first part of the signature $x = r^k \ mod \ n$.
- Compute a hash of the message M concatenated with the preceding quantity:
  $e = h(M||x)$
- Compute
  $y = e \cdot r \cdot s_i \ mod \ n$

Signing a Document

Let M a document to be signed, of arbitrary volume. The signature creation has three steps.

- Choose a random integer r and compute the first part of the signature $x = r^k \ mod \ n$.
- Compute a hash of the message M concatenated with the preceding quantity:
  $e = h(M||x)$
- Compute
  $y = e \cdot r \cdot s_i \ mod \ n$
- The *signature* of message M by user with identity Id is the couple of integers (x, y)

Checking the signature

Let M be a message and (x, y) its signature, obtained with a RSA integer n and public key k, by a user with identity string Id. The control has three steps:

1. Compute the quantity $t = (y^k) \cdot Id\ mod\ n$

Checking the signature

Let M be a message and (x, y) its signature, obtained with a RSA integer n and public key k, by a user with identity string Id. The control has three steps:

1. Compute the quantity $t = (y^k) \cdot Id \; mod \; n$
2. Compute the quantities $e = h(M||x)$ and $z = e^k \cdot x \; mod \; n$

Checking the signature

Let M be a message and (x, y) its signature, obtained with a RSA integer n and public key k, by a user with identity string Id. The control has three steps:

1. Compute the quantity $t = (y^k) \cdot Id \ mod \ n$
2. Compute the quantities $e = h(M||x)$ and $z = e^k \cdot x \ mod \ n$
3. Check that $t = z$. If the quantities are equal accept the signature as valid, otherwise reject it.

Strength and Computational characterisics

- One exponentiation replaced by a modular multiplication.

Strength and Computational characterisics

- One exponentiation replaced by a modular multiplication.
- No exponentiation in the presence of data.

Strength and Computational characterisics

- One exponentiation replaced by a modular multiplication.
- No exponentiation in the presence of data.
- As strong as factoring the RSA integer.

Strength and Computational characterisics

- One exponentiation replaced by a modular multiplication.
- No exponentiation in the presence of data.
- As strong as factoring the RSA integer.
- Zero Knowledge.

Strength and Computational characterisics

- One exponentiation replaced by a modular multiplication.
- No exponentiation in the presence of data.
- As strong as factoring the RSA integer.
- Zero Knowledge.
- Possible Id Based implementaion.

The Multi Signature Id Based version (IBMS)

One must show how to initialize parameters, how to obtain key derivation, message signing and verifying the signature.

The Multi signature Id Based version (IBMS)

The Set up Procedure
The key distribution entity runs a generator procedure for RSA parameters (N, e, d), with e a prime integer, with length strictly greater than $log_2 N/4$.
It publishes $mpk = (N, e)$ the master public key. It keeps secret d the master secret key.
We suppose that there exists a hash function $h()$ with output strings of length at least 160 bits and less than $log_2\ N$.

IBMS Key Derivation

The identity of the signer denoted by Id, is an integer, characteristic of the signer, which is unforgeable. It can be obtained a string of characters to which is associated an integer less than n.
In order to eliminate the possibility of a weak key associated to a certain identity, a hash of the identity will be used instead of the identity itself.
The charateristic quantity of the signer number i with identification string Id_i is the following:
$x_i = h(Id_i)^{(\phi(n)-1).e} \ mod \ N.$
$\phi()$ is the Euler function, we suppose that there are n persons that will sign the message m. The secret key x_i of user i will be sent through a secure and authenticated channel.

IBMS Signing a Message (inter active rounds)

On input user with secret key x_1 for Id_1, message m and cosigners with identities $Id_2, ..., Id_n$, the signer proceeds as follows

Round 1

- Local Input: $x_1, L = (Id_1, ..., Id_n), m$

Round 2 -Receive from signer j : t_j
-Send R_1
for all j

IBMS Signing a Message (inter active rounds)

On input user with secret key x_1 for Id_1, message m and cosigners with identities $Id_2, ..., Id_n$, the signer proceeds as follows

Round 1

- Local Input: $x_1, L = (Id_1, ..., Id_n), m$
- Computation: choose a random r_1 in $\mathbf{Z}_N{}^*$, compute $R_1 = r_1{}^{x_1}\ mod\ N.$ and $t_1 = h(m||R_1||L)$.

Round 2 -Receive from signer j : t_j
-Send R_1
for all j

IBMS Signing a Message (inter active rounds)

On input user with secret key x_1 for Id_1, message m and cosigners with identities $Id_2, ..., Id_n$, the signer proceeds as follows

<u>Round 1</u>

- Local Input: $x_1, L = (Id_1, ..., Id_n), m$
- Computation: choose a random r_1 in $\mathbf{Z}_N{}^*$, compute $R_1 = r_1{}^{x_1}\ mod\ N$. and $t_1 = h(m||R_1||L)$.
- Send to each signer $j\ :\ t_1$.

<u>Round 2</u> -Receive from signer $j\ :\ t_j$
-Send R_1
for all j

IBMS Signing a Message 2

Round 3 - receive R_j
- computation: Check that $t_j = h(m||R_j||L)$.
Halt the computation and transmit a signal failure if one of the equations is not checked.
-Compute $R = \prod_i R_i \ mod \ N$.
-Compute $\theta_1 = r_1 \cdot m \cdot x_1 \ mod \ N$
and send to the set of $L - 1$ other participants in the multi signature.

Round 4
Receive from signer j: θ_j
-Computation $\theta = \prod_i \theta_i \ mod \ N$
Local output $\sigma = (R, \theta)$ σ, can be computed by each participant, is the multi signature of message M by the set of L participants.

IBMS Verification

On input of the master key (N, e) and of a signature $\sigma = (R, \theta)$, and a set of signers $L = (Id_1, ..., Id_n)$ and a message m the verifier computes

1. Compute the quantity $t = (\theta^e)\ mod\ N$

IBMS Verification

On input of the master key (N, e) and of a signature $\sigma = (R, \theta)$, and a set of signers $L = (Id_1, ..., Id_n)$ and a message m the verifier computes

1. Compute the quantity $t = (\theta^e)\ mod\ N$
2. Compute $u = R \cdot m^{e.n} \cdot \prod_i h(Id_i)\ mod\ N$

IBMS Verification

On input of the master key (N, e) and of a signature $\sigma = (R, \theta)$, and a set of signers $L = (Id_1, ..., Id_n)$ and a message m the verifier computes

1. Compute the quantity $t = (\theta^e) \; mod \; N$
2. Compute $u = R \cdot m^{e.n} \cdot \prod_i h(Id_i) \; mod \; N$
3. Check that $t = u$. If the quantities are equal accept the signature as valid, otherwise reject it.