Asian J. of Adv. Basic Sci.: 2(3), 125-129
ISSN (Online): 2347 - 4114
www.ajabs.org

Implementation & Analysis of RSA and ElGamal Algorithm

Ankush Sharma, Jyoti Attri, Aarti Devi & Pratibha Sharma

Deptt. Of CSE, Career Point University, Hamirpur (H.P.) 176041
Email ID: ankushasp@gmail.com

ABSTRACT: Cryptography is an emerging technology, which is important for network security. Network security is most vital component in information security as it refers to all hardware and software function, characteristics, features, operational procedures, accountability, access control, and administrative and management policy. This paper describes the implementation of Rivest Shamir Adleman (RSA) and ElGamal Algorithm on JCryp Tool 1.0.0. In this paper comparison of these two algorithms has been done on the basis of security and time consumption for encryption and decryption.

Keywords: Cryptography, Asymmetric key, RSA, ElGamal.

INTRODUCTION

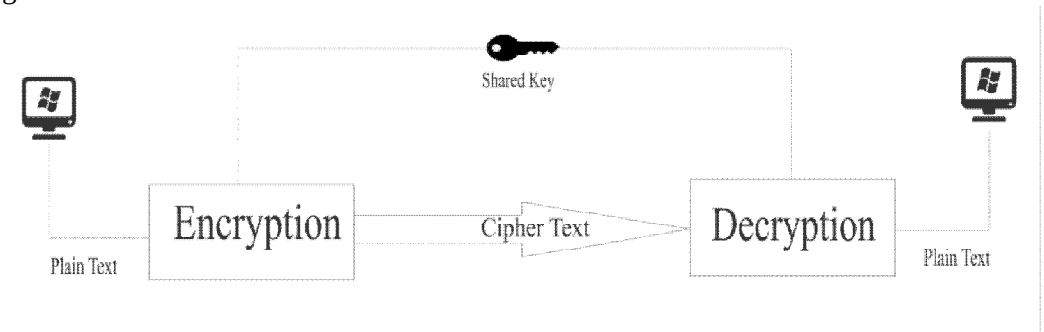
Network Security & Cryptography is a concept to protect network and data transmission over wireless network. Data Security is the main aspect of secure data transmission over unreliable network. Data Security is a challenging issue of data communications today that touches many areas including secure communication channel, strong data encryption technique and trusted third party to maintain the database. [1].In technical terms, the process of encoding plaintext message into cipher text messages is called as Encryption. The reverse process of transforming cipher text message back to plaintext message is called as Decryption[2].

1.1 Categories of Cryptography:

1.2.1. Symmetric key (also called secret-key) cryptography algorithms.

1.2.2. Asymmetric key (also called public-key) cryptography algorithms.

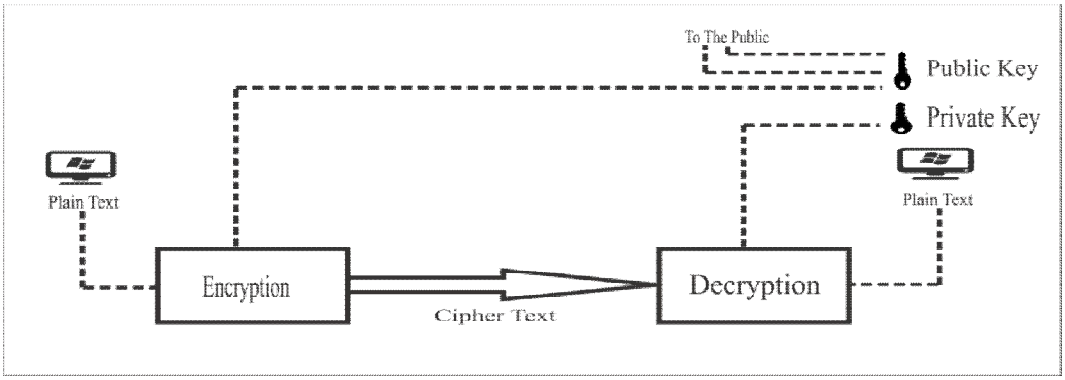
1.2.1. Symmetric Key cryptography: In symmetric-key cryptography, the same key is used by both parties. The sender uses this key and an encryption algorithm to encrypt data; the receiver uses the same key and the corresponding decryption algorithm to decrypt the data as shown in below Figure:



Also, in symmetric-key cryptography, the same key is used by the sender (for encryption) and the receiver (for decryption). The key is shared[3].

1.2.2. Asymmetric Key cryptography: In asymmetric or public-key cryptography, there are two keys: a private key and a public key. The private key is kept by the receiver. The public key is announced to the public. In Figure imagine Alice wants to send a message to Bob. Alice uses the public key to

encrypt the message. When the message is received by Bob, the private key is used to decrypt the message.



In public-key encryption/decryption, the public key that is used for encryption is different from the private key that is used for decryption. The public key is available to the public; the private key is available only to an individual[3].

RSA (Rivest-Shamir-Adleman) Algorithm: Rivest, Adi Shamir and Leonard Adleman are the developer of the RSA cryptosystem of MIT in 1997. It was described in 1978. Some of the famous security system which is composed of three faces: such as prime Key generation, Encryption and Decryption phase. In this technique we used RSA cryptosystem algorithm. In which included the private key and public key. The public key only used for encrypt the messages and it can be seen to all. It is not secret key. The private key is used for decrypt the messages. Private Key is also called the secret key[4].
Seeing from key management, RSA algorithm is more superior algorithm. Because the RSA algorithm can distribute encryption key openly, it is also very easy to update the encryption keys, and for the different communication objects, just keep the decryption keys secret[5].

Mathematical Implementation of RSA Algorithm: The RSA algorithm is based on the mathematical part that is easy to find and multiple two large prime numbers together, but it is extremely difficult to factor their product. There are some important steps are involved in a RSA algorithm to solve a problem as given below:

Step 1: Assume two large prime numbers p & q.

Step 2: Compute:

$$N = p*q$$

Where N is the factor of two large prime number.

Step 3: Select an Encryption key (e) such that it is not a factor of (p-1)*(q-1)

i.e. $\varphi(N) = (p-1)*(q-1)$

for calculating encryption exponents e, should be $1<e<\varphi(N)$ such that gcd $(e, \varphi(N))=1$

The main purpose of calculating gcd is that e&φ(N) should be relative prime. Where $\varphi(N)$ is the Euler Totient Function &e is the Encryption Key.

Step 4: Select the Decryption key (d), which satisfy the Equation d*e mod (p-1)*(q-1) = 1

Step 5: For Encryption:

$$CT \text{ (Cipher Text)} = (PT)^e \bmod N$$

Step 6: For Decryption:

$$PT \text{ (Plain Text)} = (CT)^e \bmod N$$

Example of RSA:

- Select the two different prime no. p & q:

  p = 17 & q = 19.

- Compute N = p*q

N = 17*19=323

- Compute φ(N)=(p-1)*(q-1)

  φ (N) = (17-1)*(19-1)

  φ (N) = 16*18=288

- Select any number 1<e<288 and e should be prime number.

  Let e=19.

- Compute d, **i.e.** e*d=1 mod φ(N)

  d = 91.

  The Public Key is (N = 323,e = 19)

  The Private Key is (N = 323,d = 91)

  Given Plain Text (PT) = Hello 123.

For Encryption:

CT = (Hello 123)^{19}mod 323.

CT = 13f 65 a5 a5 a8 12a 13b 32 33

For Decryption:

PT = (13f 65 a5 a5 a8 12a 13b 32 33)^{91}mod 323

PT = Hello 123.

ElGamal Algorithm: Taher Elgamal first described the ElGamal Cryptosystem in an article published in the proceedings of the CRYPTO '84, a conference on the advances of cryptology.

The proposed algorithm belongs to the family of public key cryptographic algorithms. Therefore it makes use of a key separated into a public and a private part. A fundamental aspect of this system is that the knowledge of the private part makes the decryption easy. If the private key is unknown, it is virtually impossible to decrypt the message in acceptable time.[7]

ElGamal encryption consists of three components: the key generator, the encryption algorithm, and the decryption algorithm.

Key Generation: The basic requirement for a cryptographic system is at least one key for symmetric algorithms and two keys for asymmetric algorithms. With ElGamal, only the receiver needs to create a key in advance and publish it.

Bob will take the following steps to generate his key pair:

1. Prime and group generation: First Bob needs to generate a large prime p and the generator g of a multiplicative group Z^*_p of the integers modulo p.

2. Private Key selection: Now Bob selects an integer b from the group Z by random and with the constraint $1 \leq b \leq p - 2$. This will be the private exponent.

3. Public key assembling: From this we can compute the public key part g^b mod p. The public key of Bob in the ElGamal cryptosystem is the triplet (p, g; g^b) and his private key is b.

4. Public key publishing: The public key now needs to be published using some dedicated key server or other means, so that Alice is able to get hold of it.

For Encryption: To encrypt a message M to Bob, Alice first needs to obtain his public key triplet (p; g; g^b)from a key server or by receiving it from him via unencrypted electronic mail. There is no security issue involved in this transmission, as the only secret part, b, is sent ing^b. Since the core assumption of the ElGamal cryptosystem says that it is infeasible to compute the discrete logarithm, this is safe.

For the encryption of the plaintext message M, Alice has to follow these steps:

1. Obtain the public key: As described above, Alice has to acquire the public key part (p, g, g^b) of Bob froman official and trusted key server.

2. Prepare M for encoding: Write M as set of integers (m1, m2,…..) in the range of {1,….., p – 1}. These integers will be encoded one by one.

3. Select random exponent: In this step, Alice will select a random exponent k that takes the place of the second party's private exponent in the Diffie-Hellman key exchange. The randomness here is a crucial

factor as the possibility to guess the k gives a sensible amount of the information necessary to decrypt the message to the attacker.

4. Compute public key: To transmit the random exponent k to Bob, Alice computes g^k mod p and combines it with the cipher text that shall be sent to Bob.

5. Encrypt the plaintext: In this step, Alice encrypts the message M to the cipher text C. For this, she iterates over the set created in step 2 and calculates for each of the m_i:

$$C_i = m_1 * (g^b)^k$$

The cipher text C is the set of all c_i with $0 < i \leq |M|$.

For Decryption: After receiving the encrypted message C and the randomized public key g^k, Bob has to use the encryption algorithm to be able to read the plaintext M. This algorithm can be divided in a few single steps:

1. Compute shared key: The ElGamal cryptosystem helped Alice to define a shared secret key without Bobs interaction. This shared secret is the combination of Bobs private exponent and the random exponent k chosen by Alice. The shared key is defined by the following equation:

$$(g^k)^{p-1-b} = (g^k)^{-b} = g^{-bk}$$

2. Decryption: For each of the ciphertext parts c_i Bob now computes the plaintext using

$$M_i = (g^k)^{-b} * c_i \bmod p$$

After combining all of the m_i back to M he can read the message sent by Alice.

For Example:

1. **Key Generation:**
   - Choose a large random prime number (p>256). p=263
   - Choose a generator number (g). g=67
   - Choose public key A. A = 11
   - Choose your parameter b between 2 and p-2. This parameter is used for every operation and should be used only once. Otherwise an attack would be possible using only one known plain-/ciphertext combination.

     i.e. b = 257.

     Given Plain Text (PT) = Hello 123.
2. **Encryption:**

   Given Plain Text (PT) = Hello 123.

   Cipher Text = e1e1 d9 cece 33 64 ba de 102
3. **Decryption :**
   - Compute Shared Key:-

     g^{-bk}
   - Decryption:

     $M_i = (g^k)^{-b} * c_i \bmod p$

   Cipher Text = e1e1 d9 cece 33 64 ba de 102

   Plain Text (PT) = Hello 123.

CONCLUSION

The practical implementation of both these algorithms helps reader towards the best understanding and working differences between the two asymmetric key cryptography algorithms. This paper analyze that ElGamal algorithm is more secure as compared to RSA algorithm because it generates more complex cipher text and it was also slow because when we encrypt and decrypt it, it generates more than one public keys.

REFERENCES

1. Kaushik Sumedha & Singhal Ankur "Network *Security Using Cryptographic Techniques*" Volume 2, Issue 12, December 2012, IJARCSSE.
2. kaur Kulwinder "*Performance Evaluation of Ciphers Using CRYPTOOL 2.0"* Volume 3, No. 1, AUG, 2012, IJCT.
3. Data Communication and Networking by "Behrouz A. Forouzan".
4. Urbana Ivy B.Persis, Mandiwa Purshotam. Kumar Mukesh "*A modified RSA cryptosystem based on 'n' prime numbers*" Volume1 Issue 2 Nov 2012, IJECS.
5. Saveetha P. & Arumugam S. "*Study on Improvement in RSA Algorithm and its Implementation*" Volume-3, Issue-6, 7, 8, 2012, IJCCT.
6. Yadav Prasant Singh, Sharma Pankaj, Dr Yadav K. P " *Implementation of RSA Algorithm using Elliptic Curve Algorithm for Security and Performance Enhancement*" Volume 1, Issue 4, May 2012, IJSTR.
7. Meier Andreas V. "*The ElGamal Cryptosystem*" June 8, 2005.
8. AcharyaKritika,SajwanManisha&BhargavaSanjay "Analysis of Cryptographic Algorithms for Network Security" Volume 3– Issue 2, 130 - 135, 2014,IJCATR.
9. http://m.voices.yahoo.com/comparing-symmetric-asymmetric-key-encryption-6329400.html
10. KumarYogesh, MunjalRajiv, Sharma Harsh "Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures" Vol. 11, Issue 03, Oct 2011, IJCSMS.