International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 11, Issue 2, 2025, PP 23-27
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
DOI: https://doi.org/10.20431/2349-4859.1102003
www.arcjournals.org

Possibility of Fast Signature Generation by Outsourcing

Chol-Nam Ri, Gyu-Chol Kim*

Kim Chaek University of Technology, Pyongyang, DPRK

***Corresponding Author:** ***Gyu-Chol Kim**, Kim Chaek University of Technology, Pyongyang, DPRK*

Abstract: *In this paper, we propose the method to speed up signature generation in RSA by outsourcing. We first divide the signing algorithm into two stages. One is message generating stage and the other is signing stage. Next, we modify the RSA signature so that the bulk of the calculation cost is allocated to message generating stage. This gives the possibility to propose the RSA signature schemes which have fast signature generation and very fast verification. Our schemes are suited for the applications in which a message is generated offline, but needs to be quickly signed and verified online. In this case, RSA-FDH could be applied to our schemes. Even if message is generated online (i.e., message is generated by signer), it would be possible to speed up signature generation by transferring calculation costs to external computing resources (e.g., cloud computing) instead of message generator mentioned above. In this case, RSA-PSS could be applied to our schemes because message salting is done by signer.*

Keywords: *RSA, rebalanced RSA, signature generation, cloud computing, outsourcing*

1. INTRODUCTION

RSA is a public key cryptosystem based on the difficulty of integer factorization problem, where n-bit composite number $N(= pq: p$ and q are the $n/2$-bit prime numbers) is used as a modulus number. The public key e and private key d of RSA satisfy the following equation.

$$e\,d \equiv 1\ mod\ (p-1)(q-1) \tag{1}$$

To sign a message $m \in Z_N^*$, the signer calculates $s = m^d\ mod\ N$. Verification of a signature s on a message m is carried out by checking whether or not $m = s^e\ mod\ N$.

It is easy to increase the signature verification speed by using small public exponent (e.g., $e = 3$ or $e = 65537$) in RSA. More recent researches [2, 3, 4] have occurred to propose the variants of rebalanced RSA which allow the cost of signature generation and verification to be balanced. However, for the following problems, there is no significant improvement to speed up RSA signature generation in practice. First, the key generation schemes of [2], [3] and [4] are not practical, because they cannot use the typical prime generation module. Second, it becomes to be difficult to reduce the CRT exponents even if public exponent is full sized (on the order of N), because powerful lattice based attacks[6, 7] have been proposed in recent years.

From the above considerations, we focused on making both signature generation and verification high speed in RSA. For this purpose, we modified RSA with $e = 65537$(noted as typical RSA) as follows.

We converted the private exponent $d = 65537^{-1}\ mod\ (p-1)(q-1)$ into h, d_1 and d_0 by modifying the RSA equation to

$$e(hd_1 + d_0) \equiv 1\ mod\ (p-1)(q-1) \tag{2}$$

In the signing stage, public parameter h is additionally used and signature generation protocol is changed as follows. First, message generator calculates $m_1 = H(m)^h mod\ N$ from plaintext m and sends $m||m_1$ to signer. (H denotes full domain hash function and $||$ denotes concatenation.) Next, signer calculates signature $s = H(m)^{d_0} {m_1}^{d_1}\ mod\ N(= H(m)^{(hd_1+d_0)}\ mod\ N)$ by using private key d_0 and d_1 instead of original private key d and return $m||s$. The verification (checking whether or not $s^e mod\ n = H(m)$) is identical to typical RSA.

This paper is organized as follows. In Section 2, we briefly review the rebalanced RSA and small CRT exponent attacks. In Section 3, we propose two fast variants of RSA signature(Scheme1 and Scheme2) and analyze their security. In section 4, we present the performance comparison between the proposed schemes and the other RSA variants. Finally we conclude this paper in Section 5.

2. REBALANCED RSA AND ITS VARIANTS

Rebalanced RSA[1] has achieved the fast signature generation at the cost of a significant loss of verification performance because public exponent e is full sized. Hence, the schemes to speed up signature generation without paying high price for verification (i.e., rebalanced RSA schemes with small public exponent) have been proposed with the small CRT exponent attacks related to α[3, 5, 6].

Attack 1 is resulted in finding small root x such that $gcd(A^x mod\ n - m, n) > 1$ where $A = m^e mod\ n$ in rebalanced RSA. Attack1 is not related to α and can factor the modulus n in time $O(z^{1/2} logz)$ [1, 5, Attack8.1]. In this case, $z = min(d_p, d_q)$.

Attack 2 factors the modulus by finding small root (x_1, x_2, x_3, x_4) of the equation $f(x_1, x_2, x_3, x_4) = 0$ given by

$$f = e^2 x_1 x_2 + e x_1 x_4 - e x_1 + e x_2 x_3 - e x_2 - (n-1) x_3 x_4 - x_3 - x_4 + 1 \tag{3}$$

with monomials $1,\ x_1,\ x_2,\ x_3,\ x_4,\ x_1x_2, x_1x_4, x_2x_3, x_3x_4$ and small root

$$\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}\right) = \left(d_p, d_q, k_p, k_q\right), \text{ with } \begin{cases} |x_1^{(0)}| < X_1 = n^{\delta}, \\ \left|x_2^{(0)}\right| < X_2 = n^{\delta}, \\ |x_3^{(0)}| < X_3 = n^{\alpha+\delta-1/2}, \\ |x_4^{(0)}| < X_4 = n^{\alpha+\delta-1/2}, \end{cases}$$

for some known upper bounds X_j, for $j = 1, \ldots, 4$.

And Attack 3 factors the modulus by finding small root (x, y) of the modular equation $f_e(x, y) = 0$ given by

$$f_e = (n - 1)x\, y + x + y - 1\ (mod\ e) \tag{4}$$

with monomials $1, x, y, xy$ and small root

$$(x^{(0)}, y^{(0)}) = \left(k_p, k_q\right), \text{ with } \begin{cases} |x^{(0)}| < X = n^{\alpha+\delta-1/2}, \\ |y^{(0)}| < Y = n^{\alpha+\delta-1/2}, \end{cases}$$

for some known upper bounds X and Y.

Attack 4 is resulted in finding the root $(x_{p,1}, x_{q,1}, x_{p,2}, x_{q,2}, y_p, y_q)$ of the modular equations $f_{p,1}(x_{p,1}, y_p) = f_{q,1}(x_{q,1}, y_q) = f_{p,2}(x_{p,2}, y_p) = f_{q,2}(x_{q,2}, y_q) = h(x_{p,1}, x_{q,1}, x_{p,2}, x_{q,2}) = 0$ given by

$$f_{p,1}(x_{p,1}, y_p) = n + x_{p,1}(n - y_p)\ mod\ e \tag{5}$$

$$f_{q,1}(x_{q,1}, y_q) = 1 + x_{q,1}(y_q - 1)\ mod\ e \tag{6}$$

$$f_{p,2}(x_{p,2}, y_p) = 1 + x_{p,2}(y_p - 1)\ mod\ e \tag{7}$$

$$f_{q,2}(x_{q,2}, y_q) = n + x_{q,2}(n - y_q)\ mod\ e \tag{8}$$

$$\begin{aligned} h(x_{p,1}, x_{q,1}, x_{p,2}, x_{q,2}) &= (n-1)x_{p,1}x_{p,2} + x_{p,1} + n x_{p,2}\ mod\ e \\ &= (n-1)x_{q,1}x_{q,2} + n x_{q,1} + x_{q,2}\ mod\ e \end{aligned} \tag{9}$$

with monomials $1,\ x_{p,1},\ x_{p,2},\ x_{q,1},\ x_{q,2},\ x_{p,1}y_p, x_{p,2}y_p, x_{q,1}y_q, x_{q,2}y_q, x_{p,1}x_{p,2}, x_{q,1}x_{q,2}$ and small root

$$\left(x_{p,1}^{(0)}, x_{q,1}^{(0)}, x_{p,2}^{(0)}, x_{q,2}^{(0)}\ y_p^{(0)}, y_q^{(0)}\right) = \left(k_q - 1, k_q, k_p, k_p - 1, p, q\right),$$

$$\text{with} \begin{cases} |x_{p,1}^{(0)}| < X_{p,1} = n^{\alpha+\delta-1/2}, \\ |x_{q,1}^{(0)}| < X_{q,1} = n^{\alpha+\delta-1/2}, \\ |x_{p,2}^{(0)}| < X_{p,2} = n^{\alpha+\delta-1/2}, \\ |x_{q,2}^{(0)}| < X_{q,2} = n^{\alpha+\delta-1/2}, \\ |y_p^{(0)}| < Y_p = n^{1/2}, \\ |y_q^{(0)}| < Y_q = n^{1/2}. \end{cases}$$

Since no improvements have been introduced so far, Attack 4 is known to be the state-of-the-art small CRT exponent attack [7].

3. THE PROPOSED SCHEME

3.1 Scheme 1

In this scheme, we shift the signature generation costs to message generator without effect on the security of typical RSA.

3.1.1 Key generation

The key generation algorithm takes a security parameter n(typically $n = 2048$).

Step1. Generate two distinct $(n/2)$-bit primes p and q such that $\gcd(65537, (p-1)(q-1)) = 1$ and calculate $N = pq$.

Step2. Select $e = 65537$ and calculate $d = e^{-1} mod\ (p-1)(q-1), d_p = d\ mod\ (p-1), d_q = d\ mod\ (q-1)$ and $h = 2^{\lceil n/4 \rceil}$.

Step3. Find d_{0p}, d_{0q}, d_{1p} and d_{1q} such that $d_p = hd_{1p} + d_{0p}, d_q = hd_{1q} + d_{0q}$ and $0 < d_{0p}, d_{0q}, d_{1p}, d_{1q} < h$.

Step4. Public key is (e, N, h) and private key is $(d_{0p}, d_{0q}, d_{1p}, d_{1q}, p, q)$.

Signature generation is similar to Section1. The only difference is that the signer acts as prover in Step3 and 4, which are needed to be secure against the active attack.

3.1.2 Signature generation

Message generator calculates hash value $m_1 = H(m)^h mod\ N$ from plaintext m and sends $m||m_1$ to signer who returns signature s or special symbol $\perp$ (which means reject) as follows.

Step1. Calculate $m_{0p} = H(m) mod\ p, m_{0q} = H(m) mod\ q, m_{1p} = m_1 mod\ p$ and $m_{1q} = m_1 mod\ q$.

Step2. Calculate $s_p = {m_{0p}}^{d_{0p}} {m_{1p}}^{d_{1p}} mod\ p (= {m_{0p}}^{h_p d_{1p} + d_{0p}} mod\ p)$ and $s_q = {m_{0q}}^{d_{0q}} {m_{1q}}^{d_{1q}} mod\ q (= {m_{0q}}^{h_q d_{1q} + d_{0q}} mod\ q)$.

Step3. Calculate $t_p = {s_p}^e mod\ p$ and $t_q = {s_q}^e mod\ q$.

Step4. If $t_p \neq m_{0p}$ or $t_q \neq m_{0q}$ then return $\perp$.

Step5. Return $s = \left(\left((s_p - s_q)(q^{-1} mod\ p)\right) mod\ p\right) q + s_p$.

Signature verification is identical to typical RSA which has the fastest verification among all the standardized signature schemes.

3.1.3 Security

In Scheme1, $h(= 2^{\lceil n/4 \rceil})$ does not provide any information except for the bit size of private exponent, which has been known to be approximately equal to n. Hence, RSA-FDH can be straightly applied to this scheme and following theorem is satisfied.

3.1.4 Theorem 1

In the random oracle model, Scheme1 is $(t, q_{hash}, q_{sig}, \varepsilon)$-secure under the assumption that RSA-FDH is $(t, q_{hash}, q_{sig}, \varepsilon)$-secure where q_{hash} and q_{sig} are the number of hash queries and signature queries performed by forger and where ε is the probability to break the scheme in time t.

3.2 Scheme 2

In this scheme, we shift the signature generation costs to message generator to obtain extremely fast signature generation.

3.2.2 Key generation

The key generation algorithm takes two security parameters n and k where $k \leq n/2$(typically n=2048 and k=112).

Step1. Generate two distinct $(n/2)$-bit primes p and q such that $\gcd(p-1, q-1) = 2$ and $\gcd\big(65537, (p-1)(q-1)\big) = 1$ and calculate $N = pq$.

Step2. Select $e = 65537$ and calculate $d = e^{-1} mod\ (p-1)(q-1), d_p = d\ mod\ (p-1)$ and $d_q = d\ mod\ (q-1)$.

Step3. Select k-bit numbers d_{0p}, d_{0q}, d_{1p} and d_{1q} such that $\gcd\big(d_{1p}, p-1\big) = 1, \gcd\big(d_{1q}, q-1\big) = 1$ and $d_{0p} \equiv d_{0q}\ mod\ 2$.

Step4. Calculate $h_p = \big(d_p - d_{0p}\big){d_{1p}}^{-1} mod\ (p-1)$ and $h_q = \big(d_q - d_{0q}\big){d_{1q}}^{-1} mod\ (q-1)$.

Step5. Find h such that $h_p = h\ mod\ (p-1), h_q = h\ mod\ (q-1)$ and $0 < h < (p-1)(q-1)$.

Step6. Public key is (e, N, h) and private key is $(d_{0p}, d_{0q}, d_{1p}, d_{1q}, p, q)$.

Signature generation and verification are the same as in Section 3.1. The only issue is that signature generation can be done faster than other RSA variants, because d_{0p}, d_{0q}, d_{1p} and d_{1q} are extremely small.

3.2.3 Security

Unlikely Scheme 1, h provides some information about private exponent d. It is an open problem whether there exists any efficient small CRT exponent attack to Scheme2. However, the only clear thing is that known small CRT exponent attacks to RSA such as Attack1, 2, 3 and 4 cannot be applied to Scheme2. The best known attack can be described in the following theorem.

3.2.4 Theorem 2

Let (N, e) be an RSA public key with $N = pq$(p and q are primes such that $\gcd(p-1, q-1) = 2$ and $\gcd\big(65537, (p-1)(q-1)\big) = 1$) and $e = 65537$.

Further, let $d, d_p, d_q, h, d_0, d_1, h_p, h_q, d_{0p}, d_{0q}, d_{1p}, d_{1q}$ and r be the integers such that

$d = e^{-1} mod\ (p-1)(q-1), d_p = d\ mod\ (p-1), d_q = d\ mod\ (q-1), d \equiv hd_1 + d_0 mod\ (p-1)(q-1), d_{0p} = d_0 mod\ (p-1), d_{0q} = d_0 mod\ (q-1), d_{1p} = d_1 mod (p-1), d_{1q} = d_1 mod\ (q-1), h_p = h\ mod\ (p-1), h_q = h\ mod\ (q-1)$ and $r = \min(\max(d_{0p}, d_{1p}), \max(d_{0q}, d_{1q}))$.

Then given (N, e, h) an adversary can expose the private key d in time $O(rlogr)$.

4. PERFORMANCE COMPARISON

Table 1 shows the signature generation and verification time comparison of typical RSA, rebalanced RSA, Scheme1 and Scheme2. As shown in Table 1, Scheme1 and Scheme2 are approximately 1.7 and 7.9 times faster than typical RSA, respectively, in total processing. Timings were made on 2.4GHz Core(TM) i9-12900 desktop using NTL with GMP Library for 4092 bit modulus and can be treated as a relative guideline.

Table 1. Signature generation and verification time comparison

	Typical RSA	Scheme1	Rebalanced RSA	Scheme 2
Signature generation Time	118ms	69.2ms	27.1ms	14.9ms
Signature verification Time	3.1ms	3.1ms	465ms	3.1ms
Total Processing Time= Max(Signature generation, Signature verification)	118ms	69.2ms	465ms	14.9ms

5. CONCLUSION

We have described the method to increase the signature generation speed in RSA which has the small public exponent for the fast verification. The basic idea is to transfer the calculation costs from signer to external computing resources such as cloud computing. Of course, this idea is only practical for signature schemes where the accuracy of the computed results from outside can be judged at a low computational cost such as RSA with small public exponent.

REFERENCES

[1] Boneh.D , Shacham.H. Fast variants of RSA. *CryptoBytes (The Technical Newsletter of RSA Laboratories)* Vol.5, No.1 .2002. P.1–9.

[2] Galbraith.S.D, Heneghan.C, McKee.J.F. Tunable balancing of RSA. *ACISP* 3574, 2005. P.280-292.

[3] Jochemsz.E, May.A. A polynomial time attack on RSA with private CRT-exponents smaller than $N^{0.073}$. *LNCS* 4622, Springer, 2007. P.395-411.

[4] Sun.H.M, Wu.M.E, Hinek.M.J. Trading decryption for speeding encryption in Rebalanced-RSA. *The Journal of Systems and Software* Vol.82.2009. P.1503-1512.

[5] Hinek.M.J. Cryptanalysis of RSA and its variants. *CRC Press*, 2010. P. 23-27,139-155.

[6] Takayasu.A, Lu.Y, Peng.L. Small CRT-exponent RSA revisited. *Journal of Cryptology*, Vol.32. No.4, 2019. P.1337-1382

[7] Peng.L, Takayasu.A. Generalized cryptanalysis of small CRT-exponent RSA. *Theoretical Computer Science*, Vol.795. 2019. P.432-458

Citation: *Gyu-Chol Kim et., al (2025). "Possibility of Fast Signature Generation by Outsourcing". International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), vol 11, no. 2, 2025, pp. 23-27. DOI: https://doi.org/10.20431/2349-4859.1102003.*