Encryption & Decryption using Secure RSA

Amninder Singh Narota, Roger Lee
SEITI, Department of Computer Science
Central Michigan University, U.S.A
Email: {narot1a, lee1ry}@cmich.edu

***Abstract*—The RSA algorithm is generally 6 step algorithm and the security is based on the randomly selected 2 prime numbers on the assumption that it is easy to find the multiply to prime numbers together, but it is extremely difficult to factor their product. In today's world speed of processing is decreasing at an exponential rate which makes these numbers easily crackable. To overcome this limitation we need to look at the bigger aspects for the application and even larger prime number. Working with even larger numbers are somewhat limitation to computer science since we are limited to at most 64-bit integers.**

This paper introduces the concept and implementation of RSA algorithm for security purpose. We will enhance the performance of the system by adding one more prime numbers to the algorithm and implement external libraries with low computation time to work with large numbers. As the result, it will be more secure to be decrypted by crypto analyst.

***Index Terms*—Cryptography, RSA, MT19937, Mersenne Twister, Miller-Rabin, Prime Numbers, Pseudorandom Numbers**

I. Introduction

Cryptography makes secure web sites and electronic safe transmissions possible. For a we site to be secure all of the data transmitted between the computers where the daw is kept and where it is received must be encrypted. This allows people to do online banking, online trading and make online purchases with their credit cards, without worrying that any of their account information is being compromised. Cryptography is very important to the continued growth of the internet and electronic commerce.

E-commerce is increasing at a very rapid rate. By the turn of the century, commercial transactions on the internet are expected to total hundreds of billions of dollars a year. This level of activity could not be supported without cryptographic security. It has been said that one is safer using a credit card over the internet than within a store or restaurant. It requires much more work to seize credit card numbers over computer networks than it does to simply walk by a table in a restaurant and lay hold of a credit card receipt. These levels of security, though not yet widely used, gibe the means to strengthen the foundation with which e-commerce can grow.

People use email to conduct personal and business matters on a daily basis. E-mail has no physical form and may exist electronically in more than one place at a time. This poses a potential problem as it increases the opportunity for an eavesdropper to get a hold of the transmission. Encryption protects email by rendering it very difficult to read by any unintended party. Digital signatures can also be used to authenticate the origin and the content of an e-mail message.

Cryptography is also used to regulate access to satellite and cable tv. Cable tv is set up so people can watch only the channels they pay for. Since there is a direct line from cable company to each individual subscriber's home, the cable company will only send those channels that are paid for. Many companies offer pay-per-view channels to their subscribers. Pay-per-view cable allows cable subscribers to "rent" a movie directly through the cable box. What the cable box does is decode the incoming movie, but not until the movie has been rented. If a person wants to watch a per-per-view movie, he calls the cable company and requests it. In return, the cable company sends out a signal to subscriber's cable box, which unscrambles (decrypts) the requested movie.

To achieve security there are two ways in which we can achieve.

- Encrypted file transfer.
- Strong secure protocol for transmission

II. Background

There are two types of cryptographic algorithm to accomplish these goals: Symmetric and Asymmetric cryptography. The initial unencrypted dat is referred as normal text. RSA is (*Rivest, Shamir & Adleman*) is asymmetric cryptographic algorithm developed in 1977. It generated two keys: public key for encryption and private key to decrypt message. RSA encrypt and decrypt data, second phase is encryption, where actual process of conversion of plain text to cipher text is being carried out and third phase is decryption, where encrypted text is converted in to plain text at the other side.

Secure RSA prevents files from hackers and help safe transmission of files from one end to other[12]. The algorithm introduced in this report is a modification to the existing RSA algorithm. This algorithm eliminates the need to send product of two random prime numbers in the public key. Further, this algorithm replaces the role of n in encryption and decryption by an integer.

III. Related Work

Cryptography is a process which is associated with scrambling plaintext into cipher text, then back again to plain

text. The key feature of asymmetric cryptography system is encryption and decryption procedure are done with two different keys - public key and private key. Private Key can not be derived with hep of public key that provides much strength to security of cryptography.

This is one main difference between symmetric and asymmetric cryptography, but that difference makes whole process different. This difference is small but it is enough that it has implications throughout the security. Mainly, symmetric cryptography is seen as faster, more lightweight and better suited for applications that have a lot of data to transfer, while at the same time, it is known to be less secure and more open to wider areas of attacks because of maintenance for a private key required. This drawback is removed by asymmetric cryptographic algorithm discussed in following section.

A. Elliptic curve Cryptosystem (ECC)

Over the past 30 years, public key cryptography has become a mainstay for secure communications over the Internet and throughout many other forms of communications. It provides the foundation for both key management and digital signatures. In key management, public key cryptography is used to distribute the secret keys used in other cryptographic algorithms (e.g. DES). For digital signatures, public key cryptography is used to authenticate the origin of data and protect the integrity of that data. For the past 20 years, Internet communications have been secured by the first generation of public key cryptographic algorithms developed in the mid-1970's. Notably, they form the basis for key management and authentication for IP encryption (IKE/IPSEC), web traffic (SSL/TLS) and secure electronic mail.

The majority of public key systems in use today use 1024-bit parameters for RSA and Diffie-Hellman. The US National Institute for Standards and Technology has recommended that these 1024-bit systems are sufficient for use until 2010. After that, NIST recommends that they be upgraded to something providing more security. The question is what should these systems be changed to? One option is to simply increase the public key parameter size to a level appropriate for another decade of use. Another option is to take advantage of the past 30 years of public key research and analysis and move from first generation public key algorithms and on to elliptic curves.

Elliptic Curve Cryptography provides greater security and more efficient performance than the first generation public key techniques (RSA and Diffie-Hellman) now in use. As vendors look to upgrade their systems they should seriously consider the elliptic curve alternative for the computational and bandwidth advantages they offer at comparable security. It consists of both encryption and signature algorithms.

B. ELGamal System

The ElGamal System [4] provides an alternative to rya for public key encryption.

1) Security of RSA depends on the presumed difficulty of factoring large integers.
2) Security of ElGamal algorithm depends on the difficulty of computing discrete logs in a large prime modulus.

The ElGamal signature algorithm is similar to encryption algorithm in that the public key and private key have the same form; however encryption is not the same as signature verification, nor is decryption the same as signature creation. Signature creation depends on the ElGamal signature algorithm. The main disadvantage of ElGamal is the need for randomness and it's slower speed. ElGamal has the disadvantage that the cipher text is twice as long as the plain text. ElGamal is not semantically secure.

C. DSS (Digital Signature Standard)

A digital signature is represented in a computer as a string of binary digits. A digital signature[9] is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. An algorithm provides capability to generate and verify signatures. Signature generation[9] makes use of a private key to generate a digital signature. Signature verification[9] makes use of a public key which corresponds to, but is not the same as, the private key. Each user possesses a private and public key pair. Public key are assumed to be known to the public in general. Private keys are never shared. Anyone can verify the signature of a user by employing the user's public key. Signature generation can be performed only by the possessor of the private key.

The advantages of this system are

- The length of signature is shorter.
- The key generation is faster.
- The processing time code is less.

Drawbacks of DSS are

- DSS and RSA are not compatible.
- The verification process is slower than RSA

D. Diffie-Hellman key agreement protocol

Although DiffieHellman key agreement itself is an anonymous (non-authenticated) key-agreement protocol, it provides the basis for a variety of authenticated protocols, and is used to provide perfect forward secrecy in Transport Layer Security's ephemeral modes (referred to as EDH or DHE depending on the cipher suite). In the original description papers, the Diffie-Hellman exchange by itself does not provide authentication of the communicating parties and is thus susceptible to a man-in- the-middle attack. An attacking person in the middle may establish two different Diffie-Hellman key exchanges, with the two members of the party "A" and "B", appearing as "A" to "B", and vice versa, allowing the attacker to decrypt [5] (and read or store) then re-encrypt the messages passed between them. [5] A method to authenticate the communicating parties to each other is generally needed to prevent this type of attack. In the original description papers, the Diffie-Hellman exchange by itself does not provide authentication of the communicating parties and is thus susceptible to a man-in-the-middle attack. An attacking person in the middle may

establish two different Diffie-Hellman key exchanges, with the two members of the party "A" and "B", appearing as "A" to "B", and vice versa, allowing the attacker to decrypt [5] (and read or store) then re-encrypt the messages passed between them. A method to authenticate the communicating parties to each other is generally needed to prevent this type of attack. Secure Sockets Layer (SSL)/Transport Layer Security (TLS), Diffie-Hellman protocol is used in Secure Shell (SSH), Internet Protocol Security (IPSec), Public Key Infrastructure (PKI).

E. The Sieve of Eratosthenes

Eratosthenes gave a method to generate all prime numbers between 1 and n. The algorithm is as follows:

STEP 1: Write down all numbers from 2 to n.
STEP 2: Take the first uncrossed number, say P and cross all multiple of P, except P.
STEP 3: Repeat STEP 2 until no number crosses out.

Since the time complexity is very large for this algorithm, this method was not reasonable for this project.

Following is the algorithm[10]:

```
input: n
output: list of primes
eratos(n)
  a[1] := 0
  for i := 2 to n do a[i] := 1
  p := 2
  while p2 < n do
    j := p2
    while j < n do
      a[j] := 0
      j := j+p
    repeat p := p+1 until a[p] = 1
  return a
```

The complexity of the algorithm is $O(n(\log n)(\log\log n))$[11] bit operations with a memory requirement of $O(n)$.Time complexity in RAM machine model is $O(n \log\log n)$ operations; this is a direct consequence of the fact that the prime harmonic series asymptotically approaches $\frac{1}{(ln(\ln(N)))}$. The segmented version of the sieve of Eratosthenes, with basic optimizations, uses $O(n)$ operations and $O(n^{\frac{1}{2}} \log\log n/\log n)$ bits of memory.

F. Fermat's Little Theorem

"*Let p be a prime which does not divide the integer a, then* $a^{p-1} = 1 \mod p$". The result is trivial (both sides are zero) if p divides a. If p does not divide a, then we need only multiply the congruence in Fermat's Little Theorem by a to complete the proof. This theorem satisfies only the subset of prime numbers since there are few composite numbers which also satisfy this property. Those numbers were discovered by Robert Carmichael. The smallest carmichael is 561 ($3 \times 11 \times 17$). In 1994 it was proved that there are infinite carmaichael numbers.

G. Gauss Theorem

Carl Friedrich Gauss considered the question of prime-counting function that gives the number of primes less than or equal to x, for any real number x which is as follows:

$$\frac{\pi(x)\log(x)}{x} \to 1; \text{ as } x \to \infty \tag{1}$$

$$\pi(x) \sim \frac{x}{\log(x)} \tag{2}$$

Equation (1) equates to 1 as x approaches to infinity. For example:

$$\pi(100) = 25$$
$$\frac{100}{\log(100)} \approx 22$$

$$\pi(1000000000) = 50847534$$
$$\frac{1000000000}{\log(1000000000)} \approx 48254942$$

40 years later Gauss came up with even better approximation for $\pi(x)$ which is as follows:

$$\pi(x) \sim Li(x)Li(x) = \int_2^x \frac{1}{\log(t)}\partial(t) \tag{3}$$

for example:

$$Li(1000000000) = 50849234$$

Error difference is just 1700

IV. Method

A. Secure RSA File Transmission

MREA[3] is a asymmetric-key crypto system, meaning that for communication, two keys are required: a public key and a private key. furthermore, unlike RSA, it is one way, the public key is used only for encryption, and the private key is used only for decryption. Following is a key generation algorithm for MREA crypto system. We have removed the drawback using MREA for safe transmission of file from one user to another.

Secure RSA file transmission can be summarized as follows:

STEP 1: *Choose four large prime numbers p, q, r and s randomly and independently of each other. All primes should be of equivalent length.*"
STEP 2: *Compute $n = p \times s, m = r \times s, \phi = (p-1) \times (q-1)$ and $\lambda = (r-1) \times (s-1)$*
STEP 3: *Choose an integer $e, 1 < e < \phi$, such that GCD(e, ϕ) = 1*
STEP 4: *Compute the secret exponent $d, 1 < d < \phi$, such that $e \times d(\mod \phi) = 1$*
STEP 5: *Select an integer $g = m + 1$*
STEP 6: *Compute the modular multiplicative inverse: $\mu = \lambda^{-1}(\mod m)$*

The public key is (n, m, g, e) and private key is (d, λ, μ).

B. Encryption:

Let F be a file to be encrypted where the contents of file are taken into string S. Select random number r, where $r < m$ Compute cipher text as:

$$c = g^{s^e \mod n} \times r^m \mod m^2$$

C. Decryption:

Compute origin message:

$$S = (\frac{c^\lambda \mod m^2 - 1}{m} \times \mu \mod m)^d \mod n$$

We found that STEP 4 of algorithm of Section IV-A can be further enhanced by third prime number. STEP 4 can be written as: $e \times d \mod \phi = 1$, *i.e*

$$e \times d = 1 + x(\phi) \tag{4}$$

$$d = \frac{1 + x(\phi)}{e}$$

since $d < \phi$

$$\frac{1 + x(\phi)}{e} < \phi$$
$$x(\phi) < e(\phi) - 1$$
$$x < \frac{e(\phi) - 1}{\phi}$$

since ϕ is very large $\frac{1}{\phi}$ can be ignored and value of x will be:

$$x < e \tag{5}$$

Value of d in STEP 4 can be written as

$$d = \frac{1 + x(\phi)}{e};\ \text{where } x < e \tag{6}$$

D. Pseudorandom Number Generation

The Mersenne twister[8] is pseudorandom number generator. It is, by far, the most widely used Pseudorandom number generator. It's name derives from the fact that it's period length is chosen to be 24th Mersenne Prime though the "guarantee" isn't there anymore. MT19937 is a variant of the twisted generalised feedback shift-register algorithm. It has passed the DIEHARD statistical tests. MT19937 uses 624 words of state per generator and is comparable in speed to the other generators. The original generator used a default seed of 4357 and "choosing **s** equal to zero".[8]

It is quite possible to have a random generator that produces the exactly same values for two different seeds. This isn't an issue unless we depend on some behaviour of the randomness. The main point of of course being cryptography is crypto graphical random number generators try very hard to be very random even if we run 10 generators in parallel. However, this might defeats the purpose of repeatability. The authors claim speeds 1.5 to 2 times faster than *Advanced Encryption Standard in counter mode*. The values for parameters $w, n, m, r, a, u, s, b, t, c$ and l are taken from *Table II Parameters and k-distribution of Mersenne Twisters*[8].

The MT19937 algorithm is inherently 32-bit, but works nicely on 64-bit systems. While it uses the all 32 bits to express pseudo-random numbers, rand() does not. By design it will only return numbers in the range 0 ... INT32_MAX, effectively using only 31 bits of randomness. This is a well known point of criticism for rand(). This assumes that the compiler is using two's complement for encoding negative numbers.

MT19937 has to following statistical property which makes it preferable over other pseudorandom number generators.[1]

- **Probability Density Function:** $2.386863070221 \times 10^{-7} e^{-1.78980156233 \times 10^{-13}} (x - 2.07283382666440 \times 10^8)^2$
- **Cumulative Distributive Function (CDF):** $\frac{1}{2} erfc(4.230604640398 \times 10^{-7} (2.07283382666440 \times 10^8 - x))$

E. Miller-Rabin Primality Test

In Miller-Rabin primality test the number to be tested say p and y be a non trivial square root of $1 \pmod p$.[7] Then we must have that $y^2 = 1 \pmod p$ and so $(y-1)(y+1) = 0 \pmod p$. This implies that either $y = 1 \mod p$ or $y = -1 \mod p$, which implies that y is a trivial square root.

Thus, if there is a non trivial square root of $1 \mod p$, then p has to composite. For an example of a non trivial square root of a composite, consider $p = 15$. We have that $4^2 = 16 = 1 \pmod{15}$, thus 15 is composite. The fact about non trivial square roots can be used to prove that if p is prime, then for any a relatively prime to p, some power of a from a given set of powers must be -1 or a specific odd power of a must be 1.

If for some a none of the above set of powers is -1 and the specific odd power is not 1, then it must be the fact that p is composite. It can also be shown that for composite p, the chances of finding such a is at least $\frac{3}{4}$. This a is the witness in the primality test and is not necessarily a non-trivial square root of $1 \mod p$.

F. Karatsuba Multiplication

Karatsuba's algorithm for fast multiplication was first published in "*Multiplication of Many-Digital Numbers by Automatic Computers*"[6], Proceedings of the USSR Academy of Sciences. The algorithm space for this algorithm is surprisingly rich. There are many methods methods of integer multiplication beyond what we learnt in third grade and this is one of them. Taking an example of $x = 5678$ and $y = 1234$. Now we will split first number x into two halves as $a = 56$ and $b = 78$ similarly for y, $c = 12$ and $d = 34$

STEP 1: Compute $a \times c = 672$
STEP 2: Compute $b \times d = 2652$
STEP 3: Compute sum of $(a + b) \times (c + d) = 134 \times 46 = 6164$
STEP 4: Compute STEP 3 - STEP 2 - STEP 1 = 2840

STEP 5: Pad result of STEP1 with 4 zeros, pad 2 zeros for result from STEP 4 and add them along with result from STEP 2 and resultant is the required result which is 7006652.

Assuming that we replace two of the multiplications with only one makes the program faster. The question is how fast. Karatsuba improves the multiplication process by replacing the initial complexity of $O(n^2)$ by $O(n^{\log 3})$, which as you can see on the diagram below is much faster for big n.

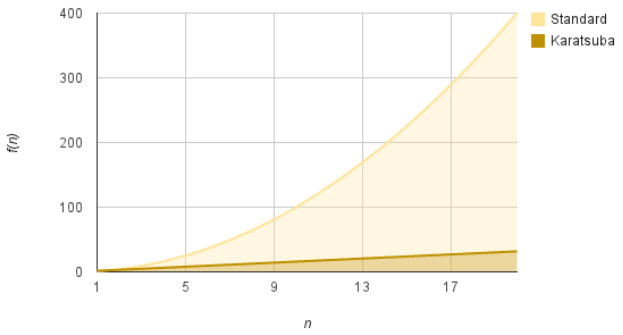

Fig. 1. Karatsuba Multiplication Performance

Figure 1 shows the performance of Karatsuba Multiplication over conventional multiplication.

V. Summary and Further Study

Our method for doing simple benchmarking and Mersenne Twister implementation generates numbers at a rate steadily over 200 million pseudo-random numbers per second. The other implementations we performed around 180M numbers/second, which is significant either. The test code reports mean and standard deviation of performance of MT19937 is shown in Figure 2.[1]

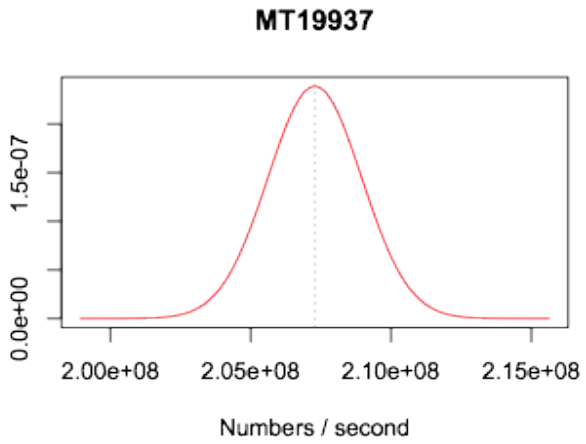

Fig. 2. Mersenne Twister 19937 Performance

Also we tested to generation large prime number and our system showed significant performance to generate 34th mersenne prime with negligible load on the system. Fig. 3 shows the performance graph of the test conducted to compute time (in nano seconds) to generate large primes.

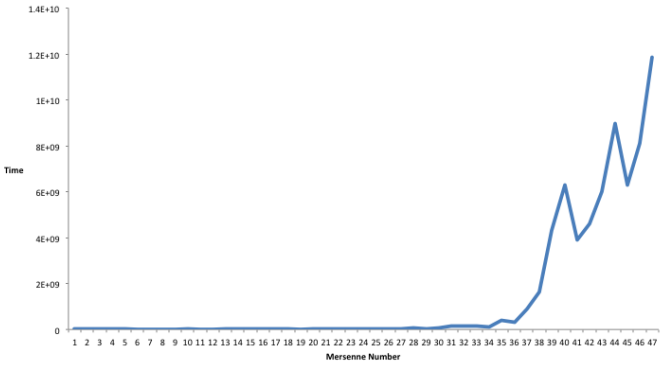

Fig. 3. Time VS Mersenne Prime Number Generation

Karatsuba Multiplication[2] algorithm mentioned in section IV-F was implemented to enhance the performance of multiplication also showed significant performance. For key generation we worked to generate keys of size unto 512 bit keys and showed significant performance based on time of computation.

Secure RSA algorithm is used to encrypt files and transmit encrypted files to other end where it is decrypted. Our system works on the large numbers. It has broad development prospects. The system application was designed to take the efficiency and reusability into account. Great level of security is achieved using this algorithm. Secure RSA algorithm for file transmission algorithm can be used where high security file transmission required in public forums.

References

[1] W. Alpha. Statistical performance of mersenne twister mt19937.

[2] G. CESARI and R. MAEDER. Performance analysis of the parallel karatsuba multiplication algorithm for distributed memory architectures. *Journal of Symbolic Computation*, 21(4–6):467 – 473, 1996.

[3] R. Dhakar, A. Gupta, and P. Sharma. Modified rsa encryption algorithm (mrea). In *Advanced Computing Communication Technologies (ACCT), 2012 Second International Conference on*, pages 426–429, Jan 2012.

[4] K. Huang and R. Tso. A commutative encryption scheme based on elgamal encryption. In *Information Security and Intelligence Control (ISIC), 2012 International Conference on*, pages 156–159, Aug 2012.

[5] E. R. R. Inc. Diffie-hellman key agreement method. *The Internet Society*, 1999.

[6] A. Karatsuba and Y. Ofman. Multiplication of many-digital numbers by automatic computers. *Proceedings of USSR Academy of Sciences*, 145(7):293–294, 1962.

[7] B. Kleinberg. Introduction to algorithms (cs 482). *The Miller-Rabin Randomized Primality Test*, 05 2010.

[8] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, Jan. 1998.

[9] U. D. of Commerce. Digital signature standard. *Federal Information Processing Standard Publications*, pages 186–2, January 2000.

[10] U. of Tennessee. Sieve of eratosthenes.
[11] P. Univeristy. Sieve of eratosthenes.
[12] X. Zhou and X. Tang. Research and implementation of rsa algorithm for encryption and decryption. In *Strategic Technology (IFOST), 2011 6th International Forum on*, volume 2, pages 1118–1121, Aug 2011.