On the Insecurity of a Server-Aided RSA Protocol

Phong Q. Nguyen[1⋆] and Igor E. Shparlinski[2⋆⋆]

[1] CNRS/Département d'Informatique, École normale supérieure
45 rue d'Ulm, 75005 Paris, France
pnguyen@ens.fr http://www.di.ens.fr/~pnguyen/
[2] Department of Computing, Macquarie University
Sydney, NSW 2109, Australia
igor@ics.mq.edu.au http://www.comp.mq.edu.au/~igor/

Abstract. At Crypto '88, Matsumoto, Kato and Imai proposed a protocol, known as RSA-S1, in which a smart card computes an RSA signature, with the help of an untrusted powerful server. There exist two kinds of attacks against such protocols: passive attacks (where the server does not deviate from the protocol) and active attacks (where the server may return false values). Pfitzmann and Waidner presented at Eurocrypt '92 a passive meet-in-the-middle attack and a few active attacks on RSA-S1. They discussed two simple countermeasures to thwart such attacks: renewing the decomposition of the RSA private exponent, and checking the signature (in which case a small public exponent must be used). We present a new lattice-based provable passive attack on RSA-S1 which recovers the factorization of the RSA modulus when a very small public exponent is used, for many choices of the parameters. The first countermeasure does not prevent this attack because the attack is a one-round attack, that is, only a single execution of the protocol is required. Interestingly, Merkle and Werchner recently provided a security proof of RSA-S1 against one-round passive attacks in some generic model, even for parameters to which our attack provably applies. Thus, our result throws doubt on the real significance of security proofs in the generic model, at least for server-aided RSA protocols. We also present a simple analysis of a multi-round lattice-based passive attack proposed last year by Merkle.

Keywords: Cryptanalysis, RSA signature, Server-aided protocol, Lattices.

1 Introduction

Small units like chip cards or smart cards have the possibility of computing, storing and protecting data. Today, many of these cards include fast and secure coprocessors allowing to quickly perform the expensive operations needed

⋆ Work supported in part by the RNRT "Turbo-signatures" project of the French Ministry of Research.
⋆⋆ Work supported in part by the Australian Research Council.

C. Boyd (Ed.): ASIACRYPT 2001, LNCS 2248, pp. 21–35, 2001.

by public key cryptosystems. However, a large proportion of the cards consists of 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 (for example, 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 follows rigorously the protocol are called *passive attacks*, while attacks where the server may return false computations are called *active attacks*. Attacks are called multi-round when they require several executions of the protocol between the same parties.

Most of the SASC protocols proposed for RSA signatures have been shown to be either inefficient or insecure (see for instance the two recent examples [13,10]), which explains why, to our knowledge, none of these protocols has ever been used in practice. Many of these protocols are variants of the protocols RSA-S1 and RSA-S2 proposed by Matsumoto, Kato and Imai [8] at Crypto '88, which use a random linear decomposition of the RSA private exponent. At Eurocrypt '92, Pfitzmann and Waidner [15] presented several natural meet-in-the-middle passive attacks and some efficient active attacks against RSA-S1 and RSA-S2. To prevent such attacks, they discussed two countermeasures which should be used together: one is to renew the decomposition of the private exponent at each signature, the other is to check the signature before the end of the protocol, which is a well-known countermeasure but requires a very small public exponent since the check is performed by the card.

The first countermeasure was effective against the original active attacks of [15], but Merkle [10] showed last year at ACM CCS '00 that the resulting scheme was still insecure. Indeed, he presented an efficient lattice-based multi-round passive attack, which was successful (in practice) against many choices of the parameters. Merkle's paper [10] included an analysis of the attack, inspired by well-known lattice-based methods [5] to solve the subset sum problem. However, the analysis was rather technical and not exactly correct (it assumed a distribution of the parameters which was not the one induced by the protocol). We present a simple analysis of a slight variant of Merkle's attack, which enables to explain experimental results, and to provide provable results for certain choices of the parameters.

The main contribution of this paper is a new lattice-based passive attack which recovers the private exponent (like Merkle's attack), but only in the case a very small public exponent is used (which is the second countermeasure). Interestingly, this attack is only one-round in the sense that a single execution of the protocol is sufficient, whereas Merkle's attack is multi-round, requiring many signatures produced by the card with the help of the same server. Consequently, the first countermeasure has no impact on this new attack. And these results point out the limits of the *generic model*, as applied to the security analysis of server-aided RSA protocols. Indeed, Merkle and Werchner [11] proved at PKC '98 that the RSA-S1 protocol was secure against one-round passive attacks

in the generic model, in the sense that all generic attacks have complexity at least that of a square-root attack (better than the meet-in-the-middle attack presented by Pfitzmann and Waidner [15]). Roughly speaking, in this context, generic attacks (see [11] for a precise definition) do not take advantage of special properties of the group used. However, our attack shows that the RSA-S1 scheme is not even secure against one-round passive attacks in the standard model of computation. In particular, the attack provably works against certain choices of the parameters to which the square-root attack cannot apply. Thus, contrary to what Merkle and Werchner claimed in [11], the generic model is not appropriate for investigating the security of server-aided RSA protocols.

The rest of the paper is organized as follows. In Section 2, we make a short description of the RSA-S1 server-aided protocol and review some useful background. We refer to [8,15] for more details. In Section 3, we present our variant of Merkle's lattice-based attack, together with an analysis. In Section 4, we present our new lattice-based attack on low-exponent RSA-S1.

2 Background

2.1 The RSA-S1 Server-Aided Protocol

Let N be an RSA-modulus and let φ denote the Euler function. Let e and d be respectively the RSA public and private exponents:

$$ed \equiv 1 \pmod{\varphi(N)}.$$

For an integer s we denote by $[s]$ the set of integers of the interval $[0, s-1]$ and by $[s]_{\pm}$ the set of integers of the interval $[-s+1, s-1]$.

Let k, ℓ and m be positive integers and let $\mathcal{B}_{k,\ell,m}$ be the set of vectors

$$\mathbf{f} = (f_1, \ldots, f_m) \in \left[2^{\ell}\right]^m$$

with $\gcd(f_1, \ldots, f_m, \varphi(N)) = 1$ and with

$$\sum_{i=1}^{m} \mathrm{wt}(f_i) = k, \tag{1}$$

where $\mathrm{wt}(f)$ denotes the Hamming weight, that is, the sum of binary digits of an integer $f \geq 0$.

The RSA-S1 server-aided protocol from [8] computes an RSA signature $x^d \pmod{N}$ with the help of an (untrusted) server in the following way:

The RSA-S1 Protocol.

Step 1 The card selects a vector $\mathbf{f} = (f_1, \ldots, f_m) \in \mathcal{B}_{k,\ell,m}$ at random accordingly to any fixed probability distribution.

Step 2 The card sends a vector $\mathbf{d} = (d_1, \ldots, d_m) \in [\varphi(N)]^m$ chosen uniformly at random from the set of vectors satisfying the congruence

$$\sum_{i=1}^{m} f_i d_i \equiv d \pmod{\varphi(N)}, \tag{2}$$

if possible. Otherwise the card returns to Step 1.

Step 3 The card asks the server to compute and return $z_i \equiv x^{d_i} \pmod{N}$, $i = 1, \ldots, m$.

Step 4 The card computes

$$x^d \equiv \prod_{i=1}^{m} z_i^{f_i} \pmod{N}.$$

Our description follows the presentation of [10] rather than the one of the original paper [8]. For instance, [8] asks that $\sum_{i=1}^{m} \mathrm{wt}(f_i) \le k$ instead of (1) but this difference is marginal as all our results can easily be adapted to this case.

For Step 4, the card mainly has two possibilities, due to memory restrictions. One is the square-and-multiply method, which requires at most $k\ell$ modular multiplications and very little memory. The other is the algorithm of [4], which enables to compute $\prod_{i=1}^{m} z_i^{f_i} \pmod{N}$ efficiently but requires more memory than the square-and-multiply method. When using this algorithm, to optimize the choice of the parameters, one should remove the restriction (1) and replace the choice $f_i \in \left[2^\ell\right]$ by $f_i \in [h]$ where h is some small integer, not necessarily a power of 2. The algorithm then requires at most $m+h-3$ modular multiplications, and the temporary storage of either m or $h-1$ elements, according to whether the card stores all the m elements $z_1, \ldots, z_m$, or the $h-1$ elements $t_j = \prod_{f_i=j} z_i$, $1 \le j < h$ (which must be computed upon reception of the z_i's). Other known tricks to speed-up the computation of products of exponentiations (see [6] and [9, Sect. 14.6]) do not seem to be useful in this context.

The protocol requires the transfer of approximately $2m \log N$ bits. Since the bandwidth of a cheap smartcard is typically 9600 bauds, this means that m must be restricted to low values. For instance, with a 1024-bit modulus, the value $m = 50$ already represents 10.7 seconds.

2.2 Passive Attacks on RSA-S1

Notice that the protocol is broken as soon as the f_i's are disclosed. Indeed, the integer $\sum_{i=1}^{m} f_i d_i$ is congruent to the RSA private exponent modulo $\varphi(N)$, and therefore enables to sign any message (and this can be checked thanks to the public exponent e). And, of course, one may further recover the factorization of N in randomized polynomial time, from $e \sum_{i=1}^{m} f_i d_i - 1$ which is a non-zero multiple of $\varphi(N)$ (see for instance [9, Section 8.2.2]).

The authors of [8] claimed that the only possible passive attack was to exhaustive search the f_i's, which requires roughly C operations where:

$$C = \binom{m\ell}{k}.$$

But obviously, one can devise simple meet-in-the-middle passive attacks. Pfitzmann and Waidner [15] noticed that one could split $(f_1, \ldots, f_m)$ as $(g_1, \ldots, g_m) + (h_1, \ldots, h_m)$ where $\sum \mathrm{wt}(g_i) \leq \sum \mathrm{wt}(h_i) = \lceil k/2 \rceil$, and deduced an attack with time and space complexity roughly:

$$\binom{m\ell}{\lceil k/2 \rceil}.$$

The attack of [15] is however not optimal: the complexity can easily be improved using a trick used by Coppersmith [18] in a meet-in-the-middle attack against the discrete logarithm problem with low Hamming weight. By choosing random subsets of cardinality $\lceil m\ell/2 \rceil$ inside $\{1, \ldots, m\ell\}$, one obtains a randomized meet-in-middle-attack with time and space complexity roughly:

$$\sqrt{k}\binom{\lceil m\ell/2 \rceil}{\lceil k/2 \rceil}.$$

Thus, we obtain an attack of complexity roughly the square root $\sqrt{C}$ of that of exhaustive search. Therefore in our numerical experiments we mainly consider sets of parameters for which $C \geq 2^{120}$. Note however that even with $C \approx 2^{100}$, the square-root attack is not much practical, due to memory constraints.

In [11], Merkle and Werchner proposed an adaptation of generic algorithms (see [17]) to server-aided RSA protocols, and showed that any one-round passive generic attack on RSA-S1 had complexity at least $\Omega(\sqrt{C})$.

In [15], Pfitzmann and Waidner also presented a few active attacks which cannot be avoided by increasing the parameters contrary to the passive attacks mentioned previously. They discussed two countermeasures to prevent their own active attacks:

- Renewing the decomposition of the private exponent d at each execution of the protocol, as described in Steps 1 and 2.
- Verifying the signature $x^d \pmod N$ before releasing it, by computing $(x^d)^e \pmod N$ and checking that it is equal to x. This countermeasure is well-known and requires a very small public exponent e (otherwise there is no computational advantage in using the server to compute $x^d \pmod N$).

The second countermeasure seems necessary but is not sufficient to prevent one of the active attacks of [15], and it creates the attack of Section 4. The first countermeasure prevents all the active attacks of [15], but creates the passive attack of Merkle [10], which we analyze in Section 3. Interestingly, it seems that the attacks of Section 3 and 4 do not apply to the RSA-S2 protocol, which is a CRT variant of RSA-S1 (see [8,15]). The situation is reminiscent of that of RSA with small private exponent, in which the best attack known [3] fails if the private exponent is small modulo both $p-1$ and $q-1$.

2.3 Lattices

Our attacks are based on lattice basis reduction, a familiar tool in public-key cryptanalysis. We give a brief overview of lattice theory (see the survey [14] for a

list of references). In this paper, we call a *lattice* any subgroup of $(\mathbb{Z}^n, +)$: in the literature, these are called integer lattices. For any set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{Z}^n$, we define the set of all integral linear combinations:

$$L(\mathbf{b}_1, \ldots, \mathbf{b}_d) = \left\{ \sum_{i=1}^{d} n_i \mathbf{b}_i : n_i \in \mathbb{Z} \right\}.$$

By definition, $L(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is a lattice, called the lattice spanned by the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$. A *basis* of a lattice L is a set of linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$ such that:

$$L = L(\mathbf{b}_1, \ldots, \mathbf{b}_d).$$

In any lattice, there is always at least one basis, and in general, there are in fact infinitely many lattice bases. But all the bases of a lattice L have the same number of elements, called the *rank* or *dimension* of the lattice. All the bases also have the same d-dimensional volume, which is by definition the square root of the determinant $\det_{1 \le i,j \le d} \langle \mathbf{b}_i, \mathbf{b}_j \rangle$, where $\langle , \rangle$ denotes the Euclidean inner product. This volume $\mathrm{vol}(L)$ is called the volume or determinant of the lattice. When the lattice dimension d is equal to the space dimension n, this volume is simply the absolute value of the determinant of any lattice basis.

For a vector $\mathbf{a}$, we denote by $\|\mathbf{a}\|$ its Euclidean norm. A basic problem in lattice theory is the shortest vector problem (SVP): given a basis of a lattice L, find a non-zero vector $\mathbf{v} \in L$ such that $\|\mathbf{v}\|$ is minimal among all non-zero lattice vectors. Any such vector is called a shortest lattice vector. It is well-known that the Euclidean norm of a shortest lattice vector is always less than $\sqrt{d}\mathrm{vol}(L)^{1/d}$, d denoting the lattice dimension. In "usual" lattices, one does not expect the norm of a shortest lattice vector to be much less than this upper bound.

Many attacks in public-key cryptanalysis work by reduction to SVP, or to approximating SVP (see the survey [14]). The shortest vector problem was recently shown to be NP-hard under randomized reductions [1], and therefore, it is now widely believed that there is no polynomial-time algorithm to solve SVP. However, there exist polynomial-time algorithms which can provably approximate SVP. The first algorithm of that kind was the celebrated LLL lattice basis reduction algorithm of Lenstra, Lenstra and Lovász [7]. We use the best deterministic polynomial-time algorithm currently known to approximate SVP, which is due to Schnorr [16] and is based on LLL:

Lemma 1. *There exists a deterministic polynomial time algorithm which, given as input a basis of an s-dimensional lattice L, outputs a non-zero lattice vector $\mathbf{u} \in L$ such that:*

$$\|\mathbf{u}\| \le 2^{O\left(s \log^2 \log s / \log s\right)} \min \left\{ \|\mathbf{z}\| : \quad \mathbf{z} \in L, \mathbf{z} \ne 0 \right\}.$$

Recently, Ajtai *et al.* [2] discovered a randomized algorithm which slightly improves the approximation factor $2^{O\left(s \log^2 \log s / \log s\right)}$ to $2^{O(s \log \log s / \log s)}$. In practice, the best algorithm to approximate SVP is a heuristic variant of Schnorr's algorithm [16]. Interestingly, these algorithms typically perform much better

than theoretically expected: they often return a shortest lattice vector, provided that the lattice dimension is not too large. Hence, it is useful to predict what can be achieved efficiently if an SVP-oracle (that is, an algorithm which solves SVP) is available. For instance, this was done for the subset sum problem [5]. However, unless the lattice dimension is extremely small, it is hard to predict beforehand whether an SVP-instance is solvable in practice, which means that experiments are always necessary in this case.

3 An Analysis of Merkle's Multi-round Attack

3.1 Merkle's Attack

The attack of Merkle [10] is based on the following observation: Because for each $\mathbf{f} = (f_1, \ldots, f_m) \in \mathcal{B}_{k,\ell,m}$ and $\mathbf{d} = (d_1, \ldots, d_m) \in [\varphi(N)]^m$

$$0 < \sum_{i=1}^{m} f_i d_i < k2^{\ell}\varphi(N)$$

we have

$$\sum_{i=1}^{m} f_i d_i \equiv d + j\varphi(N)$$

with $j \in \left[k2^{\ell}\right]$, that is, j cannot take too many distinct values.

It is shown in [10] that regardless of the distribution of the vectors $\mathbf{f} \in \mathcal{B}_{k,\ell,m}$ with probability at least $1/k2^{\ell}$ for two pairs $\mathbf{f}_1 = (f_1, \ldots, f_m)$, $\mathbf{d}_1 = (d_1, \ldots, d_m)$, and $\mathbf{f}_2 = (f_{m+1}, \ldots, f_{2m})$, $\mathbf{d}_2 = (d_{m+1}, \ldots, d_{2m})$ of vectors produced by the above protocol we have the following equation (over the integers rather than modulo N):

$$\sum_{i=1}^{m} f_i d_i = \sum_{i=m+1}^{2m} f_i d_i. \tag{3}$$

In fact, any rule to select the above vectors gives rise to a collision after at most $k2^{\ell}$ executions of the protocol. Besides, the "birthday paradox" suggests that a collision is likely to happen after roughly $k^{1/2}2^{\ell/2}$ executions of the protocol.

The linear equation (3) is unusual because each f_i is small (compared to the d_i's), and this can be interpreted in terms of lattices. More precisely, it is argued in [10] that $(\mathbf{f}_1, \mathbf{f}_2)$ is the shortest vector in a particular lattice related to the homogeneous equation (3) and the congruences

$$\sum_{i=1}^{m} f_i d_i \equiv \sum_{i=m+1}^{2m} f_i d_i \equiv d \pmod{\varphi(N)}. \tag{4}$$

However, the analysis presented by Merkle is not sufficient, because it assumes a distribution of the parameters which is not the one of the protocol (see [10,

Theorem 2.1]). And no result is proposed without SVP-oracles. Hence, Merkle's attack, as presented in [10], is not a proved attack, even under the assumption of an SVP-oracle, which is not so unusual for a lattice-based attack. Nevertheless, the experiments conducted by Merkle (see [10]) showed that the attack was successful in practice against many choices of the parameters. Thus, it was interesting to see whether Merkle's attack could be proved, with or without SVP-oracles. Here, we provide a proof, for a slight variant of Merkle's attack. The analysis we present can in fact be extended to the original attack, but our variant is slightly simpler to describe and to analyze, while the difference of efficiency between the two attacks is marginal.

3.2 A Variant of Merkle's Attack

We work directly with the lattice corresponding to (3): Let $\mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)$ be the $(2m-1)$-dimensional lattice formed by all vectors $\mathbf{z} \in \mathbb{Z}^{2m}$ with

$$\sum_{i=1}^{m} z_i d_i = \sum_{i=m+1}^{2m} z_i d_i.$$

This lattice is the simplest case of an orthogonal lattice (as introduced in [12]), and one can compute a basis of such lattices in polynomial time. It can easily be showed that the volume of the lattice is given by:

$$\mathrm{vol}(\mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)) = \frac{\left(d_1^2 + \ldots + d_{2m}^2\right)^{1/2}}{\gcd(d_1, \ldots, d_{2m})}.$$

Thus, one would expect its shortest non-zero vector to have a norm around:

$$(2m-1)^{1/2} \mathrm{vol}\left(\mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)\right)^{1/(2m-1)} \approx (2m-1)^{1/2} \varphi(N)^{1/(2m-1)}.$$

On the other hand, the vector $\mathbf{f} = (f_1, \ldots, f_{2m})$ belongs to this lattice, and has a norm of at most $k^{1/2} 2^{\ell}$. Hence, if $k^{1/2} 2^{\ell}$ is much smaller than $(2m-1)^{1/2} \varphi(N)^{1/(2m-1)}$, we expect $\mathbf{f}$ to be the shortest vector of $\mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)$, and if it is smaller enough, then the gap between $\mathbf{f}$ and the other lattice vectors guarantees that the algorithm of Lemma 1 will find it. Once $\mathbf{f}$ is known, one can derive the value $\sum_{i=1}^{m} f_i d_i$, which is congruent to the RSA private exponent modulo $\varphi(N)$, and therefore enables to sign any message. And one may further recover the factorization of N in randomized polynomial time, from $e \sum_{i=1}^{m} f_i d_i - 1$ which is a non-zero multiple of $\varphi(N)$ (see for instance [9, Section 8.2.2]).

In [10], the original attack of Merkle worked with a slight variant of the lattice $\mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)$, to take advantage of the fact that $f_i \in \left[2^{\ell}\right]$ and not $f_i \in \left[2^{\ell}\right]_{\pm}$. Such a trick was used for the subset sum problem [5]. However, this trick is not as useful here, because the distributions are different. This means that the difference between our variant and the original attack is marginal.

3.3 Theoretical Results

The previous reasoning can in fact be made rigorous by a tight analysis, which gives rise to the following result:

Theorem 1. *There is a deterministic algorithm $\mathcal{A}$ which, given as input an RSA modulus N, together with a public exponent e, and a set $\mathcal{D}$ of $k2^\ell$ vectors $\mathbf{d} \in [\varphi(N)]^m$ corresponding to a certain set $\mathcal{F}$ of vectors $\mathbf{f} \in \mathcal{B}_{k,\ell,m}$ generated by $k2^\ell$ independent executions of* RSA-S1*, outputs a value $\mathcal{A}(\mathcal{D})$ in time polynomial in $k, 2^\ell, m, \log N$ such that:*

$$\Pr_{\mathcal{D}}\left[\mathcal{A}(\mathcal{D}) \equiv d \pmod{\varphi(N)}\right] \geq 1 - \frac{k^{m+2} 2^{2\ell(m+2)+O(m^2 \log^2 \log m/\log m)}}{\varphi(N)}$$

where the probability is taken over all random choices of $\mathcal{D}$ for the given $\mathcal{F}$.

Proof. Given a set $\mathcal{D}$ of $k2^\ell$ vectors $\mathbf{d}$ associated with the protocol RSA-S1, which corresponds to a certain set $\mathcal{F}$ of $k2^\ell$ (unknown) vectors $\mathbf{f} \in \mathcal{B}_{k,\ell,m}$, the algorithm $\mathcal{A}$ selects all possible pairs of such vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ and uses the algorithm of Lemma 1 to find a short vector $\mathbf{u}$ in the $(2m-1)$-dimensional lattice $\mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)$ formed by all vectors $\mathbf{z} \in \mathbb{Z}^{2m}$ such that

$$\sum_{i=1}^{m} z_i d_i = \sum_{i=m+1}^{2m} z_i d_i.$$

We know that there is at least one pair $(\mathbf{d}_1, \mathbf{d}_2)$ such that the equation (3) holds. Notice that for any $\mathbf{f} \in \mathcal{B}_{k,\ell,m}$, we have

$$\|f\|^2 = \sum_{i=1}^{m} f_i^2 < 2^\ell \sum_{i=1}^{m} f_i \leq k2^{2\ell}. \tag{5}$$

Thus, if we apply the algorithm of Lemma 1 to $\mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)$, we obtain a vector $\mathbf{u} = (u_1, \ldots, u_{2m})$ such that:

$$\begin{aligned}
\|\mathbf{u}\|^2 &\leq 2^{O(m \log^2 \log m/\log m)} \min\left\{\|\mathbf{z}\|^2, \quad \mathbf{z} \in \mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)\right\} \\
&\leq 2^{O(m \log^2 \log m/\log m)} \left(\|\mathbf{f}_1\|^2 + \|\mathbf{f}_2\|^2\right) \\
&\leq k2^{2\ell+O(m \log^2 \log m/\log m)}.
\end{aligned}$$

Therefore, there exists some integer $U = k^{1/2} 2^{\ell+O(m \log^2 \log m/\log m)}$ such that $|u_i| < U$ for $i = 1, \ldots, 2m$, that is, $\mathbf{u} \in [U]_{\pm}^{2m}$.

We write $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2)$ where $\mathbf{u}_1, \mathbf{u}_2 \in [U]_{\pm}^{m}$ and say that $\mathbf{u}$ is *similar* to the concatenation $(\mathbf{f}_1, \mathbf{f}_2)$ if either $\mathbf{u}_1$ is non-zero and parallel to $\mathbf{f}_1$, or $\mathbf{u}_2$ is non-zero and parallel to $\mathbf{f}_2$. Notice that if one knows a vector $\mathbf{u} \neq 0$ similar to $\mathbf{f}_1, \mathbf{f}_2$, one obtains at most 2^ℓ possible values for either $\mathbf{f}_1$ or $\mathbf{f}_2$. And if $\mathbf{f}_1$ or $\mathbf{f}_2$ is correct, then $\langle \mathbf{f}_1, \mathbf{d}_1 \rangle$ or $\langle \mathbf{f}_2, \mathbf{d}_2 \rangle$ is congruent to d modulo $\varphi(N)$, which can be checked by signing a message. Hence it is enough to show that with probability at least

$1 - k^{m+2}2^{2\ell(m+2)+O(m^2 \log^2 \log m/\log m)}\varphi(N)^{-1}$ the vector $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2)$ returned by the algorithm of Lemma 1 is similar to $(\mathbf{f}_1, \mathbf{f}_2)$.

First for $\mathbf{f}_1, \mathbf{f}_2 \in \mathcal{B}_{k,\ell,m}$ we estimate the size of the set $\mathcal{E}(\mathbf{f}_1, \mathbf{f}_2)$ of pairs of vectors $\mathbf{d}_1, \mathbf{d}_2 \in [\varphi(N)]^m$ such that for some $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2) \in [U]_{\pm}^{2m}$ which is not similar to $(\mathbf{f}_1, \mathbf{f}_2)$ we have the equation

$$\sum_{i=1}^{m} u_i d_i = \sum_{i=m+1}^{2m} u_i d_i. \tag{6}$$

Let us fix a nonzero vector $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2) \in [U]_{\pm}^{2m}$ and a vector $(\mathbf{f}_1, \mathbf{f}_2) \in \mathcal{B}_{k,\ell,m}$ which are not similar. Without loss of generality we may assume that $\mathbf{u}_2 \neq 0$ and is not parallel to $\mathbf{f}_2$ and that $f_{2m} \neq 0$. Then excluding d_{2m} from (6) using (3), we obtain an equation

$$\sum_{i=1}^{m} c_i d_i = \sum_{i=m+1}^{2m-1} c_i d_i \tag{7}$$

with $c_i = u_i - f_i u_{2m}/f_{2m}$, $i = 1, \ldots, 2m-1$. By our assumption, for at least one $i \geq m+1$, the coefficient $c_i \neq 0$. Without loss of generality we may assume that $c_{2m-1} \neq 0$. Then the first congruence in (4) gives us at most $2^{\ell}\varphi(N)^{m-1}$ possible values for $\mathbf{d}_1 = (d_1, \ldots, d_m)$. Indeed, assuming that $f_m \neq 0$ and selecting the integers $d_1, \ldots, d_{m-1} \in [\varphi(N)]$ arbitrarily, we obtain a congruence of the form $f_m d_m \equiv D \pmod{\varphi(N)}$ which has at most $\gcd(f_m, \varphi(N)) \leq f_m < 2^{\ell}$ solutions $d_m \in [\varphi(N)]$. Finally, for any of $\varphi(N)^{m-2}$ possible choices of $d_{m+1}, \ldots, d_{2m-2} \in [\varphi(N)]^{m-2}$ the equation (7) gives at most one value for d_{m-1} and then the second congruence in (4) gives us at most $\gcd(f_{2m}, \varphi(N)) \leq f_{2m} < 2^{\ell}$ possible values for d_{2m}. So the total number of solutions for such $\mathbf{u}$ is at most $2^{2\ell}\varphi(N)^{2m-3}$. The total number of such vectors is at most U^{2m}. Thus we finally derive

$$\begin{aligned} \#\mathcal{E}(\mathbf{f}_1, \mathbf{f}_2) &\leq (2U)^{2m} 2^{2\ell} \varphi(N)^{2m-3} \\ &\leq k^m 2^{2\ell(m+1)+O(m^2 \log^2 \log m/\log m)} \varphi(N)^{2m-3}. \end{aligned}$$

For each vector $\mathbf{f} \in \mathcal{B}_{k,\ell,m}$ there are exactly $\varphi(N)^{m-1}$ vectors $\mathbf{d} \in [\varphi(N)]^m$ satisfying the congruence (2). Therefore, the probability that there is a pair of vectors $\mathbf{f}_1, \mathbf{f}_2 \in \mathcal{F}$ such that the corresponding vectors $\mathbf{d}_1, \mathbf{d}_2 \in \mathcal{D}$ satisfy $\mathbf{d}_1, \mathbf{d}_2 \in \mathcal{E}(\mathbf{f}_1, \mathbf{f}_2)$ is at most

$$\begin{aligned} &\frac{(\#\mathcal{F})^2 k^m 2^{2\ell(m+1)+O(m^2 \log^2 \log m/\log m)} \varphi(N)^{2m-3}}{\varphi(N)^{2m-2}} \\ &\qquad = k^{m+2} 2^{2\ell(m+2)+O(m^2 \log^2 \log m/\log m)} \varphi(N)^{-1}, \end{aligned}$$

and the result follows. □

Assuming that an SVP-oracle is available, we derive much stronger estimates.

Theorem 2. *There is a deterministic algorithm $\mathcal{A}$ which, given an access to an SVP-oracle and as input an RSA modulus N, together with a public exponent e, a set $\mathcal{D}$ of $k2^\ell$ vectors $\mathbf{d} \in [\varphi(N)]^m$ corresponding to a certain set $\mathcal{F}$ of vectors $\mathbf{f} \in \mathcal{B}_{k,\ell,m}$ generated by $k2^\ell$ independent executions of* RSA-S1*, outputs a value $\mathcal{A}(\mathcal{D})$ in time polynomial in $k, 2^\ell, m, \log N$ such that:*

$$\Pr_{\mathcal{D}}[\mathcal{A}(\mathcal{D}) \equiv d \pmod{\varphi(N)}] \geq 1 - \frac{k^{m+2}2^{2(\ell m+2\ell+m)}}{\varphi(N)}$$

where the probability is taken over all random choices of $\mathcal{D}$ for the given $\mathcal{F}$.

As in [10], instead of waiting for $k2^\ell$ executions of RSA-S1 one may also restrict to only two executions, which yields the following version of Theorems 1 and 2:

Theorem 3. *There is a deterministic algorithm $\mathcal{A}$ which, given as input an RSA modulus N, together with a public exponent e, a pair of vectors $\mathbf{d}_1, \mathbf{d}_2 \in [\varphi(N)]^m$ corresponding to a pair of vectors $\mathbf{f}_1, \mathbf{f}_2 \in \mathcal{B}_{k,\ell,m}$ generated by two independent executions of* RSA-S1*, outputs a value $\mathcal{A}(\mathbf{d}_1, \mathbf{d}_2)$ in time polynomial in $k, 2^\ell, m, \log N$ such that:*

$$\Pr_{\mathbf{d}_1,\mathbf{d}_2}[\mathcal{A}(\mathbf{d}_1, \mathbf{d}_2) \equiv d \pmod{\varphi(N)}] \geq \frac{1}{k2^\ell} - \frac{k^m 2^{2\ell(m+1)+O(m^2 \log^2 \log m/\log m)}}{\varphi(N)}$$

where the probability is taken over all random choices of $\mathbf{d}_1, \mathbf{d}_2$ for the given $\mathbf{f}_1, \mathbf{f}_2$.

Theorem 4. *There is a deterministic algorithm $\mathcal{A}$ which, given access to an SVP-oracle and as input an RSA modulus N, together with a public exponent e, a pair of vectors $\mathbf{d}_1, \mathbf{d}_2 \in [\varphi(N)]^m$ corresponding to a pair of vectors $\mathbf{f}_1, \mathbf{f}_2 \in \mathcal{B}_{k,\ell,m}$ generated by two independent executions of* RSA-S1*, makes a single call to the SVP-oracle with the lattice $\mathcal{L}(\mathbf{d}_1, \mathbf{d}_2)$ and outputs a value $\mathcal{A}(\mathbf{d}_1, \mathbf{d}_2)$ in time polynomial in $k, 2^\ell, m, \log N$ such that:*

$$\Pr_{\mathbf{d}_1,\mathbf{d}_2}[\mathcal{A}(\mathbf{d}_1, \mathbf{d}_2) \equiv d \pmod{\varphi(N)}] \geq \frac{1}{k2^\ell} - \frac{k^m 2^{2(\ell m+\ell+m)}}{\varphi(N)}$$

where the probability is taken over all random choices of $\mathbf{d}_1, \mathbf{d}_2$ for the given $\mathbf{f}_1, \mathbf{f}_2$.

Notice that unless k (and thus $\ell \geq k/m$) is exponentially large compared to m, which is completely impractical, the terms k^{m+2} and k^m in the bounds of Theorems 1 and 3 respectively, can be included in the term $2^{O(m^2 \log^2 \log m/\log m)}$.

3.4 Experiments

In practice, the attack is as efficient as Merkle's original attack, due to the fact that strong lattice basis reduction algorithms behave like oracles for the shortest vector problem up to moderate dimension. In [10], Merkle reported the experimental results presented in Table 1. Notice however that none of the sets of parameters of Table 1 leads to an efficient protocol (for the card).

Table 1. Experiments with Merkle's attack

m	k	ℓ	Success (%)	Complexity of the sqrt attack
25	28	11	100	2^{62}
32	26	10	100	2^{62}
38	26	9	100	2^{63}
42	26	8	100	2^{63}
48	26	7	70	2^{63}
56	26	6	10	2^{63}

4 A New One-Round Attack on Low Exponent RSA-S1

4.1 Description of the Attack

We now assume that a very small public exponent e is used. We also assume that the secret primes p and q defining $N = pq$ have approximately the same length. Let $s = p + q = O(N^{1/2})$. We have $\varphi(N) = N - s + 1$. When the RSA-S1 protocol is performed once, we have:

$$\sum_{i=1}^{m} f_i d_i \equiv d \pmod{\varphi(N)},$$

and therefore,

$$\sum_{i=1}^{m} f_i e d_i \equiv 1 \pmod{\varphi(N)}.$$

From (5) we see that there exists $r \in \left[k2^{\ell}e\right]$ such that

$$\sum_{i=1}^{m} f_i e d_i = 1 + r\varphi(N) = 1 + r(N - s + 1).$$

Hence

$$\sum_{i=1}^{m} f_i e d_i = 1 + r - rs \pmod{N}, \tag{8}$$

where $|1 + r - rs| = O(k2^{\ell}eN^{1/2})$. We thus obtain a linear equation modulo N where the unknown coefficients f_i and $1 + r - rs$ are all relatively small. This suggests to define the $(m+1)$-dimensional lattice $\mathcal{L}_{e,N}(\mathbf{d})$ spanned by the rows of the following matrix:

$$\begin{pmatrix} N & 0 & 0 & \dots & 0 \\ ed_1 & eR & 0 & \dots & 0 \\ ed_2 & 0 & eR & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ ed_m & 0 & \dots & 0 & eR \end{pmatrix}$$

where $R = \lfloor N^{1/2} \rfloor$. Obviously, the volume of this lattice is $\mathrm{vol}(\mathcal{L}_{e,N}(\mathbf{d})) = e^m N R^{m/2}$. Therefore, one would expect its shortest vector to be of norm roughly $(m+1)^{1/2} e^{m/(m+1)} N^{(m+2)/(2m+2)}$. On the other hand, the lattice contains the target vector

$$\mathbf{t} = (1 + r - rs, f_1 eR, \ldots, f_m eR),$$

whose norm is $\|\mathbf{t}\| = O\left(k 2^\ell e N^{1/2}\right)$ because of (5). Hence, the target vector is likely to be the shortest vector in this lattice if $k e^{1/(m+1)} 2^\ell$ is much smaller than $m^{1/2} N^{1/(2m+2)}$. Note that this condition is satisfied for sufficiently large N and that it is very similar to the heuristic condition we obtained in Section 3.2, which suggests that the efficiency of the attacks of Section 4 and 3 should be comparable. In case the target vector is really much smaller than the other lattice vectors, then the algorithm of Lemma 1 finds it. Once the target vector is known, we can recover a private exponent equivalent to d thanks to $\sum_{i=1}^m f_i d_i$, which enables to sign any message, as in Merkle's attack. Again, one may further derive a not too large multiple of $\varphi(N)$, which yields the factorization of N in randomized polynomial time.

4.2 Theoretical Results

The previous attack can be proved, using the same counting arguments of the proof of Theorem 1:

Theorem 5. *There is a deterministic algorithm $\mathcal{A}$ which, given as input an RSA modulus $N = pq$ such that $p + q = O(N^{1/2})$, together with a public exponent e, and a vector $\mathbf{d} \in [\varphi(N)]^m$ corresponding to a certain vector $\mathbf{f} \in \mathcal{B}_{k,\ell,m}$ generated by* RSA-S1*, outputs a value $\mathcal{A}(\mathbf{d})$ in time polynomial in $k, 2^\ell, m, \log N$ such that:*

$$\Pr_{\mathbf{d}}\left[\mathcal{A}(\mathbf{d}) \equiv d \pmod{\varphi(N)}\right] \geq 1 - \frac{k^{m+1} e^{m+1} 2^{\ell(m+1)+O(m^2 \log^2 \log m/\log m)}}{N^{1/2}}$$

where the probability is taken over all random choices of $\mathbf{d}$ for the given $\mathbf{f}$.

Proof. The algorithm $\mathcal{A}$ starts by applying the algorithm of Lemma 1 to find a short vector $\mathbf{w} \neq 0$ in the $(m+1)$-dimensional lattice $\mathcal{L}_{e,N}(\mathbf{d})$. Since $\mathbf{t}$ is a lattice vector and because $p + q = O(N^{1/2})$, we have:

$$\|\mathbf{w}\| \leq 2^{O(m \log^2 \log m/\log m)} \|\mathbf{t}\| = k e 2^{\ell + O(m \log^2 \log m/\log m)} N^{1/2}.$$

By definition of the lattice, $\mathbf{w}$ is of the form:

$$\mathbf{w} = (u_0 N + \sum_{i=1}^m e d_i u_i, u_1 eR, \ldots, u_m eR),$$

where each u_i is an integer.

Therefore, there exists some integer $U = k e 2^{\ell + O(m \log^2 \log m/\log m)}$ such that $|u_i| < U$ for $i = 1, \ldots, 2m$. Thus $\mathbf{u} = (u_1, \ldots, u_m) \in [U]_\pm^m$. We may assume

that $\|\mathbf{w}\| < N$ otherwise the right hand side of the inequality of the theorem is negative, making the bound trivial. Then necessarily $\mathbf{u} \neq 0$. We also have

$$\sum_{i=1}^{m} ed_i u_i \equiv w_0 \pmod{N} \tag{9}$$

for some $w_0 \in [W]_{\pm}$ where $W = O\left(ke2^{\ell+O(m \log^2 \log m/ \log m)} N^{1/2}\right)$.

Clearly, we may assume that $2^{\ell} \leq \min\{p, q\}$ otherwise the result is trivial. Thus for any $i = 1, \ldots, m$ with $f_i \neq 0$ we have $\gcd(f_i, N) = 1$. As before we see that for each w_0 and for each $\mathbf{u} \in [U]_{\pm}^m$ not parallel to $\mathbf{f}$ there are at most $\varphi(N)^{m-2}$ vectors $\mathbf{d} \in [\varphi(N)]^m$ satisfying both (8) and (9). Therefore the total number of vectors $\mathbf{d} \in [\varphi(N)]^m$ which satisfy (8) and at least one congruence (9), for some $w_0 \in [W]_{\pm}$ and some nonzero vector $\mathbf{u} \in [U]_{\pm}^m$ not parallel to $\mathbf{f}$, is at most

$$2^{m+1} W U^m \varphi(N)^{m-2} = k^{m+1} e^{m+1} 2^{\ell(m+1)+O(m^2 \log^2 \log m/ \log m)} N^{1/2} \varphi(N)^{m-2}.$$

Taking into account that $\varphi(N) \geq N/2$ we obtain the desired result. □

Of course, the same proof provides a stronger result if an SVP-oracle is available:

Theorem 6. *There is a deterministic algorithm $\mathcal{A}$ which, given access to an SVP-oracle and as input an RSA modulus $N = pq$ such that $p + q = O(N^{1/2})$, together with a public exponent e, vector $\mathbf{d} \in [\varphi(N)]^m$ corresponding to a certain vector $\mathbf{f} \in \mathcal{B}_{k,\ell,m}$ generated by* RSA-S1*, makes a single call to the SVP-oracle with the lattice $\mathcal{L}_{e,N}(\mathbf{d})$ and outputs a value $\mathcal{A}(\mathbf{d})$ in time polynomial in $k, 2^{\ell}, m, \log N$ such that:*

$$\Pr_{\mathbf{d}}[\mathcal{A}(\mathbf{d}) \equiv d \pmod{\varphi(N)}] \geq 1 - \frac{k^{m+1} e^{m+1} 2^{\ell(m+1)+O(m)}}{N^{1/2}}$$

where the probability is taken over all random choices of $\mathbf{d}$ for the given $\mathbf{f}$.

Certainly one can obtain similar results when the primes p and q are not balanced, although the probability of success decreases.

4.3 Experiments

We made a few experiments with a (balanced) 1024-bit RSA modulus and a public exponent $e = 3$, using Victor Shoup's NTL library [19]. The experiments have confirmed the heuristic condition. By applying standard floating point LLL reduction, and improved reduction if necessary, we have been able to recover the private exponent for all the parameters considered by Merkle in his own experiments [10] (see Table 1). The success rate has been 100%, except with the case $(m, k, \ell) = (56, 26, 6)$ where it is 65% (for this case, Merkle only achieved a 10% success rate). We also made some experiments on other (more realistic) sets of parameters. For instance, over 100 samples, we have always been able to recover the factorization with $(m, k, \ell) = (60, 30, 3)$, $(70, 30, 2)$ and $(80, 40, 1)$. The attack takes at most a couple of minutes, as the lattice dimension is only $m+1$. These results show that no set of parameters for RSA-S1 provides sufficient security without being impractical for the card.

References

1. M. Ajtai, 'The shortest vector problem in L_2 is NP-hard for randomized reductions', *Proc. 30th ACM Symp. on Theory of Comput.*, ACM, 1998, 10-19.
2. M. Ajtai, R. Kumar and D. Sivakumar, 'A sieve algorithm for the shortest lattice vector problem' *Proc. 33rd ACM Symp. on Theory of Comput.*, ACM, 2001, 601–610.
3. D. Boneh and G. Durfee, 'Cryptanalysis of RSA with private key d less than $N^{0.292}$', *Proc. of Eurocrypt '99*, Lect. Notes in Comp. Sci., Vol. 1592, Springer-Verlag, Berlin, 1999, 1–11.
4. E. Brickell, D.M. Gordon, K.S. McCurley, and D. Wilson, 'Fast exponentiation with precomputation', *Proc. Eurocrypt '92*, Lect. Notes in Comp. Sci., Vol. 658, Springer-Verlag, Berlin, 1993, 200–207.
5. M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.-P. Schnorr, and J. Stern, 'Improved low-density subset sum algorithms', *Comput. Complexity*, **2** (1992), 111–128.
6. D. M. Gordon, 'A survey of fast exponentiation methods', *J. of Algorithms*, **27** (1998), 129–146.
7. A. K. Lenstra, H. W. Lenstra and L. Lovász, 'Factoring polynomials with rational coefficients', *Mathematische Annalen*, **261** (1982), 515–534.
8. T. Matsumoto, K. Kato, and H. Imai, 'Speeding up secret computations with insecure auxiliary devices', *Proc. Crypto '88*, Lect. Notes in Comp. Sci., Vol. 403, Springer-Verlag, Berlin, 1990, 497–506.
9. A. J. Menezes, P. C. van Oorschott and S. A. Vanstone, *Handbook of applied cryptography*, CRC Press, Boca Raton, FL, 1996.
10. J. Merkle, 'Multi-round passive attacks on server-aided RSA protocols', *Proc. 7th ACM Conf. on Computer and Commun. Security*, ACM, 2000, 102–107.
11. J. Merkle and R. Werchner, 'On the security of server-aided RSA protocols', *Proc. PKC '98*, Lect. Notes in Comp. Sci., Vol.1431, Springer-Verlag, Berlin, 1998, 99–116.
12. P. Q. Nguyen and J. Stern, 'Merkle-Hellman revisited: A cryptanalysis of the Qu–Vanstone cryptosystem based on group factorizations', *Proc. Crypto '97*, Lect. Notes in Comp. Sci., Vol.1294, Springer-Verlag, Berlin, 1997, 198–212.
13. P. Q. Nguyen and J. Stern, 'The Béguin–Quisquater server-aided RSA protocol from Crypto'95 is not secure', *Proc. Asiacrypt '98*, Lect. Notes in Comp. Sci., Vol.1514, Springer-Verlag, Berlin, 1998, 372–379.
14. P. Q. Nguyen and J. Stern, 'The two faces of lattices in cryptology', *Proc. CALC '01*, Lect. Notes in Comp. Sci., Vol.2146, Springer-Verlag, Berlin, 2001, 146–180.
15. B. Pfitzmann and M. Waidner, 'Attacks on protocols for server-aided RSA computation', *Proc. Eurocrypt '92*, Lect. Notes in Comp. Sci., Vol.658, Springer-Verlag, Berlin, 1993, 153–162.
16. C. P. Schnorr, 'A hierarchy of polynomial time basis reduction algorithms', *Theor. Comp. Sci.*, **53** (1987), 201–224.
17. V. Shoup, 'Lower bounds for discrete logarithms and related problems', *Proc. Eurocrypt '97*, Lect. Notes in Comp. Sci., Vol.1233, Springer-Verlag, Berlin, 1997, 256–266.
18. D. Stinson, 'Some baby-step giant-step algorithms for the low Hamming weight discrete logarithm problem', To appear in *Mathematics of Computation.*
19. V. Shoup, 'NTL computer package version 5.0', *Available from* `http://www.shoup.net/`.