METHOD AND DEVICE FOR AUTHENTICATING WITH SYMMETRICAL ALGORITHM

The present invention relates to a method for authenticating with symmetrical algorithm having the main characteristic that, for each mutual authentication of two parties, commonly referred to as A and B, a cryptographic computation with a variable key, referred to as K', is carried out.

The invention more precisely relates to the cryptographic computations cited above during an authentication of the two parties, referred to as A and B, independent of each other in the context of a data exchange. This can be between a PC and a server, a reader and a server, a chip card and a chip card reader such as, for example, a bank note dispenser in the context of the use of a chip card referred to as A by a user who, wishing to procure a few banknotes, goes to a place having a banknote dispenser, referred to as B.

More particularly, the two parties A and/or B can be considered to be a chip card and/or a reader.

It is known to experts that the problems of security are crucial for correct functioning of appliances of the banknote dispenser and chip card type. These concerns are the subject of continuous and ever more effective protective provisions for making an attack more and more difficult to commit.

To explain this state of affairs, in the technological field of cryptology, it is known that the two parties who are to communicate with each other are referred to as A and B.

In more technical and scientific terms, the method consists of a certain number of steps.

In the detail, A, who possesses a secret key K, chooses a random value R, or random number, also referred to as a message.

A sends this random value R to B, who also possesses the same secret key K.

B computes a result, by means of a symmetrical enciphering algorithm ALG, referred to as r, by means of the following formula:

$$r = ALG[K](R).$$

r is the result of the enciphering of the message R with the algorithm ALG and the secret key K. The algorithm known as ALG is a symmetrical enciphering algorithm. This may be DES (Data Encryption Message), triple DES, IDEA, etc.

This operation can be carried out since the secret key K is known to the two parties A and B and only by A and B.

Next, B sends the result r to A. The latter, A, then computes the result r' by means of the following formula:

$$r' = ALG[K](R).$$

If the number r is equal to the result r' then B is authenticated.

The authentication of B by A is effected in this way.

However, this operation is not sufficient since it is vulnerable to current measurements and therefore to the attacks of a possible pirating. An attacker, or pirate, measures the electric current consumption of the chip. According to the curve obtained, he can deduce information on the operations and the data used by the processor. In order to effect precise measurements, the attacker must make several measurements and filter them.

More precisely, the problem lies in the computation of r, that is to say on the formula:

$$r = ALG[K](R).$$

This is because current measurements are easy to carry out at this level and it is consequently possible to know the data through the computation, in particular because the key K is constant.

In order to obtain a pertinent measurement, the pirate must carry out several measurements and filter them in order to extract the pertinent information therefrom. Because of the use of the same constant key K, all the measurements use the same key K and

therefore the result of the filtering is characteristic of the key K.

The invention proposes a first particularity which consists of a succinct modification which is carried out in the formula for computation of r' of the authentication protocol, which is the object of the invention.

The party A, who possesses the secret key K, sends a random value or randomly chosen number R_1, to the party B. The latter also possesses the same secret key K.

B chooses a random number R_2, and then computes the number K', which is also a secret key, using the following formula:

$K' = R_2$ xor K, xor being the mathematical "exclusive OR".

Next, B computes a result r resulting from the formula:

$r = ALG[K'](R_1)$

B then sends r and R_2 to A.

The latter computes, by means of the same symmetrical enciphering algorithm ALG, K' and r' using the following formulae:

$K' = R_2$ xor K

and

$r' = ALG[K'](R_1)$

On the assumption that r is equal to r', then B is authenticated by A.

The impossibility of attack at the present time stems from the fact that, since K' changes at each

authentication, the current consumption of the computation of r and r' is different at each execution of the authentication.

However, the computation of K' remains vulnerable to attacks involving current consumption. The invention therefore proposes a second particularity not related to the first described above, concerning the computation of K'.

This is because the invention uses an enciphering system independent of and/or dependent on the authentication system described above.

It consists of a computation of K' which is made random.

For this purpose, the secret key K is decomposed into a set of n subkeys k_i, i being the subkeys index, by means of the formula:

$K = k_1 \text{ xor } k_2 \text{ xor } \ldots \text{ xor } k_n$.

Thus the computation of K' is possible as being, in another formulation:

$K' = R_2 \text{ xor } k_1 \text{ xor } \ldots \text{ xor } k_n$.

Because of the substitutable nature of the operator xor, it is possible to change the computation order in order to obtain a variable computation at each authentication.

So that a link is established between K and K', the algorithm used comprises an initialisation phase and a subset of loops.

The initialisation of the algorithm must be explained, in general terms first of all, and then by

taking a particular case of explanation, then transposable to the generality.

The following remarks explain the initialisation.

In general terms, a first table k, referred to hereinafter in the description as k[], is used; this table contains the values of the n subkeys k_i.

A second table, referred to as to_do, referred to hereinafter as to_do[], contains n Booleans. Each Boolean contains the true value referred to as "True" or "T", below and in Figure 2. The tables k[] and to_do[] contain the same number n of elements, representing the n subkeys k_i and the n Booleans.

The value R_2 is allocated to K', or more precisely $K' = R_2$.

The loop of the algorithm is described below and is depicted in Figure 1:

The first step, or step a, consists of the fact that, as long as there remains an element in the table to_do[] at the value "T", then a random number i, lying between 1 and n, is chosen.

The following step, step b, is the test of equality of the element i in the table to_do[] and the value "T", referred to hereinafter in the description and in the figure.

If the above equality test is true, two operations are performed:

- The first, step c, is the allocation to the variable K' of the result of the mathematical "exclusive OR" between K' and the i^{th} element of the

table k[], which is in fact the computation of the following formula:

K' = K' xor k[i]

- The second, step d, is the allocation to the element of index i of the table to_do[] of the value "False", hereinafter referred to in the description and in the figure as "F": to_do[i] = "F".

If the equality test of step b is false, then the computation system returns to the first step, or step a.

This algorithm is not in constant time since it is possible to execute more loops than there are subkeys k_i.

The invention also relates to an authentication system with symmetrical enciphering algorithm between two entities or parties A and B, possessing the same secret key K, which implements the method described above.

The invention will now be described with a precise example embodiment, which is the case for which n, the number of subkeys, is equal to two, n = 2, in relation to Figure 2, a figure describing only two loops.

The initialisation remains identical to the general case described above; it is mentioned in Figure 2 by the reference A or 10.

The loop of the algorithm is effected as follows:

Two computations of the algorithm are possible.

Either the following operation will be performed:

$K' = R_2 \text{ xor } k_1 \text{ xor } k_2$;

Or the following operation will be performed:

$K' = R_2 \text{ xor } k_2 \text{ xor } k_1$;

Thus the pirate does not know which computation will be performed initially and therefore cannot use several measurements for effecting filterings.

The probability of filling the two elements in a single transit of the loop or first attempt is zero, or, in a more explicit and visual manner (cf Figure 2), the probability of putting the two elements of the table to_do[] to the value "F" in a single attempt is zero. It is therefore impossible to obtain, in a single looping, the two identical values "F", "F".

Next, the probability of putting the two elements of the table to_do[] to the value "F" in two loops or attempts is equal to one half, or 1/2.

This is because, during the first loop, a random number i equal to 1 (20) or 2 (21), i=1 or 2, is chosen; then the value "F" is put in one of the two elements of the table to_do[]; this first loop is mentioned in Figure 2 by the reference B.

During a second loop (cf Figure 2), a random number i equal to 1 or 2, i=1 or 2, is chosen; then the value "F" is put in one of the two elements of the table to_do[], according to the random number chosen; this second loop is mentioned in Figure 2 by the reference C.

Thus there are two cases terminated (31, 32), that is to say two values "F" in two elements and two non-terminated cases (30, 33); the computation of K' is not terminated. The probability of obtaining this result is one half.

In addition, the probability of filling two elements in three loops or attempts is equal to one quarter, or 1/4 (not depicted in Figure 2).

This is because a random number i equal to 1 or 2 is chosen; the value "F" is put either in the first element or in the second element of the table to_do[].

Thus, like during the second loop, there are two cases terminated, that is to say two values "F" in two elements, and two cases not terminated. The computation of K' is not terminated.

In very general terms, the probability of filling the two elements in k loops is equal to $1/2^{k-1}$.

It is advantageous to know an average S of loops to be performed in order to fall on two elements each comprising the value "F".

For this, the mathematical expectation is computed and is formulated as follows:

$$S = \sum_{i=2}^{\infty} i \times \frac{1}{2^{i-1}}$$

This mathematical expectation is the weighted sum of the probabilities.

It is equal, once computed, to three.

The conclusion is therefore: S = 3

The computation of K' is effected in three loops on average.

CLAIMS

1. A method for authenticating with symmetrical enciphering algorithm between two entities or parties A and B, possessing the same secret key K, characterised in that it performs the following steps for the authentication:

a) the party A, possessing the secret key K, sends a random value or randomly chosen number R_2 to the party B, the latter also possessing the same secret key K;

b) the party B chooses a random number R_2 and computes the number K', which is also a secret key, using the following formula:

$K' = R_2$ xor K, xor being the equivalent of the mathematical "exclusive OR";

c) next B computes, by means of a symmetrical enciphering algorithm ALG, a result r resulting from the formula:

$r = ALG[K'](R_1)$;

d) the said party B next sends r and R_2 to the said party A;

e) the said party A computes, by means of the same symmetrical enciphering algorithm ALG, K' and r' using the following formulae:

$K' = R_2$ xor K

and

$r' = ALG[K'](R_1)$;

f) if r is equal to r', then the said party B is authenticated by the said party A.

2. A method according to Claim 1, characterised in that the key K is decomposed into a set of n subkeys k_i, i being the subkeys index, by means of the formula:

$K = k_1 \text{ xor } k_2 \text{ xor } \ldots \text{ xor } k_n$.

3. A method according to Claim 1 or 2, characterised in that it comprises a system of computing the key K' made random by the order of use of the subkeys k_i, i being the subkeys index, in the formulation:

$K' = R_2 \text{ xor } k_1 \text{ xor } \ldots \text{ xor } k_n$.

4. A method according to Claim 3, characterised in that the secret key K' is computed by means of an algorithm which comprises an initialisation phase and a set of loops.

5. A method according to Claim 4, characterised in that the initialisation phase comprises:

- a first table k, or k[], containing the n subkeys k_i, and

- a second table known as to_do, or to_do[], containing n Booleans, each Boolean containing the true value called "True" or "T",

and characterised in that the tables k[] and to_do[] contain the same number n of elements, representing the n subkeys k_i and the n Booleans.

6. A method according to Claim 4, characterised in that the algorithm loop comprises the following steps

a) the first step consists of the fact that, as long as there remains an element in the table to_do[]

* C L M S *

at the value "T", a random number i, lying between 1 and n, is then chosen;

b) the second step consists of the test of the element i in the table to_do[] at the value "T";

c) if the result of the second step is true, then the third step consists of two operations:

- the first is the allocation, to the variable K', of the result of the mathematical "exclusive OR" between K' and the i^{th} element of the table k[], which is the computation of the following formula:

K' = K' xor k[i]

- the second is the allocation to the element of index i of the table to_do[] of the value "False", referred to hereinafter in the description and in the figure as "F": to_do[i] = "F".

7. A method according to Claim 4, characterised in that the algorithm loop comprises the following steps:

a) the first step consists of the fact that, as long as there remains an element in the table to_do[] at the value "T", a random number i, lying between 1 and n, is then chosen;

b) the second step consists of the test of the element i in the table to_do[] at the value "T";

c) if the result of the second step is false, then the computation system returns to the first step.

8. A system for authenticating with a symmetrical enciphering algorithm between two entities or parties A and B, possessing the same secret key K, characterised

in that it implements the method for authenticating according to any one of Claims 1 to 7.

* C L M S *