Proceedings of ICCES'08, pp.291-297

Faster RSA Algorithm for Decryption Using Chinese Remainder Theorem

G.N. Shinde[1] and H.S. Fadewar[2]

Summary

E-business security is an overarching business issues that, based of an analyzed risks, and establishes the threat acceptance and reduction parameters for the safe use of technology. As an overarching issue, e-business security can be thought of as being absolutely fundamental to the effective and efficient use of Information Technology in support of e-business.

This paper proposed four time faster RSA-CRT algorithm for decryption of data and effective representation of encryption using Chinese Remainder Theorem (CRT) for the data security. The algorithm is implemented in Java source code.

keywords: Encryption, Decryption, RSA, and CRT

Introduction

The most active subjects in the security related communities are the necessary protection against the data thieves. This gives an importance and the value of exchanged data over the Internet or other media types.

This paper tries to increase a fair performance of the most commonly used RSA(Rivest-Shamir-Adleman) algorithm in the data encryption field. Cryptography is usually referred to as "the study of secret", while now a day is most attached to the definition of encryption. Encryption is the process of converting plain text "unhidden" to a cryptic text "hidden" to secure it against data thieves[1]. This process has another part where cryptic text needs to be decrypted on the other end to be understood. Figure 1. shows the simple flow of commonly used encryption algorithms.

Plain Text Chipper Text Plain Text

Encryption Decryption

Figure 1: Encryption-Decryption Flow.

Cryptography Goals

Every security system must provide a bundle of security function that can assure the secrecy of the system. These functions are usually referred to as the goals of the security system. These goals can be listed under the following five categories:

Authentication: This means that before sending and receiving data using the system, the receiver and sender identity should be verified.

[1] Principal, Indira Gandhi (Sr.) College, CIDCO, New Nanded.(M.S.)

[2] Mahatma Gandhi College, Amedpur Dist Latur (M.S.)

Secrecy or confidentiality: Usually this function is how most people identify a secure system. It means that only the authenticated people are able to interpret the message contents and no one else.

Integrity: Integrity means that the content of the communicated data is assured to be free from any type of modification between the end points. The basic form of integrity is packet check sum in Ipv$ packets.

Service Reliability and Availability: Since secure system usually gets attached by intruders, which may affect their availability and type of service to their users. Such system should provide a way to grant their users the quality of service they expect [2].

Non-Repudiation: This function implies that neither the sender nor the receiver can falsely deny that have sent a certain message [3].

RSA Cryptosystem

The RSA system is an asymmetric public key cryptosystem; this means that there is any number of pairs of algorithms (E, D) both defined on the same set of values. E is the public encryption algorithm and D is the private decryption algorithm [4]. These satisfy:

- Encryption followed by decryption works: If c=E(m) is the chipper text corresponding to some plaintext m, then m=D(c) i.e. m=D(E(m))
- Can encrypt efficiently: For any message m, there is an efficient algorithm to calculate E(m).
- Can decrypt efficiently: or any message or cipher text x, there is an efficient algorithm to calculate D(x).
- Public and private keys stay that way: From knowledge of E, there is no efficient way to discover D.
- Signing followed by verifying works: The set of messages m is the same as the set of cipher texts c=E(m), for all m, so that the decryption algorithm can be applied to a message, resulting in what is called a signed message or signature. If s=D (m) is the signature corresponding to some plaintext m, then m=E(s)[4].

Algorithm: RSA cryptosystem construction

Step 1. Choose random large prime integers p and q of roughly the same size, but not too close together.

Step 2. Calculate the product $n = pq$.

Step 3. Choose a random encryption exponent e less than n that has no factors in [5] common with either $p-1$ or $q-1$.

Step 4. Calculate the decryption exponent Ed mod $(p-1)(q-1)=1$.

Step 5. The encryption function is $E(m)=m^e$ mod n, or any message m.

Step 6. The decryption function is $D(c)=c^d$ mod n, for any cipher text c.

Step 7. The public key is the pair of integers (n, e).

Step 8. The private key is the triple of integers (p, q, d).

Chinese Remainder Theorem (CRT)

A common math puzzle is to find a positive integer x.

which when divided by 2, 3, 5 gives remainder 1 and is divisible by 7. Does a solution necessarily exist? If yes, is there more than one solution? Such questions are formally studied using the Chinese Reminder Theorem[6].

Given a system of congruence to different moduli:

$x == a_1$ mod m_1,
$x == a_2$ mod m_2,
$\ldots$
$x == a_r$ mod m_r,

and if each pair of moduli are relatively prime: $gcd(mi,mj)=1$ or $i \neq j$, has exactly one common solution modulo $M=m_1 * m_2 * m_3 \ldots m_r$ and any two solution are congruent to one another modulo M.

Chinese Remainder Theorem in RSA-CRT

In RSA-CRT, it is a common practice to employ the Chinese Remainder Theorem during decryption. It results in a decryption much faster than modular exponentiation. RSA-CRT differs from the standard RSA in key generation and decryption. The value of d, the secret exponent cannot be made short. As soon as $d < N^{0.292}$, RSA system can be totally broken. Keeping this in mind we make use the following scheme[7].

RSA-CRT key generation

1. let p and q be very be two large primes of nearly the same size such that $gcd(p-1,q-1)=2$.

2. Compute $N=p*q$.

3. Pick two random integers d_p and d_q such that $gcd(d_p,p-1)=1$, $gcd(d_q,p-1)=1$ and $d_p == d_q$ mod 2.

4. Find d such that $d == d_p \bmod (p-1)$ and $d == d_q \bmod (q-1)$.
5. Compute $e = d^{-1}$ (mod (N)).

The public key is $< N, e >$ and the private key is $< p, q, d_p, d_q >$. Since $gcd(d_p, p-1) = 1$ and $d == d^p \bmod p-1$, we have $gcd(d, p-1) = 1$. Similarly, $gcd(d, q-1) = 1$. Hence $gcd(d, (N)) = 1$ and by step 5, e can be computed.

To apply the Chinese Remainder Theorem in step 4, the respective moduli have to be relatively prime in pairs for a solution to necessarily exist. We observe that $(p-1)$ and $(q-1)$ are even and that we cannot directly apply the Chinese Remainder Theorem. However, $gcd((p-1)/2, (q-1)/2) = 1$. Since $gcd(d_p, p-1) = 1$ and $gcd(d_q, q-1) = 1$, essentially d_p, d_q are odd integers and $d_p - 1$, $d_q - 1$ are even integers. We have $gcd(d, p-1) = 1$ which implies that d is odd and $(d-1)$ is even [8].

To find a solution to
$d = d_p \bmod p-1$, and $d == d_q \bmod q-1$
We find a solution to
$d - 1 == (d_p - 1) \bmod (p-1)$.
$d - 1 == (d_q - 1) \bmod (q-1)$.
By applying the cancellation law and taking the common factor 2 out, we have
$x = d' == (d-1)/2 == (d_p - 1)/2 \bmod (p-1)/2$,
$x = d' == (d-1)/2 == (d_q - 1)/2 \bmod (q-1)/2$,
Using Chinese Remainder Theorem we find d such that $d = (2 * d') + 1$.

RSA-CRT Decryption

Let M be the plaintext and C the cipher text. If C is not dividable by p and $d_p == d \bmod p-1$, then $C^{dp} == C^d \bmod p$. For decryption we find

1. M_p= C^{dp} (mod p)= C^d (mod p) and. M_q= C^{dq} (mod q)= C^d (mod q).
2. Then using Chinese Remainder Theorem, we find a solution.

M=M_p(mod p)= C^d (mod p),
M=M_q= C^{dq} (mod q)= C^d (mod q).

Security

The CRT version of decryption requires the prime's p and q, as well as the decryption exponent d, so this might seem to be an extra source of insecurity. However, it is simple to factor the modulus n given the decryption exponent d, so no security is lost in using this method [9].

Performance

Theory predicts that the CRT decryption should be four times as faster than RSA. The average decryption time for the normal method is about 0.157 seconds per decryption and for the CRT method is 0.046 second per decryption, giving speedup by a factor of about 3.4 [10].

RSA-CRT Java Implementation

Here is an altered Java implementation of the RSA cryptosystem using Chinese Remainder Theorem to speed up decryption process for this implementation uses the Java ***BigInteger*** library class. This is just a skeleton implementation that creates keys from scratch and uses them, but does not save keys to a file for repeated use, or fetch such keys from the file.

This code implements RSA using 3 Java classes[5]:

- **RSAPublicKey:** The data and methods needed for RSA public keys, with the modulus *n* and exponent *e*, along with a username to keep the keys straight. The important methods are encryption and verification.
- **RSAPrivateKey:** This extends the previous class to add the primes *p* and *q*, and the decryption exponent *e* as data members. Important methods include decryption and signing, along with key generation.
- **RSATest:** A class to test out the system with realistic key size (1024 bits).

A Test Run

```
% Javac RSAublicKey.java
% Javac RSAPrivateKey.Java
% javac RSATest.java
% java RSATest

Message m:
12345678909876543210123456789098765432101234567890
98765432102345678909876543 21

Alice encrypts m; Bob Decryptit:
62338756326752740557713183298294849049819906374359
259444456444183746063611277 7

Original message back decrypted:
12345678909897654321012345678909765432112345678909
87654321012345678909876543 21

Original message back verified:
12345678909876543210123456789098765432101234567890
98765432102345678909876543 21
```

```
Bob signs and encrypts m or Alice: Alice verifies
signature and decrypts:
Message signed and encrypted

Using bob's secrete key and Alice public key:
273343686041287035582131939498827019828348248292 59
6875685127460868394031846 68

Original message back, verified and decrypted,
using Alice's secrete key and bob's public key:
12345678909876543210123456789098765432101234567890 9
876543210234567890987654321
```

Conclusion

E-business security is the necessary protection against data thefting in IT. This paper proposes a four times faster RSA-CRT algorithm for decryption than RSA using Chinese Remainder Theorem. Also it is found that encryption is more effective by using CRT.

References

1. C. Lamprecht "Investigating the efficiency of Cryptographic algorithm in Online Transaction"ISN 1473-804X online, 1473-8031 I.J. of Simulation Vol.7, No.2.
2. http://www.tml.hut.fi/studies/T-110.501/2001/papers/index.html.
3. Ghasem S. Alijani, "Design and Implementation of an Information Security Model for E-Business", Information System Education Journal, Vol 4, no.4 Feb, 8, 2006.
4. A. Sengupta, " E-Commerce Security- A Lie Cycle Approach", Sadhana Vol. 30, Parts 2 & 3, April 2005 pp.119-140.
5. Dirk Balfanz, " A Security Infrastructure or Distributed Java Application".
6. L. Ertaual and N. Chavan, " Security of Ad Hoc Networks and Threshold Cryptography", in MOBIWAC 2005.
7. Alexander May, "Cryptanalysis of Unbalanced RSA with Small CRT-Exponent", CRYPTO 2002, LNCS 2442, pp 242-256, 2002.
8. Johannnes Blomer, Martin Otto, "a new CRT-RSA Algorithm Secure Against Bellcore", CC'03, October 27-30, Washington, DC, USA.
9. Dan Boneh and Hovav Shacham, winter/Spring 2002. Fast Variants o RSA. CryptoBytes- Vol 5, No. 1 Winter/Spring 2002. Pg 1-9.

http:/www.rsasecurity.com/rsalabs/cryptobytes/CryptoBytes_janury_2002_final.pdf.

10. Abdel-Karim Al Tamimi, “Performance Analysis of DataEncryptionAlgorithms”,
http://www.cse.wustl.edu/∼jain/cse56706/ftp/encryption_perf/index.html.