A Comprehensive Study of Backdoors for RSA Key Generation

Ting-Yu Lin, Hung-Min Sun and Mu-En Wu
Department of Computer Science
National Tsing Hua University, Hsinchu, Taiwan 300
Email: hmsun@cs.nthu.edu.tw[2]; {ting; mn}@is.cs.nthu.edu.tw[1,3]

Abstract

Young and Yung devised various backdoors for RSA key generation. However, due to their backdoor constructing method, the two MSBs of the modulus n generated by their systems have different distribution than that generated by the normal RSA. Thus, someone may observe the abnormal distribution of n to detect the existence of backdoors. Recently, Crepéau and Slakmon further suggested four simple ways to construct RSA backdoor mechanisms. Although their schemes have roughly the same running time as the normal RSA, they are still not Strong-SETUPs. Hence, in this paper, we propose three RSA backdoor cryptosystems. The first two, RSA-SBLT and RSA-SBES, have the same advantages as Crépeau and Slakmon's RSA-HP$_\beta$ but provide new ways to construct the backdoors. As for our third backdoor cryptosystem, RSA-BDH, we exploit the relationship between p and q to devise a backdoor which can solve the problem occurring in Young and Yong's methods successfully. Moreover, we prove that RSA-BDH is a real Strong-SETUP based on DH assumption.

Keywords: RSA, SETUP, Backdoor, Lattice Reduction, Diffie-Hellman

1 Introduction

RSA[15] is a widely used public-key cryptosystem in the world, but should the RSA system in cryptographic devices always be trusted? In general, a cryptographic device remains a black-box such that the users are obliged to trust the internal design of the device and have no access to verify the authenticity and integrity of the software. However, it is possible for a cryptographic device to contain a backdoor mechanism through which users' key information can be leaked to the manufacturer. Note that by "manufacturer" we mean the maker of the cryptographic device.

The backdoor mechanism can give the manufacturer an exclusive advantage to obtain the secret leaked information. However, it can also be employed in the design of "auto-escrowing key" systems [19]. A key escrow system [2][7][9][13] is a system in which third-party agencies, such as law enforcement agencies, would have backdoor keys to read encrypted messages. Basically, the backdoor keys are escrowed among two or more agencies. When the key escrow agencies want to examine some suspicious communication, they can combine their shares of the backdoor keys and recover the information. There are many advantages to apply the backdoor mechanisms to the key escrow systems, including involving no lengthy communications between users and escrow agents, letting users be free to generate their own keys at any time, and keeping the private keys secret even though the device is reverse engineered.

In 1993, Anderson [1] proposed a RSA trapdoor in which p and q are generated according to a specific formula. By holding a secret constant used in the formula, the manufacturer can factor the RSA modulus n and break the system. However, his scheme has been proven to be insecure by Kaliski [12] since anyone can detect the existence of the trapdoor using lattice reduction techniques.

In 1996, Young and Yung [19] presented a notion called "SETUP" (Secretly Embedded Trapdoor with Universal Protection). A SETUP is a mechanism which enables the manufacturer to obtain the user's secret from some stage of the output process of the device in an unnoticeable fashion, yet protects against attacks by others. With a SETUP, the manufacturer has the relative advantage compared to the true attacker to obtain the user's secret.

Young and Yung have presented various backdoors for RSA key generation since 1996 [19][20][21] [22]. In their methods, the manufacturer owns a public and private key pair of some public key system, such as RSA, ElGamal, Rabin, and ECDDH. The manufacturer's public key is used to encrypt the information with which p can be generated and the encrypted information is embedded in the modulus n. Hence, by decrypting the information embedded in the modulus n, the manufacturer can easily derive p and recover the user's private key.

In some of Young and Yung's backdoor mechanisms [20][21][22], even though the device is reverse engineered, anyone except the manufacturer can not distinguish whether a given key is a SETUP key or not in polynomial time. The kinds of SETUP mechanisms are further called Strong-SETUP. However, in the methods, since p and n are randomly generated first and then q is decided by $q = \left\lfloor \frac{n}{p} \right\rfloor$, the two MSBs of n have uniform distribution which does not correspond to the distribution of the two MSBs of the normal n. In the normal RSA key generation where 512-bit p and q are randomly chosen, the first two MSBs of n must equal 00 or 01 with the probability 0.38, 10 with the probability 0.48, and 11 with the probability 0.14. Hence, anyone can detect the existence of SETUP by observing the abnormal distribution of n even without reverse-engineering.

Besides the above problem, the running time of Young and Yung's systems does not match the normal RSA key-generation time. To overcome the drawback, Crépeau and Slakmon [8] proposed four simple backdoor schemes: RSA-HSD$_\beta$, RSA-HSPE$_\beta$, RSA-HSE$_\beta$, and RSA-HP$_\beta$. For each scheme, a specific RSA attack [3][4][18][5][14] is employed and all the information required to mount the attack is permutated using the manufacturer's secret key and then embedded in the public key e or n. Since RSA-HSD$_\beta$, RSA-HSE$_\beta$, and RSA-HP$_\beta$ do not involve complicated steps, the running times are roughly equivalent to the normal RSA key-generation time. However, there are still some drawbacks. First, all the backdoors are not Strong-SETUPs, and anyone reverse-engineering the device can easily distinguish whether a casual key set is the output of the backdoor scheme or not. Second, only a limited range of e can be generated in the first three schemes. The limitation of e is impractical because $e = 2^{16} + 1$ is usually chosen in the RSA cryptosystem. Third, there is no proof that the information embedded in e (or n) is undetectable.

Based on the above arguments, we propose three new RSA backdoor cryptosystems. All of our backdoors are embedded in n, but not in e. The first two, RSA-SBLT and RSA-SBES, are similar to Crépeau and Slakmon's backdoors[8] since both of them involve no complicated steps such that their running time is close to the normal RSA key-generation time. However, we provide new different ways to construct the backdoors. RSA-SBLT is based on the lattice attack and RSA-SBES is based on the exhaustive search attack.

As for our third cryptosystem, RSA-BDH, we employ Diffie-Hellman(DH) assumption to construct a Strong-SETUP mechanism. Our backdoor is embedded in such a way that there exists a specific interrelationship between p and q. We can prove that this interrelationship is undetectable based on DH assumption. Besides, as long as the modulus r used in RSA-BDH is large enough, we can assure that p and q are uniformly distributed. Thus, different from Young and Yung's backdoors, the abnormal distribution of MSBs of n does not occur in RSA-BDH.

The remainder of the paper is organized as follows. In Section 2, we introduce the notation used in the paper. In Section 3, we focus on the lattice-based attack against small CRT-exponent RSA and the exhaustive search attack on small k RSA which will be applied to constructing our

backdoors later. In Section 4 and 5, we present three new backdoor cryptosystems for the RSA key generation, including two simple backdoors and a Strong-SETUP mechanism. Moreover, the proof of how the third backdoor cryptosystem satisfies the property of Strong-SETUP is demonstrated in Section 6. Finally we conclude the paper and consider an open problem in Section 7.

2 Notation

In the following, we introduce the notation used in the paper.

n:	the RSA modulus satisfying $n = pq$, where p and q are 512-bit primes.
E:	the secret kept by the manufacturer in RSA-SBLT and RSA-SBES.
K:	the parameter used in RSA-SBES.
l_n, l_e, l_E, l_K:	the bit-length of n, e, E, and K respectively.
l_{CRT-d}:	the bit-length of CRT-exponent d_p (or d_q) (Note that $d_p \equiv d \bmod p-1$ and $d_q \equiv d \bmod q-1$.)
m:	the security parameter. (Considering the computational ability today, m is often set to 40 since the exhaustive search in 2^{40} is feasible.)
r:	the modulus used in RSA-BDH. (Note that $r = 2r_1 + 1$ is a safe prime.)

3 Previous Attacks on RSA

In the section, we introduce two attacks on RSA which will be used later for devising our first two backdoor mechanisms. One is lattice-based attack which is proposed by Coppersmith [6] and can be applied to breaking small CRT-exponent RSA [17]. The other is exhaustive search attack employed to break small k RSA. Note that "small k RSA" means all RSA variants whose k's are smaller than 2^{40}. Now, we first describe the main theorem employed in the lattice-based attack as follows.

Theorem 3.1 (Trivariate Linear Modular Equation [11]) Let $f(x, y, z) \in Z[x, y, z]$. For every $\varepsilon > 0$, 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 $f_i(x, y, z)$ for $i = 1,2,3$ such that each root (x_0, y_0, z_0) of $f(x, y, z) \bmod M$ is also a root of each $f_i(x, y, z) \bmod M$, and if $|x_0| < X$, $|y_0| < Y$, $|z_0| < Z$ and $XYZ < M^{1-\varepsilon}$ for some bounds X, Y, and Z, then (x_0, y_0, z_0) is also a root of each of the three polynomials over the integers. If these three polynomials are also algebraically independent, then we can compute (x_0, y_0, z_0).

We omit the proof of Theorem 3.1 and refer the reader to [11]. In CRT-key equations, since $(d_p \equiv d \bmod(p-1))$ and $(d_q \equiv d \bmod(q-1))$, there exist two integers k_p and k_q such that $ed_p - 1 + k_p = k_p p$ and $ed_q - 1 + k_q = k_q q$. Multiplying together the two equations yields (after some rearrangement)

$$e^2 d_p d_q + e\,(d_p\,(k_q - 1) + d_q(k_p - 1)) - k_p k_q\,(N - 1) - k_p - k_q + 1 = 0 \tag{1}$$

Applying Theorem 3.1 to the equation (1), we deduce some insecure relationships between the RSA key lengths. We summarize them in Case 1. For more details, see [17].

Case 1: (for small CRT-exponent RSA)

If the lengths of keys satisfy the following inequalities, small CRT-exponent RSA could be broken with the lattice-based attack. Note that m is set to 40 such that the exhaustive search is feasible.

$$5l_{CRT-d} + 2l_e < 2l_n + m$$
$$3l_{CRT-d} + 2l_e < \tfrac{3}{2}l_n + m_1 \text{ and } l_e > \tfrac{l_n}{4} - m_2, \text{ where } m_1 + m_2 = m$$
$$6l_{CRT-d} + 3l_e < \tfrac{5}{2}l_n + m$$

The first inequality $5l_{CRT-d} + 2l_e < 2l_n + m$ can be derived by applying Theorem 3.1 to the equation (1), and setting the modulus M to e^2. Similarly, the second and the third inequalities can be derived by setting the modulus M to e and n respectively.

In addition to the attack on small CRT-exponent RSA, we introduce another attack on small k RSA which is referred to in [16] as follows.

Case 2: (for small k RSA)
Considering the RSA equation $ed = k(n + 1 - (p + q)) + 1$ modulo e, we get the equation:

$$(p + q) \equiv n + 1 + k^{-1} \bmod e \tag{2}$$

As long as the two subsequent conditions are both satisfied, we can break the RSA cryptosystem.

1. k is so small enough that an exhaustive search can be mounted to try all its possible values.
2. The bit length of $(p + q)$ is smaller than or equivalent to the bit length of e.

According to the above two conditions, since $p + q < e$ or $p + q \approx e$, the value $p + q$ can be computed exactly either by $(n + 1 + k^{-1} \bmod e)$ or by $(n + 1 + k^{-1} \bmod e) + e$. Moreover, since k is small enough, we can try all possible values of k to compute all possible values of $(p + q)$. By checking whether $\alpha^{n+1-(p+q)} \equiv 1 \bmod n$ for a random α, we can find the correct value of $(p + q)$ and break the system.

Considering the computational ability today, we can exhaustively search a number whose bit-length is less than or equivalent to 40. Thus we suggest that the parameter k used in our backdoor mechanism should be smaller than or equivalent to 2^{40}.

Note that in general, the bit-length of d is as long as that of k. However, there still exist some RSA variants with "small k" but "larger d", such as the RSA variant in [16]. What we use to devise our backdoor RSA-SBES is such a RSA variant.

4 The Proposed Simple RSA Backdoors

Based on the attack techniques introduced in the previous section, we devise two simple RSA backdoor cryptosystems. One is RSA-SBLT(Simple Backdoor based on Lattice Technique), and the other is RSA-SBES(Simple Backdoor based on Exhaustive Search). Both of them embed the backdoor in the RSA modulus n so the choice of the public key e is not limited. Besides, they have roughly the same running time as the normal RSA. In the following, we introduce them respectively.

4.1 Simple Backdoor Based on Lattice Technique (RSA-SBLT)

According to the lattice-based attacks on small CRT-exponent RSA shown in Case 1 of Section 3, there are three insecure relationships among the key lengths. Letting $l_e = \frac{l_n}{2}$, we rewrite the relationships as follows: (1) $5l_{CRT-d} < l_n + m$; (2) $6l_{CRT-d} < l_n + 2m_1$ and $\frac{l_n}{4} > -m_2$,where $m = m_1 + m_2$; (3) $6l_{CRT-d} < l_n + m$. Obviously, the third relationship is contained within the second relationship. Therefore, we can reduce the three relationships to

$$l_{CRT-d} < \tfrac{(l_n+m)}{5}. \tag{3}$$

$$l_{CRT-d} < \tfrac{l_n}{6} + \tfrac{m}{3}. \tag{4}$$

Based on the above two insecure relationships, we devise a new backdoor mechanism, called RSA-SBLT. We first introduce the parameters which the manufacturer needs to set beforehand. Then,

the algorithm, and how to use the backdoor to derive private keys are given subsequently.

Preprocessing: Parameters Set by Manufacturers

a. Set $l_E = \frac{l_n}{2}$ and choose a random prime E of l_E bits. (In order to make p and q generated by the system have the same bit-length $\frac{l_n}{2}$, l_E must be set to $\frac{l_n}{2}$.)

b. Set $m = 40$. (m should be set such that the exhaustive search in 2^m is feasible.)

c. Set l_{CRT-d} and l_n such that either $l_{CRT-d} < \frac{(l_n+m)}{5}$ or $l_{CRT-d} < \frac{l_n}{6} + \frac{m}{3}$ is satisfied. (Note that to facilitate the manufacturer's attack, l_{CRT-d} and l_n should be set more conservatively.)

Algorithm of RSA-SBLT:

Step 1. Randomly choose a number K_p of l_{CRT-d} bits such that $\gcd(E, K_p) = 1$.

Step 2. Compute d_p and p' such that $Ed_p - K_p p' = 1$, where $2^{l_{CRT-d}-1} < d_p < 2^{l_{CRT-d}}$ and $2^{\frac{l_n}{2}-1} < p' < 2^{\frac{l_n}{2}}$.

Step 3. If $p' + 1$ is not a prime, go to step 1. Otherwise, set $p = p' + 1$.

Step 4. Randomly choose a number K_q of l_{CRT-d} bits such that $\gcd(E, K_q) = 1$.

Step 5. Compute d_q and q' such that $Ed_q - K_q q' = 1$, where $2^{l_{CRT-d}-1} < d_q < 2^{l_{CRT-d}}$ and $2^{\frac{l_n}{2}-1} < q' < 2^{\frac{l_n}{2}}$.

Step 6. If $q' + 1$ is not a prime, go to step 4. Otherwise, set $q = q' + 1$.

Step 7. Perform normal RSA key generation except setting the modulus $n = pq$ generated before.

Usage of RSA-SBLT Backdoor: (Input: (n, E))

Since the insecure relationships among key lengths are assured in the process of the key generation, the lattice-based attack can be applied to solving $E^2 d_p d_q + E\left(d_p\left(K_q - 1\right) + d_q(K_p - 1)\right) - K_p K_q\left(n - 1\right) - K_p - K_q + 1 = 0$ while E and n are known. Thus, the manufacturer holding the secret E can solve the equation and further factor the modulus n.

Note that since the manufacturer's secret E is chosen to be a prime and larger than both K_p and K_q, the condition $\gcd(E, K_p) = 1$ in Step1 and the condition $\gcd(E, K_q) = 1$ in Step 4 are always true. That is, the repeated steps for satisfying the relative-prime relationships can be reduced and the running time of the scheme can be closer to that of the normal RSA key generation.

4.2 Simple Backdoor Based on Exhaustive Search (RSA-SBES)

Based on the exhaustive search attack on small k RSA as described in Case 2 of Section 3, we present a new backdoor cryptosystem, RSA-SBES. In the following, we first explain the parameters which need to be set by the manufacturer beforehand. Then, the details of our algorithm, and how to use the backdoor to derive private keys are given.

Preprocessing: Parameters Set by Manufacturers

a. Set $l_E = \frac{l_n}{2}$ and randomly choose a prime number E of l_E bits. (In order to make p and q generated by the system have the same bit-length $\frac{l_n}{2}$, l_E must be set to $\frac{l_n}{2}$.)

b. Set $l_K = 40$. (l_K should be set such that the exhaustive search in 2^{l_K} is feasible.)

Algorithm of RSA-SBES

Step 1. Randomly choose a prime p of $\frac{l_n}{2}$ bits.

Step 2. Randomly choose a number K of l_K bits such that $\gcd(E, K(p-1)) = 1$.

Step 3. Compute d and q' such that $Ed - K(p-1)q' = 1$, where $2^{l_K+\frac{l_n}{2}-1} < d < 2^{l_K+\frac{l_n}{2}}$ and $2^{\frac{l_n}{2}-1} < q' < 2^{\frac{l_n}{2}}$.

Step 4. If $q'+1$ is not a prime, go to Step 2. Otherwise, set $q = q'+1$.

Step 5. Perform normal RSA key generation except setting the modulus $n = pq$ produced before.

Usage of RSA-SBES Backdoor: (Input: (n, E))

Since the manufacturer holds the secret E and $K < 2^{l_K}$, he can try all possible values of K to compute either $(n + 1 + K^{-1} \bmod E)$ or $(n + 1 + K^{-1} \bmod E) + E$ to derive $(p + q)$. By checking whether $\alpha^{n+1-(p+q)} \equiv 1 \bmod n$ for a random α, the exact value of $(p+q)$ can be found, as mentioned in Case 2 of Section 3. Once $(p + q)$ is found, the manufacturer can derive the user's private key.

Note that the manufacturer's secret E is chosen to be a prime as in RSA-SBLT since once E is a prime, the repeated steps for satisfying the relative-prime relationships can be reduced and the running time of the scheme is closer to that of the normal RSA key generation.

5 RSA Backdoor based on Diffie-Hellman (RSA-BDH)

In Young and Yung's methods [19][20][21][22], since p and n are randomly chosen first and then q is decided by $q = \left\lfloor \frac{n}{p} \right\rfloor$, the two MSBs of n have abnormal distribution compared with that of the normal RSA where p and q are randomly chosen first. Thus, everyone can detect the existence of SETUP by checking the distribution of MSBs of n. In the section, we present a RSA backdoor which can solve the problem. We prove that our backdoor mechanism is a real Strong-SETUP based on the intractability of Diffie-Hellman problem. In the following, we first introduce some parameters which the manufacturer needs to set beforehand. Then, the details of the algorithm, and how to use the backdoor to derive private keys are presented. As for the related security analysis, we provide complete proofs in the next section.

Preprocessing: Parameters Set by Manufacturers

a. Select a safe prime r of $\frac{l_n}{2}$ bits, where $r = 2r_1 + 1$.

b. Let G be a cyclic group of Z_r^* of order $r-1$ and select a generator g of G.

c. Select an integer x in $[0, r-2]$ and keep it secret. (Note that since x does not appear in the mechanism of the system, it can not be obtained through reverse-engineering.)

d. Set $Y = g^x \bmod r$.

e. Select an odd number w in $[0, r-2]$.

Algorithm of RSA-BDH

Step 1. Randomly choose an integer c in $[0, r-2]$.

Step 2. Randomly choose two integers t_1 and t_2 in $[0, 1]$.

Step 3. If $(g^{c+wt_1}Y^c \bmod r)$ and $(g^{-wt_2}Y^{-c} \bmod r)$ are both primes, go to step 4. Otherwise, go to step 1.

Step 4. Set $p = (g^{c+wt_1}Y^c \bmod r)$, $q = (g^{-wt_2}Y^{-c} \bmod r)$, and $n = pq$.

Step 5. Perform normal RSA key generation except using the modulus n produced before.

Usage of RSA-BDH Backdoor: (Input: (n, g, x, r, w))

Step 1. Set $p' = n^{(1+x)} \bmod r$. If $(n \bmod p') = 0$, go to Step 5. Otherwise, go to the next step.

Step 2. Set $p' = p'g^w \bmod r$. If $(n \bmod p') = 0$, go to Step 5. Otherwise, go to the next step.

Step 3. Set $p' = (ng^w)^{1+x} \bmod r$. If $(n \bmod p') = 0$, go to Step 5. Otherwise, go to the next step.

Step 4. Set $p = ((ng^{-w})^{1+x}g^w \bmod r)$ and $q = \frac{n}{p}$, and the attack is finished.

Step 5. Set $p = p'$ and $q = \frac{n}{p'}$.

Regarding the usage of the backdoor, provided that $t_1 = 0$ and $t_2 = 0$, the manufacturer can easily recover p in Step 1 since $n = pq \equiv (g^cY^c)(Y^{-c}) \equiv g^c \bmod r$ and $\left(n^{1+x} \bmod r\right) = ((g^c)^{1+x} \bmod r) = (g^cY^c \bmod r) = p$. Analogously, when $t_1 = 1$ and $t_2 = 1$, p can be derived in Step 2 since $n = pq \equiv (g^{c+w}Y^c)(g^{-w}Y^{-c}) \equiv g^c \bmod r$ and $(n^{1+x}g^w \bmod r) = ((g^c)^{1+x}\, g^w \bmod r) = (g^{c+w}Y^c \mod r) = p$. When $t_1 = 0$ and $t_2 = 1$, p can be computed in Step 3 since $n = pq \equiv (g^cY^c)(g^{-w}Y^{-c}) \equiv g^{c-w} \bmod r$ and $((ng^w)^{1+x} \mod r) = ((g^{c-w}g^w)^{1+x} \mod r) = ((g^c)^{1+x} \bmod r) = (g^cY^c \bmod r) = p$. When $t_1 = 1$ and $t_2 = 0$, p can be obtained in Step 4 since $n = pq \equiv (g^{c+w}Y^c)(Y^{-c}) \equiv g^{c+w} \bmod r$ and $((ng^{-w})^{1+x}\, g^w \bmod r) = ((g^{c+w}g^{-w})^{1+x}\, g^w \mod r) = ((g^c)^{1+x}\, g^w \bmod r) = (g^{c+w}Y^c \bmod r) = p$.

In the next section, we will give the detailed security analysis of RSA-BDH further and show how RSA-BDH satisfies the property of Strong-SETUP.

6 Security Analysis of RSA-BDH

To ensure that anyone except the manufacturer can not distinguish the keys generated by RSA-BDH from the keys generated by the normal RSA system, there are two things which must be proved. First, the distributions of p and q generated by RSA-BDH are uniform as the distributions of p and q generated by the normal RSA system. Second, except the relation $n = pq$, there are no other interrelationships among p, q, and n which can be perceived by anyone excluding the manufacturer. Therefore, in the following two subsections, we show how RSA-BDH satisfies the two things with complete proofs.

6.1 Distributions of p and q

In RSA-BDH, since the generation of p and q involves the modulus r, the numbers between r and $2^{\frac{l_n}{2}} - 1$ can not be chosen as p and q. To prevent the abnormal condition from being detected, r must be selected as large as possible within $2^{\frac{l_n}{2}}$ such that the remaining portion between r and $2^{\frac{l_n}{2}} - 1$ are negligible compared with the whole range between $2^{\frac{l_n}{2}-1}$ and $2^{\frac{l_n}{2}} - 1$ from which the normal p and q can be chosen. That is, r must be selected such that $(2^{\frac{l_n}{2}} - r)/2^{\frac{l_n}{2}-1}$ is negligible. In such a way, we can regard the possibility of the abnormal condition being perceived as negligible.

Moreover, we must make sure that p and q are uniformly distributed. We prove it as follows.

Theorem 6.1 In RSA-BDH, $p(= g^{c+wt_1}Y^c \bmod r)$ and $q(= g^{-wt_2}Y^{-c} \bmod r)$ are uniformly distributed in $[1, r-1]$ as long as x does not equal r_1, $r_1 - 1$, $2r_1$, and $2r_1 - 1$, where $Y = g^x \bmod r$.

Proof. We only show that p is uniformly distributed in $[1,\, r-1]$. As for q, the proof is analogous and we omit it. Since g is a generator of G, to show that $p(= g^{c(1+x)+wt_1} \bmod r)$ is uniformly distributed in $[1,\, r-1]$ is equivalent to show that $(c(1+x)+wt_1 \bmod r-1)$ is uniformly distributed in $[0,\, r-2]$. We prove this with the aid of the function $f : A \to B$, where $A = [0, r-2] \times [0, 1]$ and $B = [0,\, r-2]$. The function f is defined as follows:

$$(c, t_1) \mapsto c(1+x) + wt_1 \bmod r - 1$$

Note that the mapping f stands for the relation between A and B. If f is onto and for every element in B, there are exactly two corresponding elements in A, we can claim that $c(1+x)+wt_1 \bmod r-1$ is uniformly distributed in B as long as (c, t_1) is uniformly distributed in A. To prove that f satisfies the above properties, let us consider the following two cases:

Case 1. *when x is odd:*

Given any odd number β_1 in B, we have the equation $c(1+x) + wt_1 = \beta_1 \bmod r - 1$. Since $c(1+x)$ and $r-1$ are even and β_1 is odd, wt_1 must be odd, that is, $t_1 = 1$. Solving for c, we get $c = (\beta_1 - w)(1+x)^{-1} \bmod r_1$. Similarly, given any even number β_2 in B, we have the equation $c(1+x)+wt_1 = \beta_2 \bmod r-1$. Since $c(1+x)$, $r-1$ and β_2 are all even, $t_1 w$ must be even, too. That is, $t_1 = 0$. Solving for c, we get $c = \beta_2(1+x)^{-1} \bmod r_1$.

Case2. *when x is even:*

Given any number β in B, we have the equation $c(1+x) + wt_1 = \beta \bmod r - 1$. No matter β is even or odd, we get $c = (\beta - wt_1)(1+x)^{-1} \bmod r - 1$, where t_1 is 0 or 1.

From case 1 and case 2, we know that while x is odd, given any β in B, there exist exactly two corresponding elements (c, t_1) and $(c + r_1, t_1)$ in A such that $f(c, t_1) = f(c + r_1, t_1) = \beta$; while x is even, given any β in B, there are exactly two corresponding elements $(c, 0)$ and $(c, 1)$ in A such that $f(c, 0) = f(c, 1) = \beta$. Thus, f is onto and maps exactly every two elements in A to one element in B. Since c and t are chosen randomly in our scheme, $c(1+x) + wt_1 \mod r - 1$ is uniformly distributed. Therefore, $p \equiv g^{c(1+x)+wt_1} \bmod r$ is also uniformly distributed in $[1, r-1]$. ■

In fact, it hardly happens that x equals r_1, $r_1 - 1$, $2r_1 - 1$, or $2r_1$. Moreover, as long as the manufacturer does not choose any of these values as his secret x, p and q are uniformly distributed in $[1,\, r-1]$.

6.2 Interrelationships among p, q, and n

Now, we want to show that except the relation $n = pq$, there is no other interrelationship among p, q, and n which can be detected by anyone excluding the manufacturer. To achieve the goal, we must prove the following six things: (a) p can not be deduced from q without the information n; (b) q can not be deduced from p without the information n; (c) n can not be deduced from q without the information p; (d) n can not be deduced from p without the information q; (e) p can not be deduced from n without the information q; (f) q can not be deduced from n without the information p. In fact, (a) is true if and only if (c) is true due to the relation $n = pq$. The similar condition occurs in (b)(d) and (e)(f). Thus, what remains to be proved is (a), (b), and (e). To prove them, we exploit the intractability of the DH problem [10]. The descriptions of the DH problem and the DH assumption are stated as follows:

DH (Diffie-Hellman) problem (in G): *Let $r = 2r_1 + 1$ be a safe prime, and G be a cyclic group of order $r-1$ with a generator g. Given group elements $g^a \bmod r$ and $g^b \bmod r$, find $g^{ab} \bmod r$.*

DH (Diffie-Hellman) assumption: *There exists no Turing Machine which can solve DH problem in probabilistic polynomial time.*

Besides DH assumption, we also need to exploit the intractability of RDH(Reversion of DH) problem. We state RDH problem and prove it is equivalent to DH problem in G as follows.

RDH (Reversion of Diffie-Hellman) problem (in G): *Let $r = 2r_1 + 1$ be a safe prime, and G be a cyclic group of order $r - 1$ with a generator g. RDH problem is as follows: given group elements $g^a \bmod r$ and $g^{ab} \bmod r$, find $g^b \bmod r$.*

Theorem 6.2 *The RDH problem is equivalent to the DH problem in G.*

Proof. Refer to Appendix A. ■

With the aid of the intractability of the DH and RDH problem, we can prove (a), (b), and (e) now. For convenience of proving, we do not consider the terms g^{wt_1} in p and g^{wt_2} in q, that is, regarding p as $(g^c Y^c \bmod r)$ and q as $(Y^{-c} \bmod r)$. Basically, g^{wt_1} and g^{wt_2} act as the constant 1 or g^w, and neglecting them does not affect the correctness of our proof. In the following, we prove (a) with Theorem 6.3 and Corollary 6.1, (b) with the Theorem 6.4 and Corollary 6.2, and (e) with Theorem 6.5 respectively.

Theorem 6.3 *Without the information n, p can be deduced from q if and only if the RDH problem in G can be solved.*

Proof. Suppose n is unknown and p can be deduced from q. Then there exists an oracle A such that $A(q, g, Y) = p$, i.e.

$$A(g^{-xc}, g, g^x) = (g^{c(1+x)} \bmod r)$$

Considering the RDH problem, given $(g^a \bmod r)$ and $(g^{ab} \bmod r)$, we can compute $(g^b \bmod r)$ using the oracle A since $(g^b \bmod r) = (g^{b(1+a)} g^{-ab} \bmod r) = (A(g^{-ab}, g, g^a) g^{-ab} \bmod r)$. That is, the RDH problem in G can be solved if p can be deduced from q without the information n.

Assuming that the RDH problem in G can be solved, there exists an oracle B such that $B(g^a, g^{ab}) = (g^b \bmod r)$. We can deduce p from q using the oracle B because $p = (g^c Y^c \bmod r) = (g^c g^{xc} \bmod r) = (B(g^x, g^{xc}) g^{xc} \bmod r) = (B(Y, q^{-1}) q^{-1} \bmod r)$. Therefore, if the RDH problem in G can be solved, p can be derived from q. ■

From Theorem 6.2 and 6.3, it follows that

Corollary 6.1 *Without the information n, p can be deduced from q in polynomial time if and only if the DH assumption in G is false.*

Theorem 6.4 *Without the information n, q can be deduced from p if and only if the RDH problem in G can be solved.*

Proof. If q can be deduced from p without the information n, there exists an oracle A such that $A(p, g, Y) = q$, i.e. $A(g^{c(1+x)}, g, g^x) = (g^{-xc} \bmod r)$. When considering the RDH problem, given $(g^a \bmod r)$ and $(g^{ab} \bmod r)$, we can compute $(g^b \bmod r)$ using the oracle A since $(g^b \bmod r) = (g^{ab} g^{-(a-1)b} \bmod r) = (g^{ab} A(g^{ba}, g, g^{a-1}) \bmod r) = (g^{ab} A(g^{ba}, g, g^a g^{-1}) \bmod r)$. Therefore, if q can be deduced from p without the information n, the RDH problem in G can be solved.

Assume that the RDH problem in G can be solved now. Let B be an oracle solving the RDH problem in G, i.e. $B(g^a, g^{ab}) = (g^b \bmod r)$. Using the oracle B, we can derive q from p since $q = (Y^{-c} \bmod r) = (g^{-xc} \bmod r) = (g^{c(-x-1)} g^c \bmod r) = ((g^{c(x+1)})^{-1} B(g^{x+1}, g^{c(x+1)}) \bmod r) = (p^{-1} B(Yg, p) \bmod r)$. Therefore, if the RDH problem in G can be solved, we can derive q from p. ■

From Theorem 6.2 and 6.4, it follows that

Corollary 6.2 *Without the information n, q can be deduced from p in polynomial time if and only if the DH assumption in G is false.*

Theorem 6.5 *Without the information q, p can be deduced from n if and only if the DH problem in G can be solved.*

Proof. Assume that the DH problem can be solved by an oracle A, i.e. $A(g^a, g^b) = (g^{ab} \bmod r)$. Using the oracle A, we can derive the value p from n since $p = (g^c Y^c \bmod r) = (g^c (g^x)^c \bmod r) = (g^c A(g^x, g^c) \bmod r) = (nA(Y, n) \bmod r)$. Thus, if the DH problem in G can be solved, p can be deduced form n without the information of q.

Suppose that p can be deduced from n without the information of q now. That is, there exists an oracle B such that $B(n, g, Y) = p$, i.e. $B(g^c, g, g^x) = (g^c g^{xc} \bmod r)$. Given $(g^a \bmod r)$ and $(g^b \bmod r)$, we can compute $(g^{ab} \bmod r)$ using the oracle B since $(g^{ab} \bmod r) = (g^b g^{ab} g^{-b} \bmod r) = (B(g^b, g, g^a) g^{-b} \bmod r)$. That is, we can solve the DH problem in G using the oracle B. Thus, if p can be deduced form n without the information of q, the DH problem in G can be solved. ■

According to Theorem 6.5, we can ensure that anyone except the manufacturer and the keys owner can not factor n in polynomial time based on the DH assumption. Moreover, from Corollary 6.1, 6.2, and Theorem 6.5, we can ensure that anyone excluding the manufacturer is unable to perceive the existence of the SETUP by observing the extra interrelationships among p, q, and n as long as the DH assumption is true.

From Corollary 6.1, 6.2, Theorem 6.1 and 6.5, it follows that

Theorem 6.6 *When r is large enough such that $(2^{\frac{ln}{2}} - r)/2^{\frac{ln}{2}-1}$ is negligible, RSA-BDH is a Strong-SETUP.*

Note that different from Young and Yung's methods, RSA-BDH does not generate a random n ahead but generates p and q which are uniformly distributed. Thus, RSA-BDH does not have the problem that the two MSBs of n have abnormal distribution.

7 Conclusion and Open Problem

In the paper, we propose three RSA backdoor cryptosystems: RSA-SBLT, RSA-SBES, and RSA-BDH. In RSA-SBLT and RSA-SBES, we provide two new ways to embed the backdoors. One is based on the lattice attack and the other is based on the exhaustive search attack. Since both of the systems own the advantages of Crépeau and Slakmon's schemes that the running time is roughly the same as the normal RSA key-generation time, anyone can hardly perceive the existence of the backdoor with time analysis.

In our third backdoor cryptosystem, RSA-BDH, we devise a new backdoor by exploiting the relationship between p and q. We solve the problem occurring in Young and Yung's scheme that the MSBs of n have abnormal distribution compared with that of the normal RSA. Hence, others can not detect the existence of backdoors by observing the distribution of MSBs of n. Moreover, we have proved that RSA-BDH is a real Strong-SETUP based on DH assumption. Note that in all of our proposed systems, the backdoor is embedded in n, but not in e. Thus, e can be chosen by users and this is more practical while small e is often used in the RSA cryptosystem for reducing encryption time. In fact, in addition to the purpose of stealing secrets, the backdoor mechanisms can also be applied to key escrow systems for the social security purpose.

To make a backdoor mechanism satisfy the property of Strong-SETUP, there must be some time-consuming steps involved in the construction. This is the dilemma in devising backdoor mechanisms.

However, does there exist a backdoor mechanism which not only satisfies the property of Strong-SETUP but also has the same key-generation time as the normal RSA? This remains an open problem for future research.

References

[1] R. J. Anderson, "A practical RSA trapdoor," *Electronic Letters*, vol. 29, no. 11, p. 995, 1993.

[2] A. Shamir, "Partial Key Escrow: A new approach to software key escrow," *In Key Escrow Conference*, 1995.

[3] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $n^{0.292}$," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp.1339-1349, 2000.

[4] D. Boneh, G. Durfee, and Y. Frankel, "An attack on RSA given a small fraction of the private key bits," *in Advances in Cryptology - ASIACRYPT'98*, LNCS 1514, pp.25-34, 1998.

[5] D. Coppersmith, "Finding a small root of a bivariate integer equation: factoring with high bits known," *in Advances in Cryptology - EUROCRYPT'96*, LNCS 1070, pp.178-189, 1996.

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

[7] Z. Chai, Z. Cao, and R. Lu, "ID-based threshold decryption without random oracles and its application in key escrow," *Proceedings of the 3rd International Conference on Information Security*, November 14-16, 2004.

[8] C. Crépeau and A. Slakmon, "Simple backdoors for RSA key generation," *Topics in Cryptology-CT-RSA 2003*, LNCS 2612, pp.403-416, 2003.

[9] D.E. Denning, "The US key escrow encryption technology," *Computer Communications*, vol. 17, no.7, pp.453-457, 1994.

[10] W. Diffie and M. Hellman, "New directions in cryptography," *In IEEE Trans. on Information Theory*, 22(6), pp. 644-654, 1976.

[11] M. J. Hinek, "Lattice attacks in cryptography: a partial overview," *Technical Report CACR 2004-08*, University of Waterloo, 2004.

[12] B. S. Kaliski, "Anderson's RSA trapdoor can be broken," *Electronics Letters*, vol. 29, no.15, p. 1387, 1993.

[13] J. Nechvatal, "A public-key-based key escrow system," *Journal of Systems and Software*, vol. 35, no. 1, pp. 73-83, 1996.

[14] R. L. Rivest and A. Shamir, "Efficient factoring based on partial information," *in Advances in Cryptology – EUROCRYPT'85*, LNCS 219, pp. 31-34, 1985.

[15] R. Rivest, A. Shamir, and L. Aldeman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM* , vol. 21, no.2, pp.120-126, 1978.

[16] H. M. Sun and C. T. Yang, "RSA with balanced short exponents and its application to entity Authentication," *Public Key Cryptography 2005*, pp.199-215, 2005.

[17] H. M. Sun, M. J. Hinek, and M. E. Wu, "On the design of rebalanced RSA-CRT," *Technical Report CACR 2005-35*. The paper is available from http://www.cacr.math.uwaterloo.ca/techreports/2005/cacr2005-35.pdf.

[18] M. Wiener, "Cryptanalysis of short RSA secret exponents," *IEEE Transactions on Information Theory*, vol. 36, no. 3, pp. 553-558, 1990.

[19] A. Young and M. Yung, "The dark side of "black-box" cryptography, or: should we trust Capstone?," *in Advances in Cryptology - CRYPTO'96*, LNCS 1109, pp.89-103, 1996.

[20] A. Young and M. Yung, "Kleptography: Using cryptography against cryptography," *Advances in Cryptology - EUROCRYPT'97*, LNCS 1233, pp.62-74, 1997.

[21] A. Young and M. Yung, "A space efficient backdoor in RSA and its applications," *Selected Areas in Cryptography 2005*, LNCS 3897, pp. 128-143, 2006.

[22] A. Young and M. Yung, "Malicious cryptography: kleptographic aspects," *CT-RSA 2005*, LNCS 3376, pp. 7-18, 2005.

Appendix A: The Proof of Theorem 6.2

Proof. Assuming that the RDH problem in G can be solved, there exists an oracle A such that $A(g, r, g^a, g^{ab}) = (g^b \bmod r)$. Given $(g^x \bmod r)$ and $(g^y \bmod r)$, we can use the oracle A to compute $(g^{xy} \bmod r)$ with the following method:

1. Check if $(g^x \bmod r)$ or $(g^y \bmod r)$ equals $(g^{r_1} \bmod r)$ or $(g^{2r_1} \bmod r)$. If any condition is satisfied, we can easily compute $(g^{xy} \bmod r)$ since either x or y is known.

2. Otherwise, let $x = 2^u m_1$ and $y = 2^v m_2$ where m_1 and m_2 are both odd. Without loss of generality, we assume $u \geq v$. The value $(g^{xy^{-1}} \bmod r)$ exists since when $v = 0$, $(g^{xy^{-1}} \bmod r) = (g^{2^u m_1 m_2^{-1}} \bmod r)$ where $\gcd(m_2, r-1) = 1$ and when $v > 0$, $(g^{xy^{-1}} \bmod r) = (g^{(2^{u-1}m_1(2^{v-1}m_2)^{-1} \bmod r_1)} \bmod r)$ where $\gcd(2^{v-1}m_2, r_1) = 1$. Now that $(g^{xy^{-1}} \bmod r)$ exists, it can be computed by using $A(g, r, g^y, g^x)$. Besides, we can use the oracle A to derive $(g^{x^2} \bmod r)$ since $A(g^x, r, g, g^x) = ((g^x)^x \bmod r) = (g^{x^2} \bmod r)$. Once $(g^{xy^{-1}} \bmod r)$ and $(g^{x^2} \bmod r)$ are known, $(g^{xy} \bmod r)$ can be easily derived because $A(g, r, g^{xy^{-1}}, g^{x^2}) = (g^{xy} \bmod r)$.

Therefore, if RDH problem in G can be solved, DH problem in G can be solved.

Now, assuming that DH problem in G can be solved, which means there exists an oracle B such that $B(g, r, g^x, g^y) = (g^{xy} \bmod r)$. Given $(g^a \bmod r)$ and $(g^{ab} \bmod r)$, we can use the oracle B to derive $(g^b \bmod r)$ since $B(g^a, r, g, g^{ab}) = (g^b \bmod r)$. Thus, if DH problem in G can be solved, RDH problem in G can be solved. ■