Comparison among different Cryptographic Algorithms using Neighborhood-Generated Keys

Lalit Singh
M.Tech(CSE)
BTKIT Dwarahat

R.K. Bharti,Ph.D
(Astt. Professor)
BTKIT Dwarahat

ABSTRACT

The prevention of information from unauthorized access is the main concern in the area of cryptography. There are various algorithms i.e. Elgamal, RSA, diffie Hellman, Knapsack are proposed with varying key length and key management methods. The size of encrypted message, key management method and algorithm efficiency is critical in respect of implementation in organizations that store large volumes of data. This paper proposes the comparison among various algorithms for different cryptography algorithms used in generating the neighborhood keys.

Keywords

Cryptography, neighborhood keys, Elgamal, diffie Hellman, RSA, Knapsack, Security, key management.

1. INTRODUCTION

The term “Cryptography” is drawn from the Greek words “crypto” which means “Secret” and “graphy” which means “writing”. In Cryptography, original message is referred to as the plain text while the coded message is known as the cipher text. The process of conversion of plaintext into cipher text is called Enciphering or Encryption; and the process of restoring the plaintext from the cipher text is called Deciphering or decryption [1]. Cryptography is the methods that allow information to be sent in a secure from in such a way that the only receiver able to retrieve this information [2]. A Cryptographic system can also be viewed as a mathematical function which is used for both encryption and decryption [3].

In Cryptography, the process of encryption and decryption implicates the utilization of keys. For encrypting and decrypting the message these keys are used in either symmetric or asymmetric way. Symmetric cryptography is a form of cryptosystem which performs encryption and decryption by employing the use of a unique key. It means an identical key is used by the sender and the receiver for encrypting and decrypting the message respectively. This type of cryptosystem is also known as conventional encryption.

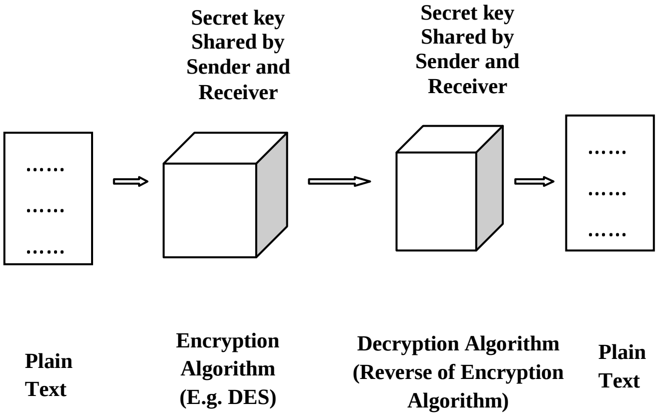
Secret key Shared by Sender and Receiver
Secret key Shared by Sender and Receiver
……
……
……
……
……
……
Plain Text
Encryption Algorithm (E.g. DES)
Decryption Algorithm (Reverse of Encryption Algorithm)
Plain Text

Fig 1. A Simplified Model of Conventional Encryption

Asymmetric cryptography, in contrast, entails the generation of two keys – Public key and private key for encryption and decryption of message respectively [4]. For the encryption of message public key is used and it may be distributed freely, while private key is provided to the intended receiver which decrypts the message securely. Rivest- Shamir- Adleman (RSA), Diffie Hellman key exchange algorithm, Knapsack algorithm and Elgamal algorithm etc. are well known algorithms that employ the use of asymmetric cryptography.

RSA, most frequently used algorithm, is an internet encryption and authentication system which employs the use of an algorithm refined by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977[5]. It has been included as part of the Web browsers from Microsoft and Netscape.

Diffie–Hellman key exchange, one of the earliest practical examples of key exchange, is an explicit technique for exchanging cryptographic keys. This technique refuses two parties, sender and receiver which do not have prior knowledge of each other, to mutually develop a secret shared key over an insecure communication channel [6] which can be used to encrypt subsequent communications by employing the use of a symmetric key cipher[7].

The ElGamal encryption system, based on the Diffie Hellman key exchange, is an asymmetric key encryption algorithm for public-key-cryptography and was described in 1984 by Taher Elgamal. The key length of The Elgamal can range from 256-bit to arbitrarily long [8].

The first algorithm for public key encryption was developed by Ralph Merkle and Martin Hellman and is called Knapsack problem**.** It is a simple problem based on the knapsack problem.

2. DIFFERENT METHODS

2.1 Diffie–Hellman Key Exchange Algorithm

Diffie–Hellman establishes a shared secret that can be used for secret communications by exchanging data over a public network [9,10]. The algorithm itself is limited to the exchange of secret value.

There are two publicly known numbers:
A prime number q and an integer α that is a primitive root of q.
Suppose the user A and B wish to exchange a key.
USER A,
User A select a random integer private X_A [11] , X_A<q
And calculate public key,$Y_A= \alpha^{X_A}$ mod q.
USER B
Similarly user B select private X_B, X_B<q.
And calculate public key, $Y_B=\alpha^{X_B}$ moq q.
Calculation of secret key by user A
K= $(Y_B)^{X_A}$ mod q
Calculation of secret key by user B
K = $(Y_A)^{X_B}$ mod q.
These two calculations produce same results.

Example: suppose prime number, q=5 and the primitive root of q is 2, α=2.
And user A and B select private key X_A=3 and X_B=4.
Then each computes its public keys respectively.
A computes, $Y_A= (\alpha)^{X_A}$ mod q = $(2)^3$mod 5= 8 mod 5 =3.
B computes, $Y_B = (\alpha)^{X_B}$ mod q = $(2)^4$mod 5= 16 mod 5 =1
After they exchange public key, each can compute the common secret key.
A compute, k = $(Y_B)^{X_A}$ mod q = $(1)^3$ mod 5 = 1 mod 5 = 1.
B compute, k = $(Y_A)^{X_B}$ mod q = $(3)^4$ mod 5 = 81 mod 5 = 1.

2.2 ElGamal Encryption Algorithms

ElGamal encryption is one of many encryption schemes which utilize randomization in the encryption process [12]. The Elgamal algorithm is also a public key algorithm, which can be used for both: Digital signature as well as encryption. Its security is based on the difficulty of computing discrete logarithms in a finite field.

To generate a key pair:
First select a prime number p and two random number g and x, so that both g and x are less than p.
Then find out:

$$Y = g^x \text{ mod p.}$$

The public key becomes g,y and p. both g and p can be shared in a group of users. The private key is x.
For encryption
Firstly we require a plaintext message M for encryption and then select a random number k such that k is relatively prime to (p-1).
Then find out:

$$a = g^k \text{ mod p.}$$
$$b = y^k \text{ M mod p.}$$

Then the pair (a,b) becomes the cipher text.
Note: To decrypt (a,b) to find out the plaintext M,
Calculate:

$$M = b/a^x \text{ mod p.}$$

Example: suppose p =11, g = 2 and x = 3.
Then
y = g^x mod p = $(2)^3$ mod 11 = 8 mod 11 = 8.
So the public keys are (2, 8, 11) and the private (secret) key is 3.
Sender receives the public keys (2,8,11).
Chooses random value k = 4 and calculates a and b for the plaintext (M = 7).
a = g^k mod p = 2^4 mod 11 = 16 mod 11 = 5.
b = (y^k M) mod p = ($7*8^4$) mod 11 = 7*4096 mod 11 = 28672 mod 11 = 6.
Sender sends (a,b) as (5,6).
For decryption
M=b*(a^-1/x)) mod p=6*(5^-1/3) mod 11= 6*5^(11-1-3) mod 11= 6*5^7 mod 11=6*78125 mod 11=468750 mod 11=7 =M(plain text).

2.3 RSA

The RSA scheme is block cipher in which the plaintext and cipher text are integers between 0 and n-1 for some n. a typical size for n is 1024 bits, or 309 decimal digits. That is, n is less than 2^{1024} [13, 14].
The public key and private keys are carefully generated using the RSA algorithm. They can be used to encrypt information.

Key generation:
Step 1: Select two prime number, p and q. ($p \neq q$).
Step 2: Calculate, n = p*q.
Step 3: Calculate, $\phi(n)$ = (p-1)*(q-1).
Step 4: Select e such that e is relatively prime to $\phi(n)$ and less than $\phi(n)$.
And
Gcd(e , $\phi(n)$) = 1 , $1< e < \phi(n)$

Step 5: Calculate d such that
$d \equiv e^{-1} (\text{mod } \phi(n))$
So that
d.e[mod $\phi(n)$]=1, is that, d is the multiplicative inverse of e in mod $\phi(n)$.
Step 6: Get public key as KU = {e , n}.
Step 7: Get private key as PK = {d , n}.

Encryption and decryption are of the following form, for some plaintext block M and cipher text block C.
C = M^e mod n.
Similarly
M = C^d mod n.
Where C is the cipher, M is the message, e is the public key, *d* is the private key, and n is the modulus [15].

Example:
Step 1: suppose two prime number p=11 and q=3.
Step 2: calculate, n=p*q=11*3=33.
Step 3: calculate, $\phi(n)$ = (p-1)(q-1) = (11-1)(3-1) = 10*2 =20.
Step 4: choose e = 3.
Check gcd ($\phi(n)$,e) = gcd (20,3) = 1.

Step5: calculate, d
$d*e \bmod \phi(n) = 1$
So that d=7.
Step 6: the public key is {3, 33}.
Step 7: the private key is {7, 33}.

Encryption for plaintext block M=7.
$C = M^e \bmod n = 7^3 \bmod 33 = 343 \bmod 33 = 13.$
Decryption for cipher text block C = 13.
$M = C^d \bmod n = 13^7 \bmod 33 = 62748517 \bmod 33 = 7.$

2.4 The Knapsack problem

The Knapsack problem or Rucksack problem is a problem in combinatorial optimization. Given a pile of item [16], each with different weights is it possible to put some of them in a bag (i.e. knapsack) in such a way that the knapsack has a certain weight.
That is, if m_1, m_2, m_3…….. m_n are the given values and S is the sum, find out bi so that

$S = b_1m_1 + b_2m_2 + b_3m_3 +$ ---------------- $+ b_nm_n$.
Each bi can be 0 or 1.
A 1 indicates that the item is in the knapsack and a 0 indicates that it is not. A block of plain text equal in length to the number of items in the pile would select the items in the knapsack. The cipher text is the resulting sum.

Example: knapsack example

Plain text	0 1 1 0 1 1	1 1 1 0 0 0
Knapsack (w)	1 6 8 12 15 20	1 6 8 12 15 20
Cipher text	0 +6 + 8 + 0 +15+20 = 49	1 + 6 + 8 + 0 + 0 + 0=15

3. EXECUTION TIME DIFFERENCE IN DIFFERENT ASYMMETRIC ALGORITHMS:

Plain text	Execution time for RSA Algo in millisec.	Execution time for Elgamal Algo in millisec.	Execution time for knapsack Algo in millisec
4	7.25	4.34	7.84
5	7.50	5.60	8.48
6	8.02	6.37	8.97
7	7.52	6.04	7.26
8	7.89	5.70	8.54
9	7.90	6.09	8.50
10	8.90	7.58	8.36

4. RESULT & DISCUSSION

After executing for various different type and length of plain text the execution time is calculated by making a corresponding computer program and comparison is made. The table and the line chart shown in figure 2.

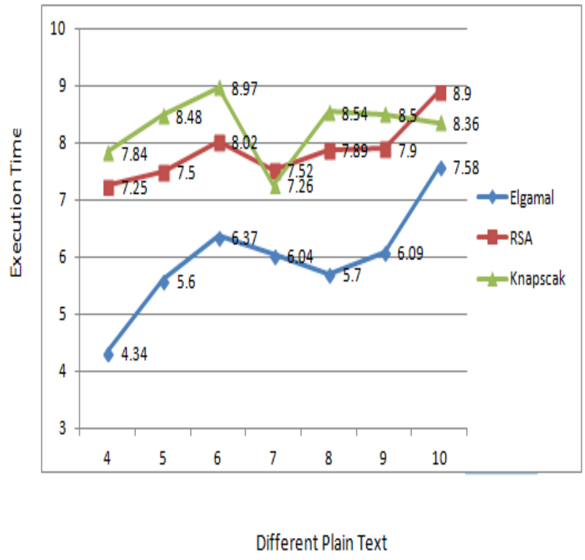

Fig: 2 Execution time is measured in milliseconds

Paper represents the execution time for various algorithms for different plain texts, which reveals that time varying with respect to plain text characteristics. The execution time is also varied with respect to key length. To optimize the efficiency of algorithm an efficient key and its key management is required.

5. CONCLUSION

A new technical findings and viewpoint: efficiency, key length, plain text type dependency is found when implementing the any cryptographic algorithms. The all these technical viewpoints can further extended by focus on using intelligent systems such as Artificial Neural Network, Decision Tree algorithm Genetic Algorithm, etc. To encrypt message using keys generated by these algorithms.

6. REFERENCES

[1] Samuel King Opoku "A ROBUST CRYPTOGRAPHIC SYSTEM USING NEIGHBORHOOD-GENERATED KEYS" International Journal of Research in Computer Science EISSN 2249-8265 Volume 2 Issue 5 (2012) pp. 1-9.

[2] Vishwa gupta, Gajendra Singh and Ravindra Gupta, " Advance cryptography algorithm for improving data security" ijarcsse Volume 2, Issue 1, January 2012.

[3]Vashitva Kumar Srivastava , Amitesh Kumar Srivastava, and Majhar Khan, " A Symmetric Key Cryptographic Algorithm" International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 8, October – 2012.

[4] Prof K.Govinda , Dr.E.Sathiyamoorth " Multilevel Cryptography Technique Using Graceful Codes" Journal of Global Research in Computer Science Volume 2, No. 7, July 2011

[5] B.Persis Urbana Ivy, Purshotam Mandiwa. Mukesh Kumarv"A modified RSA cryptosystem based on 'n' prime numbers" International Journal of Engineering and

Computer Science ISSN: 2319-7242 Volume1 Issue 2 Nov 2012 Page No. 63-66.

[6] M.V.Siva Prasad , Dr. A.Vinaya Babu & G.Satyanarayana "AN ENHANCED AND SECURE PROTOCOL FOR AUTHENTICATED KEY EXCHANGE " Journal of Theoretical and Applied Information Technology.

[7] Makhamisa Senekane, Sehlabaka Qhobosheane and B.M. Taele "Elliptic Curve Diffie-Hellman Protocol Implementation Using Picoblaze" IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.6, June 2011

[8] Mustafa DÜLGERLER & M. Nusret SARISAKAL **"** A SECURE E-MAI APPLICATION USING THE ELGAMAL ALGORITHM: MD MESSAGE CONTROLLER**"** journal of electrical & electronics engineering year 2003 volume 3 number 1

[9] L.Thulasimani & M.Madheswaran "Implementation of an Energy Efficient Reconfigurable Authentication Unit for Software Radio" International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1375-1380.

[10] A.Brillia, D. Jagadiswary, R. Muthu Venkata Krishnan**"**Implementing Cryptographic Techniques in Message Passing Interface Systems**"** International Journal of Science and Modern Engineering (IJISME) ISSN: 2319-6386, Volume-1, Issue-4, March 2013.

[11] Thanuja R & Dilip Kumar S "A NEW APPROACH TO DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM" International Journal of Engineering Research and Applications (IJERA) Vol. 1, Issue 3, pp.534-535

[12] Rashmi singh, shiv kumar " Elgamal's Algorithm in Cryptography" International Journal of Scientific & Engineering Research Volume 3, Issue 12, December-2012

[13] Mehdi Hojabri & Mona Heidari Department of CS and SE Andhra University "Union of RSA algorithm, Digital signature And KERBEROS in cloud security" International Conference on Software Technology and Computer Engineering (STACE-2012), ISBN : 978-93-81693-68-1, 22nd July 2012,Vijayawada.

[14] M. Renukadevi, N. Bhaskar, R. Prabu "ANOMALY PROTECTION USING BATCHING STRATEGIES" Journal of Computer Applications (JCA)ISSN: 0974-1925, Volume IV, Issue 4, 2011.

[15]Sonal Sharma, Jitendra singh yadav and Prashant Sharma "Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm" International Journal of Advanced Research in Computer Science and Software Engineering 2 (8), Volume 2, Issue 8**,** August-2012, pp. 134-138.

[16] Robert M. Nauss "An Efficient Algorithm for the 0-1 Knapsack Problem" *Management Science* September 1976 vol. 23 no. 1 27-31http://mansci.journal.informs.org/content/23/1/27.full.pdf+html