Security Analysis of CRT-Based Cryptosystems

Katsuyuki Okeya[1] and Tsuyoshi Takagi[2]

[1] Hitachi, Ltd., Systems Development Laboratory,
292, Yoshida-cho, Totsuka-ku, Yokohama, 244-0817, Japan
ka-okeya@sdl.hitachi.co.jp
[2] Technische Universität Darmstadt, Fachbereich Informatik,
Hochschulstr. 10, D-64283 Darmstadt, Germany
takagi@informatik.tu-darmstadt.de

Abstract. We investigate the security of several cryptosystems based on the Chinese remainder theorem (CRT) against side channel attack (SCA). Novak first proposed a simple power analysis against the CRT part using the difference of message modulo p and modulo q. In this paper we apply Novak's attack to the other CRT-based cryptosystems, namely Multi-Prime RSA, Multi-Exponent RSA, Rabin cryptosystem, and HIME(R) cryptosystem. Novak-type attack is strictly depending how to implement the CRT. We examine the operations related to CRT of these cryptosystems, and show that an extended Novak-type attack is effective on them. Moreover, we present a novel attack called zero-multiplication attack. The attacker tries to guess the secret prime by producing ciphertexts that cause a multiplication with zero during the decryption, which is easily able to be detected by power analysis. We examine the zero-multiplication attack on the above cryptosystems. Finally, we propose countermeasures against these attacks. The proposed countermeasures are based on the ciphertext blinding, but they require no inversion operation. The overhead of the proposed scheme is only about 1% to 5% of the whole decryption.

Keywords: RSA, Multi-Prime RSA, Factoring, Chinese Remainder Theorem, Side Channel Attacks, PKCS #1

1 Introduction

RSA cryptosystem is the most famous public-key cryptosystem in practical use, and it is implemented in plenty of security applications. Especially, security solutions with smart cards have been focused because of its flexibility and high security. However, recent research results point out weakness of RSA implementations on memory constraint devices against side channel attack (SCA) [Koc96, KJJ99,JLQ99,BDL01], etc. Several experimentation ensure practical feasibility of SCA [Nov02,BB03,ABF$^+$02]. These attacks are particularly effective on the implementation using the Chinese remainder theorem (CRT), which accelerates the decryption speed [PKCS]. The attack on RSA-CRT can factor the public modulus, and RSA cryptosystem is completely broken. We have to carefully deal with these attacks.

M. Jakobsson, M. Yung, J. Zhou (Eds.): ACNS 2004, LNCS 3089, pp. 383–397, 2004.

The decryption algorithm of RSA-CRT consists of two parts: (C1) to decrypt m_p, m_q (message modulo p, q) from ciphertext c, (C2) to recover the proper message m from m_p, m_q using CRT, where p, q are secret primes. The most side channel attacks deal with the first part, e.g., a timing attack using the difference of timing between $c < p$ and $c > p$ [Koc96,BB03], the power analysis on the modular multiplication $c^{d_p} \bmod p$ with secret exponent d_p [MDS99,BLW02], the timing attack using the final subtraction of Montgomery multiplication (See, for example, [Sch00]). However, Novak first proposed the attack on the second part [Nov02]. CRT is usually implemented the Garner algorithm [PKCS], and it causes operation $m_q - m_p$. He showed that a characteristic function δ of $m_q - m_p < 0$ can be detected by power analysis, and the modulus can be factored using δ. A standard countermeasure against these attacks is to randomize the ciphertext using the blind signature technique [Koc96,Kal96]. A message is randomized by multiplying r^e and the randomization is removed by multiplying r^{-1} after decryption. A drawback of this method is the computation of the inverse r^{-1}, which is not usually equipped on smartcards designed for RSA cryptosystem.

Incidentally, several cryptosystems based on CRT have been proposed in order to accelerate the decryption speed of RSA-CRT moreover, namely Multi-Prime RSA [PKCS] and Multi-Exponent RSA [Tak98]. Multi-Prime RSA utilizes a public modulus comprised several pair-wise distinct secret primes. Multi-Prime RSA is practically used, e.g., Compaq implemented it for a SSL sever [Com], Sun offers it in the specification of Java Cryptography Architecture [JCA]. Multi-Exponent RSA uses a modulus of form p^2q. The message modulo p^2 is recovered from modulo p using fast Hensel lifting, and the total decryption time of Multi-Exponent RSA is faster than that of Multi-Prime RSA for small exponent e. Other CRT-based cryptosystems are Rabin cryptosystem and HIME(R) cryptosystem [NSS01]. Their advantage over RSA is that they can be proven as secure as factoring problem in the sense of one-wayness or semantic security against chosen ciphertext attack.

1.1 Contribution of This Paper

In this paper, we investigate the security of the above CRT-based cryptosystems against SCA. First, the operations related to CRT of the cryptosystems are examined in the sense of Novak's attack. It is not obvious to construct a Novak-type attack on CRT-based cryptosystems, because the CRT computes with two numbers of different bit-length in these cryptosystems. Note that Novak's attack assume that these numbers are of nearly equal bit-length. In addition, the Novak-type attack is strictly depending how to implement the second CRT. We examine the operations related to CRT of these cryptosystems, and show that extended Novak-type attack is effective on them. Secondly, we present a novel attack called *zero-multiplication attack*. The attacker tries to guess the secret prime by producing ciphertexts that cause a multiplication with zero during the decryption. Note that Goubin [Gou03] proposed a side channel attack using a point with the zero value and an enhancement was proposed by Akishita-Takagi [AT03]. The crucial point of Goubin's attack is that a zero-valued register, which is known for the attacker, appears during the computation. In other words, Goubin's attack is a differential power analysis (DPA) using the *data* zero. On the other hand, the proposed zero-multiplication attack utilizes the *instruction* of the multiplication by zero for revealing the secret. That is, the zero-multiplication attack is a simple power analysis (SPA) using the instruction of zero. In fact, the zero-multiplication appears at CRT with small messages like $m < p \wedge m < q$, namely $m_q - m_p = 0$. We point out that Multi-Exponent RSA and HIME(R) cryptosystem involve additional zero-multiplications arisen from the Hensel lifting. Finally, we propose novel countermeasures against these attacks. The countermeasures are based on the ciphertext blinding technique, but they require no inversion operation. We can randomize ciphertexts and remove the randomization using only modular multiplication. This provides us a practical implication, because no library of computing an inversion is usually equipped on smartcards designed for RSA cryptosystem. The overhead of the proposed scheme is only about 1% to 5% of the whole decryption.

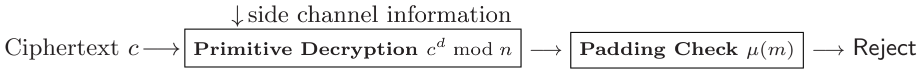
The paper is organized as follows: Section 2 reviews the RSA cryptosystems using RSA-CRT. Section 3 extends Novak's attack to Multi-prime RSA, and proposes zero-multiplication attack on it. Section 4 examines the proposed attacks on the CRT-based cryptosystems. Section 5 proposes countermeasures against these attacks.

2 RSA Cryptosystem with CRT

In this section we review the RSA cryptosystem using the Chinese remainder theorem (RSA-CRT).

Let $n = pq$ be the RSA modulus, where p, q are two secret primes that have the same bit length. Let e, d be the integers such that $ed = 1 \bmod \phi(n)$, where $\phi(n) = (p-1)(q-1)$. The public key and the secret key of the RSA cryptosystem are (e, n) and d, respectively. The message $m \in \mathbb{Z}_n$ is encrypted by computing $c = m^e \bmod n$. The integer c is called ciphertext. The person who knows the secret key d can decrypt the ciphertext by computing $m = c^d \bmod n$.

In order to make the RSA cryptosystem semantic secure against the chosen ciphertext attack, we usually deploy the OAEP padding [PKCS]. We convert the message m to PKCS format $\mu(m)$ before computing encryption. In the decryption, if $c^d \bmod n$ does not satisfy the PKCS format, c is rejected as invalid ciphertext. Note that the attacker is able to choose any ciphertext c as the input for primitive computation $c^d \bmod n$ (the manipulated ciphertexts are eventually rejected at the padding check with overwhelming probability). Although classical chosen ciphertext attacks, e.g. [Dav82] are not feasible, chosen ciphertext attacks using side channel information on the primitive decryption are feasible.


↓side channel information
Ciphertext c
Primitive Decryption $c^d \bmod n$
Padding Check $\mu(m)$
Reject

If we use the Chinese remainder theorem for the decryption of RSA, its speed can be accelerated with additional memory. Let $d_p = d \bmod p - 1$ and $d_q =$

$d \bmod q-1$. RSA-CRT deciphers $m_p = m \bmod p$ and $m_q = m \bmod q$ instead of computing $m = c^d \bmod n$. Indeed we can obtain them by $m_p = c^{d_p} \bmod p$ and $m_q = c^{d_q} \bmod q$. Then the proper message is recovered by applying the Chinese remainder theorem for m_p and m_q. We describe the standard algorithm of the decryption of the RSA-CRT [PKCS]:

```
RSA-CRT_Decryption
Input: ciphertext $c$, secret keys $p, q, d_p, d_q, pInv$
Output: message $m$
1. $m_p \leftarrow c^{d_p} \bmod p$,  $m_q \leftarrow c^{d_q} \bmod q$,
2. $h \leftarrow (m_q - m_p) * (pInv) \bmod q$,  $m_{pq} \leftarrow m_p + p * h$
3. return($m_{pq}$)
```

The Chinese remainder theorem at Step 2 is computed using Garner's algorithm [PKCS]. We pre-compute $pInv = p^{-1} \bmod q$, and Step 2 requires only two multiplications of $\mathbb{Z}_n$. Because the bit-size of p, q is half of n, the running time of computing $c^{d_p} \bmod p$ is about $2^3 = 8$ times faster. The total improvement of the running time is about 4 times.

2.1 Known Attacks

We review several known attacks against the RSA-CRT.

Timing Attack: Kocher proposed a timing attack of computing $c^{d_p} \bmod p$ [Koc96]. If $c < p$ holds, then we do not reduce c modulo p. There is a difference of timing between $c < p$ and $c > p$. The attacker can recover p by the binary search. This attack is called the *timing attack*. Recently Boneh et al. showed an experimental result of this timing attack against the server-client model — several implementation of SSL are vulnerable [BB03].

Fault Attack: If we can manipulate one bit of the register for $m_p = m \bmod p$ (say m', the related fake message), then the modulus can be factored by computing $gcd(m - m', n)$ due to $m' = m \bmod q$ and $m' \neq m \bmod p$ [JLQ99]. This attack is called the *fault attack*. This attack was extended to more sophisticated fault attack [BDL01], etc. Aumüller et al. showed an experimental result of this attack [ABF$^+$02]. They also proposed a countermeasure, which checks every process during the decryption, e.g. $m_p = m \bmod p$, $m^e = c \bmod p$, etc.

SPA/DPA: We can break the secret key by utilizing the side channel information related to the secret key, e.g., the simple power analysis (SPA), the differential power analysis (DPA) [KJJ99]. Messerges et al. showed the modular multiplication $c^d \bmod n$ is vulnerable against SPA/DPA [MDS99]. A DPA against the modular multiplication $c^{d_p} \bmod p$ was demonstrated by den Boer et al. [BLW02]. The ciphertext blinding method resists this type of attacks. The other countermeasure is the *exponent blinding method*, which randomizes the secret exponent by computing $d' = d + \phi(n)r$ for some integer r (or we can use a randomized representation of d, for example, MIST [Wal02]).

Timing Attack against Montgomery Multiplication: Schindler et al. pointed out the weakness of the implementation using the Montgomery multiplication

(See, for example, [Sch00]). The attacker tries to guess the secret key by observing the final subtraction of Montgomery multiplication. A countermeasure is to always perform the final subtraction, and then we choose the proper residue.

Novak Attack: Novak proposed an SPA against Step 2, namely the Chinese remainder theorem [Nov02]. He focused on the following implementation of $m_q - m_p \bmod q$; first compute $y = m_q - m_p$ and then $y = y + q$ if $y = m_q - m_p < 0$ holds. The experimental result shows the side channel information of $y = m_q - m_p < 0$ can be detected by SPA. He developed a binary search algorithm of finding secret key q with about $\log q$ calls. We should note that Novak's attack is effective for $m_q \approx m_p$ only, because y often takes different signs. A countermeasure against SPA is to always compute $y' = y + q$, and then we choose y' if and only if $m_q - m_p < 0$. Note that the exponent blinding method or MIST does not resist Novak attack.

Remark 1. The timing attack and Novak attack are effective on the chosen ciphertext setting. However, they are not feasible to the probabilistic signature, e.g., RSA-PSS [PKCS]. Even if the attacker chooses a message m, it is randomized by padding function ρ such that $\rho(m)$. The attacker cannot control the size of $\rho(m)$. Very recently, Fouque et al. proposed an extension of Novak attack on RSA with the randomly chosen messages, but this attack is restricted to the unbalanced modulus s.t. $p \not\approx q$ [FMP03].

3 Multi-prime RSA

In this section we investigate the security of Multi-Prime RSA against SCA. We assume the chosen ciphertext setting, that is, the attacker can freely choose ciphertexts for revealing the secret.

The public modulus of general Multi-Prime RSA consists of the product of several pair-wisely distinct secret primes [PKCS]. The current practically relevant Multi-Prime RSA modulus is a 1024-bit modulus $n = pqr$ with the same size secret primes p, q, r. In this paper we discuss this modulus, but the attack can be easily extended to other types.

The public-key of Multi-Prime RSA is (n, e), where $n = pqr$. The secret key is $(p, q, r, d_p, d_q, d_r, pInv, pqInv)$, where $d_p = e^{-1} \bmod p-1$, $d_q = e^{-1} \bmod q-1$, $d_r = e^{-1} \bmod r-1$, $pInv = p^{-1} \bmod q$, and $pqInv = (pq)^{-1} \bmod r$. A message $m \in \mathbb{Z}_n$ is encrypted by $c = m^e \bmod n$, which is equal to the RSA encryption. The ciphertext c is decrypted as follows:

```
Multi-Prime_RSA_Decryption
Input: ciphertext $c$, secret key $(p, q, r, d_p, d_q, d_r, pInv, pqInv)$
Output: message $m$
1. $m_p \leftarrow c^{d_p} \bmod p$,  $m_q \leftarrow c^{d_q} \bmod q$,  $m_r \leftarrow c^{d_r} \bmod r$
2. $h \leftarrow (m_q - m_p) * (pInv) \bmod q$,  $m_{pq} \leftarrow m_p + p * h$
3. $h \leftarrow (m_r - m_{pq}) * (pqInv) \bmod r$,  $m_{pqr} \leftarrow m_{pq} + (pq) * h$
4. return($m_{pqr}$)
```

Each modular multiplication modulo p, q, r is about 27 times faster than $c^d \bmod n$, because we choose these primes have the same size. Thus, the decryption algorithm of Multi-Prime RSA is about 9 times faster than that of RSA for the modulus n with same bit length.

At Step 2, we use the Chinese remainder theorem for p and q. Novak's attack can detect the approximation of prime q and we can factor n into q and pr. In this case, the number field sieve can factor pr much faster than pqr. We consider that the Multi-Prime RSA is broken, since it does not keep the expected security.

At Step 3, we compute CRT for pq and r using pre-computed value $pqInv = (pq)^{-1} \bmod r$. If we can develop a Novak-type algorithm for Step 3, the Multi-Prime RSA is also no longer secure. Note that a straight-forward extension of Novak's attack fails, because $m_{pq} > m_r$ holds in most cases, namely the value $m_{pq} - m_r$ is positive with high probability. In order to construct a Novak-type attack, we need to overcome this problem. In addition, Novak's attack against Step 3 strongly depends on how to implement "$h \leftarrow (m_r - m_{pq}) * (pqInv) \bmod r$". There are several ways to implement it. The possible ways are as follows:

(MP1) $y \leftarrow m_{pq} \bmod r$, $t \leftarrow m_r - y \bmod r$, $h \leftarrow t * pqInv \bmod r$,
(MP2) $y \leftarrow -m_{pq} \bmod r$, $t \leftarrow m_r + y \bmod r$, $h \leftarrow t * pqInv \bmod r$,
(MP3) $y \leftarrow m_r - m_{pq}$, $t \leftarrow y \bmod r$, $h \leftarrow t * pqInv \bmod r$.

The first way is a natural implementation, since we can reuse the modular subtraction module, because we should implement the modular subtraction $m_q - m_p \bmod q$ of Step 2. When we compute $-m_{pq} \bmod r$ in the second algorithm, it is usually computed $y = m_{pq} \bmod r$ and then $r - y$. This procedure avoids treating a signed integer. The third way is a straight-forward implementation.

3.1 Novak-Type Attack on Multi-prime RSA

In the following we investigate Novak's attack against the first implementation (MP1). For m in $\mathbb{Z}_n$, we define the following characteristic function:

$$\delta(m) = \begin{cases} 1 & \text{if } (m \bmod r) - ((m \bmod pq) \bmod r) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Because the integer m_{pq} is reduced modulo pq before computing modulo r, it differently behaves from Novak's attack. Indeed, we have the following proposition. In ascending order of m, the sign $\delta(m)$ has the pattern

$$1, .., 1, 0, .., 0, 1, .., 1, 0, .., 0, 1, ..,$$

and it is changed 1 to 0 if and only if $r|m$ holds. In other words, the attacker can factor n into r and pq if he/she detects such m.

Proposition 1. *Let $N = P_1 * P_2$, where* $\gcd(P_1, P_2) = 1$ *and $P_1 > P_2$. For M in $\mathbb{Z}_N$, define $\delta[P_1, P_2](M) = 1$ if $M \bmod P_2 - ((M \bmod P_1) \bmod P_2) \geq 0$, otherwise $\delta[P_1, P_2](M) = 0$. Then we have following properties:*

(1) $\delta[P_1, P_2](M) = 1$ *holds for all* $M < P_1$,
(2) For all $P_1 \leq M < N$, *we have*

$$\delta[P_1, P_2](kP_2) = ... = \delta[P_1, P_2](kP_2 + l) = 0,$$
$$\delta[P_1, P_2](kP_2 + l + 1) = ... = \delta[P_1, P_2]((k + 1)P_2 - 1) = 1,$$

where k, l *are integers with* $0 < k < P_1$ *and* $0 < l < P_2 - 2$.

Proof. Let $N = LP \cup UP$, where $LP = \{0, 1, ..., P_1 - 1\}$ and $UP = \{P_1, P_1 + 1, ..., N - 1\}$. For successive $M = 0, 1, 2, ..., N - 1$, we evaluate the values (M mod P_1) mod P_2 and M mod P_2 in the following. If $M \in LP$, then (M mod P_1) mod $P_2 = M$ mod P_2 holds due to M mod $P_1 = M$. Thus $\delta(M) = 1$ for all $M \in LP$. Next we consider the case of $M \in UP$. Let $f(M) = (M \bmod P_2) - ((M \bmod P_1) \bmod P_2)$. Then $\delta(M) = 1$ iff $f(M) \geq 0$. Note that $M \bmod P_1 \neq M \bmod P_2$ for $M \in UP$ due to $gcd(P_1, P_2) = 1$. Then the value (M mod P_1) mod P_2 is never equal to (M mod P_2) for $M \in UP$ because of the Chinese remainder theorem. Therefore, we obtain $f(kP_2) < 0$, namely $\delta(kP_2) = 0$, and $f(kP_2 - 1) = P_2 - 1 - ((kP_2 - 1 \bmod P_1) \bmod P_2) > 0$, namely $\delta(kP_2 - 1) = 1$, where k is a positive integer. Next we consider $f(kP_2 + l)$ for $0 < l \leq P_2 - 2$. For $0 < l \leq P_2 - 2$, we have two cases:

(i)$M = kP_2 + l$ is not divisible by P_1, (ii)$M = kP_2 + l$ is divisible by P_1.

Let $g(M) = M$ mod P_1. In the first case, we have $g(M+1) = g(M)+1$ and $g(M)$ is monotonously increasing for $M = kP_2 + l, l = 1, ..., P_2 - 2$. If we reduce them modulo P_2, then all numbers are pair-wised different due to $P_1 > P_2$. Thus, two sets (M mod P_1) mod P_2 and M mod P_2 have the following pattern:

$$(M \bmod P_1) \bmod P_2 = \{s, s + 1, ..., P_2 - 1, 0, 1, ..., s - 1\},$$
$$M \bmod P_2 = \{0, 1, ..., P_2 - s - 1, P_2 - s, P_2 - s + 1, ..., P_2 - 1\},$$

where for some $0 < s < P_2$. (M mod P_1) mod P_2 is the set of s-left-sift of M mod P_2. Thus the corresponding δ sequence is $\delta[P_1, P_2] = \underbrace{0, .., 0}_{P_2 - s}, \underbrace{1, .., 1}_{s}$. We have obtained the desired sequence. Next we discuss the second case. There is only one integer which is divisible by P_1 in any interval with length P_2. Let t be the number divisible by P_1 in interval $[kP_2, ..., kP_2 + P_2 - 1]$. The integers $M < t$ have the same pattern above. For $M \geq t$ we always have $f(M) > 0$, namely $\delta[P_1, P_2](M) = 1$. Thus we have the following pattern:

$$(M \bmod P_1) \bmod P_2 = \{P_1 - v, P_1 - v + 1, .., P_1 - 1, 0, 1, ..., u - 1\},$$
$$M \bmod P_2 = \{0, 1, ..., v - 1, P_2 - u, P_2 - u + 1, ..., P_2 - 1\},$$

where $v = t \bmod P_2$ and $u = P_2 - v$. Thus the corresponding δ sequence is $\delta[P_1, P_2] = \underbrace{0, .., 0}_{v}, \underbrace{1, .., 1}_{u}$. Consequently we have proved the proposition. □

If we choose $P_1 = pq$ and $P_2 = r$ we can construct the Novak-type attack from this proposition. The condition $P_1 > P_2$ is satisfied, because the three primes are chosen with same bit length. We describe the Novak-type attack in the following. It is modified from the original Novak attack in order to reduce the number of the oracle calls.

```
Novak_Type_Attack
INPUT: modulus $n$, public exponent $e$, upper bound $B$ of $r$.
OUTPUT: secret prime $r$ such that $n = pqr$.
1. Randomly choose $x_0 \in \mathbb{Z}_n$
   1.1. Compute $\delta(x_0)$ of $c_0 \leftarrow x_0^e \bmod n$ using SPA
   1.2. Set $\delta(x_1) \leftarrow \delta(x_0)$
2. While $\delta(x_1) = \delta(x_0)$
   2.1. If $\delta(x_0) = 1$, randomly choose $x_1$ s.t. $x_1 > x_0$ and $x_1 - x_0 < B$
          else randomly choose $x_1$ s.t. $x_1 < x_0$ and $x_0 - x_1 < B$
   2.2. Compute $\delta(x_1)$ of $c_1 \leftarrow x_1^e \bmod n$ using SPA
3. $LB \leftarrow x_0, UB \leftarrow x_1$.
   3.1. If $\delta(x_0) = 0$ then $LB \leftarrow x_1, UB \leftarrow x_0$.
4. While $LB \neq UB$ do:
   4.1. $m \leftarrow \lceil (LB + UB)/2 \rceil$.
   4.2. Compute $\delta(m)$ of $c \leftarrow m^e \bmod n$ using SPA.
   4.3. If $\delta(m) = 1$ then $LB \leftarrow m$; otherwise $UB \leftarrow m$.
5. Compute $r \leftarrow \gcd(n, m)$.
6. Return($r$).
```

At Step 1 we choose an initial integer x_0 randomly from $\mathbb{Z}_n$, and compute $\delta(x_0)$ of $c_0 \leftarrow x_0^e \bmod n$ using SPA. At Step 1 we try to find integer x_1, which satisfies $\delta(x_0) \neq \delta(x_1)$ and $|x_0 - x_1| < B$, where B is the upper bound of secret prime r. At step 3 we assign upper bound UB and lower bound LB of the target whose signs are exactly opposite. Step 4 is the main loop. We find the target based on the binary search of UB and LB. If $UB = LB(= m)$ holds, then we obtain the target. From the above lemma, the secret prime r yields by computing $gcd(m, n) = r$.

We estimate the average oracle calls. At Step 4 we requires at most $\lceil log_2 r \rceil$ oracle calls because $|UB - LB| < B$, where $B = 2^{\lceil log_2 r \rceil}$. We assume that $pq \bmod r$ is randomly distributed in modulo r. Then at Step 2 we can obtain x_1 with a few trials, because the probability of finding x_1 at Step 2 is about 1/2 on average due to randomness of x_1 and $s = pq \bmod r$. At Step 1 we use only one oracle call. Thus we need about $(\log_2 n)/3$ oracle calls on average.

Remark 2. If we modify the characteristic function δ, the attack described above is basically applicable to implementation (MP2) in the previous section, because the following two conditions are equivalent: $m_r - (m_{pq} \bmod r) \geq 0$ and $m_r + (r - (m_{pq} \bmod r)) \geq r$.

On the other hand, our attack is not applicable to implementation (MP3), because $m_r - m_{pq}$ is negative with high probability. However, we show a different attack on (MP3) in the next section.

3.2 Zero-Multiplication Attack

In this section we deal with the SPA using the multiplication with zero.

In the previous section we discussed that the Novak-type attacks are applicable to the decryption algorithm of Multi-Prime RSA if it deploys a subtraction related to the secret primes, e.g. $m_r - (m_{pq} \bmod r) \bmod r$. However, its practical feasibility causes a controversy, because we can avoid the operation as follows: at first we always compute $t = m_r - (m_{pq} \bmod r)$ and $t' = t + r$, and then "if $t < 0$ then $t = t'$". It is more difficult to detect the last operation using SPA.

However we can mount Novak's attack to the stronger SPA using the multiplication with zero. We call it *zero-multiplication attack* in this paper.

We have the following observation for the Multi-Prime RSA. If we choose $m < r$, then $h(m) = m_r - m_{pq} \bmod r = 0$ holds. Then we compute $0 * pqInv \bmod r$ at Step 3 of the decryption. The chosen ciphertext attack is allowed to generate the ciphertext c with $c = m^e \bmod n$ and $m < r$. Thus the binary search on r is possible using the SPA, and thus the secret prime r can be found. We show an algorithm to find the secret key r in the following. Define $\delta_{ZERO}(x) = 1$ if $h(x) = 0$ otherwise $\delta_{ZERO}(x) = 0$.

```
Zero_Multiplication_Attack
INPUT: modulus $n$, public exponent $e$, bit-length $L$ of $r$.
OUTPUT: secret prime $r$ such that $n = pqr$.
1. Set $x \leftarrow 0$
2. For $i = L - 1$ down to 0
  2.1. Set $y \leftarrow x + 2^i \bmod n$ and $c \leftarrow y^e \bmod n$
  2.2. Compute $\delta_{ZERO}(y)$ of $c$ using SPA
  2.3. If $\delta_{ZERO}(x) = 1$ holds, then set $x \leftarrow y$
3. Return($x$).
```

This attack is also applicable to the secret prime p at Step 2. If m satisfies both $m < p$ and $m < q$, then we always have $h = m_q - m_p = 0$.

Remark 3. Implementation (MP3) in the previous section is vulnerable against zero-multiplication attack. Because $y \leftarrow m_r - m_{pq}$ is always 0 if $m < r$ satisfies.

4 Application to Other CRT-Based Cryptosystems

There are several cryptosystem based on the Chinese remainder theorem, namely Rabin cryptosystem, Multi-Exponent RSA [Tak98], and HIME(R) cryptosystem [NSS01]. We discuss the effectiveness of the Novak-type attack and the zero-multiplication attack on them. We keep assuming the chosen ciphertext setting in this section.

4.1 Rabin Cryptosystem

We discuss the Novak attack against the Rabin cryptosystem. Let p, q be primes with $p \bmod 4 = q \bmod 4 = 3$. The public-key and secret-key of the Rabin cryptosystem are n and (p, q), respectively. A message $m \in \mathbb{Z}_n$ is encrypted by

$c = m^2 \bmod n$. This encryption function is a 4 : 1 mapping, and thus there are 4 different solutions for $c = x^2 \bmod n$. The 4 messages are decrypted as follows:

```
Rabin_Decryption
Input: ciphertext $c$, secret key $(p, q, d_p, d_q, pInv)$
Output: messages $m$
1. $m_p \leftarrow c^{(p+1)/4} \bmod p$, $m_q \leftarrow c^{(q+1)/4} \bmod q$
2. $h \leftarrow (m_q - m_p) * (pInv) \bmod q$, $m \leftarrow m_p + p * h$
3. Compute $\bar{m}$ from Step 1 to Step 2 for $(-m_p, m_q)$
4. Set $m_1 \leftarrow m$,   $m_2 \leftarrow \bar{m}$,   $m_3 \leftarrow n - m$,   $m_4 \leftarrow n - \bar{m}$
5. Find proper $m$ from $m_1, m_2, m_3, m_4$
6. Return$(m_1, m_2, m_3, m_4)$
```

At Step 1 the message modulo p, q are recovered. At Step 2 we compute the original m using the Chinese remainder theorem. The original Novak attack is applicable to Step 2. In order to recover other 3 different solutions for a given ciphertext c, we perform Step 1,2 for messages $(-m_p, m_q)$ and we obtain $\bar{m}$. Then all 4 solutions are $m_1 = m, m_2 = \bar{m}, m_3 = n - m, m_1 = n - \bar{m}$.

Two negative integer $-m_p$ is usually converted to its positive representative class, namely $p - m_p$. There is additionally one more possible oracle:

(R1) if $m_q - (p - m_p) < 0$, then $y = m_q - (p - m_p), h = y + q$.

Using this oracle we can construct a Novak-type attack. If the sign of the oracle changes 1 to 0, then the secret prime p or q appears. Indeed we have the following proposition, which is similarly to Proposition 1.

Proposition 2. *Let $n = pq$ be the RSA modulus. For m in $\mathbb{Z}_N$, define $\delta_{Rabin}(m) = 1$ if $m \bmod q - (p - (m \bmod p)) \geq 0$, otherwise $\delta_{Rabin}(m) = 0$. The sequence of $\delta_{Rabin}(m)$ has the consecutive pattern $\underbrace{0, .., 0}_{a_i}, \underbrace{1, .., 1}_{b_i}$ for successive $m = q, q+1, ..., n-1$, where $0 < a_i, b_i < \max(p, q)$ and $i = 0, 1, 2, ...$ The integer g that satisfies $\delta_{Rabin}(g-1) = 1$ and $\delta_{Rabin}(g) = 0$ is divisible by p or q.*

Proof. The proof is quite similar to that of Proposition 1. Thus we only describe the sketch of it. There are two cases: $p < q$ and $p > q$. At first we deal with the case of $p > q$. The sequences of $m \bmod q$ and $(p - (m \bmod p))$ are as follows:

$$m \bmod q = 0, 1, ..., q-2, q-1,$$
$$p - m \bmod p = s, s-1, ..., 2, 1, p, p-1, ..., t+1, t,$$

where $0 < s, t < p$. Therefore the sequence of δ associated to it has following fixed pattern: $\underbrace{0, .., 0}_{a_1}, \underbrace{1, .., 1}_{b_1}, \underbrace{0, .., 0}_{a_2}, \underbrace{1, .., 1}_{b_2}$ for some $0 < a_1, b_1 < p$ and $0 \leq a_2, b_2 < p$. The signs are changed at most twice modulo q. If $a_2 \neq 0$ holds, then the integer g such that $\delta(g-1) = 1$ and $\delta(g) = 0$ always satisfies $p|g$ or $q|g$. Next we deal with the case of $p < q$. The δ sequence is as follows: $\underbrace{0, .., 0}_{a_1}, \underbrace{1, .., 1}_{b_1}, ..., \underbrace{0, .., 0}_{a_i}, \underbrace{1, .., 1}_{b_i}$, where $0 < a_1, b_1, ..., a_{i-1}, b_{i-1} < p$ and $0 \leq a_i, b_i < p$ for some i. The signs are changed at most $\lceil q/p \rceil + 1$ modulo q. The integer g such that $\delta(g-1) = 1$ and $\delta(g) = 0$ always satisfies $p|g$ for the first $(i-1)$ changes of the sign and $q|g$ for the last one. Consequently, we proved the proposition. □

The zero-multiplication attack is also applicable to Step 2, because $m_q - m_p$ is zero if m satisfies both $m < p$ and $m < q$. However, it is not clear to find a zero multiplication for the other three Chinese remainder theorems. For example, without knowledge of p, q we are not able to find the message m that satisfies $p - m_p = m_q$.

4.2 Multi-exponent RSA

We discuss the variant of RSA using modulo p^2q proposed by Takagi [Tak98]. In this paper we call it Multi-Exponent RSA according to [BS02]. The message m is recovered from messages m_{p^2} modulo p^2 and m_q modulo q. The message m_{p^2} is lifted from the message m_p using the Hensel lifting, which requires only quadratic complexity $\mathcal{O}((\log p)^2)$. We present a modified version in the following. The public key is equal to that of the original RSA cryptosystem, namely (e, n) but $n = p^2q$ where p, q have the same bit length. The secret key is $(p, q, d_p, d_q, p^2Inv, eInv)$, where $d_p = e^{-1} \bmod (p-1), d_q = e^{-1} \bmod (q-1)$, $p^2Inv = (p^2)^{-1} \bmod q, eInv = e^{-1} \bmod p$. A message $m \in \mathbb{Z}_n$ is encrypted $c = m^e \bmod n$. The ciphertext c is decrypted as follows:

```
Multi-Exponent_RSA_Decryption
Input: ciphertext $c$, secret key $(p, q, d_p, d_q, p^2Inv, eInv)$
Output: message $m$
1. $k \leftarrow c^{d_p - 1} \bmod p$, $m_p \leftarrow ck \bmod p$, $m_q \leftarrow c^{d_q} \bmod q$
2. $g \leftarrow c - m_p^e \bmod p^2$, $b \leftarrow g * k * eInv \bmod p^2$, $m_{p^2} \leftarrow m_p + b$
3. $h \leftarrow (m_q - m_{p^2}) * (p^2Inv) \bmod q$, $m \leftarrow m_{p^2} + p^2 * h$
4. Return($m$)
```

At Step 1 we decrypt message modulo p, q, and the additional information $k = c^{d_p - 1} \bmod p$. At Step 2 we compute message modulo p^2 using the Hensel lifting. Note that m_{p^2} is uniquely represented as $m_{p^2} = m_p + b$, where b is divisible by p and $0 \leq b/p < p$. Thus we have relationship: $c \bmod p^2 = m_{p^2}^e \bmod p^2 = m_p^e + e m_p^{e-1} b \bmod p^2$, and thus we obtain $b = g((e m_p^{e-1})^{-1} \bmod p) \bmod p^2$, where $g = c - m_p^e \bmod p^2$. Because $(m_p^{e-1})^{-1} \bmod p = c^{d_p - 1} \bmod p$, we can correctly decrypt m_{p^2} at Step 2. At Step 3 we compute the Chinese remainder theorem for m_{p^2} and m_q. Thus the Novak-type attack is applicable to Step 3. Note that there is a multiplication with zero at Step 3 if the message is smaller than q. Therefore we can find the secret prime q using the zero-multiplication attack described in Section 3.2.

We discuss the zero-multiplication attack on Step 2. There is the following relation:

(ME1) if $m < p$, then $m_p = m, g = c - m_p^2 \bmod p^2 = 0$.
(ME2) if $m > p$, then $m_p \neq m, g = c - m_p^2 \bmod p^2 \neq 0$ with overwhelming probability over $m \in \mathbb{Z}_n$.

Thus there are two zero-multiplications at Step 3 if $m < p$ holds. The attacker detects whether g is zero or not. Define $\delta_{MERSA}(x) = 1$ if $x < p$ otherwise $\delta_{MERSA}(x) = 0$. Therefore, we can construct a binary search algorithm for p described Section 3.2 using this characteristic function δ_{MERSA}.

4.3 HIME(R) Cryptosystem

We discuss the security of HIME(R) cryptosystem developed [NSS01]. HIME(R) is a provably secure cryptosystem, which is IND-CCA2 under the factoring assumption of modulus p^2q. The decryption algorithm utilizes the Chinese remainder theorem and the Hensel-like lifting, and thus it is faster than RSA-CRT with the same modulus size. We describe the HIME(R) primitive in the following with a modification. Let p, q be primes with $p \bmod 4 = q \bmod 4 = 3$ with the same bit length. Let $pInv = p^{-1} \bmod q$ and $2Inv = 2^{-1} \bmod p$. The public-key and secret key of HIME(R) is n and $p, q, pInv, 2Inv$, respectively. A message is $m \in \mathbb{Z}_n$ is encrypted by $c = m^2 \bmod n$. This encryption function is same as the Rabin encryption $(4 : 1$ mapping). The message c is decrypted as follows:

```
HIME(R)_Primitive_Decryption
Input: ciphertext $c$, secret key $(p, q, pInv, 2Inv)$
Output: message $m$
0. Check $c \bmod p$ and $c \bmod q$ are quadratic residue
1. $k \leftarrow c^{(p-3)/4} \bmod p$, $m_p \leftarrow c * k \bmod p$, $m_q \leftarrow c^{(q+1)/4} \bmod q$
2. $h \leftarrow (m_q - m_p) * (pInv) \bmod q$, $m_{pq} \leftarrow m_p + p * h$
3. $g \leftarrow c - m_{pq}^2 \bmod n$, $b \leftarrow g * k * (2Inv) \bmod n$, $m \leftarrow m_{pq} + b$
4. Compute $\bar{m}$ from Step 2 to Step 3 for $(-m_p, m_q)$
5. Set $m_1 \leftarrow m$,   $m_2 \leftarrow \bar{m}$,   $m_3 \leftarrow n - m$,   $m_4 \leftarrow n - \bar{m}$
6. Find proper $m$ from $m_1, m_2, m_3, m_4$
7. Return($m$)
```

At Step 0 we check the ciphertext c is quadratic residue or not. At Step 1 we compute the message modulo p and q, and additionally $k = c^{(p-3)/4} \bmod p$. At Step 2 the message modulo pq are recovered using the Chinese remainder theorem. Here we can apply the original Novak attack. Note that an integer $m \in \mathbb{Z}_n$ is uniquely represented as $m = m_{pq} + b$, where $m_{pq} = m \bmod pq$, b is divisible by pq and $0 \leq b/pq < p$. At Step 3 we find the unique integer b for given m_{pq} and c. From $c \bmod n = m^2 \bmod n = m_{pq}^2 + 2m_{pq}b \bmod n$, we obtain $b = (c - m_{pq}^2)((2m_p)^{-1} \bmod p) \bmod n$. Here we have $k = m^{(p-3)/2} \bmod p = m^{(p-1)/2-1} \bmod p = \pm m_p^{-1} \bmod p$ due to $m^{(p-1)/2} = \pm 1 \bmod p$. We can correctly recover the message m at Step 3. At Step 4 and 5 we generate other 3 candidates of the proper message. The Novak-type attack described in Section 4.1 is applicable to the message $(-m_p, m_q)$ at Step 4.

There are several operations related to the secret keys p, q. We examine the zero-multiplication attack on Step 3. Recall $c = m^2 \bmod p^2q$, since c is the ciphertext of m. Then we have following relationship:

(H1) if $m < pq$, then $m_{pq} = m, c - m_{pq}^2 \bmod p^2q = 0$.
(H2) if $m > pq$, then $m_{pq} \neq m, c - m_{pq}^2 \bmod p^2q \neq 0$ with overwhelming probability over $m \in \mathbb{Z}_n$.

Thus in Step 3 two zero-multiplications appear if $E(m) = c - m_{pq}^2$ is zero. The attacker detects whether $E(m)$ is zero or not. Define $\delta_{HIME}(x) = 1$ if $E(x) = 0$ otherwise $\delta_{HIME}(x) = 0$. Therefore, we can construct a binary search algorithm for pq described Section 3.2 using this characteristic function δ_{HIME}.

5 New Countermeasures

In this section we discuss how to randomize a ciphertext of RSA cryptosystem.

A standard way is the ciphertext blinding method (see Section 2.1). A drawback of this scheme is the computation of the inverse $r^{-1} \bmod n$. A library that computes an inversion is not usually equipped on smartcards, so that the designer has to additionally develop it. While we can compute $r^{-1} \bmod n$ using the modular exponentiation $r^{\phi(n)-1} \bmod n$, it requires a large overhead.

We present a randomization method, which requires no modular inversion. The proposed algorithm is as follows:

```
Ciphertext_Blinding_Without_Inversion
Input: public keys $n, e$, ciphertext $c$, secret keys $p, q$
Output: message $m$ s.t. $c = m^e \bmod n$
1. Compute $s \leftarrow r^{e-1} \bmod n$ for random $r \in \mathbb{Z}_n^*$ and $t \leftarrow s * r * c \bmod n$
2. Compute $u \leftarrow t^{d-1} \bmod n$
3. Compute $v \leftarrow u * s \bmod n$ and $m \leftarrow v * c \bmod n$
4. Return($m$)
```

At Step 1 the ciphertext c is blinded by $r^e \bmod n$ such that $t = r^e c \bmod n$, but we store the value $s = r^{e-1} \bmod n$. At Step 2 we decrypt message t using exponent $d-1$ instead of d, namely $u = t^{d-1} \bmod n$. Note that $u * s \bmod n = (m^e r^e)^{d-1} r^{e-1} \bmod n = m^{1-e} \bmod n$. Thus at Step 3 we can recover message m by $u * s * c \bmod n$. The attacker tries to analyze the computation of Step 4, but ciphertext c is randomized as $r^e c \bmod n$. At Step 2 we can also compute $t^{d-1} \bmod n$ using the Chinese remainder theorem. In that case, we have to modify the secret key d_p and d_q to $d_p = d - 1 \bmod p - 1$ and $d_q = d - 1 \bmod q - 1$, respectively. We also note that public exponent e has to be known, which is not always the case in real-life applications.

This countermeasure is efficient for small encryption exponent. If we choose standard $e = 2^{16} + 1$, then it requires about only 20 multiplications of $\mathbb{Z}_n$. Therefore the overhead is about 5% of the whole decryption computation of RSA with CRT. We require 2 registers of $\mathbb{Z}_n$ for auxiliary paramters (s, u).

In the following we discuss other possible randomization schemes. If we store $(S, T) = (r^e \bmod n, r^{-1} \bmod n)$ in non-volatile memory for a random integer $r \in \mathbb{Z}_n$, then we can randomize a ciphertext c by computing $m = (S^t c)^d T^t \bmod n$, where t is a small random exponent. If we choose 16-bit or 32-bit t, then the overhead of this scheme is about 47 or 98 multiplications of $\mathbb{Z}_n$ (namely about 12% or 26% comparing with RSA-CRT), respectively. We require 2 registers of $\mathbb{Z}_n$ for auxiliary paramters $(S^t c, T^t)$. Consequently, our proposed scheme is more efficient than these schemes.

5.1 Application to Rabin Cryptosystem

The proposed countermeasure is also applicable to Rabin or HIME(R) cryptosystem. The encryption exponent of these schemes is 2, so that this countermeasure is particularly effective. In that case we have to choose quadratic residue r, otherwise the valid ciphertext is not decrypted. Indeed we can construct it as follows:

```
Rabin_Ciphertext_Blinding_Without_Inversion
Input: public key $n$, ciphertext $c$, secret keys $p, q$
Output: message $m$ s.t. $c = m^2 \bmod n$
1. Compute $s \leftarrow r^2 \bmod n$ for random $r \in \mathbb{Z}_n^*$
2. Randomize the ciphertext $t \leftarrow s^2 * c \bmod n$
3. Compute $u_p \leftarrow t^{(p-3)/4} \bmod p$ and $u_q \leftarrow t^{(q-3)/4} \bmod q$
4. Compute 4 different $w \bmod n$ corresponding to $u_p, u_q$ using CRT
5. Compute $m \leftarrow w * c * s \bmod n$
6. Return($m$)
```

At Step 1 we generate a random quadratic residue s. At Step 2 the ciphertext c is randomized by computing $t = s^2 c \bmod n$, so that the attacker cannot manipulate the randomized message $ms \bmod n$. At Step 3 we decrypt the inversion of the randomized message $(ms)^{-1} \bmod p$ and $(ms)^{-1} \bmod q$, respectively. Note that $t^{(p-3)/4} \bmod p = t^{(p+1)/4-1} \bmod p = (\pm ms)t^{-1} \bmod p = \pm(ms)^{-1} \bmod p$. At Step 4 we recover 4 different messages related to $w = (ms)^{-1} \bmod n$. At Step 5 we compute the proper message m by $m = w * c * s \bmod n$.

The overhead of the proposed method is only 5 multiplications of $\mathbb{Z}_n$. It is about 1% of the whole decryption computation.

References

[ABF$^+$02] C. Aumüller, P. Bier, W. Fischer, P. Hofreiter, and J.-P. Seifert, "Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures," CHES 2002, LNCS 2523, pp.260-275, 2003.

[AT03] T. Akishita and T. Takagi, "Zero-Value Point Attacks on Elliptic Curve Cryptosystem", ISC 2003, LNCS2851, pp. 218-233, 2003.

[BLW02] B. den Boer, K. Lemke, and G. Wicke, "A DPA Attack against the Modular Reduction within a CRT Implementation of RSA," CHES 2002, LNCS 2523, pp.228-243, 2003.

[BDL01] D. Boneh, R. DeMillo, R. Lipton, "On the Importance of Eliminating Errors in Cryptographic Computations." J. of Cryptology, 14(2), pp.101-119, 2001.

[BB03] D. Boneh and D. Brumley, "Remote Timing Attacks are Practical," 12th Usenix Security Symposium, pp.1-14, 2003.

[BS02] D. Boneh and H. Shacham, "Fast Variants of RSA," CRYPTOBYTES, Vol.5, No.1, pp.1-9, 2002.

[Com] MultiPrime^{TM}, Compaq AXL300 Accelerator. `http://www.compaq.com/products/servers/security/axl300/`

[Dav82] G. Davida, "Chosen Signature Cryptanalysis of the RSA (MIT) Public Key Cryptosystem," TR-CS-82-2, University of Wisconsin, 1982.

[FMP03] P.-A. Fouque, G. Martinet, G. Poupard, "Attacking Unbalanced RSA-CRT using SPA," CHES 2003, LNCS 2779, pp.254-268, 2003.

[Gou03] L. Goubin, "A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems", PKC 2003, LNCS 2567, pp. 199-211, 2003.

[JCA] Java Cryptography Architecture, `http://java.sun.com/products/jdk/1.2/docs/guide/security/CryptoSpec.html`

[JLQ99] M. Joye, A.K. Lenstra, and J.-J. Quisquater, "Chinese Remaindering Based Cryptosystems in the Presence of Faults," Journal of Cryptology 12(4), pp.241-245, 1999.

[Kal96] B. Kaliski, "Timing Attacks on Cryptosystems," RSA Laboratories Bulletin, No.2, 1996.

[Koc96] C. Kocher, "Timing attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems," CRYPTO '96, LNCS 1109, pp.104-113, 1996.

[KJJ99] C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," CRYPTO '99, LNCS 1666, pp.388-397, 1999.

[MDS99] T. Messerges, E. Dabbish, R. Sloan, "Power Analysis Attacks of Modular Exponentiation in Smartcards," CHES'99, LNCS 1717, pp.144-157, 1999.

[NSS01] M. Nishioka, H. Satoh, and K. Sakurai, "Design and Analysis of Fast Provably Secure Public-Key Cryptosystems Based on a Modular Squaring," ICISC 2001, LNCS 2288, pp.81-102, 2001.

[Nov02] R. Novak, "SPA-Based Adaptive Chosen-Ciphertext Attack on RSA Implementation," PKC 2002, LNCS 2274, pp.252-262, 2002.

[PKCS] Public-Key Cryptography Standards, PKCS # 1, Amendment 1: Multi-Prime RSA, RSA Laboratories.

[Sch00] W. Schindler, "A Timing Attack against RSA with the Chinese Remainder Theorem," CHES 2000, LNCS 1965, pp.109-124, 2000.

[Tak98] T. Takagi, "Fast RSA-type cryptosystem modulo $p^k q$," CRYPTO '98, LNCS 1462, pp.318-326, 1998.

[Wal02] C. Walter, "MIST: An Efficient, Randomized Exponentiation Algorithm for Resisting Power Analysis," CT-RSA 2002, LNCS 2271, pp.53-66, 2002.