Elgamal’s Algorithm in Cryptography

Rashmi Singh, Shiv Kumar
(M.Tech.) Mewar University
NH - 79 Gangrar,(Rajasthan) - 312901
Ph. 9694430530
Email_id: rashmikamal011@gmail.com

Abstract— Cryptography, over the ages, has been an art practiced by many who have devised ad hoc techniques to meet some of the information security requirements. The last twenty years have been a period of transition as the discipline moved from an art to a science. There are now several international scientific conferences devoted exclusively to cryptography and also an international scientific organization, the International Association for Crypto logic Research (IACR), aimed at fostering research in the area. In this age of viruses and hackers, or electronic eavesdropping and electronic fraud, security is paramount.

El-Gamal encryption/decryption is based on the difficulty of the discrete algorithm problem where it is straight forward to raise numbers of large powers but it is much harder to do the inverse computation of the discrete logarithm. The El-Gamal algorithm depends on certain parameters which are affecting the performance, speed and security of the algorithm.

The importance of these parameters and role it takes in the security and the complexity of the system and the analyzed, particularly the effect of changing the length of the modulo number and the private key number are investigated.

.**Index Terms**— Cryptography, security, public key cryptography, private key cryptography, modulo, encryption, decryption

—————————— ◆ ——————————

1 INTRODUCTION

ElGamal encryption is one of many encryption schemes which utilizes randomization in the encryption process. Others include McEliece encryption (x8.5), and Goldwasser-Micali (x8.7.1), and Blum-Goldwasser (x8.7.2) probabilistic encryption. Deterministic encryption schemes such as RSA may also employ randomization in order to circumvent some attacks (e.g., see x8.2.2(ii) and x8.2.2(iii)). The fundamental idea behind randomized encryption (see Definition 7.3) techniques is to use randomization to increase the cryptographic security of an encryption process through one or more of
the following methods –
Increasing the effective size of the plaintext message space;

•Precluding or decreasing the effectiveness of chosen-plaintext attacks by virtue of a one-to-many mapping of plaintext to cipher-text; and
•Precluding or decreasing the effectiveness of statistical attacks by leveling the a priori probability distribution of inputs..

1.1SECURITY:

The problem of breaking the ElGamal encryption scheme, i.e., recovering m given p,g,(g^x) and a, b is equivalent to solving the Diffie-Hellman problem (see x3.7). Infact, the ElGamal encryption scheme can be viewed as simply comprising a D.

Diffie-Hellman key exchange to determine a session key (g ^ (x*k)), and then encrypting the message by multiplication with that session key. For this reason, the security of the El-Gamal encryption scheme is said to be based on the discrete logarithm problem in Zp*, although such an equivalence has not been proven.

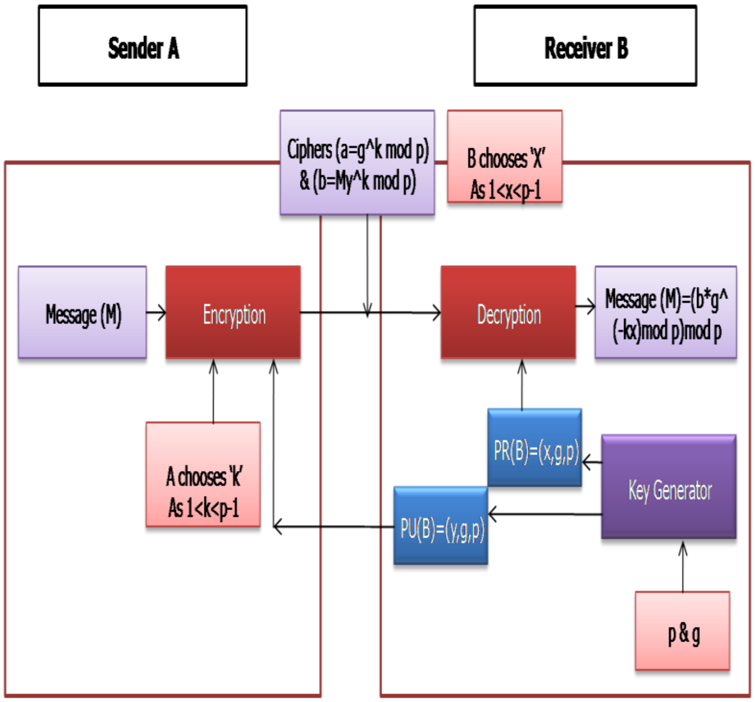
It is critical that different random integers k be used to encrypt different messages. Suppose the same k is used to encrypt two mess ages m1 and m2 and the resulting cipher-text pairs are (a1,b1) and (a2,b2). Then b1/b2= m1/m2, and m2 could be easily computed if m1 were known

1.2 RECOMMENDED PARAMETER SIZES:

Given the latest progress on the discrete logarithm problem in Zp*(x3.6), a 512-bit modulus p provides only marginal security from concerted attack. As of 1996, a modulus p of at least 768 bits is recommended. For long-term security, 1024-bit or larger moduli should be used. For common system-wide parameters even larger key sizes may be warranted.

This is because the dominant stage in the index-calculus algorithm (x3.6.5) for discrete logarithms in Zp* is the pre-computation of a database of factor base logarithms, following which individual logarithms can be computed relatively quickly. Thus computing the database of logarithms for one particular modulus p will compromise the secrecy of all private keys derived using p.

1.3 Model of Elgamal's Algorithm:



2 Description of Elgamal's Cryptosystem:

Taher ElGamal [ElG85] proposed the following way to make Diffie-Hellman into an encryption scheme. The El-Gamal Cryptosystem is implicitly based on the difficulty of finding a solution to the discrete logarithm in F¤p: given a primitive element a of F¤p and another element b.

2.1 Determination of Public key and Private key:

A. Choose a large prime number p such that ø(p) has a large prime factor. Choose g: primitive root of p.
B. Choose an integer x such that 1 <x< ø(p) . Compute y=g^x mod p.
C. The triplet (x,g,p) forms the private key and the triplet (y,g,p) forms the public key of the user'A'.
D. User 'A' keeps the private key (x,g,p) secret and makes the public key (y,g,p) available to all those users with whom 'A' intends to communicate..

2.2 Encryption (Say By User 'B'):

When any user (say user B) possessing A's public key (y,g,p) intends to send a message M(0≤M<p) to user 'A' user 'B' proceeds as follows:-

A. Choose a random integer k such that (1<k< ø(p)).

B. Knowing public key (y,g,p) of intended recipient 'A'. user 'B' computes the cipher-text, which comprises of a pair of integers :-

(g^k mod p , My^k mod p)

C. User 'B' transmits the cipher-text to the intended recipient User 'A'.

The first component of the cipher-text i.e. g^k mod p is called CLUE. It contains clue of the random value k, which is not known to the intended recipient of nth cipher-text. The intended recipient will use the clue for extraction of plaintext from second component of the cipher-text..

2.3 Decryption (Say By User 'B'):

User 'A' receives the cipher-text (g^k mod p, M^k mod p). 'A' proceeds to decrypt the received cipher-text using its private key(x,g,p).
1) Choose (-x)mod ø(p) = ø(p)-x = p-1-x
2) Compute (g^k mod p)^(-x mod ø(p)) mod p = g^ -kx mod p.
3) Compute M= ((My^k mod p).(g^-kx mod p))mod p.

2.4 Working of Elgamal's Algorithm:

Message decrypted at the recipient end
= ((My^k mod p) (g^-kx mod p)) mod p
= ((Mg^kx mod p)(g^-kx mod p)) mod p
= ((Mg^kx g^-kx) mod p = M mod p
=M (same as the message sent)

Thus, the recipient is able to extract the original plaintext message M from the received cipher-text. Proved that the message received at the recipient end is same as the message sent.

2.5 Working of Elgamal's Algorithm:

Suppose a user 'A' computes its private and public keys as follows:-

2.5.1. Determination Public Key and the Private Key

Let p=11; g=2
Let x=5 Y=g^x mod p = 2^5 mod 11 = 32 mod 11=10
Thus, 'A' is the PRIVATE KEY =(x,g,p) = (5,2,11)
And 'A' s public key = (y,g,p)= (10,2,11)

2.5.2. Encryption

Suppose a user 'B' wants to send message M= 3 to 'A' , then 'B' will encrypt M a follows and transmit C to A:-
'B' will select an integer k such that 0<k<ø(p).
Let k=7
C= (g^k mod p , My^k mod p)
= (2^7 mod 11, 3x 10^7 mod 11)
= (128 mod 11, (100x100x100x30)mod 11)
=(7,8)

ElGamal's encryption is based on DLP(Discrete logarithmic problem.)

2.5.3. DECRYPTION

Knowing (x,g,p), user 'A' will decrypt the cipher-text C as follows:-

Compute (-x)mod ø(p)= (-x)mod 10=10-x=10-5=5.

Compute (g^k)^-x mod p =(7)^5 mod 11 = (49x49x7)mod 11

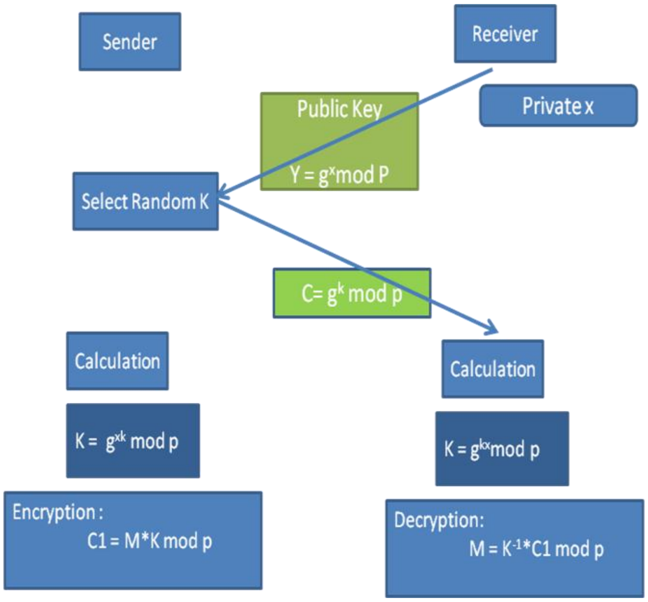
Compute (M.y^k.g^-xk)mod p

= (8x 10)mod 11

=3

=M(Original Message)

2.5.4. Encryption and Decryption in Elgamal's using DH-protocol



2.6 Performance of Elgamal's Algorithm:

2.6.1.Simulation Result

Bit length	(p) length	(g) length	(y) length	(y1) length	(y2) length
1	1	1	1	1	4
10	3	1	2	3	7
20	6	1	6	5	10
50	15	1	15	15	20
100	31	1	30	30	35
128	37	2	37	37	42
130	38	1	37	37	42
150	42	1	42	42	46

The importance of Elgamal's parameters is analyzed by changing only one parameter at a time & keeping others unchanged .The algorithm mainly depends on the prime number whose value depends on the number of bits used to generate this prime. It also depends on the private key, the input message & the cut length of the block message.

Initially, the parameters are set such that the plain text is (How Are You), the cut-length is equal to the whole message length & the private key is chosen as five characters.

2.6.2.Changing the length of the prime number (p):

The value of a prime number p plays an important role in forming the public key (y,g,p) & hence the encrypted message pair say – (y1,y2). This prime number is a function of number of bits used to generate it. Changing this bit number is directly reflected on the length of other parameters, shown in table1.

The length of the prime number p is directly proportional to its generating number of bits since the maximum value of (p) is restricted by 2^bit (table). The length of the encrypted message also increases as a function of prime number. The security is improved by increasing this cipher text. The length of the integer (g) is not directly affected by the number of bits, whereas the length of the public key (y) increases in a manner similar to that of prime number (p).

The result obtained in the table, shows that (p), (y1), &(y) have almost the same values, whereas (y2) behaves in a similar way but with larger values. On the other hand, the result of (g) is almost constant for all the values of modulo (p).

2.6.3.Changing the private key length:

The private key plays an important role in generating the public key & hence the cipher text. To study the performance of this private key, the previous parameters are kept as stated before, while the private key length is changed. The value of (bit), which is used to generate the prime number (p) is fixed to 128, then the secret key is chosen of variable length. This change has shown no clear effect on the other parameters.

2.6.4.Changing the cut length:

For a short message, as one given here , changing the cut-length has no clear effect on the other parameters. However, it may be only the execution time which may be varying as the cut-length changes.

3. Conclusion

The ElGamal algorithm can be use as RSA algorithm for public key encryption because:

- RSA encryption depends on the difficulty of factoring large integers while
- ElGamal encryption depends on on the difficulty of computing dicrete logs in a large prime modulus.

ElGamal is nothing but the advance version of Diffie- Hellmen key exchange protocol. But , ElGamalis not good because its cipher text is two times longer than the plain text. ElGamal is good because it gives different cipher text for same plain text each time.For image data, the size of the cipher text is very huge & reshaping the encrypted data was not understood.
ElGamal 's encryption is very simple because it is multiplication of message and symmetric key(i.e c=m*k)

4 Acknowledgment

Foremost, I would like to express my sincere gratitude to my advisor Dr. D.B. Ojha for the continuous support of my M.Tech study, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my M.Tech study.

Besides my advisor, I would like to thank the rest of my department professors or lecturers: Prof. R.P.Ojha and Santosh Upadhyay and others, for their encouragement, insightful comments, and hard questions.

Last but not the least; I would like to thank my best friend, Shiv Kumar supporting me spiritually throughout my life.

5 References

[1] W. Diffie, M.E. Hellman, "Privacy and Authentication: An Introduction to Cryptography", in Proceedings IEEE, Vol 67 (3) Mar 1979 pp 397-427

[2] W Diffie, M. E. Hellman," New Directions in Cryptography", IEEE Transactions on Information Theory, 22 (1976), 644-654.

[3] R.L Rivest, A Shamir, L Adleman," A Method for obtaining Digital Signatures and Public Key Cryptosystems", Communications of the ACM, 21 (1978), 120-126.

[4] T ElGamal,"A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms", IEEE Transactions on Information Theory, 31 (1985), 469-47

[5] 'JAVA 2 - BLACK BOOK- jdk 6 Edition'.

[6] 'COMPLETE REFFERENCE (J2SE)', Herbert Schildt.

[7] Hand book of cryptography at -- www.carc.math.uwaterloo.ca/hac

[8] java.sun.com/j2se/1.4.2/docs/api/java/.../BigInteger.html

[9] java.sun.com/docs/books/tutorial/networking/index.html

[10] java.sun.com/javase/6/docs/api/overview-summary.html

[11] http://www.ics.uci.edu/goodrich/teach/ics247WO3/notes/elgamal.pdf

[12] http://www.x5.net/faqs/crypto/q29.html

- Rashmi Singh is currently pursuing masters degree program in Computer science and engineering in Mewar University, India, PH-+919694430530. E-mail: rashmikamal011@gmail.com
- Shiv Kumar is currently pursuing masters degree program in Computer science and engineering in Mewar University, India, PH-+919711237004. E-mail: shivkumar004@gmail.com