# Improved Factoring Attacks on Multi-prime RSA with Small Prime Difference

Mengce Zheng[1,2(✉)], Noboru Kunihiro[2], and Honggang Hu[1]

[1] Key Laboratory of Electromagnetic Space Information, CAS, University of Science and Technology of China, Hefei, China
hghu2005@ustc.edu.cn

[2] The University of Tokyo, Tokyo, Japan
zheng@it.k.u-tokyo.ac.jp, kunihiro@k.u-tokyo.ac.jp

**Abstract.** In this paper, we study the security of multi-prime RSA with small prime difference and propose two improved factoring attacks. The modulus involved in this variant is the product of $r$ distinct prime factors of same bit-size. Zhang and Takagi (ACISP 2013) showed a Fermat-like factoring attack on multi-prime RSA. In order to improve the previous result, we gather more information about the prime factors to derive $r$ simultaneous modular equations. The first attack is based on combining $r$ equations to solve one multivariate modular equation by a generic lattice approach. Since the equation form is similar to multi-prime $\Phi$-hiding problem, we propose the second attack by applying the optimal linearization technique. We also show that our attacks can achieve better bounds in the experiments.

**Keywords:** Cryptanalysis · Multi-prime RSA · Small prime difference · Factoring attack · Lattice · Linearization technique

## 1 Introduction

### 1.1 Background

RSA [20] is a famous public key cryptosystem that has been widely used in various settings. However, the original RSA is not fit for some constrained environments. Since people need faster and more efficient RSA encryption/decryption processes, several variants have been proposed and surveyed [3]. In this paper, we focus on a variant called multi-prime RSA. It is described as follows.

**Key Generation.** Generate $r$ distinct primes $p_1, p_2, \ldots, p_r$ of same bit-size and modulus $N = \prod_{i=1}^{r} p_i$. Pick a random number that is coprime to $\varphi(N) = \prod_{i=1}^{r}(p_i - 1)$ as the public key $e$ and compute the corresponding private key $d = e^{-1} \bmod \varphi(N)$.

**Encryption.** Transform the message string into an integer $M \in \mathbb{Z}_N$ and compute the ciphertext as $C = M^e \bmod N$.


J. Pieprzyk and S. Suriadi (Eds.): ACISP 2017, Part I, LNCS 10342, pp. 324–342, 2017.
DOI: 10.1007/978-3-319-60055-0_17

**Decryption.** Compute $M_i = C^{d_i} \bmod p_i$ for $d_i = d \bmod (p_i - 1)$, $1 \leq i \leq r$. Combine $M_i$'s by the Chinese Remainder Theorem to obtain the plaintext $M = C^d \bmod N$.

This variant modifies the modulus to $N = p_1 p_2 \cdots p_r$ for $r \geq 3$. It was patented by Compaq [5], using a modulus of the form $N = p_1 p_2 p_3$. We then discuss the performance of multi-prime RSA. The advantage is the efficiency when using Chinese Remainder Theorem in its decryption process. From [3], we know that the asymptotic speedup over the standard RSA is approximately $\frac{r^2}{4}$. Moreover, ordinary attacks such as small private exponent attack and partial key exposure attack are less effective as $r$ increases. But $r$ should not be unrestrictedly large because of the Elliptic Curve Method [18]. Since factoring a multi-prime RSA modulus using ECM is much easier with increasing $r$, one might choose $r = 3$, 4 and 5 for most settings. Generally speaking, multi-prime RSA with appropriate $r$ might be a practical alternative for reducing the decryption costs.

Without loss of generality, we have $p_1 < p_2 < \cdots < p_r$ and $\frac{1}{2}N^{\frac{1}{r}} < p_1 < N^{\frac{1}{r}} < p_r < 2N^{\frac{1}{r}}$. The second one indicates that the prime factors are balanced, which means that they are roughly of same bit-size. The prime difference $\Delta$ is defined as $\Delta := \max_{i \neq j} |p_i - p_j| = p_r - p_1 = N^{\gamma}$ for $0 < \gamma < \frac{1}{r}$. The security of multi-prime RSA has been investigated for small private exponent [4,13,14] and for small prime difference [1,22,25,26].

Prime difference was introduced by de Weger [11] to show that one can find an enhanced small private exponent attack with small prime difference. As for multi-prime RSA, it is also applied to obtain some improvements. Thereafter we review some related previous attacks. Suppose that $N$ is a multi-prime RSA modulus with $r$ prime factors. Let $e \approx N$ be a valid public key and $d = N^{\delta}$ be its corresponding private key.

**Bahig-Bhery-Nassr** [1]**.** Given the prime difference $\Delta = N^{\gamma}$ and the public key $(N, e)$, then multi-prime RSA is insecure if $\gamma$ and $d$ satisfy

$$2d^2 + 1 < \frac{N^{\frac{2}{r} - \gamma}}{6r}.$$

**Zhang-Takagi** [25,26]**.** Given the prime difference $\Delta = N^{\gamma}$ and the public key $(N, e)$, then $d$ can be probabilistically found in time polynomial in $\log N$ if $\gamma$ and $\delta$ satisfy

$$\delta < 1 - \sqrt{1 + \gamma - \frac{2}{r}}.$$

The bound was later refined to

$$\delta < 1 - \sqrt{1 + 2\gamma - \frac{3}{r}} \text{ for } \gamma \geq \frac{3}{2r} - \frac{1 + \delta}{4},$$
$$\delta < \frac{3}{r} - \frac{1}{4} - 2\gamma \text{ for } \gamma < \frac{3}{2r} - \frac{1 + \delta}{4}.$$

They also presented a Fermat-like factoring attack for

$$\gamma < \frac{1}{r^2}.$$

**Takayasu-Kunihiro** [22]**.** Given the prime difference $\Delta = N^\gamma$ and the public key $(N, e)$, then $d$ can be probabilistically found in time polynomial in $\log N$ if $\gamma$ and $\delta$ satisfy

$$\delta < 1 - \sqrt{1 + 2\gamma - \frac{3}{r}} \text{ for } \frac{3}{2}(\frac{1}{r} - \frac{1}{4}) \le \gamma < \frac{1}{r},$$
$$\delta < 1 - \frac{2}{3}(\sqrt{(7 + 8\gamma - \frac{12}{r})(1 + 2\gamma - \frac{3}{r})} - 1 - 2\gamma + \frac{3}{r}) \text{ for } \gamma < \frac{3}{2}(\frac{1}{r} - \frac{1}{4}).$$

Notice that the condition $\frac{3}{2r} - \frac{1+\delta}{4}$ in Zhang-Takagi attack degenerates to $-\frac{\delta}{4}$ for $r = 6$, and the condition $\frac{3}{2}(\frac{1}{r} - \frac{1}{4})$ in Takayasu-Kunihiro attack degenerates to 0 for $r = 4$. Thus, Zhang-Takagi attack and Takayasu-Kunihiro attack depend on $\delta$ with $\gamma < \frac{1}{r}$ for larger $r$. In such cases, factoring attacks with quite small $\gamma$ are much more effective without any restriction on $\delta$. The distinction is the dependence on the private exponent and this is also the advantage of factoring attacks.

### 1.2 Our Contribution

In this paper, we aim to factor the multi-prime RSA modulus with small prime difference. More concretely, $N$ can be factored in polynomial time under which condition when given the multi-prime RSA modulus $N$ that is the product of $r$ distinct primes and its prime difference $N^\gamma$.

Let $x_i = p_i - p$ for $i = 1, 2, \dots, r$ with $|x_i| = |p_i - p| < p_r - p_1 = N^\gamma$ for $p = [N^{\frac{1}{r}}]$. At ACISP 2013, Zhang and Takagi [25] solved $x_i$ from each equation and computed prime factors by $p_i = x_i + p$. In our opinion, they only made use of partial information about prime factors with prime difference. In contrast, we transform the knowledge of all balanced prime factors with prime difference into the following modular equations.

$$\begin{cases} x_1 + p = 0 \bmod p_1, \\ x_2 + p = 0 \bmod p_2, \\ \qquad \vdots \\ x_r + p = 0 \bmod p_r. \end{cases}$$

Our factoring problem is somewhat similar to multi-prime $\Phi$-hiding problem introduced by Kiltz et al. [16] because of the modular equation form. The definition of multi-prime $\Phi$-hiding problem is given. Let $N = p_1 \cdots p_r$ be a composite integer (of unknown factorization) with $r$ distinct prime factors of same bit-size. Given $N$ and a prime $e$, decide whether $e$ divides $p_i$ for $1 \le i \le r - 1$ or not.

In order to solve multi-prime $\Phi$-hiding problem, one can try to solve the following simultaneous equations and then conclude that $e$ is $\Phi$-hidden in $N$ or not.

$$\begin{cases} ex_1 + 1 = 0 \bmod p_1, \\ ex_2 + 1 = 0 \bmod p_2, \\ \quad\vdots \\ ex_{r-1} + 1 = 0 \bmod p_{r-1}. \end{cases}$$

There exist some differences between these two problems. In $\Phi$-hiding problem, since it is not necessary to know the exact values of the unknowns but enough to know if the equations can be solved, one can perform a linearization on the product $\prod_{i=1}^{r-1}(ex_i+1)$ and then decide if $\prod_{i=1}^{r-1}(ex_i+1) = 0 \bmod p_1p_2\cdots p_{r-1}$ can be solved. Thus, it is like a "decision"-form problem. Our factoring problem is like a "search"-form one because we must extract the value of every unknown variable. In our optimized method, we can transform the factoring problem into a "decision"-form problem and then apply the optimal linearization technique.

Another difference is that we do not have $ex_r + 1 = 0 \bmod p_r$ in $\Phi$-hiding problem. This special feature can be applied to improve the bound [24]. However we can not directly use the same technique to solve the factoring problem.

Our improvements are based on two ideas. The first one is a direct method by gathering all the equations together to solve an $r$-variate modular equation. The drawback of this method is that the running time is exponential in $r$. So we provide an optimized method by combining fewer equations. Inspired by Tosu and Kunihiro [23], we can benefit from the optimal linearization technique with fewer unknowns and less cost. Thus, we will obtain a great speedup and efficient performance in the practical implementation.

We show that multi-prime RSA modulus with small prime difference can be efficiently factored in the following cases due to various $r$'s.

- For $r \leq 6$, we have
$$\gamma < \frac{2}{r(r+1)}.$$
- For $r \geq 7$ and an optimal $l$, we have
$$\gamma < \frac{2}{l+1}\left(\frac{1}{r}\right)^{\frac{l+1}{l}}.$$
- For much larger $r$ and the base of natural logarithm e, we have
$$\gamma < \frac{2}{\mathrm{e}r(\log r + 1)}.$$

## 2 Preliminaries

### 2.1 Lattice Based Method

We briefly introduce lattice based method including the LLL algorithm [17], Coppersmith's technique [6–8], Howgrave-Graham's lemma [15] and Coron's reformulation [9,10].

The technique is to construct a set of polynomials modulo $R$ sharing the common roots and then reduce them to the equations over the integers. After transforming known parameters into constructed polynomials' coefficients that form a lattice basis matrix with dimension $w$. One can compute some short lattice vectors whose norm is expected to be sufficiently small by the LLL algorithm. Eventually, one can solve the desired roots. The LLL algorithm proposed by Lenstra, Lenstra and Lovász [17] is practically used for finding approximately small lattice vectors.

**Lemma 1.** *Let $\mathcal{L}$ be a lattice with determinant $\det(\mathcal{L})$. The LLL algorithm outputs a reduced basis $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_w)$ in polynomial time, and for $1 \le i \le w$, the reduced basis vectors satisfy*

$$\|\boldsymbol{v}_1\|, \|\boldsymbol{v}_2\|, \ldots, \|\boldsymbol{v}_i\| \le 2^{\frac{w(w-1)}{4(w+1-i)}} \det(\mathcal{L})^{\frac{1}{w+1-i}}.$$

The following lemma presented by Howgrave-Graham [15] helps us to judge whether the roots of a modular equation are also roots over the integers. To a given polynomial $g(x_1, \ldots, x_n) = \sum a_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}$, its norm is defined as $\|g(x_1, \ldots, x_n)\|^2 := \sum |a_{i_1,\ldots,i_n}|^2$.

**Lemma 2.** *Let $g(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be an integer polynomial that is a sum of at most $m$ monomials. Suppose that*

1. $\|g(x_1X_1, \ldots, x_nX_n)\| \le \frac{R}{\sqrt{m}}$,
2. $g(x_1^{(0)}, \ldots, x_n^{(0)}) = 0 \bmod R$ *for* $|x_1^{(0)}| \le X_1, \ldots, |x_n^{(0)}| \le X_n$.

*Then we have $g(x_1^{(0)}, \ldots, x_n^{(0)}) = 0$ over the integers.*

The above fundamental lemmas give us the final condition, which is roughly $\det(\mathcal{L}) < R^w$. Some RSA cryptanalytic applications [2,8,12] are derived from such lattice based method. But Boneh and Durfee [2] have noted that solving multivariate equations is heuristic because the polynomials derived from lattice reduction algorithms are not guaranteed to be algebraically independent. In order to extract the exact roots in practice, we rely on the following assumption.

**Assumption 1.** *The polynomials derived from the LLL algorithm in lattice based method are algebraically independent. Furthermore, the solution can be efficiently found by Gröbner basis computations.*

Our improved attacks can be reduced to solving multivariate linear equations that was studied by Herrmann and May [12].

**Lemma 3.** *Let $\epsilon > 0$ and let $N$ be a sufficiently large composite integer (of unknown factorization) with a divisor $p \geq N^{\beta}$. Furthermore, let $f(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be a linear polynomial in $n$ variables. Under Assumption 1, we can find solutions $(x_1^{(0)}, \ldots, x_n^{(0)})$ of the equation $f(x_1, \ldots, x_n) = 0 \bmod p$ with $|x_1^{(0)}| \leq N^{\eta_1}, \ldots, |x_n^{(0)}| \leq N^{\eta_n}$ if*

$$\sum_{i=1}^{n} \eta_i \leq 1 - (n+1)(1-\beta) + n(1-\beta)^{\frac{n+1}{n}} - \epsilon.$$

*The time complexity is polynomial in $\log N$ and $(\mathrm{e}/\epsilon)^n$.*

The lattice based algorithm was later improved by Lu et al. [19] and Takayasu and Kunihiro [21]. Since the cryptanalysis is based on approximations, we neglect the lower order terms and remove $\epsilon$ in our methods for simplicity.

### 2.2 Some Notations

We introduce the following notations for our methods.

- $p$ denotes the value of rounding $N^{\frac{1}{r}}$ to the nearest integer and it is mentioned above as $p = [N^{\frac{1}{r}}]$.
- $\sigma_i^k$ denotes the elementary symmetric polynomial in $k$ variables $y_1, \ldots, y_k$ of degree $i$ and it is defined by $\sigma_i^k := \sum_{\lambda \subset \{1,2,\ldots,k\}, |\lambda|=i} \left(\prod_{j\in\lambda} y_j\right)$.
- $Q_k$ denotes the product of $k$ prime factors that are chosen from $p_1, p_2, \ldots, p_r$ and hence $Q_k$ is a divisor of $N$.
- $Q_k'$ denotes the numerical value of the left side after solving the equation and hence $Q_k'$ is a multiple of $Q_k$.

## 3 Improved Factoring Attacks

### 3.1 The Direct Method

As mentioned before, we gather all the equations together to solve an $r$-variate modular equation. More concretely, we present the following factoring attack.

**Proposition 1.** *Let $N = p_1 \cdots p_r$ be a multi-prime RSA modulus for $p_1 < \cdots < p_r$ and $p_r - p_1 = N^{\gamma}$ for $0 < \gamma < \frac{1}{r}$. Then under Assumption 1, $N$ can be factored in time polynomial in $\log N$ but exponential in $r$ if*

$$\gamma < \frac{2}{r(r+1)}.$$

Our approach utilizes the equation form of multi-prime $\Phi$-hiding problem. Let $e$ be the inverse of $p$ modulo $N$, namely $e = p^{-1} \bmod N$. Then $y_i + p = 0 \bmod p_i$ can be rewritten as $ey_i + 1 = 0 \bmod p_i$ and we obtain

$$\begin{cases} ey_1 + 1 = 0 \bmod p_1, \\ \qquad\vdots \\ ey_r + 1 = 0 \bmod p_r. \end{cases}$$

Combining all equations together gives us

$$\prod_{i=1}^{r}(ey_i+1)=\sum_{i=1}^{r}e^i\sigma_i^r+1=0 \bmod N.$$

We have $e=p^{-1} \bmod N$ that is equivalent to $ep=1 \bmod N$. It can be reduced to $\sum_{i=1}^{r}e^i\sigma_i^r+ep=0 \bmod N$ and further

$$\sum_{i=1}^{r}e^{i-1}\sigma_i^r+p=0 \bmod N.$$

Regarding each $\sigma_i^r$ as a new variable makes $\sum_{i=1}^{r}e^{i-1}\sigma_i^r+p$ a linear equation. We then figure out each $\eta_i$ of $|\sigma_i^r|<N^{\eta_i}$ for $i=1,\ldots,r$ and apply Lemma 3 with $\beta=1$. It is not hard to know that $\eta_i=i\gamma$ for $1\le i\le r$. Thus, the final condition is $\sum_{i=1}^{r}i\gamma<1$, which can be simplified to

$$\gamma<\frac{2}{r(r+1)}.$$

After solving the linear equation, we obtain the values of $\sigma_1^r,\ldots,\sigma_r^r$. Then we extract $x_1,\ldots,x_r$ by solving $x^r-\sigma_1^r x^{r-1}+\cdots+(-1)^r\sigma_r^r=0$ over the integers. Finally, we compute the prime factors $p_1,\ldots,p_r$ for $p_i=x_i+p$. The full description of the algorithm is given in Appendix A.1.

The running time depends on reducing the basis matrix and extracting the common roots. The LLL algorithm can output the desired polynomials in time polynomial in $\log N$ but exponential in $r$. This may be a drawback due to large $r$ and forces us to find more efficient method. The Gröbner basis computation for finding the common roots is usually polynomial time in practice. Additionally, one can obtain more polynomials derived from the LLL algorithm and hence the Gröbner basis computation is suggested rather than resultant computation.

### 3.2 The Optimized Method

As described in the direct method, we still solve the factoring problem in the view of a "search"-form problem. Its drawback is that the time complexity is exponential in $r$. Consequently, the factoring attack becomes less efficient for larger $r$.

When considering taking fewer equations to form one modular equation, we have some interesting observations. We randomly choose $k$ $(2\le k\le r-1)$ equations and obtain a new equation $F(y_1,\ldots,y_k)=0 \bmod Q_k$. Fortunately, it is enough to know the numerical value $Q_k'$ of the left side and not necessary to know exact values of $y_1,\ldots,y_k$. Then, computing the greatest common divisor $\gcd(Q_k',N)$ gives us all combinations of $k$ prime factors that indicate every prime factor.

In fact, the factoring problem is refined to become of "decision"-form. Thus, we can employ the optimal linearization similar to the technique proposed by

Tosu and Kunihiro [23] when solving multi-prime $\Phi$-hiding problem. The idea is to examine all possible linearization cases to find the optimal setting when it can be efficiently solved. We present the optimized factoring attack below.

**Proposition 2.** *Let $N = p_1 \cdots p_r$ be a multi-prime RSA modulus for $p_1 < \cdots < p_r$ and $p_r - p_1 = N^\gamma$ for $0 < \gamma < \frac{1}{r}$. Then under Assumption 1, $N$ can be factored in time polynomial in $\log N$ with an optimal $l$ if*

$$\gamma < \frac{2}{l+1}\left(\frac{1}{r}\right)^{\frac{l+1}{l}}.$$

We consider combining $k$ equations and performing a linearization of $l$ ($2 \le l \le k$) variables. Note that the parameters $k$ and $l$ need to be decided later. First, we have $(y_1 + p)(y_2 + p) \cdots (y_k + p) = 0 \bmod Q_k$. It can be rewritten as $\sum_{i=0}^{k} p^{k-i}\sigma_i^k = 0 \bmod Q_k$. The expansion is

$$\sigma_k^k + p\sigma_{k-1}^k + p^2\sigma_{k-2}^k + \cdots + p^k = 0 \bmod Q_k.$$

Then, we apply a linearization for the case of $l$ variables. Let $t_1, \ldots, t_{l+1}$ be the integers satisfying $t_1 = k > t_2 > \cdots > t_{l+1} = 0$. We obtain

$$p^{k-t_1}u_1 + p^{k-t_2}u_2 + \cdots + p^{k-t_l}u_l + p^k = 0 \bmod Q_k,$$

where $u_i := \sum_{j=t_{i+1}+1}^{t_i} p^{t_i-j}\sigma_j^k$ for $1 \le i \le l$. For $|y_i| < N^\gamma$, $p \approx N^{\frac{1}{r}}$ and $\gamma < \frac{1}{r}$, we know that the bound is $|u_i| < N^{\frac{t_i - t_{i+1}-1}{r} + (t_{i+1}+1)\gamma}$. In other words, we have

$$\eta_i = \frac{t_i - t_{i+1} - 1}{r} + (t_{i+1}+1)\gamma.$$

Thus, we can find the roots of the linear equation by Lemma 3 with $\beta = \frac{k}{r}$ and above $\eta_i$ if $\sum_{i=1}^{n}\eta_i < 1 - (l+1)(1-\beta) + l(1-\beta)^{\frac{l+1}{l}}$.

Then we have

$$\gamma < \frac{l \cdot \left(\frac{k+1}{r} + (1-\frac{k}{r})^{\frac{l+1}{l}} - 1\right)}{l + \sum_{i=2}^{l} t_i}.$$

The above bound reaches its maximum by setting $(t_1, t_2, t_3, \ldots, t_l)$ to be $(k, l-1, l-2\ldots, 1)$. The condition now is

$$\gamma < \frac{2}{l+1}\left(\frac{k+1}{r} + (1-\frac{k}{r})^{\frac{l+1}{l}} - 1\right).$$

We can further optimize $k$ to obtain the best bound on $\gamma$ by calculating the derivative on $k$. It can be verified that $k = r-1$ is the most suitable choice. Thus, we derive the condition

$$\gamma < \frac{2}{l+1}\left(\frac{1}{r}\right)^{\frac{l+1}{l}}.$$

It means that we need to solve

$$u_1 + p^{r-l}u_2 + \cdots + p^{r-2}u_l + p^{r-1} = 0 \bmod Q_{r-1}.$$

The optimal value of $l$ can be discovered by numerical computation. For each positive integer $r \le 10$, the optimal cases are $l = 2$ for $r = 3, 4, 5$, and $l = 3$ for $r = 6, 7, 8, 9, 10$. To be specific, we show the final equations need to be solved in our optimized method as follows.

- For $r = 3, 4, 5$, that is

$$u_1 + p^{r-2}u_2 + p^{r-1} = 0 \bmod Q_{r-1}.$$

- For $r = 6, 7, 8, 9, 10$, that is

$$u_1 + p^{r-3}u_2 + p^{r-2}u_3 + p^{r-1} = 0 \bmod Q_{r-1}.$$

As analyzed in [23], we set $l \approx \log r$ for much larger $r$ and the condition is approximated

$$\gamma < \frac{2}{\mathrm{e}r(\log r + 1)},$$

where e is the base of natural logarithm. Therefore, we also present the factoring attack for much larger $r$.

**Proposition 3.** *Let $N = p_1 \cdots p_r$ be a multi-prime RSA modulus for $p_1 < \cdots < p_r$ and $p_r - p_1 = N^\gamma$ for $0 < \gamma < \frac{1}{r}$. Then under Assumption 1, $N$ can be factored in time polynomial in $\log N$ for much larger $r$ if*

$$\gamma < \frac{2}{\mathrm{e}r(\log r + 1)}.$$

After solving the modular equation, we obtain the values of $u_1, \ldots, u_l$. Then we know all combinations of $r-1$ prime factors by $\gcd(Q'_{r-1}, N)$. Finally, we compute each prime factor by $\frac{N}{\gcd(Q'_{r-1}, N)}$.

Note that we can find all prime factors by solving the linear equation once because every combination (or product) of $r-1$ prime factors is equivalent to each other. Using $l \approx \log r$ implies that our method works in time polynomial in $\log N$ and $r$.

### 3.3 Discussions

Compare with the direct method, we have two improvements in our optimized method. Firstly, we decrease the number of unknown variables and significantly improve the practical performance for larger $r$. Secondly, we can achieve a better bound for much larger $r$ at the same time. But for $r \le 6$, the direct method offers a higher bound and hence the factoring attack is still in polynomial time.

Note that the unknown variables $u_i$'s in the optimized method are quite unbalanced. So we make further improvement by applying better lattice constructions proposed by Takayasu and Kunihiro [21]. Here we omit the complicated analysis and show another advantage. For $r \leq 10$, the optimal $l$ is always 2. It means that the final equation we need to solve in the optimized method is $u_1 + p^{r-2}u_2 + p^{r-1} = 0 \bmod Q_{r-1}$. Thus, we further reduce the running time of the optimized factoring attack. The full description of the optimized algorithm and detailed lattice construction are given in Appendix A.2.

Table 1 shows the comparison of upper bound on $\gamma$ due to above factoring attacks for $r \leq 10$. The fourth column provides the results using better lattice construction that is discussed above. It is visible that our methods are superior.

**Table 1.** The comparison of upper bound on $\gamma$ due to above factoring attacks

| $r$ | Section 3.1 | Section 3.2 | Section 3.3 | Zhang-Takagi [25] |
|---|---|---|---|---|
| 3 | **0.1666** | 0.1283 | — | 0.1111 |
| 4 | **0.1000** | 0.0833 | 0.0835 | 0.0625 |
| 5 | **0.0666** | 0.0596 | 0.0608 | 0.0400 |
| 6 | **0.0476** | 0.0458 | 0.0474 | 0.0277 |
| 7 | 0.0357 | 0.0373 | **0.0387** | 0.0204 |
| 8 | 0.0277 | 0.0312 | **0.0327** | 0.0156 |
| 9 | 0.0222 | 0.0267 | **0.0282** | 0.0123 |
| 10 | 0.0181 | 0.0232 | **0.0248** | 0.0100 |

### 3.4 Experimental Results

We now state some experimental results to show the practical performance of our methods. These experiments are carried out under Sage 7.3 running on a laptop with Intel Core i7 CPU 2.70 GHz and 8 GB RAM. The numbers we used are chosen uniformly at random and Assumption 1 is found to hold for the experiments.

During the experiments, we collect many polynomials satisfying our requirements. In other words, we obtain enough sufficiently short vectors after running the LLL algorithm. Hence, we extract the common roots by Gröbner basis computation and finally attain the factorization of multi-prime RSA modulus.

We provide the experimental results on two attacks according to Sects. 3.1 and 3.2 (actually refined by Sect. 3.3), namely the $\gamma_{e1}$-column and $\gamma_{e2}$-column, respectively. The $\gamma_{zt}$-column provides the experimental bound of Zhang-Takagi method. The results about the comparison are showed in Table 2.

We firstly comment the experiments for $r = 3$. We reduce a 220-dimensional lattice for the direct method while we use a lattice whose dimension is 300 for the optimized method. A 1536-bit multi-prime RSA modulus can be successfully factored by a 174-bit prime difference by the direct method. While using the

**Table 2.** The experimental results of upper bound on $\gamma$

| $r$ | $\gamma_{zt}$ | $\gamma_{e1}$ | $\gamma_{e2}$ |
|---|---|---|---|
| 3 | 0.1109 | 0.1132 | 0.1120 |
| 4 | 0.0620 | — | 0.0750 |
| 5 | 0.0396 | — | 0.0533 |
| 6 | 0.0275 | — | 0.0337 |
| 7 | 0.0202 | — | 0.0286 |

optimized method, a 172-bit difference leads to the factorization of a 1536-bit modulus. Thus, we conclude that the direct method performs better for $r = 3$ with roughly similar lattice setting. On the other hand, we observe that the optimized method runs much faster, which is predicted above.

For $4 \leq r \leq 7$, we use the optimized method with lattices whose dimension is around 300 since it is more efficient. We carry out experiments for much smaller moduli with almost the same lattice setting and they work much better. We also do experiments for moduli of the same size with various lattice dimensions for $r = 3, 4$. The results become better as the lattice dimension increases. So the lattice dimension may be a critical limitation that influences the practical performance of lattice based methods. The optimized bounds for $4 \leq r \leq 7$ showed in the $\gamma_{e2}$-column are those observed in the experiments with much smaller moduli. More details are given in Appendix B.

## 4 Conclusions

Factoring attack works better than small private exponent attack on multi-prime RSA with much smaller prime difference, and the former removes the restriction on the private exponents. We further upgrade the insecure bound on the prime difference and propose improved factoring attacks based on lattice approach and the optimal linearization technique.

To summarize, our factoring attacks make significant improvements by taking full knowledge of the small prime difference. We combine more equations rather than only one equation to solve the factoring problem. Furthermore, applying the optimal linearization technique on unknown variables helps us to reduce the time cost and obtain better results.

For our factoring attacks on multi-prime RSA modulus with $r$ primes, solving an $r$-variate linear equation constructed by $r$ simultaneous modular equations is preferred for $r \leq 6$. And solving an $l$-variate ($l$ depends on $r$) linear equation constructed by $r - 1$ equations is preferred for $r \geq 7$. Both factoring attacks can be done in polynomial time.

**Acknowledgments.** The first author is supported by China Scholarship Council Grant No. 201606340061. This research was partially supported by JST CREST Grant Number JPMJCR14D6, Japan and JSPS KAKENHI Grant Number 16H02780, and National Natural Science Foundation of China (Grant Nos. 61522210, 61632013), 100 Talents Program of Chinese Academy of Sciences, and the Fundamental Research Funds for the Central Universities in China (Grant No. WK2101020005).

# A Algorithms

## A.1 The Direct Method

**Algorithm 1.** The direct method (Sect. 3.1)

**Input:** Multi-prime RSA modulus $N$ with $r$ and small prime difference $N^{\gamma}$.
**Output:** The factorization $N = p_1 \cdots p_r$.

1: Compute $p = [N^{\frac{1}{r}}]$ and $e = p^{-1} \bmod N$.
2: Construct the linear modular equation with unknown variables $\sigma_i^r$:

$$\sigma_1^r + e^1\sigma_2^r + \cdots + e^{r-1}\sigma_r^r + p = 0 \bmod N.$$

3: Figure out $\eta_i$'s that are related to the bounds $N^{\eta_i}$ on $\sigma_i^r$ for $1 \le i \le r$:

$$|\sigma_i^r| < N^{i\gamma}.$$

4: Extract each $\sigma_i^r$ by applying Lemma 3.
5: Solve $x^r - \sigma_1^r x^{r-1} + \cdots + (-1)^r\sigma_r^r = 0$ over the integers.
6: Set $p_i = p + x_i$ in increasing order with roots $x_i$ for $1 \le i \le r$.

## A.2 The Optimized Method

In Takayasu-Kunihiro lattice construction, we carefully work out the selection of polynomials by considering the sizes of root bounds. For example, we deal with $u_1 + p^{r-2}u_2 + p^{r-1} = 0 \bmod Q_{r-1}$ in our optimized method. We use $u_2^{i_2}(u_1 + p^{r-2}u_2 + p^{r-1})^{i_1}N^{\max\{t-i_1,0\}}$ as the shift polynomials with positive integers $m$ and $t$ that will be optimized later. The indexes $i_1$ and $i_2$ satisfy $0 \le i_1 + i_2 \le m$ and $0 \le \gamma_1 i_1 + \gamma_2 i_2 \le \beta t$ in order to select as many helpful polynomials as possible and to let the basis matrix be triangular.

Thus, the shift polynomials modulo $p^t$ have the common roots for $u_1$ and $u_2$. We span a lattice by the coefficient vectors of above shift polynomials and the equations are derived from the reduced LLL basis vectors. The small roots can be easily recovered by Gröbner basis computation.

**Algorithm 2.** The optimized method (Sect. 3.2)

**Input:** Multi-prime RSA modulus $N$ with $r$ and small prime difference $N^{\gamma}$.
**Output:** The factorization $N = p_1 \cdots p_r$.

1: Compute $p = [N^{\frac{1}{r}}]$.
2: Choose an optimal $l$ according to $r$.
3: Construct the linear modular equation with unknown variables $u_i$:

$$u_1 + p^{r-l}u_2 + \cdots + p^{r-2}u_l + p^{r-1} = 0 \bmod Q_{r-1}.$$

4: Figure out $\eta_i$'s that are related to the bounds $N^{\eta_i}$ on $\sigma_i^r$ for $1 \leq i \leq l$ with known $(t_1, t_2, t_3, \ldots, t_l, t_{l+1}) = (r-1, l-1, l-2 \ldots, 1, 0)$:

$$|u_i| < N^{\frac{t_i - t_{i+1} - 1}{r} + (t_{i+1}+1)\gamma}.$$

5: Extract each $u_i$ by using Takayasu-Kunihiro lattice construction.
6: Compute $Q'_{r-1} = u_1 + p^{r-l}u_2 + \cdots + p^{r-2}u_l + p^{r-1}$ with roots $\{u_1, \ldots, u_l\}$.
7: Set $p_i = N/\gcd(Q'_{r-1}, N)$ in increasing order for $1 \leq i \leq r$.

# B More Details About the Experimental Results

More graphs about the experimental results are showed below. Firstly, as showed in Figs. 1 and 2, upper bound on $\gamma$ gets better when the lattice dimension increases. For the direct method, upper bound on $\gamma$ remains stable when the lattice dimension is between 50 and 170. For the optimized method, the value is between 60 and 300.

We then show the experimental results for $r = 3$ using the direct method in Fig. 3. As the size of the modulus increases, $\gamma$ finally arrives around 0.113. This value is beyond the asymptotic bound $\frac{1}{9}$ of previous Zhang-Takagi method.

The remaining graphs are related to the experiments for $3 \leq r \leq 7$ with various moduli using the optimized method. The lattice dimension of each experiment is set around 300. From Figs. 4, 5, 6, 7 and 8, we find that upper bound on $\gamma$ is higher for smaller modulus and then goes to a lower value. Also it will finally arrive at a certain value that may be determined by the lattice dimension.

Another observation is that these lattices whose dimension is around 300 seem less effective for moduli with larger bit-size. To be specific, it is less effective for the moduli of greater than 500-bit when $r = 3$. The critical bit-size is 700-bit for $r = 4, 5$ and 1000-bit for $r = 6, 7$. Thus, we guess that the lattices used in our experiments are effective for the prime factor of less than 160-bit. To obtain desired upper bounds, we need to apply some lattices with huge dimension.

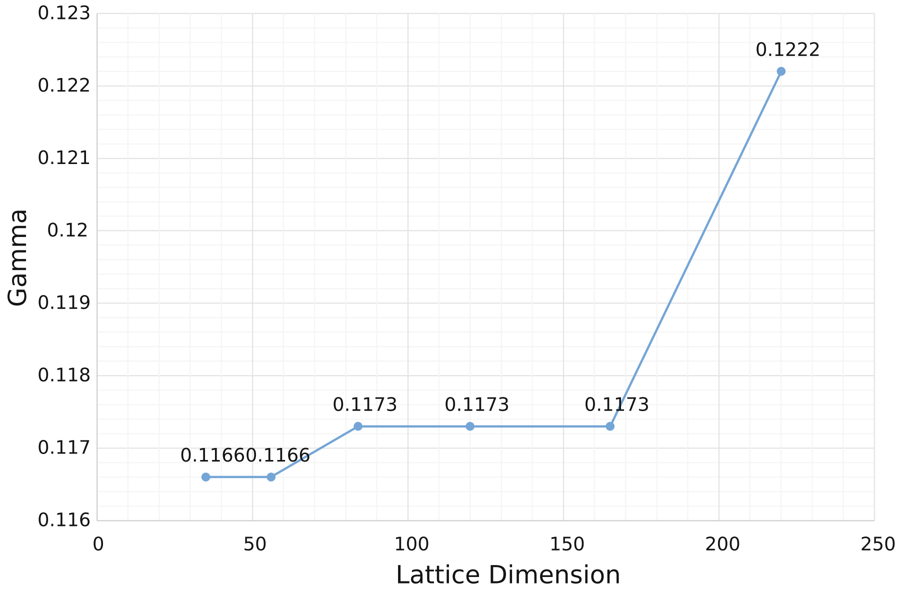


**Fig. 1.** The experimental results of upper bound on $\gamma$ with various lattice dimensions and the same bit-size moduli for $r = 3$ using the direct method

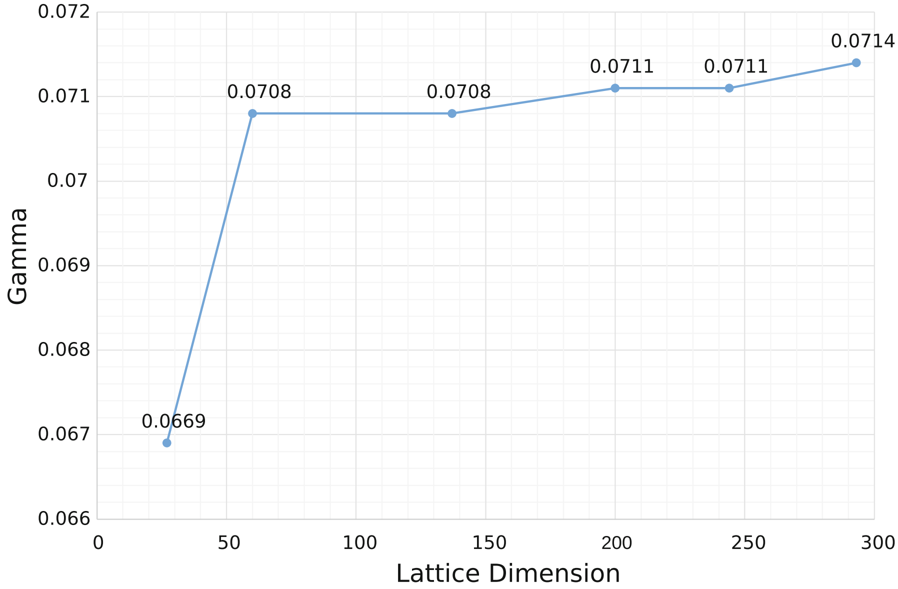


**Fig. 2.** The experimental results of upper bound on $\gamma$ with various lattice dimensions and the same bit-size moduli for $r = 4$ using the optimized method

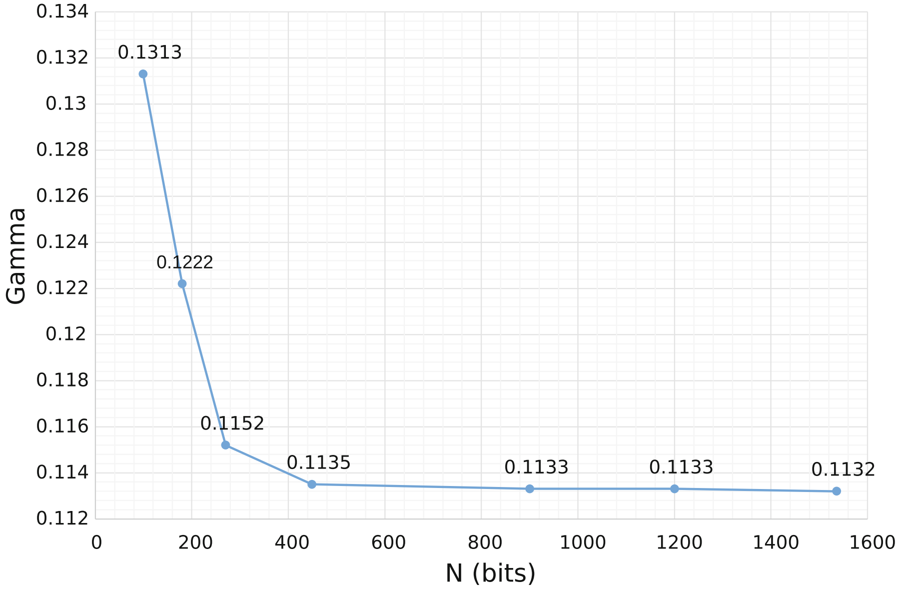


**Fig. 3.** The experimental results of upper bound on $\gamma$ with various moduli for $r = 3$ using the direct method

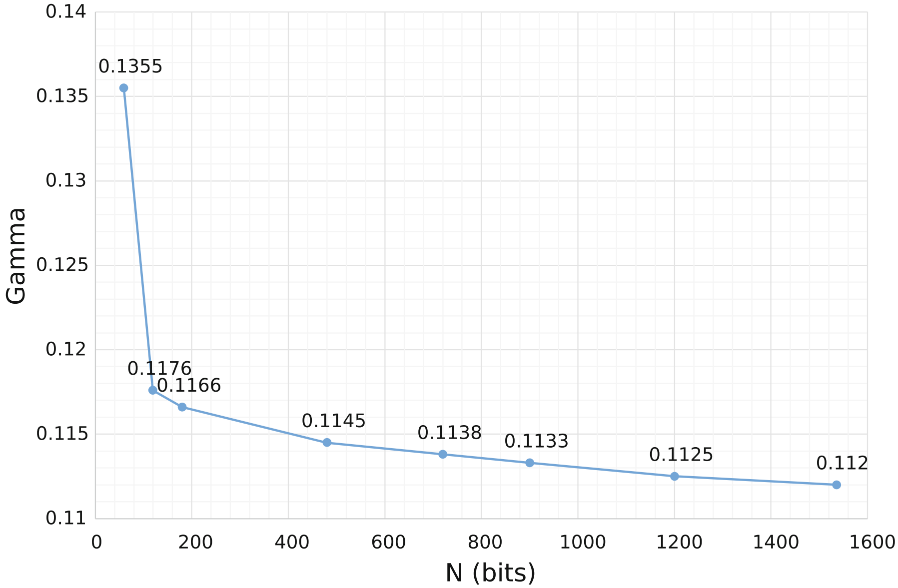


**Fig. 4.** The experimental results of upper bound on $\gamma$ with various moduli for $r = 3$ using the optimized method

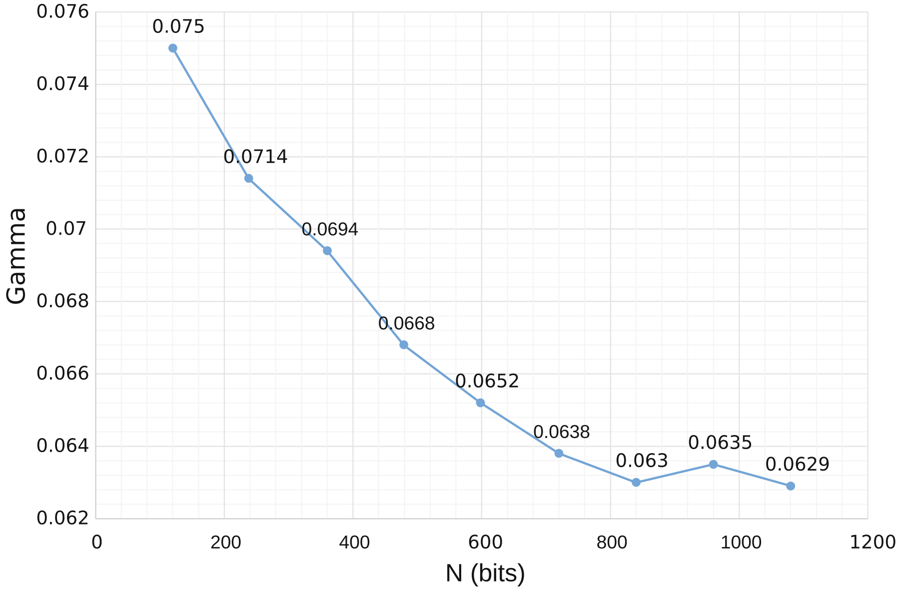


**Fig. 5.** The experimental results of upper bound on $\gamma$ with various moduli for $r = 4$ using the optimized method

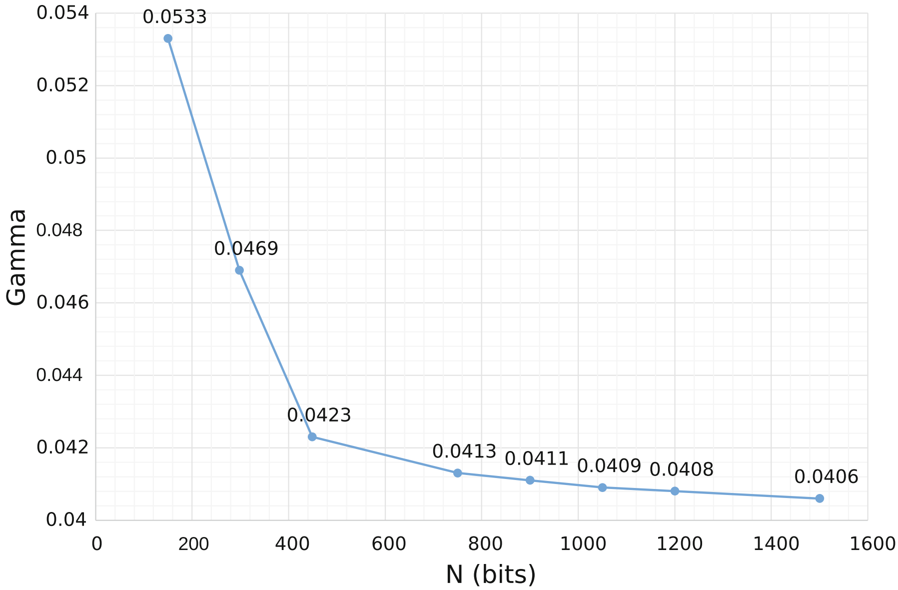


**Fig. 6.** The experimental results of upper bound on $\gamma$ with various moduli for $r = 5$ using the optimized method

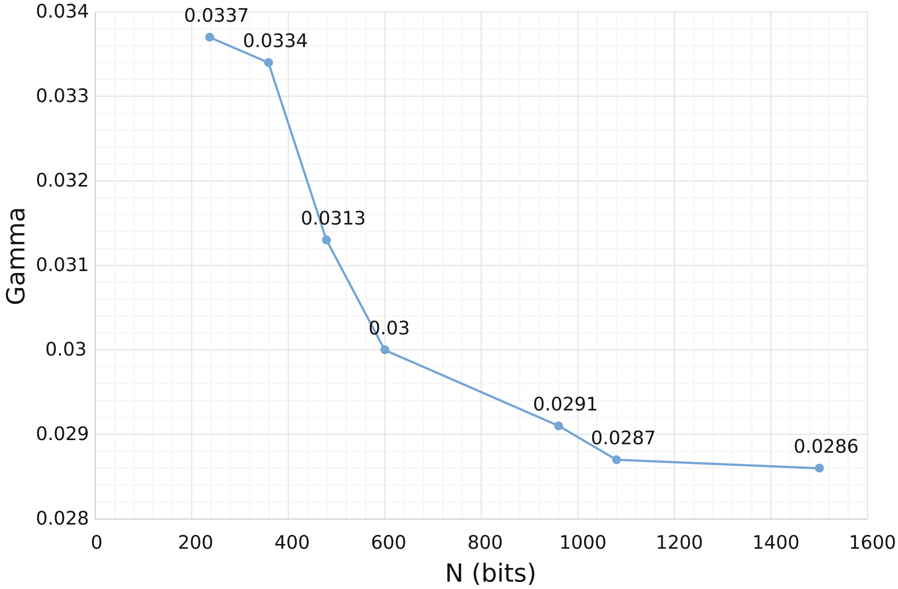


**Fig. 7.** The experimental results of upper bound on $\gamma$ with various moduli for $r = 6$ using the optimized method

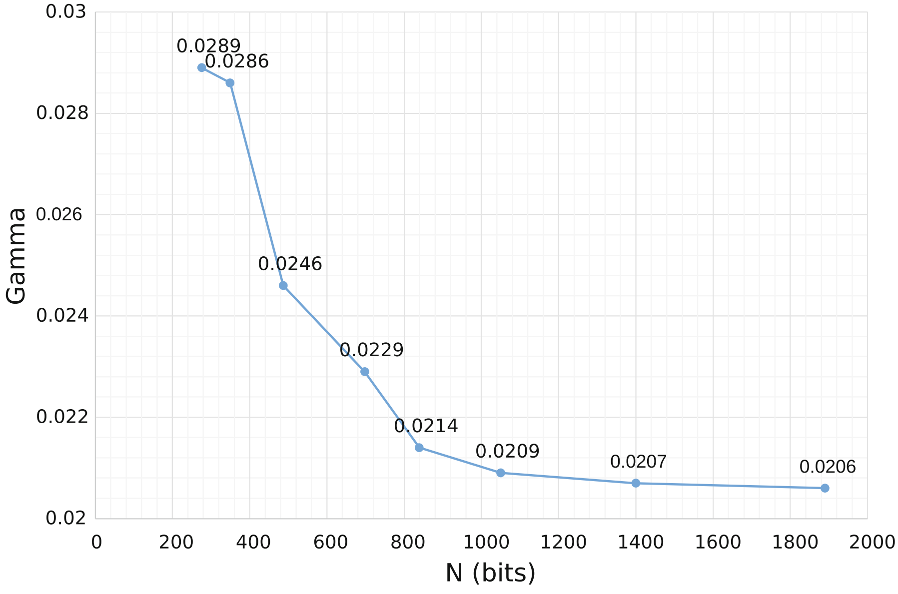


**Fig. 8.** The experimental results of upper bound on $\gamma$ with various moduli for $r = 7$ using the optimized method

## References


1. Bahig, H.M., Bhery, A., Nassr, D.I.: Cryptanalysis of multi-prime RSA with small prime difference. In: Chim, T.W., Yuen, T.H. (eds.) ICICS 2012. LNCS, vol. 7618, pp. 33–44. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34129-8_4
2. Boneh, D., Durfee, G.: Cryptanalysis of RSA with private key $d$ less than $N^{0.292}$. IEEE Trans. Inf. Theory **46**(4), 1339–1349 (2000)
3. Boneh, D., Shacham, H.: Fast variants of RSA. CryptoBytes **5**(1), 1–9 (2002)
4. Ciet, M., Koeune, F., Laguillaumie, F., Quisquater, J.J.: Short private exponent attacks on fast variants of RSA. Technical report, UCL Crypto Group Technical Report Series CG-2002/4, Université Catholique de Louvain (2002)
5. Collins, T., Hopkins, D., Langford, S., Sabin, M.: Public key cryptographic apparatus and method, US Patent#5,848,159 (1997)
6. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 178–189. Springer, Heidelberg (1996). doi:10.1007/3-540-68339-9_16
7. Coppersmith, D.: Finding a small root of a univariate modular equation. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 155–165. Springer, Heidelberg (1996). doi:10.1007/3-540-68339-9_14
8. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptology **10**(4), 233–260 (1997)
9. Coron, J.-S.: Finding small roots of bivariate integer polynomial equations revisited. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 492–505. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24676-3_29
10. Coron, J.-S.: Finding small roots of bivariate integer polynomial equations: a direct approach. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 379–394. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74143-5_21
11. De Weger, B.: Cryptanalysis of RSA with small prime difference. Appl. Algebra Eng. Commun. Comput. **13**(1), 17–28 (2002)
12. Herrmann, M., May, A.: Solving linear equations modulo divisors: on factoring given any bits. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 406–424. Springer, Heidelberg (2008). doi:10.1007/978-3-540-89255-7_25
13. Hinek, M.J.: On the security of multi-prime RSA. J. Math. Cryptology **2**(2), 117–147 (2008)
14. Hinek, M.J., Low, M.K., Teske, E.: On some attacks on multi-prime RSA. In: Nyberg, K., Heys, H. (eds.) SAC 2002. LNCS, vol. 2595, pp. 385–404. Springer, Heidelberg (2003). doi:10.1007/3-540-36492-7_25
15. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (1997). doi:10.1007/BFb0024458
16. Kiltz, E., O'Neill, A., Smith, A.: Instantiability of RSA-OAEP under chosen-plaintext attack. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 295–313. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14623-7_16
17. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
18. Lenstra Jr., H.W.: Factoring integers with elliptic curves. Ann. Math. **126**(3), 649–673 (1987)
19. Lu, Y., Zhang, R., Peng, L., Lin, D.: Solving linear equations modulo unknown divisors: revisited. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 189–213. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48797-6_9

20. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
21. Takayasu, A., Kunihiro, N.: Better lattice constructions for solving multivariate linear equations modulo unknown divisors. In: Boyd, C., Simpson, L. (eds.) ACISP 2013. LNCS, vol. 7959, pp. 118–135. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39059-3_9
22. Takayasu, A., Kunihiro, N.: General bounds for small inverse problems and its applications to multi-prime RSA. In: Lee, J., Kim, J. (eds.) ICISC 2014. LNCS, vol. 8949, pp. 3–17. Springer, Cham (2015). doi:10.1007/978-3-319-15943-0_1
23. Tosu, K., Kunihiro, N.: Optimal bounds for multi-prime $\Phi$-hiding assumption. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 1–14. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31448-3_1
24. Xu, J., Hu, L., Sarkar, S., Zhang, X., Huang, Z., Peng, L.: Cryptanalysis of multi-prime $\Phi$-hiding assumption. In: Bishop, M., Nascimento, A.C.A. (eds.) ISC 2016. LNCS, vol. 9866, pp. 440–453. Springer, Cham (2016). doi:10.1007/978-3-319-45871-7_26
25. Zhang, H., Takagi, T.: Attacks on multi-prime RSA with small prime difference. In: Boyd, C., Simpson, L. (eds.) ACISP 2013. LNCS, vol. 7959, pp. 41–56. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39059-3_4
26. Zhang, H., Takagi, T.: Improved attacks on multi-prime RSA with small prime difference. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **97**(7), 1533–1541 (2014)