

## On the Design of RSA With Short Secret Exponent

HUNG-MIN SUN, WU-CHUAN YANG\* AND CHI SUNG LAIH\*

*Department of Computer Science and Information Engineering  
National Cheng Kung University  
Tainan, 701 Taiwan*

*E-mail: hmsun@mail.ncku.edu.tw*

*\*Department of Electrical Engineering  
National Cheng Kung University  
Tainan, 701 Taiwan*

*E-mail: wcyang77@ms32.hinet.net*

*E-mail: laihcs@eembox.ncku.edu.tw*

Based on continued fractions Wiener showed that a typical RSA system can be totally broken if its secret exponent  $d < N^{0.25}$  where  $N$  is the RSA modulus. Recently, based on lattice basis reduction, Boneh and Durfee presented a new short secret exponent attack which improves Wiener's bound up to  $d < N^{0.292}$ . In this paper we show that it is possible to use a short secret exponent which is lower than these bounds while not compromising the security of RSA, provided that  $p$  and  $q$  differ in size and are large enough to defend against factoring algorithms. As an example, an RSA system with  $d$  of 192 bits,  $p$  of 256 bits, and  $q$  of 768 bits is secure against all the existing short secret exponent attacks. On the other hand, in order to balance between and minimize the overall computation of encryption and decryption, we propose a secure variant of RSA such that both  $e$  and  $d$  are the same size, e.g.,  $\log_2 e \approx \log_2 d \approx 568$  for a 1024-bit RSA modulus. Moreover, a generalization of this variant is presented for designing the RSA system with  $\log_2 e + \log_2 d \approx (\log_2 N) + l_k$  where  $l_k$  is a predetermined constant, e.g., 112. Compared with a typical RSA system in which  $e$  is the same order of magnitude as  $N$  if  $d$  is first selected, these variants of RSA have the advantage that the overall computation can be significantly reduced. As an example, we can construct a secure RSA system with  $p$  of 256 bits,  $q$  of 768 bits,  $d$  of 256 bits, and  $e$  of 880 bits.

**Keywords:** data encryption and cryptography, RSA, short secret exponent, continued fraction, lattice basis reduction

### 1. INTRODUCTION

The concept of the public-key cryptosystem was proposed by Diffie and Hellman [6] in 1976. Since then, a number of public-key cryptosystems have been proposed to realize the notion of public-key cryptosystems. Many of these systems are either insecure due to some known attacks or impractical due to large key or low information rate. The RSA public-key cryptosystem was invented by Rivest, Shamir, and Adleman [21] in 1978. Since then, the RSA system has been the best known and most widely accepted

---

Received October 12, 1999; accepted June 5, 2000.  
Communicated by Hsu-Chun Yen.

public key cryptosystem. Usually, the RSA system is deployed in application systems for providing privacy and/or ensuring authenticity of digital data. Hence, many practical issues have been considered when implementing RSA, e.g., how to reduce the storage requirement for RSA modulus [17, 29], how to use short public exponent for reducing the encryption execution time (or signature-verification time) [2-4, 10, 11], and how to use short secret exponent for reducing the decryption execution time (or signature-generation time) [1, 30, 31].

In this paper, we are interested in the use of short secret exponent because it is particularly advantageous when there is a large difference in computing power between two communicating devices. For example, it would be desirable for a smart card to have a short secret exponent in order to speedup the decryption or the generation of signatures in the smart card, and for a larger computer to have a short public exponent in order to speedup the encryption or the verification of signatures required in the smart card. We are also interested in the use of balanced and minimized public and secret exponents that have approximately equal lengths and are as short as possible. The primary motivation for this is to satisfy the requirement of those applications whose computing power of two communicating devices is approximately equal. In particular, it is advantageous when a sequence of encryptions and decryptions (or signature generations and verifications) are required to run synchronously, i.e., no party is idle during communication. Inspired by the above concept, we are also interested in balancing and minimizing the encryption time and the decryption time when there is a difference in computing power between two communicating devices. Compared to a balanced system, it is that if more computation is performed during encryption, then less will be required for decryption, and vice versa. Therefore, there should be a trade-off between encryption and decryption, such that the overall computation is constant. Consequently, our concern is how to reduce the overall computation and how to distribute the computational load between encryption and decryption. If the distributed computation is roughly proportional to the computing power of the two communicating devices, we can balance the time required for encryption and decryption even if there is a difference in computing power between these two devices.

We first describe a simplified version of the RSA system. Let  $N = pq$  be the product of two large primes. If both  $p$  and  $q$  are 512 bits long, then  $N$  is about 1024 bits long. Let  $e$  and  $d$  be two integers satisfying  $ed = 1 \pmod{\phi(N)}$ , where  $\phi(N) = (p-1)(q-1)$  is the Euler totient function of  $N$ . Here we call  $N$  the *RSA modulus*,  $e$  the *public exponent*, and  $d$  the *secret exponent*. The public key is the pair  $(N, e)$  and the secret key is  $d$ . For simplicity, we assume the owner of the secret key is Alice. To provide privacy, one can encrypt a message  $M$  into a ciphertext  $C$  by:  $C = M^e \pmod N$ , while only Alice can decrypt the ciphertext  $C$  into the plaintext  $M$  by:  $M = C^d \pmod N$ . To ensure authenticity of digital data, only Alice can sign a document  $M$  to obtain a signature  $S$  by:  $S = M^d \pmod N$ , while one can verify the validity of Alice's signature  $S$  on  $M$  by checking if  $M = C^e \pmod N$  satisfies a predetermined redundancy scheme.

For a fixed modulus size, the RSA encryption or decryption time is roughly proportional to the number of bits in the exponent. 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 short public exponent attacks [11]. The most powerful attack on short public exponent is due to Coppersmith, Franklin, Patarin and Reiter [4]. Under their attack, the RSA system is

insecure for all public exponents of length up to around 32 bits. Hence, it is suggested that public exponents be longer than 32 bits. Note that these short public exponent attacks succeed only in the encryption of the RSA primitive. Furthermore, they do not work if sufficient random padding is used, such as in the RSA with the protection of the standards PKCS#1 v2.0 or IEEE P1363.

On the other hand, to reduce the decryption time (or the signature-generation time), one may also wish to use a small secret exponent  $d$ . Unfortunately, based on the convergents of the continued fraction expansion of a given number, Wiener [31] showed that the RSA system can be totally broken if  $d < N^{0.25}$ . Verheul and Tilborg [30] proposed an extension of Wiener's attack that allows the RSA system to be broken when  $d$  is a few bits longer than  $0.25 \log_2 N$ . For  $d > N^{0.25}$ , their attack need do an exhaustive search for about  $2t+8$  bits, where  $t = \log_2(d/N^{0.25})$ . If  $t = 20$  (which corresponds to an order of magnitude  $2^{48}$ ) is feasible for doing an exhaustive search, then the RSA system with  $d < 2^{20} N^{0.25}$  is insecure. Thus, this gives a 20-bit improvement on Wiener's bound. Recently, based on lattice basis reduction, Boneh and Durfee [1] proposed a new attack on the use of short secret exponent. They improved Wiener's bound up to  $N^{0.292}$ , i.e., the RSA system can be totally broken if  $d < N^{0.292}$ . This gives a 43-bit improvement on Wiener's bound if  $N$  is the size of 1024 bits. In general, the use of short secret exponent encounters a more serious security problem than the use of short public exponent.

In this paper, we show that it is possible to use a short secret exponent which is below both Wiener's bound and Boneh and Durfee's bound while not compromising the security of RSA provided that  $p$  and  $q$  differ in size and are large enough to defend against the factoring algorithms based on elliptic curves. As an example, when  $p$  is 256 bits long and  $q$  is 768 bits, then  $d$  of 192 bits is large enough to defend against the existing short secret exponent attacks. In this study of balanced and minimized public and secret exponents, we propose a secure variant of RSA such that  $e$  and  $d$  are of the same size, e.g.,  $\log_2 e \approx \log_2 d \approx 568$  for a 1024-bit RSA modulus. After analyzing of the proposed RSA variant according to the ways of attacking short secret exponent RSA, we conclude that the proposed scheme is secure enough to defeat all existing short secret exponent attacks. Finally, the trade-off between the lengths of the secret exponent and public exponent is analyzed. We show that it is possible to design a secure RSA system with  $\log_2 e + \log_2 d \approx \log_2 N + l_k$  where  $l_k$  is a predetermined constant, e.g., 112. Compared with a typical RSA system in which  $e$  is of the same order of magnitude as  $N$  if  $d$  is first selected, these variants of RSA have the advantage that the overall computations can be significantly reduced. As an example, we can construct a secure RSA system with  $p$  of 256 bits,  $q$  of 768 bits,  $d$  of 256 bits, and  $e$  of 880 bits.

The remainder of this paper is organized as follows. In section 2, we review some well-known attacks on the use of short secret exponent. In section 3, we propose and analyze a construction of RSA system to defend against these types of attacks. In section 4, we present a variant of RSA in which the length of the secret exponent and the public exponent can be balanced and minimized. In section 5, the trade-off between the lengths of the secret exponent and public exponent is analyzed. Finally, section 6 gives our conclusions.

## 2. OVERVIEW OF PREVIOUS WORK

Because the security analysis of our schemes is related to Wiener's attack [31], Verheul and Tilborg's attack [30], and Boneh and Durfee's attack [1] on the use of short secret exponent, we briefly review these attacks. Additionally, we introduce the basic concept of unbalanced RSA which was proposed by Shamir [24].

### 2.1 Wiener's Attack and its Extension on Short Secret Exponent

Wiener [31] proposed a clever attack on the use of a small  $d$  in the typical RSA system. His attack is based on approximations using continued fractions to find the numerator and denominator of a fraction in polynomial time when a sufficiently close estimate of the fraction is known. He showed that the RSA system can be totally broken if the secret exponent is up to approximately one-quarter as many bits as the modulus and if  $p$  and  $q$  are of approximately the same size. For simplicity, we slightly modify Wiener's attack in the following way. Let  $ed = k\phi(N)+1$  in a typical RSA system. Hence  $\gcd(d,k) = 1$ . We can rewrite this equation as:  $ed = k(N - (p + q) + 1) + 1$ . Therefore,

$|\frac{e}{N} - \frac{k}{d}| = \delta$ , where  $\delta = \frac{k}{d} \frac{p+q-1-\frac{1}{k}}{N}$ . It is known that for a rational number  $x$  such that  $|x - \frac{B}{A}| < \frac{1}{2A^2}$ , where  $\gcd(A, B)=1$ ,  $\frac{B}{A}$  can be obtained as convergent of the continued fraction expansion of  $x$ . (For further discussion of continued fractions, we refer the reader to [31]). As pointed out by Pinch [20], if  $p < q < 2p$  and  $d < \frac{1}{3} N^{0.25}$ , then  $p+q-1 < 3\sqrt{N}$  and  $k < d < \frac{1}{3} N^{0.25}$ . Therefore,  $|\frac{e}{N} - \frac{k}{d}| \leq \frac{1}{dN^{0.25}} < \frac{1}{3d^2} < \frac{1}{2d^2}$ . Thus  $\frac{k}{d}$  can be found because it is one of the  $\log N$  convergents of the continued fraction for  $\frac{e}{N}$ .

The extension of Wiener's attack, proposed by Verheul and Tilborg [30], basically follows Wiener's approach except that they proposed a more general method to compute the convergents of the continued fraction expansion of the same number as in Wiener's attack up to the point where the denominator of the convergent exceeds approximately  $N^{0.25}$ . For  $d > N^{0.25}$ , their attack need do an exhaustive search for about  $2t+8$  bits, where  $t = \log_2(d/N^{0.25})$ . Because Verheul and Tilborg's attack is not directly related to our work, we omit reviewing the details of their attack here.

### 2.2 Boneh and Durfee's Attack on Short Secret Exponent

Based on solving the small inverse problem, Boneh and Durfee [1] proposed a new attack on short RSA secret exponent, which leads to a tighter bound than that proposed by Wiener. They concluded that if  $e \approx N$  and  $d < N^{0.292}$ , then the secret exponent  $d$  can be found efficiently. In a typical RSA system,  $ed = k\phi(N)+1$ . So,  $ed = k((N+1)-(p+q))+1$ . Let  $A=N+1$ ,  $s=-(p+q)$ , and  $t=k$ . Then  $ed + t(A+s)=1$ . Thus,  $t(A+s) \equiv 1 \pmod{e}$ . If both  $t$  and  $s$  are much smaller than  $e$ , the problem can be viewed as follows:

given an integer  $A$ , find an element close to  $A$  whose inverse modulo,  $e$ , is small. This problem is usually referred as the *small inverse problem*. Let  $e \approx N^\alpha$  and  $d < N^\beta$ . So far, Boneh and Durfee have shown that if  $\beta < \frac{7}{6} - \frac{1}{3}(1+6\alpha)^{1/2}$ , then the small inverse problem can be solved. Consequently, RSA is insecure whenever  $d < N^{0.285}$  (which can be slightly improved up to  $N^{0.292}$ ) if  $\alpha = 1$ .

### 2.3 Unbalanced RSA System

It is generally accepted that RSA moduli are composed of two large primes of the same size. Shamir [24] proposed a variant of the typical RSA, called *unbalanced RSA*, in which the two primes differ widely in size, e.g.,  $\log_2 q = 10 \cdot \log_2 p$ . His motivation is to provide higher security without increasing computational cost.

In general, all the existing factoring algorithms for breaking RSA can be classified into two types: algorithms whose running time depends on the smaller factor  $p$ , and algorithms whose running time depends on the size of the modulus  $N$ . The fastest factoring algorithm of the first type is based on *elliptic curves*. Its asymptotic running time is  $\exp(O((\log_2 p)^{1/2}(\log_2 \log_2 p)^{1/2}))$ . This algorithm is usually referred as the *elliptic curve method* (ECM). So far, the largest factor that has ever been found in practice with this algorithm is about 53 digits ( $\approx 176$  bits) long [8]. Therefore, if we choose  $p$  to be larger than 256 bits, the elliptic curve method becomes infeasible.

The fastest factoring algorithm of the second type is based on the *general number field sieve* (GNFS). Its asymptotic running time is  $\exp(O(\log_2 N)^{1/3}(\log_2 \log_2 N)^{2/3})$ . So far, the largest RSA modulus that has ever been factored with this algorithm is 130 digits ( $\approx 432$  bits) long [5]. Recently, the largest RSA modulus that has ever been factored with line sieving and lattice sieving is 155 digits ( $\approx 512$  bits)[32].

In light of the rapid development of computer techniques and factoring algorithms, it is clear that the standard 512-bit RSA modulus no longer provides adequate security and must be significantly increased. Generally, for a large RSA modulus the GNFS attack is much more efficient than the ECM attack. Therefore, there is no need to increase the sizes of the RSA modulus and its prime factors at the same rate.

At the Eurocrypt'99 rump session, Shamir [25] announced his design for a special hardware device, called "TWINKLE" device which can execute sieve-based factoring algorithms approximately two to three orders of magnitude faster than a conventional fast PC. If the device can be implemented efficiently, this new technique will increase the size of factorable numbers by 100 to 200 bits for a GNFS attack. Basically, RSA system is secure provided that the smaller factor  $p$  and the RSA modulus  $N$  are large enough to make an ECM attack and a GNFS attack infeasible respectively.

Gilbert *et al.* [9] have pointed out that Shamir's unbalanced RSA suffers from some weaknesses. However, these weaknesses come from decrypting only modulo  $p$  (and thus limiting the plaintexts to integers smaller than  $p$ ). Our schemes proposed in this paper do not suffer from the same weaknesses.

Note that some fast and practical public-key cryptosystems [14, 19, 28], which rely on the difficulty of factoring numbers of the type  $p^2q$ , were recently proposed. These cryptosystems also use the same concepts for making the factors short, but at the same time, large enough so that an ECM attack is infeasible.

### 3. RSA WITH SHORT SECRET EXPONENT

In this section, we propose an unbalanced RSA system such that the use of short secret exponent is still secure against all existing short secret exponent attacks and their extension. We show that when  $p$  is the size of 256 bits and  $q$  is the size of 768 bits, then using 192 bits for  $d$  is sufficient.

#### 3.1 The Proposed Scheme (Scheme I)

We propose to construct the unbalanced RSA as follows:

- Step 1. Randomly select a prime  $p$  and a prime  $q$  ( $p < q$ ) such that  $p$  and  $N$  is large enough to make an ECM attack and a GNFS attack infeasible, respectively, e.g.,  $p$  and  $q$  are 256 bits and 768 bits long. Therefore,  $N$  is about 1024 bits long.
- Step 2. Randomly select a short secret exponent  $d$  such that  $\log_2 d + \log_2 p > \frac{1}{3} \log_2 N$  (see Section 3.4) and  $d > 2^\gamma p^{0.5}$ , where  $\gamma$  is a security parameter, e.g.,  $\gamma = 64$ ; hence,  $d$  is 192 bits long. Note that it is necessary that  $\gamma$  satisfies the following inequality:  $32\alpha\gamma \log_e 2 \gg 3(1-\alpha-2\gamma \log_e 2)^2$ , where  $\alpha \approx \log_e q$ . Here  $\log_e$  denotes the logarithm with base  $e$ , the public exponent. We give the details in Section 3.3.
- Step 3. Find  $e$  such that  $ed = 1 \pmod{\phi(N)}$ , where  $\phi(N) = (p-1)(q-1)$ . Generally,  $e$  will be of the same order of magnitude as  $\phi(N)$ . Here we assume  $e \geq \phi(N)/2 + 1$  (the probability of occurrence of this case is 1/2). If not, we repeat Step 2 again.

It is clear that this construction leads to a short secret exponent, e.g., a 192-bit  $d$  for a 1024-bit RSA modulus, which is far below the lower bounds proposed by Wiener (256 bits [31]) and by Boneh and Durfee (299 bits [1]). Note that if  $\gamma \approx 0.5 \log_2 p$ , to our best knowledge, no information can be obtained to break the resulting RSA system until now. The details are explained in Section 3.3. So, the RSA system with  $p$  of 256 bits,  $q$  of 768 bits,  $d$  of 192 bits (due to 128-bit  $\gamma$ ) and  $e$  of 1024 bits is quite secure.

#### 3.2 Defending Against Wiener's Attack and its Extension

Because  $d > 2^\gamma p^{0.5}$ , it is clear that  $\frac{1}{d} > 2^{2\gamma} \frac{1}{d^2}$ . Since  $ed - k\phi(N) = 1$ ,  $k\phi(N) = ed - 1$ , so we obtain  $\frac{k}{d} = \frac{e - \frac{1}{d}}{\phi(N)} \geq \frac{\frac{\phi(N)^p}{2} + 1 - \frac{1}{d}}{\phi(N)} \geq \frac{\frac{\phi(N)}{2}}{\phi(N)} \geq \frac{1}{2}$ . Thus  $|\frac{e}{N} - \frac{k}{d}| = \frac{k}{d} \frac{p+q-1-\frac{1}{k}}{N} > \frac{k}{d} \frac{q}{N} = \frac{k}{d} \frac{1}{p} > \frac{1}{2} 2^{2\gamma} \frac{1}{d^2} = 2^{2\gamma} \frac{1}{2d^2} \gg \frac{1}{2d^2}$ . If  $\gamma$  is sufficiently large, the value  $|\frac{e}{N} - \frac{k}{d}|$  will be much larger than  $\frac{1}{2d^2}$ . Thus, Wiener's attack does not apply to Scheme I.

### 3.3 Defending Against Boneh and Durfee's Attack

Following Boneh and Durfee's approach, let  $A = N + 1$ ,  $s = -(p+q)$ , and  $t = -k$ . Thus  $t(A+s) = 1 \pmod{e}$ . Let  $|s| < e^\alpha$  and  $|t| < e^\beta$ . The sufficient condition for solving the small inverse problem is:  $4\alpha(2\beta + \alpha - 1) < 3(1 - \beta - \alpha)^2$ . The derivation of this condition is given in Appendix A.

In our construction  $e \approx N$ ,  $q \approx |s| \approx e^\alpha$ ,  $d \approx |k| \approx |t| \approx e^\beta$ , giving  $p = \frac{N}{q} \approx \frac{e}{e^\alpha} \approx e^{1-\alpha}$ .

It follows that  $d \approx 2^\gamma p^{0.5} \approx 2^\gamma e^{0.5(1-\alpha)}$ . Let  $2^\gamma \approx e^{\gamma'}$ , i.e.,  $\gamma' \approx \gamma \log_e 2$ , giving  $d \approx e^{\gamma'+0.5(1-\alpha)}$ , and  $2\beta \approx 2\gamma'+1-\alpha$ . This allows the sufficient condition for solving the small inverse problem to be reduced to  $32\alpha\gamma' < 3(1 - \alpha - 2\gamma')^2$ . In order to defend against Boneh and Durfee's attack, it is necessary that  $\gamma$  is large enough so that the following inequality holds:  $32\alpha\gamma \log_e 2 \gg 3(1 - \alpha - 2\gamma \log_e 2)^2$ . As an example, we assume  $p$ ,  $q$ ,  $\gamma$  and  $d$  are 256 bits, 768 bits, 64 bits, and 192 bits long, respectively. Thus,  $\alpha = 0.75$  and  $\beta = 0.1875$ . It is clear that  $4\alpha(2\beta + \alpha - 1) = 0.375 \gg 3(1 - \beta - \alpha)^2 = 0.117186$ , and so Boneh and Durfee's attack can not succeed.

An important observation proposed by Boneh and Durfee [1] is that the unique solution of the small inverse problem encodes enough information to find  $d$ . Therefore, a strong defense to Boneh and Durfee's attack is to make the small inverse problem fail to have a unique solution. This is why Boneh and Durfee believed that a typical RSA with  $d \approx N^{0.5}$  is strongly secure against short secret exponent attacks. If we let  $\gamma$  be a few bits larger than  $0.5 \log_2 p$ , then  $d > p$ . Without loss of generality, assume  $|t| \approx d > p \approx e^{1-\alpha}$ , then  $|t| > e^{1-\alpha}$ . The resulting small inverse problem is:  $t(A + s) = 1 \pmod{e}$ , where  $|t| > e^{1-\alpha}$  and  $|s| \approx e^\alpha$ , will no longer have a unique solution. As a result, if  $d \approx p$ , the resulting RSA is strongly secure against Boneh and Durfee's attack, even if their attack can be up to  $d < N^{0.5}$ .

### 3.4 Defending Against the Cubic Attack

Here we consider a kind of attack, named the cubic attack.

Because  $ed = k(p-1)(q-1) + 1$  and  $N = pq$ , we can obtain the following system of modular equations:

- (1)  $k(p-1)(q-1) + 1 = 0 \pmod{e}$
- (2)  $pq = N \pmod{e}$ .

Combining (1) and (2), we can obtain the following cubic equation in two variables  $k$  and  $p$ :

- (3)  $k(p-1)(N-p) + p = 0 \pmod{e}$

Coppersmith [2] has shown how to solve such cubic equations heuristically if  $\log_2 k + \log_2 p < \frac{1}{3} \log_2 e$ . To defend against Coppersmith's attack, we need the constraint:  $\log_2 d + \log_2 p > \frac{1}{3} \log_2 N$  because  $\log_2 k \approx \log_2 d$  and  $\log_2 e \approx \log_2 N$  in Scheme I. On the other hand, if one can know the exact value of  $k$ , then equation (3) can be reduced to a quadratic equation in a single variable  $p$ , and hence, can be solved

provided that  $e$  either is prime, or can be factored and doesn't have too many prime factors. Obviously, we must make  $k$  unknown to an attacker. In Scheme I, because  $k$  is of the same order of magnitude as  $d$ , it is large enough to make an exhaustive search infeasible. This makes Scheme I secure against the cubic attack.

#### 4. RSA WITH BALANCED PUBLIC EXPONENT AND SECRET EXPONENT

For a typical RSA, it is almost impossible to create a pair  $(e, d)$  such that both are simultaneously much shorter than  $\phi(N)$ . This is because if  $d$  is randomly chosen first, then  $e$  will be of the same order of magnitude as  $\phi(N)$ , and vice versa. In this section, we are interested in constructing RSA with balanced and minimized public and secret exponents, such that both are approximately  $(\frac{1}{2}\log_2 N + 56)$  bits long, without compromising the security of RSA.

##### 4.1 Basic Theorems

**Theorem 1.** If  $a$  and  $b$  are relatively prime, we can find integers  $u$  and  $v$  such that

$$au - bv = 1.$$

*Proof:* This is a well-known theorem. For simplicity, we omit the proof and refer the reader to [12]. In addition, Euclid's algorithm [12] can be used to find adaptive integers  $u$  and  $v$ . □

In the following, we give a generalization of Theorem 1 as Theorem 2.

**Theorem 2.** Let two integers  $a, b > 1$ . If  $\gcd(a, b) = 1$ , 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  $h \geq 1$ .

*Proof:* See Appendix B.

##### 4.2 The Proposed Scheme (Scheme II)

Traditionally, when constructing RSA,  $p$  and  $q$  are selected first. After that, either select the secret exponent  $d$  and then determine the public exponent  $e$ , or vice versa. Thus either  $e$  or  $d$  is of the same order of magnitude as  $\phi(N)$ . In the following we propose a new construction of RSA such that the length of the public exponent  $e$  and the length of the secret exponent  $d$  can be balanced and minimized. Departing from traditional constructions, we first select  $p$  and  $d$ , and then determine  $e$  and  $q$ .

We assume  $p$  and  $q$  are approximately  $(\frac{1}{2}\log_2 N - 112)$  and  $(\frac{1}{2}\log_2 N + 112)$  bits long respectively. Here we assume that  $p$  and  $N$  are large enough to make both an ECM

attack and a GNFS attack infeasible, e.g.,  $p$  and  $N$  are about 400 bits and 1024 bits long, respectively. Our construction is as follows:

- Step 1. Randomly select a prime number  $p$  of  $(\frac{1}{2}\log_2 N - 112)$  bits.
- Step 2. Randomly select a number  $k$  of 112 bits.
- Step 3. Randomly select a number  $d$  of  $(\frac{1}{2}\log_2 N + 56)$  bits such that  $\gcd(k(p-1), d) = 1$ .
- Step 4. Based on Theorem 2, we can uniquely determine two numbers  $u'$  and  $v'$  such that  $du' - k(p-1)v' = 1$ , where  $0 < u' < k(p-1)$  and  $0 < v' < d$ .
- Step 5. If  $\gcd(v'+1, d) \neq 1$ , then go to Step 3.
- Step 6. Randomly select a number  $h$  of 56 bits, compute  $u = u' + hk(p-1)$  and  $v = v' + hd$ .
- Step 7. If  $v+1$  isn't a prime number, then go to Step 6.
- Step 8. Let  $e = u$ ,  $q = v + 1$ , and  $N = pq$ , then  $p$ ,  $q$ ,  $e$ ,  $d$ , and  $N$  are the parameters of RSA.

In this construction  $e$  and  $d$  satisfy  $ed = k(p-1)(q-1) + 1 = k\phi(N) + 1$ . Therefore, the equation  $ed = 1 \pmod{\phi(N)}$  still holds, as a typical RSA. Obviously, both  $e$  and  $d$  obtained from this construction are approximately  $(\frac{1}{2}\log_2 N + 56)$  bits long, and  $p$  and  $q$  are approximately  $(\frac{1}{2}\log_2 N - 112)$  bits and  $(\frac{1}{2}\log_2 N + 112)$  bits long respectively. As an example, if  $\log_2 N \approx 1024$ , then  $d$  is 568 bits long,  $p$  is 400 bits long,  $e$  is about 568 bits long and  $q$  is about 624 bits long. A concrete example for this case is given in Appendix C. In order to measure the efficiency of the proposed scheme, we ran some experiments to test the average times required to find a suitable  $h$  in Step 6 for obtaining a prime  $q$ . Upon testing 100 samples, we find that on average we need to try 487.48 times for Step 6 when  $N$  is of 1024 bits long. A comparative result is 566.31 times of selecting a random number of 624 bits and testing whether the number is a prime. This shows that both have roughly the same cost for obtaining a prime  $q$ . Note that in Step 5, if  $\gcd(v'+1, d) \neq 1$ , then it is impossible to find  $h$  such that  $v' + hd + 1$  is a prime. In addition, the prime  $p$  generated in Step 1 can be determined arbitrarily, e.g., by selecting a strong prime  $p$ , but the prime  $q$  generated in Step 8 cannot. Fortunately, for an RSA key the requirement that  $p$  and  $q$  be strong primes is no longer needed due to [22, 26, 27].

Note that compared with the RSA using CRT-based implementations, Scheme II apparently does not provide better efficiency. However the CRT-based RSA needs to keep more secrets ( $p$  and  $q$ ) than the typical RSA. Moreover, the CRT-based RSA usually incurs some additional security problems [15], and even some error detection techniques are applied to it.

### 4.3 Defending Against Wiener's Attack and its Extension

The proposed RSA (Scheme II) in Section 4.2 is very different from a typical RSA. Generally, in a typical RSA, if  $d$  is randomly chosen first, then  $e$  will be of the same order of magnitude as  $\phi(N)$ , and vice versa. This implies that a typical RSA has the relationship  $ed = k\phi(N)+1$ , where  $k$  is of the same order of magnitude as  $\min\{e, d\}$ . In Scheme II, the relationship among these parameters is  $ed = k\phi(N) + 1$  where  $k$  is of 112 bits.

Here we examine the security of Scheme II following the line of attack proposed by Wiener on short RSA secret exponent.

It is clear that  $|\frac{e}{N} - \frac{k}{d}| = \frac{k}{d} \frac{p+q-1-\frac{1}{k}}{N} > \frac{k}{d} \frac{q}{N} = \frac{k}{d} \frac{1}{p}$ . Without loss of generality, we assume that  $k > 2^{111}$ ,  $2^{-113} N^{0.5} < p < 2^{-112} N^{0.5}$  and  $2^{55} N^{0.5} < d < 2^{56} N^{0.5}$ . It follows that  $\frac{1}{p} > \frac{2^{112}}{N^{0.5}}$  and  $\frac{1}{N^{0.5}} > 2^{55} \frac{1}{d}$ , and  $\frac{k}{d} \frac{1}{p} > 2^{111} \frac{1}{d} \frac{2^{112}}{N^{0.5}} > 2^{279} \frac{1}{2d^2} \gg \frac{1}{2d^2}$ . Now, we can see that  $|\frac{e}{N} - \frac{k}{d}|$  will be much larger than  $\frac{1}{2d^2}$ , and Wiener's attack does not apply to Scheme II.

#### 4.4 Defending Against Boneh and Durfee's Attack

As for Section 3.3, the sufficient condition for solving the small inverse problem is:  $4\alpha(2\beta + \alpha - 1) < 3(1 - \beta - \alpha)^2$ . Due to the difficulty of obtaining a general proof, we only show that Scheme II with  $N$  of 1024 bits is secure against Boneh and Durfee's Attack. For Scheme II, if  $\log_2 N \approx 1024$ , then  $d$  is 568 bits long,  $p$  is 400 bits,  $e$  is about 568 bits and  $q$ , about 624 bits. Thus  $\alpha = \frac{624}{568}$  and  $\beta = \frac{112}{568}$ . It is clear that  $4\alpha(2\beta + \alpha - 1) = 2.1664 \gg 3(1 - \beta - \alpha)^2 = 0.2625$ . So, Boneh and Durfee's attack doesn't apply to Scheme II.

#### 4.5 Defending Against the Cubic Attack

Here we refer to Section 3.4 for the cubic attack. In Scheme II, because  $\log_2 k + \log_2 p = \frac{1}{2} \log_2 N \gg \frac{1}{3} \log_2 e$ , Coppersmith's attack cannot work here. In addition, because  $k$  is 112 bits long, it is large enough to make an exhaustive search infeasible. Hence, Scheme II is also secure against the cubic attack.

### 5. TRADE-OFF BETWEEN PUBLIC EXPONENT AND SECRET EXPONENT

In general, given  $p$ ,  $q$ , and a short secret exponent  $d$ , it is very likely that the corresponding public exponent  $e$  is of the same order of magnitude as  $\phi(N)$ . But from Section 4, we know that it is possible for us to use median public and secret exponents in a RSA system so that the overall computation required for encryption and decryption are minimized and balanced without compromising with the security. Therefore, one may be desire to have secret and public exponents which differ in size, but have the overall computation still minimized, e.g.,  $d \approx N^{0.25}$  and  $e \approx N^{0.86}$ . In this section, we investigate the trade-off between the length of the public exponent and the length of the secret exponent while maintaining the minimization of the overall computation.

### 5.1 The Proposed Scheme (Scheme III)

Generalizing Scheme II, we give an efficient construction for RSA such that  $\log_2 e + \log_2 d \approx \log_2 N + l_k$ , where  $l_k$  is a predetermined constant.

- Step 1. Randomly select a prime number  $p$  of length  $l_p$  ( $l_p < \frac{1}{2} \log_2 N$ ) such that it is large enough to make an ECM attack infeasible, e.g.,  $l_p = 256$ .
- Step 2. Randomly select a number  $k$  of length  $l_k$ , e.g.,  $l_k = 112$ .
- Step 3. Randomly select a number  $d$  of length  $l_d$  such that  $\gcd(k(p-1), d) = 1$ , e.g.,  $l_d = 256$ .
- Step 4. Based on Theorem 2, we can uniquely determine two numbers  $u'$  and  $v'$  such that  $du' - k(p-1)v' = 1$ , where  $0 < u' < k(p-1)$  and  $0 < v' < d$ .
- Step 5. If  $\gcd(v'+1, d) \neq 1$ , then go to Step 3.
- Step 6. Randomly select a number  $h$  of length  $\log N - l_p - l_d$ , then compute  $u = u' + hk(p-1)$  and  $v = v' + hd$ .
- Step 7. If  $v+1$  is not a prime number, go to Step 6.
- Step 8. Let  $e = u$ ,  $q = v + 1$ , and  $N = pq$ , then  $p$ ,  $q$ ,  $e$ ,  $d$ , and  $N$  are the parameters of RSA.

From steps 1-3, we know that  $k$ ,  $p$ , and  $d$  are  $l_k$  bits,  $l_p$  bits, and  $l_d$  bits long. Obviously,  $e$  and  $q$  obtained from the above construction are roughly  $\log N + l_k - l_d$  and  $\log N - l_p$  bits long, respectively. These parameters  $l_k$ ,  $l_p$ , and  $l_d$  must satisfy the following requirements:

- (1)  $l_k \gg l_p - l_d + 1$ .
- (2)  $\alpha$  and  $\beta$  must satisfy:  $4\alpha(2\beta + \alpha - 1) \gg 3(1 - \beta - \alpha)^2$ , where  $\alpha = \frac{\log_2 N - l_p}{\log_2 N + l_k - l_d}$  and  $\beta = \frac{l_k}{\log_2 N + l_k - l_d}$ .
- (3)  $k$  is large enough to make an exhaustive search infeasible, e.g.,  $l_k = 112$ .

We give a detailed discussion for the above requirements in Sections 5.2, 5.3, 5.4, and 5.5, respectively.

As an example, if  $k$ ,  $p$ , and  $d$  are 112 bits, 256 bits, and 256 bits long, then  $e$  is about 880 bits long and  $q$  is about 768 bits long. A concrete example for this case is given in Appendix D. In order to measure the efficiency of the proposed scheme, we also ran some experiments to test the average times required to find a suitable  $h$  in Step 6 for obtaining a prime  $q$ . Upon testing 100 samples, our results indicate that on average we need to try 743.56 times for Step 6. A comparative result is 696.86 times of selecting a random number of 768 bits and testing whether the number is a prime. This shows that both have approximately the same cost in order to obtain a prime  $q$ .

Note that if one wishes to have a smaller public exponent and a larger secret exponent, the one needs only to modify this construction by interchanging the positions of  $e$  and  $d$ , i.e., first fix  $e$  and  $p$  and then determine  $d$  and  $q$ .

### 5.2 Defending Against Wiener's Attack and its Extension

Referring to Section 3.2, it is clear that  $|\frac{e}{N} - \frac{k}{d}| = \frac{k}{d} \frac{p+q-1-\frac{1}{k}}{N} > \frac{k}{d} \frac{q}{N} = \frac{k}{d} \frac{1}{p}$ .

Without loss of generality, we assume that  $k > 2^{l_k-1}$ ,  $2^{l_p-1} < p < 2^{l_p}$  and  $2^{l_d-1} < d < 2^{l_d}$ ,

from which  $\frac{1}{p} > 2^{-l_p}$  and  $2^{-l_d+1} > \frac{1}{d}$ . Obviously,  $\frac{k}{d} \frac{1}{p} > 2^{l_k-l_p-1} \frac{1}{d} = 2^{l_k-l_p} \frac{1}{2d}$ ,

and from requirement (1)  $l_k \gg l_p - l_d + 1$ , we have  $\frac{k}{d} \frac{1}{p} \gg 2^{-l_d+1} \frac{1}{2d} > \frac{1}{2d^2}$ .

$|\frac{e}{N} - \frac{k}{d}|$  is much larger than  $\frac{1}{2d^2}$ , and we can conclude that Wiener's attack does not

apply to Scheme III.

### 5.3 Defending Against Boneh and Durfee's Attack

Because  $k$ ,  $p$ , and  $d$  are  $l_k$  bits,  $l_p$  bits, and  $l_d$  bits long,  $e$  and  $q$  obtained from Scheme III will be roughly  $\log_2 N + l_k - l_d$  bits and  $\log_2 N - l_p$  bits long, respectively. Note that  $q > p$ . Following Boneh and Durfee's approach, let  $A = N + 1$ ,  $s = -(p + q)$ , and  $t = -k$ . Thus,  $t(A + s) = 1 \pmod{e}$ . Let  $|s| < e^\alpha$  and  $|t| < e^\beta$ . Therefore,  $\alpha \approx \frac{\log_2 N - l_p}{\log_2 N + l_k - l_d}$  and  $\beta \approx \frac{l_k}{\log_2 N + l_k - l_d}$ . As described in Section 3.3, to defend against Boneh and Durfee's attack,  $\alpha$  and  $\beta$  must satisfy  $4\alpha(2\beta + \beta - 1) \gg 3(1 - \beta - \alpha)^2$ . As an example, if  $k$ ,  $p$ , and  $d$  are 112 bits, 256 bits, and 256 bits long (hence  $e$  and  $q$  are about 880 bits and 768 bits long), then  $\alpha \approx \frac{768}{880}$  and  $\beta \approx \frac{112}{880}$ . It is clear that  $4\alpha(2\beta + \alpha - 1) = 0.4443 \gg 3(1 - \beta - \alpha)^2 = 0$ .

### 5.4 Defending Against the Cubic Attack

Here we refer to Section 3.4 for the cubic attack. In Scheme III, because  $\log_2 k + \log_2 p > \frac{1}{3} \log_2 N > \frac{1}{3} \log_2 e$ , Coppersmith's attack cannot work here, and besides,  $l_k = 112$  makes an exhaustive search infeasible. Thus, Scheme III is secure against the cubic attack.

**Remark:** In [23], Sakai, Morii, and Kasahara proposed a key generation algorithm for RSA cryptosystem which can make  $\log_2 e + \log_2 d \approx \log_2 N$ . Their scheme has the following properties:

$$(1) \quad ed = \frac{k(p-1)(q-1)}{2g} + 1, \text{ where } g \text{ is a large prime and } g|(p-1), g|(q-1)$$

$$(2) \quad \log_2 p \approx \log_2 q \approx \frac{1}{2} \log_2 N$$

It should be noticed that Wiener [31] has pointed out that making  $g$  large (and hence  $\gcd(p-1, q-1)$  is large) may cause some security problems. For example, one can find  $g$  from  $N-1$  by using factoring algorithms because  $g$  divides  $pq-1 = (p-1)(q-1) + (p-1) + (q-1)$ . If  $g$  is not large enough to defend against an ECM attack, e.g.,  $g$  of 110 bits and 120 bits in Sakai et al.'s scheme,  $g$  can easily be found from  $N-1$ . Even if  $g$  is large enough to defend against an ECM attack, e.g., 250 bits, it is still possible to factor  $N-1$ , and hence, obtain  $g$  because  $N-1$  may possibly contain only some small prime factors excluding  $g$ . Once  $g$  is founded, Wiener's attack will work efficiently [26]. A possible solution to repair their schemes is to make  $g$  much larger and let  $N-1$  contain at least two large prime factors whose product is large enough to prevent factoring. However, in some literature and current practical use, e.g., X9.31, it is usually recommended to make  $\gcd(p-1, q-1)$  small in order to guard against the relevant attacks such as repeat encryption attacks.

## 6. CONCLUSIONS

Our results show that it is possible to use a short secret exponent which is below both Wiener's bound and Boneh and Durfee's bound while not compromising the security of RSA provided that  $p$  and  $q$  differ in size. For example,  $p$  of 256 bits,  $q$  of 768 bits, and  $d$  of 192 bits are large enough to defend against all existing short exponent attacks. We also propose an efficient construction of RSA such that both  $e$  and  $d$  are of the same order of magnitude as  $2^{56}N^{0.5}$ . As an example, a secure RSA with parameters  $e$  of 568 bits,  $d$  of 568 bits,  $p$  of 400 bits, and  $q$  of 624 bits is achievable. Finally, the trade-off between the length of secret exponent and the length of public exponent is analyzed. According our analysis, it is possible to design a secure RSA system with  $\log_2 e + \log_2 d \approx \log_2 N + l_k$  where  $l_k$  is a predetermined constant, e.g., 112, provided that  $p$  and  $q$  differ in size. As an example, we can construct a secure RSA with  $p$  of 256 bits,  $q$  of 768 bits,  $d$  of 256 bits, and  $e$  of 880 bits.

An important observation obtained in this paper is that making the size of  $p$  and  $q$  different enables RSA to defend against all existing short secret exponent attacks. Although this also reduces the strength of RSA against factoring, at present  $p$  of 256 bits is large enough to defend against an ECM attack. The same concept of making the factors short but large enough such that an ECM attack is infeasible can also be found in an unbalanced RSA system [20] and other efficient public-key cryptosystems [14, 19, 28] with the type  $N = p^2q$ .

## ACKNOWLEDGMENTS

We are grateful to Dr. Marc Joye and Prof. Sung-Ming Yen for their valuable comments. This research was supported in part by the National Science Council, Taiwan, under contract NSC-89-2213-E-006-118 and NSC89-2213-E-006-010.

## REFERENCES

1. D. Boneh and G. Durfee, "Cryptanalysis of RSA with private exponent  $d < N^{0.292}$ ," *EUROCRYPT'99*, 1999, LNCS 1592, Springer-Verlag, pp. 1-11.
2. D. Coppersmith, "Finding a small root of a univariate modular equation," in *Proceedings of EUROCRYPT'96*, 1996, LNCS 1070, Springer-Verlag, pp. 155-165.
3. D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities," *Journal of Cryptology*, Vol. 10, 1997, pp. 233-260.
4. D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, "Low-exponent RSA with related messages," in *Proceedings of EUROCRYPT'96*, 1996, LNCS 1070, Springer-Verlag, pp. 1-9.
5. J. Cowie, B. Dodson, R. Elkenbracht-Huizing, A. K. Lenstra, P. L. Montgomery, and J. Zayer, "A world wide number field sieve factoring record: on to 512 bits," in *Proceedings of ASIACRYPT '96*, 1996, LNCS 1163, Springer-Verlag, pp. 382-394.
6. W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. IT-22, 1976, pp. 644-654.
7. S. Cavallar, W. Lioen, H. te Riele, B. Dodson, A. Lenstra, P. Leyland, P. L. Montgomery, B. Murphy, and P. Zimmermann, "Factorization of RSA-140 using the number field sieve," in *Proceedings of ASIACRYPT '99*, Springer-Verlag, 1999, pp. 195-207.
8. ECMNET Project; <http://www.loria.fr/~zimmerma/records/ecmnet.html>
9. H. Gilbert, D. Gupta, A. Odlyzko, and J. J. Quisquater, "Attacks on Shamir's RSA for paranoids," *Information Processing Letters*, Vol. 68, 1998, pp. 197-199.
10. J. Hastad, "On using RSA with low exponent in a public key network," in *Proceedings of CRYPTO'85*, 1986, LNCS, Springer-Verlag, pp. 403-408.
11. J. Hastad, "Solving simultaneous modular equations of low degree," *SIAM Journal of Computing*, Vol. 17, 1988, pp. 336-341.
12. I. N. Herstein, *Topics in Algebra*, Xerox Corporation, 1975.
13. N. Howgrave-Graham, "Finding small roots of univariate modular equations revised," in *Proceedings of Cryptography and Coding*, 1997, LNCS 1355, Springer-Verlag, pp. 131-142.
14. D. Hühnlein, M. J. Jacobson, S. Paulus, and T. Takagi, "A cryptosystem based on non-maximal imaginary quadratic orders with fast decryption," in *Proceedings of EUROCRYPT'98*, 1998, LNCS 1403, Springer-Verlag, pp. 294-307.
15. M. Joye, J. J. Quisquater, S. M. Yen, and M. Yung, "Security paradoxes: how improving a cryptosystem may weaken it," in *Proceedings of the Ninth National Conference on Information Security*, 1999, pp. 27-32.
16. A. Lenstra, H. Lenstra, and L. Lovasz, "Factoring polynomial with rational coefficients," *Mathematische Annalen*, Vol. 261, 1982, pp. 515-534.
17. A. Lenstra, "Generating RSA moduli with a predetermined portion," in *Proceedings of ASIACRYPT'98*, 1998, LNCS 1514, Springer-Verlag, pp. 1-10.
18. L. Lovasz, *An Algorithmic Theory of Number, Graphs and Convexity*, SIAM Publications, 1986.
19. T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Proceedings of EUROCRYPT'98*, 1998, LNCS 1403, Springer-Verlag, pp. 308-318.

20. R. Pinch, "Extending the Wiener attack to RSA-type cryptosystems," *Electronics Letters*, Vol. 31, 1995, pp. 1736-1738.
21. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communication of ACM*, Vol. 21, 1978, pp. 120-126.
22. R. Rivest and R. D. Silverman, "Are strong primes needed for RSA?," in The 1997 RSA Laboratories Seminar series, Seminar Proceedings, 1997.
23. R. Sakai, M. Morii, and M. Kasahara, "New key generation algorithm for RSA cryptosystem," *IEICE Transactions on Fundamentals*, Vol. E77-A, 1994, pp. 89-97.
24. A. Shamir, "RSA for paranoids," *CryptoBytes*, Vol. 1, 1995, pp. 1, 3-4.
25. A. Shamir, "Factoring large numbers with the TWINKLE device," presented at *Eurocrypt '99*, 1999.
26. R. D. Silverman, "Fast generation of random, strong RSA primes," *CryptoBytes*, Vol. 3, 1997, pp. 9-13.
27. R. D. Silverman, "The requirement for strong primes in RSA," RSA Laboratories Technical Note, 1997.
28. T. Takagi, "Fast RSA-type cryptosystem modulo  $p^2q$ ," in *Proceedings of CRYPTO'98*, 1998, LNCS 1462, Springer-Verlag, pp. 318-326.
29. S. A. Vanstone and R. J. Zuccherato, "Short RSA keys and their generation," *Journal of Cryptology*, Vol. 8, 1995, pp. 101-114.
30. E. Verheul and H. van Tilborg, "Cryptanalysis of less short RSA secret exponents," *Applicable Algebra in Engineering, Communication and Computing*, Springer-Verlag, Vol. 8, 1997, pp. 425-435.
31. M. Wiener, "Cryptanalysis of short RSA secret exponents," *IEEE Transactions on Information Theory*, Vol. 36, 1990, pp. 553-558.
32. Factorization of RSA-155, <http://www.rsa.com/rsalabs/html/rsa155.html>, Aug. 1999.

## APPENDIX A: SOLVING THE SMALL INVERSE PROBLEM

Boneh and Durfee introduced a method based on Coppersmith's approach [3], (as presented by Howgrave-Graham [13]) to solve the small inverse problem.

Because Boneh and Durfee only solved special cases for  $x(A+y) = 1 \pmod{e}$  when  $x < e^\beta$  and  $|y| < e^{0.5}$ , and when  $x < 2e^{1+(\beta-1)/\alpha}$  and  $|y| < 2e^{0.5\alpha}$ , we now generalize their method to solve the general case when  $x < e^\beta$  and  $|y| < e^\alpha$ .

The small inverse problem can be restated as follows:

Given a polynomial  $f(x, y) = x(A+y) - 1$ , find an  $(x_0, y_0)$  satisfying  $f(x_0, y_0) = 0 \pmod{e}$  where  $x_0 < e^\beta$ ,  $|y_0| < e^\alpha$ .

For simplicity, we only follow Boneh and Durfee's derivation without introducing the theoretical part of their method. A more detailed discussion is given in [1].

All we need to do is to find a polynomial with a small norm that has  $(x_0, y_0)$  as a root modulo  $e^m$  for some positive integer  $m$ . Given an integer  $m$ , we define the polynomials

$$g_{i,k}(x, y) = x^i f^k(x, y) e^{m-k} \quad \text{and} \quad h_{j,k}(x, y) = y^j f^k(x, y) e^{m-k}, \quad \text{for } k = 0, \dots, m.$$

Here  $g_{i,k}(x, y)$  is called  $x$ -shifts and  $h_{j,k}(x, y)$  is called  $y$ -shifts. A lattice  $L$  can

be spanned by the coefficient vectors corresponding to these polynomials. Our goal is to find a lattice that has sufficiently small vectors and then uses the LLL lattice basis reduction algorithm [14, 16] to find them. Therefore, to solve the small inverse problem, the lattice spanned by these polynomials must have a sufficiently small determinant.

For the spanned lattice by these polynomials, the determinant of the submatrix corresponding to all  $x$  shifts is :  $\det_x = e^{m(m+1)(m+2)/3} \cdot X^{m(m+1)(m+2)/3} \cdot Y^{m(m+1)(m+2)/6}$ .

Similarly, the determinant of the submatrix corresponding to all  $y$  shifts is

$$\det_y = e^{tm(m+1)/2} \cdot X^{tm(m+1)/2} \cdot Y^{t(m+1)(m+t+1)/2}.$$

Let  $X = e^\beta$  and  $Y = e^\alpha$ . Ignoring low order terms, we obtain:

$$\det_x \approx e^{\frac{1}{6}(2+2\beta+\alpha)m^3} \quad \text{and} \quad \det_y \approx e^{\frac{1}{2}(1+\beta+\alpha)tm^2 + \frac{1}{2}\alpha mt^2}.$$

$$\text{Therefore, } \det(L) = \det_x \det_y \approx e^{\frac{1}{6}(2+2\beta+\alpha)m^3 + \frac{1}{2}(1+\beta+\alpha)tm^2 + \frac{1}{2}\alpha mt^2}.$$

The dimension is  $w = (m+1)(m+2)/2 + t(m+1)$ . Ignoring low order terms, we obtain  $w \approx \frac{1}{2}m^2 + tm$ .

To solve the small inverse problem, we must have  $\det(L) < e^{mw}$ . Thus,

$$\frac{1}{6}(2+2\beta+\alpha)m^3 + \frac{1}{2}(1+\beta+\alpha)tm^2 + \frac{1}{2}\alpha mt^2 < \frac{1}{2}m^3 + tm^2.$$

$$\text{Therefore, } (2\beta + \alpha - 1)m^2 + 3(\beta + \alpha - 1)tm + 3\alpha t^2 < 0.$$

The function  $(2\beta + \alpha - 1)m^2 + 3(\beta + \alpha - 1)tm + 3\alpha t^2$  has the maximal value when  $t = \frac{1-\beta-\alpha}{2\alpha}m$ . So, we must make

$$(2\beta + \alpha - 1)m^2 + 3(\beta + \alpha - 1)\left(\frac{1-\beta-\alpha}{2\alpha}\right)m^2 + 3\alpha\left(\frac{1-\beta-\alpha}{2\alpha}m\right)^2 < 0, \text{ which implies}$$

that  $m^2(4\alpha(2\beta + \alpha - 1) - 3(1 - \beta - \alpha)^2) < 0$ . So,  $4\alpha(2\beta + \alpha - 1) < 3(1 - \beta - \alpha)^2$ .

Note that this result means that if  $\alpha$  and  $\beta$  satisfy the above inequality, then the small inverse problem can be solved. The result is derived from the determinant of a lattice. As described in [1], the lattice contains a sub-lattice with a smaller determinant, so the result can be improved. Here we omit the details on this improvement because it is very complex to derive and the improvement seems to be small compared with the original result.

## APPENDIX B: PROOF OF THEOREM 2

**Theorem 2.** Let two integers  $a, b > 1$ . If  $\gcd(a, b) = 1$ , 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  $h \geq 1$ .

**Proof:** Since  $\gcd(a, b) = 1$ ,  $a$  has a unique inverse (modulo  $b$ ). Let  $u_1 = a^{-1} \bmod b$  and  $0 < u_1 < b$ . Define  $v_1 = \lfloor au_1/b \rfloor$ . Hence, it follows that  $au_1 - bv_1 = au_1 \bmod b = 1$ .

Moreover, since  $au_1 > 0$ , we must have  $v_1 \neq 0$ , and since  $bv_1 = au_1 - 1 < ab$ , we have  $v_1 < a$ ; therefore,  $0 < v_1 < a$ . For  $h \geq 2$ , we define  $u_h = u_1 + (h-1)b$  and  $v_h = v_1 + (h-1)a$ , which satisfies  $au_h - bv_h = 1$  where  $(h-1)b < u_h < hb$  and  $(h-1)a < v_h < ha$ .

### APPENDIX C: AN EXAMPLE FOR $p$ OF 400 BITS, $q$ OF 624 BITS, $d$ OF 568 BITS, AND $e$ OF 568 BITS

```

p = 0000cd0a 73cb74b6 27aa29e7 9ba13cab d73f4b67 92abde25 c2dcc2dd 68f7a477
    9cc6f0a0 d5eeea7c 7c740c8c b370a2e1 6112a393
q = 0000807e 4aac8213 62d7d547 4e4dac07 1ea03096 0f13c597 A619a6d7 4c8a3e5b
    dcd00bcb dcfb0758 555f6b4e 23cc4f6a 5221fa87 bfef172d b815a296 4c5c5be7
    61a22fe4 53808fac 0a2fb2d2 548285af
d = 008dc2d0 0c1e3027 e0a43f18 022896a0 35379c76 b1e5577c 71038464 bf9ef9a6
    00bb3aa0 bb4f590d ef8311ab 95282426 7277f349 200c5d67 5e23dc05 9613dcc
    ae0a5dad 1209cc53
e = 00b335b3 9edd0f90 546f4a51 2ec2a0dd 191e1fb0 38f6b5dd b93f5156 7ecdc538
    355a67b6 d7fbbee3 0926925c b0112914 bbe9f4bf a1a61f92 53dfab7e d9c40261
    6fc3d7a8 f77c025f

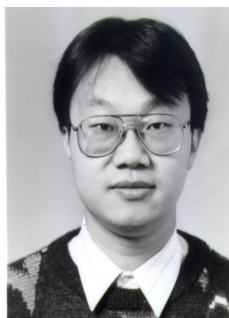
```

### APPENDIX D: AN EXAMPLE FOR $p$ OF 256 BITS, $q$ OF 768 BITS, $d$ OF 256 BITS, AND $e$ OF 880 BITS

```

p = f80dd4da c85afb9d 019d0f24 92c03006 c5baef83 7cfc15eb 2e17b1c1 1fb166e3
q = 96d12784 058456cf 00e17f03 b6402825 00a95a1a 772f7059 ea78ac03 57e49dbf
    feaff1d1 b556e47f 855e8d74 9905753b 12a46068 ce6df746 0e85602c 8f4ed8ac
    ed6b7f21 2fb1d58f ca645447 ae39277d d01e681a e8a630c6 8c158859 c2e4b743
d = bd82175c 6d9bd203 9ce3f83b cdbceb8e 51c82b29 7f4e237d b0eb3518 807c02bf
e = 0000c3b8 1c856425 ff98f54d 605ebe3e 58fd6381 acd328b8 0c4c1d7d ebba6832
    061d6fa7 baa8b814 65a82be5 93cdc56a 21ac87e7 693e97e9 3632dfc7 47572a58
    f3683163 cd312935 bd24a7ac 08204830 1ba73867 da7456d7 f5efcada 715ad9a0
    cec3edd3 e773421b 2c699c42 ef62ebff

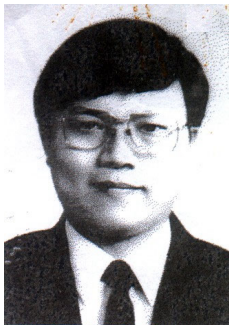
```



**Hung-Min Sun (孫宏民)** received his B.S. degree in applied mathematics from National Chung-Hsing University in 1988, his M.S. degree in applied mathematics from National Cheng-Kung University in 1990, and his Ph.D. degree in computer science and information engineering from National Chiao-Tung University in 1995, respectively. He was an associate professor with the Department of Information Management, Chaoyang University of Technology from 1995 to 1999. Currently he is teaching at the Department of Computer Science and Information Engineering, National Cheng Kung University. His research interests include cryptography, information theory, network security, reliability, and distributed systems.



**Wu-Chuan Yang (楊吳泉)** was born in Tainan, Taiwan, in 1966. He received his B.S. and M.S. degree in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan, in 1988 and 1991 respectively. He has been an instructor in Nan Jeon Junior College of Technique and Commerce, since 1991. Currently, he is working toward his Ph.D. degree in National Cheng Kung University. His research interests include cryptography, algorithm, information and network security.



**Chi-Sung Laih (賴溪松)** was born on June 4, 1956 in Chiayi, Taiwan, Republic of China. He received his B.S., M.S. and Ph.D. degrees all in Electrical Engineering from National Cheng Kung University in 1984, 1986 and 1990, respectively. Since September 1986, he has been on the faculty of the Department of Electrical Engineering at National Cheng Kung University, Tainan, Taiwan, and currently is a professor. From August 1993 to January 1997, he was an adjunct research fellow at Engineering and Technology Promotion Center of the National Science Council of the Republic of China. Currently, he is the director of computer and network center. From February 1997, he was the director of Project Management, office of Research and Development at National Cheng Kung University. From June 1997, he was elected as the Chairman of Chinese Cryptology and Information Security Association (CCISA). His research interests include Cryptology, Information Security, Error Control Codes and Communication Systems. Dr. Laih is a member of IEEE, ACM and IACR. He was the winner of the 1991 and 1997 Acer Long Term Award for Outstanding M.S. Thesis Supervision, the winner of Graduate Team of TI-Taiwan 1994 DSP Design Championship and the winner of 1997 and 1999 Outstanding Paper Award and 1996 of CCISA. He also obtained the 1997-1998 and 1999-2000 Outstanding Research Award of the National Science Council of the Republic of China. He received 1999 Outstanding Talent Award in Information Science, Republic of China.