ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

Performance Analysis of Goldwasser-Micali Cryptosystem

Shruthi R[1], Sumana P[2], Anjan K Koundinya[3]

Department of Computer Science and Engineering, R.V. College of Engineering, Bangalore

Abstract: Probabilistic encryption is the use of randomness in an encryption algorithm, so that when encrypting the same message several times it will, in general, yield different ciphertexts. To be semantically secure, that is, to hide even partial information about the plaintext, an encryption algorithm must be probabilistic. The Goldwasser–Micali cryptosystem is an asymmetric key encryption algorithm developed by ShafiGoldwasser and Silvio Micali in 1982. Goldwasser-Micali has the distinction of being the first probabilistic public-key encryption scheme which is provably secure under standard cryptographic assumptions. However, it is not an efficient cryptosystem, as ciphertexts may be several hundred times larger than the initial plaintext. The aim of this paper is to outline the key concepts involved in Goldwasser-Micali encryption algorithm and compare it with RSA . The metrics used for comparison are encryption time,decryption time and size of cipher text with varying plain text sizes which are the key considerations for choosing an encryption algorithm. The reading will be recorded for drawing inferences.

I. INTRODUCTION

One of the drawbacks to the RSA encryption algorithm as originally defined is that it leaks a single plaintext bit in every ciphertext. This bit is the Jacobi symbol of the plaintext, and is either "1" or "−1." Since e is odd it is straightforward to see that J (m/n) =J (m^e/n) for all valid RSA plaintexts m.

This observation pointed to a problem in public key cryptography in general. It should not be possible for an adversary to so much as even distinguish one encryption from another. This problem can be formulated as an experiment. Let an adversary choose any two different plaintexts m_1 and m_2, let the encryption algorithm choose one of the messages randomly, encrypt it, give the resulting ciphertext to the adversary, and then let the adversary guess which message was encrypted. In a truly secure public key cryptosystem the adversary should be able to guess with probability significantly greater than 1/2 which message was encrypted. In RSA, the adversary can choose a messagem1such that J(m_1/n) = 1 and another message m_2 such that J(m_2/n) =−1 and then distinguish correctly everytime.

The GM cryptosystem was the first cryptosystem to provably solve this problem. It was presented by Goldwasser and Micali along with a rigorous definition of security known as semantic security and a proof that the GM cryptosystem is semantically secure against plaintext attacks.

The objectives of the paper are to give a detailed explanation of the concepts involved in Goldwasser-Micali Encryption algorithm, to outline the steps involved in encryption and decryption using Goldwasser-Micali algorithm and to compare and analyze RSA versus Goldwasser-Micali algorithm.

The paper is organised as follows, Section II discusses the various concepts such as Homomorphic Cryptosystem, Quadratic Residues and Jacobi symbols. Section III outlines the pseudocode of the Goldwasser-Micali encryption and decryption algorithm. Section IV describes the details of the implementation which includes hardware and software requirements of the test machine, programming languages and libraries used. Section V deals with the results and analysis which explains the details of the comparisons that were done between RSA and Goldwasser-Micali algorithm. Section VII is Conclusion which gives the outcome of the work carried out and future enhancements.

II. CONCEPTS IN GM

A.XOR Homomorphism

The system is homorphic in that multiplying ciphertexts is equivalent to XORing plaintexts. Note the following congruences:

$$E_{GM}(b_1,r_1;g,N)\cdot E_{GM}(b_2,r_2;g,N)\equiv g^{b_1+b_2}(r_1r_2)^2 \pmod N$$

$$\equiv E_{GM}(b_1\oplus b_2, r_1 r_2;g,N)\pmod N$$

where b1 and b2 are the bits of the input and r1 and r2 are the random blinding factors.

The last equality holds because only the least bit of b1+b2 matters in determining quadratic residuosity, and ⊕ is equivalent to modulo 2 addition[3].

B.Quadratic Residue

A number is a quadratic residue modulo an odd prime p if it is the square of some number modulo p.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

The Legendre symbol is defined as

$$\left(\frac{x}{p}\right) = \begin{cases} 0 & if\ x \equiv 0 \pmod p \\ 1 & if\ x\ is\ a\ quadratic\ residue\ modulo\ p \\ -1 & if\ x\ is\ a\ quadratic\ non\ residue\ modulo\ p \end{cases}$$

By Euler's criterion, we compute $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \pmod p$.

In the case of composite modulus, the Jacobi symbol is used instead of the Legendre symbol.[3]

C.Jacobi Symbol

For N = pq, where p and q are odd primes, the Jacobi symbol is

$$\left(\frac{x}{N}\right) = \begin{cases} 0 & if\ \gcd(x,N) > 1 \\ 1 & if\ \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) \\ -1 & if\ \left(\frac{x}{p}\right) = -\left(\frac{x}{q}\right) \end{cases}$$

Lemma: $x$is a quadratic residue modulo Nif $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = 1$. If $x$is a quadratic residue modulo Nthen it is a quadratic residue modulo N.

Proof: If $x \in$ QR(N) then $x = y^2 + kN = y^2 + kpq$ for some y, k, so $x \equiv y^2 \pmod p$ and $x \bmod p \in$ QR(p). The same holds for q, so $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = 1$. Given x such that $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = 1$, we know that there exist a and b such that $a^2 \equiv x \pmod p$and $b^2 \equiv x \pmod q$. By the Chinese Remainder Theorem, there exists a $y$such that$y \equiv a \pmod p$ and $y \equiv b \pmod q$. Since,$y^2 \equiv x \pmod p$and $y^2 \equiv x \pmod q$, we know that $y^2 \equiv x \pmod N$, and therefore $x \in$ QR(N). If $x \in$ QR(N^2) then $x = y^2 + kN^2$ for some y, k, so $x \bmod N \in$ QR(N).[3]

III.GM Algorithm

A. Key generation for Goldwasser-Micali probabilistic encryption

Each entity creates a public key and corresponding private key. Each entity A should do the following[4]:

1. Select two large random and distinct primes p and q, each roughly the same size.
2. Compute$n = pq$.
3. Select $y \in Z_n$ such that $y$is a quadratic non-residue modulo$n$and the Jacobi sumbol $\left(\frac{y}{n}\right) = 1$(y is a pseudosquare modulo$n$).
4. A's public key is $(n, y)$and$A$'s private key is the pair (p,q).

B. Goldwasser-Micali probabilistic public-key encryption

B encrypts a message m for A, which A decrypts.[4]

1. Encryption: B should do the following:

(a) Obtain A's authentic public key (n, y).

(b) Represent the message m as a binary string m = $m_1 m_2 \ldots m_t$ of length t.

(c) For i from 1 to t do:

i. Pick an$x$ belongs to Z_n^* at random.

ii. If $m_i = 1$ then set $c_i \leftarrow y x^2 \bmod n$, otherwise set $c_i \leftarrow x^2 \bmod n$.

(d) Send the t-tuple c=($c_1, c_2, \ldots c_t$) to A.

2. Decryption: To recover plaintext m from c, A should do the following

(a) For i from 1 to t do:

i. Compute the Legendre symbol $e_i = \left(\frac{c_i}{p}\right)$

ii. If $e_i = 1$ then set $m_i \leftarrow 0$; otherwise set $m_i \leftarrow 1$

(b) The decrypted message is m = $m_1 m_2 \ldots m_t$.[4]

C. Proof that Decryption works

If a message bit m_i is 0, then $c_i = x^2 \bmod n$is a quadratic residue modulo n. If a message bit m_i is 1, then since y is a pseudosquare modulo n, $c_i = yx^2 \bmod n$ is also a pseudosquare modulo n. c_i is a quadratic residue modulo n if and only if c_i is a quadratic residue modulo p, or equivalently $\left(\frac{c_i}{p}\right) = 1$. Since A knows p, she can compute this legendre symbol and hence recover the message bit m_i[4].

D. Security of Goldwasser-Micali probabilistic Encryption

There is a simple reduction from breaking this cryptosystem to the problem of determining whether a random value modulo N with Jacobi symbol $+1$ is a quadratic residue. If an algorithm A breaks the cryptosystem, then to determine if a given value x is a quadratic residue modulo N, we test A to see if it can break the cryptosystem using (x,N) as a public key. If x is a non-residue, then A should work properly. However, if x is a residue, then every "ciphertext" will simply be a random quadratic residue, so A cannot be correct more than half of the time. Furthermore, this problem is random self-reducible, which ensures that for a given N, every public key is just as secure as every other public key.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

The GM cryptosystem has homomorphic properties, in the sense that if c_0, c_1 are the encryptions of bits m_0, m_1, then $c_0c_1 \bmod N$ will be an encryption of $m_0 \oplus m_1$. For this.reason, the GM cryptosystem is sometimes used in more complex cryptographic primitives. Since x is selected at random from Z_n^*, $x^2 \bmod n$ is a random quadratic residue modulo n, and $yx^2 \bmod n$ is a random pseudosquare modulo n. Hence, an eavesdropper sees random quadratic residues and pseudosquares modulo n. Assuming that the quadratic residuosity problem is difficult, the eavesdropper can do no better than guess each message bit. More formally, if the quadratic residuosity problem is hard, then the Goldwasser-Micali encryption scheme is semantically secure.

IV. IMPLEMENTATION ENVIRONMENT

A. *Functional Requirements*

Based on the chosen prime numbers p and q, in the range of 1 to N (where N isproduct of p and q) all the quadratic non-residues are printed. The user has to choose one among the displayed quadratic non residues (z). The public key is (N,z) and private key is(p,q). The user then inputs the plaintext which is encrypted using the keys generated. The ciphertext hence obtained is decrypted to get back the plaintext.

B. *Software Requirements*

Python version 2.7- Python is a remarkably powerful dynamic programming language that isused in a wide variety of application domains. Python is often compared to Tcl, Perl, Ruby, Scheme or Java. Some of its key distinguishing features include:

- very clear, readable syntax
- strong introspection capabilities
- intuitive object orientation
- natural expression of procedural code
- full modularity, supporting hierarchical packages
- exception-based error handling
- very high level dynamic data types
- extensive standard libraries and third party modules for virtually every task
- extensions and modules easily written in C, C++ (or Java for Jython, or .NET languages for Iron Python)
- embeddable within applications as a scripting interface

V. RESULTS AND ANALYSIS

This section with the details of the experiments that were conducted and demonstrates the performance analysis ofGoldwasser-Micali cryptosystem. The parameters considered to evaluate the performance are plaintext size versus ciphertext size, encryption time, decryption time and time to generate Jacobi symbols.

A. *Encryption Time*

The graph of time taken for encryption against size of plaintext in bytes has been plotted as shown in figure 1. The value and N and z were fixed as 2537 and 2 respectively. Encryption time of RSA remained almost constant between plaintext sizes 2 to 6 and 8 to 26 bytes. It increased drastically from around 3 milli seconds to 14 milli seconds when plaintext size was increased from 6 bytes to 8 bytes.

In contrast Goldwasser-Micali had greater varying encryption times reaching a maximum of 26 milli second plain text of 18 bytes and minimum of 3.8 milli seconds for plain text of 4 bytes.

The average encryption time for RSA and Goldwasser-Micali were found to be 12.77 ms and 14.9 ms respectively. This indicates Goldwasser-Micali takes slightly more time compared for encryption than RSA but it is also more secure against attacks.

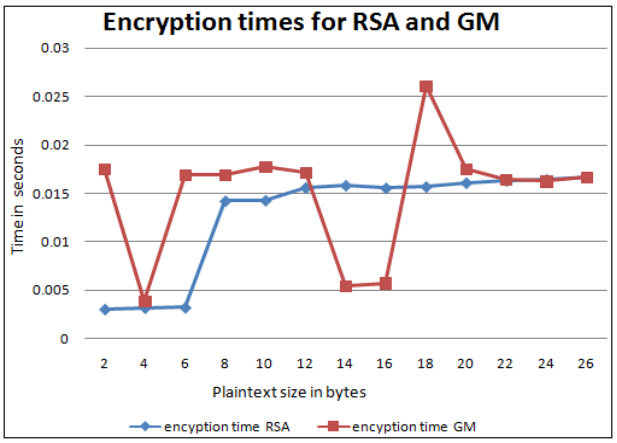

Fig 1: Encryption times for RSA and GM

B. *Decryption Time*

The graph of time taken for decryption against size of plaintext in bytes has been plotted as shown in figure 2. The value and N and z were fixed as 2537 and 2 respectively. Decryption time of RSA remained almost constant between plaintext sizes 2 to 26 bytes.

In contrast Goldwasser-Micali had greater varying encryption times reaching a maximum of 27.5milli second plain text of 6 bytes and minimum of 3.6milli seconds for plain text of 20 bytes.

The average encryption time for RSA and Goldwasser-Micali were found to be 16 ms and 17.65ms respectively.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

This indicates GoldwasserMicali takes slightly more time compared even for decryption than RSA but it is also more secure against attacks.

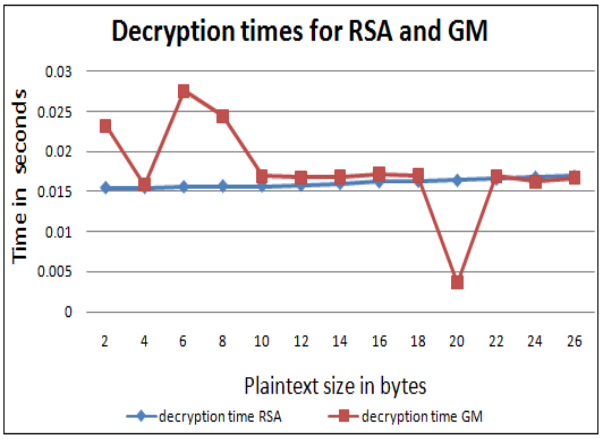

Fig 2: Decryption times for RSA and GM

C. Plaintext size versus Ciphertext size

RSA and Goldwasser Micali show a linear increase in the number of blocks of ciphertext generated for increasing plain text sizes. However the increase is more pronounced in the case of Goldwasser Micali. This can be attributed to the generation of a random value for every bit of plain text in Goldwasser micali.

Hence for applications in which Ciphertext size is more important than security RSA is preferred of GoldwasserMicali.

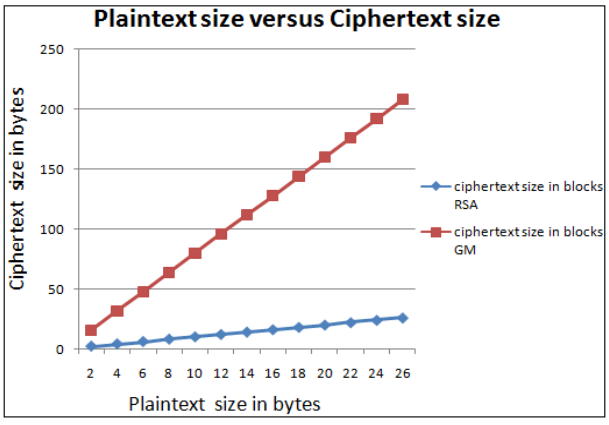

Fig 3: Plaintext versus Ciphertext size for RSA and GM

D. Time taken to generate Jacobi symbols

The graph of time taken to generate Jacobi symbols against various values of N has been plotted as shown in figure 4.

This time does not depend only on the value of N but also the prime factors of considered in the code. It can be seen that as N is increased to higher values the rate of increase in the time to generate Jacobi symbols increases. Hence a drastic increase from 8 seconds to 102 seconds when N's value is increased from 2537 to 11413.

This time is an overhead in Goldwasser micali compared to RSA and hence as we saw previously encryption time in Goldwasser Micali is more compared to RSA.

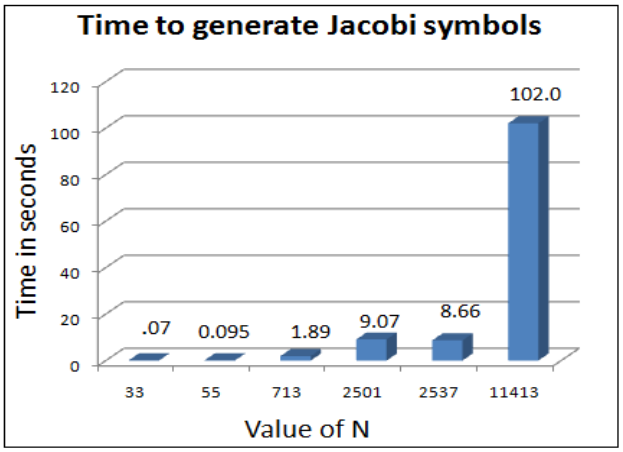

Fig 4: Time to generate Jacobi symbols

VI. FUTURE ENHANCEMENTS

Currently RSA algorithm is used in the SSL layer for encrypting the data. As it is inferred that Goldwasser Micali algorithm is more secure than the RSA algorithm from the above results, we would like to implement this algotithm in the SSL layer to enhance the performance of it.

VII. CONCLUSION

Goldwasser and Micali develop a bit encryption function based on the number theoretic problem of quadratic residuosity. The method has many useful properties, but there is one major drawback: for a given security parameter N, the probabilistic encryption of each bit is N bits long, requires N random bits, and uses several operation on N bit integers.A major disadvantage of the Goldwasser-Micali scheme is the message expansion by a factor of lg n bits. Some message expansion is unavoidable in a probabilistic encryption scheme because there are many ciphertexts corresponding to each plaintext[4].

Paper[6] describes a dense method of probabilistic encryption which, unlike the method of

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

Goldwasser and Micali, is capable of encrypting more than 1 bit at time. For any given k and security parameter N, this new method allows the encryption of k bits of information into an N + k bit ciphertext using N + k random bits and operations on N + k bit integers. Thus, for any desired security parameter N, the ratio of plaintext size to ciphertextsize(as well as to random bits required or to the size of the integers computed upon) can be made arbitrarily close to one.

There are also some applications where one bit at a time probabilistic encryption is unsuitable regardless of efficiency. Paper [6] describes two such applications - non-interactive verifiable secret sharing and a method for obtaining verifiable secret-ballot elections, in which the dense probabilistic encryption method described here can be used while there is no apparent way of developing similar solutions with bitwise probabilistic encryption.

VIII. REFERENCES

[1] Dr.Adam.L.Young and Dr.Moti.L.Young, "Malicious Cryptography exposing cryptovirology," Wiley Publishing, Inc., Indianapolis, Indiana, 2004 cited at http://www.scribd.com/doc/73563406/163/C-1-9-The-Goldwasser-Micali-Algorithm

[2] ShafiGoldwasser and Silvio Micali, "Probabilistic Encryption" Laboratory of computer science, Massachusetts institute of technology, Cambridge, Vol.28, No.2, April 1984

[3] Michael Todd Malkin, "Cryptographic Methods In Multimedia Identification And Authentication," A Dissertation submitted to the department of computer science and the committee on graduate studies of Stanford university, Sept 2006

[4] A. Menezes, P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography," Chapter 8, CRC Press, 1996

[5] Manuel Blum and Silvio Micali, "How To Generate Cryptographically Strong Sequences Of Pseudo Random Bits" Department of Elecrical Engineering and Computer Sciences, University of California, Berkeley, IEEE 1982

[6] Josh Benaloh, "Dense Probabilistic Encryption," Clarkson University

[7] ShafiGoldwasser , OdedGoldreich and Silvio Micali, "How to construct Random Functions," Massachusetts Institute of Technology, Cambridge, Massachusetts, Journal of the Association for Computing Machinay, Vol.33, No.4, October 1986, pp.792-807

[8] ShafiGoldwasser , Andy Yao and Silvio Micali, "Strong Signature Schemes," Lab for Computer Science, MIT, Computer Science Department, University of Toronto, IBM Research, San Jose, ACM 1983

[9] ShafiGoldwasser , Ronald L. Rivest and Silvio Micali, "A DIGITAL SIGNATURE SCHEME SECURE AGAINST ADAPTIVE CHOSEN-MESSAGE ATTACKS,", Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Vol. 17, No. 2, April 1988

BIOGRPHIES

Anjan K Koundinya has received his B.E degree from Visveswariah Technological University, Belgaum, India in 2007 And his master degree from Department of Computer Science and Engineering, M.S. Ramaiah Institute of Technology, Bangalore, India. He has been awarded Best Performer PG 2010 and rank holder for his academic excellence. His areas of research includes Network Security and Cryptology, Adhoc Networks, Mobile Computing, Agile Software Engineering and Advanced Computing Infrastructure. He is currently working as Assistant Professor in Dept. of Computer Science and Engineering, R V College of Engineering.

Shruthi R has received her B.E degree in Computer Science and Engineering from R V College of Engineering , Bangalore, India in 2013. Her areas of research interests include Computer Networks, Cryptography, Network Security, Virtualization and Storage Technologies. She is currently working as a Member Technical Staff in the Research and Development department of VMware Software India Pvt. Ltd.

Sumana P has received her B.E degree in Computer Science and Engineering from R V College of Engineering , Bangalore, India in 2013. Her areas of research interests include Computer Networks, Cryptography, Network Security, AdHoc and Wireless Sensor Networks. She is currently working as a Test Associate Engineer Dell International Services