The Béguin-Quisquater Server-Aided RSA Protocol from Crypto '95 is not Secure

Phong Nguyen and Jacques Stern

École Normale Supérieure
Laboratoire d'Informatique
45, rue d'Ulm
F – 75230 Paris Cedex 05
{Phong.Nguyen,Jacques.Stern}@ens.fr
http://www.dmi.ens.fr/~{pnguyen,stern}/

Abstract. A well-known cryptographic scenario is the following: a smart card wishes to compute an RSA signature with the help of an untrusted powerful server. Several protocols have been proposed to solve this problem, and many have been broken. There exist two kinds of attacks against such protocols: passive attacks (where the server follows the instructions) and active attacks (where the server may return false values). An open question in this field is the existence of efficient protocols (without expensive precomputations) provably secure against both passive and active attacks. At Crypto '95, Béguin and Quisquater tried to answer this question by proposing an efficient protocol which was resistant against all known passive and active attacks. In this paper, we present a very effective lattice-based passive attack against this protocol. An implementation is able to recover the secret factorization of an RSA-512 or RSA-768 key in less than 5 minutes once the card has produced about 50 signatures. The core of our attack is the basic notion of an orthogonal lattice which we introduced at Crypto '97 as a cryptographic tool.

1 Introduction

Small units like chip cards or smart cards have the possibility of computing, storing and protecting data. Today, some of these cards include fast and secure coprocessors allowing to quickly perform the expensive operations needed by public key cryptosystems. But most of the cards are cheap cards with too limited computing power for such tasks. To overcome this problem, extensive research has been conducted under the generic name "server-aided secret computations" (SASC). In the SASC protocol, the client (the smart card) wants to perform a secret computation (*e.g.*, RSA signature generation) by borrowing the computing power of an untrusted powerful server without revealing its secret information. One distinguishes two kinds of attacks against such protocols: attacks where the server respects the instructions are called *passive attacks*, while attacks where the server may return false computations are called *active attacks*.

The first SASC protocol was proposed by Matsumoto, Kato and Imai [9] in the case of RSA signatures [14]. Pfitzmann and Waidner [13] presented several

K. Ohta and D. Pei (Eds.): ASIACRYPT'98, LNCS 1514, pp. 372–379, 1998.

passive attacks against all the protocols of [9], and Anderson [1] described an efficient active attack against one of the protocols of [9]. Several new protocols such as [8,5,4,2,7] have been proposed since. Among these, the protocol of Béguin and Quisquater [2] was quite attractive: it was relatively efficient (since it was based on the fast exponentiation algorithm due to Brickell, Gordon, McCurley and Wilson [3]), did not require expensive precomputations (contrary to most of the proposed protocols), and was secure against all known passive and active attacks, including some lattice-based passive attacks.

We present a very effective lattice-based passive attack against this protocol. Our implementation shows that a server is able to recover the secret factorization of the RSA key (512, 768 or 1024 bits) in less than 5 minutes, once the card has produced about 50 signatures, for all the choices of parameters suggested by Béguin and Quisquater. To run the attack, the server needs to store very few information. The core of our attack is the basic notion of an orthogonal lattice which we recently introduced as a cryptographic tool in [10]. As in [10,12,11], this technique enables us to use the linearity hidden in the protocol, and results in a simple heuristic attack which is devastating in pratice. An open question remains: does there exist a server-aided RSA signature protocol which is both efficient (without requiring expensive precomputations) and provably secure against passive and active attacks ?

The rest of the paper is organized as follows. In section 2, we make a short description of the Béguin-Quisquater server-aided RSA signature protocol. We refer to [2] for more details. In section 3, we recall some facts from [10] about the notion of an orthogonal lattice. Finally, we present our attack in section 4 and the experiments in section 5.

2 The Béguin-Quisquater Protocol

Let $n = pq$ be a RSA public modulus with a secret exponent s and a public exponent v. We have $sv \equiv 1 \pmod{\phi(n)}$ with $\phi(n) = (p-1)(q-1)$. Denote by $\ell(x)$ the bit-length of an integer x. Let $t = \max(\ell(p), \ell(q)) - 1$. In practice, one can assume that $\ell(p) = \ell(q) = t+1$. Using the Extended Euclidean Algorithm, compute integers w_p and w_q less than n in absolute value such that $w_p + w_q = 1$, p divides w_p and q divides w_q. Thus, if $y_p \equiv y \pmod{p}$ and $y_q \equiv y \pmod{q}$ then $y \equiv y_p w_q + y_q w_p \pmod{n}$. The protocol uses two integer parameters m and h, and is as follows:

1. The card receives M to sign.
2. The card chooses random integers $a_0, \ldots, a_{m-1}$ in $\{0, \ldots, h\}$ and $x_0, \ldots, x_{m-1}$ such that $\ell(x_i) \leq t - \log_2(mh) - 2$.
3. The card computes $s_1 = \sum_{i=0}^{m-1} a_i x_i$.
4. The card sends $M, n, x_0, \ldots, x_{m-1}$ to the server.
5. The server returns $z_0, \ldots, z_{m-1}$ where $z_i = M^{x_i} \bmod n$.
6. The card computes $z_p = \prod_{i=0}^{m-1} z_i^{a_i} \bmod p$ and $z_q = \prod_{i=0}^{m-1} z_i^{a_i} \bmod q$ using the algorithm of [3] for fast exponentiation with precomputation.

7. The card computes $s_2 = s - s_1$ and represents s_2 under the form:

$$\begin{aligned}\sigma_p &= s_2 \mod (p-1) + \varrho_p(p-1)\\ \sigma_q &= s_2 \mod (q-1) + \varrho_q(q-1)\end{aligned}$$

   where ϱ_p is a random number in $\{0, \ldots, q-2\}$ and ϱ_q is a random number in $\{0, \ldots, p-2\}$.
8. The card sends σ_p and σ_q to the server.
9. The server computes and sends to the card $y_p = M^{\sigma_p} \mod n$ and $y_q = M^{\sigma_q} \mod n$.
10. The card computes $S_p = y_p z_p \mod p$ and $S_q = y_q z_q \mod q$.
11. Next, the card computes $S = w_q S_p + w_p S_q \mod n$.
12. The card verifies $M \equiv S^v \mod n$.
13. If the verification is correct, then the card transmits S.

In their paper [2], Béguin and Quisquater analyzed several passive and active attacks, including some lattice-based passive attacks. They concluded that their protocol was secure against all known passive and active attacks, for 4 different sets of parameters (valid for both RSA-512 and RSA-768), which are summarized in the following table:

	Case 1	Case 2	Case 3	Case 4
h	10	7	17	11
m	19	22	25	29

The resulting protocol was quite efficient. It only required about 30 modular multiplications for the card. The needed RAM and the data transfers between the card and the server were small, and the precomputations were not expensive.

3 The Orthogonal Lattice

We recall a few useful facts about the notion of an orthogonal lattice, which was introduced as a cryptographic tool in [10]. Let L be a lattice in $\mathbb{Z}^n$ where n is any integer. The orthogonal lattice $L^\perp$ is defined as the set of elements in $\mathbb{Z}^n$ which are orthogonal to all the lattice points of L, with respect to the usual dot product. We define the lattice $\bar{L} = (L^\perp)^\perp$ which contains L and whose determinant divides the one of L. The results of [10] which are of interest to us are the following two theorems:

Theorem 1. *If L is a lattice in $\mathbb{Z}^n$, then $\dim(L) + \dim(L^\perp) = n$ and:*

$$\det(L^\perp) = \det(\bar{L}).$$

Thus, $\det(L^\perp)$ divides $\det(L)$. This implies that if L is a low-dimensional lattice in $\mathbb{Z}^n$, then a reduced basis of $L^\perp$ will consist of very short vectors compared to a reduced basis of L. In practice, most of the vectors of any reduced basis of $L^\perp$ are quite short, with norm around $\det(\bar{L})^{1/(n-\dim L)}$.

Theorem 2. *There exists an algorithm which, given as input a basis* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ *of a lattice* L *in* $\mathbb{Z}^n$*, outputs an LLL-reduced basis of the orthogonal lattice* $L^\perp$*, and whose running time is polynomial with respect to* n*,* d *and any upper bound of the bit-length of the* $\|\mathbf{b}_j\|$*'s.*

In practice, one obtains a simple and very effective algorithm (which consists of a single lattice reduction, described in [10]) to compute a reduced basis of the orthogonal lattice, thanks to the celebrated LLL algorithm [6]. This means that, given a low-dimensional L in $\mathbb{Z}^n$, one can easily compute many short and linearly independent vectors in $L^\perp$.

4 A Simple Attack

Throughout the attack, only steps 2, 3, 7 and 8 of the protocol will be of interest.

Assume that the card computes $r+1$ signatures. Denote by $s_1^{[i]}, \varrho_p^{[i]}, \varrho_q^{[i]}, \sigma_p^{[i]}$ and $\sigma_q^{[i]}$ the values used by the card to compute the i-th signature. Define the following vectors in $\mathbb{Z}^r$ which consist of successive differences:

$$\begin{aligned}
\Delta s_1 &= \left(s_1^{[2]} - s_1^{[1]}, s_1^{[3]} - s_1^{[2]}, \ldots, s_1^{[r+1]} - s_1^{[r]}\right) \\
\Delta \sigma_p &= \left(\sigma_p^{[2]} - \sigma_p^{[1]}, \sigma_p^{[3]} - \sigma_p^{[2]}, \ldots, \sigma_p^{[r+1]} - \sigma_p^{[r]}\right) \\
\Delta \sigma_q &= \left(\sigma_q^{[2]} - \sigma_q^{[1]}, \sigma_q^{[3]} - \sigma_q^{[2]}, \ldots, \sigma_q^{[r+1]} - \sigma_q^{[r]}\right)
\end{aligned}$$

By definition of the $\sigma_p^{[i]}$'s and $\sigma_q^{[i]}$'s, the following equations hold:

$$\Delta\sigma_p + \Delta s_1 \equiv 0 \pmod{p-1} \tag{1}$$

$$\Delta\sigma_q + \Delta s_1 \equiv 0 \pmod{q-1} \tag{2}$$

The server knows $\Delta\sigma_p$ and $\Delta\sigma_q$ by step 8, but not Δs_1. These vectors were also considered by Béguin and Quisquater when they analyzed some lattice-based passive attacks, but this is the only similarity between these attacks and the attack we present. We will see that short vectors orthogonal to $\Delta\sigma_p$ (resp. $\Delta\sigma_q$) give information on q (resp. p). If we find enough such independent vectors, then q (resp. p) is revealed. Fortunately, the previous section shows that it is not hard to do so, provided that r is sufficiently large.

We start with two simple remarks:

Lemma 3. *Let* $\mathbf{u} \in \mathbb{Z}^r$*. If* $\mathbf{u} \perp \Delta\sigma_p$ *then* $\mathbf{u} \perp \Delta s_1$ *or* $\|\mathbf{u}\| \geq (p-1)/\|\Delta s_1\|$.

Proof. By (1), we have $\mathbf{u}.\Delta s_1 \equiv 0 \pmod{p-1}$ and the result follows by Cauchy-Schwarz. □

Lemma 4. *Let* $\mathbf{u} \in \mathbb{Z}^r$*. If* $\mathbf{u} \perp \Delta s_1$ *then* $(q-1)$ *divides* $\mathbf{u}.\Delta\sigma_q$.

Proof. Straightforward from (2). □

This shows that if $\mathbf{u} \perp \Delta\sigma_p$ then $(q-1)$ divides $\mathbf{u}.\Delta\sigma_q$, or $\|\mathbf{u}\| \geq (p-1)/\|\Delta s_1\|$. We notice that the latter case implies that $\mathbf{u}$ is relatively long, because the entries of Δs_1 are smaller than $p-1$, as the following lemma shows:

Lemma 5. *Each entry of Δs_1 is in absolute value less than 2^{t-2}.*

Proof. In Step 2, each $s_1^{[i]}$ is a sum of m integers of form ax where $0 \leq a \leq h$ and $\ell(x) \leq \lfloor t - \log_2(mh) - 2 \rfloor$. Therefore:

$$0 \leq s_1^{[i]} \leq mh 2^{\lfloor t - \log_2(mh) - 2 \rfloor} \leq 2^{t-2}.$$

The result follows. □

Actually, the previous upper bound is quite pessimistic. In practice, experiments show that when the choices of Step 2 are indeed random, the entries of Δs_1 are in absolute value less than 2^{t-5}, and on the average around 2^{t-6}. This has to be compared with $\ell(p) = \ell(q) = t+1$. This phenomenon is explained by the following technical lemma:

Lemma 6. *If the random choices of Step 2 are independent and uniformly distributed, then the entries of Δs_1 have zero mean and a variance equal to*

$$\frac{(2h+1)(2^k-1)(2^{k+1}-1)}{18} mh + \frac{2^{2k}h^2}{8}(m^2-m),$$

where k is the integer $\lfloor t - \log_2(mh) - 2 \rfloor$.

Proof. A simple calculation shows that:

$$E(a_i) = \frac{h}{2} \qquad E(a_i^2) = \frac{h(2h+1)}{6}$$

$$E(x_i) = \frac{2^k-1}{2} \qquad E(x_i^2) = \frac{(2^k-1)(2^{k+1}-1)}{6}$$

Therefore $E(s_1) = \frac{(2^k-1)}{4} mh$ and by independence,

$$E(s_1^2) = E\left(\left(\sum_{i=0}^{m-1} a_i x_i\right)^2\right) = mE(a_0^2)E(x_0^2) + (m^2-m)E(a_0)^2E(x_0)^2.$$

Hence, each entry of Δs_1 has zero mean and a variance equal to $2E(s_1^2)$. □

Let σ be the standard deviation of the entries of Δs_1. The following table gives the value of $(t+1) - \log_2 \sigma$ (which indicates the size difference between $q-1$ and the entries of Δs_1) for the 4 different choices of parameters. This value is almost independent of t: there is no difference between RSA-512, RSA-768 and RSA-1024.

	Case 1	Case 2	Case 3	Case 4
h	10	7	17	11
m	19	22	25	29
$(t+1) - \log_2 \sigma$	5.4	5.7	5.2	5.7

Thus, an orthogonal vector to $\Delta\sigma_p$ (resp. $\Delta\sigma_q$) is either relatively long, or such that $q-1$ (resp. $p-1$) divides its dot product with $\Delta\sigma_q$ (resp. $\Delta\sigma_p$). Note that, since vectors $\Delta\sigma_p$ and $\Delta\sigma_q$ are generated using the random values $\rho_p^{[i]}$'s and $\rho_q^{[i]}$'s, there is no intrinsic reason why a vector orthogonal to one of them should also be orthogonal to the other. Thus, if $\mathbf{u}$ is orthogonal to $\Delta\sigma_p$, the dot product $\mathbf{u}.\Delta\sigma_q$ is a non zero multiple of $q-1$. This implies that if we find several short vectors orthogonal to $\Delta\sigma_p$ (resp. $\Delta\sigma_q$), then $q-1$ (resp. $p-1$) will be revealed by simple gcds.

The previous section shows that one can expect to find (in polynomial time) many independent vectors orthogonal to $\Delta\sigma_p$ with norm around

$$\|\Delta\sigma_p\|^{1/(r-1)} \approx (2^{2t}\sqrt{r})^{1/(r-1)}.$$

When r is sufficiently large, the vectors are short enough to reveal $q-1$, and therefore the factorization. Finally, our attack is the following:

1. Compute a reduced basis of $(\Delta\sigma_p)^{\perp}$.
2. Consider the shortest vectors in this basis (a few are enough) and compute their dot product with $\Delta\sigma_q$.
3. Compute the gcd of all these dot products and check whether it is $q-1$.

In practice, only Step 1 takes a little time. Note that the server only needs to store the $\sigma_p^{[i]}$'s and the $\sigma_q^{[i]}$'s (not even the signatures) to run the attack.

5 Experiments

We implemented the attack using the NTL package [15] which includes efficient lattice-reduction algorithms. We used the LLL floating point version with extended exponent to compute orthogonal lattices, since the entries of $\Delta\sigma_p$ were too large (about the size of n) for the usual floating point version. In practice, the attack reveals the secret factorization as soon as r (the number of signatures) is large enough, and the total computation time is less than 5 minutes on a UltraSparc-I clocked at 167 MHz, when r is less than 70. It actually takes more time to generate the signatures along with the different parameters than to recover the factorization.

The following table shows the practical number of RSA signatures which are necessary to make the attack successful, for different key sizes and choices of parameters.

Minimal number of signatures

	Case 1	Case 2	Case 3	Case 4
h	10	7	17	11
m	19	22	25	29
RSA-512	53	50	56	53
RSA-768	54	52	56	54
RSA-1024	62	60	63	62

When r reaches these values, at least the 10 shortest vectors of the reduced basis are also orthogonal to Δs_1. Generally, 5 of them are enough to reveal $q-1$.

When r is larger, most of the vectors of the reduced basis are very short and have similar norms, and their dot product with $\Delta\sigma_q$ is a non-zero multiple of $q-1$. As previously, we only need a few of them to discover the factorization.

6 Conclusion

We presented a simple passive attack against the Béguin-Quisquater server-aided RSA protocol. It is based on the basic notion of an orthogonal lattice. This notion was introduced as a useful tool in a paper published last year, which cryptanalyzed a knapsack-like cryptosystem proposed by Qu and Vanstone. We applied this technique in a different manner, but the success of our attack relies on the main property of orthogonal lattices as well: given a low-dimensional lattice, one can easily find many short and linearly independent vectors in the corresponding orthogonal lattice.

The attack has been implemented, and is devastating in practice, for all the choices of parameters suggested by Béguin and Quisquater. Once the card has produced about 50 signatures, the server can quickly recover the secret factorization of the RSA key, without storing much information. This shows that the Béguin-Quisquater server-aided RSA protocol is not secure, and stresses the importance of provable security as opposed to security against all known attacks. The existence of a server-aided RSA signature protocol which is both efficient (without requiring expensive precomputations) and provably secure against passive and active attacks remains open.

References

1. R. J. Anderson. Attack on server assisted authentication protocols. *Electronic Letters*, 28(15):1473, 1992.
2. P. Béguin and J.-J. Quisquater. Fast server-aided RSA signatures secure against active attacks. In *Proc. of Crypto '95*, volume 963 of *LNCS*, pages 70–83. Springer, 1995.
3. E. Brickell, D. M. Gordon, K. S. McCurley, and D. Wilson. Fast exponentiation with precomputation. In *Proc. of Eurocrypt '92*, volume 658 of *LNCS*, pages 200–207. Springer, 1993.

4. J. Burns and C. J. Mitchell. Parameter selection for server-aided RSA computation schemes. *IEEE Transactions on Computers*, 43, 1994.
5. S. Kawamura and A. Shimbo. Fast server-aided secret computation protocols for modular exponentiation. *IEEE Journal on Selected Areas Communications*, 11, 1993.
6. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.
7. C. H. Lim and P. J. Lee. Security and performance of server-aided RSA computation protocols. In *Proc. of Crypto '95*, volume 963 of *LNCS*, pages 70–83. Springer, 1995.
8. T. Matsumoto, H. Imai, C.-S. Laih, and S.-M. Yen. On verifiable implicit asking protocols for RSA computation. In *Proc. of Auscrypt '92*, volume 718 of *LNCS*, pages 296–307. Springer, 1993.
9. T. Matsumoto, K. Kato, and H. Imai. Speedings up secret computation with insecure auxiliary devices. In *Proc. of Crypto '88*, volume 403 of *LNCS*, pages 497–506. Springer, 1989.
10. P. Nguyen and J. Stern. Merkle-Hellman revisited: a cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 198–212. Springer-Verlag, 1997.
11. P. Nguyen and J. Stern. Cryptanalysis of a fast public key cryptosystem presented at SAC' 97. In *Proc. of SAC '98*, LNCS. Springer-Verlag, 1998.
12. P. Nguyen and J. Stern. Cryptanalysis of the Ajtai-Dwork cryptosystem. In *Proc. of Crypto '98*, volume 1462 of *LNCS*. Springer-Verlag, 1998.
13. B. Pfitzmann and M. Waidner. Attacks on protocols for server-aided RSA computation. In *Proc. of Eurocrypt '92*, volume 658 of *LNCS*, pages 153–162. Springer, 1993.
14. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.
15. V. Shoup. NTL computer package version 2.0. Can be obtained at `http://www.cs.wisc.edu/~shoup/ntl`.