A "Paradoxical" Indentity-Based Signature Scheme Resulting from Zero-Knowledge

Louis Claude Guillou [1] and Jean-Jacques Quisquater [2]

[1] Centre Commun d'Etudes de Télédiffusion et Télécommunications
CCETT, BP 59; F–35 512 Cesson-Sevigné Cédex, France

[2] Philips Research Laboratory Brussels
Avenue Van Becelaere, 2; B–1 170 Brussels, Belgium
E-mail: jjq@prlb2.uucp

ABSTRACT

At EUROCRYPT'88, we introduced an interactive zero-knowledge protocol (Guillou and Quisquater [13]) fitted to the authentication of tamper-resistant devices (*e.g.* smart cards, Guillou and Ugon [14]).

Each security device stores its secret *authentication number*, an RSA-like signature computed by an authority from the device identity. Any transaction between a tamper-resistant security device and a verifier is limited to a unique interaction: the device sends its *identity* and a random *test number*; then the verifier tells a random large *question*; and finally the device answers by a *witness number*. The transaction is successful when the test number is reconstructed from the witness number, the question and the identity according to numbers published by the authority and rules of redundancy possibly standardized.

This protocol allows a cooperation between users in such a way that a group of cooperative users looks like a new entity, having a shadowed identity the product of the individual shadowed identities, while each member reveals nothing about its secret.

In another scenario, the secret is partitioned between distinct devices sharing the same identity. A group of cooperative users looks like a unique user having a larger public exponent which is the greater common multiple of each individual exponent.

In this paper, additional features are introduced in order to provide: firstly, a mutual interactive authentication of both communicating entities and previously exchanged messages, and, secondly, a digital signature of messages, with a non-interactive zero-knowledge protocol. The problem of multiple signature is solved here in a very smart way due to the possibilities of cooperation between users.

The only secret key is the factors of the composite number chosen by the authority delivering one authentication number to each smart card. This key is not known by the user. At the user level, such a scheme may be considered as a keyless identity-based integrity scheme. This integrity has a new and important property: it cannot be misused, *i.e.* derived into a confidentiality scheme.

Keywords: cryptology, factoring, complexity, randomization, zero-knowledge interactive proofs, identity-based system, public key system, intégrity, identification, authentication, digital signature.

S. Goldwasser (Ed.): Advances in Cryptology - CRYPTO '88, LNCS 403, pp. 216-231, 1990.

1 Introduction

Some problems are very asymmetric: although only inefficient methods are known for solving these problems, any proposal is easily tested in order to know whether it is a solution or not. There are two methods in order to prepare an instance of such a complex problem:

— either you prepare the instance by yourself;

— or an authority does it for you, in relation with your identity.

In the *first method*, each user picks a *trap* at random, and then deduces the text of a problem having this trap as solution. This method leads to systems where each user has his own secret key. An authority manages the system by registering the users and their public keys in a publicly available register.

> Factoring large integers is a pretty well known example of such a complex problem. The following operations are rather easy to do: selecting at random two large prime integers and computing their product. But only inefficient methods are known to factor large composite integers. Outside number theory, many other complex problems are available.

In the *second method*, each user relies upon a trusted authority, like a bank, a credit card company, a telephone operator or a transportation authority; after the signature of a contract specifying the rights and obligations of each party, the authority delivers to the new user a tamper-resistant security device, *e.g.* a smart card, storing a secret identity-based authentication number. This alternate method leads to a *keyless* system. Only the authority has a secret key while each card holds its own authentication number which is not a trap. Other identity-based systems have been investigated (Shamir [17], Desmedt and Quisquater [4], Quisquater [15]), but our approach is different: here we are authenticating the security device only, not its holder.

How, without revealing it, can a tamper-resistant security device convince any verifier that it knows the authentication value corresponding to its identity?

According to the zero-knowledge techniques (Goldwasser, Micali and Rackoff [8], Goldreich, Micali and Wigderson [10]), the device convinces the verifier without revealing anything on the specific value of the authentication number which remains thus an efficient identification element as long as the secret is unrevealed and as long as the (instance of the) problem remains unsolved. The *knowledge* of the authentication number makes the difference between the tamper-resistant device and the outside.

After an *interactive process*, the verifier has nothing else but an intimate conviction which cannot be transmitted to anybody else. The interactive process may be used, not only to check the identity of the device, but also to check messages endorsed by the device. This method of proof is "non-transitive".

After a *non-interactive process*, like a signature, the verifier is convinced and can convince a judge that a genuine device signed the message. A knowledge is clearly transmitted along with each signature; but while proving that the device knows its authentication number, the signature still transmits no knowledge at all on specific value of the underlying authentication number which may be used indefinitely as an identification element of the device.

The zero-knowledge techniques are very efficient in various processes aiming at protecting the integrity of data ans systems:

identification, authentication and signature.

2 The GQ authentication scheme

We found an interactive protocol aiming at verifying the presence of a secret authentication number in a tamper-resistant security device claiming its identity (Guillou and Quisquater [13]).

Each tamper-resistant security device (*e.g.* a smart card) holds its unique authentication number B related to its identity I by the following simple equation:

$$B^v \cdot J \bmod n = 1, \text{ with } J = Red(I),$$

where,

n: is a composite number;

v: is an exponent, both published by the authority and known to each verifier;

J: is the "shadowed" identity of the device, that is to say a number as large as n, including the claimed identity I, half shorter than n, completed by a redundancy (the shadow) depending on I (Guillou and Quisquater [11], Guillou, Davio and Quisquater [12]). Redundancy rules *Red* (or how constructing J from I) are published or preferably standardized.

NOTE: Let us mention that ISO is standardizing a "digital signature scheme with shadow " (see ISO-DP 9796) in the Working Group JTC1/SC20/WG2 (public-key techniques).

The authentication transaction between the verifier and the device is limited to a unique interaction, which was not the case with the previous proposals (Fiat-Shamir [5], [6]). Here is the interactive protocol described in [13]:

1. The card I transmits its identity I and a *test number* T which is the v^{th} power in Z_n of an integer r picked at random in Z_n^*.

2. The verifier asks a *question* d which is an integer picked at random from 0 to $v-1$.

3. The card I sends a *witness number* t which is the product in Z_n of the integer r by the d^{th} power of the authentication number B.

4. In order to verify such a witness number t, the verifier computes the product of the d^{th} power of the shadowed identity J by the v^{th} power of witness t, that is:

$$\begin{aligned} J^d \cdot t^v \bmod n &= J^d \cdot (r \cdot B^d)^v \bmod n \\ &= (J \cdot B^v)^d \cdot r^v \bmod n \\ &= T. \end{aligned}$$

The proof of security relies on three basic facts:

— A device knowing the authentication number can easily answer correctly any question.

— A lucky guesser has an evident winning strategy by choosing first any witness number before deducing a test number according to the guessed question.

— Knowing two correct witness numbers according to any two different questions for the same test number (anyone) reveals the authentication number.

Let us define a *cheater* as a device trying to fool the verifier, while not knowing the specific value of B.

On one hand, any cheater having guessed the question d can obviously prepare a good looking pair T and t by, firstly, picking t at random in Z_n and, secondly, deducing T by computing exactly as the verifier will do.

On the other hand, having two witnesses t' and t'' corresponding to two different questions d' and d'' for the same test number T gives a significant (and generally total) knowledge about the authentication number B (see the proof in the next section).

Any cheater is thus able to prepare in advance exactly one witness number (at least one, but not two). A lucky cheater thus fools the verifier by guessing one question amongst v possible questions. At each transaction, the verifier has $(v-1)$ chances on v to defeat a cheater. Thus, when the size of v, also named *depth of the authentication number*, is sufficient to reach directly the required level of security, there is no need to repeat the interaction.

In the GQ scheme, the size of required memory and the volume of transmitted data are reduced to *minimum minimorum*. It is well fitted to smart card authentication.

3 Security of the GQ scheme

Now let us consider more precisely the conditions on v and the factors of n in the GQ scheme. Let us consider that n has only two prime factors: p and q.

Let us consider that v is an odd integer which is an RSA-like exponent, so that: $\gcd(p-1,v) = \gcd(q-1,v) = 1$. The case where v is an even integer integer will be considered in the full paper; the exponent v may even be a power of two.

Let us consider carefully the verification formula when v is an RSA-like exponent:

$$\mathcal{F}_d(t) = J^d \cdot t^v \bmod n.$$

A collision is a set of four integers:

$$\{t', t'', d', d''\} \quad t', t'' \text{ in } Z_n^*; \quad 0 \leq d'' < d' \leq v-1$$

such that,

$$\mathcal{F}_{d'}(t') = \mathcal{F}_{d''}(t'')$$

which is,

$$J^{d'} \cdot t'^v \bmod n = J^{d''} \cdot t''^v \bmod n$$

and may be transformed in

$$J^{(d'-d'')} \cdot (t'/t'')^v \bmod n = 1.$$

According to the Bezout formula, there exists a unique pair of integers k, $0 \leq k \leq v-1$, and m, $0 \leq m \leq d'-d''-1$, easily computed by the Euclidean algorithm, such that:

$$m \cdot v - k \cdot (d'-d'') = \pm \gcd(v, d'-d'').$$

Let us raise the equation to the power k and substitute.

$$\begin{aligned} 1 &= J^{k\cdot(d'-d'')} \cdot (t'/t'')^{k\cdot v} \bmod n \\ &= J^{m\cdot v \pm \gcd(v,d'-d'')} \cdot (t'/t'')^{k\cdot v} \bmod n \\ &= J^{\pm \gcd(v,d'-d'')} \cdot \{J^m \cdot (t'/t'')^k\}^v \bmod n. \end{aligned}$$

Thus:

$$B^{\pm \gcd(v,d'-d'')} = J^m \cdot (t'/t'')^k \bmod n.$$

When v is prime, any collision provides B. When v is composite, generally any collision provides B as well, and in some cases, a partial knowledge of B is obtained as a power of B of a rank dividing v.

Knowing any collision in $\mathcal{F}$ is thus equivalent to knowing B or a power of B of a rank dividing v.

For a given user, J and v are fixed: the function $\mathcal{F}$ from t to $\mathcal{F}_d(t)$ is a set of permutations of Z_n indexed by d, $0 \leq d \leq v-1$.

In a way similar to what is done in the GMR scheme (Goldwasser, Micali and Rivest [9]), by composing the basic permutation $\mathcal{F}$ indexed by d, $0 < d < n/2$, a large family of permutations $\mathcal{F}$ indexed by D may be constructed. Let D be an integer written on k v–ary digits, from the most significant one $d(k-1)$ to the least significant one $d(0)$, where k is the integer such that $v^{k-1} \leq D < v^k$:

$$\mathcal{F}_D(x) = \mathcal{F}_{d(0)}(\mathcal{F}_{d(1)}(...\mathcal{F}_{d(k-1)}(x)...)) = J^D \cdot x^{v^k} \bmod n.$$

Knowing any collision in this composed family leads generally to knowing the solution β to the equation:

$$J \cdot \beta^{v^k} \bmod n = 1.$$

The authentication number B, such that $J \cdot B^v \bmod n = 1$, is easily deduced from β.

Collision-resistance of this set is equivalent to computing the authentication number B by inverting an RSA instance ([16]).

4 Protocols of cooperation between entities

4.1 Entities with same exponent and different identities

Let us consider two tamper-resistant security devices, each one storing its unique authentication number (B_1 or B_2) related to its identity (I_1 or I_2) by the following equations:

$$B_1^v \cdot J_1 \bmod n = 1, \text{ with } J_1 = Red(I_1),$$
$$B_2^v \cdot J_2 \bmod n = 1, \text{ with } J_2 = Red(I_2).$$

The two entities, cooperating on a *shared* Personal Computer, are negotiating an authentication transaction with a verifier according to the following protocol:

1. Entity I_1 transmits its identity I_1 and a test number T_1 which is the v^{th} power in Z_n of an integer r_1 picked at random in Z_n^*.

   Entity I_2 transmits its identity I_2 and a test number T_2 which is the v^{th} power in Z_n of an integer r_2 picked at random in Z_n^*.

   The Personal Computer sends to the verifier the two identities I_1 and I_2 and the *common test number* T computed from:

   $$\begin{aligned} T &= T_1 \cdot T_2 \bmod n \\ &= (r_1 \cdot r_2)^v \bmod n \\ &= r^v \bmod n \end{aligned}$$

   where r is used for the (implicit) common random number $r_1 \cdot r_2 \bmod n$.

2. The verifier asks a *question* d which is an integer picked at random from 0 to $v-1$.

3. Entity I_1 sends a witness number t_1 which is the product in Z_n of integer r_1 by the d^{th} power of authentication number B_1.

   Entity I_2 sends a witness number t_2 which is the product in Z_n of integer r_2 by the d^{th} power of authentication number B_2.

The Personal Computer sends to the verifier the *common witness number* t:

$$\begin{aligned} t &= t_1 \cdot t_2 \bmod n \\ &= (r_1 \cdot B_1^d) \cdot (r_2 \cdot B_2^d) \bmod n \\ &= (r_1 \cdot r_2) \cdot (B_1 \cdot B_2)^d \bmod n \\ &= r \cdot (B_1 \cdot B_2)^d \bmod n. \end{aligned}$$

4. In order to check such a witness number t, the verifier computes the product of the d^{th} power of the shadowed identity J_1 and J_2 by the v^{th} power of witness t, that is:

$$\begin{aligned} J_1^d \cdot J_2^d \cdot t^v \bmod n &= J_1^d \cdot J_2^d \cdot (r_1 \cdot B_1^d \cdot r_2 \cdot B_2^d)^v \bmod n \\ &= (J_1 \cdot B_1^v)^d \cdot (J_2 \cdot B_2^v)^d \cdot r^v \bmod n \\ &= T. \end{aligned}$$

This protocol of cooperation, easily extensible to any number of cooperating entities, indicates a new direction in multiple signature schemes.

4.2 Two entities with the same identity and different exponents

Let us now consider two tamper-resistant devices, each one storing its unique authentication number (B_1 and B_2) related to the same identity I by one of the following simple equations (let us consider that v_1 and v_2 are prime together):

$$B_1^{v_1} \cdot J \bmod n = 1 \text{ and } B_2^{v_2} \cdot J \bmod n = 1, \text{ with } J = Red(I).$$

The cooperation may simulate an entity having identity I with the exponent $v = v_1 \cdot v_2$,

$$B^v \cdot J \bmod n = 1,$$

with B_1 equal to $B^{v_2} \bmod n$ while B_2 is equal to $B^{v_1} \bmod n$.

The two entities, cooperating on a shared Personal Computer, are negociating an authentication transaction with a verifier according the following protocol:

1. Entity 1 transmits its identity I and a test number T_1 which is the v^{th} power in Z_n of an integer r_1 picked at random in Z_n^*.

   Entity 2 transmits its identity I and a test number T_2 which is the v^{th} power in Z_n of an integer r_2 picked at random in Z_n^*.

   The shared Personal Computer sends to the verifier the common identity I and the *common test number* T computed from:

$$\begin{aligned} T &= T_1^{v_2} \cdot T_2^{v_1} \bmod n \\ &= (r_1 \cdot r_2)^{v_1 \cdot v_2} \bmod n \\ &= (r_1 \cdot r_2)^v \bmod n \\ &= r^v \bmod n, \end{aligned}$$

where r is used for the (implicit) common random number $r_1 \cdot r_2 \bmod n$.

2. The verifier asks a *question* d which is an integer picked at random from 0 to $v-1$.
   The shared Personal Computer translates the question: $d_1 = d/v_2 \bmod v_1$ for the entity 1 and $d_2 = d/v_1 \bmod v_2$ for the entity 2.

3. Entity 1 sends a witness number t_1 which is the product in Z_n of integer r_1 by the d_1^{th} power of authentication number B_1.

   Entity I_2 sends a witness number t_2 which is the product in Z_n of integer r_2 by the d_2^{th} power of authentication number B_2.

   The Personal Computer sends to the verifier the *common witness number* t:

$$\begin{aligned} t &= t_1 \cdot t_2 \bmod n \\ &= r_1 \cdot r_2 \cdot B_1^{d_1} \cdot B_2^{d_2} \bmod n \\ &= r \cdot B^{d_1 \cdot v_2 + d_2 \cdot v_1} \bmod n. \end{aligned}$$

4. Let us call d' the integer $d_1 \cdot v_2 + d_2 \cdot v_1$. In order to check such a witness number t, the verifier computes the product of the d^{th} power of the shadowed identity J by the v^{th} power of witness t, that is:

$$\text{Is the test number } T \text{ equal to } J^{d'} \cdot t^v \bmod n \ ?$$

   Proof:

$$\begin{aligned} J^{d'} \cdot t^v \bmod n &= J^{d_1 \cdot v_2 + d_2 \cdot v_1} \cdot (r_1 \cdot B_1^{d_1} \cdot r_2 \cdot B_2^{d_2})^{v_1 \cdot v_2} \bmod n \\ &= (J \cdot B_1^{v_1})^{d_1 \cdot v_2} \cdot (J \cdot B_2^{v_2})^{d_2 \cdot v_1} \cdot (r_1 \cdot r_2)^v \bmod n \\ &= T. \end{aligned}$$

□

This protocol of cooperation may easily be extended to any number of cooperating entities.

Let us remark that the protocols of cooperation solve many problems of subliminal channels in the sense of Simmons or Desmedt. One cooperating entity is then a one-way active warden (see more in the full paper).

5 Interactively authenticating both cards and messages

The authentication described in the basic method convinces the verifier that an entity knowing the authentication number is involved in the transaction.

But the interaction of simultaneous processes may be misleading: everybody knows the strategy used by the child playing chess simultaneously against two masters. The first master opens the first play, then the child reproduces this opening

on the second table. The second master replies, and the child repeats this reply on the first table. While knowing nothing in chess skill, the child will not loose both plays. We must be careful in the design of a protocol, so as to avoid to give to a child the merits of a master.

Let us transpose the problem. A kitchener using a security device provided by a banker is buying oranges at a grocery, the grocer being a member of the Organization: at the same time, another member of the same Organization is negociating diamonds in a jewelry, the jeweler being unaware of any problem. When the payment operation is ready, the jeweler verifies the authenticity of the security device of the man buying diamonds. But in fact, this "security" device is connected via a full duplex radiating channel to the grocery POS terminal. And owing to this hidden synchronization, the jeweler is preparing a bill on kitchener's account number, both kitchener and jeweler being unaware of the problem. Y. Desmedt noted this problem in the rump session of CRYPTO '87.

By linking transaction purpose and buyer identity in a unique authentication process, the fraud prepared by the Organization will no more succeed. The kitchener is buying oranges, while the jeweler is selling diamonds. This message authentication must convince the verifier that the message is really sent by the entity owning the right authentication number.

Such an extension implies a hash function. Some papers (Goldreich, Goldwasser and Micali [7]) are dealing with functions *statistically undistinguishable* from really random functions *with polynomially limited resources.* Let us suppose that such a good one-way hash function h exists, while, today, no such a function is ready for standardization.

NOTE. Hash functions h may be implemented either in prover's PC or in the card. The user must control the parameters sent to the hash function. In the example, the user holds a portable device in which the card is inserted and where the hash function h is implemented.

This is a message authentication (the basic idea was already present in Fiat-Shamir [6]):

1. The user claims the message M, the identity I and the verification number V.
   At each treatment, the card picks at random an integer r in Z_n and computes a test T by raising it to the v^{th} power in Z_n. The portable device of the user computes as the verification number V the hashing of M and T:
   $$V = h(M,T) = h(M, r^v \bmod n).$$

2. The verifier asks a question d.
   The verifier picks at random an integer d from 0 to $v-1$ and transmits it.

3. The user shows a witness t.
   The card computes as witness t the product of random elements r by the d^{th}

power of the authentication number B:

$$t = r \cdot B^d \bmod n.$$

4. The verifier reconstructs the test number T from the question d, the identity I and the witness t. Next, the verifier reconstructs the verification number V from the message M and the test T:

$$\text{Is } V \text{ equal to } h(M, J^d \cdot t^v \bmod n) \text{ ?}$$

This is still a zero-knowledge interactive protocol.

Let us now introduce a non-interactive zero-knowledge protocol: the hash function h may be used by the prover himself to compute directly the *question* d. Some of these ideas on non-interactivity were already formulated in Fiat and Shamir [6].

6 Swapping to signatures by removing interactivity

The integrity of a transmission system is threatened in various ways:

- false information may be introduced in the system;
- a wire-tapped message may be replayed;
- the sender may be impersonated;
- false signature may be forged.

By a *signature operation*, the sender prepares a signed message.
By a *verification operation*, the receiver checks the signed message.
When the integrity is threatened, at least the receiver must protect his operation.

Each operation may be described
as an algorithm controlled by parameters such a key.
In order to protect an operation, the key at least should be kept secret.

Each signature scheme implies three fundamental operations ([12]): the *key production*, the *signature* and the *verification*. In each signature system, there are five types of partners: the *prover*, the *verifier*, the *cheater*, the trusted *authority* managing the identities of users and hot lists and the *judge* evaluating disputes and repudiations.

In an interactive authentication process, the verifier reacts in a random way. Let us use a hash function to replace the interactivity between the prover and the verifier. We are facing now a signature scheme based on a non-interactive zero-knowledge technique. Our contribution in this field is not the basic ideas ([8], [17]), but rather a first synthesis between two basic ideas.

Let us consider the security level (related to the value of v): in a proximity relation with a policeman, nobody will try to show a forged driving licence with probability $1 - 10^{-4}$ of being caught. Some people may try up 10 000 times to remotely access to a database, and in a remote control, the question must then be 20 bit long. But in a signature scheme where a simulation may be secretly forged off-line, the level of security must be raised to 60 bit long questions. Even with the most powerful computers, it is unrealistic to try 10^{18}.

Here is the *signature operation*:

1. At each signature, the card picks at random an element of Z_n, and computes as the *test* T the v^{th} power of r in Z_n, transmitted to the PC.

2. The PC (or the card depending upon the application) hashes the message M and the test T in an integer d uniformly selected from 0 to $v-1$. This integer is transmitted to the card as the question d.

3. The card computes as the *witness* t the product in Z_n of the integer r by the d^{th} power of the authentication number B. The consecutive computations are

$$T = r^v \bmod n; d = h(M,T); t = r \cdot B^d \bmod n.$$

The signed message consists of the message M followed by a very compact appendix including the identity I, the question d and the witness t.

The *verification operation* consists of reconstructing the test T from the witness t, the question d and the identity I, knowing n, v and the redundancy rules.

This method is still *zero-knowledge about the authentication number* included in the card. Even an enemy using a stolen card, while producing signatures, will learn nothing about the specific value of the authentication number. Ans when a card is hashing itself M and T, the property is still maintained, because the same hashing should have been done outside the card. While making forgery easy, a weak hash function should not endanger the secret.

7 The identity-based signature scheme

This signature scheme *with appendix* is a probabilistic scheme based upon an underlying signature scheme *shadow* ([12]).

The underlying signature scheme is based on user's identities. For a bank, such an identity includes an account number, a validity period and a usage code, associated with the serial number of the chip embedded in the card.

We now propose to use as hash function the collision-resistant permutations analyzed in the second paragraph and related to the underlying signature scheme with shadow. Thus the security of the hash function is homogeneous with the security of the zero-knowledge scheme.

Resulting from hashing the message M and the test T, the question d is an element of Z_n. A shortening of the question d should result in a partial collision in Z_n, which does not give the authentication number. The proof of equivalence would thus disappear. In order to accept such large questions, v is a prime between $v/2$ and n.

The resulting signature scheme is *paradoxical*:

- An enemy having received as many signatures of messages of his choice as he wants is not able to produce only one additional signature unless he has broken the underlying problem and reconstructed the authentication number of the user.

- A user trying to repudiate one signature by producing a second message with the same appendix, should reveals a collision and thus his authentication number.

This is the *signature scheme*:

1. At each signature, the card picks at random an integer r in Z_n and computes as *test* T the p^{th} power of r in Z_n, transmitted to the PC.

2. The PC hashes the message M and the test T by computing as question d the product in Z_n of the M^{th} power of J by the v^{th} power of T ($v^{k-1} \leq M \leq v^k$).

3. The card computes as *witness* t the product in Z_n of r by the d^{th} power of the authentication number B.

Let us summarize these successive computations (k is such that $v^{k-1} \leq M \leq v^k$):

$$T = r^v \bmod n; d = J^M \cdot T^{v^k} \bmod n; t = r \cdot B^d \bmod n.$$

NOTE. At each signature, the integer r is picked at random in Z_n. In a practical implementation in a smart card, the random generation is difficult to control. A deterministic production of r should be very useful. How to specify a secure deterministic generation of r? Such a computation should imply both the authentication number B as a secret seed and the whole message M to be signed which should include at least a time stamp.

In such an implementation, for security reasons, the whole process should be performed inside the card, like this one:

At each signature, the card receives as argument a message M to be signed.

1. From this message M and from the authentication number B, the card generates an integer r in Z_n.

2. The card raises the integer r to the v^{th} power in Z_n to get the test number T.

3. The card computes as the *question* d the product in Z_n of the M^{th} power of J by the $(p^k)^{\text{th}}$ power of T ($v^{k-1} \leq M \leq v^k$).

4. The card computes as the *witness* t the product in Z_n of r by the d^{th} power of the authentication number B. After this sequence, the cards delivers the question d and the witness t.

The *verification operation* includes the successive reconstructions of the test T and the question d.

1. The test T is reconstructed as the product in Z_n of the d^{th} power of the shadowed identity J by the V^{th} power of the witness t.

2. The question d is reconstructed as the product in Z_n of the M^{th} power of the shadowed identity J by the $(v^k)^{\text{th}}$ power of the test number T.

Let us summarize these computations:

$$J^d \cdot t^v \bmod n = J^d \cdot (B^d \cdot r)^v \bmod n = (J \cdot B^v)^d \cdot r^v \bmod n = r^v \bmod n = T,$$

and,

$$d = J^M \cdot T^{(v^k)} \bmod n.$$

The whole verification collapses in a simple equation:

$$\text{Is the question } d \text{ equal to } J^{M+d \cdot v^k} \cdot t^{v^{k+1}} \bmod n \text{ ?}$$

8 Exchange authentication: *a priori* versus *a posteriori*?

Some proposals are made today to standardize authentication protocols beginning by an authentication sequence keying a pair of communicating entities. Subsequently doing the difference with the other entities on the network, this key ensures integrity of subsequent transmissions by ciphering either exchanged data or at least an imprint computed by hashing these data. This shared key must be kept secret. *A priori* authentication is mandatorily a procedure establishing a shared secret key in the pair of communicating entities. Such methods sadly confuse integrity and confidentiality while public key techniques seem powerful to provide separate solutions to the two classes of threatens against confidentiality on one hand and integrity on the other hand.

When a *priori* authentication is needed to limit misusing of gate resources by intruders, this authentication should not be used to ensure integrity of subsequent exchanges. But a second (a *posteriori*) authentication should rather be performed after the exchanges in order to check both integrity of previous exchanges and identification of communicating entities.

Operation sequencing is correct only in an a *posteriori* authentication when the authentication protocol occurs after the exchange of information. The zero-knowledge techniques are typically used after an exchange of clear information.

In a *posteriori* authentication, another subtlety appears between:

- *keyed systems*, where each user owns his secret key, like a composite number, usable for general purposes with the help of a registration authority.
- and *keyless systems*, based upon identities, where each user owns an authentication number delivered by a trusted authority for some dedicated purposes.

In a keyed system, confidentiality and integrity are both provided. The only solution (the RSA scheme) proposed today in CCITT X.509 (authentication framework) is in this category. While being useful in some circumstances, such a method is not dedicated to integrity: confidentiality is easily obtained.

In a keyless identity-based system, the communicating entities are not able to produce a common secret key: secrecy cannot be derived from the scheme.

Let us notice that in both cases, an authority (either a general multi-purpose authority or several dedicated authorities) has to play a prominent part! The keyless systems with multiple authorities fit better with the bright proposals of Chaum ([2]) on privacy protection. This is also an important point!

Integrity techniques are typically used on various remote control sysytems in such a way that no assumption has to be done on the security of networks and terminals used in the transaction. Why some assumptions should be done on the morality of potential users?

It seems to us that a *good* integrity scheme does not have to do any assumption on the integrity of the potential users.

Thus the conjunction of zero-knowledge techniques and identity-based techniques solves some political problems due to the use of cryptologic techniques on public networks. At least one signature scheme exists which cannot be misused and illegally transformed into a confidentiality scheme.

An identity-based scheme should be taken into account in X.509.

References

[1] Gilles Brassard, David Chaum and Claude Crépeau, *Minimum disclosure proofs of knowledge*, July 1987.

[2] David Chaum, *Security without identification: transaction systems to make Big Brother obsolete*, Comm. of ACM, **28**, Oct. 1985, pp. 1030–1044.

[3] Ivan Bjerre Damgård, *Collision-free hash functions and public-key signature schemes*, EUROCRYPT '87, to appear.

[4] Yvo Desmedt and Jean-Jacques Quisquater, *Public-key systems based on the difficulty of tampering*, Advances in cryptology, Proceedings of CRYPTO '86, Lectures notes in computer science, N° 263, Springer-Verlag, pp. 186–194.

[5] Amos Fiat and Adi Shamir, *How to prove yourself: practical solutions to identification and signature problems*. Springer Verlag, Lecture notes in computer science, N° 263, Advances in cryptology, Proceedings of CRYPTO '86, pp. 186–194, 1987.

[6] Amos Fiat and Adi Shamir, *Unforgeable proofs of identity*, 5th SECURICOM, Paris, 1987, pp. 147–153.

[7] Oded Goldreich, Shafi Goldwasser and Silvio Micali, *How to construct random functions*, 25th, IEEE symposium on foundations of computer science, 1984, pp. 464–479.

[8] Shafi Goldwasser, Silvio Micali and Charles Rackoff, *The knowledge of interactive proof systems*, 17th ACM symposium on theory of computing, 1985, pp. 291–304.

[9] Shafi Goldwasser, Silvio Micali and Ronald Rivest, *A paradoxical signature scheme*, 25th IEEE symposium on foundations of computer science, 1984, pp. 441–448.

[10] Oded Goldreich, Silvio Micali and Avi Wigderson, *Proofs that yields nothing but the validity of the proof*, Workshop on probabilistic algorithms, Marseille, March 1986.

[11] Louis C. Guillou and Jean-Jacques Quisquater, *Efficient digital public-key signatures with shadow*, Springer Verlag, Lecture notes in computer science, Advances in cryptology, Proceedings of CRYPTO '87, p. 223.

[12] Louis C. Guillou, Marc Davio and Jean-Jacques Quisquater, *Public-key techniques*, Cryptologia, to appear.

[13] Louis C. Guillou and Jean-Jacques Quisquater, *A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory*, EUROCRYPT '88, to appear.

[14] Louis C. Guillou and Michel Ugon, *Smart card: a highly reliable and portable security device*, CRYPTO '86, Lecture notes in computer science, N° 263, Springer-Verlag, pp. 464-479.

[15] Jean-Jacques Quisquater, *Secret distribution of keys for public-key system*, Springer Verlag, Lecture notes in computer science, N° 293, Advances in cryptology, Proceedings of CRYPTO '87, pp. 203–208, 1987.

[16] Ronald Rivest, Adi Shamir and Leonard Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. of ACM, **21**, Feb. 1978, pp. 120–126.

[17] Adi Shamir, *Identity-based cryptosystems and signatures schemes*, Springer Verlag, Lecture notes in computer science, N° 196, Advances in cryptology, Proceedings of CRYPTO '84, pp. 47–53, 1985.

[18] H. C. Williams, *A modification of the RSA public-key cryptosystem*, IEEE Trans. on Information Theory, **IT-26**, Nov. 1980, pp. 726–729.