Can O.S.S. be Repaired ?
- Proposal for a New Practical Signature Scheme -

David Naccache

Gemplus Card International
1 Place de Navarre, F-95200, Sarcelles, FRANCE.
E-mail : 100142.3240@compuserve.com

Abstract. This paper describes a family of new **Ong-Schnorr-Shamir-Fiat-Shamir-like** [1] identification and signature protocols designed to prevent forgers from using the **Pollard-Schnorr** attack [2].

Our first signature scheme (and its associated identification protocol) uses x, which is secret-free, as a commitment on which k will depend later. Therefore, the original quadratic equation is replaced by $x^2 - k(x)y^2 \equiv m \bmod n$ where k(x) is a non-polynomial function of x and since the **Pollard-Schnorr algorithm** takes as input value k (to output x and y), it becomes impossible to feed *à-priori* k(x) which is output-dependent.

The second signature method takes advantage of the fact that although an attacker can generate valid OSS signatures (solutions {x,y} of $x^2 - k y^2 \equiv m \bmod n$), he has no control over the internal structure of x and y and in particular, if we restrict the solution space by adding extra conditions on x and y, it becomes very difficult to produce forged solutions that satisfy the new requirements.

1. Introduction

In 1985, **Ong, Schnorr and Shamir** proposed a digital signature scheme which seemed to be very efficient [1]. In their system, the public-key consisted of a couple of integers k and n where n is an RSA [7] modulus of length N (bits) whose factorisation is kept secret.

A signature {x,y} of a message m (hashed value of a primitive file *M*) was considered valid if :

$$x^2 - k y^2 \equiv m \bmod n \qquad (1)$$

and it was shown by the authors that general solutions of this equation can be generated by the signer provided that he knows a secret u such that $u^2 k \equiv 1 \bmod n$.

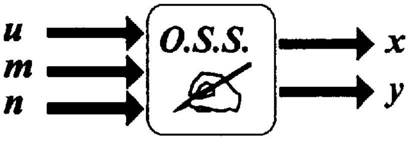

Figure 1 : The original OSS scheme

In 1987, **Pollard and Schnorr** [2], exhibited a fast probabilistic algorithm for computing solutions of equation (1) and thereby broke OSS. This attack was later extended by **Adleman, Estes and McCurley** [3].

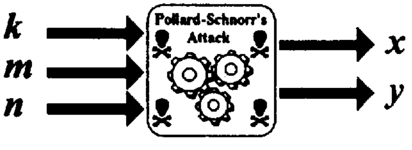

Figure 2 : Sketch of the attack introduced by Pollard and Schnorr

T. Helleseth (Ed.): Advances in Cryptology - EUROCRYPT '93, LNCS 765, pp. 233-239, 1994.

In this article, we describe a couple of signature methods (and an associated identification protocol which is zero-knowledge if the challenge size is bounded) intended to prevent forgery by this family of attacks.

The first (standard) signature protocol uses x (which is secret-free) as a commitment on which k will depend later. Therefore, the original quadratic equation is replaced by $x^2 - k(x)\, y^2 \equiv m \bmod n$ where k(x) is a non-polynomial function of x and since the **Pollard-Schnorr** attack takes as an input a value k (to output x and y), it is impossible to input in advance k(x) which is output-dependent and not yet known.

The second (interactive) signature method takes advantage of the fact that the attacker has no control over the solutions of the congruence $x^2 - k\, y^2 \equiv m \bmod n$. In particular, it is hard to produce an x such that the sub-equation $x = r + \frac{m}{r} \bmod n$ admits a solution r with a given internal redundancy.

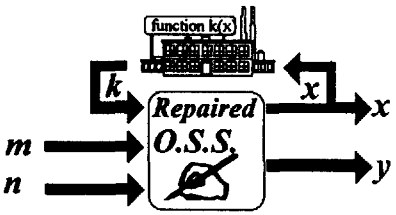

Figure 3 : Our repair strategy

2. Definitions

The system authorities select and publish a one-way function f hashing N-z bit strings into z bit words. Practically, we recommend $z \approx 160$ for $N = 512$ (for instance, SHA or a DES-based hashing).

As in the case of the **Fiat-Shamir** protocol [5], each user is provided with a set of c secret keys $u_1,\ldots,u_c$ and the corresponding public keys $k_1,\ldots,k_c$ (such that $u_i^2 k_i \equiv 1 \bmod n$ for all i) are communicated to the verifiers by any desired means (for instance, ID-based as suggested by **Shamir** in [4] but the key transfer protocol suggested in the original **Fiat-Shamir** [5], should be modified as described in [8]).

3. Protocol #1 (Standard Digital Signature)

① The signer picks t random numbers $\{r_i\}$, computes the set $\{x_i\}$ where : $x_i = r_i + \frac{m}{r_i} \bmod n$, hashes $f(\{x_i\})=\{e_i\}$ (where each e_i is of size c) and calculates $y_i = \prod_{e_{i,j}=1} u_j \left(r_i - \frac{m}{r_i} \right) \bmod n$ for i=1 to t.

② The signature $\{\{x_i\},\{y_i\}\}$ is checked by re-computing $f(\{x_i\}) = \{e_i\}$ and verifying that $x_i^2 - y_i^2 \prod_{e_{i,j}=1} k_j \equiv 4m \bmod n$ for i = 1,...,t.

4. Security and Efficiency

The security level of this protocol is $\frac{1}{2^{tc}}$ (with typically tc ≈ 80) but the size of the signature ("multiple evidence" that the signer affixed honestly his signature on m) is 2Nt.

This yields, (for instance : c=10, t=8), a system where 8192 bit signatures are generated in average $\frac{t(2+c)}{2}$ (=48) modular multiplications with an Nc (=5120) bit secret-key.

The key size and number of multiplications required to implement the new scheme are equivalent to those of the **Fiat-Shamir** but the computation of modular inverses (by using the extended Euclidean algorithm) is much slower than the squaring operation and the size of the resulting signatures is a bit less than the double of **Fiat-Shamir** ones (Things are much better for the identification protocol of section 5 which requires exactly the same amount of transmission as a **Fiat-Shamir**). However, the following variant offers exactly the same amount of transmission as a **Fiat-Shamir** but requires t additional modular squarings : $\{r_i\}$ and $\{x_i\}$ are as in section 5 but $\{e_i\}$ is the hashed value of $\{x_i^2 \bmod n\}$, the definition of the y_i remains unchanged (use the new $\{e_i\}$!) and the signature $\{\{e_i\},\{y_i\}\}$ is checked by comparing $\{e_i\}$ to $f(\{4m + y_i^2 \prod_{e_{i,j}=1} k_j \bmod n\})$.

Note that :

☞ The new scheme is in public domain (no patents) and can be freely used and implemented.

☞ We assume that the verifier checks out trivial weak instances (eg. x= ±2 and y= 0 in protocol 2 etc.).

☞ A simple and practical technique for delegating the computation of $\frac{m}{r}$ mod n to the verifier (which in many practical cases is much more powerful than the prover) is described in section 9.

5. Protocol #2 (Identification and Security Proof)

Repeat t times :

① The prover picks a random r, computes $x = r + \frac{1}{r} \bmod n$ and sends this value to the verifier.

② The verifier replies with a random binary string e, of length c.

③ The prover computes and sends $y = \prod_{e_i=1} u_i \left(r - \frac{1}{r}\right) \bmod n$.

④ And the verifier checks if : $x^2 - y^2 \prod_{e_i=1} k_i \equiv 4 \bmod n$.

The security of the identification scheme can be proved by transforming any algorithm breaking the protocol into a scheme for extracting roots modulo n (For simplicity, we consider the case c=1 since extension to bigger challenges is straightforward) :

Breaking the algorithm means being able to commit in advance a number x (no matter what its internal structure is) such that whatever c will be, both y_1 and y_2 (such that $x^2 - y_2^2 \equiv 4 \bmod n$ and $x^2 - y_1^2 k \equiv 4 \bmod n$) are efficiently computable.

Subtracting the two equations, we get $y_1^2 k \equiv y_2^2 \bmod n$ which yields $\frac{y_2}{y_1} \equiv \sqrt{k} \bmod n$.

6. Protocol #3 (Interactive Signature)

① The signer picks $t > 1$ random N-z bit numbers $h_1,\ldots,h_t$, computes $r_i = 2^{N-z} f(h_i) + h_i$ and $x_i = r_i + \frac{m}{r_i} \bmod n$ for $i=1,\ldots,t$ and sends $x_1,\ldots,x_t$ to the receiver.

② The receiver replies with a randomly chosen index $1 \le j \le t$.

③ The signer reveals all the h_i except h_j.

④ The receiver checks that all the x_i (except x_j) are coherent with the above definition and if a false x_i is detected at this point, the signer is rejected and the protocol is aborted.

⑤ The receiver picks a random binary string e, of length c, and sends it to the signer.

⑥ The signer replies with $y_j = \left(r_j - \frac{m}{r_j}\right) \prod_{e_i = 1} u_i \bmod n$.

⑦ The receiver checks that $x_j^2 - y_j^2 \prod_{e_i = 1} k_i \equiv 4m \bmod n$ and if this test holds, he accepts the interactive signature $\{x_j, y_j, e\}$ of the message m.

7. Security

If the sender uses the **Pollard-Schnorr** attack his chances to remain undetected by the receiver are $\frac{1}{t 2^c}$.

Therefore, the receiver can convince himself, with any desired probability, that the sender actually used a redundant random to generate x_j and y_j.

In case of dispute (the signer pretends that $\{x_j, y_j, e\}$ is a forgery), a judge (knowing the prime factors of n) can solve $X^2 - X x_j + m \equiv 0 \bmod n$ and check that one of the solutions X presents the redundancy of step ①

(That is, $\exists$ h such that $X^2 - X x_j + m \equiv 0 \bmod n$ where $X = 2^{N-z} f(h) + h$).

If yes, the signer is cheating and if not, the receiver used the **Pollard-Schnorr** method and is falsely accusing the signer.

Note that due to the interactive nature of the protocol an attacker is prevented from using pre-processing. The practical significance of this observation is a significant reduction in the size of the key (parameter c).

Also, it should be observed that although the receiver of the signature is convinced that the signature is valid, he cannot transmit this conviction to anybody else (except the judge).

8. Other Possible Research Directions

Except the extension of our scheme to higher degrees (for instance $x^3+ky^3+k^2z^3-3kxyz \equiv m \mod n$ with keys $\{u,w\}$ such that $u^3=k \mod n$, $w^3=1 \mod n$ and $1+w+w^2=0 \mod n$ as suggested by **Ong, Schnorr and Shamir** in [6]), other mono-key OSS-like variants are now being investigated.

These are based on a solution (for x and r) of the equation :

$$r+\frac{g(m,x)}{r}=x \mod n \qquad (2)$$

Where g is a public function.

Such a scheme should work as follows :

① The signer solves the equation $r+\frac{g(m,x)}{r}=x \mod n$ for $\{x, r\}$.

② Then, he computes $y=u\left(r-\frac{g(m,x)}{r}\right) \mod n$ and sends the signature $\{x, y\}$ to the receiver.

③ The signature is checked by comparing that $x^2-ky^2\equiv 4\,g(m,x) \mod n$.

Although still incomplete, we demonstrate this idea with the concrete example $g(m,x)=m\oplus x$ (where "⊕" stands for a bitwise xor) and argument why we believe that efficient algorithms for solving $r+\frac{m\oplus x}{r}=x \mod n$ may exist.

Denoting : $x=\sum_{i=0}^{N-1}2^i x_i$ and $m=\sum_{i=0}^{N-1}2^i m_i$, a simple trick for getting rid of the "⊕" in the sub expression $x_i\oplus m_i$ is the observation that :

$$x_i\oplus m_i = x_i(1-2m_i)+m_i.$$

Replacing this into (2), we get : $r+\sum_{i=0}^{N-1}2^i\left(\frac{x_i(1-2m_i)+m_i}{r}-x_i\right)=0 \mod n$

or (with $R=\frac{1}{r} \mod n$) : $r+\sum_{i=0}^{N-1}2^i\left(x_i(R-2Rm_i-1)+Rm_i\right)=0 \mod n$

and finally : $$\sum_{i=0}^{N-1}x_i2^i(2Rm_i-R+1)=r+Rm \mod n \qquad (3)$$

Defining $a_i=2^i\,(2Rm_i-R+1)$ and $b=r+R\,m$, equation (3) clearly appears as a knapsack problem :

$$\text{Find } \bar{x}=\left\langle x_0,x_1,\ldots,x_{N-1}\right\rangle\in\{0,1\}^N \text{ such that } \sum_{i=0}^{N-1}x_ia_i=b \mod n$$

for which efficient algorithms may exist (under certain assumptions...) for small N (condition that can be softened by leaving some liberty to m).

The idea is that the forger should be unable to "mix" the number-theoretic operations of the **Pollard-Schnorr** algorithm with the knapsack solution but this is not a sufficient argument for proving security and a couple of open questions still persists :

1. Can the linearity in the proposed example be used by a forger in order to attack the system efficiently ?

2. Give a characterisation of the functions g such that repairing OSS by solving $r + \frac{g(m,x)}{r} = x \mod n$ *is secure and still computable efficiently.*

9. Delegating the Extended Euclidean Algorithm

In many cases, the verifier (for instance, a smart-card reader, a terminal or a PC) is much more powerful than the prover (typically a smart-card) and therefore it seems attractive to delegate the computation of the term $\ell = \frac{m}{r} \mod n$ to the verifier :

① The signer picks a random d, computes and sends $s = r\,d \mod n$.

② The powerful verifier computes $v = \frac{m}{s} \mod n$ and sends v to the signer.

③ The signer retrieves $\ell = v\,d \mod n$ and checks that $\ell\, r \equiv m \mod n$.

Practically, this protocol presents the second advantage of not forcing the signer to keep in memory the whole message m : The signer can *secretly* and randomly select a group of 10 bytes in m and check in step ③ that these 10 bytes actually match with those of $\ell\, r \mod n$.

Note that :

☞ The size of d can be reduced to accelerate the multiplications.

☞ For $t>1$, the computation of the inverse of $\prod_{i=1}^{t} r_i$ allows to retrieve all the m/r_i by inter-multiplications.

If r is a (sufficiently big) prime, a second (theoretically) interesting delegation protocol is :

① The signer picks a big random prime d, computes and sends $s = r\,d$ (not modulo !).

② The verifier computes $v = \frac{m}{s} \mod n$ and sends v to the signer.

③ The signer retrieves $\ell = v\,d \mod n$ and checks that $\ell\, r \equiv m \mod n$.

10. Implementation

GemenOSS is one of the public key implementations realised by GEMPLUS CARD INTERNATIONAL.
The whole family includes ***Guillou-Quisquater*** signature and identification protocols implemented within the MIMOSA smart-card, a one-time identification scheme with low memory consumption based on the authentication tree concept, a superfast version of the DSS [9], trading the modular inverse computation against one modular multiplication and the transmission of a few bytes, and an RSA prototype due to be issued in the near future.

GemenOSS implements our identification scheme with tc = 20 and $|n|$ = 512 bits. The verifier (a Compaq Deskpro 4/66i) and the prover (68HC05 clocked at 4 MHz) communicate via a 115,200 baud interface and calculations are done in parallel whenever possible.

Big numbers are manipulated in a redundant (proprietary) format wherein the number of 8-bit by 8-bit multiplications in each big modular multiplication grows as $\log\left(\frac{(\alpha+n)^2}{n-3.02\sqrt{n-\alpha}}\right)$ but transmission is polynomial in parameter α.

The prototype is expected to complete an identification session in about 1.7 seconds when operating at the best α communication / multiplication trade-off point.

11. Conclusion

We demonstrated a family of protocols that allow to reuse quadratic equations modulo n for digital signatures.

The cost of "repairing" the OSS is very acceptable and can be expressed differently (various trade-offs are possible) in terms of key size, number of modular multiplications and transmission overhead.

Due to progress made since the publication of the original OSS scheme, the author strongly encourages the cryptographic community to attack the proposed protocols (#1, #2 and #3) and try to degrade the basic security probabilities (respectively : 2^{-tc}, 2^{-tc} and $2^{-c}/t$) obtained by a brutal application of [2] and [3].

12. Acknowledgements

We thank **Claus Schnorr, Adi Shamir, David M'raïhi** and **Beni Arazi** for their useful remarks, **Jacques Stern** for outlining a subtle attack against a preliminary version of protocol 2 and **Serge Vaudenay** for warning against the choice of weak {x, y} values by a cheating prover (or signer).

References

[1] **H. ONG, C. SCHNORR & A. SHAMIR**, ***"An efficient signature scheme based on quadratic equations"*** in Proceedings of the 16th Symposium on the Theory of Computing, Washington, 1984, pp. 208-216.

[2] **J. POLLARD & C. SCHNORR**, ***"An efficient solution of the congruence $x^2 + k\,y^2 \equiv m$ mod n"***, IEEE Transactions on Information Theory, vol. IT-33, no. 5., September 1987, pp 702-709.

[3] **L. ADLEMAN, D. ESTES & K. McCURLEY**, ***"Solving bivariate quadratic congruences in random polynomial time"***, Mathematics of Computation, vol. 48, no. 177, January 1987, pp 17-28.

[4] **A. SHAMIR**, ***"Identity-Based Cryptosystems and Signature Schemes"***, Proceedings of Crypto'84, Lecture Notes in Computer Science, no. 196, Springer-Verlag 1985.

[5] **A. FIAT & A. SHAMIR**, ***"How to Prove Yourself : Practical Solutions to Identification and Signature Problems"***, Proceedings of Crypto'86, Lecture Notes in Computer Science, no. 263, Springer-Verlag 1986.

[6] **H. ONG, C. SCHNORR & A. SHAMIR**, ***"Efficient Signature Schemes Based on Polynomial Equations"***, Proceedings of Crypto'84, Lecture Notes in Computer Science, no. 196, Springer-Verlag 1985.

[7] **R. RIVEST, A. SHAMIR & L. ADLEMAN**, ***"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems"***, *Comm: ACM 21, 2 (Feb. 1978), pp 120-126.*

[8] **D. NACCACHE**, ***"Unless Modified Fiat-Shamir is Insecure"***, *Proceedings of the Third Symposium on State and Progress of Research in Cryptography : SPRC'93, Fondazione Ugo Bordoni (1993), pp 172-180.*

[9] **D. NACCACHE & D. M'RAÏHI**, ***"A strictly DSS-compatible scheme without 1/ k mod q"***, *to appear.*