Survey of Cryptographic Algorithms and Applications for Network Security

N.Venkatesan[1] Dr. M. Prabakaran[2]
[1]Research Scholar[2]Research Supervisor
[1,2]Department of Computer Science
[1,2]Bharathidasan University,Government Arts College, Ariyalur, Tamil Nadu, India

Abstract—*Cryptographic algorithms are used to provide security for various applications. The basic services of these algorithms include confidentiality, authentication, integrity, non-repudiation etc. This paper presents a detailed survey of the cryptographic algorithms which are mainly classified as encryption, digital signature and secret sharing. Under each category, a survey on the existing works on the cryptographic algorithms has been done. It also list outs the challenges involved in each technique and presents a comparison table of all the techniques.*

Keywords—Encryption, Digital Signature, Secret Sharing, Symmetric Key, Public Key, AES, Elliptic Curve Cryptography

I. INTRODUCTION

Nowadays, a lot of Internet applications such as on-line shopping, stock trading, internet banking, and so forth are emerged that demand end-to-end secure connections, should be confidential, to ensure data authentication, accountability and confidentiality, integrity, and availability. Security in networking depends on cryptography (means secret writing). The cryptography is defined as the science and art of transforming messages to make them secure and immune to attack.

A cryptographic algorithm (also known as cipher) is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key to encrypt the plaintext. A single plaintext is encrypted to different ciphertext with different keys [1].

The following are four basic services provided by cryptography:

1) Confidentiality verifies whether data remain secret and protects the transmitted data so that the intruder cannot read the encrypted message from the cipher text. While providing confidentiality, an intruder should not know the source and destination, frequency, and length of the data flow.
2) Authentication verifies whether the communicating entity is authentic and finds the origin of the recipient of a message.
3) Integrity ensures that the data received is the same as that of the one sent by the authorized entity. The recipient should confirm that the message has not been modified in the transmission. An intruder will not substitute the fake message for the actual one.
4) Nonrepudiation ensures that the receiver can prove that the message was sent by the specified party. Likewise, the sender can prove that the message was received by the specified party [2].

Cryptography is used in web browser, web server, email client, email server, and so forth.

II. CLASSIFICATION OF CRYPTOGRAPHIC ALGORITHMS

Cryptography involves encryption, digital signature, and secret sharing. Figure 1 shows the classification of cryptographic algorithms.

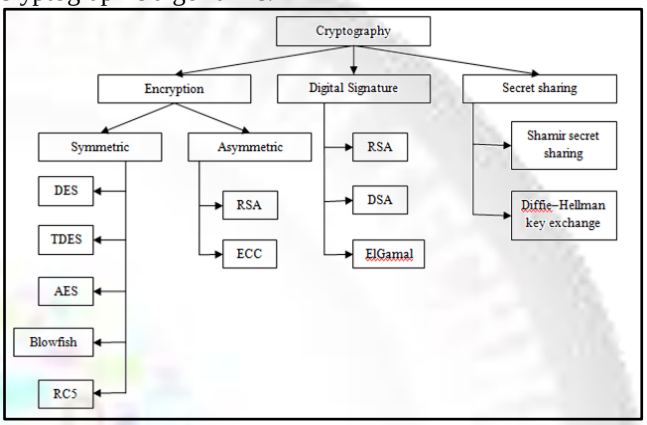

Fig. 1: Classification of cryptographic algorithms

A. Encryption

Encryption is the process of converting normal text to encrypted text whereas decryption is the process of converting encrypted text to normal text. When a sender sends a Hello message to a recipient, the plaintext is converted to ciphertext by using a key (encryption). The ciphertext is transmitted over the transmission medium. On receiving this message, recipient converts the ciphertext back to the plaintext using the same algorithm and key that was used to encrypt the message.

A key is a numeric or alphanumeric text or may be a special symbol. The strength of the encryption algorithm depends on the secrecy and length of the key, the initialization vector, and how they all work together. Avalanche effect is defined as the property of any encryption algorithm where a small change in the key or the plaintext produces a significant change in the cipher text [3].

Encryption algorithms are classified into two groups, namely, symmetric key and asymmetric key encryption.

- In symmetric key encryption (also known as conventional encryption), encryption and decryption are performed using the same key. AES, Blowfish, DES, T-DES, and RC5 are some of the symmetric key encryption algorithms.
- In asymmetric encryption (also known as public-key encryption), encryption and decryption are performed using the different keys: a public key and a private key [4]. The advantage of asymmetric encryption is that it allows people who have no pre-existing security arrangement to exchange messages securely. There is no need for sharing the secret keys among sender and receiver via some secure channel. All communications

involve only public keys, and no private key is ever transmitted or shared. RSA and ECC are some of the asymmetric key encryption algorithms.

B. Digital Signatures

Public key cryptography provides a method for employing digital signatures. Digital signature is used to verify the authenticity of the information's origin and to verify whether the information is complete. Therefore, public key digital signatures offer authentication, data integrity, and non-repudiation, which prevents the sender from claiming that he or she did not actually send the information.

A digital signature serves the same purpose as a handwritten signature. A digital signature is secured than the handwritten signature as it attests to the contents of the information and to the identity of the signer. This technique deals in software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.

A digital signature is said to be valid if it satisfy the following properties: it must verify the author and the date and time of the signature; it must to authenticate the contents at the time of the signature; and it must be verifiable by third parties, to resolve disputes

Signature schemes can be deterministic or randomized. In deterministic signature scheme, computing a signature on a message will always give the same result, whereas randomized signature scheme will give a different result each time you compute it. Signature schemes may or may not support message recovery. If they support message recovery, given the signature, anyone can recover the message on which it was generated; if they do not support message recovery, then the verifier needs to know or guess the message before he can perform the verification. RSA, DSA, and ElGamal are some of the digital signatures algorithms.

By using digital signature, business has not to wait for paper documents to be sent by any postal services. Transmission over a network is cheaper by Digital Signature is much cheaper than others. Using digital signatures alters the risks of documents being decoded, read, removed, or altered while in transmission. With the help of time stamping the digital signatures, you will get the messages in correct time when the documents are signed [5].

C. Secret Sharing

Secret sharing is defined as the methods for distributing a secret among a group of participants, each of whom is allocated a share of the secret. When a sufficient number of shares are combined together and individual shares are of no use on their own, the secret can be reconstructed only. In secret sharing scheme, there are one dealer and n players where the dealer provides a share of the secret to the players only when particular conditions are fulfilled. The dealer accomplishes this by giving each player a share such that any group of t (for threshold) or more players can reconstruct the secret together but no group of fewer than t players can. Such a system is called a (t, n)-threshold scheme [6].

Secret sharing schemes are used for storing information that is highly sensitive and highly important. Some of the examples of secret sharing schemes are encryption keys, missile launch codes, and numbered bank accounts. All these applications needs the information to be kept highly confidential since their exposure can be disastrous. Also, they should not be lost.

For simultaneously achieving high levels of confidentiality and reliability, conventional methods for encryption are not suitable. This is due to one must choose between keeping a single copy of the key in one location for maximum secrecy, or keeping multiple copies of the key in different locations for greater reliability while storing the encryption key. When the reliability of the key is increased by storing multiple copies, confidentiality is reduced due to the formation of additional attack vectors. There are more opportunities for a copy to fall into the wrong hands. Secret sharing schemes address this problem in conventional secret sharing schemes and allow arbitrarily high levels of confidentiality and reliability to be achieved.

Secret sharing schemes are significant in cloud computing domain. By using a threshold secret sharing mechanism, a key can be distributed over many servers. The key is then reconstructed if needed. Secret sharing is suitable for sensor networks where the links are liable to be tapped by sending the data in shares which makes the task of the eavesdropper harder. In such networks, the security can be made greater by continuous changing of the way the shares are constructed. Shamir secret sharing and Diffie–Hellman key exchange are some of the secret sharing algorithms.

III. Analysis of Cryptographic Algorithms

This section presents the detailed description of each algorithm.

A. Encryption Algorithms

This section provides the description of encryption algorithms.

1) Symmetric Key Encryption Algorithms:

The symmetric key encryption algorithms are illustrated below:

a) DES

The Data Encryption Standard (DES) was designed to protect sensitive non-classified information used by the US Government and by enterprises. DES is also known as the Data Encryption Algorithm (DEA). DES uses 64-bit keys in which only 56 bits are actually used for the encryption/decryption of data and the remaining bits are used for key integrity checks.

DES's round function operates on 32-bit half-blocks and consists of the following four operations. First, the block is expanded from 32 bits to 48. Next, 48 bits of round key are mixed using exclusive-or. The result is then passed through a row of eight S-boxes, each of which takes a 6-bit input and provides a 4-bit output. Finally, the bits of the output are permuted according to a fixed pattern [7]. The round keys are derived from the user-supplied key by using each user keybit in about 14 different rounds according to a slightly irregular pattern.

In the past, it was widely used in banking, government, embedded applications, and automatic teller machine networks. However, it is considered to be insecure for most applications nowadays.

b) TDES

Triple DES algorithm (TDES) is a modern derivative of the original DES algorithm where three DES keys are used consecutively in encrypt, decrypt and encrypt mode. Triple DES works with either 112-bit keys or 168-bit keys. Because of certain properties of the TDES, the effective key-strength of a 168-bit key is 112 bits and the effective key strength of a 112 bit key is 80 bits. Hence, TDES with 112 bits should no longer be used; however, TDES with 168 bits is considered to be secure [8]. Both DES and TDES are block ciphers. As these algorithms are very efficiently implemented in hardware, it is commonly used in embedded systems.

c) Blowfish

Blowfish is a symmetric block cipher that can be effectively used for encryption and safeguarding the data. It uses a variable-length key, from 32 bits to 448 bits, making it suitable for securing data. Blowfish was designed as a fast, free alternative to existing encryption algorithms. It is unpatented and license-free, and is available free for all uses. Though it suffers from weak keys problem, no attack is known to be successful against it. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found.

Blowfish was intended to be a completely free–unpatented, unlicensed, and uncopyrighted–alternative to DES. Later, it is used in some systems, both public and private. Blowfish is easy to scale up to a 128-bit block and down to smaller block sizes. Since Blowfish is intended for huge microprocessors with a lot of memory, it is suitable for recent applications. Blowfish can achieve an efficiency of data encryption up to 4 bits per clock. It can also be used in applications where there is a strong communication link and where the key will not be changed too frequently. It is fast as it encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte. It is compact because it can run in less than 5K of memory. It simply uses addition, XOR, lookup table with 32-bit operands [9].

d) AES

The Advanced Encryption Standard (AES) is a symmetric block cipher that uses the same key for both encryption and decryption. However, AES is quite different from DES in a number of ways. AES standard can only accept a block size of 128 bits and a choice of three keys: 128, 192, and 256 bits. A number of AES parameters depend on the key length.

The input is a single 128 bit block both for decryption and encryption and is known as the in matrix. This block is copied into a state array which is modified at each stage of the algorithm and then copied to an output matrix. Both the plaintext and key are depicted as a 128 bit square matrix of bytes. This key is then expanded into an array of key schedule words (the w matrix). It must be noted that the ordering of bytes within the in matrix is by column. The same applies to the w matrix.

e) RC5

RC5 is a fast block cipher designed to be suitable for both software and hardware implementation. It is a parameterized algorithm, with a variable number of rounds, a variable block size, and a variable-length secret key. One significant feature of the design of RC5 is its simplicity. Encryption is based on only three operations, namely, addition, exclusive-or, and rotation. Thus, it makes RC5 both easy to implement and more amenable to analysis than many other block ciphers. The connection between simplicity of design and simplicity of analysis was indeed one of Rivest's goals. Another distinguished feature of RC5 is the heavy use of data-dependent rotations in encryption [10].

RC5 may be subject to timing attacks if RC5 is implemented on platforms for which the time for computing a single rotation is proportional to the rotation amount. However, RC5 can easily be implemented in such a way as to be invulnerable to timing attacks. RC5 has the attractive feature that the length of the key can be varied (unlike with DES for instance) and so the level of security against these attacks can be tuned to suit the application.

2) Asymmetric Key Encryption Algorithms

The asymmetric key encryption algorithms are illustrated below:

a) RSA

RSA was developed and named after by Ron Rivest, Adi Shamir and Len Adleman. Since this time the RSA algorithm has reigned supreme as the most widely accepted and implemented general-purpose approach to public key encryption. The RSA scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and n − 1 for some n with the help of an expression with exponentials.

Plaintext is encrypted in blocks with a binary value less than n. The private key consists of {d, n} and public key is {e, n}. For example, user A has published his public key and then user B wishes to send the message M to A. B calculates ciphertext $C = M^e \pmod n$ and transmits C. On receipt of the ciphertext C, user A decrypts the message $M = C^d \pmod n$.

RSA obtains its security from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large prime numbers. Recovering the plaintext from one key and the ciphertext is equivalent to factoring the product of two primes [11]. RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n. The security of RSA depends on the strengths of two separate functions, namely, encryption function and key generation.

b) Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is a term used to describe a suite of cryptographic tools and protocols whose security is based on special versions of the discrete logarithm problem. It does not use numbers modulo p. ECC is based on a special class of mathematical structures known as elliptic curves. An elliptic curve is characterised by a relatively simple mathematical equation of a particular form, producing curves. The key feature of the mathematics behind ECC is that repeated multiplication can be computed relatively easily while the reverse operations are computationally difficult. When compared to RSA is that the number of key bits can remain relatively low.

The discrete logarithm problem is much harder when applied to points on an elliptic curve. This prompts switching from numbers modulo p to points on an elliptic curve. An equivalent security level can be obtained with shorter keys if elliptic curve-based variants are used. The shorter keys result in ease of key management and efficient computation, making elliptic-curve-based variants of encryption scheme highly attractive for application where computing resources are constrained. The lack of a sub-exponential attack on ECC offers potential reductions in processing power and memory size. These advantages are especially important in applications on constrained devices [12].

B. Digital Signature Algorithms

This section provides the description of digital signature algorithms.

1) ElGamal:

The ElGamal digital signature scheme (also known as Elliptic Curve Variant) is a digital signature scheme that depends on the difficulty of computing discrete logarithms. This scheme allows that a verifier can confirm the authenticity of a message sent by the signer sent to him over an insecure channel [13]. It derives the strength from the assumption that the discrete logarithms cannot be found in practical time frame for a given number, while the inverse operation of the power can be computed efficiently.

Each user of ElGamal cryptosystem generates the key pair by the following steps. Generally a prime number p of 1024 to 2048 bits length is chosen. After that, a generator element g is chosen that ranges between 1 and p-1. It is a generator of the multiplicative group of integers modulo p. The private key x is chosen which is $1 < x < p-1$. The value y is calculated to find ElGamal public key.

In terms of processing speed, Elgamal is quite slow. It is used mainly for key authentication protocols. Due to higher processing efficiency, Elliptic Curve variants of ElGamal are becoming increasingly popular.

2) DSA

Digital Signature Algorithm (DSA) is an efficient variant of ElGamal digital signature with bit length 320. Signature verification in DSA requires only two modular exponentiations with exponents of bit length 160. DSA was advanced by the NSA to be used by the United States government as a standard for virtual signatures. This signature is based on the ElGamal Signature Algorithm. This algorithm performs only the operation of signing the message itself whereas RSA covers signing in encryption and encryption of the message contained.

As DSA manages virtual signatures, it is preferred in faster key technology. This is because the fact that DSA produces the keys right away. During decryption, DSA is faster particularly because of the single function. Digital signature works fine with DSA while verification of the virtual signature is quicker while RSA is hired.

C. Secret Sharing Algorithms

This section provides the description of secret sharing algorithms.

1) Shamir secret sharing:

Shamir's secret sharing scheme is a threshold scheme based on polynomial interpolation that allows a dealer D to distribute a secret value s to n players, such that at least $t < n$ players are necessary to reconstruct the secret. The protocol is information theoretically secure, that is, any fewer than t players cannot gain any information about the secret by themselves.

To share the secret among players, such that t players are required to reconstruct the secret. Dealer creates a random polynomial f(x) of degree t-1. This polynomial is constructed over a finite field, such that the coefficient a_0 is the secret s and all other coefficients are random elements in the field; the field is known to all participants. Dealer publicly chooses n random distinct evaluation points and secretly distributes the share to each player. In order to reconstruct the secret from each subset of t shares out of n shares, Lagrange interpolation is used.

This algorithm provides information theoretic security. Given any t shares, the polynomial is uniquely determined and hence the secret a_0 can be computed. However, given t-1 or fewer shares, the secret can be any element in the field and those shares do not supply any further information regarding the secret. Here, each share is exactly the same size as the secret.

Additional shares may easily be created by calculating the polynomial in additional points. Different weights can be assigned by the number of shares to different authorities. Shamir's secret sharing scheme has the homomorphism property. It has an efficient distributed mechanism for arithmetic calculations [14].

2) Diffie–Hellman Key Exchange:

Diffie–Hellman key exchange (D–H) is one of the earliest practical examples of key exchange implemented within the field of cryptography. The D–H method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Diffie–Hellman key agreement itself is an anonymous (non-authenticated) key-agreement protocol as it provides the basis for a variety of authenticated protocols. Also it is used to provide perfect forward secrecy in Transport Layer Security's ephemeral modes. The method was followed afterwards by RSA, an implementation of public key cryptography using asymmetric algorithms.

D–H method establishes a shared secret that can be used for secret communications while exchanging data over a public network. The crucial part of the process is that sender and recipient exchange their secret numbers. Finally this generates an identical key that is computationally difficult to reverse for another party that might have been listening in on them. The sender and recipient now use this common secret to encrypt and decrypt their sent and received data. It may even be public [15].

IV. CURRENT CHALLENGES OF CRYPTOGRAPHIC ALGORITHMS

A. Challenges in Encryption Algorithms

The issues in encryption algorithms are as follows:

– DES was proved insecure for large corporations or governments; however for backward compatibility, and

cost of upgrading, DES should still be preferred, outweighing the risk of exposure.
- Asymmetric encryption techniques are about 1000 times slower than symmetric encryption techniques which make it impractical while encrypting large amounts of data.
- In order to obtain the same security strength as symmetric, asymmetric must use a stronger key than symmetric encryption technique.
- DES algorithm consumes least encryption time, and AES algorithm has least memory usage while encryption time difference is very minor in AES and DES algorithm.
- There is no significant difference when the results are displayed either in hexadecimal base encoding or in base 64 encoding.
- In case of changing data type such as image instead of text, it was found that Blowfish has disadvantage over other algorithms in terms of time consumption.
- In case of changing key size, in AES algorithms, higher key size leads to clear change in the battery and time consumption.
- Avalanche effect is very high for AES as compared to DES whereas memory requirement and simulation time for DES is greater than that of AES, which shows AES is better than DES.
- AES is ideal for encrypting messages sent between objects via chat-channels, and is useful for objects that involve monetary transactions.
- Blowfish has a better performance than other common encryption algorithms used. Since Blowfish has not any known security weak points so far, this makes it an excellent candidate to be considered as a standard encryption algorithm.
- AES showed poor performance results compared to other algorithms since it requires more processing power.
- Computation costs are relatively low when compared to other signature and key exchange options. These features make ECC ideal to be implemented in small devices such as smart cards and mobile gadgets.
- ECC is complicated and tricky to implement securely, particularly for the standard curves. In ECC, signing with a broken random number generator compromises the key. There occur some patent problems, especially for binary curves.
- RSA is very slow in key generation, signing and decryption, which are slightly tricky to implement security.

B. Challenges in Digital Signature Algorithms

The issues in digital signature algorithms are as follows:
- Digital signatures have a limited time and come with its expiry.
- Both sender and receiver must have to buy authorized certificates for the effective use of digital signature.
- Sender and receiver both have to buy authorized software too, to make transmission smoother and easier.
- In cases where demands regarding computer-generated and technology-based issues are weak or even non-existent, the exchange in such jurisdictions becomes very risky for those who use digitally signed electronic documents.
- The generation of an ElGamal key pair is comparatively simpler than the equivalent process for RSA.
- RSA has the issue in factoring the numbers.
- When faster encryption is needed, RSA is favoured as it encrypts each message and signature for signing in.
- The generation process and verification process of digital signature needs substantial quantity of time; hence, the speed of communication will decrease.
- If a user changes his private key after every fixed break of period, then the record of all these changes must be reserved.
- Loading of all the preceding keys is another overhead that is cause when there is need for a previously sent message.
- RSA consumes longest encryption time and memory usage is also very high, but output byte is least in RSA algorithm.
- The main attraction of ECC over RSA and DSA is that the best known algorithm for solving the underlying hard mathematical problem in ECC by using full exponential time. RSA and DSA take sub-exponential time. This shows that significantly smaller parameters can be used in ECC than in other systems such as RSA and DSA, but with equivalent levels of security.
- In ElGamal digital signature scheme, a third party can forge signatures either by finding the signer's secret key or by finding collisions in the hash functions. Both problems are difficult to handle.
- The signer should carefully choose a different random number uniformly for each signature. That random number and even its partial information should not be leaked. If the information leaked, an attacker can deduce the secret key with less difficulty. The attacker can easily find the key when two messages are sent with same random number.
- For the same level of security as provided by RSA, very short keys are required in ElGamal scheme.

C. Challenges in Secret Sharing Algorithms

- A major problem with secret-sharing schemes is that the shares' size in the best known secret-sharing schemes realizing general access structures is exponential in the number of parties in the access structure. Thus, the known constructions for general access structures are impractical.

Table 1 shows the comparison on features, advantages, disadvantages and applications of cryptographic algorithms

Algorithm	Type	Features	Advantages	Disadvantages	Applications
DES	Symmetric Key Encryption	Uses 64-bit keys	Consumes least encryption time	Insecure for most applications nowadays	Banking, government, and automatic teller machine networks

TDES	Symmetric Key Encryption	Uses 112-bit keys or 168-bit keys	Can be efficiently implemented in hardware	Memory requirement and simulation time is more	Embedded systems
Blowfish	Symmetric Key Encryption	Uses a variable-length key from 32 bits to 448 bits	Invulnerable against differential related-key attacks	Suffers from weak keys problem	Used in Security Sockets Layer (SSL)
AES	Symmetric Key Encryption	Uses 128, 192 and 256 bit keys	Less memory usage	Avalanche effect is very high. Needs more processing power	VoIP, PDAs, cell phones
RC5	Symmetric Key Encryption	Variable key length	Due to variable key size, the level of security can be tuned to suit the application.	Susceptible to a differential attack	Used in image cryptosystem
RSA	Asymmetric Key Encryption and Digital Signature	The security depends on the strengths of encryption function and key generation	Produces less output bytes and provides slow signing	Consumes longest encryption time and memory usage is more	Used in electronic commerce protocols
ECC	Asymmetric Key Encryption		Computation costs are relatively low	Complicated and tricky to implement securely, particularly true for the standard curves	Small devices such as smart cards and mobile gadgets
ElGamal	Digital Signature	1024 to 2048 bits length	High processing efficiency	Processing speed is quite slow	Mainly used for key authentication protocols
DSA	Digital Signature	320 bit length	Faster in signing	Slower when validating the signature	Applications where faster key technology is needed
Shamir secret sharing	Secret Sharing	Provides information theoretic security	Extensible, dynamic and flexible	The verification of correctness of the retrieved shares during the reconstruction process is difficult	Used in applications where a group of mutually suspicious individuals with conflicting interests must cooperate.
Diffie–Hellman key exchange	Secret Sharing	Two parties with no prior knowledge establish shared secret keys over insecure channel	Uses lightweight two-pass protocol with only a public key	Susceptible to man-in-the-middle attack	Applied to many security protocols such as SSL, secure shell (SSH), and IP Sec.

Table 1: Comparison of cryptographic algorithms

V. CONCLUSION

This paper presents a detailed survey of the various types of cryptographic algorithms such as encryption, digital signature, and secret sharing. To provide the security to the network and data cryptographic algorithms are used. In this paper, a survey on the existing works on the cryptographic algorithms has been done. All the techniques are useful for

real-time cryptography. Each technique is unique in its own way, which might be suitable for different applications and has its own advantages and disadvantages. The challenges in each technique are then discussed. In our future work, this concept will be further explored and a combination of algorithms will be applied to form a method for providing a more secure environment.

REFERENCES

[1] Gurpreet Singh and Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security", International Journal of Computer Applications, Volume 67, No.19, 2013.
[2] G.Naga Satish, Ch.V.Raghavendran, P.T.K.Mehar, P. Suresh Varma, "Secret Key Cryptographic Algorithm", International Journal of Computer Science, Information Technology and Management, Vol. 1 No. 1-2, 2012.
[3] Akash Kumar Mandal, Chandra Parakash and Mrs. Archana Tiwari, "Performance Evaluation of Cryptographic Algorithms: DES and AES", IEEE Students' Conference on Electrical, Electronics and Computer Science, pp. 1-5, 2012.
[4] William Stallings, "Cryptography and etwork Security: Principles and Practice", Pearson Education/Prentice Hall, 5th Edition.
[5] Shivendra Singh, Sarfaraz Iqbal, Arunima Jaiswal, "Survey on Techniques Developed using Digital Signature: Public key Cryptography", International Journal of Computer Applications, Volume 117, No. 16, 2015.
[6] Amos Beimel, "Secret-Sharing Schemes: A Survey".
[7] G. Julius Caesar and John F. Kennedy, "Security Engineering: A Guide to Building Dependable Distributed Systems", Chapter 5: Cryptography.
[8] William Barker, William Burr, William Polk and Miles Smid, "Recommendation for Key Management (Part 1: General)", NIST Special Publication SP800-57, Elaine Barker, 2007.
[9] Sankeeth Kumar Chinta, "Blowfish", 2015.
[10] Burton S. Kaliski and Yiqun Lisa Yin, "On the Security of the RC5 Encryption Algorithm", RSA Laboratories Technical Report TR-602, Version 1.0, 1998.
[11] K.Sivaraman, "A Comparision Study of RSA and DSA Algorithm in Mobile Cloud Computing", International Journal of Pure and Applied Mathematics, Volume 116 No. 8, 2017, 247-253.
[12] Anoop MS, "Elliptic Curve Cryptography: An Implementation Guide".
[13] Arnold Reinhold et al, "ElGamal Signature Scheme".
[14] Dr. Dahlia Malkhi, "An advance course in computer and network security", Lecture notes, The Hebrew University of Jerusalem Secret Sharing.
[15] K.Suganya and K.Ramya, "Performance study on Diffie Hellman Key Exchange Algorithm", International Journal for Research tn Applied Science And Engineering Technology (IJRASET), Vol. 2 Issue III, March 2014.