Self-certified public keys

Marc Girault
Service d'Etudes communes de la Poste et de France Télécom (SEPT)
Groupement PEM
42 rue des Coutures, B.P. 6243
14066 CAEN-CEDEX, France

Abstract *We introduce the notion, and give two examples, of self-certified public keys, i.e. public keys which need not be accompanied with a separate certificate to be authenticated by other users. The trick is that the public key is computed by both the authority and the user, so that the certificate is "embedded" in the public key itself, and therefore does not take the form of a separate value.*

Self-certified public keys contribute to reduce the amount of storage and computations in public key schemes, while secret keys are still chosen by the user himself and remain unknown to the authority. This makes the difference with identity-based schemes, in which there are no more certificates at all, but at the cost that secret keys are computed (and therefore known to) the authority.

1. Introduction

A lot of public-key schemes have been designed since their discovery in 1976 [DH]. In such schemes, every user has a key-pair (s,P). The first one, s, is a secret key, only known to this user. The second one, P, is a public key, which anybody may know. The two keys, secret and public, are mathematically strongly connected, but knowing only the second one is insufficient to retrieve the first one in reasonable time.

By definition, public keys need not be protected for confidentiality ; on the contrary, they have to be made as public as possible. But this "publicity" makes them particularly vulnerable to active attacks, such as the substitution of a "false" public

D.W. Davies (Ed.): Advances in Cryptology - EUROCRYPT '91, LNCS 547, pp. 490-497, 1991.

key to a "true" one in a directory. This is why, in addition to the key-pair (s,P) and his identification string (or identity) I, the attributes of a user must also contain a "guarantee" that P is really the public key of user I, and not the one of an impostor I'.

Depending on the form of this guarantee, we may distinguish several types of schemes. Their common point is to require the existence of an authority, in which every user "trusts" (in a sense to define ...). It seems difficult to get rid of such a requirement, except that there may be several authorities, each user trusting at least one of them.

In certificate-based schemes, the guarantee G takes the form of a digital signature of the pair (I,P), often called certificate, computed and delivered by the authority. In such a case, the four attributes (I, s, P and G) are distinct ; three ones are public (I, P and G) and should be available in a directory. When somebody needs, for example, to authenticate user I, he gets his public triplet (I,P,G), checks G with the help of the authority's public key, that everybody is supposed to know, and afterwards makes use of P to authenticate this user. This is CCITT X509 approach [CCI].

In identity-based schemes, introduced by Shamir [Sh] in 1984 (see e.g. [FS] or [GQ]), the public key is nothing but the identity I (i.e. $P=I$). And the guarantee is nothing but the secret key itself (i.e. $G=s$), so that only two attributes exist (I and s) instead of four. This approach is very attractive, since there is no certificate to store and to check, but has its drawbacks. In particular, the authority can impersonate any user at any moment since secret keys are calculated by it.

In this paper, we propose a new type of scheme, intermediary between certificate-based and identity-based ones. In such schemes, the guarantee is equal to the public key (i.e. $G=P$), which therefore may be said *self-certified*, and each user has three attributes (I, s and P).

Schemes using self-certified public keys are neither certificate-based, since there is no separate certificate, nor identity-based, since the public key is not restricted to the identity. As a consequence, they contribute to reduce the amount of storage and computations (in particular, they do not require hash-functions at authority level) while secret keys are still chosen by the user himself and remain unknown to the authority.

Our paper is organized as follows : after the present introduction (section 1), we give some general features on public-key schemes using self-certified public keys (section 2), then provide two examples of such schemes (sections 3 and 4). The first one is based on factorization and discrete logarithm problems. The second one is only based on discrete logarithm problem.

2. General

The authority mentioned in the introduction is the link between all the users connected to a same network. Thanks to it, two people who have never "met" before and who share nothing (except universal parameters related to the authority) may set up an authenticated or confidential channel. Of course, this works if and only if users trust this authority. But what does "trust" mean here ?

With secret-key schemes, such authorities know all the secret keys held by the users and these users must have every confidence in it. With public-key schemes, this drastic condition can be greatly relaxed, but a careful analysis shows that several levels of trust can be defined. In this paper, we essentially distinguish three levels (1, 2 and 3).

At level 1, the authority knows (or can easily compute) users' secret keys and, therefore, can impersonate any user at any time without being detected. At level 2, the authority does not know (or cannot easily compute) users' secret keys. Nevertheless, the authority can still impersonate a user by generating false guarantees (e.g. false certificates). This is why we also require, to reach level 3, the frauds of the authority to be detectable. More precisely, a public-key scheme will be said of level 3 if the authority cannot compute users' secret keys, and if it can be proven that it generates false guarantees of users if it does so.

Clearly, the level 3 is the most desirable one, and is achieved by certificate-based schemes. Indeed only the authority is able to produce certificates. As a consequence, the existence of two (or more) different certificates for the same user is in itself a proof that the authority has cheated.

Now, as the storage and the verification of certificates lead to additional parameters to store, exchange and more computations to perform, it would be pleasant to design schemes which are not certificate-based, while they still achieve level 3. Identity-based schemes fail to do that since they only achieve level 1, which may be highly insufficient in some applications. The two schemes that we are going to present now use self-certified public keys and reach level 3.

Before describing these schemes, we wish to point out that, as in certificate-based schemes and contrary to identity-based ones, the channel used by a user and the authority needs not be a confidential one.

3. A scheme using RSA/Rabin digital signature scheme

3.1 Set-up

At SECURICOM'89 conference, Paillès and the author have presented a scheme

[PG] using self-certified public keys (but without employing this name). A similar scheme has also been proposed by Tanaka and Okamoto at SECURICOM'90 [TO]. A new version of PG scheme has been presented at ESORICS'90 conference [GP] (or [Gi]). All these proposals use RSA digital signature scheme but only reach level 2. The reason why is the possibility for each user to create other valid public keys linked to his identity I, after he has been given one by the authority. As a consequence, a judge cannot distinguish between a cheating authority and a cheating user (we will now call Alice).

In our scheme, the authority generates a RSA key-pair [RSA], that is a large integer n, product of two prime factors p and q, an integer e coprime to $p-1$ and $q-1$, and the converse d of e modulo $(p-1)(q-1)$. Then it computes an integer g of maximal order in the multiplicative group $(Z/nZ)^*$, with usual notations. The parameters n, e, and g are published by the authority whilst p, q and d are kept secret.

Actually, this RSA key could be replaced by a Rabin's one [Ra] (i.e. with $e = 2$), in order to reduce the number of multiplications to carry out, but only RSA case is described here.

Now, Alice randomly chooses a (say) 150-bit secret key s, computes the integer $v = g^{-s} \pmod n$ and gives v to the authority. Then she proves to the authority that she knows s without revealing it, by using the protocol described below (paragraph 3.2). Afterwards the authority computes Alice's public key P as a RSA signature of the modular difference of g^{-s} and I :

$$P = (g^{-s} - I)^d \bmod n$$

So the following equation, called (E), holds :

$$P^e + I = g^{-s} \pmod n$$

Now, this set-up will enable Alice to prove her identity with the help of a minimum-knowledge identification protocol, or exchange secret keys with other users in an authenticated manner.

3.2 Identification protocol

Alice proves to Bob she is Alice, by convincing him that she knows a discrete logarithm modulo n (the one of $v = (P^e + I)$), while ignoring the factorization of n. This can be done with the following protocol, related to Beth's [Be] or Schnorr's one [Sc], and also to a protocol from Okamoto and Ohta [OO] :

1) Alice sends I and P to Bob, who computes $v = (P^e + I) \bmod n$.

2) Alice selects a (say) 220-bit random integer x, computes $t = g^x \pmod n$, and sends t to Bob.

3) Bob selects a (say) 30-bit random integer c and sends it to Alice.

4) Alice computes $y = x + sc$ and sends it to Bob.

5) Bob checks that $g^y v^c = t \pmod n$.

It can be proven that :

- Alice will be accepted by Bob with probability almost 1 (completeness)
- an impostor, who does not know s, will be detected with probability $1-2^{-30}$ (soundness)
- the protocol hardly reveals anything about s (minimum-knowledgeness).

The same protocol (steps 2 to 5) is used by Alice and the authority at the set-up of the scheme (see paragraph 3.1).

Note that there is no certificate to check : public-keys are "self-certified". Note also that another user cannot infer Alice's secret key from her public key, provided discrete logarithm problem is hard. Moreover, though it holds the factors of n, the authority is also unable to compute s (from $g^{-s} \pmod n$) if these factors are large enough (say 350 bit).

Of course, the authority can still compute "false" public keys linked to Alice, by choosing a number s' and computing P' as described in paragraph 3.1. But, since only the authority is able to produce valid keys (i.e. satisfying equation (E)), the existence of two (or more) different valid public keys for the same user is in itself a proof that the authority has cheated. This shows that the scheme reaches level 3, as promised.

3.3 Key exchange protocol

Let (I,s,P) be the attributes of Alice, (I',s',P') those of Bob. They can simply exchange an authenticated key by choosing :

$$K = (P^e + I)^{s'} = (P'^e + I')^s = g^{-ss'} \pmod n$$

This protocol is clearly related to Diffie-Hellman's one, but, contrary to it, makes Alice sure that she shares K with Bob and conversely.

4. A scheme using El-Gamal digital signature scheme

Contrary to the preceding one, this scheme is only based on the difficulty to compute discrete logarithms modulo a prime number. It results from a combination of Beth's scheme and Horster-Knobloch "testimonial scheme" [HK].

Beth's scheme is an identity-based identification scheme, in that Alice's secret key is computed by the authority as an El-Gamal [El] signature of her identity I. Actually, only one part of this signature is the secret key, and the other part has to be transmitted to those who wish to authenticate Alice. In our paradigm, this second part can be viewed as a self-certified public key. But only level 1 is achieved.

In order to reach level 3, we can combine Beth's scheme with the so-called testimonial digital signature scheme. This gives the following :

The authority generates a large prime p such that $p-1$ has also a large prime factor (e.g. $(p-1)/2$) and a primitive element g of $(Z/pZ)^*$. Then it selects an integer a in $[0,p-2]$ and computes $b = g^a \pmod p$. The parameters p, g, and b are published by the authority whilst a is kept secret.

Now, Alice chooses a random integer h, computes $u = g^h \pmod p$ and gives u to the authority. The authority chooses a random integer k, computes $P = u^k \pmod p$ and solves in x the equation :

$$aP + k\alpha = I \mod (p-1)$$

Then the authority returns (P,I,x) to Alice, who calculates : $s = xh^{-1} \pmod p$ so that :

$$b^P P^s = g^I \pmod p$$

Alice's secret key is s and her self-certified public key is P. Note that the pair (s,P) is an El-Gamal authority's signature of I, but that the authority ignores s ! (A sort of paradox ...). That makes the difference with original Beth's scheme and explains why level 3 is reached by this one.

Now, Alice can be authenticated by Bob, using Beth's protocol, as described in [Be].

Acknowledgements

I would like to thank E. Okamoto for his careful analysis of section 3, and H.J. Knobloch for having informed me of the testimonial scheme.

References

[Be] T. Beth, "A Fiat-Shamir-like authentication protocol for the ElGamal scheme", Advances in Cryptology, Proc. of EUROCRYPT'88, LNCS 330, Springer-Verlag, 1988, pp.77-86.

[CCI] "The Directory-Authentication Framework", CCITT Recommendation X509.

[DH] W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, Vol.IT-22, Nov.1976, pp.644-654.

[El] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", Advances in Cryptology, Proc. of CRYPTO'84, LNCS 196, Springer-Verlag, 1985, pp.10-18.

[FS] A. Fiat and A. Shamir, "How to prove yourself : Practical solutions to identification and signature problems", Advances in Cryptology, Proc. of CRYPTO'86, LNCS 263, Springer-Verlag, 1987, pp.186-194.

[Gi] M. Girault, "An identity-based identification scheme based on discrete logarithms modulo a composite number", Proc. of EUROCRYPT'90, LNCS 473, Springer-Verlag, 1991, pp.481-486.

[GP] M. Girault and JC. Paillès, "An identity-based identification scheme providing zero-knowledge authentication and authenticated key exchange", Proc. of ESORICS'90, pp.173-184.

[GQ] L.C. Guillou and J.J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory", Advances in Cryptology, Proc. of EUROCRYPT'88, LNCS 330, Springer-Verlag, 1988, pp.123-128.

[HK] P. Horster and H.J. Knobloch, "Discrete logarithm based protocols", these Proceedings.

[OO] T. Okamoto and K. Ohta, "How to utilize the randomness of zero-knowledge proofs", Proc. of CRYPTO'90, to appear.

[PG] J.C. Paillès and M. Girault, "CRIPT : A public-key based solution for secure data communications", Proc. of SECURICOM'89, pp.171-185.

[Ra] M.O Rabin, "Digitalized signatures and public-key functions as intractable as factorization", MIT, Lab. for Computer Science, MIT/LCS/TR-212, Jan.1979.

[RSA] R.L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", CACM, Vol.21, n°2, Feb.1978, pp.120-126.

[Sc] C.P. Schnorr, "Efficient identification and signatures for smart cards", Advances in Cryptology, Proc. of CRYPTO'89, LNCS 435, Springer-Verlag, pp.239-252.

[Sh] A. Shamir, "Identity-based cryptosystems and signature schemes", Advances in Cryptology, Proc. of CRYPTO'84, LNCS 196, Springer-Verlag, 1985, pp.47-53.

[TO] K. Tanaka and E. Okamoto, "Key distribution system using ID-related information directory suitable for mail systems", Proc. of SECURICOM'90, pp.115-122.