Cryptanalysis of a Verifiably Committed Signature Scheme based on GPS and RSA

Julien Cathalo*, Benoît Libert** and Jean-Jacques Quisquater

Université catholique de Louvain
Place du Levant 3
1348 Louvain-la-Neuve, Belgium
{cathalo,libert,jjq}@dice.ucl.ac.be

Abstract. This paper describes a powerful attack on a verifiably committed signature scheme based on GPS and RSA proposed in Financial Cryptography 2001. Given any partial signature, the attacker can extract the corresponding full signature. The attack works provided the attacker previously obtained a full signature of a special form, which can be done simply by eavesdropping a very small number of full signatures. For example, with the originally recommended parameters choice, 66% of the signatures are of this form. As a consequence, two "fair" protocols using this primitive do not satisfy the fairness property. Of independent interest, our attack shows that special attention should be paid when building cryptographic protocols from GPS and RSA.

Keywords: Verifiably Committed Signatures, Optimistic Fair Exchange, GPS Signature, Cryptanalysis

1 Introduction

Consider the following situation. Alice wants to buy a CD on Bob's web site, but Alice and Bob do not trust each other. Alice does not want to sign the payment unless she is sure that she will get the CD, and Bob does not want to send the CD unless he is sure that Alice is going to pay him. This is an example of an important issue in electronic commerce and digital rights management: the problem of exchanging items and information in a fair way, which is known as fair exchange. For such situations, a trusted third party, Charlie, is always required. A trivial fair exchange protocol would involve Charlie in any step of the exchange, but it is clear that this solution is not practical. It is thus natural to try and minimize the intervention of Charlie, by designing protocols where Charlie only intervenes in case of problem. Such protocols are called *optimistic* fair exchange protocols.

Several solutions have been proposed to achieve this [1–6, 13]. They all rely on some cryptographic primitive allowing the commitment to a signature. This notion was generalized by Dodis and Reyzin in [7], who proposed a model for non-interactive

* supported partly by *Communauté francaise de Belgique - Actions de Recherche Concertées* and partly by *Walloon Region / WIST-MAIS project*
** supported by *DGTRE / First Europe project*

fair exchange protocols. The model relies on a kind of cryptographic primitives called *verifiably committed signatures* that generalize verifiably encrypted signatures and two-signatures.

The principle of verifiably committed signatures can be summarized as follows. The participants in the protocol are Alice the signer, Bob the verifier, and Charlie the semi-trusted arbitrator. The signer Alice wants to give Bob a signature on a message m, but only if Bob fulfills some obligation I in exchange. Bob does not want to fulfill I until he is sure that he will eventually get Alice's signature on m. In order to do this, Alice first computes a partial signature σ' on m. Bob can check that this partial signature corresponds to Alice's valid signature on m, but cannot extract the valid signature from the partial one. Then Bob fulfills I. Alice then sends the final signature σ on m back to Bob. In case Alice aborts the protocol, Bob can call the arbitrator Charlie, and if Bob can prove that he fulfilled I, Charlie extracts the full signature from the partial one. Thus Charlie is only needed in case of problem.

We recall that a verifiably committed signature scheme should satisfy the following (informal) security requirements [7]:

- *Security against Alice:* Alice should not be able to produce a valid partial signature σ' which Charlie cannot convert into a valid full signature σ.
- *Security against Bob:* Bob should not be able to produce a valid partial signature σ' which he did not get from Alice, and Bob should not be able to produce a valid full signature σ which he did not get from Alice or Charlie.
- *Security against Charlie:* Charlie should not be able to produce a valid full signature σ without seeing a valid partial signature σ' computed by Alice.

In [12], Markowitch and Saeednia proposed an optimistic fair exchange scheme. The scheme relies on a verifiably committed signature scheme that they introduce[1]. The verifiably committed signature scheme is constructed from the basic RSA encryption scheme [15] and the GPS signature scheme [9, 14], and was also used later in the context of a non-repudiation protocol [11].

In this paper, we show that the security against Bob is not satisfied in the verifiably committed signature from [12]. We describe an attack allowing to extract the full signature σ from a partial signature σ'. Our attack is extremely efficient (it only requires that the attacker previously obtained a very small number of full signatures, and has a computational cost of a few exponentiations), and breaks the fairness property of two protocols [12, 11].

Of independent interest, we show how the combination of GPS and RSA can be very sensitive to attacks. Informally, the problem of extracting a RSA root of a GPS commitment has some poor security properties: there is a (non-negligible) subset of the instances of this problem for which the knowledge of a solution allows the computation of the solution for *any* instance.

The remaining of the paper is organized as follows: section 2 briefly describes the basic GPS signature scheme, section 3 describes the verifiably committed signature scheme based on GPS and RSA, section 4 explains the attack and discusses its efficiency, section 5 describes some attempts at repairing the scheme, and section 6 concludes.

[1] The committed signature scheme was not presented separately but appeared as an element of the optimistic fair exchange protocol; nevertheless, from the claimed security properties, it is clear that it is a verifiably committed signature scheme.

2 The GPS Signature Scheme

In this section we briefly describe the GPS signature scheme. It is derived from the GPS identification scheme [9, 14] using the standard method from Fiat and Shamir [8]. It works as follows.

Setup The TTP generates an RSA modulus $n = pq$ where $p = 2p' + 1, q = 2q' + 1$ and p, p', q, q' are primes. It chooses three integers A, B and S such that $A >> BS$, and a hash function h such that the outputs of h are uniformly distributed over $[0, B[$. The TTP chooses an integer α of large order modulo n. It publishes n, α, h. To generate her key pair, the signer chooses a random integer $x \in [0, S[$ as her secret key and computes $y = \alpha^x \bmod n$ as her public key.

Sig To compute a signature on a message m, the signer takes a random integer $r \in [0, A[$ and computes $t = \alpha^r \bmod n$, and $z = r + h(t, m)x$. The signature is (t, z).

Ver To verify a signature (t, z) on m, the verifier checks that $h(t, m) \in [0, B[$, that $z \in [0, A + (B - 1)(S - 1)[$ and that $\alpha^z \equiv ty^{h(t,m)} \pmod n$.

The security of the scheme against existential forgeries under adaptative chosen message attacks was proven (in the random oracle model) in [14] under the hypothesis that computing discrete logarithms with short exponents modulo n is hard.

3 Description of the Committed Signature Scheme

This section describes the verifiably committed signature scheme of [12]. In this scheme, the idea is to use GPS signatures combined with the RSA encryption primitive, a valid signature being a GPS signature along with the RSA decryption of the corresponding commitment. As a consequence, the hardness of extracting a full signature from a partial signature implicitly relies on the problem of computing the RSA decryption of a GPS commitment given the corresponding challenge and answer, but without knowing nor the GPS private key or the RSA private key.

We now describe the committed signature scheme according to the formal model of [7].

Setup Charlie generates an RSA modulus $n = pq$ where $p = 2p' + 1, q = 2q' + 1$ and p, p', q, q' are primes. He chooses a hash function h. Charlie chooses an integer α of order $p'q'$. He also chooses an integer c coprime to $p'q'$, and computes $d = c^{-1} \bmod p'q'$. Charlie publishes n, α, h and c. Charlie's secret arbitration key is d. To generate her key pair, Alice chooses a random integer x as her secret key and computes $y = \alpha^x \bmod n$ as her public key.

PSig To compute a partial signature on a message m, Alice takes a random integer r and computes $t = \alpha^{cr} \bmod n$, and $z = cr + h(t, m)x$. The partial signature is (t, z).

PVer To verify a partial signature (t, z) on m, Bob - or any verifier - checks if $\alpha^z \equiv ty^{h(t,m)} \pmod n$.

Sig To compute a full signature on m corresponding to her partial signature (t, z), Alice computes $t' = \alpha^r \bmod n$, with the value of r used in the partial signature process. The full signature is (t', z) with the same z than in the partial signature.

Ver To verify a full signature, (t', z) on m, Bob - or any verifier - checks if $\alpha^z \equiv t'^c y^{h(t'^c \bmod n, m)} \pmod n$.

Res In case Alice refuses to open her signature to Bob, Charlie uses the partial signature (t, z) and his secret arbitration key d to compute the corresponding final signature (t', z) where $t' = t^d \bmod n$.

The signature protocol is as follows: Alice computes a partial signature (t, z), where $t = \alpha^{cr} \bmod n$, sends this partial signature to Bob, and stores the value $t' = \alpha^r \bmod n$. Bob checks that $\alpha^z \equiv ty^{h(t,m)} \pmod n$, then fulfills his obligation I. Alice sends t' to Bob, and Bob checks that $t'^c \equiv t \pmod n$.

Remarks:

1. Notice that this verifiably committed signature scheme does not exactly fit the model of [7], since both the Sig and Ver algorithm depend on the public arbitration key c.
2. Since $z \bmod c = h(t, m)x \bmod c$, where $z, h(t, m)$ and c are public, it is clear that some information on $x \bmod c$ leaks. For this reason, the authors of [12] recommend to take $c = 3$ to minimize the amount of information leaked on the secret x.

4 Extracting Full Signatures from Partial Signatures

In this section we describe an attack that allows an attacker to extract the full signature (t', z) on m from the partial signature (t, z). The main idea of the attack is that computing t', the c-th root of $t = \alpha^z y^{-e} \bmod n$, is simple when one knows the c-th root of $\alpha^{ex \bmod c} y^{-e \bmod c} \bmod n$ thanks to the euclidian division. We now explain this more in the detail.

4.1 Preliminary: the F function

Let F be the application from $[0, c[$ to $[0, n[$ defined by:

$$F(\overline{e}) = \left(\alpha^{\overline{e}x \bmod c} y^{-\overline{e}}\right)^d \bmod n$$

We are going to show how the properties of this function allow attacking the scheme. In fact the two first properties already show the insecurity of the scheme for small values of c; thus the third property may be viewed as an enhancement of the attack.

A first remark is that $F(0) = 1$..

Property 1 *Knowing a full signature (t', z) allows computing $F(e \bmod c)$, where $e = h(t'^c \bmod n, m)$.*

Proof: Since $t'^c \equiv \alpha^z y^{-e} \bmod n$, we have:

$$\begin{aligned}(t'\alpha^{-(z \text{ div } c)} y^{e \text{ div } c})^c &\equiv \alpha^{z \bmod c} y^{-(e \bmod c)} \pmod n \\ &\equiv \alpha^{ex \bmod c} y^{-(e \bmod c)} \pmod n \qquad (1)\end{aligned}$$

Thus $F(e \bmod c) = t'\alpha^{-(z \text{ div } c)} y^{e \text{ div } c} \bmod n$. □

Property 2 *Given a partial signature (t, z), the knowledge of $F(e \bmod c)$ where $e = h(t \bmod n, m)$ allows computing the corresponding full signature (t', z).*

Proof: From equation (1), we have

$$t' = \alpha^{z \text{ div } c} y^{-(e \text{ div } c)} F(e \bmod c) \bmod n$$

□

Property 3 *Knowing $F(e_0 \bmod c)$ where $e_0 = h(t_0'^c \bmod n, m)$ is coprime to c for some full signature (t_0', z_0) allows computing $F(e)$ for any $e \in [0, c[$.*

Proof: We can compute $k = {e_0}^{-1} e \bmod c$, and we have:

$$\begin{aligned} F(e_0 \bmod c)^{ck} &= \left(\alpha^{e_0 x \bmod c} y^{-e_0}\right)^k \bmod n \\ &= \left(\alpha^{k(e_0 x \bmod c) \text{ div } c} y^{-(ke_0 \text{ div } c)}\right)^c \alpha^{ke_0 x \bmod c} y^{-(ke_0 \bmod c)} \bmod n \\ &= \left(\alpha^{k(z_0 \bmod c) \text{ div } c} y^{-(ke_0 \text{ div } c)}\right)^c (F(ke_0 \bmod c))^c \bmod n \end{aligned}$$

Thus

$$F(e) = F(ke_0 \bmod c) = F(e_0 \bmod c)^k \alpha^{-(k(z_0 \bmod c) \text{ div } c)} y^{ke_0 \text{ div } c} \bmod n \qquad (2)$$

□

4.2 Attack strategy

The attack proceeds in two steps. In the first step, the attacker eavesdrops a small number of full signatures from the signer. In the second step, he asks the signer any number of committed signatures and extracts the corresponding full signature.

1. In this step, the attacker eavesdrops a number of full signatures until he finds a signature (t_0', z_0) (on a message m_0) such that $e_0 \bmod c$ is coprime with c, where $e_0 = h(t_0'^c \bmod n, m_0)$. Once he has obtained such a signature, he computes $F(e_0 \bmod c)$ and stores the triple $(e_0 \bmod c, z_0 \bmod c, F(e_0 \bmod c))$.
2. In this step, the attacker plays the role of the verifier. Once he gets a committed signature (t, z), he computes $F(e \bmod c)$ where $e = h(t \bmod n, m)$, using the stored triple $(e_0 \bmod c, z_0 \bmod c, F(e_0 \bmod c))$ and equation (2). He then uses this value of $F(e \bmod c)$ to extract the full signature from the committed one, according to property 2. The attacker thus obtains the full signature without fulfilling his obligation. This step obviously succeeds with probability 1, and can be done with any number of committed signatures.

4.3 An example with artificially small parameters

We take $c = 3$, and very small values for the other parameters. Alice, whose private key is $x = 110$, takes a random $r = 213$, and computes $t = \alpha^{3 \times 213} \bmod n$. She computes $h(t, m) = 16$, and $z = cr + h(t, m)x = 3 \times 213 + 16 \times 110 = 2399$. The full signature is t' such that $t'^3 \equiv \alpha^{2339} y^{-16} \bmod n$. Since $2339 = 799 \times 3 + 2$ and $16 = 5 \times 3 + 1$, the attacker computes $\tau = t' \alpha^{-799} y^5 \bmod n$. This τ is such that $\tau^3 \equiv \alpha^2 y^{-1} \pmod n$.

Now, given a partial signature $(\widetilde{t}, \widetilde{z} = 1933)$ such that $h(\widetilde{t}, \widetilde{m}) = 14$, he has $\widetilde{t} = \alpha^{1933} y^{-14} \bmod n = \alpha^{3 \times 644 + 1} y^{-3 \times 4 - 2} \bmod n$, thus he needs to find a c-th root of $\alpha y^{-2} \bmod n$. This root is $\tau^2 \alpha^{-1}$ thanks to equation (2).

Thus the attacker can compute $\widetilde{t'} = \alpha^{644} y^{-4} \tau^2 \alpha^{-1} \bmod n$, where $\widetilde{t'}^c \bmod n = \widetilde{t}$, that is, $\widetilde{t'}$ is the full signature corresponding to the committed one.

4.4 Efficiency: discussion on the c parameter

Since the second step always succeeds once the first step ended successfully, the efficiency of the attack only depends on the number of full signatures the attacker has to obtain in the first step. For each signature (t', z), the probability that $e \bmod c$ is coprime with c is $\varphi(c)/c$. Thus the first step succeeds after at most l tries with probability

$$P_l = 1 - \left(1 - \frac{\varphi(c)}{c}\right)^l$$

This formula allows us to evaluate the efficiency of the attack depending on the value of the c parameter.

The $c = 3$ case It is natural to study this single case since it is the value suggested by the authors. We have $\varphi(3) = 2$, thus the attack succeeds after at most l tries with probability

$$P_l = 1 - \frac{1}{3^l}$$

Thus eavesdropping one signature allows a success probability of 66%, while 5 signatures allow a probability of success of more than 99%.

The $c \leq 2^{10}$ case In [12], it is specified that c must be a very small integer. Thus we can assume that $c < 2^{10}$. To compute a lower bound of $\varphi(c)/c$ for values of c between 2 and 2^{10}, we simply compute all the possible values of $\varphi(c)/c$. The lower bound is reached for $c = 210$. Thus we have that

$$\forall c \in [1, 2^{10}], \quad \frac{\varphi(c)}{c} \geq \frac{\varphi(210)}{210} = \frac{8}{35} \approx 0.22857$$

While with a single signature, the lower bound on the probability of success is already 22%, we can see that, with 10 signatures, we have a probability of success of at least 92%.

5 Attempts to repair the scheme

5.1 Increasing the value of c

A natural idea to repair the scheme, since the efficiency of our attack depends on $\varphi(c)/c$, is to minimize this value by taking a larger and specific value for c. Of course, since $x \bmod c$ leaks during any execution of the protocol, we have to take a larger value for x. More precisely, we have to ensure that $x \operatorname{div} c$ is large enough (160 bits) because $y\alpha^{-x \bmod c} = (\alpha^c)^{x \operatorname{div} c} \pmod n$ thus finding x is equivalent to solve the discrete logarithm of $y\alpha^{-x \bmod c} \bmod n$ in basis $\alpha^c \bmod n$. This implies a performance decrease since the size of the commitments must be such that $r >> 2^{|h|}c$.

But we have the following well-known bound: when c is any number such that $c \geq 5$,

$$\varphi(c) \geq \frac{c}{6 \ln \ln c} \tag{3}$$

Thus even if c is 4000 bit long, then 2.1% of the full signatures are such that e is coprime with c. Thus with 33 full signatures the attacker will succeed with probability 1/2.

In short, taking a larger value for c does not prevent the attack, but just makes it a little less efficient. We conclude that the scheme is insecure for *any* value of c.

5.2 Other attempts

One might also try to remove some multiplicative properties to prevent the attack, but these multiplicative properties seem necessary to ensure the verifiability of partial signatures.

Another attempt to repair the scheme makes use of non-interactive proofs-of-knowledge. The idea is that the attack is possible because t', the c-th root of t, is revealed by Alice. Thus we keep the same protocol except for the full signature process. Alice proves that she knows t', but this time she does not reveal it. This can be done with a non-interactive zero-knowledge proof-of-knowledge of c-th root of t, such as a non-interactive version of the Guillou-Quisquater [10] protocol, for example. However, this does not lead to an efficient verifiably committed signature scheme.

6 Conclusion

We showed that the security against the verifier was not verified in the committed signature scheme from [12] (also used in [11]) by describing a very simple and fast method to extract a full signature from a partial signature.

This attack is a new example of a well-known fact: constructing a new cryptographic primitive from two secure primitives can lead to a totally insecure result. But, interestingly, we notice that GPS and RSA can be securely combined; in fact, the RSA primitive was already used in the initial design of GPS, with self-certified public keys [9].

We conclude that the security of a cryptographic protocol should not rely on the hardness of computing the RSA decryption of a GPS commitment.

References

1. N. Asokan, V. Shoup, and M. Waidner. Optimistic Fair Exchange of Digital Signatures. In K. Nyberg, editor, *Advances in Cryptology - Eurocrypt 98*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606. Springer-Verlag, 1998.
2. N. Asokan, V. Shoup, and M. Waidner. Optimistic Fair Exchange of Digital Signatures. *IEEE Journal on Selected Areas in Communication*, 18(4):593–610, 2000.
3. G. Ateniese. Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures. In G. Tsudik, editor, *Sixth ACM Conference on Computer and Communication Security and Privacy*, pages 138–146. ACM, November 1999.
4. F. Bao, R. Deng, and W. Mao. Efficient and Practical Fair Exchange Protocols with Off-line TTP. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 77–85, 1998.
5. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In E. Biham, editor, *Advances in Cryptology - Eurocrypt 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer-Verlag, 2003.
6. J. Camenisch and I.B. Damgård. Verifiable Encryption, Group Encryption, and their Application to Group Signatures and Signature Sharing Schemes. In T. Okamoto, editor, *Advances in Cryptology - Asiacrypt 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer-Verlag, 2000.

7. Y. Dodis and L. Reyzin. Breaking and Repairing Optimistic Fair Exchange from PODC 2003. In M. Yung, editor, *ACM Workshop on Digital Rights Management (DRM)*, 2003. available at http://eprint.iacr.org/2003/146/.
8. A. Fiat and A. Shamir. How to prove yourself : practical solutions of identification and signature problems. In G. Brassard, editor, *Advances in Cryptology - Proceedings of CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1987.
9. M. Girault. Self-Certified Public Keys. In D.W. Davies, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 1991*, volume 0547 of *Lecture Notes in Computer Science*, pages 490–497. Springer, 1991.
10. L. Guillou and J.-J. Quisquater. A "Paradoxical" Identity-Based Signature Scheme Resulting from Zero-Knowledge Minimizing Both Transmission and Memory. In C.G. Günther, editor, *Advances in Cryptology - Proceedings of CRYPTO 1988*, pages 216–231. Springer, 1988.
11. O. Markowitch and S. Kremer. An Optimistic Non-Repudiation Protocol with Transparent Trusted Third Party. In G.I. Davida and Y. Frankel, editors, *Proceedings of the 4th International Conference on Information Security (ISC 2001)*, volume 2200 of *Lecture Notes in Computer Science*, pages 363–378. Springer, 2001.
12. O. Markowitch and S. Saeednia. Optimistic Fair Exchange with Transparent Signature Recovery. In P.F. Syverson, editor, *Proceedings of Financial Cryptography 2001*, volume 2339 of *Lecture Notes in Computer Science*, pages 339–350. Springer, 2002.
13. J.M. Park, E. Chong, H. Siegel, and I. Ray. Constructing Fair Exchange Protocols for E-Commerce via Distributed Computation of RSA Signatures. In *Proceedings of the 22th Annual ACM Symposium on Principles of Distributed Computing (PODC 2003)*, pages 172–181, July 2003.
14. G. Poupard and J. Stern. Security Analysis of a Practical *on the fly* Authentification and Signature Generation. In K. Nyberg, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 1998*, volume 1403 of *Lecture Notes in Computer Science*, pages 422–436. Springer, 1998.
15. R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. LCS TM82, MIT Laboratory for Computer Science, Cambridge, Massachusetts, 1977.