Small Public Exponent Brings More: Improved Partial Key Exposure Attacks against RSA

Yansong Feng[1,2], Abderrahmane Nitaj[3], and Yanbin Pan[1,2]

[1] Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
[2] School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China
{fengyansong,panyanbin}@amss.ac.cn
[3] Normandie Univ, UNICAEN, CNRS, LMNO, 14000 Caen, France
abderrahmane.nitaj@unicaen.fr

Abstract. Let (N, e) be a public key of the RSA cryptosystem, and d be the corresponding private key. In practice, we usually choose a small e for quick encryption. In this paper, we improve partial private key exposure attacks against RSA with MSBs of d and small e. The key idea is that under such a setting we can usually obtain more information about the prime factors of N and then, by solving a univariate modular polynomial equation using Coppersmith's method, N can be factored in polynomial time. Compared to previous results, we reduce the number of the leaked bits in d that are needed to mount the attack by $\log_2(e)$ bits. For $e = 65537$, previous work required an additional enumeration of 17 bits to achieve our new bound, resulting in a 2^{10} (or 1,024) x increase in time consumption. Furthermore, our experiments show that for a 1024-bit modulus N, our attack can achieve the theoretical bound on a simple personal computer, which verifies the new method.

Keywords: RSA, Factorization, Coppersmith's method, Partial key attack

1 Introduction

The RSA cryptosystem, one of the most worldwide used public key cryptosystems, was proposed by Rivest, Shamir and Adleman [RSA78] in 1978. Its security is based on the hardness of the factorization problem. In the key generation phase of RSA, Alice first selects two prime numbers p and q, computes the public modulus $N = pq$, and then chooses a random integer e coprime to $\phi(N) = (p-1)(q-1)$ as the public exponent, and computes d such that $ed \equiv 1 \mod \phi(N)$, as the secret exponent.

As a famous public-key encryption scheme, there has been much research about the cryptanalysis of RSA. Wiener's attack [Wie90] showed that RSA could be broken when the secret exponent is small, typically $d < N^{0.25}$. Therefore, it is insecure to choose a small d to reduce the cost of decryption. As a follow-up result, Boneh and Durfee [BD99] gave a new method showing that Wiener's attack can be extended to $d < N^{0.292}$, which still remains the best bound despite several efforts [HM09,HM10,KSI11,TK16]. Note that in all these attacks, the adversary has no additional available bits of the secret key $(p, q, \phi(N), d)$. Moreover, Rivest et al. [RSA78,Mil75] showed that one can factor N probabilistically in polynomial time when d is known. Then Coron and May [May04,CM07] proposed a deterministic algorithm to factor N with the known d.

In a so-called partial key exposure attack, one can obtain some information about the secret key, e.g. via some side-channel leakage like [ZvdPYS22]. Coppersmith [Cop96,Cop97] showed a polynomial-time attack exists when only half-bits of the prime p are given. As a direct application of Coppersmith's result, Boneh et al. [BDF98a] showed that only a quarter of the least significant bits (LSB) of d is enough to factor N when e is sufficiently small. For the most significant bits (MSBs) case, they showed that when $e = N^{\alpha}$ with $\frac{1}{4} < \alpha < \frac{1}{2}$, only αn MSBs of d are required to factor N where $n = \log_2(N)$. Later, several results were proposed for larger e [BM03] or even full size $e \approx N$ [EJMdW05]. In [EJMdW05], Ernst et al. also studied the attacks with leaked MSBs/LSBs of small d, which has been further improved by [Aon09]. Also, there are several results about partial key exposure attack for larger d [TK19,STK20].

Since small e is the usual setting in practice to increase the efficiency of encryption, such as the parameters in the TLS/SSL protocol for Apple, Microsoft, IACR, and Arxiv, we are more interested in the case when $1 < e < N^{\frac{1}{4}}$. An interesting open problem was once proposed by Boneh et al. [BDF98a] in 1998: is it enough to mount an attack given $\frac{1}{4}n$ bits of d in positions $\frac{1}{4}n$ to $\frac{1}{2}n$ when e is constant small? Later, they gave a positive answer to this question in the full version of their paper [BDF98b]: roughly $\frac{3}{4}n$ MSBs of d are needed to mount their attack when $1 < e < N^{\frac{1}{4}}$. Moreover, they try to find p with $\frac{1}{4}n - \log_2(e)$ bits is known in Theorem 3.3 in [BDF98b]. If noncontinuous MSBs of d are allowed to be obtained, the cost of time in their attack is $e \cdot \text{poly}(n)$ when nearly $\frac{1}{4}n$ MSBs are known or $\text{poly}(n)$ when $\log_2(e) + \frac{1}{4}n$ MSBs are known, where $\text{poly}(n)$ denotes polynomial time in n. A natural question is whether these MSBs are necessary for factoring RSA modulus in polynomial time when e is small.

In this paper, we present a negative answer to the above question, that is, we can break RSA even with fewer leaked bits of d when e is small. Note that we can usually find the exact value of k such that the key equation $ed = 1 + k(N - (p+q) + 1)$ holds when e is small and enough MSBs of d are known. It is well known that the MSBs (or LSBs) of d can yield nearly the same size as the MSBs (or LSBs) of p under some reasonable conditions, then Coppersmith's classical result [Cop97] can be employed. However, besides the MSBs (or LSBs) of p that we can obtain from the MSBs (or LSBs) of d, we can also get $(p+q) \mod e$ from the key equation, which will yield $p \mod e$ when the factorization of e is known. Then we present an efficient algorithm based on Coppersmith's method to recover p using its MSBs (or LSBs) and $p \mod e$. With the help of $p \mod e$, the whole attack needs fewer leaked bits in d than previous attacks. Simply speaking, the key idea in our attack is to extract and explore the additional information of $p \mod e$.

Based on the former idea, we find that given an n-bit RSA modulus $N = pq$ with $q < p < 2q$ and $p - q = N^{\frac{1}{2}-\theta}$ with $0 < \theta < \frac{1}{4}$, let $e = N^{\alpha}$ ($\alpha < \frac{1}{4}$) be a small public exponent with known factorization and r distinct prime factors, and $d = N^{\delta}$ be a private exponent. We show that we can factor N with time polynomial in 2^r and $\log_2(N)$ in the following two cases.

Continuous MSBs leaked. We know $(\delta - \gamma)n$ MSBs of d with

$$\gamma < \delta + \alpha - \theta - \frac{3}{4},$$

where γn is the number of unknown bits of d. That is to say, we only need $(\frac{3}{4} - \alpha)n$ bits compared with $\frac{3}{4}n$ in [BDF98b] when $\theta \approx 0$.

Noncontinuous MSBs leaked. We know $(\delta + \alpha - 1)n$ MSBs of d and $(\frac{5}{4} + \theta - \alpha - \delta)n$ bits after $(\delta - \frac{1}{2})n$ MSBs of d. In other words, our attack only requires $(\frac{1}{4} + \theta)n$ known bits of d in total.

Finally, we also provide an algorithm for both MSBs and LSBs cases. Our attack only requires $(\frac{1}{4} + \theta)n$ bits of d like MSBs case but needs $e \cdot \text{poly}(n)$ time.

We would also like to point out that

- It is not so hard to factor e in practice since we just consider small e. For example, if $n = 1024$, then e has at most 256 bits, which is very easy to factor.
- In practice, it seems reasonable to assume $\theta \approx 0$. A rough estimation shows that $p - q < \frac{1}{2^c}\sqrt{N}$ with probability about $\frac{1}{2^c}$, in which $\theta \approx \frac{c}{n}$. However, to eliminate θ completely, we need to enumerate almost c bits of d to increase the number of known bits.
- As in [BDF98b], we also consider the full size d ($\delta \approx 1$) and $\theta \approx 0$. Our method reduces $\log_2(e)$ MSBs compared with Boneh et al.'s attack [BDF98b]. That is, for the continuous MSBs case, we just need $(\frac{3}{4} - \alpha)n$ bits compared with $\frac{3}{4}n$ MSBs in [BDF98b]. If noncontinuous MSBs can be obtained, nearly $0.25n$ MSBs are enough for our attack, whereas $\log_2(e) + 0.25n$ MSBs are needed for their attack [BDF98b]. We provided a summary of comparison in Figure 1.

To achieve our new bound, previous work [BDF98b] required an additional enumeration of $\log_2 e$ bits to achieve our new bound. For the widely used $e = 65537$, our new algorithm eliminates this enumeration, which leads to a 2^{10} (or 1,024) x improvement in the running time.

Moreover, our attack can achieve the theoretical bound for 1024 bits N, which means our method is practical. We provide an efficient open source implementation of our algorithm in SageMath. The source code is available at:

https://github.com/fffmath/MSBsOfPrivateKeyAttack.

With this implementation, we conducted several experiments, and the experimental results can be found in Section 4.

[4] We estimate the time consumption to be $2^{16} \times 2000$ s and $2^{128} \times 100$ s, respectively.

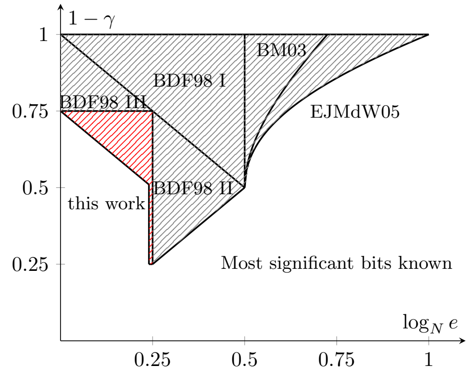

Fig. 1: The results for known MSBs of d.

$\log_2 e$	[BDF98b]'s Bound	Ours	Leaked MSBs	Time in [BDF98b]	Ours
17	768	752	768	2095.55 s	1.5 s
			752	-	1028.72 s $\approx$ 0.29 h
129	768	640	768	111.64 s	0.12 s
			640	-	4057.90 s$\approx$1.13 h

Table 1: Comparison of running time for 1024-bit N(" - " means longer than 24 h[4]).

Roadmap. Our paper is organized as follows. We provide some necessary background for our approaches in Section 2. In Section 3, we present our main result for the MSBs case and then generalize it to both MSBs and LSBs case. Section 4 describes the experiments that validate our analysis. Finally, we provide a brief conclusion in Section 5.

2 Notations and Preliminaries

Let $\mathbb{Z}$ denote the ring of integers. We use lowercase bold letters (e.g., $\mathbf{v}$) for row vectors and uppercase bold letters (e.g., $\mathbf{A}$) for matrices.

MSBs abbreviates the most significant bits and LSBs abbreviates the least significant bits. $\Omega(\cdot)$ denotes the lower bound of the asymptotic complexity, and $\mathcal{O}(\cdot)$ (Big-O) denotes the upper bound of the asymptotic complexity.

For any polynomial $h(x_1 \cdots, x_k) \in \mathbb{Z}[x_1, \cdots, x_k]$, we use $\|h(x_1 \cdots, x_k)\|$ to denote the Euclidean norm of the coefficient vector of $h(x_1 \cdots, x_k)$. That is, for $h(x_1 \cdots, x_k) = \sum h_{i_1,\cdots,i_k} {x_1}^{i_1} \cdots {x_k}^{i_k}$, it holds that

$$\|h(x_1 \cdots, x_k)\| = \sqrt{\sum h^2_{i_1,\cdots,i_k}}.$$

2.1 Lattices, SVP, and LLL

Let $m \geq 1$ be an integer. A lattice is a discrete additive subgroup of $\mathbb{R}^m$. An equivalent definition is presented as follows.

Definition 1 (Lattice). *Let $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n} \in \mathbb{R}^m$ be n linearly independent vectors with $n \leq m$. The lattice $\mathcal{L}$ spanned by $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$ is the set of all integer linear combinations of $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$, i.e.,*

$$\mathcal{L} = \left\{ \mathbf{v} \in \mathbb{R}^m \mid \mathbf{v} = \sum_{i=1}^{n} a_i \mathbf{v_i}, a_i \in \mathbb{Z} \right\}.$$

We call n as the rank of $\mathcal{L}$ and m as the dimension of $\mathcal{L}$. The lattice $\mathcal{L}$ is said to be full rank if $n = m$. Define $\mathbf{B} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_n \end{pmatrix}$, which is denoted as the matrix basis of $\mathcal{L}$. The determinant of $\mathcal{L}$ is defined as $\det(\mathcal{L}) = \sqrt{\det\left(\mathbf{B}\mathbf{B}^T\right)}$, where $\mathbf{B}^T$ is the transpose of $\mathbf{B}$. If $\mathcal{L}$ is full rank, this reduces to $\det(\mathcal{L}) = |\det(\mathbf{B})|$.

The Shortest Vector Problem (SVP) is one of the famous computational problems in lattices.

Definition 2 (SVP). *Given a lattice $\mathcal{L}$, the Shortest Vector Problem (SVP) asks to find a non-zero lattice vector $\mathbf{v} \in \mathcal{L}$ of minimum Euclidean norm, i.e., find $\mathbf{v} \in \mathcal{L} \backslash \{\mathbf{0}\}$ such that $\|\mathbf{v}\| \leq \|\mathbf{w}\|$ for all non-zero $\mathbf{w} \in \mathcal{L}$.*

SVP has been proven NP-hard under randomized reduction [Ajt98]. Nevertheless, there exist algorithms to efficiently find a relatively short vector, such as the famous LLL algorithm introduced by Lenstra, Lenstra, and Lovász [LLL82] in 1982. The following result [May03] presents the upper bound for the norm of the i-th vector in the LLL-reduced basis using the determinant of the lattice.

Lemma 1 (LLL Algorithm). *Given an n-dimensional lattice $\mathcal{L}$, we can find an LLL-reduced basis $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$ of $\mathcal{L}$ in polynomial time, which satisfies*

$$\|\mathbf{v_i}\| \leq 2^{\frac{n(n-1)}{4(n+1-i)}} \det(\mathcal{L})^{\frac{1}{n+1-i}}, \quad \textit{for} \quad i = 1, \ldots, n.$$

2.2 Coppersmith's Method

Suppose $f \in \mathbb{Z}[x_1, ...x_k]$ is a polynomial with a small root $\mathbf{u} = (u_1, \ldots, u_k) \in \mathbb{Z}^k$ modulo some integer M. Here, a small root means $|u_i| < X_i$ for known bound X_i, for $i = 1, \ldots, k$. To find such a root, Coppersmith's method is usually employed. Below we will give a brief introduction to Coppersmith's method for solving modular equations. More details can be found in [May03].

Coppersmith's method first constructs a lattice $\mathcal{L}$ with the coefficient vector of a system of polynomials that has the same small root $\mathbf{u}$ of f modulo M^m

where m is some positive integer. For example, the polynomials can be selected as:

$$g_{[i_1,\ldots,i_k,i]} = x_1^{i_1} \cdot \ldots \cdot x_k^{i_k} f^i M^{m-i}, \text{ for } i = 0, \ldots, m.$$

Note that each $g_{[i_1,\ldots,i_k,i]}$ has the same small root $\mathbf{u}$ of f modulo M^m.

Coppersmith's method tries to find the short vectors, or equivalently, the short polynomials $g_1, \ldots, g_k$, in the lattice $\mathcal{L}$ by applying the LLL algorithm. Using the following result, due to Howgrave-Graham, and Lemma 1, we just need $\det(\mathcal{L}) < M^{m \dim(\mathcal{L})}$ to ensure that $g_1, \ldots, g_k$ have the same small root $\mathbf{u}$ with f, not only modulo M^m but also over $\mathbb{Z}$.

Lemma 2 (Howgrave-Graham [How97]). *Let $g(x_1, \ldots, x_k) \in \mathbb{Z}[x_1, \ldots, x_k]$ be a polynomial with at most ω monomials. Let M be a positive integer. If there exist k integers $(u_1, \ldots, u_k)$ satisfying the following two conditions:*

1. *$g(u_1, \ldots, u_k) \equiv 0 \mod M$,*
2. *there exist k positive integers $X_1, \ldots, X_k$ such that $|u_i| < X_i$ for $i = 1, \ldots, k$, and $\|g(x_1X_1, \ldots, x_kX_k)\| < \frac{M}{\sqrt{\omega}}$,*

then $g(u_1, \ldots, u_k) = 0$ holds over $\mathbb{Z}$.

Lastly, Coppersmith's method computes the desired root $\mathbf{u} = (u_1, \ldots, u_k)$ by solving the system of polynomial equations $g_i(x_1, \ldots, x_k) = 0$ for $i = 1, \ldots, k$.

In the multivariate scenario, that is, for $k > 1$, we usually assume the ideal generated by $g_1, \ldots, g_k$ being zero-dimensional, allowing us to compute the small root $\mathbf{u}$ by Gröbner basis [MR09,MNS21,MNS22,MN23]. However, when f is univariate ($k = 1$), we can directly compute the root of g_1 over $\mathbb{Z}$ without any assumption. In this paper, we just use Coppersmith's method for univariate polynomials as described in the following lemma, whose detailed proof can be found in the proof of Theorem 7 in [May03] or an analogous proof showed in [LZPL15]. We use the latter in this paper.

Lemma 3. *Suppose N has an unknown divisor $b > N^\beta$ and f is a monic and univariate polynomial with degree r, then we can find all solutions x_0 with*

$$|x_0| \leq N^{\frac{\beta^2}{r}}$$

of the equation $f(x) \equiv 0 \mod b$ in polynomial time of $(r, \log_2(N))$.

Our analysis is based on $\beta = \frac{1}{2}$ and $\deg(f) = 1$. More specifically, to solve $f(x) \equiv 0 \mod b$ with $|x| < X$, we choose the coefficient vectors of the following polynomials $g_i(xX)$ as the lattice basis matrix.

$$g_i(x) = f^i(x) N^{\max\{\lfloor \frac{m}{2} \rfloor - i, 0\}} \text{ for } i = 0, \ldots, m.$$

Additionally, we provide an example with $m = 8$, $\beta = \frac{1}{2}$ and $\deg(f) = 1$ in Table 2 for better understanding.

g_i	1	x	x^2	x^3	x^4	x^5	x^6	x^7	x^8
$g_0(xX)$	N^4	0	0	0	0	0	0	0	0
$g_1(xX)$	$*$	N^3X	0	0	0	0	0	0	0
$g_2(xX)$	$*$	$*$	N^2X^2	0	0	0	0	0	0
$g_3(xX)$	$*$	$*$	$*$	NX^3	0	0	0	0	0
$g_4(xX)$	$*$	$*$	$*$	$*$	X^4	0	0	0	0
$g_5(xX)$	$*$	$*$	$*$	$*$	$*$	X^5	0	0	0
$g_6(xX)$	$*$	$*$	$*$	$*$	$*$	$*$	X^6	0	0
$g_7(xX)$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	X^7	0
$g_8(xX)$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	X^8

Table 2: The matrix of the lattice with $m = 8$.

2.3 Factoring RSA Modulus with Some Hints

Let us introduce the (textbook) RSA scheme briefly, which consists of three polynomial-time algorithms: Key Generation, Encryption and Decryption.

Key Generation: Alice randomly selects two primes p and q with $q < p < 2q$, and then computes $N = pq$ and $\phi(N) = (p-1)(q-1)$. After that, Alice chooses a random integer e such that $\gcd(e, \phi(N)) = 1$, and compute d such that $ed \equiv 1 \mod \phi(N)$. The public key is (e, N) and the private key is d.

Encryption: To encrypt a message m, Bob computes the ciphertext $C = m^e \mod N$ and sends it to Alice.

Decryption: Alice compute $M^d \mod N$ to get the message m.

Note that textbook RSA could be insecure when using low encryption exponents (e.g., $e = 3$) or when facing the chosen plaintext attack. To avoid these problems, practical RSA implementations typically choose $e = 2^16+1$ and embed padding into the value m before encrypting it [Ble98,Mac13].

The security of the RSA cryptosystem is based on the problem of factoring large numbers and the RSA problem. Although it seems hard to factor big RSA modulus directly up to now, Coppersmith's method is usually employed to factor N when some additional hints are given. For example, when given half of the most significant bits of a prime factor, we can find the factorization of N [Cop96,Cop97,May03].

Lemma 4. *Given a n-bit RSA modulus $N = pq$ with $q < p < 2q$, if p_0 is an approximation of p such that $|p - p_0| < N^{\frac{1}{4}}$, then one can find p and factor N in polynomial time.*

The following Corollary shows that we can factor N when $|p - q|$ is small enough, which can also be derived by Fermat's Factoring Method [dW02].

Corollary 1. *Given n-bit RSA modulus $N = pq$ with $q < p < 2q$, suppose $|p - q| < N^{\frac{1}{4}}$, then one can find p and factor N in polynomial time.*

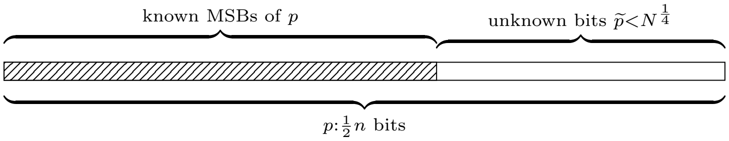

Fig. 2: Illustration of known bits of p for Lemma 4.

Proof. Note that $|p - \lceil\sqrt{N}\rceil| < |p - q| < N^{\frac{1}{4}}$. Therefore, $\lceil\sqrt{N}\rceil$ is an approximation of p satisfying Lemma 4. Then the corollary follows. □

2.4 Deriving Information from the Approximation

It is well known that the MSBs of d can be used to yield a good approximation of k, which satisfies $ed = 1 + k\phi(N)$.

Lemma 5 (Lemma 4.2 in [BDF98b]). *Given n-bit RSA modulus $N = pq$ with $q < p < 2q$, let $e = N^{\alpha}$ be a public exponent with $\alpha < 1$, and $d \leq N^{\delta}$ be a private exponent satisfying $ed \equiv 1 \mod (p-1)(q-1)$. Let d_0 be an approximation of d such that $|d - d_0| < N^{\gamma}$. If $\alpha + \delta < \frac{3}{2}$ and $\gamma < 1 - \alpha$, then*

$$\frac{ed - 1}{(p-1)(q-1)} = \left[\frac{ed_0 - 1}{N}\right] + k_1,$$

with a constant additive error $|k_1| \leq 14$.

Next, we generalize Lemma B.1 in [BDF98b] as below to show how to yield the MSBs of p from the approximation of $p + q$. For completeness, we present its proof, which is almost the same as the proof of Lemma B.1 in [BDF98b].

Lemma 6 (Lemma B.1 in [BDF98b]). *Given a n-bit RSA modulus $N = pq$ with $q < p < 2q$ and $p - q = N^{\frac{1}{2}-\theta}$ with $0 < \theta < \frac{1}{4}$, let $2(p+q) > S \geq p + q \geq 2\sqrt{N}$ be an approximation of $p + q$ with $|S - (p+q)| < N^{\beta}$. Define $p_0 = \frac{1}{2}(S + \sqrt{S^2 - 4N})$. Then p_0 is an approximation of p such that*

$$|p - p_0| < \frac{1}{2}\left(1 + 9N^{\theta}\right)N^{\beta}.$$

Proof. First, observe that $(p+q)^2 = (p-q)^2 + 4N$. Hence $(p+q)^2 > 4N$. Also, observe that $q < p < 2q$ implies $q < \sqrt{N}$, and $p + q < 3q < 3\sqrt{N}$. Denote $S' = p + q$ and $D' = \sqrt{S'^2 - 4N}$. We have $|S - S'| < N^{\beta}$, $S + S' < 3S' < 9\sqrt{N}$ and $D' = p - q = N^{\frac{1}{2}-\theta}$. Then $p = \frac{1}{2}(S' + D')$. Let $D = \sqrt{S^2 - 4N} \geq 0$, and

$p_0 = \frac{1}{2}(S + D)$. Then, since $D^2 - D'^2 = S^2 - S'^2$, we get

$$\begin{aligned}
|p - p_0| &\leq \frac{1}{2}(|S - S'| + |D - D'|) \\
&= \frac{1}{2}(|S - S'| + \frac{|D^2 - D'^2|}{D + D'}) \\
&= \frac{1}{2}(|S - S'| + \frac{|S - S'|(S + S')}{D + D'}) \\
&\leq \frac{1}{2}(1 + \frac{S + S'}{D + D'})N^\beta \\
&< \frac{1}{2}(1 + \frac{9\sqrt{N}}{N^{\frac{1}{2}-\theta}})N^\beta \\
&= \frac{1}{2}\left(1 + 9N^\theta\right)N^\beta.
\end{aligned}$$

This concludes the proof. □

Note that we ask $0 < \theta < \frac{1}{4}$ due to Corollary 1.

3 Factoring RSA Modulus with Known Bits of *d*

In practical implementations of RSA, the public exponent e is often chosen to be a small value like $e = 2^{16} + 1 = 65537$. This choice of e ensures efficient encryption and verification processes due to its low Hamming weight (only two bits set to 1). However, although this choice does not directly influence the security but also raises the problem to balance efficiency and security.

In this section, we focus on the partial key exposure attacks against RSA with small public exponent. More precisely, we consider the RSA modulus $N = pq$ with $p < q < 2q$, and $p - q = N^{\frac{1}{2}-\theta}$ with $0 < \theta < \frac{1}{4}$ due to Corollary 1, $e = N^\alpha$ with $0 < \alpha < \frac{1}{4}$, and in addition, an approximation of d is known.

In the following, we first introduce our improvement for the MSBs case by presenting a simple lemma that allows us to recover p using information from both $p \mod e$ and the MSBs of p. Additionally, note that some MSBs remain unused (See Section 4.2.8 in [MH24]), we can also complete the attack with less MSBs of d under conditions allowing for non-continuous leakage. Finally, we generalize the attack to the case when both MSBs and LSBs are leaked.

3.1 Factoring RSA Modulus with Known MSBs of *d*

We start with the following lemma, which shows that one can recover p using $p \mod e$ and additional MSBs of p.

Lemma 7. *Given a n-bit RSA modulus $N = pq$ with $q < p < 2q$, let $e = N^\alpha$ be a public exponent with $\alpha < \frac{1}{4}$. Suppose γn MSBs of p and $\overline{p} \equiv p \mod e$ are known, then one can find p and factor N in polynomial time of n when $\gamma > \frac{1}{4} - \alpha$.*

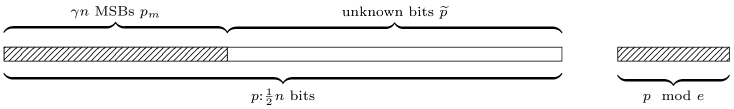

Fig. 3: Illustration of known bits of p and known $p \mod e$ for Lemma 7.

Proof. As in Figure 3, let $p = p_m 2^{\frac{1}{2}n - \gamma n} + \widetilde{p}$ where $p_m \approx N^{\gamma}$ is known. Then $p_0 = p_m 2^{\frac{1}{2}n-\gamma n}$ is an approximation of p with $|p - p_0| < N^{\frac{1}{2}-\gamma}$.

Since $\overline{p} \equiv p \mod e$, we can write $p = te + \overline{p}$. Denote $t_0 = \frac{p_0 - \overline{p}}{e}$. Then t_0 is an approximation of t with

$$\begin{aligned}|t - t_0| &= |t - \frac{p_0 - \overline{p}}{e}| \\ &= |\frac{te - (p_0 - \overline{p})}{e}| \\ &= |\frac{(p - \overline{p}) - (p_0 - \overline{p})}{e}| \\ &= |\frac{p - p_0}{e}| \\ &< N^{\frac{1}{2}-\gamma-\alpha}.\end{aligned}$$

Then, one can write $t = t_0 + \widetilde{t}$ with an unknown $\widetilde{t}$ satisfying $\widetilde{t} < N^{\frac{1}{2}-\gamma-\alpha}$.

Consider the univariate polynomial $g(x) = (t_0 + x)e + \overline{p}$. Then $\widetilde{t}$ is a solution of the equation of $g(x) \equiv 0 \mod p$. Since $\gcd(e, N) = 1$, then one can compute $e^{-1} \mod N$ and $f(x) = e^{-1} g(x) \mod p$. Note that f is monic, and has the same root as g. Since $\gamma > \frac{1}{4} - \alpha$, we have

$$\frac{1}{2} - \gamma - \alpha < \frac{1}{4},$$

which yields that $\widetilde{t}$ is a small root of $f(x) = 0 \mod p$. Therefore, we can find $\widetilde{p}$ and then factor N by Lemma 3. □

Below we present our main theorem. For simplicity, we assume e is a prime number.

Theorem 1. *Given a n-bit RSA modulus $N = pq$ with $q < p < 2q$ and $p - q = N^{\frac{1}{2}-\theta}$ with $0 < \theta < \frac{1}{4}$, let $e = N^{\alpha}$ be a small prime public exponent with $\alpha < \frac{1}{4}$, and $d = N^{\delta}$ be a private exponent satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. Let d_0 be an approximation of d such that $|d - d_0| < N^{\gamma}$. Then one can factor N in polynomial time if*

$$\gamma < \delta + \alpha - \theta - \frac{3}{4}.$$

Proof. Let d_0 be an approximation of d such that $|d - d_0| < N^{\gamma}$. Without loss of generality, we can assume that $d - d_0 \geq 0$. If $d < d_0$, then $d_0' = d_0 - N^{\gamma}$ is also a good approximation of d such that $0 \leq d - d_0' < N^{\gamma}$.

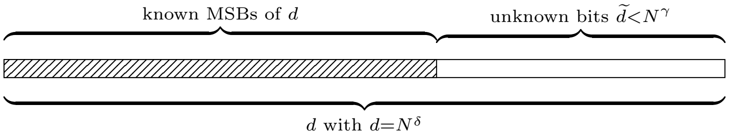

Fig. 4: Illustration of known bits of d for Theorem 1

Let $d = d_0 + \widetilde{d}$ where $0 \le \widetilde{d} < N^{\gamma}$ is unknown. The key equation of RSA implies that there exists an integer k such that

$$ed - k(p-1)(q-1) = 1.$$

We next show how to factor N step by step.

Step 1. Determine the candidate k. Since $\gamma < \delta + \alpha - \theta - \frac{3}{4}$, we have $\gamma < 1 + \alpha - \frac{3}{4} < 1 - \alpha$. Besides, it holds that $\alpha + \delta < \frac{1}{4} + 1 < \frac{3}{2}$. By Lemma 5, we have $k = \left[\frac{ed_0 - 1}{N}\right] + k_1 \approx N^{\alpha+\delta-1}$ with $|k_1| \le 14$. Then we can enumerate these candidates and k is one of them. Below, we assume the correct value of k is known.

Step 2. Recover the MSBs of p. The equation $ed - k(p-1)(q-1) = 1$ with $d = d_0 + \widetilde{d}$ gives

$$e\widetilde{d} + k(p+q) = k(N+1) - ed_0 + 1. \qquad (1)$$

Define $S = N + 1 - \frac{ed_0 - 1}{k}$. Then by Equation (1) we have

$$S - (p+q) = \frac{e\widetilde{d}}{k}.$$

Since $0 \le \widetilde{d} < N^{\gamma}$, and $k \approx N^{\alpha+\delta-1}$, then

$$0 \le S - (p+q) < \frac{N^{\alpha+\gamma}}{N^{\alpha+\delta-1}} = N^{1+\gamma-\delta}.$$

Since $S \ge p + q > 2\sqrt{N}$, then $S^2 > 4N$. Define $D = \sqrt{S^2 - 4N}$, and $p_0 = \frac{1}{2}(S + D)$. By Lemma 6, we get

$$|p - p_0| < \frac{1}{2}\left(1 + 9N^{\theta}\right) N^{1+\gamma-\delta}.$$

It shows that $\frac{1}{2}n - (1 + \gamma + \theta - \delta)n$ MSBs bits of p are known.

Step 3. Recover $\overline{p} \equiv p \mod e$. Consider the following equation again

$$ed = 1 + k(N - (p+q) + 1).$$

Note that $\gcd(k, e) = 1$. Similar to Theorem 7 in [BDF98a], we can compute

$$p + q \equiv N + 1 - k^{-1} \mod e,$$

which yields $s = p + q \mod e$. Then we can find $\overline{p} \equiv p \mod e$ by solving the following modular equation

$$x^2 - sx + N \equiv 0 \mod e. \tag{2}$$

Step 4. Factor N. To apply Lemma 7, we set $\frac{1}{2} - (1 + \gamma + \theta - \delta) > \frac{1}{4} - \alpha$, that is

$$\gamma < \delta + \alpha - \theta - \frac{3}{4}.$$

Then, we can factor N in polynomial time using Lemma 7. □

Especially when we consider $e = 2^{16} + 1$, as a prime number, Theorem 1 works directly. As in [BDF98b], as long as we can solve Equation 2 or know the factorization of e, the proof will work regardless of whether e is prime or not. By an analogous proof, we have the following Theorem 2.

Theorem 2. *Given n-bit RSA modulus $N = pq$ with $q < p < 2q$ and $p - q = N^{\frac{1}{2}-\theta}$ with $0 < \theta < \frac{1}{4}$, let $e = N^\alpha$ ($\alpha < \frac{1}{4}$) be a small public exponent with r known distinct prime factors and $d = N^\delta$ be a private exponent satisfying $ed \equiv 1 \mod (p-1)(q-1)$. Let d_0 be an approximation of d such that $|d - d_0| < N^\gamma$. Then one can factor N in time polynomial in $\log_2(N)$ and 2^r if*

$$\gamma < \delta + \alpha - \theta - \frac{3}{4}.$$

3.2 Factoring RSA Modulus with Less Known MSBs of d

Theorem 1 shows that we need $(\delta - \gamma)n = (\frac{3}{4} - \alpha + \theta)n$ MSBs of d. In fact, we can use $\frac{kN}{e}$ to get an approximation of d with $|d - \frac{kN}{e}| < N^{\delta - \frac{1}{2}}$, which means $\frac{1}{2}n$ MSBs of d is known. Therefore, we do not need the $\frac{3}{2}n - \alpha n - \delta n$ bits after $(\alpha + \delta - 1)n$ MSBs of d. That is, we only need $(\frac{3}{4}n - \alpha n) - (\frac{3}{2}n - \alpha n - \delta n) = \delta n - \frac{3}{4}n$ bits of d. We rewrite this result in the following Theorem.

Theorem 3. *Given a n-bit RSA modulus $N = pq$ with $q < p < 2q$ and $p - q = N^{\frac{1}{2}-\theta}$ with $0 < \theta < \frac{1}{4}$, let $e = N^\alpha$ be a prime public exponent with $\alpha < \frac{1}{4}$, and $d = N^\delta$ be a private exponent satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. One can factor N in polynomial time if $(\delta + \alpha - 1)n$ MSBs of d and $(\frac{5}{4} + \theta - \alpha - \delta)n$ bits after $(\delta - \frac{1}{2})n$ MSBs of d are known. That is, we only need $(\frac{1}{4} + \theta)n$ bits of d.*

Proof. Denote by $d_m^{(1)} \approx N^{\delta+\alpha-1}$ the known MSBs of d. Then $d_0 = d_m^{(1)} 2^{(1-\alpha)n}$ is an approximation of d with $|d - d_0| < N^{1-\alpha}$. Then, by Lemma 5, we can determine a few candidates of k. Below, we assume the correct value of k is known.

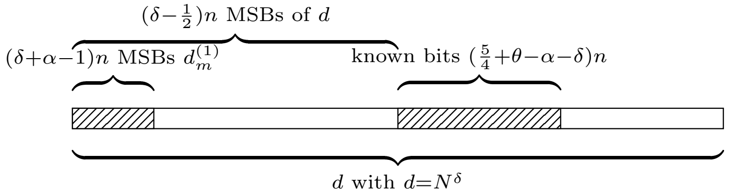
$(\delta-\frac{1}{2})n$ MSBs of d
$(\delta+\alpha-1)n$ MSBs $d_m^{(1)}$
known bits $(\frac{5}{4}+\theta-\alpha-\delta)n$
d with $d=N^{\delta}$

Fig. 5: Illustration of known bits of d for Theorem 3.

With the correct k, we claim that $d_0 = \left[\frac{kN}{e}\right]$ is a good approximation of d since

$$\begin{aligned}|d-d_0| &\approx |d-\frac{kN}{e}|\\ &= |\frac{ed-kN}{e}|\\ &= |\frac{k\phi(N)+1-kN}{e}|\\ &= |\frac{k(p+q-1)-1}{e}|\\ &< |\frac{k(p+q)}{e}|\\ &< |p+q|\\ &\approx N^{\frac{1}{2}}.\end{aligned}$$

Note $d = N^{\delta}$, then $d_0 = \left[\frac{kN}{e}\right]$ has almost the same $(\delta-\frac{1}{2})n$ MSBs of d. By enumeration, we can assume the $(\delta-\frac{1}{2})n$ MSBs of d are known.

Now we have

$$\left(\delta-\frac{1}{2}\right)n+\left(\frac{5}{4}+\theta-\alpha-\delta\right)n=\left(\frac{3}{4}-\alpha+\theta\right)n$$

MSBs of d. Then we have an approximation d_0 such that $|d-d_0| < N^{\delta+\alpha-\theta-\frac{3}{4}}$. Hence, by Theorem 1, we can factor N in polynomial time. □

Similarly, when the factorization of e is known, we have

Corollary 2. *Given a n-bit RSA modulus $N = pq$ with $q < p < 2q$ and $p - q = N^{\frac{1}{2}-\theta}$ with $0 < \theta < \frac{1}{4}$, let $e = N^{\alpha}$ be a public exponent with $\alpha < \frac{1}{4}$, and r known distinct prime factors, and $d = N^{\delta}$ be a private exponent satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. One can factor N with time polynomial in $\log_2(N)$ and 2^r if $(\delta+\alpha-1)n$ MSBs of d and $(\frac{5}{4}+\theta-\alpha-\delta)n$ bits after $(\delta-\frac{1}{2})n$ MSBs of d are known.*

Note that Corollary 2 implies that only $(\frac{1}{4}+\theta)n$ bits of d are needed. Note also that when e is enumerable and $d \approx N$, our Theorem 3 produces the same result as Theorem 3.3 in [BDF98b], requiring only $\frac{1}{4}n$ bits in the positions from $\frac{1}{4}n$ to $\frac{1}{2}n$.

However, when $e \approx N^{\frac{1}{4}}$, $d \approx N$, $\theta \approx 0$, only $\frac{1}{4}n$ MSBs are needed in our Theorem 3, which is much better than Theorem 3.3 in [BDF98b] and achieves the bound in Theorem 4.1 in [BDF98b].

3.3 Factoring RSA Modulus with MSBs and LSBs of d

We generalize Lemma 7 in the following lemma, which shows one can recover p with $p \mod e$ and additional MSBs and/or LSBs of p.

Lemma 8. *Given a n-bit RSA modulus $N = pq$ with $q < p < 2q$, let $e = N^{\alpha}$ be a public exponent with $\alpha < \frac{1}{4}$. Suppose $\gamma_1 n$ MSBs, $\gamma_2 n$ LSBs of p and $p \mod e$ are known, then one can find p and factor N when $\gamma_1 + \gamma_2 > \frac{1}{4} - \alpha$.*

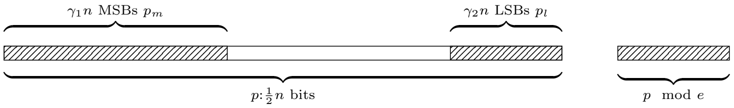

Fig. 6: Illustration of known bits of p and known $p \mod e$ for Lemma 8.

Proof. As in Figure 6, let $p = p_m 2^{\frac{1}{2}n-\gamma_1 n} + \widetilde{p}2^{\gamma_2 n} + p_l$ where $p_m \approx N^{\gamma_1}$ and $p_l \approx N^{\gamma_2}$ are known. Then $p_0 = p_m 2^{\frac{1}{2}n-\gamma_1 n}$ is an approximation of p with $|p - p_0| < N^{\frac{1}{2}-\gamma_1}$.

Since $\overline{p}^{(1)} \equiv p \mod e$ and $\overline{p}^{(2)} \equiv p_l \mod 2^{\gamma_2 n}$ are known, we can compute $\overline{p} \equiv p \mod e2^{\gamma_2 n}$ by Chinese Remainder Theorem.

We can write p as $p = te2^{\gamma_2 n} + \overline{p}$. Denote $t_0 = \frac{p_0 - \overline{p}}{2^{\gamma_2 n}e}$. Then t_0 is an approximation of t with

$$|t - t_0| = |t - \frac{p_0 - \overline{p}}{2^{\gamma_2 n}e}| = |\frac{p - p_0}{2^{\gamma_2 n}e}| < N^{\frac{1}{2}-\gamma_1-\gamma_2-\alpha}.$$

So there exists an unknown integer $\widetilde{t} < N^{\frac{1}{2}-\gamma_1-\gamma_2-\alpha}$ such that $t = t_0 + \widetilde{t}$.

Consider the polynomial $g(x) = (t_0 + x)e + \overline{p}$. Then $\widetilde{t}$ is a root of $g(x) \equiv 0 \mod p$. Since $\gcd(e, N) = 1$, one can compute $e^{-1} \mod N$ and $f(x) = e^{-1}g(x) \mod p$. Note that f is monic, and has the same root as g. Since $\gamma_1 + \gamma_2 > \frac{1}{4} - \alpha$, we have

$$\frac{1}{2} - \gamma_1 - \gamma_2 - \alpha < \frac{1}{4},$$

which yields that $\widetilde{t}$ is a small root of $f(x) = 0 \mod p$. Therefore, we can find $\widetilde{p}$ and then factor N by Lemma 3. □

Based on Theorem 3, we will demonstrate that the known bits after the $(\delta - \frac{1}{2})n$ MSBs of d can be generalized to other positions.

Theorem 4. *Given a n-bit RSA modulus $N = pq$ with $q < p < 2q$ and $p - q = N^{\frac{1}{2}-\theta}$ with $0 < \theta < \frac{1}{4}$, let $e = N^{\alpha}$ be a prime public exponent with $\alpha < \frac{1}{4}$, and $d = N^{\delta}$ be a private exponent satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. Suppose $N \equiv 3 \mod 4$ and $(\delta + \alpha - 1)n$ MSBs of d, $\gamma_1 n$ bits after $(\delta - \frac{1}{2})n$ MSBs of d and $\gamma_2 n$ LSBs of d are known, one can factor N in $\mathcal{O}(e \cdot poly(n))$ if*

$$\gamma_1 + \gamma_2 > \frac{5}{4} + \theta - \alpha - \delta$$

where $poly(n)$ denotes polynomial time in n. That is to say, we only need $(\frac{5}{4} + \theta - \alpha - \delta)n + (\delta + \alpha - 1)n = (\frac{1}{4} + \theta)n$ bits of d totally.

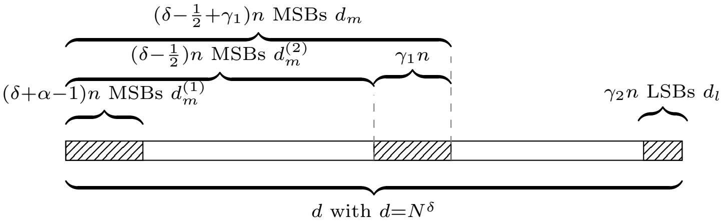
$(\delta-\frac{1}{2}+\gamma_1)n$ MSBs d_m
$(\delta-\frac{1}{2})n$ MSBs $d_m^{(2)}$
$\gamma_1 n$
$(\delta+\alpha-1)n$ MSBs $d_m^{(1)}$
$\gamma_2 n$ LSBs d_l
d with $d=N^{\delta}$

Fig. 7: Illustration of known bits of d for Theorem 4.

Proof. Consider the following equation again

$$ed - k(p-1)(q-1) = 1.$$

We next show how to factor N step by step.

Step 1. Determine the candidate k. Denote by $d_m^{(1)} \approx N^{\delta+\alpha-1}$ the known MSBs of d. Then $d_0^{(1)} = d_m^{(1)} 2^{(1-\alpha)n}$ is an approximation of d with $|d - d_0^{(1)}| < N^{1-\alpha}$. We have $k = \left[\frac{ed_0^{(1)}-1}{N} \right] + k_1 \approx N^{\alpha+\delta-1}$ with $|k_1| \leq 14$ by Lemma 5. Then we can enumerate these candidate k's. Below we assume the correct value of k is known.

Step 2. Recover $(\delta - \frac{1}{2})n$ MSBs of d. Like proof in Theorem 3, we use $d_0^{(2)} = \left[\frac{kN}{e}\right]$ to get an approximation of d with $|d - d_0^{(2)}| < N^{\frac{1}{2}}$. Note $d \approx N^{\delta}$, then $d_0^{(2)} = \left[\frac{kN}{e}\right]$ has the same $(\delta - \frac{1}{2})n$ MSBs of d.

Step 3. Recover the MSBs of p. Now we have $(\delta - \frac{1}{2} + \gamma_1)n$ MSBs, denoted as d_m with $d_m \approx N^{\delta-\frac{1}{2}+\gamma_1}$. Note $d_0 = d_m 2^{(\frac{1}{2}-\gamma_1)n}$ is an approximation of d with $|d - d_0| < N^{\frac{1}{2}-\gamma_1}$. We write d as $d = d_0 + \widetilde{d}$ with $0 \leq \widetilde{d} < N^{\frac{1}{2}-\gamma_1}$.

Define $S = N + 1 - \frac{ed_0-1}{k}$. Then by Equation (1) we have

$$S - (p+q) = \frac{e\widetilde{d}}{k}.$$

Since $0 \leq \widetilde{d} < N^{\frac{1}{2}-\gamma_1}$, we know that

$$0 \leq S - (p+q) < N^{\frac{3}{2}-\gamma_1-\delta}.$$

Define $D = \sqrt{|S^2 - 4N|}$, and $p_0 = \frac{1}{2}(S+D)$. By Lemma 6, we get

$$|p - p_0| < \frac{1}{2}\left(1 + 9N^{\theta}\right) N^{\frac{3}{2}-\gamma_1-\delta}.$$

It shows that $\frac{1}{2}n - (\frac{3}{2} - \gamma_1 - \delta + \theta)n = (\gamma_1 + \delta - \theta - 1)n$ MSBs of p are known.

Step 4. Recover the LSBs of p. Since d_l is known in the following equation:

$$ed_l \equiv 1 + k(N + 1 - p - q) \mod 2^{\gamma_2 n}$$

Like Theorem 3.1 in [BDF98b], it yields $\gamma_2 n$ LSBs of p.

Step 5. Recover $p \mod e$. Consider the following equation again

$$ed = 1 + k(N - (p+q) + 1).$$

Note that $\gcd(k, e) = 1$. Similar to Theorem 7 in [BDF98a], we can compute

$$p + q \equiv N + 1 - k^{-1} \mod e,$$

which yields $s = p + q \mod e$. Then we can find $p \mod e$ by solving Equation (2).

Step 6. Factor N. Now we have $(\gamma_1 + \delta - \theta - 1)n$ MSBs, $\gamma_2 n$ LSBs and p mod e. To apply Lemma 8, we set $(\gamma_1 + \delta - \theta - 1) + \gamma_2 > \frac{1}{4} - \alpha$, that is, when

$$\gamma_1 + \gamma_2 > \frac{5}{4} + \theta - \alpha - \delta.$$

we can factor N in polynomial time due to Lemma 8. □

Remark 1. Note that we have $N \equiv 3 \mod 4$ to solve Step 4 for convenience, just like Theorem 3.1 in [BDF98b]. The $e \log_2(e)$ in algorithm complexity comes from solving the LSBs of p using the LSBs of d. In Step 4, if k is a multiple of a power of 2, additional solutions will be generated. However, unlike Theorem 3.1 in [BDF98b], we now have an approximate value of k, so we do not need to enumerate k from 1 to e. To be honest, we do not achieve an improvement in time complexity because $\log_2(e)$ itself is also in $\mathcal{O}(n)$.

When the factorization of e is known, we have the same result as Corollary 2. Suppose e has r distinct prime factors, one can factor N with time polynomial in $\log_2(N)$ and 2^r under the same known information as in Theorem 4.

4 Experiments

We provide some experiments to verify the correctness of our analysis. The source code for the experiments is open-sourced and available at

https://github.com/fffmath/MSBsOfPrivateKeyAttack.

Our experimental environment is Ubuntu 22.04 (WSL) on a 12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz with SageMath 10.3. We employed the flatter algorithm [RH23] as the lattice basis reduction algorithm in Coppersmith's method.

4.1 Experiments for Theorem 1

For convenience, we assumed k to be known in all experiments; thus, additional time would be needed to enumerate k in a real attack.

For 1024-bit N, we selected different bit-size of e and known MSBs of d for the parameters. The results are presented in Table 3. More precisely, we conducted experiments for 1024-bit N with e of different bit sizes, specifically $e = 2^{256} + 1, 2^{128} + 1, 2^{16} + 1$. For convenience of comparison, we used the same p and q, with $p - q \approx N^{1/2}$.

Here we use $\omega(n)$ to count the number of distinct prime factors. For example, $\omega(2^2 \times 3) = \omega(2 \times 3) = 2$.

Bit-size of e	Known MSBs	Lattice Dim.	$\omega(e)$	Total Time (s)
257	768	3	2	0.64
257	528	30	2	3.48
257	518	100	2	203.91
129	768	3	2	0.12
129	643	200	2	2440.19
17	768	35	1	1.5
17	758	55	1	14.34
17	756	100	1	126.67

Table 3: Experimental results for 1024-bit modulus

We also provided experimental results over a larger RSA modulus, see the following Table 4.

$\log_2 N$	$\log_2 e$	Known MSBs	Lattice Dim.	Time for Factoring e (s)	Total Time (s)
2048	17	1536	75	0.1	50.21
2048	129	1418	75	0.1	391.71
2048	257	1200	75	1.1	406.07
3072	17	2304	75	0.1	245.72
4096	17	3072	75	0.1	385.32

Table 4: Experimental results over a larger RSA modulus

Although we mainly care about small e, especially $e = 65537$, our algorithm has a significant advantage for larger e. As shown in Table 1, the larger our e, the greater the time advantage, because using the algorithm in [BDF98b] requires enumerating the $\log_2 e$ bits. Note that for large e, we must consider the running time of factoring e. Up to now, the best record for factoring is RSA-250 (829 bits) [BGG+20]. As Table 4, for $e < 2^{256}$, it's fast to factor e. Compared to [BDF98b], this will reduce the need to enumerate 256 bits to achieve our new bound!

For cases close to the theoretical bound, we need to consider unavoidable approximation errors, which are usually a few bits. In such cases, appropriate enumeration is necessary. For a 1024-bit N, according to our theory, we will finally solve the equation $x + C \equiv 0 \mod p$, with a bound of $x < N^{\frac{1}{4}} \approx 2^{256}$. However, in practice, due to some approximation errors like $N \neq 2^{1024}$, the bound from Theorem 1 may lead us to a result of 2^{260}. At this point, we need to enumerate four bits to make it less than 2^{256}.

Below are the experimental results presented in Table 5. We consider the theoretical bounds for different bit sizes of e. Due to approximation errors, the actual solutions (the third column in Table 5) are larger than 256 bits, which means it's difficult to use Coppersmith's method directly. In such cases, we choose an appropriate number of bits to enumerate. For example, we choose $752 + \mathbf{6}$ to represent enumerating 6 bits. In other words, we now have 2^6 (or 64) candidate $752 + \mathbf{6}$ MSBs of d. We will run through all candidates until we obtain the factorization of N.

Bit-size of e	Leaked MSBs of d	Solution's bound	Run	Total Time (s)
257	512	260	512+**8**	11744.21
129	640	257	640+**6**	4057.90
17	752	258	752+**6**	1028.72

Table 5: Experimental results for 1024-bit modulus

Based on the above data, we can achieve theoretical bounds practically in real-world attacks.

4.2 Experiments for Lemma 5

As we suppose k is known in Section 4.1, here we provide some experiments for tests of $|k_0 - k|$. In Figure 8, we selected different p and q, and various bit lengths of e. Under the condition of Lemma 5, we plotted the true values of k and the approximated values of k computed using the approximation of d. These are denoted as *True Values* and *Computed Values* respectively. The closer the computed value is to the line $y = x$, the smaller their difference. We plotted error bars to show that the difference is indeed within ±14, satisfying Lemma 5.

5 Conclusion

In this paper, we focused on the partial key exposure attack in the case of small e. Let (N, e) be a public key of the RSA cryptosystem. For the MSBs case, we reduce the number of the leaked bits in d that are needed to mount the attack

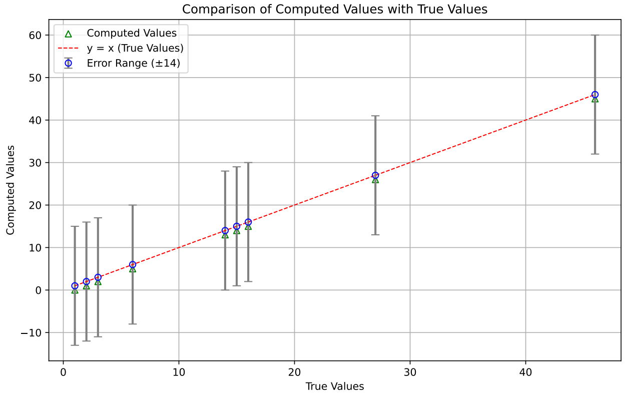

Fig. 8: Comparison between k and the approximation of k.

by $\log_2(e)$ bits compared with previous work. Moreover, we extended our results to both MSBs and LSBs case under the same condition as [BDF98b]. Finally, we provided experimental verification of our ideas and showed that for 1024 bits N, we can achieve the theoretical bound practically.

References

Ajt98. Miklós Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 10–19. ACM, 1998. doi:10.1145/276698.276705.

Aon09. Yoshinori Aono. A new lattice construction for partial key exposure attack for RSA. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*, volume 5443 of *Lecture Notes in Computer Science*, pages 34–53. Springer, 2009. doi:10.1007/978-3-642-00468-1_3.

BD99. Dan Boneh and Glenn Durfee. Cryptanalysis of RSA with private key less than $N^{0.292}$. In *EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 1999. doi:10.1007/3-540-48910-X_1.

BDF98a. Dan Boneh, Glenn Durfee, and Yair Frankel. An attack on RSA given a small fraction of the private key bits. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology - ASIACRYPT '98, International Conference on the Theory and Applications of Cryptology and Information Security, Beijing, China, October 18-22, 1998, Proceedings*, volume 1514 of *Lecture Notes in Computer Science*, pages 25–34. Springer, 1998. doi:10.1007/3-540-49649-1_3.

BDF98b. Dan Boneh, Glenn Durfee, and Yair Frankel. Exposing an RSA private key given a small fraction of its bits, 1998. Full version of the work from Asiacrypt'98, available at `http://crypto.stanford.edu/~dabo/abstracts/bits_of_d.html`.

BGG^{+}20. Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, and Paul Zimmermann. Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment. In *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part II 40*, pages 62–91. Springer, 2020. `doi:10.1007/978-3-030-56880-1_3`.

Ble98. Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1. In *Advances in Cryptology—CRYPTO'98: 18th Annual International Cryptology Conference Santa Barbara, California, USA August 23–27, 1998 Proceedings 18*, pages 1–12. Springer, 1998. `doi:10.1007/BFB0055716`.

BM03. Johannes Blömer and Alexander May. New partial key exposure attacks on RSA. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 27–43. Springer, 2003. `doi:10.1007/978-3-540-45146-4_2`.

CM07. Jean-Sébastien Coron and Alexander May. Deterministic polynomial-time equivalence of computing the RSA secret key and factoring. *J. Cryptol.*, 20(1):39–50, 2007. `doi:10.1007/S00145-006-0433-6`.

Cop96. Don Coppersmith. Finding a small root of a univariate modular equation. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 155–165. Springer, 1996. `doi:10.1007/3-540-68339-9_14`.

Cop97. Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptol.*, 10(4):233–260, 1997. `doi:10.1007/S001459900030`.

dW02. Benne de Weger. Cryptanalysis of RSA with small prime difference. *Appl. Algebra Eng. Commun. Comput.*, 13(1):17–28, 2002. `doi:10.1007/S002000100088`.

EJMdW05. Matthias Ernst, Ellen Jochemsz, Alexander May, and Benne de Weger. Partial key exposure attacks on RSA up to full size exponents. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 371–386. Springer, 2005. `doi:10.1007/11426639_22`.

HM09. Mathias Herrmann and Alexander May. Attacking power generators using unravelled linearization: When do we output too much? In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 487–504. Springer, 2009. doi:10.1007/978-3-642-10366-7_29.

HM10. Mathias Herrmann and Alexander May. Maximizing small root bounds by linearization and applications to small secret exponent RSA. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, pages 53–69. Springer, 2010. doi:10.1007/978-3-642-13013-7_4.

How97. Nick Howgrave-Graham. Finding small roots of univariate modular equations revisited. In Michael Darnell, editor, *Cryptography and Coding, 6th IMA International Conference, Cirencester, UK, December 17-19, 1997, Proceedings*, volume 1355 of *Lecture Notes in Computer Science*, pages 131–142. Springer, 1997. doi:10.1007/BFB0024458.

KSI11. Noboru Kunihiro, Naoyuki Shinohara, and Tetsuya Izu. A unified framework for small secret exponent attack on RSA. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 260–277. Springer, 2011. doi:10.1007/978-3-642-28496-0_16.

LLL82. Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982. doi:10.1007/BF01457454.

LZPL15. Yao Lu, Rui Zhang, Liqiang Peng, and Dongdai Lin. Solving linear equations modulo unknown divisors: revisited. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 189–213. Springer, 2015. doi:10.1007/978-3-662-48797-6_9.

Mac13. Edmond K Machie. *Network security traceback attack and react in the United States Department of Defense network*. Trafford Publishing, 2013.

May03. Alexander May. *New RSA Vulnerabilities Using Lattice Reduction Methods*. PhD thesis, University of Paderborn, 2003. URL: https://d-nb.info/972386416/34.

May04. Alexander May. Computing the RSA secret key is deterministic polynomial time equivalent to factoring. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 213–219. Springer, 2004. doi:10.1007/978-3-540-28628-8_13.

MH24. Gabrielle De Micheli and Nadia Heninger. Survey: Recovering cryptographic keys from partial information, by example. *IACR Communications in Cryptology*, 1(1), 2024. doi:10.62056/ahjbksdja.

Mil75. Gary L. Miller. Riemann's hypothesis and tests for primality. In William C. Rounds, Nancy Martin, Jack W. Carlyle, and Michael A. Harrison, editors, *Proceedings of the 7th Annual ACM Symposium on Theory of Computing, May 5-7, 1975, Albuquerque, New Mexico, USA*, pages 234–239. ACM, 1975. doi:10.1145/800116.803773.

MN23. Jonas Meers and Julian Nowakowski. Solving the Hidden Number Problem for CSIDH and CSURF via Automated Coppersmith. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023,* *Proceedings, Part IV*, volume 14441 of *Lecture Notes in Computer Science*, pages 39–71. Springer, 2023. doi:10.1007/978-981-99-8730-6_2.

MNS21. Alexander May, Julian Nowakowski, and Santanu Sarkar. Partial key exposure attack on short secret exponent CRT-RSA. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 99–129. Springer, 2021. doi:10.1007/978-3-030-92062-3_4.

MNS22. Alexander May, Julian Nowakowski, and Santanu Sarkar. Approximate divisor multiples–factoring with only a third of the secret CRT-exponents. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 147–167. Springer, 2022. doi:10.1007/978-3-031-07082-2_6.

MR09. Alexander May and Maike Ritzenhofen. Implicit factoring: On polynomial time factoring given only an implicit hint. In *PKC 2009*, volume 5443 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2009. doi:10.1007/978-3-642-00468-1_1.

RH23. Keegan Ryan and Nadia Heninger. Fast practical lattice reduction through iterated compression. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 3–36. Springer, 2023. doi:10.1007/978-3-031-38548-3_1.

RSA78. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. doi:10.1145/359340.359342.

STK20. Kaichi Suzuki, Atsushi Takayasu, and Noboru Kunihiro. Extended partial key exposure attacks on RSA: Improvement up to full size decryption exponents. *Theor. Comput. Sci.*, 841:62–83, 2020. URL: https://doi.org/10.1016/j.tcs.2020.07.004, doi:10.1016/J.TCS.2020.07.004.

TK16. Atsushi Takayasu and Noboru Kunihiro. How to generalize RSA cryptanalyses. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part II*, volume 9615 of *Lecture Notes in Computer Science*, pages 67–97. Springer, 2016. doi:10.1007/978-3-662-49387-8_4.

TK19. Atsushi Takayasu and Noboru Kunihiro. Partial key exposure attacks on RSA: achieving the Boneh-Durfee bound. *Theor. Comput. Sci.*, 761:51–77, 2019. URL: https://doi.org/10.1016/j.tcs.2018.08.021, doi:10.1016/J.TCS.2018.08.021.

Wie90. Michael J. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Trans. Inf. Theory*, 36(3):553–558, 1990. doi:10.1109/18.54902.

ZvdPYS22. Yuanyuan Zhou, Joop van de Pol, Yu Yu, and François-Xavier Standaert. A third is all you need: Extended partial key exposure attack on CRT-RSA with additive exponent blinding. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV*, volume 13794 of *Lecture Notes in Computer Science*, pages 508–536. Springer, 2022. doi:10.1007/978-3-031-22972-5_18.