On the Improvement of the BDF Attack on LSBS-RSA

Hung-Min Sun[1], Mu-En Wu[1,2], Huaxiong Wang[2,3], and Jian Guo[2]

[1] Department of Computer Science,
National Tsing Hua University, Taiwan
hmsun@cs.nthu.edu.tw, mn@is.cs.nthu.edu.tw
[2] School of Physical & Mathematical Sciences,
Nanyang Technological University, Singapore
{hxwang,guojian}@ntu.edu.sg
[3] Centre for Advanced Computing - Algorithms and Cryptography
Department of Computing
Macquarie University, Australia

Abstract. An (α, β, γ)-LSBS RSA denotes an RSA system with primes sharing α least significant bits, private exponent d with β least significant bits leaked, and public exponent e with bit-length γ. Steinfeld and Zheng showed that LSBS-RSA with small e is inherently resistant to the BDF attack, but LSBS-RSA with large e is more vulnerable than standard RSA. In this paper, we improve the BDF attack on LSBS-RSA by reducing the cost of exhaustive search for k, where k is the parameter in RSA equation: $ed = k \cdot \varphi(N) + 1$. Consequently, the complexity of the BDF attacks on LSBS-RSA can be further reduced. Denote σ as the multiplicity of 2 in k. Our method gives the improvements, which depend on the two cases:

1. In the case $\gamma \leq \min\{\beta, 2\alpha\} - \sigma$, the cost of exhaustive search for k in LSBS-RSA can be simplified to searching k in polynomial time. Thus, the complexity of the BDF attack is independent of γ, but it still increases as α increases.
2. In the case $\gamma > \min\{\beta, 2\alpha\} - \sigma$, the complexity of the BDF attack on LSBS-RSA can be further reduced with increasing α or β.

More precisely, we show that an LSBS-RSA is more vulnerable under the BDF attack as $\max\{2\alpha, \beta\}$ increases proportionally with the size of N. In the last, we point out that although LSBS-RSA benefits the computational efficiency in some applications, one should be more careful in using LSBS-RSA.

Keywords: RSA, partial key exposure (PKE), the BDF attack, least significant bit (LSB), LSBS-RSA, exhaustive search.

1 Introduction

RSA [12] is the most widely used public key cryptosystem in the world. It is not only built into several operating systems, such as Microsoft, Apple, Sun, and

Novell, but is also used for securing web traffic, e-mail, smart cards and IC cards. Since the encryption and decryption in RSA require taking heavy exponential multiplications modulus of N, the efficiency problem is the main disadvantage of using RSA. In order to overcome these drawbacks, many researchers have studied variants of RSA which reduce the computational costs [10], [11]. In general, the RSA encryption and decryption time are roughly proportional to the number of bits in public and secret exponents, respectively. To reduce the encryption time (or the signature-verification time), one may wish to use a small public exponent e. The smallest possible value for e is 3, however, it has been proven to be insecure against some small public exponent attacks [9]. Therefore, a more widely accepted and used public exponent is $e = 2^{16} + 1 = 65537$ or larger but far smaller than $\varphi(N)$.

In 1998, Boneh, Durfee, and Frankel [1], [2] first proposed the partial key exposure (PKE) attacks on RSA. They showed that for low public exponent RSA, given a fraction of the bits of the private exponent, an adversary can recover the entire private key and thus break the RSA. We call their methods the BDF attacks throughout this paper. More results of the partial key exposure attacks on RSA were proposed in 2003, and 2005 by Blömer & May [3], and Ernst, Jochemsz, May, & Weger [8], respectively.

In this paper, we improve the BDF attack on LSBS-RSA. An LSBS-RSA denotes an RSA system with modulus primes sharing a number of least significant bits (LSBs), *i.e.* $p-q = r \cdot 2^{\alpha}$ for some odd integer r, and $\alpha > 1$, where r, $\alpha \in \mathbb{N}$. This concept was first proposed by Steinfeld and Zheng [13] to improve the efficiency of a server aided RSA signature generation (SASG) [4]. In [13] and [14], Steinfeld and Zheng analyze the complexity of the BDF attack on LSBS-RSA. Their results show that low public exponent LSBS-RSA is inherently resistant to the partial key exposure attacks. That means, the BDF attacks will be less effective for LSBS-RSA with small e than for standard RSA. However, this is not true for large public exponent LSBS-RSA. LSBS-RSA with large e is more vulnerable under such attacks than standard RSA. In this paper, we give the detailed analysis to further support Steinfeld and Zheng's argument. We improve the BDF attack by reducing the cost of exhaustive search for k in LSBS-RSA, where k is the parameter in RSA equation: $ed = k \cdot \varphi(N) + 1$.

Denote σ as the multiplicity of 2 in k. Our improvements depend on the two cases: $\gamma \leq \min\{\beta, 2\alpha\} - \sigma$ and $\gamma > \min\{\beta, 2\alpha\} - \sigma$:

In the case $\gamma \leq \min\{\beta, 2\alpha\} - \sigma$, the cost of exhaustive search for k in LSBS-RSA can be simplified to searching k in polynomial time. Thus, the complexity of the BDF attack in this case can be further reduced. On the other hand, in the case $\gamma > \min\{\beta, 2\alpha\} - \sigma$, the complexity of searching k in LSBS-RSA still can be improved instead of finding k by exhaustive search totally. Thus, the BDF attack on LSBS-RSA in this case is improved as well. Furthermore, we show that an LSBS-RSA is more vulnerable under the BDF attack as $\max\{2\alpha, \beta\}$ increases proportionally with the size of N.

The remainder of this paper is organized as follows. In Section 2, we briefly review theorems and lemmas related to the BDF attack. In Section 3, we revise

the BDF attack on LSBS-RSA and show the complexity analysis in Section 4. In Section 5, further discussions about the feasibility and the efficiency are proposed. Finally, we conclude this paper and give some open problems in Section 6.

2 Preliminary

2.1 RSA, LSBS-RSA and Some Notations

In standard RSA, let N $(= p \times q)$ be the product of two large primes p and q. The public exponent e and the private exponent d satisfy $e \times d \equiv 1 \pmod{\varphi(N)}$, where $\varphi(N) = (p-1) \times (q-1)$ is the Euler totient function of N. Here, N is called the RSA modulus. The public key is the pair (N, e) that is used for encryption (or signature-verification): $c \equiv m^e \pmod{N}$, where m is the message and c is the corresponding ciphertext. The private key is the pair (N, d) that enables the decryption of ciphertext (or signature-generation): $m \equiv c^d \pmod{N}$. In the key generation of RSA, we usually select two primes (about 512 bits) p and q, and then multiply them to obtain N (about 1024 bits). Next, we pick the public exponent e first, and then compute the private exponent d by $d \equiv e^{-1} \pmod{\varphi(N)}$ by Euclidean algorithm. With high probability, no matter what size of e is chosen, the size of d is as large as the size of $\varphi(N)$ almost.

Throughout this paper, we follow the notation (α, β, γ), which is also used by Steinfeld and Zheng. An (α, β, γ)-LSBS RSA is an RSA system with the following properties:

α:	α-LSBS RSA modulus: $N = pq$, where $\lvert p - q \rvert = r \cdot 2^{\alpha}$ for some odd integer r.
β:	The β least significant bits of the private exponent d are available.
γ:	The public exponent e with bit-length γ.

In addition, we use the symbols κ and λ to denote the multiplicity of 2 in k and $\varphi(N)$, respectively. Moreover, given an integer x of m bits, whose binary representation is

$$(x)_2 = (x_m, x_{m-1}, ...x_j, ..., x_i, ..., x_2, x_1)_2,$$

where $x_i = 0$ or 1 for $i = 1, ..., m$. We call x_m the most significant bit of x and x_1 the least significant bit of x. Denote "$\mathrm{LSB}_{i\text{-}j}(x)$" as the i-th to j-th least significant bits of $(x)_2$, where $i < j$. That is,

$$\mathrm{LSB}_{i\text{-}j}(x) = (x_j, ..., x_i)_2.$$

and "$\mathrm{LSB}_i(x)$" as the i-th least significant bit of $(x)_2$. That is,

$$\mathrm{LSB}_i(x) = x_i.$$

2.2 The BDF Attack on LSBS-RSA

Here we briefly introduce the BDF attack on LSBS-RSA. All the following theorems and lemmas can be found in **[13]** and **[14].** The goal is to use the information of partial key to find $\mathrm{LSB}_{1\text{-}\frac{n}{4}}(p)$ or $\mathrm{LSB}_{1\text{-}\frac{n}{4}}(q)$. Then, use Coppersmith's method (see Theorem 1) to factor N.

Theorem 1. *(Coppersmith's method [5]) Let $N = pq$ be an n-bit RSA modulus. If $LSB_{1\text{-}\frac{n}{4}}(p)$ or $LSB_{1\text{-}\frac{n}{4}}(q)$ is given, then there exists an algorithm to factor N in polynomial time in n.*

We denote $T_{Cop}(n)$ as the complexity of the algorithm in Theorem 1. The improved versions of Coppersmith's method can be found in [6] and [7].

Lemma 1. *Consider the modular equation $x^2 \equiv c \pmod{2^r}$ and let $m_2(c)$ denote the multiplicity of 2 in c. That is, $c = c_{odd} \cdot 2^{m_2(c)}$, where c_{odd} is the largest odd factor of c. Then, the solutions are summarized in the following table:*

Conditions	*Solution #*	*Solution Forms*
If $r \leq m_2(c)$	$2^{\lfloor r/2 \rfloor}$	$x \equiv 0 \pmod{2^{\lceil r/2 \rceil}}$
If $r > m_2(c)$ *and* $m_2(c)$ *is odd*	0	-
If $r > m_2(c)$ *and* $m_2(c)$ *is even, there are three subcases:*		
subcase 1: $r = m_2(c) + 1$	$2^{\frac{m}{2}}$	$x \equiv 2^{\frac{m}{2}} \pmod{2^{\frac{m}{2}+1}}$
subcase 2: $r = m_2(c) + 2$, $c_{odd} \equiv 1 \pmod 4$	$2 \cdot 2^{\frac{m}{2}}$	$x \equiv \pm 2^{\frac{m}{2}} \pmod{2^{\frac{m}{2}+2}}$
subcase 3: $r \geq m_2(c) + 3$, $c_{odd} \equiv 1 \pmod 8$	$4 \cdot 2^{\frac{m}{2}}$	$x \equiv \pm s \cdot 2^{\frac{m}{2}} \pmod{2^{r-\frac{m}{2}}}$, *or* $x \equiv \left(\pm s + 2^{r-m-1}\right) \cdot 2^{\frac{m}{2}} \pmod{2^{r-\frac{m}{2}}}$, *where* $s^2 \equiv c_{odd} \pmod{2^{r-m}}$.
Otherwise	0	-

Proof. The proof can be found in Lemma 1 of [13], or [14].

Note that if there exist solutions for $x^2 \equiv c \pmod{2^r}$, then c and r must satisfy one of the conditions in the above table. Next, we show the properties of an α-LSBS RSA.

Lemma 2. *Let $N = pq$ denote an n-bit α-LSBS-RSA modulus. There exists an algorithm to compute the $LSB_{1\text{-}2\alpha}(p+q)$, $LSB_{1\text{-}\alpha}(p)$, and $LSB_{1\text{-}\alpha}(q)$ in polynomial time $O\left(n^2\right)$.*

Proof. Let $p = p_H \cdot 2^\alpha + l$ and $q = q_H \cdot 2^\alpha + l$. Thus, l is a solution to the modular quadratic congruence $x^2 \equiv N \pmod{2^\alpha}$, and it can be computed at most for 4 candidates in time polynomial in n^2. From

$$p \cdot q = N, \tag{1}$$

we may replace p and q by $p_H \cdot 2^\alpha + l$ and $q_H \cdot 2^\alpha + l$, respectively. This conducts to

$$\mathrm{LSB}_{1\text{-}2\alpha}\left(l \cdot (p_H + q_H) \cdot 2^\alpha + l^2\right) = \mathrm{LSB}_{1\text{-}2\alpha}(N). \tag{2}$$

Since l is an odd integer, $l^{-1} \pmod{2^{2\alpha}}$ exists. We have

$$\mathrm{LSB}_{1\text{-}2\alpha}((p_H + q_H) \cdot 2^{\alpha}) = \mathrm{LSB}_{1\text{-}2\alpha}\left(l^{-1} \cdot (N - l^2)\right). \tag{3}$$

The identity (3) shows that $\mathrm{LSB}_{1\text{-}\alpha}(p_H + q_H)$ can be totally computed from $l^{-1} \cdot (N - l^2)$. Thus, we have

$$\begin{aligned} &\mathrm{LSB}_{1\text{-}2\alpha-1}\left(\tfrac{p+q}{2}\right) \\ &= \mathrm{LSB}_{1\text{-}2\alpha-1}((p_H + q_H) \cdot 2^{\alpha-1} + l) \\ &= \mathrm{LSB}_{1\text{-}\alpha}(p_H + q_H) \,||\, \mathrm{LSB}_{\alpha}\left((p_H + q_H) \cdot 2^{\alpha-1} + l\right) \,||\, \mathrm{LSB}_{1\text{-}\alpha-1}(l). \end{aligned} \tag{4}$$

where "||" denotes the concatenation. Therefore, we get

$$\mathrm{LSB}_{1\text{-}2\alpha}(p + q) = \mathrm{LSB}_{1\text{-}2\alpha-1}\left(\tfrac{p+q}{2}\right) \,||\, 0,$$

which completes the proof.

In the following we show the result of the BDF attack on LSBS-RSA, which is called the generalized BDF attack.

Theorem 2. ***(Generalized BDF Attack, [13], [14])*** *Let $N = pq$ denote an n-bit α-LSBS RSA modulus, d is a private exponent, and e is a public exponent with bit-length γ. Given $d_0 = LSB_{1\text{-}\beta}(d)$, the Generalized BDF attack factors N within the following time complexity:*

$$\begin{aligned} &\textit{If } \beta < 2(\alpha - 1) + \gamma, \textit{ then } T_{BDF}(n) = O\left(\gamma 2^{\gamma} \cdot \lceil 2^{\frac{n}{4} - \frac{\beta}{2}} \rceil \cdot T'_{Cop}(n)\right); \\ &\textit{If } \beta \geq 2(\alpha - 1) + \gamma, \textit{ then } T_{BDF}(n) = O\left(\gamma 2^{\gamma} \cdot \lceil 2^{\frac{n}{4} + \alpha - \beta} \rceil \cdot T'_{Cop}(n)\right), \end{aligned} \tag{5}$$

where $T'_{Cop}(n) = T_{Cop}(n) + O\left(n^2\right)$, which is the complexity of Coppersmith's method plus the $O\left(n^2\right)$ for the other computations.

Note that in the case $\beta \geq 2(\alpha - 1) + \gamma$, $T_{BDF}(n)$ increases as α increases, which shows Steinfeld and Zheng's argument: low public exponent LSBS-RSA is inherently resistant to the BDF attack. In addition, as can be seen in (5), $T_{BDF}(n)$ decreases as β increases. We divide the process of the BDF attack on LSBS-RSA into three parts:

1. Exhaustive search the parameter k in RSA equation: $ed = k \cdot \varphi(N) + 1$, where $1 < k < e$.
2. With the information of k, compute $\mathrm{LSB}_{1\text{-}\frac{n}{4}}(p)$ by solving the quadratic modular equation.
3. Once $\mathrm{LSB}_{1\text{-}\frac{n}{4}}(p)$ is known, use Coppersmith's method to factor N

Since $k < e$ and $2^{\gamma-1} \leq e < 2^{\gamma}$, the step 1 requires the time complexity $O(\gamma 2^{\gamma})$ to exhaustive search for k. The step 2 requires the time complexity $O\left(2^{\frac{n}{4} - \frac{\beta}{2}}\right)$ or $O\left(2^{\frac{n}{4} + \alpha - \beta}\right)$, depends on the relations between α, β, and γ. The step 3 requires the cost $T'_{Cop}(n)$, which is the complexity of Coppersmith's method plus $O(n^2)$.

3 The Revised BDF Attack on LSBS-RSA

In this section we show the revised BDF attack on LSBS-RSA. The main improvement is to reduce the cost of searching k in LSBS-RSA. Thus, the complexity in the step 1, which is $O(\gamma 2^{\gamma})$, can be further reduced. Before that, we show the process of recovering $\mathrm{LSB}_{1\sim\frac{n}{4}}(p)$ by solving the quadratic modular equation, and then use Coppersmith's method to factor N.

3.1 The Process of the BDF Attack

From the RSA equation we have

$$ed - 1 - k\left(N + 1 - p - \tfrac{N}{p}\right) = 0.$$

Multiplying p modulo 2^{β} yields the following modular equation with root p:

$$kx^2 + (ed_0 - k(N+1) - 1)\,x + kN \equiv 0 \pmod{2^{\beta}}, \tag{6}$$

where $d_0 = \mathrm{LSB}_{1\sim\beta}(d)$ is known to the attacker.

Suppose $k = k_{odd} \cdot 2^{\kappa}$, where k_{odd} is the largest odd factor of k, and κ denotes the multiplicity of 2 in k. Eliminating the leading coefficient of (6) yields

$$x^2 + \left(k_{odd}^{-1} \cdot \tfrac{ed_0 - 1}{2^{\kappa}} - (N+1)\right)x + N \equiv 0 \pmod{2^{\beta-\kappa}}, \tag{7}$$

where k_{odd}^{-1} denotes the inverse of k_{odd} in $\mathbb{Z}^*_{2^{\beta-\kappa}}$. Consequently, (7) is reduced to

$$\left(x + \tfrac{b(k)}{2}\right)^2 \equiv c(k) \pmod{2^{\beta-\kappa}}, \tag{8}$$

where

$$b(k) = k_{odd}^{-1} \cdot \tfrac{ed_0 - 1}{2^{\kappa}} - (N+1), \text{ and}$$

$$c(k) = \left(\tfrac{b(k)}{2}\right)^2 - N.$$

Now, we solve the modular equation (8) by applying Lemma 1.
Since

$$\begin{aligned} b(k) &= k_{odd}^{-1} \cdot \tfrac{ed_0 - 1}{2^{\kappa}} - (N+1) \\ &\equiv k_{odd}^{-1} \cdot \tfrac{k((N+1)-(p+q))}{2^{\kappa}} - (N+1) \pmod{2^{\beta-\kappa}} \\ &\equiv -(p+q) \pmod{2^{\beta-\kappa}}, \end{aligned}$$

we get

$$\begin{aligned} c(k) &= \left(\tfrac{b(k)}{2}\right)^2 - N \pmod{2^{\beta-\kappa}} \\ &\equiv \left(\tfrac{p+q}{2}\right)^2 - N \pmod{2^{\beta-\kappa}} \\ &\equiv \left(\tfrac{p-q}{2}\right)^2 \pmod{2^{\beta-\kappa}}. \end{aligned}$$

Moreover, since N is an α-LSBS RSA modulus, we may write $p-q = r \cdot 2^{\alpha}$ for some odd integer r, which shows the multiplicity of 2 in $\left(\frac{p-q}{2}\right)^2$ is $2(\alpha-1)$. Consequently, according to Lemma 1, the number of the solutions of (8) depends on the two cases: $\beta-\kappa \leq 2(\alpha-1)$ and $\beta-\kappa > 2(\alpha-1)$.

In the case $\beta-\kappa \leq 2(\alpha-1)$, there are $2^{\lfloor \frac{\beta-\kappa}{2} \rfloor}$ solutions of the form

$$x + \tfrac{b(k)}{2} \equiv 0 \pmod{2^{\lceil \frac{\beta-\kappa}{2} \rceil}}$$

for the modular equation (8). Thus, the $\lceil \frac{\beta-\kappa}{2} \rceil$ least significant bits of the root, *i.e.*, p, are known to the attacker, which is the same as $\mathrm{LSB}_{1^{\frown}\lceil \frac{\beta-\kappa}{2} \rceil}\left(-\frac{b(k)}{2}\right)$. Since $\mathrm{LSB}_{1^{\frown}\frac{n}{4}}(p)$ (or $\mathrm{LSB}_{1^{\frown}\frac{n}{4}}(q)$) is the minimum requirement to apply Coppersmith's method, the remaining unknown part of p is $\mathrm{LSB}_{(\lceil \frac{\beta-\kappa}{2} \rceil+1)^{\frown}\frac{n}{4}}(p)$. Therefore, in this case the search for the parameter k with the cost $2^{\frac{n}{4}-\lceil \frac{\beta-\kappa}{2} \rceil}$ is required. We simplify the cost to $O\left(2^{\frac{\kappa}{2}} \cdot 2^{\frac{n}{4}-\frac{\beta}{2}}\right)$.

In the case $\beta-\kappa > 2(\alpha-1)$, three subcases are discussed below according to Lemma 1:

subcase 1: If $\beta-\kappa = 2(\alpha-1)+1$, there are $2^{\alpha-1}$ solutions of the form

$$x + \tfrac{b(k)}{2} \equiv 2^{\alpha-1} \pmod{2^{\alpha}}.$$

subcase 2: If $\beta-\kappa = 2(\alpha-1)+2$, and $(\alpha-1)_{odd} \equiv 1 \pmod 4$, there are $2 \cdot 2^{\alpha-1}$ solutions of the form

$$x + \tfrac{b(k)}{2} \equiv \pm 2^{\alpha-1} \pmod{2^{\alpha+1}}.$$

subcase 3: If $\beta-\kappa \geq 2(\alpha-1)+3$, and $(\alpha-1)_{odd} \equiv 1 \pmod 8$, there are $4 \cdot 2^{\alpha-1}$ solutions of the form

$$x + \tfrac{b(k)}{2} \equiv (\pm s) \cdot 2^{\alpha-1} \pmod{2^{(\beta-\kappa)-(\alpha-1)}}, \text{ or}$$

$$x + \tfrac{b(k)}{2} \equiv \left(\pm s + 2^{(\beta-\kappa)-2(\alpha-1)-1}\right) \cdot 2^{\alpha-1} \pmod{2^{(\beta-\kappa)-(\alpha-1)}}.$$

Note that s is any solution to $s^2 \equiv (\alpha-1)_{odd} \pmod{2^{(\beta-\kappa)-2(\alpha-1)}}$, where $(\alpha-1)_{odd}$ is the largest odd factor of $\alpha-1$.

In the subcase 1, $\mathrm{LSB}_{1^{\frown}\alpha}(p)$ is known to the attacker. In order to apply Coppersmith's method, the remaining unknown part of p is $\mathrm{LSB}_{(\alpha+1)^{\frown}\frac{n}{4}}(p)$. Thus, in this case it requires the search with cost $2^{\frac{n}{4}-\alpha}$.

In the subcase 2, $\mathrm{LSB}_{1^{\frown}\alpha+1}(p)$ is known to the attacker. In order to apply Coppersmith's method, the remaining unknown part of p is $\mathrm{LSB}_{(\alpha+2)^{\frown}\frac{n}{4}}(p)$. We simplify the cost to $O\left(2^{\frac{n}{4}-\alpha}\right)$.

In the subcase 3, $\mathrm{LSB}_{1^{\frown}(\beta-\kappa)-(\alpha-1)}(p)$ is known to the attacker. In order to apply Coppersmith's method, the remaining unknown part of p is $\mathrm{LSB}_{(\beta-\kappa-\alpha)^{\frown}\frac{n}{4}}(p)$. Thus, in this case it requires the search with cost $2^{\frac{n}{4}-((\beta-\kappa)-(\alpha-1))}$. We simplify the cost to $O\left(2^{\kappa} \cdot 2^{(\frac{n}{4}-\beta)+\alpha}\right)$. As a result, the complexity of the BDF attack on

(α, β, γ)-LSBS RSA is concluded as follows.

If $\beta \leq 2(\alpha - 1) + \kappa$, then

$$T_{BDF}(n) = O\left(|K_c| \cdot (2^{\frac{\kappa}{2}} \cdot 2^{\frac{n}{4} - \frac{\beta}{2}}) \cdot T'_{Cop}(n)\right); \tag{9}$$

If $\beta = 2(\alpha - 1) + \kappa + 1$, or $\beta = 2(\alpha - 1) + \kappa + 2$, then

$$T_{BDF}(n) = O\left(|K_c| \cdot (2^{\frac{n}{4} - \alpha}) \cdot T'_{Cop}(n)\right); \tag{10}$$

If $\beta \geq 2(\alpha - 1) + \kappa + 3$, then

$$T_{BDF}(n) = O\left(|K_c| \cdot (2^{\kappa} \cdot 2^{(\frac{n}{4} - \beta) + \alpha}) \cdot T'_{Cop}(n)\right), \tag{11}$$

where $|K_c|$ denotes the number of candidates of k, which is required to test by exhaustive search. Next, we show how to reduce the size of K_c in LSBS-RSA.

3.2 Searching k in LSBS-RSA

We consider the following lemma:

Lemma 3. *Consider the three positive integers A, B, and C, where $C = A \times B$. If $LSB_{1^{\sim}m}(A)$ and $LSB_{1^{\sim}m}(C)$ are given, we can compute $LSB_{1^{\sim}m-m_2(A)}(B)$ in polynomial time in m, where $m_2(A)$ denotes the multiplicity of 2 in A.*

Proof. Suppose that $A = A_1 \cdot 2^m + A_2$ and $B = B_1 \cdot 2^m + B_2$, where $A_2 = \mathrm{LSB}_{1^{\sim}m}(A)$ and $B_2 = \mathrm{LSB}_{1^{\sim}m}(B)$, respectively. We may write $A_2 = A \pmod{2^m}$ and $B_2 = B \pmod{2^m}$. Since

$$A \times B = (A_1 B_1) \cdot 2^{2m} + (A_1 B_2 + A_2 B_1) \cdot 2^m + A_2 B_2 = C,$$

we have

$$C \pmod{2^m} \equiv A_2 B_2 \pmod{2^m}. \tag{12}$$

Denote $A_2 = a_2 \cdot 2^{m_2(A_2)}$, where $m_2(A_2)$ denotes the multiplicity of 2 in A_2. Since $C = A \times B$, we may set $C = c \cdot 2^{m_2(A_2)}$. Consequently, simplifying (12) yields

$$a_2 \times B_2 \pmod{2^{m-m_2(A_2)}} = c \pmod{2^{m-m_2(A_2)}},$$

which implies

$$B_2 \pmod{2^{m-m_2(A_2)}} = a_2^{-1} \times c \pmod{2^{m-m_2(A_2)}},$$

where a_2^{-1} denotes the inverse of a_2 in $\mathbb{Z}^*_{2^{m-m_2(A_2)}}$. Note that $m_2(A_2)$ is smaller than or equal to $m_2(A)$, but the case "$m_2(A_2) = m_2(A)$" happens with probability $1 - \frac{1}{2^m}$, which is close to 1 if m is not too small. Thus, in our case we may assume that $m_2(A_2) = m_2(A)$ and get

$$\mathrm{LSB}_{1^{\sim}m-m_2(A)}(B) = a_2^{-1} \times c \pmod{2^{m-m_2(A)}}, \tag{13}$$

which completes the proof.

Moreover, if $B \leq 2^{m-m_2(A)}$, then B can be completely determined immediately. Following corollary shows our method for searching k in LSBS-RSA

Corollary 1. *In (α, β, γ)-LSBS RSA, $LSB_{1^{\sim}\min\{\beta,2\alpha\}-\sigma}(k)$ can be computed in polynomial time in n, where σ denotes the multiplicity of 2 in $\varphi(N)$.*

Proof. From RSA equation we have $ed - 1 = k \cdot \varphi(N)$. Since $d_0 = \mathrm{LSB}_{1^{\sim}\beta}(d)$ is known, we can compute $\mathrm{LSB}_{1^{\sim}\beta}(ed-1)$. In addition, $\mathrm{LSB}_{1^{\sim}2\alpha}(p+q)$ can be computed efficiently according to Lemma 2, and thus $\mathrm{LSB}_{1^{\sim}2\alpha}(\varphi(N))$ can be derived to the attacker immediately. Now, setting $C = ed - 1$, $A = \varphi(N)$, and $B = k$ in Lemma 3, we get the result:

$$\mathrm{LSB}_{1^{\sim}\min\{\beta,2\alpha\}-\sigma}(k) = (ed_0 - 1) \cdot \varphi^{-1} (\mathrm{mod}\, 2^{\min\{\beta,2\alpha\}-\sigma}), \tag{14}$$

which completes the proof.

Note that we have $k < e \approx 2^{\gamma}$ due to the process of RSA-key generation. Hence, if the public exponent e is small enough such that $\gamma \leq \min\{\beta, 2\alpha\} - \sigma$, then k can be completely determined immediately in polynomial time in n. On the other hand, if the public exponent e satisfying $\gamma > \min\{\beta, 2\alpha\} - \sigma$, Corollary 1 implies that finding k requires exhaustive search with cost $2^{\gamma-(\min\{\beta,2\alpha\}-\sigma)}$. Therefore, the size of K_c can be set to $\max\left\{1,\, 2^{\gamma-(\min\{\beta,2\alpha\}-\sigma)}\right\}$. Apply $|K_c|$ to the revised BDF attack, the corresponding complexity analysis is shown in the next section.

4 The Complexity Analysis

According to Corollary 1, the complexity of the BDF attack on LSBS-RSA is discussed in the two cases: small public exponent and large public exponent.

4.1 LSBS-RSA with Small Public Exponent e ($\gamma \leq \min\{\beta, 2\alpha\} - \sigma$)

In LSBS-RSA with small e satisfying $\gamma \leq \min\{\beta, 2\alpha\} - \sigma$, according to Corollary 1, the parameter k can be computed immediately. Hence, the term $|K_c|$ in (9), (10), and (11) can be replaced by $T_k(n)$, where $T_k(n)$ denotes the complexity of computing k from (14), which is polynomial time in n.

4.2 LSBS-RSA with Large Public Exponent e ($\gamma > \min\{\beta, 2\alpha\} - \sigma$)

For large public exponent e, *i.e.*, $\min\{\beta, 2\alpha\} - \sigma < \gamma$, we may set $|K_c| = 2^{\gamma-(\min\{\beta,2\alpha\}-\sigma)}$. Thus, the complexity of searching k in this case depends on the two cases: $\beta < 2\alpha$, and $2\alpha \leq \beta$.

In the case $\boldsymbol{\beta < 2\alpha}$ First we consider the case $\beta < 2\alpha$. According to Corollary 1, $\mathrm{LSB}_{1^{\frown}\beta-\sigma}(k)$ is known to the attacker. Thus, finding the unknown part of k requires exhaustive search with the cost $2^{\gamma-(\beta-\sigma)}$. After replacing $|K_c|$ in (9), (10), and (11) by $2^{\gamma-(\beta-\sigma)}$, we get the following results:

In the case $\beta < 2\alpha$ and $\beta \leq 2(\alpha-1)+\kappa$, we have

$$\begin{aligned} T_{BDF}(n) &= O\left(2^{\gamma-(\beta-\sigma)} \cdot 2^{\frac{\kappa}{2}+\frac{n}{4}-\frac{\beta}{2}} \cdot T'_{Cop}(n)\right) \\ &= O\left(2^{\frac{\kappa}{2}+\sigma} \cdot 2^{(\frac{n}{4}+\gamma)-\frac{3\beta}{2}} \cdot T'_{Cop}(n)\right). \end{aligned}$$

In the case $\beta < 2\alpha$ and $\beta = 2(\alpha-1)+\kappa+1$, we get $2\alpha+\kappa-1 < 2\alpha$, which implies $\kappa \leq 0$. Since κ denotes the multiplicity of 2 in k, we get $\kappa = 0$, which conducts to $\beta = 2\alpha - 1$. Thus,

$$\begin{aligned} T_{BDF}(n) &= O\left(2^{\gamma-(\beta-\sigma)} \cdot (2^{\frac{n}{4}-\alpha}) \cdot T'_{Cop}(n)\right) \\ &= O\left(2^{\sigma} \cdot 2^{(\frac{n}{4}+\gamma)-(\alpha+\beta)} \cdot T'_{Cop}(n)\right). \end{aligned}$$

In the case $\beta < 2\alpha$ and $\beta = 2(\alpha-1)+\kappa+2$, we get $2\alpha+\kappa < 2\alpha$, which implies $\kappa \leq -1$. This is a contradiction for any non-negative integer κ. The same result for the case $\beta < 2\alpha$ and $\beta \geq 2(\alpha-1)+\kappa+3$, we get $2\alpha+\kappa+1 \leq \beta < 2\alpha$. It implies that $\kappa \leq -2$, which is also a contradiction.

In the case $\boldsymbol{2\alpha \leq \beta}$ Secondly, we consider the case $2\alpha \leq \beta$. According to Corollary 1, $\mathrm{LSB}_{1^{\frown}2\alpha-\sigma}(k)$ is known to the attacker. Thus, finding the unknown part of k requires the exhaustive search with the cost $2^{\gamma-(2\alpha-\sigma)}$. Replacing $|K_c|$ by $2^{\gamma-(2\alpha-\sigma)}$ in (9), (10), and (11), we get the following results:

In the case $2\alpha \leq \beta$ and $\beta \leq 2(\alpha-1)+\kappa$, we have

$$\begin{aligned} T_{BDF}(n) &= O\left(2^{\gamma-(2\alpha-\sigma)} \cdot 2^{\frac{\kappa}{2}+\frac{n}{4}-\frac{\beta}{2}} \cdot T'_{Cop}(n)\right) \\ &= O\left(2^{\frac{\kappa}{2}+\sigma} \cdot 2^{(\frac{n}{4}+\gamma)-\left(2\alpha+\frac{\beta}{2}\right)} \cdot T'_{Cop}(n)\right). \end{aligned}$$

In the case $2\alpha \leq \beta$ and $\beta = 2(\alpha-1)+\kappa+1$, we have

$$\begin{aligned} T_{BDF}(n) &= O\left(2^{\gamma-(2\alpha-\sigma)} \cdot (2^{\frac{n}{4}-\alpha}) \cdot T'_{Cop}(n)\right) \\ &= O\left(2^{\sigma} \cdot 2^{(\frac{n}{4}+\gamma)-3\alpha} \cdot T'_{Cop}(n)\right). \end{aligned}$$

The same result above in the case $2\alpha \leq \beta$ and $\beta = 2(\alpha-1)+\kappa+2$, and thus we ignore it.

In the case $2\alpha \leq \beta$ and $\beta \geq 2(\alpha-1)+\kappa+3$, we have

$$\begin{aligned} T_{BDF}(n) &= O\left(2^{\gamma-(2\alpha-\sigma)} \cdot 2^{\kappa+\frac{n}{4}+\alpha-\beta} \cdot T'_{Cop}(n)\right) \\ &= O\left(2^{\kappa+\sigma} \cdot 2^{(\frac{n}{4}+\gamma)-(\alpha+\beta)} \cdot T'_{Cop}(n)\right). \end{aligned}$$

Condition	$T_{BDF}(n)$
$\beta \leq 2(\alpha-1)+\kappa$	$O\left(T_k(n) \cdot 2^{\frac{n}{4}-\frac{\beta}{2}} \cdot T'_{Cop}(n)\right)$
$\beta = 2\alpha+\kappa$,or $\beta = 2\alpha+\kappa-1$	$O\left(T_k(n) \cdot (2^{\frac{n}{4}-\alpha}) \cdot T'_{Cop}(n)\right)$
$\beta \geq 2\alpha+\kappa+1$	$O\left(T_k(n) \cdot 2^{\frac{n}{4}+\alpha-\beta} \cdot T'_{Cop}(n)\right)$

Table 1. The Summary of the BDF Attack on LSBS-RSA with Small Public Exponent.

4.3 Summary of the Revised BDF Attack on LSBS-RSA

We give the summary for the complexity of the revised BDF attack on LSBS-RSA. We just count in the complexity of the exponent, but eliminate the complexity of polynomial time. In addition, σ and κ are both small constants with high probability, and thus we can ignore them in the "Big O" notation.

Table 1 shows the complexity of the revised BDF attack on LSBS-RSA when $\gamma \leq \min\{\beta, 2\alpha\} - \sigma$.

As can be seen in Table 1, γ is independent to the complexity of BDF attack on LSBS-RSA. However, in case of $\beta \geq 2\alpha+\kappa+1$, $T_{BDF}(n)$ increases as α increases, which is further supporting Steinfeld and Zheng's argument [14]: Low public exponent LSBS-RSA is inherently resistant to the partial key exposure attack.

Moreover, if we set $\chi_{\max} = \max\{2\alpha, \beta\}$ and $\chi_{\min} = \min\{2\alpha, \beta\}$, then all the complexities in the exponential cost are in the interval:

$$\left[\tfrac{n}{4} - \tfrac{1}{2}\chi_{\max},\ \tfrac{n}{4} - \tfrac{1}{2}\chi_{\min}\right].$$

Therefore, we conclude that the complexity of the revised BDF attack on (α, β, γ)-LSBS RSA with small e is in the range

$$O\left(T_k(n) \cdot 2^{\frac{n}{4}-\frac{1}{2}\chi_{\max}} \cdot T'_{Cop}(n)\right) \leq T_{BDF}(n) \leq O\left(T_k(n) \cdot 2^{\frac{n}{4}-\frac{1}{2}\chi_{\min}} \cdot T'_{Cop}(n)\right). \quad (15)$$

Table 2 shows the complexity of the revised BDF attack on LSBS-RSA when $\min\{\beta, 2\alpha\} - \sigma < \gamma$. As shown in the table, the complexity of the revised BDF attack is independent to α in the case $\beta < 2\alpha$ and $\beta \leq 2(\alpha-1)+\kappa$. In the other cases, the complexity decrease as α and β increase.

All the complexities in exponential cost are in the interval:

$$\left[(\tfrac{n}{4}+\gamma) - \tfrac{3}{2}\chi_{\max},\ (\tfrac{n}{4}+\gamma) - \tfrac{3}{2}\chi_{\min}\right].$$

Therefore, we conclude that the complexity of the revised BDF attack on (α, β, γ)-LSBS RSA with large e is in the range

$$O\left(2^{(\frac{n}{4}+\gamma)-\frac{3}{2}\chi_{\max}} \cdot T'_{Cop}(n)\right) \leq T_{BDF}(n) \leq O\left(2^{(\frac{n}{4}+\gamma)-\frac{3}{2}\chi_{\min}} \cdot T'_{Cop}(n)\right). \quad (16)$$

From (15) and (16), we know that an LSBS-RSA is more vulnerable under the BDF attack as $\chi_{\max} = \max\{2\alpha, \beta\}$ increases proportionally with the size of N.

Condition I	Condition II	$T_{BDF}(n)$
$\beta < 2\alpha$	$\beta \leq 2(\alpha-1)+\kappa$	$O\left(2^{(\frac{n}{4}+\gamma)-\frac{3\beta}{2}} \cdot T'_{Cop}(n)\right)$
$\beta < 2\alpha$	$\beta = 2\alpha-1$ and $\kappa = 0$	$O\left(2^{(\frac{n}{4}+\gamma)-(\alpha+\beta)} \cdot T'_{Cop}(n)\right)$
$2\alpha \leq \beta$	$\beta \leq 2(\alpha-1)+\kappa$	$O\left(2^{(\frac{n}{4}+\gamma)-(2\alpha+\frac{\beta}{2})} \cdot T'_{Cop}(n)\right)$
$2\alpha \leq \beta$	$\beta = 2\alpha+\kappa$, or $\beta = 2\alpha+\kappa-1$	$O\left(2^{(\frac{n}{4}+\gamma)-3\alpha} \cdot T'_{Cop}(n)\right)$
$2\alpha \leq \beta$	$\beta \geq 2\alpha+\kappa+1$	$O\left(2^{(\frac{n}{4}+\gamma)-(\alpha+\beta)} \cdot T'_{Cop}(n)\right)$

Table 2. The Summary of the BDF Attack on LSBS-RSA with Large Public Exponent.

5 Further Discussions

5.1 The Relation between (α, β, γ) and $(\alpha, 0, \gamma)$-LSBS RSA

The following result shows that for $\beta \leq 2\alpha$ and small difference of γ and β, to break (α, β, γ)-LSBS RSA is as hard as to break $(\alpha, 0, \gamma)$-LSBS RSA.

Theorem 3. *(Revised Theorem 4 in [14]) In (α, β, γ)-LSBS RSA, given (N, e, d_0), suppose an algorithm $\boldsymbol{A}$ can factor N in time $T_A(n)$, where $d_0 = LSB_{1^{\sim}\beta}(d)$ and $\beta \leq 2\alpha$. Then, there exists a factoring algorithm $\boldsymbol{F}$ for $(\alpha, 0, \gamma)$-LSBS RSA, that given (N, e), factors N in time $T_F(n)$, where*

$$T_F(n) = O\left(2^{\gamma-\beta} \cdot \left(T_A(n) + n^2\right)\right).$$

Proof. The proof is almost the same as the proof of the theorem 4 in [14]. The difference is that the cost for exhaustive search for k is reduced to $O\left(2^{\gamma-\beta}\right)$ rather than $O(2^{\gamma})$. Thus, for each candidate $k_c \in K_c$, we may compute

$$d_0 = e^{-1}\left[1 + k_c(N + 1 - s_0)\right] \pmod{2^{2\alpha}}, \tag{17}$$

where $s_0 \equiv p + q \pmod{2^{2\alpha}}$ is available according to Lemma 2. Consequently, $d_0 = \mathrm{LSB}_{1^{\sim}2\alpha}(d)$ consists of the 2α least significant bits of d, which also consists of $\mathrm{LSB}_{1^{\sim}\beta}(d)$. Applying (N, e, d_0) to the input of $\mathbf{A}$ succeeds to factor N in time

$$O\left(2^{\gamma-\beta} \cdot \left(T_A(n) + n^2\right)\right),$$

which denotes the complexity of $T_F(n)$.

Note that (17) also implies that $\mathrm{LSB}_{1^{\sim}2\alpha}(d)$ is leaked in $(\alpha, 0, \gamma)$-LSBS RSA if the cost of $2^{\gamma-\beta}$ is feasible under current computational capability. Therefore, for $\beta \leq 2\alpha$ and $\gamma - \beta < E_s$, where E_s denotes the bit number of the feasible exhaustive search, Theorem 3 also shows the hardness of breaking (α, β, γ)-LSBS RSA is equivalent to that of $(\alpha, 0, \gamma)$-LSBS RSA.

5.2 Feasibility and Further Reducing the Cost of Searching k

Under the current computational capability, we may set $E_s = 64$, which means the exhaustive search for$O\left(2^{64}\right)$ is feasible. According to our result, The revised BDF attack on LSBS-RSA with small e , *i.e.*, $\gamma \leq \min\{\beta, 2\alpha\} - \sigma$, is feasible if

$$\tfrac{n}{4} - \tfrac{1}{2}\chi_{\min} \leq 64.$$

For LSBS-RSA with large e, *i.e.*, $\gamma > \min\{\beta, 2\alpha\} - \sigma$, the attack is feasible if

$$(\tfrac{n}{4} + \gamma) - \tfrac{3}{2}\chi_{\min} \leq 64.$$

We should point out that our method for finding k is still a kind of brute method. In fact, we can estimate the value of k before the exhaustive search. Denote the estimation of φ to be $\varphi_E := N+1-2\lceil\sqrt{N}\rceil$. Compute $\widetilde{k}$ and $\widetilde{d}$ by using Euclidean algorithm such that $e\widetilde{d} = \widetilde{k}(N+1-2N)+1$, where $0 < \widetilde{k} < e$ and $0 < \widetilde{d} < \varphi_E$. Then, searching k from $\widetilde{k}$ with the fixed part of least significant bits will further reduce the cost.

6 Conclusion and Future Work

In this paper we improve the BDF attack on LSBS-RSA. With our improvement, the complexity of the BDF attack is further reduced with less cost for exhaustive search. More precisely, we show that the lower bound of exponential cost in the BDF attack increases with decreasing $\max\{2\alpha, \beta\}$, and the upper bound of exponential cost in the BDF attack decreases with increasing $\min\{2\alpha, \beta\}$. Our result is further supporting the claim in [14]: Low public exponent LSBS-RSA is resistant to partial key exposure attacks but large public exponent LSBS-RSA is vulnerable under the attacks.

To further reduce the complexity of the BDF attack, we may focus on improving the efficiency of Coppersmith's method, such as [6], [7]. Moreover, an open question has been mentioned for many times: whether the information of the $\frac{n}{4}$ least significant bits of p (or q) is the minimum requirement to factor N in polynomial time? Moreover, to further extend the partial key exposure attack on LSBS-RSA, the lattice technique should be considered to analyze.

LSBS-RSA is beneficial to computational efficiency of server-aided signature generation, such as [4]. However, we believe that an RSA system with modulus primes sharing a large number of bits also raises the risk in the security [15], [16]. It is a trade-off between the efficiency and the security level. Thus, one should be more careful in using such RSA variants.

Acknowledgement

The authors would like to thank Ron Steinfeld for his helpful discussion and anonymous reviewers for their valuable comments. This work was supported in

part by the National Science Council, Taiwan, under Contract NSC 96-2628-E-007-025-MY3 and NSC 096-2917-I-007-022, the Ministry of Education of Singapore under grant T206B2204, and the Australian Research Council under ARC Discovery Project DP0665035.

References

1. D. Boneh, G. Durfee and Y. Frankel, "An Attacks on RSA Given a Small Fraction of the Private Key Bits," Advanced in Cryptology – ASIACRYPT '98, LNCS 1514, Springer-Verlag, pp.25-34, 1998.
2. D. Boneh, G. Durfee and Y. Frankel, "Exposing an RSA Privae Key Given a Small Fraction of its Bits," Full version of the work from Asiacrypt'98, available at http://crypto.stanford.edu/~dabo/abstracts/bits_of_d.html, 1998.
3. J. Blömer and A. May, "New Partial Key Exposure Attacks on RSA," Advanced in Cryptology – CRYPTO'03, LNCS 2729, Springer-Verlag, pp.27-43, 2003.
4. M. Bellare and P. Rogaway, "The exact security of digital signatures: How to sign with RSA and Rabin," Advanced in Cryptology – EUROCRYPTO'96, LNCS 1070, Springer-Verlag, pp.399-416, 1996.
5. D. Coppersmith, "Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known," Proceedings of Eurocrypt'96, LNCS 1070, pp. 178–189, 1996.
6. J-S. Coron, "Finding Small Roots of Bivariate Integer Polynomial Equations Revisited," Advanced in Cryptology – EUROCRYPTO'04, LNCS 3027, Springer-Verlag, pp.492-505, 2004.
7. J-S. Coron, "Finding Small Roots of Bivariate Integer Polynomial Equations: A Direct Approach," Advanced in Cryptology – CRYPTO'07, LNCS 4622, Springer-Verlag, pp.379-394, 2007.
8. M. Ernst, E. Jochemsz, A. May, B. de Weger, "Partial Key Exposure Attacks on RSA up to Full Size Exponents," Advanced in Cryptology – EUROCRYPT'05, Springer-Verlag, pp.371-386, 2005.
9. J. Hastad, "Solving simultaneous modular equations of low degree," SIAM J. of Computing, Vol. 17, pp.336-341, 1988.
10. H.-M. Sun, W.-C. Yang and C.-S. Laih, "On the design of RSA with short secret exponent," Proceedings of Asiacrypt'99, LNCS 1716, pp. 150–164, 1999.
11. H.-M. Sun and C.-T. Yang, "RSA with balanced short exponents and its application to entity authentication," *Public Key Cryptography - PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pp. 199–215. Springer, 2005.
12. R. Rivest, A. Shamir and L. Aldeman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM* , Vol. 21, No.2, pp.120-126, 1978.
13. R. Steinfeld, and Y. Zheng, "An Advantage of Low-Exponent RSA with Modulus Primes Sharing Least Significant Bits," in *Topic in Cryptology - CT-RSA 2001*, ser. Lecture Notes in Computer Science, D. Naccache, Ed. Heidelberg: Springer, 2001, vol. 2020, pp. 52-62.
14. R. Steinfeld, and Y. Zheng, "On the Security of RSA with Primes Sharing Least-Significant Bits," *Appl. Algebra Eng. Commun. Comput.*, Heidelberg: Springer, 2004, vol. 15, no. 3(4), pp. 179-200.
15. B. de Weger, "Cryptanalysis of RSA with small prime difference," *Applicable Algebra in Engineering, Communication and Computing*, Vol. 13, pp. 17-28, 2002.

16. Y.-D. Zhao, and W.-F. Qi, "Small Private-Exponent Attack on RSA with Primes Sharing Bits," in *Proc. Information Security Conference 2007 — ISC 2007*, ser. Lecture Notes in Computer Science, J. Garay et al., Eds. Heidelberg: Springer, 2007, vol. 4779, pp. 221-229.