International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-1 Number-2 Issue-2 December 2011

Secure Three Prime RSA from Hardware Fault Attack

Ravi Rai Chaudhary [1], Deepali Kelkar [2], Vikas Arya[3]
PG Research Group, Dept. of Computer Science, M.I.T Ujjain, M.P[1]
PG Dept. of Computer Science, M.I.T Ujjain, M.P[2]
PG Research Group, Dept. of Computer Science, VIET Ghaziabad,U.P[3]

Abstract

RSA is the most widely deployed public key cryptosystem since it was proposed in 1978.It is used for securing web traffic, e-mail, and some wireless devices. Confidentiality of information has been particularly popularized with the explosive growth of the Internet. However from the user, the Internet was based on open network architecture with computer-based nodes and without network security, and thus was vulnerable to attackers and hackers. This paper is dedicated to the attack study of the system. In this paper carry out the study of Chinese Remainder Theorem based fast RSA, Multi- power RSA, Three Prime RSA and Secure Three Prime RSA Cryptosystem. This work has been implemented using JDK1.6 as the programming environment. The output our implementation shown by respected graph. Compared to CRT based Fast RSA, Multipower RSA Three Prime RSA and generate a new counter attack to make secure to The Three Prime RSA from Hardware fault attack.

Keywords

Cryptosystem, RSA, Three Prime RSA

1. Introduction

Modular arithmetic is a fundamental component of many cryptographic schemes. One consequence of this fact is that these schemes contain mathematical properties such as associatively, commutatively and transitivity which may be exploited by both system designers and attackers. In the case of RSA, modular arithmetic allows an adversary to carefully calculate the effect faults, which occur in a signature operation. For the CRT based RSA [4] cryptosystem, used to speed up the decryption, the attacker is able to purposely induce some types of hardware fault into the system. Then the affected computation values may be used to factor the public modulus of the system. This is called hardware fault attack. By this paper carry out the study of the Chinese Remainder Theorem based Fast RSA, Multi-Power RSA cryptosystem, Three Prime RSA cryptosystem and Secure Three Prime RSA. Compared to conventional CRT-based Fast RSA and Multi-Power RSA cryptosystem, the Three-Prime RSA and Secure Three Prime RSA provide higher operation speed. The RSA cryptosystem is the de facto world-wide standard for public key encryption. The RSA cryptosystem, named after its inventors R. Rivest, A. Shamir, and L. Adleman, is the most widely used public-key cryptosystem. It may be used to provide both secrecy and digital signatures and its security is based on the intractability of the integer factorization problem. In this article carry out the study of the Chinese Remainder Theorem based fast RSA [4], Multi-Power RSA [9], Three Prime RSA [6] and Secure Three prime RSA Cryptosystem. This work has been implemented in jdk1.6 as the programming platform. The output of our implementation shown by respected graphs.

This paper is organized as follows: section2: here describes basic Two Prime RSA Algorithm. Section3: Describes Fast RSA algorithm using Chinese Remainder Theorem. In section 4: we describe the Multi Power RSA algorithm. In section 5: we describe Three Prime RSA. In section 6: will describe the comparison between all these algorithms. In section 7: described will introduced of hardware fault attack. In section 8: described about the consequences of the Counter attack on Three Prime RSA and analyzed it using graphs making it more secure.

2. RSA Algorithm

The RSA cryptosystem is the de facto worldwide standard for public key encryption. The RSA cryptosystem, named after its inventors R. Rivest, A. Shamir, and L. Adleman, is the most widely used public-key cryptosystem. It may be used to provide both secrecy and digital signatures and its security is based on the intractability of the integer factorization problem. The RSA system itself is constructed as follows:

Algorithm: Key generation for RSA public-key encryption
SUMMARY: each entity creates an RSA public key and a corresponding private key.
Each entity A should do the following:

1. Generate two large random (and distinct) primes p and q, each roughly the same size.
2. Compute $n = p*q$ and $\varphi = (p-1)(q-1)$.
3. Select a random integer e, $1 < e < \varphi$, such that $\gcd(e, \varphi) = 1$.

4. Use the extended Euclidean algorithm to compute the unique integer d, $1 < d < \varphi$, such that $ed \equiv 1 \pmod{\varphi}$.
5. A's public key is (n, e); A's private key is d.

3. Fast RSA using CRT

Another approach given by [1] and [4] to implement RSA cryptosystem, using the Chinese Remainder Theorem to speed enhancement

Algorithm: SUMMARY: B encrypts a message m for A, which A decrypts.

1) Encryption. B should do the following:
A. Obtain A's authentic public key (n, e).
B. Represent the message as an integer m in the interval [0, n − 1].
C. Compute $c = m^e \bmod n$.
D. Send the cipher text c to A.
2) Decryption. To recover plaintext m from c, A should do the following:
A. Use the private key d to recover $m = c^d \bmod n$. For that compute

$v1 = c^d \bmod p$, and $v2 = c^d \bmod q$.
Using Fermat's theorem,
$v1 = c^d \bmod (p-1) \bmod p$, and $v1 = c^d \bmod (q-1) \bmod q$.
Finally, following the Garner's algorithm, calculate
$C2 = p-1 \bmod q$, and $u = (v2 - v1).C2 \bmod q$.
The final answer is:
$m = v1 + up$.

4. Multi Power RSA

One can further speed up RSA decryption using module of the from $N=p^{b-1}q$ where p and q are n/b bits each [9]. When N is 1024 – bits long we can use at most b=3, i.e. , $N=p^2q$. the two primes p,q are then each 341 bits long.

Algorithm: B encrypts a message m for A, which A decrypts.
1. Encryption. B should do the following:
(a) Obtain A's authentic public key (n, e).
(b) Represent the message as an integer m in the interval [0, n − 1].
(c) Compute c = me mod n.
(d) Send the cipher text c to A.
2. Decryption. To recover plaintext m from c, A should do the following:
(a) Use the private key d to recover $m = c^d \bmod n$.
For that compute
$m_p = m \bmod p$ and $m_q = m \bmod q$.
Next, compute
$s_p = m_p^{dp} \bmod p$ and $s_q = m_q^{dq} \bmod q$ where $dp = d \bmod (p-1)$ and $dq = d \bmod (q1)$. Finally, s is computed as: S = CRT (sp, sq).

5. Three Prime RSA

A different approach is provided by [6] and [7]. In these articles, they are using multi-prime RSA or New three-prime RSA to speed up the decryption of the RSA cryptosystem. In New three-prime cryptosystem, p, q and r are three large prime numbers such that n = p* q* r. The signing process can be computed more efficiently by applying CRT.
Algorithm: B encrypts a message m for A, which A decrypts.
1. Encryption. B should do the following:
a) Obtain A's authentic public key (n, e).
b) Represent the message as an integer m in the interval [0, n − 1].
c) Compute $c = m^e \bmod n$.
d) Send the cipher text c to A.
2. Decryption. To recover plaintext m from c, A should do the following:
(a) Use the private key d to recover $m = c^d \bmod n$.
For that compute
$m_p = m \bmod p$, $m_q = m \bmod q$, and $m_r = m \bmod r$.
Next, compute
$sp = m_p^{dp} \bmod p$, $sq = m_q^{dq} \bmod q$, and $sr = m_r^{dr} \bmod r$ where $d_p = d \bmod (p-1)$, $d_q = d \bmod (q-1)$ and $dr = d \bmod (r-1)$.
Finally, s is computed as:
S = CRT (sp, sq, sr).

6. Comparison

RSA uses arithmetic on integer at least 1024 bit long. Modular arithmetic is a fundamental component of many cryptographic schemes. One consequence of this fact is that these schemes contain mathematical properties such as associatively, commutatively and transitivity which may be exploited by both system designers and attackers. In the case of RSA, modular arithmetic allows an adversary to carefully calculate the effect faults, which occur in a signature operation.

The table below describes the comparison between different variant of RSA through a bar chart here the blue bar indicates the Normal RSA , the green bar indicates the Fast RSA, The cyan bar indicates the Multi power RSA and the red bar indicates the Three Prime RSA operation.

Table 1: Comparison

Phase	Two prime RSA	Fast RSA	Multi power RSA	Three Prime RSA
K1	13	0	1	0
K2	25	7	9	4
K3	25	12	10	4
K4	0	0	1	0
Total Time	63	19	21	8

Here abbreviations are:
K1 = Time to Encrypt the Message
K2 = Time to Decrypt the Message
K3 = Time to Sign Message by Alice
K4 = Time for Verification By BOB

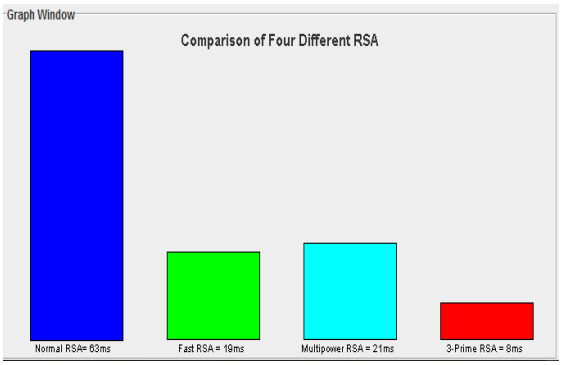

Figure 1: Graph for Comparison of the total time taken by all the algorithms.

7. Hardware Fault attack on Fast RSA and Three Prime RSA

For the CRT-based RSA cryptosystem, the attacker is able to purposely induce some types of hardware fault into the system. Then the affected computation values may be used to factor the public modulus of the system [6]. This is called hardware fault attack.

Attack on Fast RSA using CRT
For the CRT-based RSA cryptosystem, the attacker is able to purposely induce some types of hardware fault into the system. Then the affected computation values may be used to factor the public modulus of the system. This is called hardware fault attack. A faulty signature s' will be computed based on s'p (the faulty value of sp), which is induced by some error due to the interference of the attacker, and the fault-free sq. After intercepting the faulty signature s', the attacker will be able to factor the modulus of the two-prime RSA system by computing q = gcd ((s'e – m) mod n, n) and p = n/q.

Algorithm: RSA Faulty signature generation and computation of prime numbers.
SUMMARY: entity A signs a message m * M. Any entity B can verify A's signature and recover the message m from the signature.
1. Faulty Signature generation: Entity A should do the following:

a) Compute m = R (m), an integer in the range [0, n − 1].
b) Compute s_{pe} = (m_{dp} + 10) mod n and sq = mdq mod n where d_p = d mod (p-1) and dq = d mod (q-1). Here spe is the faulty value of sp.
c) Compute s' = CRT ($s_{pe,}$ sq).

2. Computation of prime numbers: To compute the values of p and q:

a) Calculate q = gcd ((s'e – m) mod n, n).
b) Compute p = n/q.

Hence, the two primes can be calculated by doing this and one can break the RSA cryptosystem using Chinese remainder theorem.

Attack on Three Prime RSA using CRT
For three-prime RSA cryptosystem with $n = p^* q^* r$, the attack involves two faulty signatures s'1 and s'2 , which are computed based on different faulty results of *si (I represents p , q or r)* respectively. The attacker can factor modulus n, only after intercepting two faulty signatures *s'1 and s' 2*, by using the hardware fault attack as follows:

Algorithm: RSA Faulty signature generation and computation of Three Prime numbers.
1. Faulty Signature generation: Entity A should do the following:

A. Compute m = R(m), an integer in the range [0, n − 1].
B. Compute spe = (mdp + 10) mod n, sp = mdp mod n, sq = mdq mod n, sqe = (mdp +10) mod n and sr = mdr mod n where dp = d mod (p-1), dq = d mod (q-1) and dr = d mod(r-1). Here spe and sqe are the faulty values of s_p and $s_{q.}$
C. Compute s1' = CRT ($s_{pe,}$ sq, sr).
D. Compute s2 ' = CRT (sp, $s_{qe,}$ sr).

2. Computation of prime numbers: To compute the values of p, q and r:

a) Calculate qr = gcd ((s1'e – m) mod n, n).
b) Compute pr = gcd ((s2'e – m) mod n, n).
c) Compute p = n/qr = n/gcd ((s1'e – m) mod n, n).
d) Compute q = n/pr = n/gcd ((s2'e – m) mod n, n).
e) Compute r = n/pq.

Hence, all the three primes can be calculated by doing this and one can break the three prime RSA cryptosystem using Chinese remainder theorems.

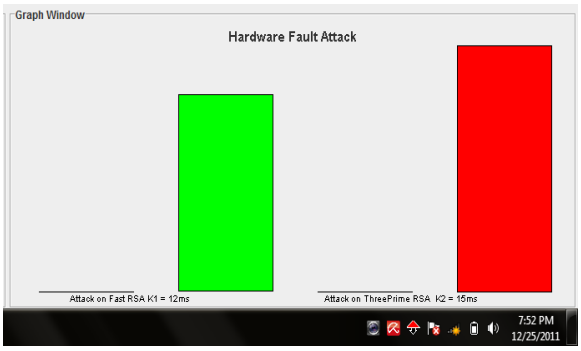

Figure 2: Graph with comparison based on Hardware Fault Attack

Table 2: Comparison based on time taken

Hardware Fault Attack	Fast RSA algorithm	Three Prime RSA algorithm
Total Time Taken	12	15

8. Proposed Work

A new approach to Secure Three Prime RSA using CRT through Hardware Fault Attack

In Zine-Eddine Abid and Wei Wang proposed a new countermeasure method without using kp and kq variables. Then, based on the properties of the FPGA implementation, they carried out a cryptanalysis and comparison study. They used a new key pair of (et, dt), where t is an integer smaller than d, and dt = d +t et = dt-1 (mod φ(n)) ds = d - t.

Based on this, the signature of the proposed j-prime RSA is calculated by the following steps.

Let k = 1, · · · , j, we have

- Step 1: Compute mik = m mod ik.
- Step 2: Compute Xik = mtik mod ik and sikt = mik ds mod (ik-1) mod ik
- Step 3: Com pute sik = (sikt * Xik) mod ik, then the signature s is computed by adding a perturbing summing term to the standard CRT operation as

s = (CRT (si1, · · · , sij) + m̃) mod n,

where m̃ = (Σ jk=1 ((sik * Xik)et mod ik -mik)) * ((n / min(i1, ⋯ ,ij)) +1).

According to this system will give the correct signature when there is no hardware fault attack. When there is hardware fault attack, the proposed countermeasure can ensure security. The attacker can keep s_{ik} correct and insert byte error or random error to make m̃ faulty. If we do not use Proposed countermeasure. Based on a faulty m̃, the complexity for the attacker to obtain any factor of the proposed Three prime RSA is O (n). Based on the above algorithm, the block diagram of the proposed three prime RSA.

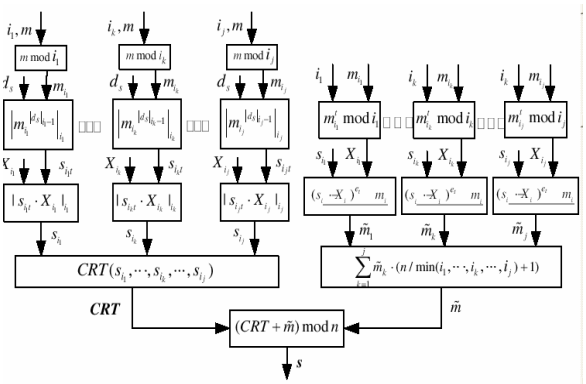

Figure 3: Proposed Cryptanalysis of Secure Three Prime Cryptosystem through Hardware Fault Attack

If one can have a faulty signature with error in the computation of mik, then we can get the value of ik without the complexity O (n) and without the erroneous m̃ with the faulty signature s' as below.

- Compute s' = (CRT (sp, sqe, sr) + m) mod n where mik = (m + 10) mod ik.
- Compute pr = gcd ((s'e – m) mod n, n).
- Compute q = n/pr = n/gcd ((s'e – m) mod n, n).
- This cryptanalytic approach is presented in detail here.
- Step 1: Generate a number n of 1024 bits from a random number generator.
- Step 2: Generate three numbers p1, q1 and r1 of approx. 1024/3 bits
- Step 3: Compute three prime numbers p, q and r of approx. equal bits of p1, q1 and r1.
- Step 4: Compute p-1, q-1 and r-1 to get φ (n).
- Step 5: Compute d*e = 1 mod φ (n) where e = 3.
- Step 6: Compute dt = d + t
- Step 7: Compute et = dt-1 (mod φ(n))
- Step 8: Compute ds = d - t.
- Step 9: Compute mpe = (m + 10) mod .
- Step 10: Compute mq = m mod q.
- Step 11: Compute mr = m mod r.
- Step 12: Compute Xpe = mtp mod p and spte = mpe ds mod (p-1) mod p.
- Step 13: Compute $X_q = m_{tq} \bmod q$ and $s_{qt} = m_q^{ds} \bmod (q-1) \bmod q$.
- Step 14: Compute $X_r = mt\ r \bmod r$ and $srt = m_r^{ds} \bmod (r-1) \bmod r$.
- Step 15: Compute $s_{pe} = (s_{pte} * X_{pe}) \bmod p$.
- Step 16: Compute $s_q = (s_{qt} * X_q) \bmod q$.
- Step 17: Compute $s_r = (s_{rt} * X_r) \bmod r$.
- Let k = 1, · · · , j, Then the faulty signature s' is computed by adding a perturbing summing term to the standard CRT operation as s'=(CRT (s_{pe}, s_q, s_r) + m̃) mod n,
- here m̃ = (Σ j k=1 ((sik * Xik)et mod ik - mik)) * ((n / min(i1, ⋯ ,ij)) +1).
- Step 18: Compute pr = gcd ((s'e – m) mod n, n).
- Step 19: Compute q = n/pr = n/gcd ((s'e – m) mod n, n).

Hence, we can break the Secure three prime RSA but breaking time will be more as compared to Two Prime RSA and get a factor without the complexity O (n).

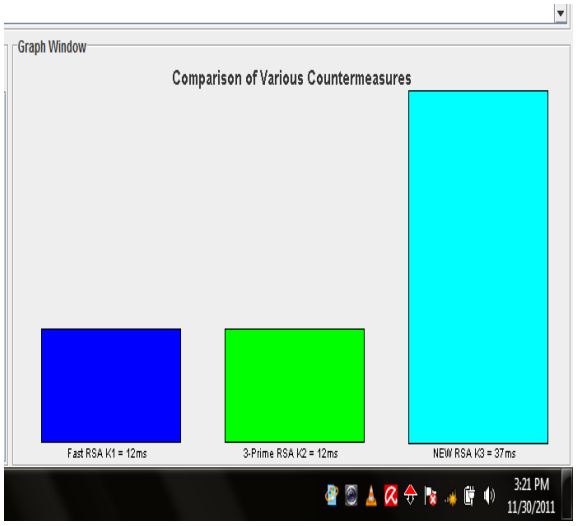

Figure 4: Graph with comparison based on Hardware Fault Attack.

Table 3: Final Comparison

Hardware fault attack	Fast RSA algorithm	Three Prime RSA algorithm	Secure Three Prime RSA
Time Taken	12 ms	15ms	35ms

Hence, we can say that by signing with another approach provided in this article, we can immune the New Three Prime RSA algorithm to a great extent as compare to the other older approaches.

9. Conclusion

So it is clear that this Secure Three-Prime RSA is too fast than the Two Prime RSA but Three Prime RSA it does not provide complete counter against the Hardware Fault Attack. But this article we make a more secure to Three Prime RSA. If one uses an encryption and verifying exponent of 3 as we are with this software, then these operations are quite fast compared with decryption and signing. A speedup by a factor of 6.5 for decryption and signing is significant. The extra algorithmic complexity is minimal, so no one would want an RSA algorithm without this speedup factor. Theory predicts that the CRT decryption should be 7 times as fast. The more complicated algorithm has various sources of extra overhead, so it is not surprising that the full speed up by a factor of 9 is not achieved. In order to factor the Three-Prime CRT-based RSA cryptosystem using Hardware Fault Attack, two faulty signatures and the corresponding factorization calculations are needed and it is proved through our implementation that the Secure Three-Prime RSA cryptosystem is more difficult to be broken than the two-prime Fast RSA.

References

[1] Handbook of Applied Cryptography, by A. Menezes, P. vanOorschot, and S. Vanstone, CRC Press, 1996.

[2] Cay S. Hostmann and Gary Cornell, .Core Java TM2 Volume1- Fundamentals., Seventh Edition, Sun Microsystems, Inc. 2005.

[3] Cay S. Hostmann and Gary Cornell, .Core Java TM2 Volume 2- Advanced Features., Seventh Edition, Sun Microsystems, Inc. 2005.

[4] Cetin Kaya Koc, .High speed RSA implementation, RSA Laboratories., CA, 1994.

[5] Boneh, Dan, and Hovav Shacham. "Fast variants of RSA." CryptoBytes 5, no. 1 (2002): 1-9.

[6] Yonghong Yang, Z. Abid and Wei Wang, .CRT-Based Three-Prime RSA with Immunity Against Hardware Fault Attack., Proceedings of the 4th IEEE International Workshop on System-on-Chip for Real- Time Applications (IWSOC.04), 2004.

[7] Anand Krishnamurthy, Yiyan Tang, Cathy Xu and Yuke Wang, An Efficient Implementation of Multiprime RSA on DSP Processor., Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, & Signal Processing, 2003, Hong Kong.

[8] S. Yen, S. Kim, S. Lim and S. Moon, . RSA Speedup with Chinese Remainder Theorem Immune against Hardware Fault Attack,. IEEE Transactions on computers, vol. 52, pp 461-472, April 2003.

[9] T. Takagi. .Fast RSA-type Cryptosystem Modulo pkq.. In H. Krawczyk, ed., Proceedings of Crypto 1998, vol. 1462 of LNCS, pp. 318.326. Springer-Verlag, Aug. 1998.

[10] Zine-Eddine Abid and Wei Wang, "Countermeasures for Hardware Fault Attack in Multi-Prime RSA Cryptosystems", in the International Journal of Network Security, Vol.6, No.2, PP.190–200, Mar.2008.