On the Fault Attacks against RSA signatures with Lower-order Bits of Messages Unknown

Abstract

This paper discusses the fault attacks on RSA signatures with partially unknown messages proposed by Coron, Joux, Kizhvatov, Naccache and Paillier (CJKNP) in CHES 2009. Our new attacks handle the case that low-order bits of messages are unknown, and the total size of unknown part can be improved from $N^{0.207}$ to $N^{0.25}$ with a single faulty signature. Furthermore, this bound is extended to $N^{\frac{1}{2}-\frac{1}{2\ell}}$ by utilizing multiple faulty signatures which is better than previous results in such case, where ℓ is the number of faulty signatures. In fact, our new results can be seen as a nice complement to previous attacks.

Keywords: RSA Signatures, Fault Attacks, ISO/IEC 9796-2, LLL Algorithm.

1 Introduction

RSA signature [16] invented by Rivest, Shamir, Adleman is undoubtedly the most widely used signature scheme. To sign a message m, the signer first encodes the message m as $\mu(m)$ by encoding function $\mu(\cdot)$, then computes the signature $\rho = \mu(m)^d \mod N$. To verify the signature, the receiver checks that $\rho^e = \mu(m) \mod N$. A sophisticated way to speed up the signature generation is to explore the Chinese Remainder Theorem(CRT). This is done by computing:

$$\rho_p = \mu(m)^{d_p} \mod p \text{ and } \rho_q = \mu(m)^{d_q} \mod q,$$

where $d_p = d \pmod{p-1}$ and $d_q = d \pmod{q-1}$. We obtain the signature

$$\rho = \rho_p + p \cdot (p^{-1} \mod q) \cdot (\rho_q - \rho_p) \mod N$$

using CRT. Such an approach, called RSA-CRT, achieves the signing time that is approximately 4 times faster than signature using the standard RSA definition.

The first fault attack against RSA-CRT signature implementation was presented by Boneh, DeMillo and Lipton in [2]. Assuming that the attacker has the ability to induce a fault when ρ_q is computed while keeping the computation of ρ_p correct, one obtains $\rho_p = \mu(m)^d \mod p$ and $\rho_q \neq \mu(m)^d \mod q$, and then computes the signature ρ using CRT such that

$$\rho^e = \mu(m) \mod p \text{ and } \rho^e \neq \mu(m) \mod q.$$

Hence, if the padding is determined, the modulus N can be factored by computing

$$\gcd(\rho^e - \mu(m) \mod N, N) = p. \tag{1}$$

It is easy to extend Boneh et al.'s fault attack to any deterministic RSA encoding, e.g. the Full Domain Hash [1] encoding where $\rho = H(m)^d \mod N$ and $H : \{0,1\}^* \mapsto \mathbb{Z}_N$ is a hash function. Even when the random nonce which is used to signature generation is sent along with signature, the attack can also work well.

In [4] , Coron, Joux, Kizhvatov, Naccache and Paillier (CJKNP) showed that how to extend Boneh *et al.*'s attack to RSA signature with unknown message part(UMP). CJKNP's attack was illustrated with a randomized version of the ISO/IEC 9796-2 standard [10]. In ISO/IEC 9796-2 the encoded message has the form:

$$\mu(m) = 6A_{16}||m[1]||H(m)||BC_{16}$$

where $m = m[1]||m[2]$ is split into two parts. CJKNP pointed out that if the unknown part of $m[1]$ is not too large (e.g. less than 160 bits for a 2048-bits RSA modulus), then the attacker can factor the RSA modulus N using a single faulty signature as in Boneh *et al.*'s attack. They also showed how to extend the attack to multiple faulty signatures, however the complexity increases exponentially with the number of faulty signatures.

At RSA 2010, Coron, Naccache and Tibouchi exhibited a simpler multiple fault attack [6], whose complexity is polynomial in number of faulty signatures. Based on the orthogonal lattice technique similar to [14, 15], their attacks allow us to tackle larger UMPs and show that in EMV case, N can be factored in a fraction of a section using ten faulty signatures.

This paper improves the CJKNP's fault attacks under the case that the partially unknown part is located in the least significant bits of message $m[1]$. It seems as a special case of CJKNP's attacks, however, in such case, we are required to solve univariate modular equations instead of bivariate equations, and the bound of UMPs can be improved from $N^{0.207}$ to $N^{0.25}$. This means that for a 1024-bit modulus N, the total size of the unknown parts can be at most 256 bits. Applying to the ISO/IEC 9796-2 signatures where the bit-size of hash function k_h is 160, the unknown part r can thus be at most 96-bit long, while the attack in [4] can work only when the bit length of r is at most 52.

Moreover, we show that how to extend the bound of unknown message using the multiple faulty signature. Based on solving simultaneous Diophantine approximation problem, the multiple faulty attack can heuristically factor N if the unknowns part is at most $N^{\frac{1}{2}-\frac{1}{2\ell}}$, where ℓ is the number of faulty signatures, which is better than the result of [6].

The rest of this paper is organized as follows. In section 2, we give a simple discription of ISO/IEC 9796-2, and recall the CJKNP's attack. Section 3 shows a new fault attack against a probabilistic variant of ISO/IEC 9796-2 standard when some LSBs of $m[1]$ are unknown, and extend the bound of the unknown part of messages by multiple faulty attacks. Finally, we give some simulation results and compare new attacks with previous attacks in section 4.

2 CJKNP's Attack on ISO/IEC 9796-2

2.1 ISO/OEC 9796-2 Standard

ISO/IEC 9796-2 is an encoding standard allowing partial or total message recovery [10, 11]. The encoding function can be used with hash function $H(m)$ of digest size k_h. For the sake of simplicity we assume that k_h, the size of m and the size of N (denoted by k) are all multiples of 8. The ISO/IEC 9796-2 encoding of a message $m = m[1]||m[2]$ is

$$\mu(m) = 6A_{16}||m[1]||H(m)||BC_{16}$$

where $m[1]$ consists of the $k - k_h - 16$ most significant bits of m and $m[2]$ represents the remaining bits of m. Notes that the original version of the standard recommended $128 \leq k_h \leq 160$ for partial message recovery (see [10]). In [5], Coron *et al.* introduced an attack against $k_h = 128$ and $k_h = 160$ with 2^{56} and 2^{61} operations respectively; After this attack published, ISO/IEC 9796-2 was amended and the current official requirement (see [11]) is now $k_h \geq 160$. In recent work, [7] described a practical forgery attack (without faults) against ISO/IEC 9796-2.

2.2 CJKNP's Attack

CJKNP's attack considers a randomized version of ISO/IEC 9796-2 standard. More precisely, the authors in [4] analyze a message $m = m[1]||m[2]$ of the form

$$m[1] = \alpha||r||\alpha', \quad m[2] = \text{DATA}$$

where r is a message part unknown to the adversary, α, α' are strings known to the adversary and DATA is some known or unknown string. The size of r is denoted by k_r and the size of $m[1]$ is $k - k_h - 16$ as required in ISO/IEC 9796-2. The encoded message is then

$$\mu(m) = 6A_{16}||\alpha||r||\alpha'||H(\alpha||r||\alpha'||\text{DATA})||BC_{16} \tag{2}$$

The total number of unknown bits in $\mu(m)$ is $k_r + k_h$. Assuming that the attacker can obtain a faulty signature satisfying (1), from (1) and (2), one can get

$$\rho^e = s + r \cdot 2^{n_r} + H(m) \cdot 2^8 \mod p.$$

where s is a known value. This shows that $(r, H(m))$ must be a solution of the equation

$$a + bx + cy = 0 \mod p$$

where $a := s - \rho^e \mod N, b := 2^{n_r}$ and $c := 2^8$ are known. This problem can be solved using the result by Herrmann and May in [8] based on Coppersmith's method [3] for finding small roots of polynomial equations. Assuming that $r < N^\gamma$, $H(m) < N^\delta$, one can get:

$$\gamma + \delta \leq \frac{\sqrt{2} - 1}{2} \approx 0.207$$

for a balanced RSA modulus.

3 New Fault Attacks on Randomized Version of ISO/IEC 9796-2

In this section we assume that lower-order bits of message $m[1]$ are unknown. This means that

$$m[1] = \alpha || r, \quad m[2] = \text{DATA}$$

where α is a string known, r is a message part unknown to the adversary. DATA is some known or unknown string. The size of r is denoted as k_r and the size of $m[1]$ is $k - k_h - 6$ as required in ISO/IEC 9796-2. The encoded message is then

$$\mu(m) = 6A_{16}||\alpha||r||H(\alpha||r)||BC_{16} \tag{3}$$

The total number of unknown bits in $\mu(m)$ is $k_r + k_h$.

The above assumption simplifies the attacking scenario by only solving univariate modular equations. We show that the bound of UMPs of $m[1]$ can be improved from $N^{0.207}$ to $N^{0.25}$ using a single faulty signature in section 3.1. Section 3.2 discusses two faults modulo different factors. Multiple faults attack is improved in section 3.3.

3.1 Single Fault Attack

We suppose that after injecting a fault the opponent is in possession of a faulty signature ρ such that

$$\rho^e = \mu(m) \mod p, \quad \rho^e \neq \mu(m) \mod q \tag{4}$$

From (3) we can write

$$\mu(m) = s + r' \cdot 2^8$$

where $s = (6A_{16}||\alpha) \cdot 2^{k_r + k_h + 8} + BC_{16}$ is a known value and $r' = r \cdot 2^{k_h} + H(m)$ is unknown. From (4) we obtain

$$\rho^e = s + r' \cdot 2^8 \mod p.$$

This shows that $x_0 = r \cdot 2^{k_h} + H(m)$ must be a solution of the equation

$$a + bx = 0 \mod p. \tag{5}$$

where $a := t - \rho^e \mod N, b := 2^8$ are known. From previous analysis, we assume $b = 1$.

Therefore, if the root x_0 of the above equation can be found, one can recover the randomized encoded message $\mu(m)$ and then get the factor p of N by computing GCD of N and $x_0 + a$ from (5).

Now, our technical goal is to obtain the useful solution x_0 satisfying equation (5). To achieve this goal, we make use of the approach of solving the modular equation with a unknown modulus. The main idea is to reduce from solving modular polynomial equations to solving univariate polynomials over integers. This means that one constructs from $f(x)$ another univariate polynomial $h(x)$ that contains all small modular roots of $f(x)$ over integers.

In the following, we will show how to find the polynomial $h(x)$. First, we construct a collection C of polynomials $g_1(x), \ldots, g_w(x)$ that all have the small roots x_0 modulo p^m. As an example, the collection is

$$\begin{aligned} g_i(x) &= N^{m-i} f^i(x), \text{ for } i = 0, \ldots, m \\ g_{m+i}(x) &= x^j f^m(x), \text{ for } j = 1, \ldots, t \end{aligned}$$

where t is an integer that has to be optimized as a function of m. Note that the ith polynomial of C is a polynomial of degree i.

Secondly, we construct the lattice L that is spanned by the coefficient vectors of $g_j(xX)$ with $x_0 \le X$. Since the polynomials of C are ordered in strictly increasing order of their degree k, the basis B of L can be written as a lower triangular matrix. For example, if $f(x) = x + a$, and $m = 2, t = 2$, the coefficient matrix is the following.

$$B = \begin{pmatrix} N^2 & 0 & 0 & 0 & 0 \\ Na & NX & 0 & 0 & 0 \\ a^2 & 2aX & X^2 & 0 & 0 \\ 0 & a^2X & 2aX^2 & X^3 & 0 \\ 0 & 0 & a^2X^2 & 2aX^3 & X^4 \end{pmatrix}$$

Clearly, the dimension of the lattice L is $m + t + 1$ and the determinant of the lattice is

$$\det(\mathcal{L}) = X^{\frac{(m+t)(m+t+1)}{2}} N^{\frac{m(m+1)}{2}}.$$

Therefore, using LLL algorithm [13] we can get a vector $\mathbf{z}$ such that

$$\|\mathbf{z}\| < 2^{\frac{m+t}{4}} \det(\mathcal{L})^{\frac{1}{m+t+1}}. \tag{6}$$

This vector yields a polynomial $h(xX)$ with small coefficients. Notice that p^m divides all $g_j(x_0)$ and $h(x)$ is an integer linear combination of $g_j(x)$. Therefore, p^m divides $h(x_0)$. From Lemma 1 below, if there exists a polynomial $h(x)$ such that its norm is less than p^m and $h(x) = 0 \mod p^m$, then $h(x) = 0$ holds over integers. Hence, we wish the norm of the first vector in LLL reduced basis to be less than p^m, so that the roots of $h(x)$ over integers must contain the solution x_0 such that p divides both N and $a + x_0$.

LLL. *([13]) Let L be a lattice of dimension n. Given a lattice basis $(b_1, b_2, \ldots, b_n)$, the LLL algorithm finds a reduced basis $(u_1, u_2, \ldots, u_n)$ in polynomial time such that*

$$\|u_1\| \leq \|u_2\| \leq \ldots \leq \|u_i\| \leq 2^{\frac{n(n-1)}{4(n+1-i)}} \det(L)^{\frac{1}{n+1-i}} \tag{7}$$

Lemma 1. (Howgrave-Graham [9]) *Let $h(x) \in \mathbb{Z}[x]$ be an integer polynomial consisting of at most ω monomials. Suppose that the following conditions hold.*

- *$h(x_0) \equiv 0 \mod R$ for some $|x_0| \leq X$ and some positive integer R*
- *$\|h(xX)\| < \frac{R}{\sqrt{\omega}}$*

Then $h(x_0) = 0$ holds over integers.

Proof. Let $h(x) := \sum_i h_i x^i$. Then,

$$\begin{aligned} |h(x_0)| &= \Big|\sum_i h_i x_0^i\Big| = \Big|\sum_i h_i X^i \Big(\frac{x_0}{X}\Big)^i\Big| \leq \sum_i \Big|h_i X^i \Big(\frac{x_0}{X}\Big)^i\Big| \\ &\leq \sum_i |h_i X^i| \leq \sqrt{\omega} \cdot \|h(xX)\| < R \end{aligned}$$

Since $h(x_0) \equiv 0 \mod R$, this implies that $h(x_0) = 0$ over the integers. □

Therefore, using equation (6) it gives the condition:

$$2^{\frac{m+t}{4}} \det(\mathcal{L})^{\frac{1}{m+t+1}} \leq \frac{p^m}{\sqrt{m+t+1}}$$

This leads to

$$2^{\frac{m+t}{4}} \Big(X^{\frac{(m+t)(m+t+1)}{2}} N^{\frac{m(m+1)}{2}}\Big)^{\frac{1}{m+t+1}} \leq \frac{p^m}{\sqrt{m+t+1}}$$

For a balanced RSA modulus N, $p \approx N^{\frac{1}{2}}$, the inequality can be written as

$$X^{\frac{(1+\tau)^2}{2}m^2+o(m^2)}N^{\frac{m^2}{2}+o(m^2)} \leq N^{\frac{1+\tau}{2}m^2+o(m^2)}$$

If we let all terms of order $o(m^2)$ contribute to ϵ, then we see that for $m \to \infty$, this bound reduces to

$$X^{(1+\tau)^2} < N^{\tau-\epsilon}$$

Let $X = N^{\beta}$. Since we put all terms that don't depend on N in error term ϵ, we see that the asymptotical bound to satisfy is

$$\beta(1+\tau)^2 < \tau$$

or equivalently

$$\beta\tau^2 + (2\beta - 1)\tau + \beta < 0.$$

The left hand side polynomial has its maximum for $\tau = \frac{1}{2\beta} - 1$. Substituting this value, we obtain $\beta < \frac{1}{4}$.

That means that for 1024-bit RSA modulus N when the total size of unknowns is at most 256 bits. In our new attack, for ISO/IEC 9796-2 with $k_h = 160$, the size of the unknown lower bits of $m[1]$ can be as large as 96 bits. In our simulation tests in section 4, the new attack can work using LLL algorithm on 22-dimensional lattice in one second if r is a 83-bit number.

3.2 Two Faults Modulo Different Factors

In this subsection, we discuss the case that two faulty signatures modulo different factors are given. Assume that we have two faulty signatures ρ and ρ', such that $\rho^e = \mu(m) \mod p$ and $\rho'^e = \mu(m') \mod q$. From the analysis in previous section, this gives two equations:

$$\begin{aligned} x_1 + a_1 &= 0 \mod p \\ x_2 + a_2 &= 0 \mod q \end{aligned}$$

where small unknowns x_1, x_2 are bounded by $x_1 \leq X_1, x_2 \leq X_2$ and $a_1, a_2 \leq N$. Multiplying the above two equations, we get a bivariate equation modulo N,

$$x_1x_2 + a_2x_1 + a_1x_2 + a_1a_2 = 0 \mod N.$$

Then x_1, x_2 can be solved by using the technique in [12] under the asymptotical condition $X_1X_2 \leq N^{\frac{2}{3}}$ holds. When $X_1 = X_2 = X$, that condition equals to $X \leq N^{\frac{1}{3}}$. That is to say, if the total unknown part x containing the unknown part of message and hash function part is smaller than $N^{\frac{1}{3}}$, one can factor N efficiently given two faulty signatures modulo different factors.

3.3 Extension to Multiple Faults Modulo the Same Factor

CJKNP gave a multiple fault attack [4], whose complexity increases exponentially with the number of faulty signatures. In [6], the authors described a simpler multiple fault attack, in which the lattice dimension is the number of faults plus one, and gave a bound $\delta \leq \frac{1}{2} - \frac{2}{\ell}$. For more details we refer the reader to [6].

In order to improve the bound of the unknown part of random nonce, we will show how to extend to multiple faults. More precisely, given ℓ faulty signatures, similar to analysis described in above, one has a collection of equations:

$$x_i + a_i = 0 \mod p, \quad \text{for } 1 \leq i \leq \ell \tag{8}$$

where a_i's are known and x_i's are unknown and small. Our goal is to recover the factor p.

In the following, we will adopt the technique of solving simultaneous Diophantine approximation problem. Let X be a suitable bound such that $x_i \leq X$ for all $i \leq \ell$. We construct the lattice $\mathcal{L}$ spanned by the rows of the following matrix

$$M = \begin{pmatrix} X & a_1 & a_2 & \dots & a_\ell \\ & N & & & \\ & & N & & \\ & & & \ddots & \\ & & & & N \end{pmatrix}$$

The dimension of the lattice $\mathcal{L}$ is $\ell + 1$ and the determinant of $\mathcal{L}$ is $N^\ell X$. From the Gaussian heuristic, the length of the smallest vector of the lattice $\mathcal{L}$ is roughly $\sqrt{\ell+1} \cdot (N^\ell X)^{\frac{1}{\ell+1}}$. From (8), there exit integers k_i for all $i \leq \ell$ satisfying $a_i + x_i = k_i \cdot p$. Therefore, in the lattice $\mathcal{L}$, we have the vector $(qX, qx_1, qx_2, \cdots, qx_\ell) = (q, -k_1, -k_2, \cdots, -k_\ell) \cdot M$. Its Euclidean norm is approximately bounded by $\sqrt{\ell+1} \cdot qX$. If the bound is less than the heuristic shortest vector length, i.e., $qX \leq (N^\ell X)^{\frac{1}{\ell+1}}$, the vector $(qX, qx_1, qx_2, \cdots, qx_\ell)$ in lattice $\mathcal{L}$ is probable to be the shortest vector. Assuming $X = N^\delta$, the condition is equal to

$$\delta \leq \frac{1}{2} - \frac{1}{2\ell}$$

That means that, if all x_i's in equation (8) satisfy $x_i \leq N^{\frac{1}{2}-\frac{1}{2\ell}}$, we can apply LLL algorithm to the above lattice and heuristically obtain a shortest vector $\mathbf{v}$. This leads to factor N by computing GCD of the two first components of $\mathbf{v}$.

4 Simulation Results

We have simulated the fault attacks described above as follows. First, we generate a correct $\rho_p = \mu(m)^d \mod p$ and a random $\rho_q \in \mathbb{Z}_q$, then using CRT compute a faulty signature ρ with 160-bit Hash function. Secondly, we compute $(\rho^e - s)2^{-8} \mod N$ which is denoted as a, where s is a known value as in section 3.1. We use the NTL library [17] LLL algorithm on a 2Ghz Intel notebook.

Notice that, although CJKNP discussed that UMP is general, we only compare new simulation results with previous attacks in [4, 6] under the assumption that UMP is in lower order bits.

4.1 Single-Fault Attack Simulations

Table 1: Single fault attack in [4]

modulus size k	UMP size k_r	m_{old}	t_{old}	ω_{old}	time
1024	6	10	3	66	4m
1024	13	13	4	105	51m
1536	70	8	2	45	39s
1536	90	10	3	66	9m
2048	158	8	2	45	55s

We see that for 1024-bit RSA modulus, in Table 1, when the UMP size k_r is 13, it requires more than 50 minutes to excute LLL algorithm on 105-dim

Table 2: New single attack

modulus size k	UMP size k_r	m_{new}	t_{new}	ω_{new}	time
1024	45	2	3	6	0.02s
1024	68	4	5	10	0.06s
1024	83	10	11	22	0.97s
1536	147	2	3	6	0.03s
1536	181	4	5	10	0.08s
1536	205	10	11	22	1.4s
2048	240	2	3	6	0.05s
2048	295	4	5	10	1.32s
2048	326	10	11	22	2.13s

lattice. In fact, exhausting a 13-bit randomizer takes 0.13 seconds. In new attack in Table 2, k_r can be 83 for only 22-dim lattice to be reduced in one second, that can not be exhaustively searched in available time. For 2048-bit RSA modulus, the attack in Table 1 can deal with 158-bit UMP in about one minute, while new attack can tackle 326-bit UMP in two seconds.

Therefore, compared with the results in two tables, new attacks can deal with more bits and are more efficient for larger modului under our attacking scenario.

4.2 Multiple-Fault Attack Simulations

From Table 3, for small $\ell \leq 4$, in our simulation results, the new attacks can work well while previous attacks in [6] are invalid since the bound $\delta < \frac{1}{2} - \frac{2}{\ell}$ is nonpositive. Moreover, the asyptotic bound $\delta < \frac{1}{2} - \frac{1}{2\ell}$ in new attacks seems more natural.

For large ℓ, the new attack has the asymptotic bound closer to $\frac{1}{2}$ than former attacks, i.e., when $\ell = 70$, $k_r + k_h$ in [6] is 0.450 while one of new attack is 0.488. For a 1024-bit RSA modulus and 160-bit hash value, the new attack can deal with the 340-bit UMP better than 300-bit UMP in previous attack. Therefore, it is considerably easier to deal with cases when $\gamma + \delta$ approaches to $\frac{1}{2}$ in the new attacks.

From simulations in Table 3, the distance between asymptotic bound and theoretical bound in new attack is much less than previous one, e.g, the

Table 3: New multiple-fault attacks and previous attacks in [6]

ℓ	$\gamma + \delta_{theory,new}$	$\gamma + \delta_{true,new}$	$\gamma + \delta_{theory,[6]}$	$\gamma + \delta_{true,[6]}$
2	0.250	0.247	-	-
3	0.333	0.329	-	-
4	0.375	0.372	-	-
8	0.437	0.434	0.250	0.214
10	0.450	0.447	0.300	0.280
14	0.464	0.461	0.357	0.330
25	0.480	0.478	0.420	0.400
70	0.492	0.488	0.471	0.450

new distance is almost 0.03, however the distance provided by [6] is about 0.2.

5 Conclusion

The paper introduced a new fault attack against a probabilistic version of ISO/IEC 9796-2 with some purticular parts of message unknown. Instead of the general case, we only analysize that some lower order bits of message are unknown to the attacker. This allows to factor the modulus N by computing small roots of univariate polynomial, and the bound of UMPs of message can be improved up to $N^{0.25}$. Furthermore, if more faulty signature are given the attack can deal with longer UMPs of message, which is heuristic but very efficient.

References

[1] M. Bellare, P. Rogaway, The exact security of digital signatures-how to sign with RSA and Rabin. Eurocrypt'96, LNCS 1070, pp. 399-416, 1996

[2] D. Boneh, R. DeMillo, R. Lipton, On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology*, vol. 14(2), pp. 101-119, 2001

[3] D. Coppersmith, Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *Journal of Cryptology*, vol. 10(4), pp. 233-260, 1997

[4] J-S. Coron, A. Joux, I. Kizhvatov, D. Naccache, P. Paillier, Fault attacks on RSA signatures with partially unknown messages, CHES 2009, LNCS 5747, pp. 444-456, 2009

[5] J.-S. Coron, D. Naccache, J. P. Stern, On the security of RSA padding, Crypto'99. LNCS 1666, pp. 1-18, 1999

[6] J-S. Coron, D. Naccache, M. Tibouchi, Fault attacks against EMV signatures, CT-RSA 2010, LNCS 5985, pp. 208-220, 2010

[7] J-S. Coron, D. Naccache, M. Tibouchi, R. P. Weinmann, Practical cryptanalysis of ISO/IEC 9796-2 and EMV signatures, Crypto'09, LNCS 5677, pp. 428-444, 2009

[8] M. Herrmann, A. May, Solving linear equations modulo divisors: on factoring given any bits, Asiacrypt'08, LNCS 5350, pp. 406-424, 2008

[9] N. Howgrave-Graham, Approximate integer common divisors, CALC 2001, LNCS 2146, pp. 51-66, 2001

[10] ISO/IEC 9796-2, Information technology-Security techniques-Digital signature schemes giving message recovery-Part 2: Mechanisms using a hash-funcion, 1997

[11] ISO/IEC 9796-2: 2002 Information technology -Security techniques-Digital signature schemes giving message recovery-Part 2: Integer factorization based mechanisms, 2002

[12] E. Jochemsz, A. May, A strategy for finding roots of multivariate polynomial with new applications in attacking RSA variants, Asiacrypt'06, LNCS 4284, pp. 267-282, 2006

[13] A. Lenstra, H. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, *Mathematische Annalen*, vol. 261, pp. 513-534, 1982

[14] P. Nguyen, J. Stern, Merkle-Hellman revisited: a cryptanalysis of the Qu-Vanstone cryptosystem based on group factorization. Crypto'97, LNCS 1294, pp. 198-214, 1997

[15] P. Nguyen, J. Stern, Cryptanalysis of a fast public key cryptosystem presented at SAC 1997, SAC 1998, LNCS 1556, pp. 213-218, 1999

[16] R. Rivest, A. Shamir, L. Aldeman, A method for obtaining digital signatures and piblic-key cryptosystems. *Communications of the ACM*, vol. 21(2), pp. 120-126, 1978.

[17] V. Shoup, Number Theory C++ Library (NTL) version 5.5.2, http://www.shoup.net/ntl/