A Note on Girault's Self-Certified Model

Shahrokh Saeednia

Université Libre de Bruxelles, Département d'Informatique
CP 212, Boulevard du Triomphe, 1050 Bruxelles, Belgium
saeednia@ulb.ac.be

Abstract. In this paper, we describe an important shortcoming of the first self-certified model proposed by Girault, that may be exploited by the authority to compute users' secret keys. We also show, while it is possible to make the attack ineffective by taking additional precautions, the resulting model loses all merits of the original model and does no longer meet the primary contribution of the self-certified notion.

Key words : Self-certified, Levels of trust, Random safe primes, Zero-knowledge, Discrete logarithm.

1 Introduction

The common problem in all public key schemes is to guarantee that a public key P is really linked to a user I and does not belong to a cheater I' whose goal is to impersonate I. The solution to this problem is obviously for an authority to provide authenticated public keys.

In the simplest approach, which is called *certificate-based*, the authority creates a *certificate* for each user, after having checked carefully his identity. In such a case, each user computes his public key P, gives it to the authority and receives a certificate of the form $C = S(P, I)$, which is actually the authority's signature on the pair of the user's public key and identity string (prepared by the authority). Whenever somebody wants to use I's public key, he first checks the validity of the certificate C using the authority's verification key (that everybody is assumed to know) and uses the public key afterwards.

This technique, though having the advantage that even the authority does not know the users' secret keys requires some amount of storage and computation, which essentially depends on the signature scheme in use.

Another approach to create authenticated keys is known as *identity-based* keys, introduced by Shamir [13]. The advantage of this method is that the users' public keys are simply their identity string and that the related secret key is computed from some trapdoor originated by the authority, so that nobody can create a valid pair of public and secret keys or determine the secret key corresponding to a given public key without knowing that trapdoor. This leads to a scheme that needs no certificate and no verification of signatures, hence reduces the amount of storage, communication and computation. The disadvantage of this method is however that the secret keys are known to the authority.

A more sophisticated approach combining the advantages of certificate-based and identity-based models is proposed by Girault [5], which is known as *self-certified*. In this type of authenticated keys, (contrary to identity-based schemes) each user chooses his secret key and computes his public key. Then, (contrary to certificate-based schemes) instead of signing the pair of public key and identity string, the authority creates a certificate from that pair, in such a way that it cannot be computed without the knowledge of some trapdoor, known only to the authority.

As we can see, the level of "trust" required in the approaches presented above is different. Girault [5] defined three levels of trust as follows:

1. The authority knows (or can easily compute) the users' secret keys and is capable of impersonating any user without being detected.
2. The authority does not know the secret keys, but it can still impersonate any user by generating false certificates that may be used without being detected.
3. The authority does not know (and cannot compute) the secret keys and if it generates false certificates for users, it can be proven.

With this definition, in schemes of level 1 and 2, the authority must be fully trusted by the users, while in schemes of level 3, it is actually considered as a potentially powerful adversary that may use any means to impersonate the users.

Hence, identity-based schemes only achieve level 1, while certificate-based and self-certified schemes attain level 3. Note that, although in the latter schemes the authority can still create false public keys and certificates linked to a given user, the existance of more than one public key for the same user means that the authority has cheated, since this may only be accomplished by the authority.

Although both certificate-based and self-certified models are of level 3, the latter presents an advantage that is about the amount of storage, communication and computation. In schemes using self-certified public keys, a single value replaces the pair of the user's public key and the related authority's signature used in certificate-based schemes. This value is then communicated at the beginning of the protocol and is subsequently used for reconstructing the user's public key. This constitutes at the same time a disadvantage of the self-certified model. In fact, in self-certified schemes, it is not necessary and not possible (except for the owner of the secret key) to verify the correctness of a certificate. So, if for example, the verification of a signature (using self-certified keys) fails, it cannot be known whether the public key or the signature is invalid.

In this paper, we show that the first self-certified model proposed by Girault in [5] is only of level 1. We also show how to make that model reach level 3 by taking some additional precautions. The resulting model, however, fails to meet the original contribution of self-certified keys (that is to reduce the amount of storage, computation and communication in public key schemes) and so, presents no more advantage over a related certificate-based model. This is because in the repaired model ,it would be mandatory to increase the size of the system parameters. This clearly leads to larger certificates and more complex computations during the execution of any protocol that uses this model.

2 Girault's self-certified model

In Girault's model, the authority is assumed to choose an RSA modulus n, as the product of two random safe primes p and q (i.e., such that $p-1=2p'$ and $q-1=2q'$, where p' and q' are also primes), an integer e co-prime to $p-1$ and $q-1$ and an integer g of maximum order in $(Z/nZ)^*$. Then it computes the inverse d of e modulo $\phi(n)$ and publishes n, e and g, while p, q and d will be kept secret.

Now, each user chooses his secret key s and computes $v = g^{-s} \pmod{n}$ and gives v to the authority. The latter checks whether there exists already a certificate for that user and if not so, computes a certificate P for the user as

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

Such a certificate is then used in various protocols, such as identification schemes, public key cryptosystems or key exchange protocols, as shown in [5]. Knowing the certificate of a user together with his identity, everybody can easily compute his public key as $v = (P^e + I) \pmod{n}$ and use it in the protocol. The genuineness of this public key is only garanteed if the protocol is successfully completed.

It is clear that in order to create a certificate without knowing d, one has to compute the eth root of $g^{-s} - I \pmod{n}$ that is actually equivalent to breaking RSA.

3 Problems and solutions

The above key generation process is claimed to be of level 3, because the authority does not know the users' secret keys and is assumed not to be able to compute them. In fact, to retrieve a secret key, the authority has to compute a discrete logarithm modulo a composite integer n (such as an RSA modulus) the factorization of which it knows. Of course, this is an infeasible task if n is large enough and fairly generated, but it may become feasible (though never easy) if the authority generates n in a special dishonest way.

More precisely, a cheating authority can choose n either as the product of some relatively small primes (1), or such that all prime factors of $p-1$ and $q-1$ are small (2). This is possible, since there is no verification from the users about the status of n. The danger in such cases is that computing discrete logarithms becomes feasible for the authority.

As Bach showed [1], to compute discrete logarithms modulo a composite integer n, it suffices to first factorize n and to solve discrete logarithms modulo each prime factor, afterwards.

Hence, the authority that already knows the prime factors of n, has just to compute discrete logarithms modulo each prime factor, to derive the users' secret keys.

Thanks to the Pohlig-Hellman algorithm [11], the authority can do so if n is chosen following (1) or (2). This is because the running time of the Pohig-Hellman algorithm is proportional to the square root of the largest prime factor of $p-1$. So, for small primes p or when $p-1$ has only small prime factors, computing discrete logarithms is feasible in practice.

There could, however, exist a problem with this scenario. If n is chosen following (1), then the factorization of n becomes feasible as well. On the other hand, if n is chosen following (2), then it is also possible to factorize n with Pollard's $p-1$ algorithm [12]. This implies that cheating users can compute d and consequently, other certificates linked to themselves, which enable them to claim later that the authority has cheated.

As a matter of fact, generating a composite number such that it is infeasible for the outside world to factorize it, but it is feasible for its generator to compute discrete logarithms modulo this number is exactly the subject of a paper by Maurer and Yacobi [9], in which they describe a scheme relevant to the identity-based model, and show how the authority can compute the secret key of each user from his identity (public key).

As discussed in [9], since the running time of Pollard's algorithm is proportional to the largest prime factor of $p-1$ rather than its square root (that is the case of Pohlig-Hellman algorithm), it is possible for the authority to choose p and q such that factoring n may be infeasible, while computing discrete logarithms modulo each prime factor is still feasible.

Thus, in the Girault's model, the authority can choose a particular n in order to compute users' secret keys and impersonate them in the future, without being detected. This precisely means that the model is only of level 1.

Therefore, it seems that the only way to prevent such attacks from the authority and make the scheme reach level 3 is that to ensure the users that "a composite number is not a Maurer-Yacobi number", or in other words, n is the product of two large *random safe* primes[1].

Fortunately, there is an interactive protocol due to Camenisch and Michels [3], allowing to prove that a number is the product of two large safe primes, without revealing them. Thus, each user, before giving v to the authority, must check the "safeness" of n by running this protocol with the authority.

Unfortunately, this is not solely sufficient because the above protocol does not allow verifying the randomness of the primes. Without a way of assuring this fact, the model cannot yet be of level 3, since there exist some special (safe) primes for which discrete logarithms are substantially easier to compute. Consequently, a cheating authority may still choose p and q (with $p = 2p' + 1$

[1] Stated that way, the relevance of this paper may be larger than analyzing only Girault's self-certified model, since there may be other models or other schemes which require a proof of this kind. However, we focus here on this model since it has been used in many applications (see, for example [7, 8, 10]).

and $q = 2q' + 1$), in such a way that other methods such as the special number field sieve can be applied to compute the logarithms.

In [6], Gordon showed that it is possible to select a prime p of 512 bits for which the discrete logarithms can be found, in a large but feasible amount of time. Let us notice that Gordon's method is originally introduced for analyzing the security of the DSA [4], in which (contrary to our case) $(p-1)/2$ is not a prime and the base g is of order l, a prime divisor of $p-1$. However, the method is essentially independent of the size and the number of prime factors of $p-1$. In fact, in order to find $\log_g y$ with special number field sieve, where g is of order l, one should proceed as follows:

1. choose a base α of maximum order $(p-1)$
2. compute $\log_\alpha y$
3. compute $\log_\alpha g$
4. derive $\log_g y$ from the results of steps 2 and 3.

Here, we will not recall Gordon's method, but just note that it has a conjectured expected running time of $\exp\{(1.00475 + o(1))(logp)^{2/5}(loglogp)^{3/5}\}$. The algorithm actually works for primes p for which there exists an irreducible monic polynomial of degree k with small coefficients, such that for some integers X and Y near $p^{1/k}$, $Y^k f(X/Y) \equiv 0 \pmod{p}$. The ideal polynomial would be of degree four.

Thus, the authority can still choose p and q of this form, which whould allow it to find the users' secret keys in a feasible amount of time.

As far as we know, there is no way of knowing whether a given n is the product of two secret primes of this kind. In [2], Blackburn and Galbraith proposed an interactive protocol for RSA key generation allowing a certification authority to verify that a generated key by a user is randomized. More precisely, the authority intervenes in the key generation phase to force each user choosing p and q at random, without gaining any information about them. Although there are some related ideas, that protocol can obviously not be applied to Girault's model, since here we have many verifiers (the users) who want to check independently that a single modulus is actually generated at random by the authority.

So, it seems that the only way of preventing this authority's attack is to choose n of larger size. Even for n of 1024 bits, this weakness remains exploitable by the authority.

For these reasons, we recommand that p and q be of 1024 bits each and that the Camenisch and Michels protocol be used to ensure the users involved in the system that the authority cannot take any advantage of choosing the system parameters for computing their secret keys.

With these precautions, Girault's model will achieve level 3, but loses its only advantage over a certificate-based model that uses a 1024-bit RSA key for signing public keys. In fact, public keys in this mode may be computed modulo a prime p of 1024 bits length and be signed afterward, providing a total of 2048-bit data. This is actually the same amount of information to be stored and communicated by its owner than in the modified self-certified model. In addition, considering the modulos used in each model, the computational cost of a protocol using the modified self-certified model would be twice of what is required for the execution of that protocol when it is adapted for using with the certificate-based model.

4 Conclusion

We showed that Girault's self-certified model, in its current form, is only of level of trust 1. To make the model of level 3, we proposed to choose the modulus n as a 2048-bit integer and add an interactive zero-knowledge proof in the key generation phase between each user and the authority; this will convince the users that n is "safe". However, the resulting model does no longer meet the primary contribution of the self-certified notion, as it loses all its merits and is even less efficient than a related certificate-based model.

References

1. E. Bach, "Discrete logarithms and factoring", Technical Report UCB/CSD 84/186, Computer Science Division (EECS), University of California, Berkeley, California, 1994
2. S.R. Blackburn and S.D. Galbraith, "Certification of secure RSA keys", *Electronics Letters*, vol. 36, 2000, pp. 29-30
3. J. Camenisch and M. Michels, "Proving in zero-knowledge that a number is the product of two safe primes", *Advances in Cryptology* (Proceedings of *EuroCrypt* '99), Lecture Notes in Computer Science, vol. 1592, Springer-Verlag, 1999, pp. 107-122
4. FIPS PUB 186, February 1991, *Digital Signature Standard*
5. M. Girault, "Self-certified public keys", *Advances in Cryptology* (Proceedings of *EuroCrypt* '91), Lecture Notes in Computer Science, vol. 547, Springer-Verlag, 1991, pp. 490-497
6. D. M. Gordon, "Designing and detecting trapdoors for discrete log cryptosystems", *Advances in Cryptology* (Proceedings of *Crypto* '92), Lecture Notes in Computer Science, vol. 740, Springer-Verlag, 1993, pp. 66-75
7. O. Baudron, F. Boudot, P. Bourel, E. Bresson, J. Corbel, L. Frisch, H. Gilbert, M. Girault, L. Goubin, J. F. Misarsky, P. Nguyen, J. Patarin, D. Pointcheval, G. Poupard, J. Stern and J. Traoré, "GPS, an asymmetric identification scheme for *on the fly* authentication of low cost smart cards", A proposal to *NESSIE*, https://www.cosic.esat.kuleuven.ac.be/nessie/
8. S. Kim, S. Oh, K. Kim and D. Won, "One-time self-certified public keys, revisited", *International Conference on Information Security and Cryptology*, (proceedings of *ICISC* '98, 1998, pp.59-69
9. U. Maurer and Y. Yacobi, "Non-interactive public key cryptography", *Advances in Cryptology* (Proceedings of *EuroCrypt* '91), Lecture Notes in Computer Science, vol. 547, Springer-Verlag, 1991, pp. 498-507
10. H. Petersen and P. Horster, "Self certified keys - Concepts and Applications", Proceedings of *Communications and Multimedia Security* '97, Chapman & Hall, 1997, pp. 102 - 116
11. S. Pohlig and M. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", *IEEE Trans. On Inf. Th.*, vol. 24, 1978, pp. 106-110

12. J. M. Pollard, “Theorems on factorization and primality testing”, *Proceedings of the Cambridge Philosophical Society*, vol. 76, 1974, pp. 521-528
13. A. Shamir, “Identity-based cryptosystems and signature schemes”, *Advances in Cryptology* (Proceedings of *Crypto* ’84), Lecture Notes in Computer Science, vol. 196, Springer-Verlag, 1985, pp. 47-53