International Transaction of Electrical and Computer Engineers System, 2015, Vol. 3, No. 1, 1-11
Available online at http://pubs.sciepub.com/iteces/3/1/1

DOI:10.12691/iteces-3-1-1

Review on Network Security and Cryptography

Shyam Nandan Kumar*

M.Tech-Computer Science and Engineering, Lakshmi Narain College of Technology-Indore (RGPV, Bhopal), MP, India
*Corresponding author: shyamnandan.mec@gmail.com

Received March 02, 2015; Revised March 12, 2015; Accepted March 17, 2015

Abstract With the advent of the World Wide Web and the emergence of e-commerce applications and social networks, organizations across the world generate a large amount of data daily. Data security is the utmost critical issue in ensuring safe transmission of information through the internet. Also network security issues are now becoming important as society is moving towards digital information age. As more and more users connect to the internet it attracts a lot of cyber-criminals. It comprises authorization of access to information in a network, controlled by the network administrator. The task of network security not only requires ensuring the security of end systems but of the entire network. In this paper, an attempt has been made to review the various Network Security and Cryptographic concepts. This paper discusses the state of the art for a broad range of cryptographic algorithms that are used in networking applications.

Keywords: *network security, cryptography, decryption, encryption*

Cite This Article: Shyam Nandan Kumar, "Review on Network Security and Cryptography." *International Transaction of Electrical and Computer Engineers System*, vol. 3, no. 1 (2015): 1-11. doi: 10.12691/iteces-3-1-1.

1. Introduction

Internet has become more and more widespread, if an unauthorized person is able to get access to this network, he can not only spy on us but he can easily mess up our lives. Network Security & Cryptography is a concept to protect network and data transmission over wireless network. A network security system typically relies on layers of protection and consists of multiple components including networking monitoring and security software in addition to hardware and appliances. All components work together to increase the overall security of the computer network. Security of data can be done by a technique called ***cryptography***. So one can say that cryptography is an emerging technology, which is important for network security.

Model for Cryptosystem Using Neural Network [1] supports high security. Neural network and cryptography together can make a great help in field of networks security. The key formed by neural network is in the form of weights and neuronal functions which is difficult to break. Here, content data would be used as an input data for cryptography so that data become unreadable for attackers and remains secure from them. The ideas of mutual learning, self-learning, and stochastic behavior of neural networks and similar algorithms can be used for different aspects of cryptography, like public-key cryptography, solving the key distribution problem using neural network mutual synchronization, hashing or generation of pseudo-random numbers. Another idea is the ability of a neural network to separate space in non-linear pieces using "bias". It gives different probabilities of activating or not the neural network. This is very useful in the case of Cryptanalysis.

Network security [2] consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password.

Cryptography is the science of writing in secret code. More generally, it is about constructing and analyzing protocols that block adversaries; [3] various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation [4] are central to modern cryptography. Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce. The development of the *World Wide Web* resulted in broad use of cryptography for e-commerce and business applications. Cryptography is closely related to the disciplines of ***cryptology*** and ***cryptanalysis***. Techniques used for decrypting a message without any knowledge of the encryption details fall into the area of cryptanalysis. Cryptanalysis is what the layperson calls "breaking the code." The areas of

cryptography and cryptanalysis together are called cryptology. ***Encryption*** is the process of converting ordinary information (called plaintext) into unintelligible text (called ciphertext). ***Decryption*** is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext. ***Cryptosystem*** is the ordered list of elements of finite possible plaintexts, finite possible cyphertexts, finite possible keys, and the encryption and decryption algorithms which correspond to each key.

The challenging problem is how to effectively share encrypted data. Encrypt message with strongly secure key which is known only by sending and recipient end is a significant aspect to acquire robust security in sensor network. The secure exchange of key between sender and receiver is too much difficult task in resource constraint sensor network. data should be encrypted first by users before it is outsourced to a remote cloud storage service and both data security and data access privacy should be protected such that cloud storage service providers have no abilities to decrypt the data, and when the user wants to search some parts of the whole data, the cloud storage system will provide the accessibility without knowing what the portion of the encrypted data returned to the user is about. This paper reviews various network security and cryptographic approaches.

In this paper sections are organized as follows: Section 2 gives the idea about types of security attacks on cloud. Section 3 deals with security services. Section 4 explains network security model. Section 5 describes the various cryptography mechanism. Section 6 gives the idea about message authentication. Section 7 shows network and internet related security approach. Firewalls technique is provide in section 8. Section 9 concludes the paper and presents avenues for future work. References for this paper are given in section 10.

2. Types of Security Attacks

2.1. Passive Attacks

This type of attacks includes observation or monitoring of communication. A passive attack attempts to learn or make use of information from the system but does not affect system resources. The goal of the opponent is to obtain information that is being transmitted. Types of passive attacks:

- **Traffic Analysis:** The message traffic is sent and received in an apparently normal fashion, and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.
- **Release of Message Contents:** Read contents of message from sender to receiver.

2.2. Active Attacks

An active attack attempts to alter system resources or affect their operation. It involves some modification of the data stream or the creation of a false stream. Types of active attacks:

- **Modification of Messages:** some portion of a legitimate message is altered, or that messages are delayed or reordered.
- **Denial of Service:** An entity may suppress all messages directed to a particular destination.
- **Replay:** It involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.
- **Masquerade:** It takes place when one entity pretends to be a different entity.

3. Security Services

It is a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers. It enhances the security of data processing and transferring.

3.1. Data Integrity

It can apply to a stream of messages, a single message, or selected fields within a message. A loss of integrity is the unauthorized modification or destruction of information.

3.2. Data Confidentiality

Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information.

3.3. Authenticity

Provide authentication to all the node and base station for utilizing the available limited resources. It also ensures that only the authorized node can participant for the communication.

3.4. Nonrepudiation

Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

3.5. Access Control

Access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

4. Network Security Model

Figure 1 shows the model of network security. A message is to be transferred from one party to another across some sort of Internet service. A third party may be responsible for distributing the secret information to the sender and receiver while keeping it from any opponent. Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

- A security-related transformation on the information to be sent. Message should be encrypted by key so that it is unreadable by the opponent.
- An encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.

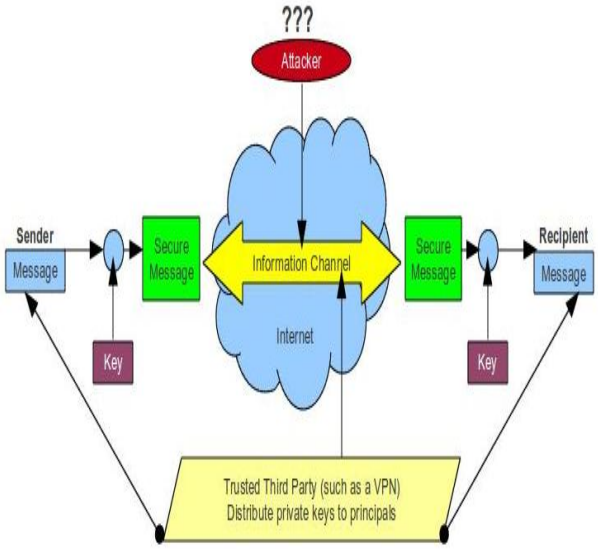


Figure 1. Model for Network Security

4.1. Need for Key Management in Cloud

Encryption provides data protection while key management enables access to protected data. It is strongly recommended to encrypt data in transit over networks, at rest, and on backup media. In particular, data to encrypt their own data. Both encryption and key management are very important to help secure applications and data stored in the Cloud. Requirements of effective key management are discuss below.

- ***Secure key stores:*** The key stores themselves must be protected from malicious users. If a malicious user gains access to the keys, they will then be able to access any encrypted data the key is corresponded to. Hence the key stores themselves must be protected in storage, in transit and on backup media.
- ***Access to key stores:*** Access to the key stores should be limited to the users that have the rights to access data. Separation of roles should be used to help control access. The entity that uses a given key should not be the entity that stores the key.
- ***Key backup and recoverability:*** Keys need secure backup and recovery solutions. Loss of keys, although effective for destroying access to data, can be highly devastating to a business and Cloud providers need to ensure that keys aren't lost through backup and recovery mechanisms.

5. Cryptography Mechanism

Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. The term is most often associated with scrambling plaintext message (ordinary text, sometimes referred to as cleartext) into ciphertext (a process called ***encryption***), then back again (known as ***decryption***). There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions, each of which is described below.

5.1. Secret Key Cryptography

With secret key cryptography, a single key is used for both encryption and decryption. As shown in Figure 2, the sender *A* uses the key *K* (or some set of rules) to encrypt the plaintext message *M* and sends the ciphertext *C* to the receiver. The receiver applies the ***same key*** *K* (or ruleset) to decrypt the cipher text *C* and recover the plaintext message *M*. Because a single key is used for both functions, secret key cryptography is also called ***symmetric encryption***.

With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that, in fact, is the secret. The biggest difficulty with this approach, of course, is the distribution of the key.

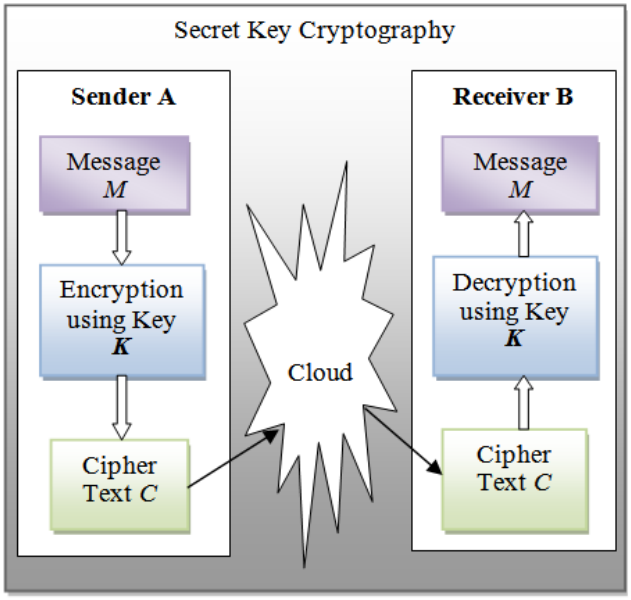


Figure 2. Secret Key Cryptography

Secret key cryptography schemes are generally categorized as being either ***stream ciphers*** or ***block ciphers.*** Stream ciphers operate on a single bit (byte or computer word) at a time and implement some form of feedback mechanism so that the key is constantly changing. A block cipher is so-called because the scheme encrypts one block of data at a time using the same key on each block. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher. Block ciphers can operate in one of several modes; the following four are the most important:

- ***Electronic Codebook (ECB)*** mode is the simplest, most obvious application: the secret key is used to encrypt the plaintext block to form a ciphertext block. Two identical plaintext blocks, then, will always generate the same ciphertext block. Although this is the most common

mode of block ciphers, it is susceptible to a variety of brute-force attacks.

- ***Cipher Block Chaining (CBC)*** mode adds a feedback mechanism to the encryption scheme. In CBC, the plaintext is exclusively-ORed (XORed) with the previous ciphertext block prior to encryption. In this mode, two identical blocks of plaintext never encrypt to the same ciphertext.
- ***Cipher Feedback (CFB)*** mode is a block cipher implementation as a self-synchronizing stream cipher. CFB mode allows data to be encrypted in units smaller than the block size, which might be useful in some applications such as encrypting interactive terminal input. If we were using 1-byte CFB mode, for example, each incoming character is placed into a shift register the same size as the block, encrypted, and the block transmitted. At the receiving side, the ciphertext is decrypted and the extra bits in the block (i.e., everything above and beyond the one byte) are discarded.
- ***Output Feedback (OFB)*** mode is a block cipher implementation conceptually similar to a synchronous stream cipher. OFB prevents the same plaintext block from generating the same ciphertext block by using an internal feedback mechanism that is independent of both the plaintext and ciphertext bitstreams.

Stream ciphers come in several flavors but two are worth mentioning here. ***Self-synchronizing stream ciphers*** calculate each bit in the keystream as a function of the previous n bits in the keystream. It is termed "self-synchronizing" because the decryption process can stay synchronized with the encryption process merely by knowing how far into the n-bit keystream it is. One problem is error propagation; a garbled bit in transmission will result in n garbled bits at the receiving side. ***Synchronous stream ciphers*** generate the keystream in a fashion independent of the message stream but by using the same keystream generation function at sender and receiver. While stream ciphers do not propagate transmission errors, they are, by their nature, periodic so that the keystream will eventually repeat.

Secret key cryptography algorithms that are in use today include:

- ***Data Encryption Standard (DES):*** DES is a block-cipher employing a 56-bit key that operates on 64-bit blocks. DES algorithm as described by Davis R. [5] takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into cipher text bit string of the same length. 3DES (Triple DES) [6] is an enhancement of DES; it is 64 bit block size with 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time.
- ***Advanced Encryption Standard (AES):*** AES [7,8] is a block cipher intended to replace DES for commercial applications. It uses a 128-bit block size and a key size of 128, 192, or 256 bits. The number of internal rounds of the cipher is a function of the key length. The number of rounds for 128- bit key is 10. Unlike its predecessor DES, AES does not use a Feistel network. Feistel networks do not encrypt an entire block per iteration, e.g., in DES, 64/2 = 32 bits are encrypted in one round. AES, on the other hand, encrypts all 128 bits in one iteration.
- ***Blowfish:*** Blowfish [9] is a symmetric 64-bit block cipher, invented by Bruce Schneier; optimized for 32-bit processors with large data caches, it is significantly faster than DES on a Pentium/PowerPC-class machine. Key lengths can vary from 32 to 448 bits in length. Blowfish, available freely and intended as a substitute for DES or IDEA, is in use in a large number of products. It is a 16-round Feistel cipher and uses large key-dependent S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the final round, each half of the data block is XORed with one of the two remaining unused P-entries.
- ***Twofish:*** [10] A 128-bit block cipher using 128-, 192-, or 256-bit keys. Designed to be highly secure and highly flexible, well-suited for large microprocessors, 8-bit smart card microprocessors, and dedicated hardware. Designed by a team led by Bruce Schneier and was one of the Round 2 algorithms in the AES process. Twofish's distinctive features are the use of pre-computed key-dependent S-boxes, and a relatively complex key schedule. One half of an n-bit key is used as the actual encryption key and the other half of the n-bit key is used to modify the encryption algorithm (key-dependent S-boxes). Twofish borrows some elements from other designs; for example, the pseudo-Hadamard transform(PHT) from the SAFER family of ciphers. Twofish has a Feistel structure like DES.
- ***Camellia:*** [11] A secret-key, block-cipher crypto algorithm developed jointly by Nippon Telegraph and Telephone (NTT) Corp. and Mitsubishi Electric Corporation (MEC) in 2000. C has some characteristics in common with AES: a 128-bit block size, support for 128-, 192-, and 256-bit key lengths, and suitability for both software and hardware implementations on common 32-bit processors as well as 8-bit processors (e.g., smart cards, cryptographic hardware, and embedded systems). Camellia is a Feistel cipher with either 18 rounds (when using 128-bit keys) or 24 rounds (when using 192 or 256-bit keys). Every six rounds, a logical transformation layer is applied: the so-called "FL-function" or its inverse. Camellia uses four 8 x 8-bit S-boxes with input and output affine transformations and logical operations. The cipher also uses input and output key whitening. The diffusion layer uses a linear transformation based on a matrix with a branch number of 5.
- ***KASUMI:*** [11,12] A block cipher using a 128-bit key and block size 64-bit, is part of the Third-Generation Partnership Project (3gpp), formerly known as the Universal Mobile Telecommunications System (UMTS). KASUMI is the intended confidentiality and integrity algorithm for both message content and signaling data for emerging mobile communications systems. KASUMI is used in the A5/3 key stream generator and in GPRS in the GEA3 key stream generator. In 2010, Dunkelman, Keller and Shamir published a new attack that allows an adversary to recover a full A5/3 key by related-key attack [13]. The core of KASUMI is an eight-round Feistel

network. The round functions in the main Feistel network are irreversible Feistel-like network transformations. In each round the round function uses a round key which consists of eight 16-bit sub keys derived from the original 128-bit key using a fixed key schedule.

5.2. Public-Key Cryptography

Public-key cryptography is a form of cryptosystem in which encryption and decryption are performed using the different keys—one a ***public key*** and one a ***private key***. These keys are mathematically related although knowledge of one key does not allow someone to easily determine the other key. As shown in Figure 3, the sender *A* uses the public key of receiver *B* (or some set of rules) to encrypt the plaintext message *M* and sends the ciphertext *C* to the receiver. The receiver applies own private key (or ruleset) to decrypt the cipher text *C* and recover the plaintext message *M*. Because pair of keys is required, this approach is also called ***asymmetric cryptography***. Asymmetric encryption can be used for ***confidentiality***, ***authentication***, or both. Applications for Public-Key Cryptosystems are given in Table 1.

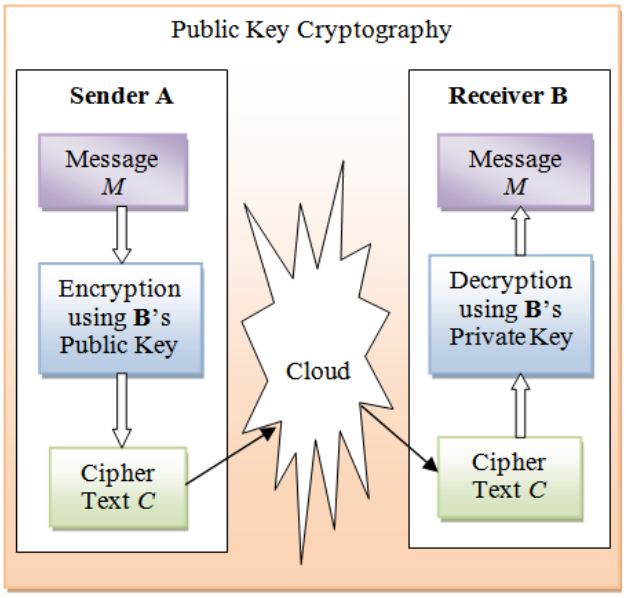


Figure 3. Public Key Cryptography

Public-key cryptography algorithms that are in use today for ***key exchange*** or ***digital signatures*** include:

5.2.1. RSA

The first, and still most common, public key cryptography implementation, named for the three MIT mathematicians who developed it — Ronald Rivest, Adi Shamir, and Leonard Adleman [14]. RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, n, that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an n with roughly twice as many digits as the prime factors. RSA has three phases: ***Key Generation***, ***Encryption***, and ***Decryption***.

5.2.1.1. Key Generation Phase

Receiver generates a public/private key pair. Algorithm is as follow:

1) Select p, q such that p and q both are prime, $p \neq q$
2) Calculate $n = p * q$
3) Calculate $f(n) = (p - 1)(q - 1)$
4) Select integer e such that $\gcd(f(n), e) = 1; 1 < e < f(n)$
5) Calculate d such that $d \equiv e\text{-}1 \pmod{f(n)}$
6) Public key PUK= (e, n)
7) Private key PRK=(d, n)

5.2.1.2. Encryption Phase

Encryption is done by sender with receiver's Public Key. Algorithm is as follow:

1) Plain Text M is known, $M < n$
2) Cipher Text C is calculated as

$$C = M^e \bmod n$$

5.2.1.3. Decryption Phase

Decryption is done by receiver using his Private Key. Algorithm is as follow:

1) Cipher Text C is known
2) Plain Text M is calculated as

$$M = C^d \bmod n$$

5.2.2. Diffie-Hellman Key Exchange

A simple public-key algorithm is Diffie-Hellman key exchange [15]. This protocol enables two users to establish a secret key using a public-key scheme based on discrete logarithms. The protocol is secure only if the authenticity of the two participants can be established. D-H is used for secret-key key exchange only, and not for authentication or digital signatures. Algorithm is as follow:

1) Select two Global Public Elements: a prime number p and an integer α that is a primitive root of p.
2) Sender Key Generation: Sender selects a random integer $XA < p$ which is private and computes $YA = \alpha\ XA \bmod p$, which is public.
3) Receiver Key Generation: Receiver selects a random integer $XB < p$ which is private and computes $YB = \alpha\ XB \bmod p$, which is public.
4) Sender calculates secret key: $K = (YB\ )\ XA \bmod p$
5) Receiver calculates secret key which is identical to sender secret key. $K = (YA)\ XB \bmod p$.

5.2.3. Elliptic Curve Cryptography

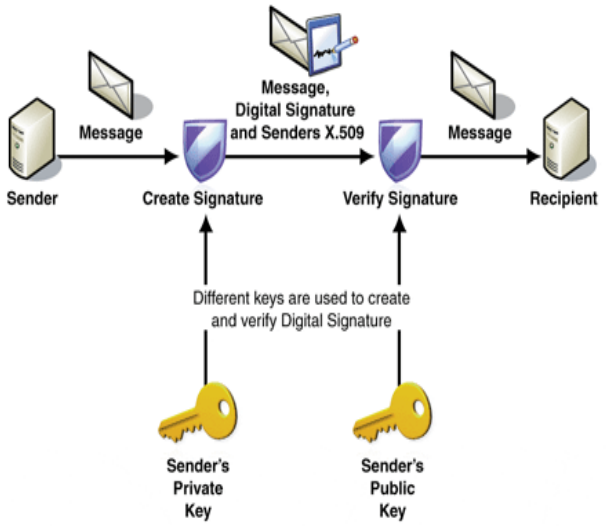
It is analog of Diffie-Hellman Key Exchange. ECC [16,17] is a public key cryptography algorithm based upon elliptic curves. Elliptic curve arithmetic can be used to develop a variety of elliptic curve cryptography (ECC) schemes, including key exchange, encryption, and digital signature. For purposes of ECC, elliptic curve arithmetic involves the use of an elliptic curve equation defined over a finite field. The coefficients and variables in the equation are elements of a finite field. Security of ECC is based on the intractability of ECDLP i.e. Elliptic Curve Discrete Logarithm Problem.

5.2.4. Digital Signature Standard

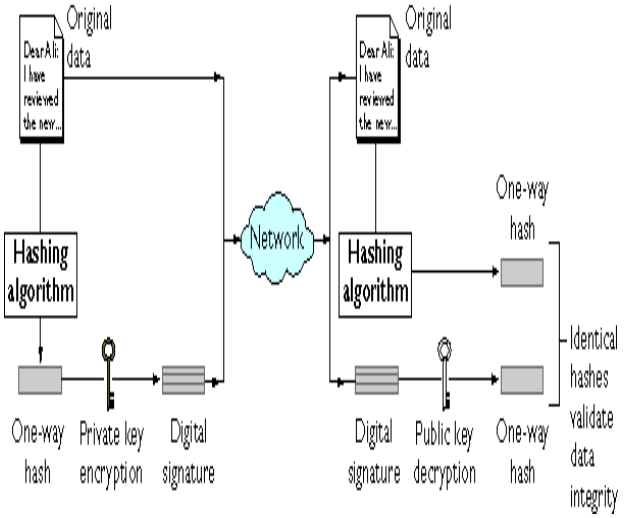
The digital signature standard (DSS) is an NIST standard that uses the secure hash algorithm (SHA) [18].

A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. Typically the signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.

Figure 4 shows the process of making and using digital signatures. Sender can sign a message using a digital signature generation algorithm. The inputs to the algorithm are the message and sender's private key. Any other user, say receiver, can verify the signature using a verification algorithm, whose inputs are the message, the signature, and sender's public key.



a). Digital Signature Without Hash Function



b). Digital Signature With Hash Function

Figure 4. Digital Signature Process

The DSS uses an algorithm that is designed to provide only the digital signature function. It cannot be used for encryption or key exchange. Nevertheless, it is a public-key technique. Figure 5 contrasts the DSS approach for generating digital signatures to that used with RSA. In the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length. This hash code is then encrypted using the sender's private key to form the signature. Both the message and the signature are then transmitted. The recipient takes the message and produces a hash code. The recipient also decrypts the signature using the sender's public key. If the calculated hash code matches the decrypted signature, the signature is accepted as valid. Because only the sender knows the private key, only the sender could have produced a valid signature.

The DSS approach also makes use of a hash function. The hash code is provided as input to a signature function along with a random number generated for this particular signature. The signature function also depends on the sender's private key PR_a and a set of parameters known to a group of communicating principals. We can consider this set to constitute a global public key PU_G. The result is a signature consisting of two components, labeled s and r.

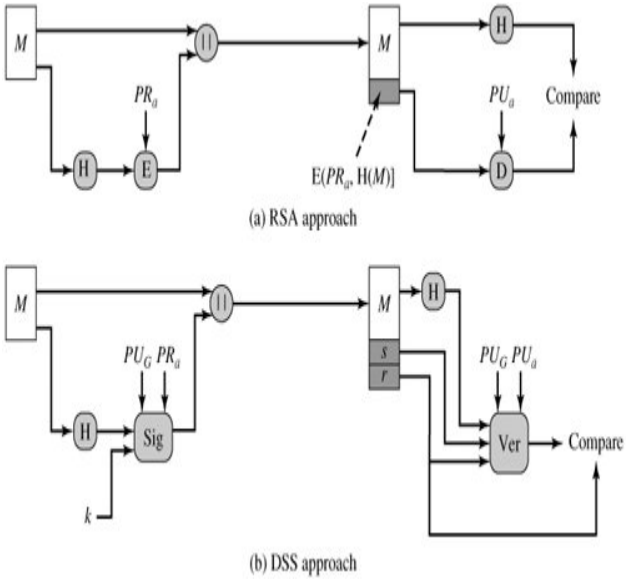


Figure 5. Digital Signature Approaches

Table 1. Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
Elliptic Curve	Yes	Yes	Yes
DSS	No	Yes	No

5.3. Hash Functions

Hash functions, also called ***message digests*** and ***one-way encryption***, are algorithms that, in some sense, use no key. A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value $h = H(M)$ as shown in Figure 6. In general terms, the principal object of a hash function is data integrity. A change to any bit or bits in results, with high probability, in a change to the hash code. Virtually all cryptographic hash functions involve the iterative use of a compression function. The compression function used in secure hash algorithms falls into one of two categories: a function specifically designed for the hash function or an algorithm based on a symmetric block cipher. SHA and Whirlpool [19] are examples of these two approaches, respectively.

The hash algorithm involves repeated use of a compression function, f, that takes two inputs (an -bit input from the previous step, called the chaining variable, and a -bit block) and produces an -bit output. At the start

of hashing, the chaining variable has an initial value that is specified as part of the algorithm. The final value of the chaining variable is the hash value. It is seen that b > n. A cryptographic hash function can be used to construct a pseudorandom function (PRF) or a pseudorandom number generator (PRNG).

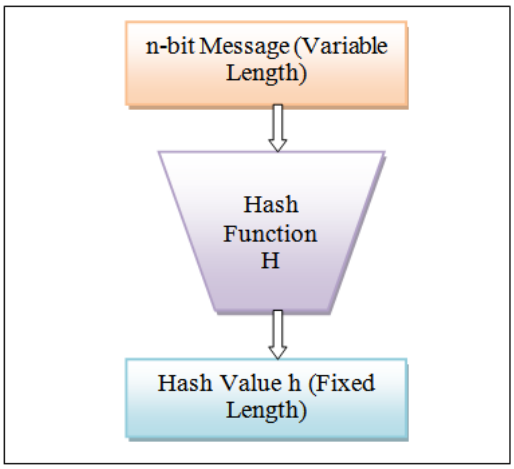


Figure 6. Block Diagram of Hash Function

5.3.1. SHA

Secure Hash Algorithm (SHA) is a family of cryptographic hash functions. Comparison of SHA Parameters is shown in Table 2. All sizes are measured in bits.

Table 2. Comparison of SHA Parameters

Algorithm	Message Digest Size	Message Size	Block Size	Word Size	No of Step
SHA-1	160	$< 2^{64}$	512	32	80
SHA-224	224	$< 2^{64}$	512	32	64
SHA-256	256	$< 2^{64}$	512	32	64
SHA-384	384	$< 2^{128}$	1024	64	80
SHA-512	512	$< 2^{128}$	1024	64	80

6. Message Authentication Code

Message authentication is a mechanism or service used to verify the integrity of a message. Message authentication assures that data received are exactly as sent (i.e., contain no modification, insertion, deletion, or replay). In many cases, there is a requirement that the authentication mechanism assures that purported identity of the sender is valid. When a hash function is used to provide message authentication, the hash function value is often referred to as a ***message digest***. More commonly, message authentication is achieved using a ***message authentication code (MAC)***, also known as a ***keyed hash function*** or ***cryptographic checksum***. Typically, MACs are used between two parties say sender and receiver, that share a secret key *K* to authenticate information exchanged between those parties. A MAC function *C* takes as input a secret key *K* and a variable-length data block or message *M* and produces a fixed-length hash value *MAC*, referred to as the *message authentication Code*. This can then be transmitted with or stored with the protected message. If the integrity of the message needs to be checked, the MAC function can be applied to the message and the result compared with the stored MAC value. Process of MAC is shown in Figure 7.

$$MAC = C(K, M)$$

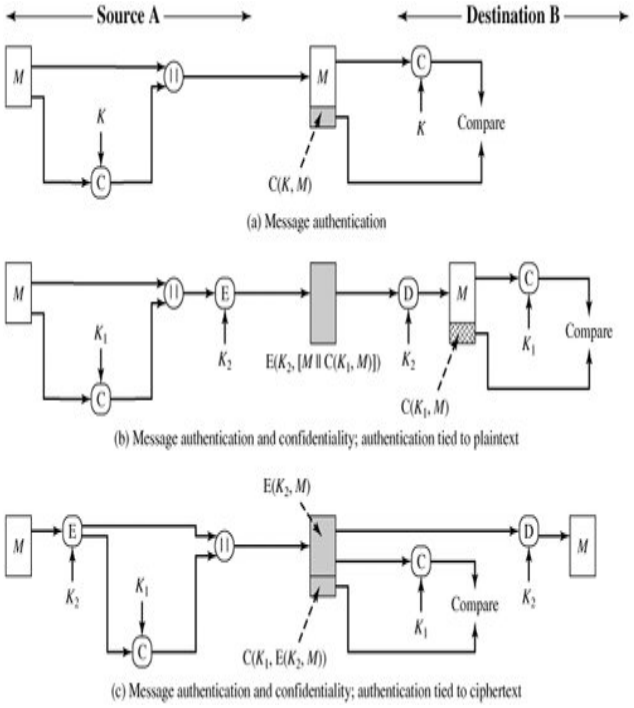


Figure 7. Working of MAC

6.1. HMAC

Hash-based message authentication code (***HMAC***) [20] is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and the authentication of a message. Any cryptographic hash function, such as MD5 or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly. The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and on the size and quality of the key.

Hash-based message authentication code (HMAC) provides the server and the client each with a public and private key. The public key is known, but the private key is known only to that specific server and that specific client. The client creates a unique HMAC, or hash, per request to the server by combing the request data and hashing that data, along with a private key and sending it as part of a request. The server receives the request and regenerates its own unique HMAC. The server compares the two HMACs, and, if they're equal, the client is trusted and the request is executed. This process is often called a ***secret handshake***.

HMAC can be expressed as:

$$HMAC(K, M) = H[K^{+} \oplus opad) \,||\, H[K^{+} \oplus ipad) \,||\, M]]$$

where

K = secret key; recommended length is ≥ n; if key length is greater than *b*-bit block, the key is input to the hash function to produce an *n*-bit key

M = message input to HMAC,

H = cryptographic hash function,

K^+ = K padded with zeros on the left so that the result is b bits in length,

$$\oplus = \text{ exclusive or } (\text{XOR}),$$

$\|$= concatenation,

opad = 01011100 (5C in hexadecimal) repeated b/8 times,

ipad = 00110110 (36 in hexadecimal) repeated b/8 times.

6.2. CMAC

Cipher-based message authentication codes (***CMAC***$_s$) [21] are a tool for calculating *message authentication codes* using a block cipher coupled with a secret key. CMAC can be used to verify both the integrity and authenticity of a message. This mode of operation fixes security deficiencies of CBC-MAC (CBC-MAC is secure only for fixed-length messages). To generate an ℓ-bit CMAC tag (t) of a message (m) using a b-bit block cipher (E) and a secret key (k), one first generates two b-bit sub-keys (k_1 and k_2).

Sub-keys ($k1$ and $k2$) Algorithm:

1) Calculate a temporary value $k_0 = E_k(0)$.
2) If msb(k_0) = 0, then $k_1 = k_0 \ll 1$, else $k_1 = (k_0 \ll 1) \oplus C$; where C is a certain constant that depends only on b. (Specifically, C is the non-leading coefficients of the lexicographically first irreducible degree-b binary polynomial with the minimal number of ones.)
3) If msb(k_1) = 0, then $k_2 = k_1 \ll 1$, else $k_2 = (k_1 \ll 1) \oplus C$.
4) Return keys (k_1, k_2) for the MAC generation process.

CMAC Tag Generation Algorithm:

1) Divide message into b-bit blocks $m = m_1 \,//\, \ldots \,//\, m_{n-1} \,//\, m_n'$ where $m_1, \ldots, m_{n-1}$ are complete blocks. (The empty message is treated as 1 incomplete block.)
2) If m_n' is a complete block then $m_n = k_1 \oplus m_n'$ else $m_n = k_2 \oplus (m_n' // 10\ldots0_2)$.
3) Let $c_0 = 00\ldots0_2$.
4) For $i = 1,\ldots, n$, calculate $c_i = E_k(c_{i-1} \oplus m_i)$.
5) Output $t = \text{msb}\ell(c_n)$.

7. Network and Internet Security

Internet security is a tree branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole. Its objective is to establish rules and measures to use against attacks over the Internet. The Internet represents an insecure channel for exchanging information leading to a high risk of intrusion or fraud, such as phishing. Different methods have been used to protect the transfer of data, including encryption.

Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority.

Types of Network Security:

7.1. Wireless Network Security

Wireless security is the prevention of unauthorized access or damage to computers using wireless networks. The most common types of wireless security are Wired Equivalent Privacy (***WEP***) and Wi-Fi Protected Access (***WPA***). WEP is a notoriously weak security standard. The password it uses can often be cracked in a few minutes with a basic laptop computer and widely available software tools. WAP security is primarily provided by the Wireless Transport Layer Security (WTLS), which provides security services between the mobile device (client) and the WAP gateway to the Internet. There are several approaches to WAP end-to-end security. One notable approach assumes that the mobile device implements TLS over TCP/IP and the wireless network supports transfer of IP packets. The WAP architecture is designed to cope with the two principal limitations of wireless Web access: the limitations of the mobile node (small screen size, limited input capability) and the low data rates of wireless digital networks. Two important WTLS concepts are the secure session and the secure connection, which are defined in the specification as:

1) ***Secure connection:*** A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session. Between any pair of parties (applications such as HTTP on client and server), there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice.
2) ***Secure session:*** An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection. There are a number of states associated with each session. Once a session is established, there is a current operating state for both read and write (i.e., receive and send). In addition, during the Handshake Protocol, pending read and write states are created. Upon successful conclusion of the Handshake Protocol, the pending states become the current states.

7.2. IP Security

Internet Protocol Security (***IPsec***) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network), or between a security gateway and a host (network-to-host). IPsec is said to be especially useful for implementing virtual private networks and for remote user access through dial-up connection to private networks. A big advantage of IPsec is that security arrangements can

be handled without requiring changes to individual user computers.

IPsec provides two choices of security service: Authentication Header (AH), which essentially allows authentication of the sender of data, and Encapsulating Security Payload (ESP), which supports both authentication of the sender and encryption of data as well. The specific information associated with each of these services is inserted into the packet in a header that follows the IP packet header. Separate key protocols can be selected, such as the ISAKMP/Oakley protocol. IPsec uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection. IPsec protects any application traffic over an IP network. Applications can be automatically secured by IPsec at the IP layer.

7.2.1. Modes of Operation

IPsec can be implemented in a ***host-to-host transport mode***, as well as in a ***network tunneling mode***.

In transport mode, only the payload of the IP packet is usually encrypted and/or authenticated. The routing is intact, since the IP header is neither modified nor encrypted; however, when the authentication header is used, the IP addresses cannot be translated, as this will invalidate the hash value. The transport and application layers are always secured by hash, so they cannot be modified in any way (for example by translating the port numbers).

In tunnel mode, the entire IP packet is encrypted and/or authenticated. It is then encapsulated into a new IP packet with a new IP header. Tunnel mode is used to create virtual private networks for network-to-network communications (e.g. between routers to link sites), host-to-network communications (e.g. remote user access) and host-to-host communications (e.g. private chat).

7.3. Electronic Mail Security

Email is vulnerable to both passive and active attacks. The protection of email from unauthorized access and inspection is known as electronic privacy. In countries with a constitutional guarantee of the secrecy of correspondence, email is equated with letters and thus legally protected from all forms of eavesdropping. With the explosively growing reliance on e-mail, there grows a demand for authentication and confidentiality services. Two schemes stand out as approaches that enjoy widespread use: Pretty Good Privacy (***PGP***) and Secure/Multipurpose Internet Mail Extension ***S/MIME***.

PGP is an open-source, freely available software package for e-mail security. It provides ***authentication*** through the use of digital signature, ***confidentiality*** through the use of symmetric block encryption, ***compression*** using the ZIP algorithm, and ***e-mail compatibility*** using the radix-64 encoding scheme. PGP incorporates tools for developing a public-key trust model and public-key certificate management.

S/MIME is an Internet standard approach to e-mail security that incorporates the same functionality as PGP. It is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.

7.4. Transport-Level Security

Transport-Level Security (***TLS***) is an IETF standardization initiative whose goal is to produce an Internet standard version of ***SSL***. Secure Socket Layer (SSL) provides security services between TCP and applications that use TCP. The Internet standard version is called Transport Layer Service (TLS). The TLS Record Format is the same as that of the SSL Record Format. SSL/TLS provides confidentiality using symmetric encryption and message integrity using a message authentication code. SSL/TLS includes protocol mechanisms to enable two TCP users to determine the security mechanisms and services they will use. HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server. Secure Shell (SSH) provides secure remote logon and other secure client/server facilities. The SSH Connection Protocol runs on top of the SSH Transport Layer Protocol and assumes that a secure authentication connection is in use. All types of communication using SSH, such as a terminal session, are supported using separate channels.

8. Firewalls

A firewall forms a barrier through which the traffic going in each direction must pass. A firewall security policy dictates which traffic is authorized to pass in each direction. Firewalls impose restrictions on incoming and outgoing Network packets to and from private networks. Incoming or outgoing traffic must pass through the firewall; only authorized traffic is allowed to pass through it. Firewalls create checkpoints between an internal private network and the public Internet, also known as choke points(borrowed from the identical military term of a combat limiting geographical feature). Firewalls can create choke points based on IP source and TCP port number. They can also serve as the platform for IPsec. Using tunnel mode capability, firewall can be used to implement VPNs. Firewalls can also limit network exposure by hiding the internal network system and information from the public Internet. A firewall may be designed to operate as a filter at the level of IP packets, or may operate at a higher protocol layer.

8.1. Characteristics of Firewalls

Firewalls can be an effective means of protecting a local system or network of systems from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet. It includes following characteristics:

- All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall.
- Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies.

- The firewall itself is immune to penetration. This implies the use of a hardened system with a secured operating system. Trusted computer systems are suitable for hosting a firewall and often required in government applications.

8.2. Types of Firewalls

A firewall may act as a packet filter. It can operate as a positive filter, allowing to pass only packets that meet specific criteria, or as a negative filter, rejecting any packet that meets certain criteria. Depending on the type of firewall, it may examine one or more protocol headers in each packet, the payload of each packet, or the pattern generated by a sequence of packets. Types of firewalls are shown in Figure 8.

8.2.1. Packet Filter

A packet filter is a first generation firewall that processes network traffic on a packet-by-packet basis. Its main job is to filter traffic from a remote IP host, so a router is needed to connect the internal network to the Internet. The router is known as a screening router, which screens packets leaving and entering the network. Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions. For example, a packet filter firewall cannot block specific application commands; if a packet filter firewall allows a given application, all functions available within that application will be permitted. Packet filter firewalls are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as network layer address spoofing. Many packet filter firewalls cannot detect a network packet in which the OSI Layer 3 addressing information has been altered. Spoofing attacks are generally employed by intruders to bypass the security controls implemented in a firewall platform.

8.2.2. Stateful Packet Inspection

In a stateful firewall the circuit-level gateway is a proxy server that operates at the network level of an Open Systems Interconnection (***OSI***) model and statically defines what traffic will be allowed. Circuit proxies will forward Network packets (formatted unit of data) containing a given port number, if the port is permitted by the algorithm. The main advantage of a proxy server is its ability to provide Network Address Translation (***NAT***), which can hide the user's IP address from the Internet, effectively protecting all internal information from the Internet. A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall, but also records information about TCP connections. Some stateful firewalls also keep track of TCP sequence numbers to prevent attacks that depend on the sequence number, such as session hijacking. Some even inspect limited amounts of application data for some well-known protocols like FTP, IM and SIPS commands, in order to identify and track related connections.

8.2.3. Application-Level Gateway

An application-level firewall is a third generation firewall where a proxy server operates at the very top of the OSI model, the IP suite application level. It is also known as application proxy. The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. A network packet is forwarded only if a connection is established using a known protocol. Application-level gateways are notable for analyzing entire messages rather than individual packets of data when the data are being sent or received.

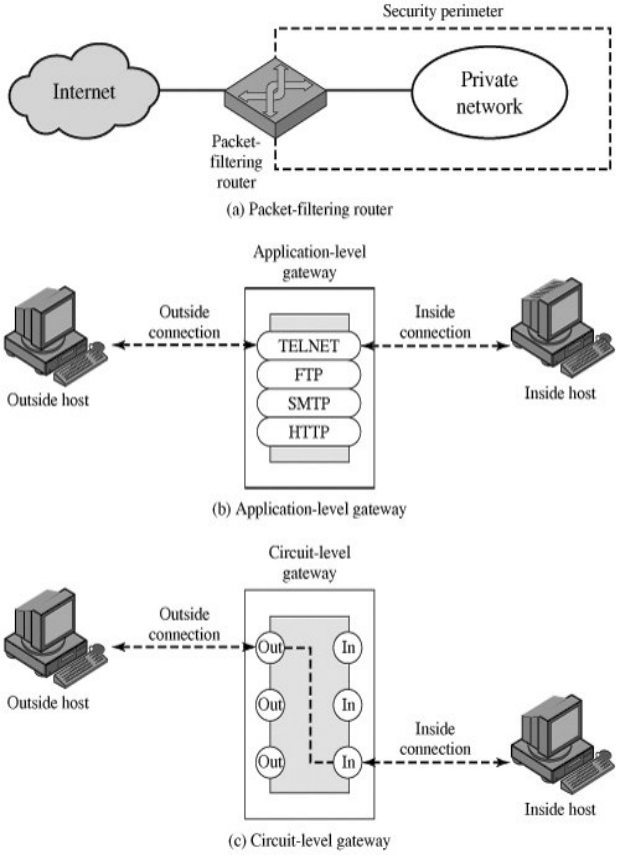


Figure 8. Types of Firewalls

9. Conclusion and Future Work

With the explosive growth in the Internet, network and data security have become an inevitable concern for any organization whose internal private network is connected to the Internet. The security for the data has become highly important. User's data privacy is a central question over cloud. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application.

The paper presented various schemes which are used in cryptography for Network security purpose. Encrypt message with strongly secure key which is known only by sending and recipient end, is a significant aspect to acquire robust security in cloud. The secure exchange of key between sender and receiver is an important task. The key management helps to maintain confidentiality of secret

information from unauthorized users. It can also check the integrity of the exchanged message to verify the authenticity. Network security covers the use of cryptographic algorithms in network protocols and network applications. This paper briefly introduces the concept of computer security, focuses on the threats of computer network security

In the future, work can be done on key distribution and management as well as optimal cryptography algorithm for data security over clouds.

References

[1] Shyam Nandan Kumar, "*Technique for Security of Multimedia using Neural Network*," Paper id-IJRETM-2014-02-05-020, IJRETM, Vol: 02, Issue: 05, pp.1-7. Sep-2014

[2] Simmonds, A; Sandilands, P; van Ekert, L (2004). "*An Ontology for Network Security Attacks*". *Lecture Notes in Computer Science*. *Lecture Notes in Computer Science* 3285: 317-323.

[3] Bellare, Mihir; Rogaway, Phillip (21 September 2005). "*Introduction". Introduction to Modern Cryptography*. p. 10.

[4] Menezes, A. J.; van Oorschot, P. C.; Vanstone, S. "*A. Handbook of Applied Cryptography*". ISBN 0-8493-8523-7.

[5] Davis, R., "*The Data Encryption Standard in Perspective," Proceeding of Communication Society magazine, IEEE*, Volume 16 No 6, pp. 5-6, Nov. 1978.

[6] S. NIST Special Publication 800-67, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, May 2004.

[7] Daemen, J., and Rijmen, V. "Rijndael: *AES-The Advanced Encryption Standard*, Springer, Heidelberg, March 2001.

[8] FIPS 197, *Advanced Encryption Standard*, *Federal Information Processing Standard*, *NIST, U.S. Dept. of Commerce*, November 26, 2001.

[9] Bruce Schneier (1993). "*Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)". Fast Software Encryption, Cambridge Security Workshop Proceedings (Springer-Verlag*): 191-204.

[10] Schneier, Bruce (2005-11-23). "*Twofish Cryptanalysis Rumors". Schneier on Security* blog. Retrieved 2013-01-14.

[11] Matsui, Mitsuru; Tokita, Toshio (Dec 2000). "*MISTY, KASUMI and Camellia Cipher Algorithm Development". Mitsibishi Electric Advance (Mitsibishi Electric corp.*) 100: 2-8. ISSN 1345-3041.

[12] *General Report on the Design, Speification and Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms*". 3GPP. 2009

[13] O. Dunkelman, N. Keller, A. Shamir, *\A practical-time attack on the KASUMI cryptosystem used in GSM and 3G telephony," Advances in Cryptology, Proceedings* Crypto'10, LNCS, T. Rabin, Ed., Springer, Heidelberg, 2010

[14] R.L.Rivest, A.Shamir, and L.Adleman, "*A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communication of the ACM*, Volume 21 No. 2, Feb. 1978.

[15] Diffie, W.; Hellman, M. (1976). "*New directions in cryptography". IEEE Transactions on Information Theory* 22 (6): 644-654.

[16] Koblitz, N., 1987. "*Elliptic curve cryptosystems. Mathematics of Computation*" 48, 203-209.

[17] Miller, V., 1985. "*Use of elliptic curves in cryptography*". CRYPTO 85.

[18] FIPS 180, *Secure Hash Standard, Federal Information Processing Standard (FIPS), Publication* 180, NIST, U.S. Dept. of Commerce, May 11, 1993.

[19] M. Lamberger, F. Mendel, C. Rechberger, V. Rijmen, M. Schl□a er, *\Rebound distinguishers: results on the full Whirlpool compression function," Advances in Cryptology, Proceedings Asiacrypt*'09, LNCS 5912, M. Matsui, Ed., Springer, Heidelberg, 2009, pp. 126-143.

[20] Bellare, Mihir; Canetti, Ran; Krawczyk, Hugo (1996). "*Keying Hash Functions for Message Authentication*".

[21] NIST Special Publication 800-38B, "*Recommendation for Block Cipher Modes of Operation": The CMAC Mode for Authentication*, May 2005.