An Approach Towards Rebalanced RSA-CRT with Short Public Exponent*

Hung-Min Sun and Mu-En Wu
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300
Email: hmsun@cs.nthu.edu.tw

Abstract

Based on the Chinese Remainder Theorem (CRT), Quisquater and Couvreur proposed an RSA variant, RSA-CRT, to speedup RSA decryption. According to RSA-CRT, Wiener suggested another RSA variant, Rebalanced RSA-CRT, to further speedup RSA-CRT decryption by shifting decryption cost to encryption cost. However, such an approach will make RSA encryption very time-consuming because the public exponent e in Rebalanced RSA-CRT will be of the same order of magnitude as $\phi(N)$. In this paper we study the following problem: does there exist any secure variant of Rebalanced RSA-CRT, whose public exponent e is much shorter than $\phi(N)$? We solve this problem by designing a variant of Rebalanced RSA-CRT with d_p and d_q of 198 bits. This variant has the public exponent $e = 2^{511} + 1$ such that its encryption is about 3 times faster than that of the original Rebalanced RSA-CRT.

Keywords: RSA, RSA-CRT, CRT, lattice basis reduction, LLL.

1 Introduction

RSA [22], the first proposed public key cryptosystem, is the most widely used public key cryptosystem. It is not only built into several operating systems, like Microsoft, Apple, Sun, and Novell, but is also used for securing web traffic, e-mail, smart cards or IC cards. Many practical issues have been considered when implementing RSA, such as how to reduce the storage requirement for RSA modulus[16][24], how to reduce the encryption time (or signature–verification time)[12], how to reduce the decryption time (or signature–generation time)[3][24], how to balance the encryption and decryption time[23], and so on.

The encryption and decryption in RSA require taking heavy exponential multiplications modulus of a large integer N which is the product of two large primes p and q. Without loss of generality, we assume N is of 1024 bits, and p and q are of 512 bits. In general, the RSA encryption and decryption time are roughly proportional to the number of bits in public and secret exponents respectively. To reduce the encryption time (or the signature-verification time), one may wish to use a small public exponent e. The smallest possible value for e is 3; however, it has been proven to be insecure against some small public exponent attacks[13]. Therefore, a more widely accepted

*This version is a revised version of the paper submitted to Eurocrypt'05. Based on the attack proposed by the reviewers of Eurocrypt'05, we make some modifications on parameters. This work is a transcription of our previous results in [27].

and used public exponent is $e = 2^{16} + 1 = 65537$. On the other hand, to reduce the decryption time (or the signature-generation time), one may also wish to use a short secret exponent d. However, the use of short secret exponent encounters a more serious security problem due to some powerful short secret exponent attacks[26][25][2][11]. For instance Wiener[26] announced an attack on short secret exponent, called continued fraction attack. He showed that RSA system can be totally broken if the secret exponent $d < \frac{1}{3}N^{0.25}$. Verheul and Tilborg[25] generalized Wiener's attack to the case where one guesses high-order bits of the prime factors. Their attack needs to do an exhaustive search for about $2t + 8$ bits, where $t = \log_2(d/N^{0.25})$. Based on the lattice basis reduction, Boneh and Durfee[2] further proposed a new attack on the use of short secret exponent. They improved Wiener's bound up to $N^{0.292}$, i.e., RSA system can be totally broken if the secret exponent $d < N^{0.292}$.

Another well-known technique[21] to reduce the decryption time is to employ the Chinese Remainder Theorem (CRT) for RSA decryption. Using this technique, two half-sized modular exponentiations are required. Let $N = pq$ be an RSA modulus and (e, d) be a pair of public exponent and secret exponent. The first modular exponentiation gives the result $C_p \equiv C^{d_p} \pmod{p}$, where $d_p \equiv d \pmod{p-1}$; the second gives the result $C_q \equiv C^{d_q} \pmod{q}$, where $d_q \equiv d \pmod{q-1}$. These two results can be easily combined[20] to obtain the final result $M \equiv C^d \pmod{N}$ by using CRT. Such an approach, called RSA-CRT, achieves 4 times faster in decryption compared to the standard RSA system. Based on CRT decryption, Wiener[26] suggested one can further reduce the decryption time by carefully choosing d, such that both $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$ are small. That is, in the key generation phase, one first selects two small CRT-exponents d_p and d_q, and then these two CRT-exponents are combined to get the secret exponent d satisfying $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$. At last, he computes the corresponding public exponent e satisfying $ed \equiv 1 \pmod{\phi(N)}$. Such a variant of RSA-CRT, called Rebalanced RSA-CRT[26][1][3], enables us to rebalance the difficulty of encryption and decryption. In other words, we can speed up the CRT decryption by shifting the decryption cost to the encryption cost. Note that in Rebalanced RSA-CRT, both d and e will be of the same order of magnitude as $\phi(N)$. The decryption time depends on the bit-size of d_p and d_q, while not on the bit-size of d. But the encryption time depends on the bit-size of e. This will make the encryption for Rebalanced RSA-CRT very time-consuming. Due to Wiener's suggestion[26], a raised open problem [26][1][2] is whether there exists any efficient attack on Rebalanced RSA-CRT. So far, the best known attack [1] on Rebalanced RSA-CRT runs in time complexity $O(\min\{\sqrt{d_p}\log_2 d_p, \sqrt{d_q}\log_2 d_q\})$ which is exponentially in the bit-size of d_p or d_q. Boneh[1][3] suggested to use d_p and d_q of 160 bits in order to defend against this attack. Under such parameters, the decryption in Rebalanced RSA-CRT will be about $\frac{512}{160} = 3.2$ times faster than that in RSA-CRT. It is still an open problem whether Rebalanced RSA-CRT using d_p and d_q of 160 bits is secure. On the other hand, based on the lattice reduction technique, May[19] showed that there is a decrease in security for Rebalanced RSA-CRT using prime factors p and q of unbalanced size. As for Rebalanced RSA-CRT with balanced prime factors p and q, May's attack is not able to work[19].

According to the key generation in Rebalanced RSA-CRT, if we first select small CRT-exponents d_p and d_q, the public exponent e will be of the same bit-size as modulus $\phi(N)$. This causes heavy encryption cost. If we can make the public exponent e much shorter than $\phi(N)$, it will be more convenient and practical in many applications. In this paper, we are interested in studying the following problem: does there exist any secure variant of Rebalanced RSA-CRT, whose public exponent e is much shorter than $\phi(N)$? We solve this problem by designing a variant of Rebalanced RSA-CRT with d_p and d_q of 198 bits. This variant has the public exponent $e = 2^{511} + 1$ such that its encryption is about 3 times faster than that of the original Rebalanced RSA-CRT.

The remainder of this paper is organized as follows. In Section 2, we briefly review previous

work, including the basic RSA, RSA with short secret exponent, RSA-CRT, and Rebalanced RSA-CRT. In Section 3, we review some well-known attacks on RSA variants. In Section 4, we propose and analyze our Scheme-A, the first Rebalanced RSA-CRT with public exponent $e = 2^{511} + 1$. Finally we conclude this paper in Section 6.

2 Overview of Some RSA Variants

2.1 RSA-Basic, RSA-Short-D, and RSA–CRT

We first review the original RSA and CRT decryption[22]. Depending on different choices for parameters and different decryption algorithms used, we classify them as RSA-Basic, RSA-Short-D, and RSA-CRT.

Key Generation in RSA

Let N be the product of two large primes p and q. Let e and d be two integers satisfying $ed \equiv 1 \pmod{\phi(N)}$, where $\phi(N) = (p-1)(q-1)$ is the Euler totient function of N. In general, N is called the RSA modulus, e is the public exponent, and d is the secret exponent.

Encryption in RSA

To encrypt a message (plaintext) M, one computes the corresponding ciphertext $C \equiv M^e \pmod{N}$.

Decryption in RSA

To decrypt the ciphertext C, the legitimate receiver computes $M \equiv C^d \pmod{N}$.

In general, d is a positive number which is smaller than $\phi(N)$. In fact, the secret exponent which can used to decrypt is not unique. For instance, we can let $d' = d - \phi(N)$. Thus d' is negative and equivalent to $d \pmod{\phi(N)}$. That means we can use d' as another secret exponent to decrypt. Indeed,

$$\begin{aligned} C^{d'} \pmod{N} &\equiv M^{e(d-\phi(N))} \pmod{N} \equiv M^{ed} \left(M^{\phi(N)}\right)^{-e} \pmod{N} \\ &\equiv M^{ed} \pmod{N} \equiv M \pmod{N} = M \end{aligned}$$

In general, we will not use such a secret exponent d' which is negative, because it is more time-consuming in decryption due to one more inverse operation required.

CRT Decryption

Based on CRT, Quisquater and Couvreur[21] proposed a fast decryption algorithm, called CRT decryption. Let $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$. The CRT decryption is as follows:

Step 1. Compute $C_p \equiv C^{d_p} \pmod{p}$.

Step 2. Compute $C_q \equiv C^{d_q} \pmod{q}$.

Step 3. Compute $\overline{M} \equiv (C_q - C_p)\, p^{-1} \pmod{q}$.

Step 4. Compute $M = C_p + \overline{M}p$.

Note that the decrypter can compute $p^{-1} \pmod{q}$ in advance. Thus the main cost for CRT decryption is in Step 1 and Step 2. Therefore, CRT decryption is approximately 4 times faster than the decryption in standard RSA[18].

Parameters for RSA-Basic, RSA-Short-D, and RSA-CRT

Here we consider three types of RSA system, called RSA-Basic, RSA-Short-D, and RSA-CRT. In RSA-Basic, one first selects the public exponent $e = 2^{16} + 1$, and thus the secret exponent d is about of 1024 bits. If CRT decryption is applied to RSA-Basic, we call it as RSA-CRT. In RSA-Short-D, one first selects a 512-bit secret exponent, and thus the public exponent is about of 1024 bits. Note that so far RSA can be totally broken if $d < N^{0.292}$[2], but as mentioned by Boneh and Durfree[2]the small inverse problem in [2] is very likely to have a unique solution when $d < N^{0.5}$. Therefore we choose a 512-bit d in RSA-Short-D for achieving high-level security. All these three RSA systems will be compared with other RSA variants in Table 2.

2.2 Rebalanced RSA-CRT

Wiener[26] suggested an RSA variant, Rebalanced RSA-CRT, to further speed up decryption by shifting the work to the encrypter. One version of this variant, which is similar to Boneh and Shacham's version[3], is described in the following.

Key Generation in Rebalanced RSA-CRT

Step 1. Randomly select two 512-bit primes $p = 2p_1+1$ and $q = 2q_1+1$ such that $\gcd(p_1, q_1) = 1$.

Step 2. Compute $p_1^{-1} \pmod{q_1}$ satisfying $p_1 p_1^{-1} \equiv 1 \pmod{q_1}$.

Step 3. Randomly select two distinct odd numbers d_p and d_q of 160 bits such that $\gcd(d_p, p-1) = 1$ and $\gcd(d_q, q-1) = 1$.

Step 4. Compute $\overline{d} \equiv (d_q - d_p)p_1^{-1} \pmod{q_1}$.

Step 5. Compute $d = d_p + \overline{d}p_1$. (Note that $\gcd(d_p, p-1) = 1$ and $\gcd(d_q, q-1) = 1$ imply $\gcd(d, (p-1)(q-1)) = 1$)

Step 6. Compute the public exponent e satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Step 7. The RSA modulus is $N = pq$, the secret key is (d_p, d_q, p, q), and the public key is (N, e).

Encryption

The encryption is the same as the encryption in standard RSA, that is $C \equiv M^e \pmod{N}$.

Decryption

Decryption is the same as the CRT decryption for RSA-CRT. The main difference between both is that in Rebalanced RSA-CRT, the CRT-exponents d_p and d_q are only of 160 bits which are much shorter than the CRT-exponents of 512 bits in RSA-CRT. Thus, decryption in Rebalanced RSA-CRT is about $\frac{512}{160} = 3.2$ times faster than that in RSA-CRT.

3 Related Attacks on RSA Variants

3.1 Short Secret Exponent Attacks

We present some short secret exponent attacks, including Wiener's continued fraction attack [26], and some lattice attacks against on RSA [2][11].

Theorem 3.1[26] In RSA system, let $N = pq$ be an RSA modulus and (e, d) be a pair of public exponent and secret exponent satisfying $ed = k\phi(N) + 1$ for some integer k. Let $|\frac{e}{N} - \frac{k}{d}| = \kappa$. If $\kappa < \frac{1}{2d^2}$, then we can efficiently recover d.

The proof of Theorem 3.1 is shown in Appendix A.1. Note that Wiener's attack can still work when k and d are negative. From this theorem we know if both $\frac{e}{N}$ and $\frac{k}{d}$ are close enough, then we can obtain the value d from one of the values of the continued fraction expansion of $\frac{e}{N}$. Wiener[26] showed the sufficient condition of $\kappa < \frac{1}{2d^2}$ is $d < \frac{1}{3}N^{0.25}$. Besides, the extension of Wiener attack was proposed by Verheul and Tilborg[25]. When $d > N^{0.25}$, their attack needs to do an exhaustive search for about $2t + 8$ bits, where $t \approx \log_2(\frac{d}{N^{0.25}})$. Here we omit reviewing this extension.

Theorem 3.2[2] In RSA system, let $N = pq$ be an RSA modulus and (e, d) be a pair of public exponent and secret exponent satisfying $ed = k\phi(N) + 1$ for some integer k. Let $|e| = N^{\alpha}$ and $|d| < N^{\gamma}$ for some α and γ. If $\gamma < \frac{7}{6} - \frac{1}{3}(1 + 6\alpha)^{1/2}$, then we can heuristically factor the RSA modulus N.

The proof of Theorem 3.2 is shown in Appendix A.2. Note that the public exponent and secret exponent in the above theorem may be negative though Boneh and Durfee did not explicitly mention in their paper. Besides, Durfee and Nguyen[11] generalized Boneh-Durfee attack to the case when the difference between the primes p and q is large. They showed that the more unbalanced the prime factors are, the more insecure the RSA system is. As for the case when p and q are balanced, their attack works up to the same bound as the Durfee-Nguyen attack[2]. We omit describing their attack here.

3.2 Attacks on Rebalanced RSA-CRT

May's Lattice Attack on Rebalanced RSA-CRT with Unbalanced p and q

Theorem 3.3[19] In RSA system, let $N = pq$ be an RSA modulus and (e, d) be a pair of public exponent and secret exponent satisfying $ed \equiv 1 \pmod{\phi(N)}$ and $d_p \equiv d \pmod{p-1}$. Let $p < N^{\beta}$ and $d_p \leq N^{\delta}$ for some integers β and δ. If these parameters satisfy the following condition: $3\beta - \beta^2 + 2\delta < 1$, then there is an algorithm to factor N in time complexity $O(\log_2 N)$.

Note that May's attack[19] is only suitable for the case when p and q are unbalanced. In this paper, we omit reviewing his work because we focus on the case when p and q are balanced.

Boneh's Factoring Attack on Rebalanced RSA-CRT

Here we review Boneh's[1] factoring attack on Rebalanced RSA-CRT in the following.

Theorem 3.4[1] In RSA system, let (N, e) be the public key with $N = pq$. Let d be the corresponding secret exponent satisfying $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$ with $d_p < d_q$.

Then given (N, e), an adversary can expose the secret exponent d in time complexity $O(\sqrt{d_p}\log_2 d_p)$.

The proof of Theorem 3.4 is shown in Appendix A.3. For current security level, we suppose 2^{80} is a safe complexity which makes an exhaustive search infeasible. Therefore, in order to achieve 2^{80} complexity for $O(\sqrt{d_p}\log_2 d_p)$, we need use d_p and d_q of 160 bits.

3.3 Partial Key Exposure Attacks

Boneh, Durfee, and Frankel[4] showed that for low public exponent RSA, given a fraction of the secret exponent bits, an adversary can recover the entire secret exponent. This kind of attack is called the partial key exposure attack. Here we focus only on those attacks for the most significant bits (MSBs) known. Note that all these partial key exposure attacks can still work when d is negative. The secret exponent d in Scheme-A is about of 832 bits. Such a secret exponent d can be regarded as exposing the 192 MSBs of a 1024-bit d (all the 192 MSBs are zero).

Theorem 3.5[4] In RSA system, let $N = pq$ be a 1024-bit RSA modulus and (e, d) be a pair of public exponent and secret exponent satisfying $ed \equiv 1 \pmod{\phi(N)}$.

1. Suppose $e \in [2^t, ..., 2^{t+1}]$ is the product of at most r distinct primes with $256 \leq t \leq 512$. Then given the factorization of e and the t MSBs of d, there is an algorithm to compute all of d in time complexity $O(2^r \log_2 N)$. (We refer the reader to the area BDF2 in Appendix A.4-Fig.1.)
2. When the factorization of e is unknown, e is in the range $[2^t, ..., 2^{t+1}]$ with $t \in 0, ..., 512$, and $d > \epsilon N$ for some $\epsilon > 0$. Then given the $1024 - t$ MSBs of d, there is an algorithm to compute all of d in time complexity $O(\frac{1}{\epsilon}\log_2 N)$. (We refer the reader to the area BDF1 in Appendix A.4-Fig. 1.)

In Appendix A.4-Fig. 1, we illustrate Boneh et al.'s results for MSBs of d. Among them, the area BDF3 was suggested by Blömer and May[7] which can be easily derived from the method in [5]. The idea is that the upper $\log_N e$ bits of d immediately yield half of the MSBs of d and the attacker can use the remaining quarter of bits to factor N.

Based on the above theorem, we know that if e is a 512-bit number of r distinct prime factors, then given the 512 MSBs of a 1024-bit d, an adversary can recover the entire d in time complexity $O(2^r \log_2 N)$. On the other hand, Blömer and May[7] proposed further result about the partial key exposure attack for MSBs of d. Their result works for public exponent e in the interval $[N^{0.5}, N^{0.725}]$ (We refer the reader to the area BM in Appendix A.4-Fig. 1.). We omit reviewing it because our public exponent e is of 512 bits and 432 bits in Scheme-A and Scheme-AI respectively.

Blömer and May[7] also presented a partial key exposure attack for MSBs of CRT-exponent d_p. Their result works for public exponent $e < N^{0.25}$. We refer the reader to Appendix A.4-Fig. 2 instead of reviewing it.

4 The Proposed Scheme-A for Rebalanced RSA-CRT with Short Public Exponent

In this section, we propose a variant of Rebalanced RSA-CRT, called Scheme-A. Scheme-A produces a 512-bit public exponent, e.g., $e = 2^{511} + 1$, two 160-bit CRT-exponents d_p, d_q and a RSA

modulus $N = pq$, where p and q are about of 512 bits. The encryption time is therefore reduced to about one-third of the time required by Rebalanced RSA-CRT. In the following, we first introduce a fundamental theorem in number theory[20] as the basis of our construction.

Theorem 4.1[20] If a and b are relatively prime, i.e. $\gcd(a, b) = 1$, then we can find a unique pair (u_h, v_h) satisfying $au_h - bv_h = 1$, where $(h-1)b < u_h < hb$ and $(h-1)a < v_h < ha$, for any integer $h \geq 1$.

4.1 The Proposed Scheme-A

The key generation in the proposed Scheme-A is as follows:

Key Generation in Scheme-A

Step 1. Randomly select an odd number e of 512 bits.

Step 2. Randomly select an odd number x of 198 bits, such that $\gcd(x, e) = 1$.

Step 3. Based on Theorem 4.1, we can uniquely determine two numbers d_p, $x < d_p < 2x$, and p', $e < p' < 2e$, satisfying $ed_p - xp' = 1$.

Step 4. If $p = p' + 1$ is not a prime number, then go to Step 2.

Step 5. Randomly select an odd number y of 198 bits, such that $\gcd(y, e) = 1$.

Step 6. Based on Theorem 4.1, we can uniquely determine two numbers d_q, $y < d_q < 2y$, and q', $e < q' < 2e$, satisfying $ed_q - yq' = 1$.

Step 7. If $q = q' + 1$ is not a prime number, then go to Step 5.

Step 8. The public key is (N, e); the secret key is (d_p, d_q, p, q).

Note that from Step 3 and Step 6, we know that $ed_p = x(p-1) + 1$ and $ed_q = y(q-1) + 1$, hence, $\gcd(e, \phi(N)) = 1$. Further we multiply these two equations, and hence obtain the following equation:

$$\begin{aligned}
&(ed_p - 1)(ed_q - 1) = x(p-1)y(q-1) \\
\Rightarrow\ & e^2 d_p d_q - ed_p - ed_q + 1 = xy(p-1)(q-1) \\
\Rightarrow\ & e(-ed_p d_q + d_p + d_q) = -xy(p-1)(q-1) + 1
\end{aligned}$$

Let $d = -ed_p d_q + d_p + d_q$ and $k = -xy$. Thus e and d satisfy $ed = k\phi(N) + 1$. Note that both d and k are negative, but this will not affect encryption and decryption. In fact, we can rewrite the above RSA equation "$ed = k\phi(N) + 1$" to "$e(d + \phi(N)) = (k + e)\phi(N) + 1$ " by adding $e\phi(N)$ to both sides. Therefore an equivalent secret exponent is $d' = d + \phi(N)$, which is of the same order of magnitude as $\phi(N)$. Such a secret exponent d' looks like an ordinary secret exponent in standard RSA.

In the following, we show that $d \pmod{p-1}$ is exactly equal to d_p ; and $d \pmod{q-1}$ is exactly equal to d_q.

Because $ed_p = x(p-1) + 1$ and $ed_q = y(q-1) + 1$, we get

$$\begin{aligned}
d \pmod{p-1} &\equiv -ed_p d_q + d_p + d_q \pmod{p-1} \\
&\equiv -[x(p-1) + 1]d_q + d_p + d_q \pmod{p-1} \equiv d_p; \text{ and} \\
d \pmod{q-1} &\equiv -ed_p d_q + d_p + d_q \pmod{q-1} \\
&\equiv -[y(q-1) + 1]d_p + d_p + d_q \pmod{q-1} \equiv d_q.
\end{aligned}$$

Special Public Exponent

In Scheme-A, the public exponent e is of 512 bits. Therefore, 768 modular multiplications are required for encryption. Instead of selecting a random e, we can select a special public exponent $e = 2^{511} + 1$. Thus only 512 modular multiplications are required for encryption. Compared to a 1024-bit public exponent, which can not be arbitrarily selected, in Rebalanced RSA-CRT, the encryption in Scheme-A is about 3 times faster than that in Rebalanced RSA-CRT. As an example, we generate an instance for Scheme-A in Appendix B.

4.2 The Expected Number of Iterations for Loop: Step 2 to Step 4 in Scheme-A

In Scheme-A there are two loops running from Step 2 to Step 4 and from Step 5 to Step 7 respectively. Here we want to show the expected number of iterations for these two loops. Because these two loops work very similarly, we only evaluate the first loop running from Step 2 to Step 4. This problem is almost equivalent to "how many random numbers of 512 bits or 513 bits are required to test in order to find a prime?". We show the expected value is 361 in Appendix C.

4.3 Security Analysis for Scheme-A

Defending against Attacks on Short Secret Exponent

First we consider Wiener's continued fraction attack. In our Scheme-A, the RSA modulus N is of 1024 bits, the public exponent e is of 512 bits, the secret exponent d is of 908 bits, and the parameter k is of 396 bits. Note that both d and k are negative. We list the intervals of these parameters as follows:

$$2^{511} \leq p, q < 2^{512}, 2^{511} \leq e < 2^{512}, -2^{908} < d \leq -2^{907}, -2^{396} \leq k < -2^{395}$$

Following Wiener's continued fraction attack[26], we get:

$$\begin{aligned} |\frac{e}{N} - \frac{k}{d}| = |\frac{ed - Nk}{Nd}| = |\frac{k}{d} \times \frac{-p - q + 1 + 1/k}{N}| > |\frac{k}{d}\frac{p}{N}| = \frac{k}{d}\frac{1}{q} \\ > \frac{1}{2^{511}}\frac{1}{2^{512}} >> \frac{1}{2d^2} \approx \frac{1}{2 \times (2^{907})^2} \end{aligned}$$

Since $|\frac{e}{N} - \frac{k}{d}| >> \frac{1}{2d^2}$, we know Wiener's attack can not be applied to Scheme-A.

Secondly, we consider the Boneh-Durfee attack[2]. From the parameters constructed by Scheme-A, we can get $\alpha \approx 0.5$ and $\gamma \approx \frac{908}{1024}$, where α and γ satisfy $|e| = N^{\alpha}$ and $|d| < N^{\gamma}$ respectively. It is clear that $\gamma \approx \frac{908}{1024} > \frac{7}{6} - \frac{1}{3}(1 + 6\alpha)^{1/2} \approx \frac{1}{2}$. So, the Boneh-Durfee attack cannot succeed.

Defending against Another Lattice Attack

Here we consider another lattice-based attack which was suggested by reviewers from Eurocrypt' 05. Because $ed_p = k_p(p-1) + 1$ and $ed_q = k_q(q-1) + 1$, we can obtain the following two modular equations:

(1). $k_p \cdot p \equiv k_p - 1 \pmod{e}$

(2). $k_q \cdot q \equiv k_q - 1 \pmod{e}$

Combine (1) and (2), we can obtain the following equation with two unknown variables $k_p k_q$ and $k_p + k_q$:

(3). $k_p k_q (N-1) \equiv -(k_p + k_q) + 1 \pmod{e}$

According to Coppersmith's technique[8] of finding the small root of a modular equation[9], the sufficient condition to solve the equation (3) is $|k_p k_q| \cdot |k_p + k_q| < e$. Obviously the proposed scheme makes $|k_p k_q| \cdot |k_p + k_q| \approx 2^{396} \cdot 2^{199} = 2^{595} >> 2^{512} \approx e$. It has 83 bits more than 512-bit public exponent. Thus our scheme has 2^{83} complexity to defend exclusive search upon this boundary condition. Note that if k_p and k_q are known, then one can compute $p = k_p^{-1}(k_p - 1) \pmod{e}$ and $q = k_q^{-1}(k_q - 1) \pmod{e}$ from equations (1) and (2). Since the length of e is 512-bit, which is longer than p and q, one can get p and q immediately. Therefore we should keep the privacy of information k_p and k_q in the proposed scheme.

Defending against Partial Key Exposure Attacks

First we consider the partial key exposure attacks for MSBs of secret exponent d[4][7]. The insecure areas of these attacks are shown in Appendix A.4-Fig. 1. Scheme-A has e of 512 bits and d of 908 bits. Such a d can be regarded as exposing the 116 MSBs of a 1024-bit d (all the 116 MSBs are zero). Therefore, our Scheme-A achieves the point $(0.5, 0.1)$ which is secure against these attacks. Secondly, we consider the partial key exposure attack for MSBs of CRT-exponent d_p[7]. The insecure area of this attack is shown in Appendix A.4-Fig. 2. Scheme-A has e of 512 bits and d_p of 198 bits. Such a d_p can be regarded as exposing the 314 MSBs of a 512-bit d_p (all the 314 MSBs are zero). Note that the fraction for exposure is measured by a denominator 1024. Our Scheme-A achieves the point $(0.5, 0.3)$ which is secure against this attack.

4.4 More Security Considerations for Rebalanced RSA-CRT and Scheme-A

In this section, we will examine the difference between our Scheme-A and the original Rebalanced RSA-CRT from the view of security. Further we will consider the security of these two RSA variants by examining if the additional leaked information in these variants is helpful for an adversary to break these two. Recall that Scheme-A has a secret exponent: $d = -ed_p d_q + d_p + d_q$, where e is of 512 bits, d_p and d_q are of 198 bits, and d is about of 908 bits. Thus $d \pmod{e} \equiv d_p + d_q$ is about of 199 bits. This is obviously different from standard RSA of which $d \pmod{e}$ is the size of order e. Considering the original Rebalance RSA-CRT, we say its secret exponent d is computed from d_p and d_q by using CRT. Such a reconstructed d will make $d \pmod{e}$ be the size of order e. Therefore it seems revealing no more available information on $d \pmod{e}$ than that in standard RSA. In the following we show that in fact, for the original Rebalanced RSA-CRT, there exists another secret exponent, $d' \equiv -ed_p d_q + d_p + d_q$, which could reveal available information on $d' \pmod{e}$ as that in our Scheme-A.

Theorem 4.2 In the original Rebalanced RSA-CRT, the public key is (N, e) and the secret key is (d_p, d_q, p, q). Let $d' = -ed_p d_q + d_p + d_q$, then d' can be used as another secret exponent to decryption.

Proof: Based on the key generation in Rebalanced RSA-CRT, we know that the reconstructed secret exponent d satisfies $d_p \equiv d \pmod{p-1}$, $d_q \equiv d \pmod{q-1}$, and $ed \equiv 1 \pmod{\phi(N)}$. Thus there exists two numbers k_p and k_q such that $ed_p = k_p(p-1) + 1$ and $ed_q = k_q(q-1) + 1$. Similar to Scheme-A, we know $e(-ed_p d_q + d_p + d_q) = -k_p k_q (p-1)(q-1) + 1$. Let $d' = -ed_p d_q + d_p + d_q$. Therefore $ed' \equiv 1 \pmod{\phi(N)}$.

Note that here e is of 1024 bits, d_p and d_q are of 160 bits, and hence d' is about of 1344 bits and $d' \pmod{e} \equiv d_p + d_q$ is about of 160 bits. Now we want to measure the amount of information on d and d' in our Scheme-A and the original Rebalanced RSA-CRT respectively. For our Scheme-A,

Table 1: Comparisons of various RSA variants in terms of encryption and decryption.

	RSA-Basic	RSA-Short-D	RSA-CRT	Rebalanced RSA-CRT	Our Scheme
Public Exponent	$2^{16}+1$	1024 bits	$2^{16}+1$	1024 bits	$2^{511}+1$
Num of Multiplication in Encryption	16+1 =17	1024×1.5 =1536	16+1 =17	1024×1.5 =1536	511+1 =512
Unit Time for Encryption	0.011	1	0.011	1	0.333
Secret Exponent	1024 bits	512 bits	-	-	-
CRT-Exponent	-	-	512 bits	160 bits	198 bits
Num of Multiplication in Decryption (Modular Size)	1024×1.5 =1536 (1024 bits)	512×1.5 =768 (1024 bits)	2×512×1.5 +2=1538 (512 bits)	2×160×1.5 +2=482 (512 bits)	2×198×1.5 +2=596 (512 bits)
Num of Operations in Decryption	$\frac{3}{2}\times \log_2 d \times (\log_2 N)^2$		$2\times\frac{3}{2}\times \log_2 d_p \times (\frac{\log_2 N}{2})^{2=\frac{3}{4}\log_2 d_p \times (\log_2 N)^2}$		
Unit Time for Decryption	1	0.5	0.25	0.078125	0.0966

we can write $d = Ae + B$, where A is about of 396 bits and B is about of 198 bits. This means the uncertainty for d is 592 bits. For the original Rebalanced RSA-CRT, similarly we can write d' $= Ae + B$, where A is about of 320 bits and B is about of 160 bits. Thus the uncertainty for d' in the original Rebalanced RSA is 480 bits.

Now we further consider if the above property in Scheme-A will lead to insecurity. More precisely, with the help of $d = Ae + B$, where e is of 512 bits, d is about of 908 bits, A is about of 396 bits and B is about of 198 bits, whether could those existing attacks on RSA variants become workable for Scheme-A? And whether does there exist any new efficient attack? To the best of our knowledge, the answer for the first question is negative. For Wiener's attack, this property can not further improve Wiener's bound because the convergent condition for continued fraction remains unchanged. For the Boneh-Durfee Attack, no obvious information from this property is available to help solve the small inverse problem. As for the partial key exposure attacks for MSBs, we can not obtain a good approximation of d by this property. So far, it is still an open problem if there exists any new efficient attack on Rebalance RSA-CRT and/or our variants.

5 Implementations and Comparisons

In order to demonstrate our key generation algorithm in the proposed scheme-A is actually feasible, we implemented our algorithm and measured the average running time. The machine used for our implementations is a personal computer (PC) with 1.72 GHz CPU and 512 MB DRAM. The programming language we used is C under NTL with GMP on Windows system using Cygwin tools. We have experimented 1000 samples for the proposed scheme. The average key-generation-time is 9 ms (milliseconds). The average number of iterations for loops running from Step 2 to Step 4 is 352.

Table 1 summarizes the parameters used in various RSA variants and gives comparisons of these RSA variants in terms of encryption and decryption. We recall that the number of binary operations to compute $Z^a \pmod b$ is $1.5 \times \log_2 a \cdot (\log_2 b)^2$. If a is the special form of $2^m + 1$, then the number of binary operations will be reduced to $(m + 1) \cdot (\log_2 b)^2$[10]. In addition, we assume that a full modular exponentiation, $Z^a \pmod b$, where both a and b are of 1024 bits, takes one unit time to compute. It is clear that the encryption in Scheme-A is about 3 times faster than that in Rebalanced RSA-CRT.

6 Conclusions

This paper presents a variant of Rebalanced RSA-CRT to further reduce the encryption cost. We do not only shorten the public exponent in Rebalanced RSA-CRT from 1024 bits down to 512 bits, but also make the public exponent to be of the special form of $2^m + 1$ where $m = 511$. Scheme-A produces a 512-bit public exponent, e.g., $e = 2^{511} + 1$, two 198-bit CRT-exponents d_p, d_q and a RSA modulus $N = pq$, where p and q are about of 512 bits. The encryption time is therefore reduced to about one-third of the time required by Rebalanced RSA-CRT.

Considering practical applications, most systems or softwares that are built for standard RSA accept such keys generated by our two variants. An exception is Microsoft^C Internet Explorer^R (IE) which accepts a maximum of 32 bits for public exponent. Therefore our RSA variants can be widely applied to several systems and softwares.

Acknowledgements:

The authors want to thank anonymous reviewers from Eurocrypt' 05 for giving us helpful comments.

References

[1] D. Boneh, "Twenty Years Attacks on the RSA Cryptosystem," Notices of the American Mathematical Society, vol. 46:2, pp. 203-213, 1999.

[2] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$," IEEE Trans. on Information, Vol. 46(4), pp. 1339-1349, 2000.

[3] D. Boneh and H. Shacham, "Fast Variants of RSA," CryptoBytes, 2002, Vol. 5, No. 1, Springer, 2002.

[4] D. Boneh, G. Durfee and Y. Frankel, "An Attacks on RSA Given a Small Fraction of the Private Key Bits," Advanced in Cryptology–ASIACRYPT '98, LNCS 1514, Springer-Verlag, pp.25-34, 1998.

[5] D. Boneh, G. Durfee and Y. Frankel, "Exposing an RSA Private Key Given a Small Fraction of its Bits," Full version of the work from ASIACRYPT'98.

[6] J. Blömer and A. May,"Low Secret Exponent RSA Revisited. CaLC , LNCS, vol. 2146 Springer-Verlag, pp.110-125, 2001.

[7] J. Blömer and A. May,"New Partial Key Exposure Attacks on RSA," Advanced in Cryptology–CRYPTO'03, LNCS 2729, Springer-Verlag, pp.27-43, 2003.

[8] D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities," Journal of Cryptology, vol. 10, pp.233-260, 1997.

[9] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, "Low-Exponent RSA with related message," Advances in Cryptology-EUROCRYPT'96, LNCS 1070, Springer-Verlag, pp.1-9. 1996.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithm,second edition, McGraw-hill Book Company, 2001.

[11] G. Durfee, P. Q. Nguyen, "Cryptanalysis of the RSA Schemes with Short Secret Exponent form Asiacrypt '99," Advances in Cryptology-Asiacrypt'00, LNCS 1976, Springer-Verlag, pp.1-11, 2000.

[12] J. Hasted, "On Using RSA with Low Exponent in a Public Key Network," in Proceedings of CRYPTO'85, Springer-Verlag, pp.403-408, 1986.

[13] J. Hastad, "Solving simultaneous modular equations of low degree", SIAM J. of Computing, Vol. 17, pp.336-341, 1988.

[14] N. Howgrave-Graham. "Finding small roots of univariate modular equations revisited," in Proceedings of Cryptography and Coding, LNCS, vol. 1355, Springer-Verlag, pp.131-142, 1997.

[15] http://www.alpertron.com.ar/ECM.HTM

[16] A. Lenstra, "Generating RSA moduli with a predetermined portion," Advances in Cryptology-ASIACRYPT'98, LNCS 1514, Springer-Verlag, pp.1-10, 1998.

[17] A. Lenstra, H. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients." Mathematische Annalen, vol. 261, pp.515-534, 1982.

[18] A. J. Menezes, P.C. van Oorschot, and S. A Vanstone, Handbook of applied Cryptography, CRC Press, 1996.

[19] A. May, "Cryptanalysis of Unbalanced RSA with Small CRT-Exponent," Advances in Cryptology-CRYPTO'02, LNCS 2442, pp.242-256, 2002.

[20] I. Niven, H. S. Zuckerman, An Introduction to the Theory of Number, John Wiley and Sons Inc,1991.

[21] J. J. Quisquater, C. Couvreur, "Fast decipherment algorithm for RSA public key cryptosystem," Electronic Letters, vol. 18, pp.905-907, 1982.

[22] R. Rivest, A. Shamir and L. Aldeman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM , vol. 21, No.2, pp.120-126, 1978.

[23] H.-M. Sun, W.-C. Yang and C.-S. Laih., "On the Design of RSA with Short Secret exponent," Advances in Cryptology-ASIACRYPT'99, LNCS 1716, pp.150-164, 1999.

[24] S. A. Vanstone and R.J. Zuccherato, "Short RSA keys and their generation," Journal of Cryptology, Vol. 8, pp.101-114, 1995.

[25] E. Verheul and H. van Tilborg, "Cryptanalysis of less short RSA secret exponents," Applicable Algebra in Engineering, Communication and Computing, vol. 8, Springer-Verlag, pp. 425-435, 1997.

[26] M. J. Wiener, "Cryptanalysis of RSA with short secret exponents," IEEE Transactions on information Theory, IT-36, pp.553-558, 1990.

[27] M- N Wu, A Study of RSA with Small CRT-Exponent, Thesis of Master Degree, National Chiao Tung University, Taiwan, June 2004.

Appendix A

(Omitted)

Appendix B: An example of Scheme-A with $e = 2^{511} + 1$, d_p and d_q of 198 bits, p and q of 512 bits.

$N =$ 47888A59 8E2424D4 A2028A18 0FD1E48F 8768E4D9 904E9DC6 7D06B001
673C503E BE99846A 39B008E2 78575949 671651C7 D69072E0 845526FB
F6E300D2 96624A1E BAF1A05B FD326D08 C1CE200C 77C0E48F 26BDBB3F
FD191E99 80402EBA 5BABE134 1B66DDC7 87FDE72E 438EE6BC C99CEDFF
47488893 B3810881 22AA6E3C 31FB5915

$p =$ 8AF105A8 85F84D30 16ED6D69 E1DA359F 09BDA979 6CDA651E DBCC4F52
994A8EB4 A70BED4B 3E3E0383 D73AF9B5 444919D1 7F00C09D F0765B4C
A1884148 3686F81B

$q =$ 83CCE530 49C698ED 36032BBF B0F21A34 70B9608C 50BCB458 CE53F6C0
3E50BE33 16D59EA5 A77FA305 DBD2D6DE E99D5A78 11BA28D1 7E49C682
B6B8D086 34F4668F

$d_p =$ 3B 511D01FC 82437878 BB60BF77 47B30EC5 0CF65340 573A7503

$d_q =$ 27 0345E74F DC319B67 4DC6FD9A 7EDF33EA 0F6BB29B 434801FF

$e = 2^{511}+1$

Appendix C

(Omitted)