A Novel Symmetric Key Cryptography using Multiple Random Secret Keys

A. S. N. Chakravarthy, Ph.D
Dept. of CSE, JNTUK-UCEV
Vizianagaram, Andhra Pradesh
India - 535003

T. Anjikumar
Dept. of CSE, SITAM-GVP
Vizianagaram, Andhra Pradesh
India - 535003

ABSTRACT
Cryptography is an essential practice required for secure communication between two parties. It preserves the confidentiality, integrity, availability, and authenticity of information thereby enhances the security of the data processing system and the information flow within an organization. Symmetric and asymmetric key cryptography are used for encryption and decryption of a message. Symmetric key cryptography is the most commonly used cryptography as it is fast and feasible for use in decrypting bulk messages and requires less computer resources. Chosen-plaintext attacks have been proposed for symmetric key cryptography as it uses single secret key to encrypt different messages. This paper introduces a novel symmetric key cryptography, where multiple random secret keys can be used to encrypt different messages, which is insecure under cipher text-only attacks, a weaker form of attack than chosen-plaintext attacks.

General Terms
Information Security, Cryptographic Algorithm.

Keywords
Symmetric cryptography, Chosen plaintext attacks, Random secret key, Block cipher.

1. INTRODUCTION
Information security is the continuing process of training due care and due diligence to protect information, and information systems, from unauthorized access, use, disclosure, destruction, modification. Information security must protect information throughout the life span of the information, from the initial creation of the information to the final disposal of the information [1]. The information must be protected while in motion and while at rest. Access to protected information must be restricted to people who are authorized to access the information. The computer programs, and the computers that process the information, must also be authorized. This requires some mechanisms to control access to protected information.

Information security uses cryptography to transform usable information into a form that is unusable by anyone other than an authorized user and this process is called as encryption. Information that has been encrypted can be transformed back into its original usable form by an authorized user, who possesses the cryptographic key, through the process of decryption [4]. There are two basic types of cryptography: Symmetric Key cryptography where we use the same key called secret key and Asymmetric Key cryptography where we use two different but mathematically related keys called private key and public key, for encryption and decryption of a message [4]. All encryption algorithms are based on substitution, where each element in the plaintext is mapped into another element and transposition, where elements in the plaintext are rearranged [4].

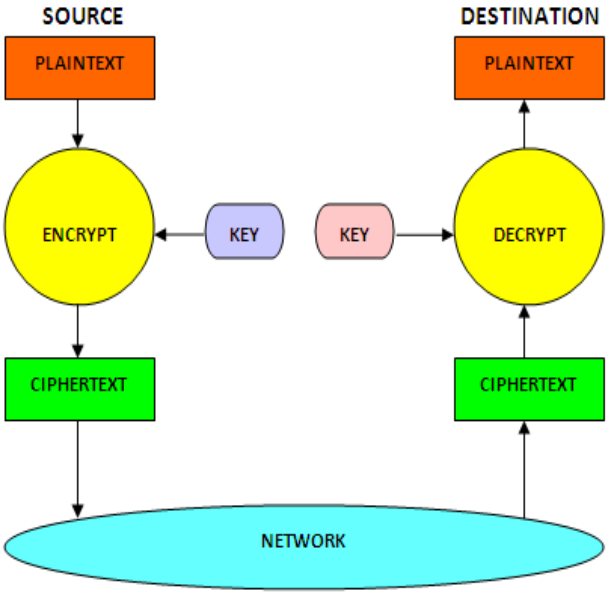

Fig 1: Simplified Model of a Cryptography

Public or asymmetric key cryptography involves the use of key pairs: one private key and one public key. Both are required to encrypt and decrypt a message [4]. The private key, is just that, private. It is not to be shared with anyone. The owner of the key is responsible for securing it in such a manner that it will not be lost [1]. On the other hand, the public key is just that, public. Public key cryptography intends for public keys to be accessible to all users. In fact, this is what makes the system strong. If a person can access anyone public key easily, usually via some form of directory service, then the two parties can communicate securely and with little effort, i.e. without a prior key distribution arrangement. The advantage of asymmetric key cryptography is increased security and convenience: private keys never need to be revealed to anyone and the public key used for encryption does not need to remain secure [4]. Asymmetric Key Encryption Scheme has become secure against adaptive chosen cipher text attack [2]. But asymmetric key encryption is slow and not feasible for use in decrypting bulk messages and requires a lot of computer resources [4].

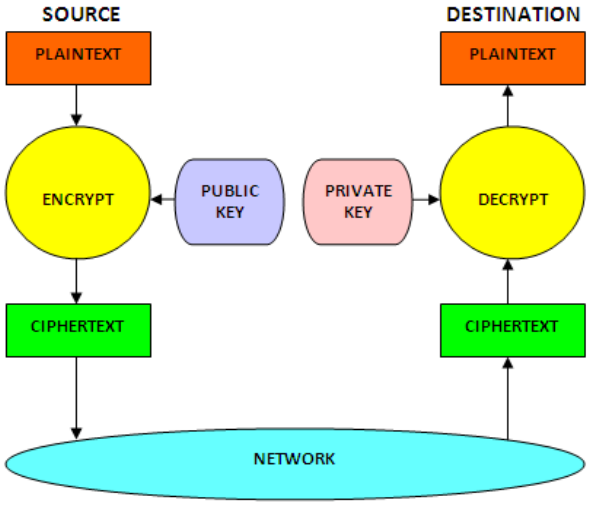

Fig 2: Simplified Model of Asymmetric Key Cryptography

In Secret key cryptography, the sender uses the secret key to encrypt the plaintext and sends the cipher text to the receiver. The receiver applies the same key to decrypt the message and recover the plaintext. Secret key cryptography is also called symmetric key encryption since a single key is used for both functions i.e. encryption and decryption [1]. With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver [1]. The advantage of symmetric key cryptography is it is fast and feasible for use in decrypting bulk messages and requires less computer resources when compared to asymmetric key cryptography but it is vulnerable to chosen plaintext attacks since the same secret key is used to encrypt and decrypt different messages [3].

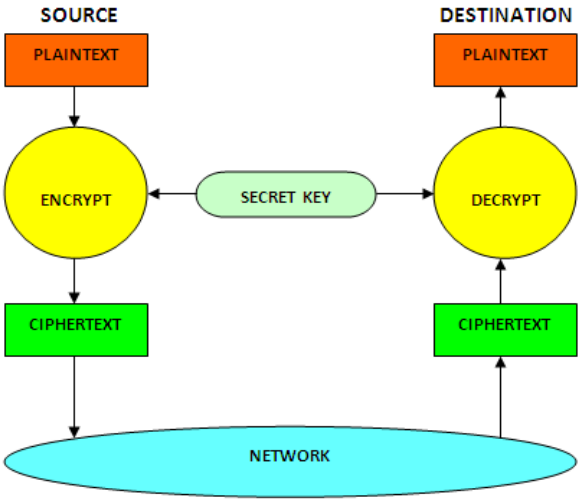

Fig 3: Simplified Model of Symmetric Key Cryptography

This paper introduces a novel symmetric key cryptography, where multiple random secret keys can be used to encrypt different messages, which is insecure under cipher text-only attacks, a weaker form of attack than chosen-plaintext attacks. This paper is organized as follows. Section 2 provides a brief description of the symmetric key cryptography. Section 3 shows the process of key distribution in the existed and the proposed symmetric key cryptography. The process of generating multiple random secret keys is described in section 4. The new symmetric cryptographic algorithm is described in section 5.The results of the experiment are shown in section 6. Finally, the paper is concluded in section 7.

2. SYMMETRIC CRYPTOGRAPHY

Symmetric key cryptography schemes are categorized into either stream ciphers or block ciphers. Stream cipher operates on a single bit at a time, and for each bit a different key is generated. In general, the same plaintext will encrypt to the different ciphertext in a stream cipher [6]. Stream ciphers come in two worth mentioning flavors i.e. Self-synchronizing stream ciphers and Synchronous stream ciphers. Each bit in the keystream is calculated as a function of the previous n bits in the keystream by the Self-synchronizing streamciphers. Since, the decryption process stay synchronized with the encryption process by knowing how far it is into the n-bit keystream, it is termed as "self-synchronizing". The keystream generated by the Synchronous stream ciphers is independent of the message stream but uses the same keystream generation function at sender and receiver. Stream ciphers do not propagate transmission errors and are periodic in nature, so that the keystream will eventually repeat.

P = Plaintext; K = Secretkey; C = Ciphertext; E = Encryption

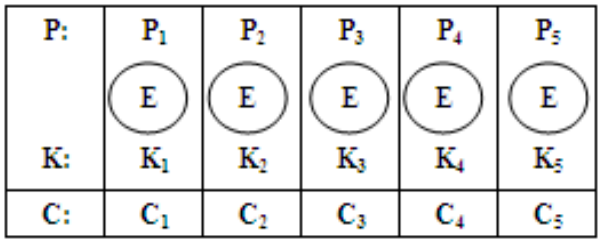

Fig 4: Simplified Model of a Stream Cipher

Block cipher operates on a block of data at a time using the same key on each block. In general, the same plaintext will encrypt to the same ciphertext in a block cipher. Block ciphers can operate in one of the four modes i.e. Electronic Codebook, Cipher Block Chaining, Cipher Feedback mode and Output Feedback [5]. Data Encryption Standard, is the most common secret-key cryptography scheme used today which is designed by IBM in the 1970s and adopted by the National Bureau of Standards in 1977 for commercial and unclassified government applications. DES has been adopted as Federal Information Processing Standard 46 and by the American National Standards Institute as X3.92. DES is a block-cipher that employs a 56-bit key which operates on 64-bit blocks. DES has a complex set of transformations and rules that were designed specifically to give way for slow software implementations and fast hardware implementations. Several variants of DES such as Triple-DES and DESX are currently in use.CAST-128(block cipher), RC2 (block cipher), RC5 (block cipher), Blowfish (block cipher), Twofish (block cipher) are some of the other secret-key cryptography algorithms that are also in use today.

P = Plaintext; K = Secretkey; C = Ciphertext; E = Encryption

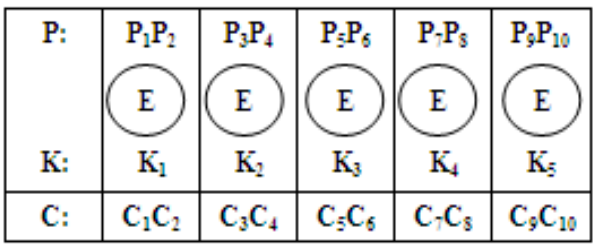

Fig 5: Simplified Model of a Block Cipher

3. KEY DISTRIBUTION

3.1 Existed System

The strength of any cryptographic system rests with the key distribution technique, a term that refers to the means of delivering a key to two parties that wish to exchange data, without allowing others to see the key [8]. One of the most common way of achieving key distribution is that for two parties A and B, if A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B [7].

For this scheme, two kinds of keys are identified:

Session key: A session key is a key used to encrypt/decrypt the user data during communication between two end parties.

Permanent key: A permanent key is a key used between entities for the purpose of distributing session keys.

The configuration consists of the following elements:

Key distribution center: The key distribution center grants permission for two systems to establish a connection and provides a one-time session key for that connection.

Front-end processor: The front-end processor performs end-to-end encryption and obtains session keys on behalf of its host or terminal.

The host transmits a connection-request packet when it wishes to set up a connection to another host. The front-end processor receives that packet, saves it and sends it to the KDC for permission to establish the connection. The communication between the FEP and the KDC is encrypted using a master key shared only by the FEP and the KDC. If the KDC approves the connection request, it generates the session key and delivers it to the two appropriate front-end processors, using a unique permanent key for each front end. The requesting front-end processor can now release the connection request packet, and a connection is set up between the two end systems. The front end processors encrypts all user data exchanged between the two end systems using the one-time session key.

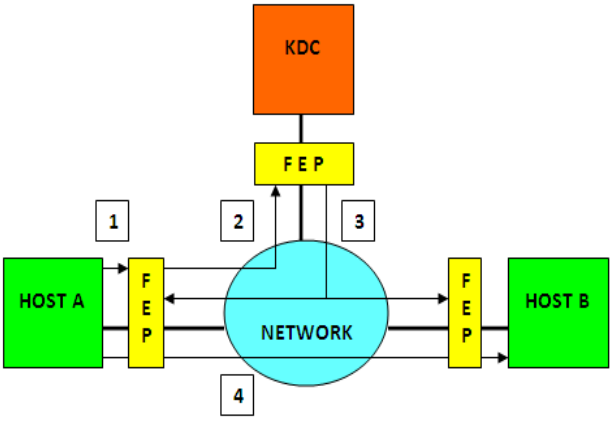

Fig 6: Simplified model of a key distribution in the existed system.

1. Host sends packet requesting connection.
2. Front end buffers packet; asks KDC for session key.
3. KDC distributes session key to both front ends.
4. Encrypted packet transmitted.

3.2 Proposed System

In the proposed system, the steps involved in establishing a connection between the two end parties is same as that of in the existed system but with some amendments in the scheme i.e. the KDC (Key Distribution Center) is substituted with RKDC (Random Key Distribution Center) and the session key is substituted with a message.

When one host wishes to set up a connection to another host, it transmits a connection-request packet. The front-end processor saves that packet and applies to the RKDC for permission to establish the connection. The communication between the FEP and the RKDC is encrypted using a master key shared only by the FEP and the RKDC. If the RKDC approves the connection request, it generates a message that contains confidential information necessary for generating multiple random secret keys and delivers it to the two appropriate front-end processors, using a unique permanent key for each front end. The requesting front-end processor can now release the connection request packet, and a connection is set up between the two end systems. The front end processors at the two end parties processes the message received from the RKDC and generates multiple random secret keys. All user data exchanged between the two end systems are encrypted by their respective front-end processors using the multiple random secret keys.

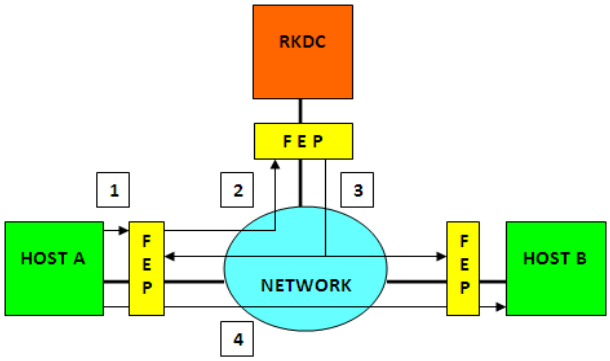

Fig 7: Simplified model of a key distribution in the proposed system.

1. Host sends packet requesting connection.
2. Front end buffers packet; asks RKDC for message.
3. RKDC distributes the message to both front ends.
4. Encrypted packet transmitted.

4. MULTIPLE RANDOM SECRET KEYS GENERATION

Only single key is used to encrypt and decrypt different messages exchanged by the two end systems in symmetric key cryptography. So, there is a possibility for the cryptanalyst to identify the key and encryption algorithm by performing cryptanalysis. Once the secret key and the encryption algorithm is known to the cryptanalyst, then he captures and decrypts all the next information exchanged between the two end systems. So, if encryption and decryption of each message is done with different secret keys, then the above problem can be solved as the cryptanalyst cannot decrypt all messages even though he knows one secret key and encryption algorithm. To do this, multiple random secret keys should be generated and the random secret key used by the two end systems should be same for each message exchange. Here, a method is proposed for generating multiple random secret keys.

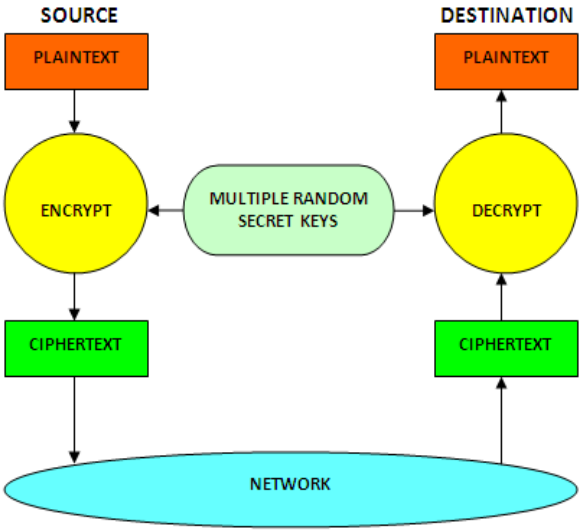

Fig 8: Proposed Model of Symmetric Key Cryptography

When one host wishes to set up a connection to another host, it transmits a connection request packet to the RKDC. If the RKDC approves the connection request, it generates an encrypted message and delivers it to the two hosts. The encrypted message generated by the RKDC will contain four values named as x,y,m,c such that the size of x,y,c will be equal to the size of the secret key that is to be generated and the value of m will vary from 1 to 9. When the two hosts receives the encrypted message, it is decrypted and the four values are passed to random key generation function which computes random secret keys.

The statements in function which computes the random secret key are as follows:

$$x = mx + c$$

$$z = y + x$$

where z is the random secret key computed from the four values m,x,y,c. The two statements are executed repeatedly for each message exchange and the value of x during the last message exchange will be used in the next message exchange. The values of x and z varies for each message exchange and their length is also limited to the length of the secret key that is accepted by the symmetric key algorithm by excluding the least/most significant bits where as the values of m,c,y are constants.

Then a connection is set up between the two end systems and all user data exchanged between them are encrypted by using the random secret keys. A new random secret key will be generated every time when a new message is to be exchanged between the two end systems and the secret key generated on the two end systems will be the same.

The advantage of using different secret keys to encrypt different messages in symmetric key cryptography is the packets will be received in the order in which they are sent and in case if any packet is lost in the network, it will be immediately detected by the destination node and sends the acknowledgement to the source node. Then the source node will send the lost packet again to the destination node thereby, avoiding the data loss on the destination node. It also provides high confidentiality and authenticity.

5. PROPOSED SYMMETRIC KEY CRYPTOGRAPHIC ALGORITHM

The new symmetric key algorithm is simple in nature and works well for large amounts of data. This algorithm is a block cipher that processes the plaintext input in fixed size blocks and produces a block of ciphertext of equal size for each plaintext block. An Extended ASCII character set which contains 256 characters is used in this algorithm during encryption and decryption of the information. To understand the algorithm, the values are represented using decimal numbers instead of binary numbers during encryption and decryption process. The secret key is considered as a string and each character in the string is allocated one byte of memory. The character is converted into number during the encryption and decryption process.

5.1 Encryption Algorithm

Plaintext: SUBSTITUTION; Secretkey: 439255234989

Step 1: Map each character in the plaintext to the corresponding digit in the secret key.

S	U	B	S	T	I	T	U	T	I	O	N
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
4	3	9	2	5	5	2	3	4	9	8	9

Step 2: Generate the ASCII value of each character in the plaintext.

S	U	B	S	T	I	T	U	T	I	O	N
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
83	85	66	83	84	73	84	85	84	73	79	78

Step 3: Add each digit in the secret key to the ASCII value of the mapped character.

4	3	9	2	5	5	2	3	4	9	8	9
+	+	+	+	+	+	+	+	+	+	+	+
83	85	66	83	84	73	84	85	84	73	79	78

Step 4: Perform Modulus operation to the resultant ASCII value with 256.

87	88	75	85	89	78	86	88	88	82	87	87
%	%	%	%	%	%	%	%	%	%	%	%
256	256	256	256	256	256	256	256	256	256	256	256

Step 5: Convert the resultant ASCII value after mod operation to its equivalent character.

87	88	75	85	89	78	86	88	88	82	87	87
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
W	X	K	U	Y	N	V	X	X	R	W	W

Step 6: The text obtained from the resultant characters is the ciphertext.

5.2 Decryption Algorithm

Ciphertext: WXKUYNVXXRWW; Secretkey: 439255234989

Step 1: Map each character in the ciphertext to the corresponding digit in secret key.

W	X	K	U	Y	N	V	X	X	R	W	W
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
4	3	9	2	5	5	2	3	4	9	8	9

Step 2: Generate the ASCII value of each character in the ciphertext.

W	X	K	U	Y	N	V	X	X	R	W	W
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
87	88	75	85	89	78	86	88	88	82	87	87

Step 3: Substract each digit in the secret key from the ASCII value of the mapped character.

87	88	75	85	89	78	86	88	88	82	87	87
-	-	-	-	-	-	-	-	-	-	-	-
4	3	9	2	5	5	2	3	4	9	8	9

Step 4: Perform Modulus operation to the resultant ASCII value with 256.

83	85	66	83	84	73	84	85	84	73	79	78
%	%	%	%	%	%	%	%	%	%	%	%
256	256	256	256	256	256	256	256	256	256	256	256

Step 5: Convert the resultant ASCII value after mod operation to its equivalent character.

83	85	66	83	84	73	84	85	84	73	79	78
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
S	U	B	S	T	I	T	U	T	I	O	N

Step 6: The text obtained from the resultant characters is the plaintext.

6. EXPERIMENTAL RESULTS

This experimental work is implemented in java and carried out on the windows operating system.

The following Figure 9 shows the encryption process at the source end i.e. converting the plaintext to ciphertext by using the first secret key and the new symmetric key algorithm.

Fig 9: Screenshot of encryption process done at the source end by using the first random secret key.

The following Figure 10 shows the decryption process at the destination end i.e. converting the ciphertext to plaintext by using the first secret key and new symmetric key algorithm.

Fig 10: Screenshot of decryption process done at the destination end by using the first random secret key.

The following Figure 11 again shows the encryption process at the source end by using the second secret key and the new symmetric key algorithm.

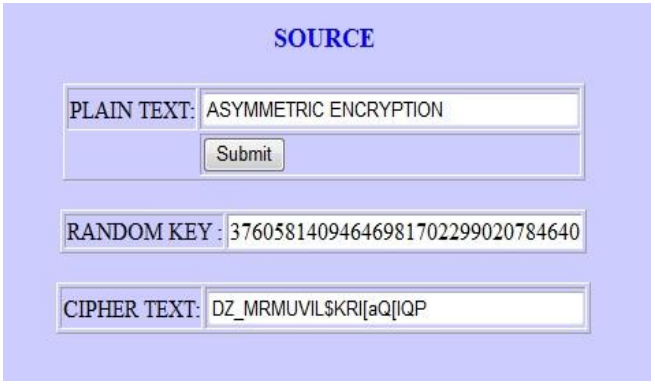

Fig 11: Screenshot of encryption process done at the source end by using the second random secret key.

The following Figure 12 again shows the decryption process at the destination end by using the second secret key and the new symmetric key algorithm.

Fig 12: Screenshot of decryption process done at the destination end by using the second random secret key.

7. CONCLUSION

Symmetric key cryptography is fast and feasible for use in decrypting bulk messages and requires less computer resources but it is vulnerable to chosen-plaintext attacks since the single secret key is used to encrypt different messages. This paper proposed a novel symmetric key cryptography, where multiple random secret keys are used to encrypt different messages, which is insecure under cipher text-only attacks, a weaker form of attack than chosen-plaintext attacks. This process is simple in nature and works well for large amount of data. The RKDC (Random Key Distribution Center) will generate message that is used to generate fixed length secret keys. There is a scope for RKDC to generate message that is used to generate variable length secret keys. The new symmetric key algorithm operates by considering every time the ASCII values of the characters. A new character set can be created by allotting new value instead of ASCII value to each character. The new character set can be used instead of ASCII character set in the new symmetric key algorithm during encryption and decryption process.

8. REFERENCES

[1] The Mohammad Zakir Hossain Sarker and Md. Shafiul Parvez, "A Cost Effective Symmetric Key Cryptographic Algorithm for Small Amount of Data", 9th International Multitopic Conference, IEEE INMIC 2005, 24-25 Dec.2005, pp. 1-6.

[2] R. Cramer, V. Shoup, "Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack" SIAM Journal on Computing Vol 33 2001.

[3] H. A. Bergen and J. M. Hogan, "A chosen plaintext attack on an adaptive arithmetic coding compression algorithm," *Comput. Security*, vol.12, pp. 157–167, 1993.

[4] J. Katz and Y. Lindell, Introduction to Modern Cryptography. London, U.K.: Chapman & Hall/CRC, 2008.

[5] S. Goldwasser and S. Micali, "Probabilistic Encryption", *Journal of Computer and System Sciences*, Vol 28, 1994, pp 270-299.

[6] S. William, Cryptography and Network Security: Principles and Practice, 2nd edition, Prentice-Hall, Inc.,1999, pp 23-50.

[7] Network Security Essentials (Applications and Standards) by William Stallings Pearson Education.

[8] Introduction to Cryptography, Buchmann, Springer.

[9] M.I. Jabiullah, S.M. Mizanur Rahman, M. Lutfar Rahman and M. Alamgir Hossain, "Secure Pseudoorandom Bit Generation for Cryptographic Application", Proceedings of International Conference on Computer and Information Technology *(ICCIT)*, Dhaka, 2001 pp 275-277.

[10] Diffie W, Hellman ME (1976) New directions in cryptography. IEEE Trans Inform Theory 22(6):644–654

[11] R. Merkle, "Secure communication over an insecure channel,"submitted to Communications of the ACM.

[12] Feistel, H. Cryptography and computer privacy. *Sci. Amer. 228, 5* (May 1973), 15-23.