Trading decryption for speeding encryption in Rebalanced-RSA

Hung-Min Sun\textsuperscript{a,}\textsuperscript{*}, Mu-En Wu\textsuperscript{b}, M. Jason Hinek\textsuperscript{c}, Cheng-Ta Yang\textsuperscript{d}, Vincent S. Tseng\textsuperscript{d}

\textsuperscript{a}Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, ROC  
\textsuperscript{b}Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC  
\textsuperscript{c}School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada  
\textsuperscript{d}Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, ROC

\begin{abstract}
In 1982, Quisquater and Couvreur proposed an RSA variant, called RSA-CRT, based on the Chinese Remainder Theorem to speed up RSA decryption. In 1990, Wiener suggested another RSA variant, called Rebalanced-RSA, which further speeds up RSA decryption by shifting decryption costs to encryption costs. However, this approach essentially maximizes the encryption time since the public exponent $e$ is generally about the same order of magnitude as the RSA modulus. In this paper, we introduce two variants of Rebalanced-RSA in which the public exponent $e$ is much smaller than the modulus, thus reducing the encryption costs, while still maintaining low decryption costs. For a 1024-bit RSA modulus, our first variant (Scheme A) offers encryption times that are at least 2.6 times faster than that in the original Rebalanced-RSA, while the second variant (Scheme B) offers encryption times at least 3 times faster. In both variants, the decrease in encryption costs is obtained at the expense of slightly increased decryption costs and increased key generation costs. Thus, the variants proposed here are best suited for applications which require low costs in encryption and decryption.
\end{abstract}

\begin{keywords}
RSA  
CRT  
Encryption  
Digital signatures  
Latent basis reduction  
Cryptanalysis
\end{keywords}

\section{Introduction}

Many practical issues have been considered when implementing the RSA such as how to reduce the storage requirement for the RSA modulus, how to reduce the encryption time (or signature-verification time), how to reduce the decryption time (or signature-generation time) ([Boneh and Shamir, 2002], how to balance the encryption and decryption time (Sun et al., 1999; Sun and Yang, 2005), and so on. In this work we are concerned with minimizing the overall cost of encryption and decryption combined (or the cost of signature-generation and verification combined) in Rebalanced-RSA.

Encryption and decryption in RSA each require an exponentiation modulo a large integer $N$ which is the product of two large primes, $p$ and $q$. The exponent in the exponentiations is the public exponent $e$ for encryption and the private exponent $d$ for decryption. Using the standard double-and-square method for modular exponentiation, the cost for encryption and decryption is roughly proportional to the number of bits in $e$ and $d$, respectively.

To simply reduce the encryption time (or the signature-verification time), one can use a small public exponent $e$. The smallest possible value for the public exponent is $e = 3$, however a more widely accepted and used small public exponent is $e = 2^{10} + 1$ ([Coppersmith et al., 1996; Coppersmith, 1997]). With an appropriate padding scheme, such as OAEP, using such small public exponents is considered safe. On the other hand, to simply reduce the decryption time (or the signature-generation time), one can use a small private exponent $d$. However, unlike the small public exponent scenario, using a very small private exponent is insecure (Boneh and Durfee, 2000; Wiener, 1990). Indeed, instances of RSA with private exponent $d < N^{0.25}$ can be efficiently broken with Wiener’s continued fraction attack (Verheul and van Tilborg, 1997; Wiener, 1990) and Boneh and Durfee’s lattice-based attack (Boneh and Durfee, 2000) shows that and instance of RSA with $d < N^{0.292}$ should be considered unsafe.

A more sophisticated way to reduce the decryption time is to employ the Chinese Remainder Theorem (CRT) ([Quisquater and Couvreur, 1982]). Using this technique, two half-sized modular exponentiations are required. Consider an instance of RSA with modulus $N = pq$, public exponent $e$, and private exponent $d$ which is roughly the same order of magnitude as $N$. To decrypt a given a ciphertext $C$ using the standard definition of RSA, one would compute the plaintext as $M = C^d \pmod{N}$. Using the CRT, one first computes $C_p = C \pmod{p}$ and $C_q = C \pmod{q}$, where $d_p = d \pmod{p - 1}$ and $d_q = d \pmod{q - 1}$, and then combines these with the CRT to obtain the final result $M = C^d \pmod{N}$. Such an approach, called RSA–CRT, achieves decryption times that are four times faster than decryption using the standard RSA definition. And, since there are no restrictions on the public exponent $e$, we are free to...
use a small public exponent, e.g., $e = 2^{16} + 1$, to reduce the encryption costs. Note that using too small public exponent, e.g., $e = 3$, is insecure due to the cryptanalysis of Coppersmith et al. (1996), Coppersmith (1997).

In 1990, Wiener (1990) proposed an RSA variant, called Rebalanced-RSA. The purpose of Rebalanced is to further reduce the decryption cost again by choosing advanced-RSA. The purpose of Rebalanced is to further reduce the insecure due to the cryptanalysis of Coppersmith et al. (1996), Coppersmith et al. (2005) proposed a similar but independent scheme in their later sections.

This paper introduces new definitions throughout this paper.

2. Preliminaries

In this section we review the original RSA along with some variants. In particular, we are interested in variants (or parameter choices) that reduce the encryption or decryption time. We do not, however, consider multi-prime or multi-power RSA here. For a survey on fast variants of RSA see Boneh and Shacham, 2002.

Section 3. In Section 3 we present our proposed variants, Scheme A and Scheme B, and analyze their security in Section 4. Some performance data and comparisons of our proposed schemes with the other RSA variants are given in Section 5. Finally, we conclude in Section 6.

2.1. Notations

We use the following notation: $N = pq$ is an n-bit RSA modulus with balanced primes (i.e., $p$ and $q$ are roughly the same size), $e$ is an $n_b$-bit public exponent, $d \equiv e^{-1} \pmod{\phi(N)}$ is a valid private exponent for the public key $(e, N)$, and $k$ is the unique integer satisfying $ed = 1 + k\phi(N)$, called the key equation. When CRT-decryption is used, $d_p$ and $d_q$ are each $n_d$-bit CRT-exponents while $k_p$ and $k_q$ are each $n_b$-bit integers satisfying $ed_p = 1 + k_p(p - 1)$ and $ed_q = 1 + k_q(q - 1)$, respectively. We call these equations the CRT key equations. Also, we often use $|x|$ to denote the bit-length of $x$. The notation introduced here will remain the same for the rest of this paper.

2.2. Original RSA

In general, there are three phases in RSA cryptosystem (Rivest et al., 1978):

2.2.1. Key generation

Choose two distinct large random primes $p$ and $q$. Their product, $N = pq$, is called the RSA modulus. Choose a large random integer $e$ that is relatively prime to $\phi(N) = (p - 1)(q - 1)$. Finally, the integer $d$ is computed as the multiplicative inverse of $e$ modulo $\phi(N)$. The pair $(e, N)$ is the public key and the pair $(d, N)$ is the private key. The integer $e$ is called the encryption (or public) exponent while the integer $d$ is called the decryption (or private) exponent. Note that with high probability the public exponent $d$ will be roughly the same order of magnitude as $\phi(N)$.

Since the public and private exponents are inverses of each other modulo $\phi(N)$, we know that $ed \equiv 1 \pmod{\phi(N)}$. We call this the key relation. Also, there must exist an integer $k$ such that $ed = 1 + k\phi(N)$. We call this the key equation.

We call the RSA primes, $p$ and $q$, balanced if they are roughly the same size. In particular, they are balanced if $0.5N^{1/2} < p < N^{1/2} < q < 2N^{1/2}$ (where $p$ and $q$ can be interchanged). We will only consider instances of RSA with balanced primes.

2.2.2. Encryption

A plaintext message $M \in \mathbb{Z}_N$ is encrypted by raising it to the eth power modulo $N$. The result, $C \equiv M^e \pmod{N}$, is called the ciphertext of $M$. Notice that $C \in \mathbb{Z}_N$ also. All variants discussed below use this algorithm for encryption.

2.2.3. Decryption

A ciphertext $C \in \mathbb{Z}_N$, for a given plaintext message $M$, is decrypted by raising it to the dth power modulo $N$. This follows since

$$C^d \pmod{N} \equiv M^{ed} \pmod{N} \equiv M^{1+k\phi(N)} \pmod{N} \equiv M \pmod{N} = M.$$

An obvious way to decrease encryption or decryption time is to simply use a small public or private exponent. We consider each as a fast variant of RSA.
2.2.4. RSA-Small-e

We call instances of RSA with a small public exponent RSA-Small-e. Encryption and decryption remain the same. Key generation is slightly modified though. For a random small public exponent of size \( n_e \) (i.e. \( e \) with bit-length \( n_e \)), first choose the RSA primes then find a random \( n_e \)-bit integer \( e \) that is relatively prime to \( \phi(n) \). The private exponent is simply computed as the inverse of \( e \) modulo \( \phi(n) \). Note that \( d \) will be roughly the same order of magnitude as \( \phi(n) \) with very high probability. For a fixed public exponent, \( e = 2^{16} + 1 \) for example, the primes are each randomly chosen so that they are relatively prime to the predetermined \( e \). Again, \( d \equiv e^{-1} \pmod{\phi(n)} \). The smallest possible value for a public exponent is \( e = 3 \), but the value \( e = 2^{16} + 1 \) is the most commonly used fixed value (Coppersmith et al., 1996; Coppersmith, 1997).

2.2.5. RSA-Small-d

We call instances of RSA with a small private exponent RSA-Small-d. As with RSA-Small-e, encryption and decryption remain the same. Key generation is the same as with RSA-Small-e also, except that the roles of \( e \) and \( d \) are interchanged. Unlike the small public exponent case though, very small values of \( d \) cannot be used safely. According to Boneh and Durfee’s attack (Boneh and Durfee, 2000), the private exponent should be greater than about \( N^{0.292} \) to be considered safe.

2.3. RSA-CRT

Based on the Chinese Remainder Theorem (CRT), Quisquater and Couvreur (1982) proposed a fast decryption algorithm called CRT-decryption. Combining this decryption method with a small public exponent leads to a significant decrease in encryption and decryption times as compared to the original RSA system. We will call instances of RSA with a small public exponent using CRT-decryption RSA-CRT.

The key generation is the same as for RSA-Small-e except that the private key is now the tuple \((d_p,d_q,p,q)\), where \(d_p \equiv d \pmod{p-1}\) and \(d_q \equiv d \pmod{q-1}\). We call \(d_p\) and \(d_q\) CRT-exponents. When \(d_p\) and \(d_q\) are roughly the same size we say that they are balanced CRT-exponents.

Encryption is the same as the original definition. For a given ciphertext \( C \in \mathbb{Z}_n\), the decryption algorithm is as follows:

Step 1. Compute \( C_p \equiv C^d \pmod{p} \).
Step 2. Compute \( C_q \equiv C^d \pmod{q} \).
Step 3. Compute \( M_0 \equiv (C_q - C_p) \cdot p^{-1} \pmod{q} \).
Step 4. Compute the plaintext \( M = C_p + M_0 \cdot p \).

Note that the value of \( p^{-1} \pmod{q} \) can be computed in advance (or can be added to the private key). Thus, the main cost for CRT decryption is in steps 1 and 2. Compared to the modular exponentiation in original RSA decryption, the bit-length of both the exponents and the modulus are half-sized. Therefore, CRT-decryption is approximately four times faster if the computations are done sequentially.

2.4. Rebalanced-RSA

In order to further speed up decryption, Wiener suggested a variant of RSA-CRT which shifts some of the work done by decryption to encryption in (Wiener, 1990). Boneh and Shacham (2002) call this variant Rebalanced-RSA.

Encryption and decryption are the same as with RSA-CRT. For key generation, to generate an instance of Rebalanced-RSA with an \( n\)-bit modulus, and \( n_a\)-bit CRT-exponents we do the following:

Step 1. Randomly select two distinct \((n/2)\)-bit primes \( p = 2p_1 + 1 \) and \( q = 2q_1 + 1 \) such that \( \gcd(p_1,q_1) = 1 \).
Step 2. Compute \( p_1^{-1} \pmod{q} \).
Step 3. Randomly select two distinct \((n/2)\)-bit odd integers satisfying \( \gcd(d_p,p−1) = 1 \) and \( \gcd(d_q,q−1) = 1 \).
Step 4. Compute \( d_q \equiv (d_q−d_p) p_1^{-1} \pmod{q_1} \).
Step 5. Compute \( d = d_q + d_0 \cdot p_1 \).
Step 6. Compute \( e = e \cdot d_1^{-1} \pmod{\phi(N)} \).

The public key is given by the pair \((e,N)\) and the private key is given by the tuple \((d_0,d_p,d_q)\). Note that \( e \) will be roughly the same order of magnitude as \( \phi(N) \) with high probability. Thus, in order to reduce the decryption time even further than RSA-CRT, the encryption time is essentially maximized.

As with RSA-Small-d, in order to be secure the values of the CRT-exponents, \( d_q \) and \( d_0 \) cannot be chosen to be arbitrarily small. We discuss this more in the next section.

2.5. Generalized Rebalanced-RSA

In this work we modify the key generation algorithm of Rebalanced-RSA so that public exponents much smaller than \( \phi(N) \) can be used. In particular, we propose two schemes for key generation which will be given in Section 3. Independently, a related scheme has been proposed by Galbraith et al. (2005). The only significant difference between our schemes and their proposed variant is the key generation algorithm. Their scheme is slightly more general than the ones proposed here, since they do not restrict themselves to balanced primes and balanced CRT-exponents, but their scheme only allows for public exponents smaller than \( N^{1/2} \) when the primes are balanced. We will refer to all of these schemes as instances of Generalized Rebalanced-RSA. In addition, Galbraith et al. consider various attacks on Generalized Rebalanced-RSA in (Galbraith et al., 2005). We give an overview of their main attacks with respect to the parameters \( n,s,n_e \) and \( k_p = \lceil k_q \rceil = \lceil k_q \rceil \). Each attack includes an exhaustive search on \( m \) bits. If \( k_p \) or \( k_q \) is known, they show that a lattice-based attack exists that can factor the modulus provided that \( n_e \geq n/4 - m \). Thus, to guard against this attack, one must ensure that \( n_e \leq n/4 - m \). When \( k_p \) and \( k_q \) are not known, there are two main attacks; a linearization attack and a lattice-based attack. The linearization attack recovers \( k_p \) and \( k_q \) whenever \( n_e \leq (n_e + m)/3 \). This attack is used in conjunction with the known \( k_p \) (or \( k_q \)) attack to ultimately factor the modulus. To guard against this attack, one must ensure that \( n_e \geq (n_e + m)/3 \) or at least ensure that \( n_e \leq n/4 - m \) so that the second part of the attack cannot be mounted. The last attack is a lattice-based attack that can be thwarted when \( n_e \geq 4n_e \geq 2n_e + 4m \). The details of this last attack are omitted in Galbraith et al. (2005).

3. The proposed schemes

In this section we present two key generation algorithms for Rebalanced-RSA which generate public exponents much smaller than \( \phi(N) \). The first key generation algorithm, Scheme A, is used to generate instances with public exponent \( N^{1/2} < e < N \) and small CRT-exponents. The second algorithm, Scheme B, is used to generate instances with public exponent \( e < N^{1/2} \) but with slightly larger CRT-exponents. Each key generation algorithm is based on the following fundamental theorem from number theory (Niven et al., 1991).

Theorem 3.1. Let \( a \) and \( b \) be relatively prime integers unequal to 1 (i.e. \( \gcd(a,b) = 1 \), and \( a \neq b \)). For every integer \( h \) there exists a unique and efficiently computable pair of integers \((u,v)\) satisfying \( au - bv = 1 \), where \( (h - 1)b < u_b < h_b \) and \( (h - 1)a < u_a < h_a \).

We now present the two key generation algorithms. The first algorithm can be used to generate instances of the Rebalanced-RSA
with public exponents larger than $N^{1/2}$ ($n_e > n/2$), while the second algorithm generates instances with public exponents smaller than $N^{1/2}$ ($n_e < n/2$). Parameter suggestions and security considerations will be discussed in Section 5.

3.1. The proposed Scheme A

The first key generation algorithm, Scheme A, takes $(n,n_e,n_d)$ as input, with $n_e > n/2$, and outputs a valid public key $(e,N)$ and corresponding private key $(d_p,d_q,p,q)$, where $|N| = n$, $|e| = n_e$ and $|d_p| = |d_q| = n_d$. The Scheme A is as follows:

**Input:** $(n,n_e,n_d)$, where $n_e > n/2$.

1. $e \leftarrow$ random $(n_e)$-bit odd integer.
2. $k_p \leftarrow$ random $(n_e)$-bit integer such that $\gcd(k_p,e) = 1$.
3. Using Theorem 4.1 (with $h = 2$), compute $(d_p,dp)$ satisfying $ed_p = kp + 1$, where $k_p < dp < 2np$ and $e < 2e$.
4. Factor $P$ as $P = kp2(n - 1)$ where $k_p$ is an $(n_e - n/2)$-bit integer and $P$ is prime. If this is infeasible, then go to Step 2.
5. $k_q \leftarrow$ random $(n_e)$-bit.
6. Integer such that $\gcd(k_q, e) = 1$.
7. Using Theorem 4.1 (with $h = 2$), compute $(d_p,d_q)$ satisfying $e = k_q + 1$, where $k_q < dq < 2nq$ and $e < 2e$.
8. Factor $Q$ as $Q = k_q2(n - 1)$ where $k_q$ is an $(n_e - n/2)$-bit integer and $Q$ is prime. If this is infeasible, then go to Step 5.

**Output:** the public key $(e,N)$ and the private key $(d_p,d_q,p,q)$.

Now we show that the output is indeed a valid public key and corresponding private key. Let $k_p = k_p2k_d$ and $k_q = k_q1k_d$. From steps 2–4 and 5–7, we know that $ed_p = kp2(n - 1) + 1$ and $ed_q = k_q + 1$, respectively. This implies $\gcd(e, \phi(N)) = 1$. Furthermore, multiplying these two equations together yields $(ed_p - 1)(ed_q - 1) = k_p(n - 1)q + 1$, which can be rewritten as $e^{-1}d_p + d_q$ and $e^{-1}d_q + d_p$ where $e < 2e$.

3.2. The proposed Scheme B

The second key generation algorithm, Scheme B, takes $(n,n_e,n_d)$ as input, with $n_e < n/2$, and outputs a valid public key $(e,N)$ and corresponding private key $(d_p,d_q,p,q)$, where $|N| = n$, $|e| = n_e$ and $|d_p| = |d_q| = n_d$. The algorithm is as follows:

**Input:** $(n,n_e,n_d)$, where $n_e < n/2$.

1. $e \leftarrow$ random $(n_e)$-bit odd integer.
2. $d_q \leftarrow$ random $(n/2 - n_e)$-bit odd integer, $d_p \leftarrow ed_q$.
3. $k_p \leftarrow$ random $(n_e + n_d - n/2)$-bit integer satisfying $\gcd(k_p, e) = 1$.

4. Using Theorem 4.1 (with $h = 2$), compute $(d_p,d_q)$ satisfying $Ed_p = k_p + 1$, where $k_p < dp < 2k_p$ and $e < 2e$ and let $dp \leftarrow d_q$.
5. $p \leftarrow q + 1$. If $p$ is not prime, then go to Step 3.
6. $d_q \leftarrow$ random $(n/2 - n_e)$-bit odd integer, $d_p \leftarrow ed_q$.
7. $k_q \leftarrow$ random $(n_e + n_d - n/2)$-bit integer satisfying $\gcd(k_q, e) = 1$.
8. Using Theorem 4.1 (with $h = 2$), compute $(d_p,d_q)$ satisfying $Ed_p = k_q + 1$, where $k_q < dq < 2k_q$ and $e < 2e$ and let $d_q \leftarrow d_q + 1$.

**Output:** the public key $(e,N)$ and the private key $(d_p,d_q,p,q)$.

That $(e,N)$ and $(d_p,d_q,p,q)$ is a valid public/private key pair follows from, essentially, the same reasoning as in Scheme A and so we omit the details here.

The key generation algorithm for this scheme is much more efficient than the one in Scheme A. The runtime of the algorithm is dominated by two loops, like Scheme A, but each iteration requires much less computational effort as there is no factoring required. And, the experimentally observed expected number of iterations for each loop is much less also. It was observed that each loop requires about $2^{10}$ iterations for this algorithm.

3.3. Sparse public exponents

In both schemes the public exponent $e$ is chosen before the RSA primes are generated. In order to speed up encryption further, we can modify step 1 of each algorithm to choose an $(n_e)$-bit odd integer that has a sparse binary representation. In particular, we can use $e = 2^{n_e - 1} + 1$, which minimizes the number of multiplications required for encryption. In the original Rebalanced-RSA, since the public exponent is computed as the inverse of a number modulo $\phi(N)$ it will be, with high probability, roughly the same order of magnitude as $\phi(N)$ and its binary representation will on average have an equal number of ones and zeros. Since $|\phi(N)| = |N| = n$, the average number of multiplications for encryption will be $1.5 \times n$. Using a sparse public exponent generated with either scheme reduces the number of multiplications to essentially $n_e$. When compared to the original Rebalanced-RSA, using a sparse public exponent with Scheme A reduces in a speed up for encryption by a factor of at most 3 (and at least 1.5), while using a sparse public exponent with Scheme B results in a speed up for encryption by a factor of at least 3. Therefore, we suggest that sparse public exponents always be used in these schemes.

Scheme A is designed to generate instances of Rebalanced-RSA with public exponents larger than $N^{1/2}$. In Appendix B, we present an alternate scheme, denoted Scheme A*, which also generates instances of Rebalanced-RSA with public exponents larger than $N^{1/2}$ that has a more efficient key generation algorithm. Unfortunately, Scheme A* does not allow the user to generate sparse public exponents.

4. Security analysis

In this section we focus on the security of Generalized Rebalanced-RSA. In particular, we investigate which combinations of...
and \( n_2 \) are secure and which are insecure. We begin by first considering the security of Rebalanced-RSA when viewed as normal RSA and then consider attacks that exploit the CRT key equations. The results of the latter attacks are essentially the same results in (Galbraith et al., 2005; Bleichenbacher and May, 2006). Here, we provide the details of their lattice-based attack and also show that their linearization attack can be viewed as a lattice-based attack as well. In addition, we also present two new attacks that slightly reduce the secure region for the parameter choices for the public and private exponents \((n, n_p)\).

### 4.1. Generalized Rebalanced-RSA as normal RSA

Multiplying together the two CRT key equations, written as \( ed_p - 1 = k_p(p - 1) \) and \( ed_q - 1 = k_q(q - 1) \), yields (after some rearrangement)

\[
e(d_p + d_q - ed_qd_p) = 1 - k_pk_q\phi(N),
\]

(4.1) which defines a valid RSA public/private exponent pair \((e, d_0 = d_p + d_q - ed_qd_p)\) with constant \( k_0 = -k_pk_q \). From the CRT key equations we have \( n_0 + n_d \geq n/2 \) and so \( |d_0| = n_0 + 2n_d \geq n/2 \). Therefore, each of the small private exponent partial key-exposure attacks requires knowing at least \( n/4 \) bits of \( d_0 \). For the small public exponent attacks when \( n_d \geq n/2 \), again, at least \( n/4 \) bits of \( d_0 \) must be known. Some stronger attacks exist when \( n_0 < n/2 \) which only require 1/4 of the bits of \( d_0 \) in the best case (from Boneh et al. (1998)). In the known LSBs attack, \( n_0 \) is very small and so \( n_d \approx n/2 \). Therefore, \( |d_0| = n \) and so \( n/4 \) bits of \( d_0 \) are required. For known MSBs, the attack is strongest when \( n_0 = n/4 \). In this case, \( n_d \geq n/4 \) and so we have \( |d_0| \approx 3n/4 \). Therefore, at least \( 3n/16 \) bits of \( d_0 \) are required. Considering all the partial key-exposure attacks, we then see that at least \( n/16 \) bits of \( d_0 \) are required (for a 1024-bit modulus this is 192 bits). Guessing these bits with an exhaustive search is as expensive as factoring a 1024-bit RSA modulus whenever \( n > 428 \). So, if we make the assumption that the implementation of the cryptosystem is resistant to side channel attacks we then suggest that the partial key-exposure attacks are not a threat to the security of the proposed schemes. It should be pointed out that in some instances \( |d_0| > n \). For these instances, the partial key-exposure attacks apply to \( d_0 \) modulo \( \phi(N) \) instead of \( d_0 \). This is not a threat, however, since in these instances \( d_0 \) modulo \( \phi(N) \) will be roughly the same order of magnitude as \( \phi(N) \) with very high probability.

The small private exponent attacks on RSA present a more serious danger. We consider Boneh and Durfee’s lattice-based attack (Boneh and Durfee, 2000; Qiao and Lam, 2000). Let \( A = N - \phi(N) = A_m + A_0 \) and \( M = N - A_m \), where \( A_m \) represents the m MSBs of \( A \) and \( A_0 \) represents the remaining bits. With this we can write the key equation as \( ed_0 - 1 = k_0(M - A_0) \). Assuming that \( M \) is known (i.e. \( A_m \) is known), the lattice-based small private exponent attacks involve trying to find all integer pairs \((x_0, y_0)\) such that \(|x_0| < X, |y_0| < Y \) and \( f(x_0, y_0) = q(x_0 - M - y_0) \equiv 1 \mod e \), where the bounds \( X \) and \( Y \) are defined so that \( X = |k_0| + Y = |A_0| \). One such pair will be \((x_0, y_0) = (-k_0, A_0)\). Once we know \( A_0 \) we also know \( \phi(N) \) which then allows us to invert \( e \) modulo \( \phi(N) \); which breaks the partial key-exposure attacks are not a threat to the security of the CRT. Theorem 4.1: [Trivariate Linear Modular Equations]: Let \( f(x, y, z) \) be a linear polynomial with integer coefficients. For every \( e > 0 \), there exists a positive \( M_0 \) such that for every integer \( M > M_0 \) that is relatively prime to at least one non-constant coefficient of \( f \) we can find three linearly independent polynomials such that each root \((x_0, y_0, z_0)\) of \( f(x, y, z) \) (mod \( M \)) is also a root of the three polynomials modulo \( M \), and if \(|x_0| < X, |y_0| < Y, |z_0| < 2X + 2Y < M^{-1} \), for some bounds \( X, Y \) and \( Z \), then \((x_0, y_0, z_0)\) is also a root of each of the three polynomials modulo \( M \). If these three polynomials are also algebraically independent then we can compute \((x_0, y_0, z_0)\).

Corollary 4.2: [Bivariate Linear Modular Equations]: Let \( f(x, y) \) be a linear polynomial with integer coefficients. For every \( e > 0 \), there exists a positive \( M_0 \) such that for every integer \( M > M_0 \) that is relatively prime to at least one non-constant coefficient of \( f \) we can find two linearly independent polynomials such that each root \((x_0, y_0)\) of \( f(x, y) \) (mod \( M \)) is also a root of each of the two polynomials modulo \( M \), and if \(|x_0| < X, |y_0| < Y < M^{-1} \), for some bounds \( X \) and \( Y \), then \((x_0, y_0)\) is also a root of each of the two polynomials modulo \( M \). If these two polynomials are also algebraically independent then we can compute \((x_0, y_0)\).
It is currently an open question to prove that the polynomials obtained are algebraically independent or not, but in practice if the solution is not too close to the bounds it is often the case that the polynomials are algebraically independent. The proof of Theorem 4.1 and Corollary 4.2 will be given in Section 4.4. We now simply use these results to show which combinations of \( n_e \) and \( n_d \) are (in)secure.

The strongest attack is obtained by considering Eq. (4.4) modulo \( e^\phi \). This is the lattice-based attack in Section 5.2 of (Galbraith et al., 2005). Here we look for small solutions of \( f(x,y,z) = ex - (N - 1)y + z + k_m \) (mod \( e^\phi \)). In particular, we are looking for \( (x_0,y_0,z_0) = (d_p(k_p - 1) - d_q(k_q - 1), -k_p k_q,0) \). Notice that \( y_0 \) and \( z_0 \) yield \( k_p \) and \( k_q \). The remaining values \( (d_p, d_q) \) are recovered by substituting \( x_0, y_0, z_0 \) into Eq. (4.3) to obtain a new equation with unknown variable \( d_p d_q \). Using \( k_p, k_q \), the equation for \( x_0 \) and this new equation allows us to recover \( d_p, d_q \) and \( d_g \) since we now have two equations in two variables. Using the bounds \( X, Y \) and \( Z \) given by \( \log_2 X = 2n_0 + n_e - n/2 \), \( \log_2 Y = 2n_0 + 2n_e - n \), and \( \log_2 Z = n/2 - m \) with Theorem 4.1, we obtain the following enabling equation to find three linearly independent polynomials with the root \( (x_0,y_0,z_0) \) over the integers:

\[
5n_0 + 2n_e < 2n + m 
\] (4.5)

Thus, if we choose \( n_e \) and \( n_d \) so that inequality (4.5) is satisfied, we can expect to do at least \( O(2^m) \) work to recover \( d_p \) and \( d_q \), which is a full break of the system. With \( m = 80 \), we then expect to do at least as much work as factoring a 1024-bit RSA modulus. Also, if we substitute \( n_e = n_d = n_e - n/2 \), which holds when the primes are balanced, into the lattice-based attack in Section 5.2 of (Galbraith et al., 2005) we obtain, essentially, the same result.

If we instead consider Eq. (4.4) modulo \( e \), we then look for small solutions to \( g(x,y) = (N - 1)x + y + k_m \) (mod \( e \)). In particular, we are looking for the root \( (x_0,y_0) = (-k_p k_q,0) \). In this case, finding \( x_0 \) and \( y_0 \) only yields \( k_p \) and \( k_q \) though. Using the bounds \( X, Y \) and \( Z \) defined by \( \log_2 X = 2n_0 + 2n_e - n \) and \( \log_2 Y = n/2 - m \) with Corollary 4.2, we obtain the following enabling equation to find two linearly independent polynomials with the root \( (x_0,y_0) \) over the integers:

\[
3n_0 + 2n_e < (3/2)n + m \quad \text{(4.6)}
\]

As above, if we choose \( n_e \) and \( n_d \) so that inequality (4.6) is satisfied, we can expect to do at least \( O(2^m) \) work to recover \( k_p \) and \( k_q \). Once \( k_p \) and \( k_q \) are known, we try to factor the modulus using the known \( k_p \) or \( k_q \) attack from Galbraith et al. (2005). This factoring attack works can be mounted when \( n_e > n/4 - m_2 \), where another exhaustive search must be carried out on \( m_2 \) bits. To impose \( O(2^m) \) work for the exhaustive searches we simply let \( m = m_1 + m_2 \). Notice that if we substitute \( n_e = n_d = n_e - n/2 \), which holds when the primes are balanced, into the linearization attack in Section 4.1 of (Galbraith et al., 2005) we obtain the same result. Thus, we have shown that Galbraith, Heneghan and McKee’s linearization attack is equivalent to a lattice-based attack.

Lastly, we consider reducing Eq. (4.4) modulo \( N \). Here we look for small solutions of \( h(x,y,z) = e^\phi x + ey + z + k_m \) (mod \( N \)), where \( k_m \) represents the \( m \) most significant bits of \( k_p k_q - k_p - k_q + 1 \) (and \( k_0 \) represents the remaining bits). In particular, we look for the root \( (x_0,y_0,z_0) = (d_p(d_p, d_q, (k_p - 1) - d_q k_q - 1), k_0') \). All of \( d_p, d_q, k_p \) and \( k_q \) can be recovered with knowledge of \( (x_0,y_0,z_0) \). Each of \( x_0, y_0, z_0 \) and \( z_0 \) yields an equation in \( (d_p, d_q, k_p, k_q) \) and substituting \( (x_0,y_0,z_0) \) into Eq. (4.3) will yield an additional equation with unknown \( k_p, k_q \). Thus, we have four equations and four unknowns which we can solve. For very small \( n_e \) values, this attack is slightly superior to the mod \( e^\phi \) attack and the mod \( e \) attack. In this region, the mod \( e \) attack can only recover \( k_p \) and \( k_q \) though. Using the bounds \( X = 2^{2n_0 + 2n_e - n/2} \) and \( Y = 2^{2n_0 + 2n_e - n/2} \) with Theorem 4.1 we obtain the following enabling equation to find three linearly independent polynomials with the root \( (x_0,y_0,z_0) \) over the integers:

\[
6n_0 + 3n_e < (5/2)n + m \quad \text{(4.7)}
\]

As with the previous cases, if we choose \( n_e \) and \( n_d \) so that inequality (4.7) is satisfied, we can expect to do at least \( O(2^m) \) work to recover \( k_p \) and \( k_q \).

### 4.3. Secure parameter choices

Here we summarize the attacks presented above in Table 1, by giving conditions on \( n_e \) and \( n_d \) so that each attack requires at least \( O(2^m) \) work. Using \( m = 80 \) results in work that is comparable to factoring a 1024-bit RSA modulus. To be considered secure, it is recommended that each inequality given below be satisfied:

In Fig. 4.2 we illustrate each of the above inequalities for a 1024-bit modulus with \( m = 80 \). Any parameter choice for \( n_e \) and \( n_d \) in the shaded region violates one or more of the inequalities and should be considered insecure. The region between the dotted line and the shaded region represents parameter choices in which the mod \( e \) attack recovers \( k_p \) and \( k_q \) but there is no known method to recover \( d_p \) and \( d_q \) (or factor \( N \)) with this knowledge. And finally, the remaining region above and to the right of all the lines represents parameter choices that can not be broken by using known methods. In order to simplify the visual representation we have been overly pessimistic and let \( m_1 = m_2 = m \) instead of considering all combinations of \( m_1 \) and \( m_2 \) such that \( m_1 + m_2 = m \). Notice that the small private exponent attack (Section 4.1) and the mod \( N \) lattice-based attack slightly extend the shaded (insecure) region of the previously known attacks (mod \( e \) and mod \( e^\phi \)). This region can be seen at the bottom-right of the shaded region when \( n_d \approx 400 \) and \( n_e \) is small (<200). Also notice that the small private exponent attack is implied by the other attacks (i.e. the region

<table>
<thead>
<tr>
<th>Method</th>
<th>Inequality</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mod(^e^))</td>
<td>(5n_p + 2n_q &gt; 2n + m)</td>
</tr>
<tr>
<td>(mod(^e^), known (k_p))</td>
<td>(3n_p + 2n_q &gt; (3/2)n + m) or (n_q &lt; n/4 - m_2), where (m = m_1 + m_2)</td>
</tr>
<tr>
<td>(mod (N))</td>
<td>(6n_p + 3n_q &gt; (5/2)n + m)</td>
</tr>
<tr>
<td>Small private exponent</td>
<td>(6n_p + 3n_q &gt; (7/3)n - m - (1/2))</td>
</tr>
<tr>
<td>Baby-step giant-step</td>
<td>(4m^2 - 4m n - 12m_1 n + 6n n^2)</td>
</tr>
</tbody>
</table>

**Fig. 4.2.** Lattice attacks on Rebalanced-RSA.
corresponding to this attack lies completely within the shaded region of the other attacks).

### 4.4. Small solutions of linear equations

In this subsection, we give a proof for Theorem 4.1. First we recall two important results. The first gives a bound on the $m$th smallest vector of an LLL-reduced lattice basis (Lenstra et al., 1982).

**Theorem 4.3. (LLL)** Let $b_1, b_2, \ldots, b_n$ be an LLL-reduced basis of a full dimensional integral lattice $L \subseteq \mathbb{Z}^n$. For $m = 1$ or $1 < m \leq n$ and $|b_m| > 2^{(n-2)/4}$, then we have the following properties: $|b_m| \leq 2^{n(m+2)}/4\det(L)^{1/(n-m+1)}$.

The next result gives a sufficient condition for roots of a polynomial modulo some integer to also be roots of the polynomial over the integers. The result is usually attributed to Howgrave-Graham (1997). We give a slightly generalized version of it.

**Theorem 4.4. (Howgrave-Graham)** Let $h(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be the sum of at most $w$ monomials and let $m$ be a positive integer. Suppose that there exists $y_1, \ldots, y_n \in \mathbb{Z}$ such that $h(y_1, \ldots, y_n) \equiv 0 \pmod{m^w}$ and $|h(x_1, \ldots, x_n)| < w^d m^w$, where $|y_i| < x_i$ for some integers $x_1, \ldots, x_n$. Then $h(y_1, \ldots, y_n) = 0$ over the integers.

We now give a proof of Theorem 4.1.

**Proof of Theorem 4.1:** Let $M$ be a positive integer with unknown factorization, and consider the linear integer polynomial $f(x,y,z) = Ax + By + Cz + D \in \mathbb{Z}[x,y,z]$, such that at least one non-constant coefficient is relatively prime to $M$. We will assume, without loss of generality, that $\gcd(C,M) = 1$. Multiplying $f(x,y,z)$ by $C^{-1}$ modulo $M$ we obtain the new polynomial $f_0(x,y,z) = ax + by + cz$, which has the same roots as $f(x,y,z)$ modulo $M$. We look for all small roots of $f_0(x,y,z)$ modulo $M$. That is, given bounds $X,Y,Z$, we wish to find all $(x_0, y_0, z_0) \in \mathbb{Z}$ such that $|x_0| < X$, $|y_0| < Y$, $|z_0| < Z$ and $f_0(x_0, y_0, z_0) \equiv 0 \pmod{M}$. In the following, let $h(x_0, y_0, z_0)$ be any such root. We will try to find the maximal bounds that lattice basis reduction techniques (based on Coppersmith’s techniques) can allow.

For some positive integer $m$, to be determined later, we consider the set of polynomials $g_{i,j,k}(x,y,z) = x^i y^j M^{-k} f_0(x,y,z)$, for non-negative integer values of $i,j,k$. Notice that for any values of $i,j \geq 0$ and $0 \leq k \leq m$, we have $g_{i,j,k}(x_0, y_0, z_0) \equiv 0 \pmod{M^m}$. We construct a lattice basis for the lattice $L$ using the coefficients of $g_{i,j,k}(x,y,z)$ for certain values of $i,j,k$. In particular, for each $0 \leq d \leq m$ we use all values $0 \leq k \leq d$ along with each combination of $i$ and $j$ such that $i+j \leq d-k$. This gives $w = m(m+1)(m+2)/6$ linearly independent vectors which form the lattice basis for $L$. The structure of the basis matrix is shown above in Table 2 (lower right-hand side of the table).

Each row in the basis matrix is the coefficient vector of one of the $g_{i,j,k}(x,y,z)$. Notice that the ordering of the columns ensures that the basis matrix is triangular and that the lattice is full dimensional (with dimension $w$). Since the matrix is triangular, we can easily compute the determinant/volume of the lattice (given by the absolute of the determinant of the basis matrix) yielding

$$\det(L) = (XYZ)^{w(m+1)(m+2)(m+3)/(w+1)}/24 = (XYZ)^{w(m+3)/4}.$$}

Applying the LLL-algorithm to the above lattice we can find three linearly independent vectors in $L$ whose sizes are bounded by $2^{(w-1)/4}\det(L)^{1/(w-2)}$, where $w$ is the lattice dimension defined above. Each of these vectors, being elements of $L$, correspond to a polynomial of the form $h(x,y,z)$ such that $h(x_0, y_0, z_0) \equiv 0 \pmod{M^m}$.

Let these polynomials be denoted by $h_1, h_2$, and $h_3$. Using Howgrave-Graham’s condition for integer roots (Theorem 4.4), we see that a sufficient condition for $(x_0, y_0, z_0)$ to be a root of these polynomials over the integers rather modulo $M^m$ is given by

$$|h_1| < 2^{(w-1)/4}\det(L)^{1/(w-2)} < M^m/M^{w/2},$$

which simplifies to

$$\det(L) < M^m/(w-2)/2^{(w-1)/(w-2)}/4\w^2.$$
\(g(x,y,z) = f(x,y) + z - 1\), with desired root \((x_0,y_0, 1)\) and apply Theorem 4.1. It is straightforward and very similar to the proof of Theorem 4.1 to prove Corollary 4.2 using only two variables so we omit the details of this proof to save space.

In Section 4.2 we also mention that an attack can be mounted by considering Eq. (4.3) as an integer linear equation in four variables. This attack yields the same result as considering Eq. (4.3) reduced modulo \(e^2\), so we omit the details.

4.5. Comparisons of this work and tunable balancing of RSA

In this work we have provided the details of the mod \(e^2\) lattice-based attack in Section 5.2 of (Galbraith et al., 2005), shown that their linearization attack is equivalent to a mod \(e\) lattice-based attack, and provided two additional attacks that slightly reduce the region of secure parameter choices (for \(n_e\) and \(n_d\)).

5. Implementations

To demonstrate the feasibility of the proposed schemes we have implemented the key generation algorithms and measured the average running time for each, A personal computer (PC) with 2.5 GHz CPU and 512 MB DRAM was used. The algorithms were implemented using NTLL (NTL, 2008) with GMP (ECM, 2008) using Cygwin tools on the Windows operating system. For Scheme A we ran the key generation algorithm 100 times obtaining an average running time for each. A personal computer (PC) with the other RSA variants in terms of the complexity of encryption and decryption for a 1024-bit modulus with security parameter \(m = 80\). Here we assume that a random \(k\)-bit exponent will require \(1.5 \times k\) multiplications. We also assume that decryption is always carried out using the Chinese Remainder Theorem. Thus, RSA-Small-\(e\) is the same as RSA-CRT. The complexity for each encryption (and decryption) is the expected number of bit operations. To compute \(2^e \mod b\) for a random exponent \(a\), we expect \(1.5 \times |a| \times |b|^2\) binary operations (Durfee and Nguyen, 2000). When the exponent is of the special form of \(2^k + 1\), the number of binary operations will be reduced to \((k+1) \times |b|^2\) bit operations. In each instance for Scheme A (and Scheme B), we use public exponents of the form \(2^k + 1\). For Scheme A we use the instance \((n_e,n_d) = (592, 190)\) and for Scheme B we use the instance \((n_e,n_d) = (170, 358)\). Both instances satisfy the security conditions listed in Section 4.3. Compared to the original Rebalanced-RSA, our instance from Scheme A encrypts about 2.6 times faster while decryption is about 1.2 times slower. For Scheme B, compared to the original Rebalanced-RSA, encryption is about nine times faster while decryption is about 2.2 times slower.

6. Conclusions

We present two key generation algorithms for Generalized Rebalanced-RSA, which allow the size of the public and CRT-exponents to be chosen by the user. The first scheme, Scheme A, generates public exponents with \(n/2 < n_e < n\) and CRT-exponents that are fairly close that of the original Rebalanced-RSA. The second scheme, Scheme B, generates public exponents with \(n_e < n/2\) and CRT-exponents that are larger than that of Scheme A.

We also present a security analysis of Generalized Rebalanced-RSA in terms of the parameters \(n_e\) and \(n_d\) that compliments the security analysis found in (Galbraith et al., 2005). It is an open question if stronger attacks on Rebalanced-RSA exist. Moreover, we show that the linearization attack proposed in (Galbraith et al., 2005) can also be viewed as a lattice-based attack, the analysis in this work generates a slightly different condition for the last attack though (our analysis leads to a factor of \(m\) instead of \(4m\)). Note that we just provide the algebraic technique to analyze the security of our schemes.

As can be derived from Table 3, the proportion of encryption cost and decryption cost in Scheme A and Scheme B are about 4.17 and 0.634, respectively. Thus, our proposed schemes are especially suitable for the environment on computational capability with such ratio, saying 4.17 or 0.63. Our schemes are feasible for applying to any RSA embedded protocol that used in the current environment, such as off-line generators, online proxy generators, and so on. However, we have to point out that we do not consider the attacks on the cases of ad-hoc, or wireless sensor network environments, although it may cause some other problems on the security consideration.

Appendix A. Examples of Schemes A and B

A.1

An example of Scheme A with \(e = 2^{591} + 1\), \(d_p\) of 190 bits and \(d_q\) of 191 bits, \(p\) and \(q\) of 513 bits.

### Table 3

<table>
<thead>
<tr>
<th></th>
<th>RSA-small-(e)</th>
<th>RSA-Short-d</th>
<th>RSA-CRT</th>
<th>Rebalanced-RSA</th>
<th>Scheme A</th>
<th>Scheme B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of multiplications</td>
<td>(e = 2^{101} + 1)</td>
<td>(n) bits</td>
<td>(e = 2^{10} + 1)</td>
<td>(n) bits</td>
<td>(e = 2^{591} + 1)</td>
<td>(e = 2^{100} + 1)</td>
</tr>
<tr>
<td>Complexity</td>
<td>17 \times 2^{20}</td>
<td>1536 \times 2^{20}</td>
<td>17 \times 2^{20}</td>
<td>1536 \times 2^{20}</td>
<td>592 \times 2^{20}</td>
<td>170 \times 2^{20}</td>
</tr>
<tr>
<td>CRT-exponent</td>
<td>(-)</td>
<td>(-)</td>
<td>(0.5 \times n) bits</td>
<td>(160) bits</td>
<td>(190) bits</td>
<td>(358) bits</td>
</tr>
<tr>
<td>Private exponent</td>
<td>(n) bits</td>
<td>(&gt;350) bits</td>
<td>(-)</td>
<td>(0.5 \times n) bits</td>
<td>(-)</td>
<td>(0.5 \times n) bits</td>
</tr>
<tr>
<td>Modulus</td>
<td>1.5 \times 2^{10}</td>
<td>&gt;1.5 \times 350</td>
<td>1.5 \times 2^{10}</td>
<td>&gt;0.469 \times 2^{10}</td>
<td>0.55 \times 2^{10}</td>
<td>1.04 \times 2^{10}</td>
</tr>
<tr>
<td>Complexity</td>
<td>1536 \times 2^{20}</td>
<td>&gt;525 \times 2^{20}</td>
<td>384 \times 2^{20}</td>
<td>&gt;120 \times 2^{20}</td>
<td>142 \times 2^{20}</td>
<td>268 \times 2^{20}</td>
</tr>
</tbody>
</table>
A.2

An example of Scheme B with \( e = 2^{160} + 1 \), \( d_p \) of 359 bits, \( d_q \) of 357 bits, \( p \) of 512 bits and \( q \) of 511 bits.

Appendix B. Scheme A*

The most costly operations for the key generation algorithm of Scheme A are the two factoring steps (4 and 7). Here we present another key generation algorithm, called Scheme A*, which is about twice as fast as the key generation of Scheme A. Like Scheme A, the public exponents in Scheme A* satisfy \( n/2 < n_e < n \). The main drawback of Scheme A*, however, is that the public exponent cannot be chosen to be sparse. Thus, the encryption costs in Scheme A* are expected to be 1.5 times larger than that in Scheme A for the same size \( n_e \).

The key generation algorithm for Scheme A* takes \((n, n_e, n_0)\) as input, with \( n > n_e > n/2 \), and outputs a public key \((e, N)\) and corresponding private key \((d_p, d_q, p, q)\). The algorithm is as follows:

**Input:** \((n, n_e, n_0)\), where \( n_e > n/2 \).

1. **Step 1.** \( p \rightarrow \) random \((n, n_e, n_0)\)-bit prime, \( d_p \rightarrow \) random \((n_e, n_0)\)-bit odd integer such that \( gcd(d_p - 1, p) = 1 \).
2. **Step 2.** \( k_{p_2} \rightarrow \) random \((n_e - n/2)\)-bit integer, using Theorem 4.1 (with \( h = 2 \)), compute \((e, k_{p_2})\) satisfying \( ed_p = k_{p_2} k_p(p - 1) + 1 \), where \( k_p(p - 1) < e < 2k_p(p - 1) \) and \( d_p < k_{p_2} < 2d_p \).
3. **Step 3.** \( k_{q_2} \rightarrow \) random \((n_0, n_0)\)-bit integer such that \( gcd(k_{q_2}, e) = 1 \).
4. **Step 4.** Using Theorem 1.1 (with \( h = 2 \)), compute \((d_q, Q)\) satisfying \( ed_q = k_{q_2} Q + 1 \), where \( k_{q_2} < d_q < 2k_{q_2} \) and \( Qe < Q < 2e \).
5. **Step 5.** Factor \( Q \) as \( Q = k_{q_2}(q - 1) \) where \( k_{q_2} \) is an \((n_0, n_0)\)-bit number and \( q \) is prime. If this is infeasible, then go to Step 3.

**Output:** the public key \((e, N)\) and the private key \((d_p, d_q, p, q)\).

The encryption of Scheme A* is at least 1.7 times faster than that in the original Rebalanced-RSA. Compared to Scheme A, Scheme A* provides much faster key generation but slower encryption time. Thus, we have a trade-off between encryption time and key generation time. The time of the key generation algorithms we have run is 912 seconds in average and it is actually near half of the time compared with Scheme A.

References


Mu-En Wu received his B.S. degree in the Department of Mathematics from National Tsing-Hua University in 2002, M.S. degree in the Department of Applied Mathematics from National Chiao-Tung University in 2004. Then he went back to Tsing-Hua to purchase his Ph.D degree and received the degree of Computer Science from National Tsing-Hua University in 2009. Now he is joining the Institute of Information Science (IIS) in Academic Sinica, Taiwan. His research interests include lattice reduction, RSA, Algorithmic number theory, Cryptography, and Algorithm.

M. Jason Hinek received his Ph.D. in computer science at the University of Waterloo, Canada, in 2007. He is currently an adjunct research fellow in the iCORE Information Security Lab at the University of Calgary, Canada.

Cheng-Ta Yang received his B.S. degree in computer and information sciences from Tung Hai University, Taichung, Taiwan, Republic of China, in 1991, and his M.S. degree in computer science and engineering from Yuan Ze University, Taiwan, in 1993. He is currently pursuing his Ph.D. degree in computer science and information engineering from National Cheng Kung University. His current research interests include information security, cryptography, electronic commerce and mobile communications.

Vincent S. Tseng received the Ph.D. degree in computer science from National Chiao Tung University, Taiwan, R.O.C., in 1997. From January 1998 to July 1999, he was a Postdoctoral Research Fellow in the Computer Science Division, University of California, Berkeley. Since August 1999, he has been on the faculty in the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan. Since February 2004, he has also served as the director for Department of Medical Informatics in National Cheng Kung University Hospital, Taiwan. He has a wide variety of research specialties covering data mining, Web technology, and biomedical informatics. He has published a number of research papers in referred journals and international conferences. Dr. Tseng is a member of ACM and honorary member of the Phi Tau Phi Society. He also served as program committee/chairs for a number of international conferences related to data mining and biomedical informatics.