A Survey and Analysis Performance of Generating Key in Cryptography

Jalpa Bariya[1] Sneha Gaywala[2]
[1,2]Department of Information and Technology
[1,2]SVIT, Gujarat Technological University

Abstract— In Digital world information security is very important, there is must require transmit secure and private information over the network. safety is also serious in broad variety of purpose. So whenever we transfer information between two parties it required cryptography techniques. In this paper we describe some different techniques and Core idea about the public encryption techniques and symmetric techniques that provide the security and how to generate key in different way and provide confidentiality and secure our data. Finally, it concludes all types of techniques and compare it.

Key words: Public key cryptography, Cryptography, symmetric key algorithm, Key generation

I. INTRODUCTION

To make a safe communiqué by which one can broadcast confidential messages secretively to communication with the other entity. Cryptography plays a most important roal in the communication channel.

Cryptography means "Hidden Writing"[1]. Cryptography is the process of exchange information in the form of plaintext into cipher text. It contain encryption and decryption to achieve security.

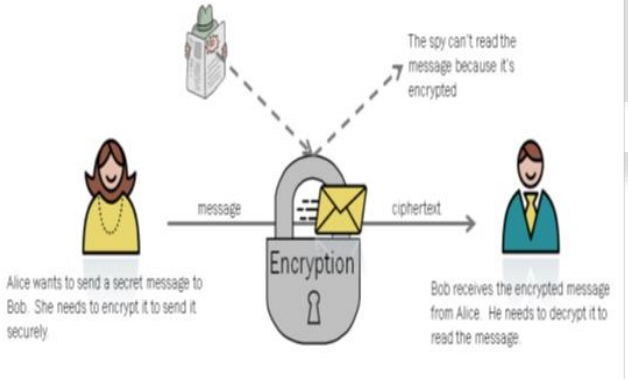


Fig. 1: Concept of Cryptography [2]

II. CLASSIFICATION OF CRYPTOGRAPHY

Cryptography mainly divided into three types symmetric key cryptography, asymmetric key cryptography and hash function. There are various algorithm related to cryptography and the classification is shown in the fig 2 [3].

Symmetric key uses same key for both encryption and decryption. Asymmetric cryptography use one key for encryption and another one is for decryption. Hash function is a computationally efficient function mapping binary string of arbitrary length to binary strings of some fixed length, called hash-value.

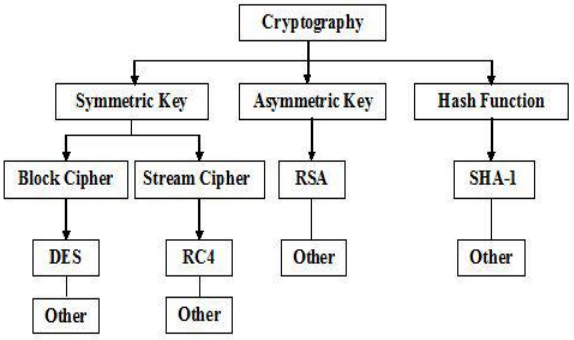


Fig. 2: Classification of cryptography [3]

III. LITERATURE SURVEY

A. Senouci Et Al.Chaotic Cryptography Using External Key[4]

The Paper has the discussion regarding 'disposition' of trusting behavior, an external secret key of variable length maximum 128-bits and used in our algorithm. The encryption of each block of plaintext has been made dependent on the secret key and the cryptosystem is further made robust against any reasonable attack by using the feedback technique.

STEP 1:

The plaintext into blocks of 8-bits, i.e., each symbol of the plaintext/ciphertext corresponding to a single block. Plaintext and ciphertext of n blocks can be represented as:

P=P1P2P3P4….Pn (plaintext)

C= C1C2C3C4…Cn (ciphertext).

Divided into blocks of 8-bits named as session keys.

K=K1K2K3K4…K16 (secret key)

STEP 2:

Xs= real number rang 0 to 1

ASCII value of nth session key

$$Xs\frac{((\mathbf{K1})2 \oplus (\mathbf{K2})2 \oplus (\mathbf{K3})2 \oplus \cdots \oplus (\mathbf{K16})2)\mathbf{10}}{\mathbf{256}} \quad (1)$$

Ns=(K1 + K2 + K3 +….. + K16)mod 256.

We choose a session key (Kr, l r 16) randomly and modify the seed values for the initial condition (Xs)

$$X=(\mathbf{Xs} + \frac{\mathbf{Kr}}{\mathbf{256}}) \text{ MOD } 1 \quad (2)$$

$$N=Ns + K \quad (3)$$

Kr = ASCII value of random session key

STEP 3:

Λ= system parameter

$\lambda i = (a * Yi + c) \bmod m \ /200 + 3.57$

a = multiplier,

m and c = constants,

λi=system parameter value of the

Yi= 0 for the encryption/decryption of the first block of plaintext/ciphertext

i = 1 while for the encryption/decryption of the remaining blocks of plaintext/ciphertext

(i = 2 to n) Yi is calculated using the standard LCG generator as defined below:

$$Yi = (a * Yi-1 + c) \bmod m \quad (4)$$

1) Encryption

Ci = (Pi + Xnew ∗ 256) mod 256

2) Decryption

Pi = (Ci + 256 − Xnew ∗ 256) mod 256

Xnew = the encryption/decryption of the next block of plaintext/cipher text

Ci – 1 = ASCII value of the previously encrypted/ decrypted cipher text block are taken

X = seed values

Ns = number of iterations

put the symbols or we can say block to the ASCII values obtained in pervious (Ci/Pi) as the cipher text/plaintext. Choose next block of plaintext/cipher text and repeat the process from until the plaintext/cipher text is exhausted.

B. Jacobian Elliptic Chebyshev Rational Maps[5]

1) Key Generation Algorithm

Key Generation takes place in three steps: Alice, in order to generate the keys, does the following:

1) Generates a large integer s
2) Selects two random numbers ω ∈ [−1 , 1] and k ∈ [0, 1], and computes Rs(ω, k).
3) Alice sets her public key to (ω, k, Rs(ω, k)) and her private key to s.

2) Encryption Algorithm

Encryption requires five steps: Bob, in order to encrypt a message, does the following:

1) Obtains Alice's authentic public key (ω, k, Rs(ω, k)).
2) Represents the message as a number M ∈ [−1 , 1].
3) Generates a large integer r.
4) Computes Rr(ω, k), Rr•s(ω, k) = Rr(Rs(ω, k), k), and X = M • Rr•s(ω, k).
5) Sends the cipher text C = (Rr(ω, k), X) to Alice.

3) Decryption Algorithm

Decryption requires two steps: 11 Alice, to recover the plaintext M from the cipher text C, does the following:

1) Uses her private key s to compute Rs•r(ω, k) = Rs(Rr(ω, k), k).
2) Recovers M by computing M = X/Rs•r(ω, k). Notice that, the value of k, which defines the form of the map, could be the same for all users of the system.

C. Spanning Tree Key Management (Stkm) [6]

The tree is built for rekeying. Each node A is loaded with three keys are shared with the base station to encrypt/decrypt the messages sent by A and BS separately Kr =shared by all nodes of the network.

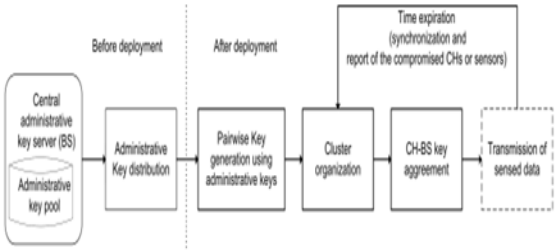
The rekeying process is launched by the base station periodically to refresh Kr.

To construct the spanning tree, a Hello message, initiated by the base station, is broadcasted within the network until all nodes join the tree.

Once a node A suspects a neighboring node (son node or father node) to be malicious, it sends a REFRESH-REQ message to its father on the tree.

This message goes upward until it reaches the BS, which finally broadcasts a REFRESH message within the tree. When a son node of the BS receives the REFRESH message, it replaces the key Kr by the new one, and then encrypts and forwards the REFRESH message of the BS to its sons. This way, the REFRESH message goes downward within the tree till reaching all the sensor nodes.

Finally, every sensor node gets a new global key.



Fig/ 3: Cluster based key renewal [6]

Suppose N wants to join the network, it broadcasts a JOIN message. Upon the reception of the JOIN message, each node A in the transmission range of the new node generates and broadcasts an acknowledge message. The new node sends out a Father message to declare the sender of his first received acknowledge message as his father. Afterwards, Afterwards, the father node adds the newly deployed node in its sons list, and the surrounding nodes that heard this Father message add the new deployed node to their neighbours list in the spanning tree.

In STKM, the storage overhead is acceptable for current nodes, as each node only needs to keep two keys with the base station, one global key and d 0 (the number of sons) keys with its sons. Moreover, STKM has a low complexity communication: each node sends a message and receives d (the number of neighbours) from its neighbours. STKM can resist against node capture attacks.

D. Exclusion Basis System (Ebs) [7]

It is a combinatorial formulation of the group key management problem each node ni is assigned.

Out of a

P= k + m (1<k, m< n)

P = pool of keys

K= keys

n= number of sensor nodes in network.

Rekeying is triggered either periodically or once one or more nodes are captured (or suspected to be captured). In the rekeying process, replacement keys are generated, encrypted with all the m keys unknown to the captured nodes and finally distributed to other nodes that collectively know the m keys.

Scalable, Hierarchical, efficient, Location-aware and Lightweight (SHELL), is based on EBS. In SHELL, nodes are grouped into clusters

Ci, (0 < i ≤ n)

N= number of nodes in a cluster

Rekeying only occurs within these cluster

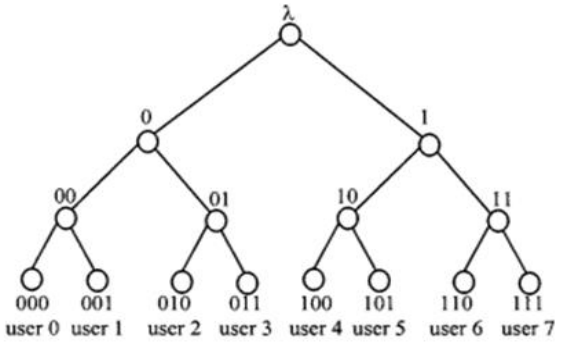


Fig. 4: Binary key [7]

The network considered in SHELL consist of a command node, cluster heads (CHs), gateways and sensor nodes. The command node is assumed to be resource rich and cannot be compromised. In SHELL, after bootstrapping, the gateway decides the number of administrative keys needed for a cluster.

Then, it sends the list of nodes along with the EBS table of key combination to the command node. The command node. The Command node designates a number of administrative keys needed for a cluster.

Then, it sends the list of nodes along with the EBS table of key combinations to the command node. The command node designates a number of gateways for each cluster Ci to generate administrative keys.

After generation, the key generation gateways encrypt and send these keys to the cluster head GCH[i], which then decrypts these messages and broadcasts their contents to the member nodes of its cluster.

E. Robust Cryptosystem Algorithm For Non-Invertible Matrices Based On Hill Cipher [8]

It is depend on the idea that on encryption side each plaintext char convert into two cipher text chars, and in the decryption side each tow cipher text chars convert into one plaintext char, in this way we can use any key matrix (invertible or not), also the key generation that always difficult when key not invertible is solved. In Hill Cipher algorithm to encrypt plaintext block of size n, we need key matrix (Knxn) with entries are between included, but the determinant must be relatively prime to , each entry in the plaintext block is between included each block of plaintext is then an n-dimensional vector .

1) Generating New Key Steps

a) At The Encryption Side

1) Send the key matrix using public key of recipient.
2) Using the key to encryption one block.
3) Using the block of data to swapping with min value of corresponding row and Summation with the max value of corresponding row to produce the new key matrix.
4) Checks if the determinant of new key zero then adds identity matrix for it, else do nothing.
5) Repeat from second to fourth step.

b) At The Decryption Side

1) Receive and decrypt the key matrix using private key of recipient.
2) Using the key to decryption one block.
3) Using the block of data that decrypted to swapping with min value of corresponding row and Summation with the max value of corresponding row to produce the new key matrix.
4) Checks if the determinant of new key zero then adds identity matrix for it, else do nothing.
5) Repeat from second to fourth step

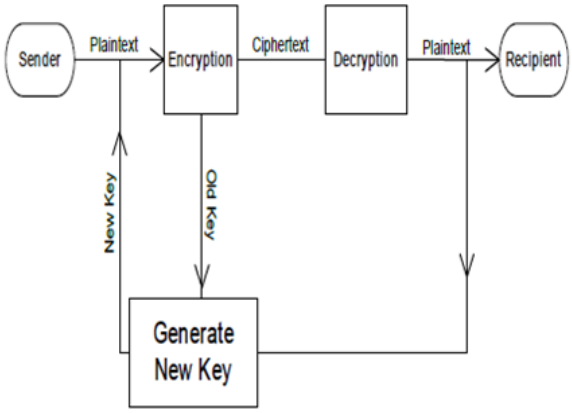


Fig. 5: Diagram [8]

The method of generating a new key is secure enough since the key changes every sent block. So the number of unknowns become more than the number of equation available to the attacker. Therefore, there is no mathematical solution for this system. In addition, the idea of generating a new key in each block has no unique inverse, because it swaps max and min unknown values with the old key values. And so, the attacker has no mathematical model to retrieve the key.

F. Quantum Key Distribution [9]

using the current computing systems classical cryptography is based on the computational difficulty to compute the secret key. Depending only on the difficulty of computational intricacy does not provide enough security because finding a fast method to calculate the secret key. it will compromise the security of the systems. Law of physics is used in Quantum computing for communication. In cryptography and key distribution quantum theorems and principles are applied. In this paper, new model for quantum key distribution are introducing among three parties or more where there is a trusted centre that providing the clients necessary secret information to securely commune with each other.

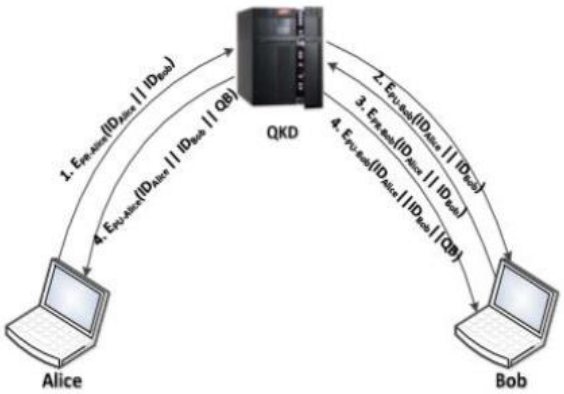


Fig. 6: User Authentication and Quantum Bases distribution [9]

Quantum key distribution protocols BB84, B92 and EPR communicate using a classical channel to compare the bases. This approach facilitates eliminating the erroneous qubits. They introduce a novel security quantum algorithm that employs public key encryption algorithm to generate keys to improve security over quantum communication channel. Moreover, the introduced algorithm enhances user's authentication and data privacy.

G. Bidisha Et Al. Secure Cryptography Using Digital Logic [10]

In this paper the algorithm that provide security to the confidential information by encryption this information twice.

In this algorithm they use binary addition, folding method, Logical XOR operation with encryption key and generation of 2's compliment of number represented in 8-bit binary equivalent value.

The algorithm contains the following steps:

1) Text Encryption/Decryption
2) Key Encryption/Decryption

1) Text Encryption Algorithm

Step:1 Read the plain text.

Step 2. Count the length of each word and store it into an array.

Step 3. Add word length with its corresponding letter's ASCII and space after this word i.e. 1st encrypted text.

Step 4. Add all the word lengths and fold the results until it becomes a single digit.

Step 5. Apply the logical XOR operation on the single digit value and all letters ASCII of the 1st encrypted text i.e. cipher text generated.

Step 6. Send the cipher text along with the 2's complement of no of words and 2's complement of each digit of array of word length as the shared link.

2) Decryption Side Algorithm

Step 1. Read the cipher text and the 2's complement of number of words and 2's complement of each digit of array of word length.

Step 2. Generate the number of words from the 2's complement of it and then generate the actual word length from the 2's complement of each elements of the array of word length.

Step 3. Add all digits of the array of world length and fold the results until it becomes a single digit.

Step 4. Apply the logical XOR operation on the single digit value and all letters ASCII of the cipher text i.e. 1st decrypted text.

Step 5. Repeat until all elements of array of word length has traversed.

Step 6. Plain text generated. Read the word length.

a. If it is not the last word length, then

- Subtract the word length from each character of the 1st decrypted text from 1st array index to (word length + 1) number of array index.
- Make (word length + 2) number index as 1st array index.

b. Otherwise, subtract the word length from each character of the 1st decrypted text from 1st index to word length number of array index.

H. Key Generation (Encryption/Decryption)

In symmetric key cryptography, we need to send the encryption key. So the private key with the cipher text. If we send the unique private key then it is easier for the hacker to generate the original text from this key. So before we send the key through the network we have encrypted the key by generate the 2's complement of each digit of the array of word length from which we generate the encryption key and 2's complement of number of fundamentals here in this array and then two as shared link.

Receiver first generate the 2's complement of second shared link i.e generate the number of element in first shared link. Then using this generates the 2's complement of each element of first shared link that contains the word length. Applying folding method receiver generates the key.

So in this paper including this features of symmetric key cryptography proposed algorithm provide a way to transfer of key securely as a shared link by encrypting the key.

The Algorithm create the cipher text twice to provide more security. Without knowing the proper steps to generating the key from the shared link, so it is almost impossible to decipher the cipher text.

	System	**Pros**	**Cons**	**Suitable for**
3.1	chaotic cryptography using external key	precise discussion of trust.	Development of the formalization for trust is not yet complete changing identities	Situational trust
3.2	Jacobian Elliptic Chebyshev Rational Maps	not need any auxiliary keys except for pre-placed keys. re-keying messages is irrelevant to current group size	For large groups only	Large group network
3.3	Spanning Tree Key Management (STKM)	low complexity communication	Complex No Collusion resistance	Wireless network
3.4	Exclusion Basis System (EBS)	a binary tree logical data structure to store keys	storage overhead at least some users know more than k keys.	Wireless network
3.5	Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher	the problem of key distribution in symmetric encryption. And we avoid the danger of third party Robust	Mathematical complexity linear nature	Peer to peer network
3.6	Quantum Key Distribution	introduce a novel security quantum algorithm that employs public key		Client to server network

		encryption algorithm to generate keys to improve security over quantum communicati on channel.		
3. 7	Secure cryptograph y algorithm using digital logic	There are also key encryption and using folding methods many attacks are possibal.	Complexity is low	Client server network

Table 1. Comparison of Some algorithm

IV. CONCLUSION

This paper has surveyed the literatures on different techniques. The centralized as well as decentralized different aggregation methods for peer to peer network. .Disadvantage of each of the protocol has been pointed out. We have attempted to integrate our understanding across the surveyed literatures any tried to find out the one system proving the privacy and with strong cryptography building blocks.

REFRANCES

[1] Mandal, B., Chandra, S., Alam, Sk.S., Patra, S.S.: A comparative and analytical study on symmetric key cryptography. In: IEEE International Conference on Electronics Communication and Computational Engineering (ICECCE 2014), pp. 131–136
[2] Kandola, Shelley. A Survey of Cryptographic Algorithms. Diss. St. Lawrence University, 2013.
[3] Saranya, K., R. Mohanapriya, and J. Udhayan. "A Review on Symmetric Key Encryption Techniques in Cryptography." International Journal of Science, Engineering and Technology Research 3.3 (2014): 539-544.
[4] Senouci, A., and A. Boukabou. "Chaotic cryptography using external key."Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on. IEEE, 2011.
[5] Bergamo, Pina, et al. "Security of public-key cryptosystems based on Chebyshev polynomials." Circuits and Systems I: Regular Papers, IEEE Transactions on 52.7 (2005): 1382-1393.
[6] He, Xiaobing, Michael Niedermeier, and Hermann De Meer. "Dynamic key management in wireless sensor networks: A survey." Journal of Network and Computer Applications 36.2 (2013): 611-622.
[7] Eltoweissy, Mohamed, et al. "Combinatorial optimization of group key management." Journal of Network and Systems Management 12.1 (2004): 33-50.
[8] Hamamreh, Rushdi A., and Mousa Farajallah. "Design of a robust cryptosystem algorithm for non-invertible matrices based on hill cipher."International Journal of Computer Science and Network Security 9.5 (2009): 11-16.
[9] Ammar Odeh, Khaled Elleithy, Muneer Alshowkan, Eman Abdelfattah, 2013, "Quantum Key Distribution by Using Public Key Algorithm(RSA)", IEEE.
[10] Mandal, Bidisha, Sourabh Chandra, and Sk Safikul Alam. "Secure Cryptographic Algorithm Using Digital Logic." Emerging Research in Computing, Information, Communication and Applications. Springer India, 2016. 501-510.

