Cryptographic Algorithms: Applications in Network Security

Saurabh Sindhu
Department of Computer Science, CRM Jat College, Hisar, Haryana, India

Divya Sindhu
Department of Computer Science, CRM Jat College, Hisar, Haryana, India

Abstract – Data security has been a major and challenging aspect in the modern era of information technology involving the internet and network applications. Especially it becomes serious in the cloud environment because the data is located in different places all over the world. The purpose of securing data is that only concerned and authorized users can access it. Different encryption algorithms provide the necessary protection against the data intruders' attacks by converting information from its normal form into an unreadable form. Security of data can be done by a technique called cryptography. Recently, the range of cryptography applications has expanded a lot in the area of network and the development of communication means. Cryptography is essentially required to ensure that data are protected against penetrations and to prevent the practice of spying. In this paper, the basic characteristics of different cryptographic algorithms i.e., Symmetric (secret) key cryptography, Asymmetric (public) key cryptography and Hashing cryptography are described. The application of these cryptographic algorithms has been explored in data and network security.

Keywords – Cryptography algorithms, Encryption, Decryption. Symmetric key, Asymmetric key, Hashing Algorithms, Network security

I. INTRODUCTION

Cryptography is the science of writing in secret code and is an ancient art [1, 2]. According to some experts, cryptography appeared spontaneously sometime after writing was invented, with applications ranging from diplomatic missives to war-time battle plans. However, new forms of cryptography came soon after the widespread development of computer communications. In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about any network, particularly the Internet [3 - 5]. There are many applications of cryptography, which include secure commerce and payments to private communications and protection of passwords.

Information Technology (IT) services such as network, storage, hardware, software and resources are provided to a customer over a network. In recent years, advances in Web Technology and the proliferation of mobile devices and sensors connected to the internet have resulted in the generation of immense data sets that need to be processed and stored (Figure 1). Moreover, business and commerce applications generate massive volume of data which has to be managed and analyzed by traditional data processing tools. For protection of data, computer and network security is a fast moving technology [6]. Network security originally focused on algorithmic aspects such as encryption and hashing techniques. As crackers troubled away at networks and systems, there is a need to protect that data against unauthorized access, alternation or interchanging [7 - 9]. Security is considered as one of the most critical features for computer network due to sensitivity and importance of data stored.

II. COMPONENTS USED IN NETWORK SECURITY

Symmetric encryption, also referred to as conventional encryption or single-key encryption, was the only type of encryption in use prior to the introduction of public key encryption in the late 1970s. Various individuals and groups to present day diplomatic, military and commercial users, have used symmetric encryption for secret communication. A symmetric encryption scheme has five ingredients.

- *Plaintext:* This is the original message or data that is fed into the algorithm as input. In cryptography data that can be read and understood without any special measures is called plaintext or clear text
- *Encryption algorithm*: The method of covering up of plaintext in such a way as to hide its substance is called encryption. Encryption is a well-known technology for protecting sensitive data. The encryption algorithm performs various substitutions and transformations on the plaintext (Figure 2).

- *Secret key*: The secret key is also input to the encryption algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.
- *Cipher text:* Encrypting plaintext results in unreadable data called cipher text. This is the scrambled message produced as output. It depends on the plaintext and the secret key. Thus, encryption is used to protect the information in hidden from anyone for whom it is not projected, even those who can see the encrypted data. For a given message, two different keys will produce two different ciphertexts.
- *Decryption algorithm*: The process of reversing cipher text to its original plain text is called decryption. This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key, and produces the original plaintext (Figure 2).

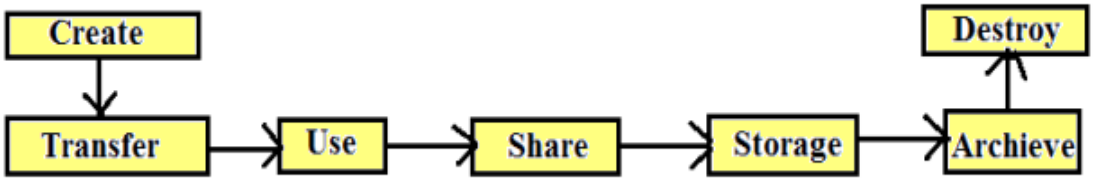

Figure 1. Life cycle of Data. A data security life cycle refers to the entire process from data creation and storage to destruction of data.

Encryption/Decryption process, in modern days is considered combination of three types of algorithms [10]. They are (i) Symmetric key algorithms where same key is used for encryption and decryption of data, (ii) Asymmetric key algorithms where a public key is used by sender to encrypt the data and a private key is used by receiver for decryption, and (iii) Hashing. Integrity of data is ensured by hashing algorithms. Thus, cryptography not only protects data from theft or alteration, but can also be used for user authentication. Cryptographic algorithms play a major role for data user security [11, 12]. As the complexity of algorithm is high, the risk of breaking the original plaintext from that of cipher text is less.

III. BASIC CONCEPTS AND PERFORMANCE OF ALGORITHMS

In today's information age, communication plays an important role in the growth of advance technologies. Therefore, privacy is needed to assure the security that is sent over communication media. Khalifa et al. [13] discussed basic concepts, characteristics and goals of various cryptography algorithms in communication. Nadeem and Javed [14] compared the performance of four secret key algorithms (DES, 3DES, AES, Blowfish) by encrypting input files of various contents and sized on different hardware program. The algorithms have been implemented in a regular language, using their standard qualifications, to allow a fair evaluation of execution speeds. Pentium-II having frequency 266 MHz and Pentium-IV with 2.4 MHz machine (running Windows XP OS) are the basis for time measurement with their goal to measure the encryption times of considered algorithms. Choi and Song [15] investigated various cryptographic algorithms suitable for wireless sensor network based on MICAz-type notes in which MD5 and RC4 showed best performance in terms of power dissipation and in terms of cryptographic processing time used.

Mellu and Mali [16] presented the fundamental mathematics behind the advanced encryption standard (AES) algorithm along with a brief description of some cryptographic primitives that are commonly used in the field of communication security. Since AES provides better security and has less implementation complexity, it has emerged as one of the strongest and most efficient algorithms. It also includes several cyber issues such as development of ciphertext as well as the analysis of AES security aspects against different kinds of attacks, the counter measures against these attacks and also highlighted some of the important security issues of AES algorithm.

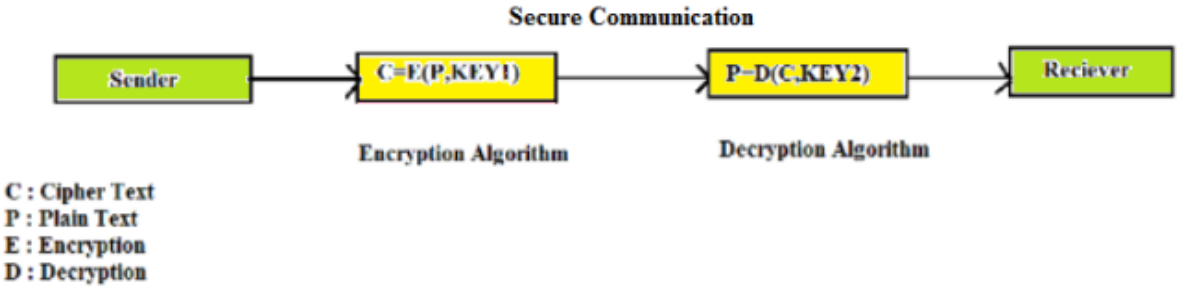

Figure 2. Use of Encryption and Decryption algorithms to obtain secure communication.

The original data/message, before being transformed is called plaintext. Plaintext is transformed into cipher text during encryption process and the cipher text is converted back into plaintext during decryption. The sender uses an encryption algorithm and the receiver uses a decryption algorithm. Thus, encryption and decryption help to secure transmission of the message to the authorized users.

Vijayalakshmi et al. [17] reported that cognitive radio networks are intelligent networks that can sense the environment and adapt the communication parameters accordingly. These networks find their applications in co-existence of different wireless networks, interference mitigation and dynamic spectrum access. Cognitive radio networks have their own set of unique security threats and challenges, such as selfish misbehaviours, self-coexistence, license user emulation and attacks on spectrum managers. Accordingly, the security protocols developed for these networks must have abilities to counter these attacks. The cognitive authentication protocol, called CoG-Auth design employs key hierarchy such as temporary keys, partial keys and session keys to fulfil the fundamental requirements of security. In this authentication, a hybrid encryption algorithm using AES and RSA was implemented and authors achieved less computational intensive, high performance, more secure and successful authentication and transmission rate. Public key cryptography (PKC) has also been employed to implement security in Cognitive Radio (CR) networks.

IV. CRYPTOGRAPHIC FUNCTIONALITIES AND EVALUATION

The properties mandatory to provide web masters and users with a mutual secure and practical authentication have been identified. The background information about the ongoing advances of browser-side cryptographic functionalities is described.

4.1. Browsers cryptographic libraries: To support the HTTPS protocol, all modern browsers provide support to some cryptographic operations (e.g. generating the client random certificate and then verify message in the Handshake phase of SSL/TLS protocol [18]). For example, one of the main cryptographic libraries is Network Security Services [19] which is a set of open source libraries designed to support cross-platform development of security-entitled applications.

4.2. Crypto API: W3C has created the Web Cryptography Working Group to develop a recommendation-track document. It defines an API that lets developers implement secure application protocols on the level of Web applications, including message privacy and authentication services, by exposing trusted cryptographic primitives from the browser.

*4.3. JavaScript cryptography***:** The use of JavaScript cryptography is a controversial issue in cryptography. Some authors [20] strongly argue that it is totally dangerous to use JavaScript cryptography inside the browser. However, other authors [21, 22] argue that claims such as JavaScript crypto is very bad for the improvement of security.

*4.4. Certificate and password managers***:** The five most popular browsers (Firefox, Chrome, Internet Explorer, Safari and Opera) provide certificate organization services. Using this built-in functionality, users can display information about the installed certificate including personal and authority certificates that the browser trusts [23] and perform all the important certificate management actions (import, export, delete). Keeping in view, the previous proposition boundaries and the ongoing advances in browser-side functionalities, properties were identified, which are required to provide web masters and users with a common secure and practical web user authentication. Browser may be built on a mechanism that solves password security weaknesses. User authentication qualifications should be stored securely and even with a database compromise.

V. Classification of Cryptographic Algorithms

There are several ways of classifying cryptographic algorithms (Figure 3). These algorithms can be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use [24, 25]. Cryptographic algorithms can be implemented in either hardware (for speed) or in software (for flexibility). Transpositions and substitutions can be implemented with simple electrical circuits. A device, known as a P-box (P stands for permutation), is used to effect a transposition on an 8-bit input. If the 8 bits are designated from top to bottom as 01234567, the output of this particular P-box is 36071245. By appropriate internal wiring, a P-box can be made to perform any transposition and do it at practically the speed of light since no computation is involved, just signal propagation.

Substitutions are performed by S-boxes. In this example, a 3-bit plaintext is entered and a 3-bit ciphertext is output. The 3-bit input selects one of the eight lines exiting from the first stage and sets it to 1; all the other lines are 0. The second stage is a P-box. The third stage encodes the selected input line in binary again. If the eight octal numbers 01234567 were input one after another, the output sequence would be 24506713. In other words, 0 has been replaced by 2, 1 has been replaced by 4 etc. Again, by appropriate wiring of the P-box inside the S-box, any substitution can be accomplished. Furthermore, such a device can be built in hardware and can achieve great speed since encoders and decoders have only one or two (subnano second) gate delays and the propagation time across the P-box may well be less than 1 pico second. By inclusion of a sufficiently large number of stages in the product cipher, the output can be made to be an exceedingly complicated function of the input.

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 cipher text when using the same key in a block cipher whereas the same plaintext will encrypt to different cipher text in a stream cipher.

VI. Key Based Different Encryption and Decryption Algorithms

The three types of algorithms using key for encryption and decryption are as follows:

- Symmetric (Secret) Key Cryptography (SKC): It uses a single key for both encryption and decryption
- Asymmetric (Public) Key Cryptography (PKC): It uses one key for encryption and another for decryption
- Hash Functions: It uses a mathematical transformation to irreversibly "encrypt" information

6.1 Symmetric (secret) Key Cryptography

This cryptographic method uses two different algorithms for encryption and decryption respectively, and a same key is used both for the sender and the receiver. The sender uses this key and an encryption algorithm to encrypt data, the receiver uses the same key and the corresponding decryption algorithm to decrypt that data [26, 27]. Because a single key is used for both functions, secret key cryptography is also called symmetric encryption. With this form of cryptography, 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. The description of some widely used symmetric key cryptographic algorithms are given below:

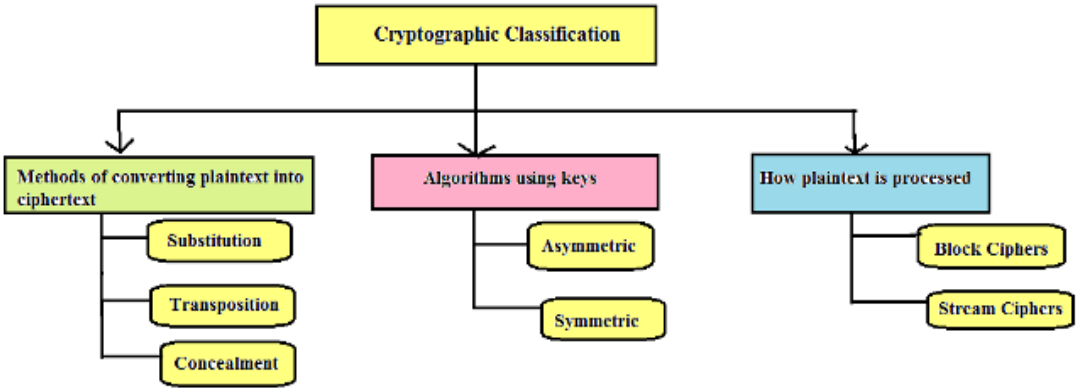

Figure 3. Classification of algorithms based on keys and applications

6.1.1. Advanced Encryption Standard (AES)

In 1997, NIST (National Institute of Standards and Technology) initiated a process to develop a new secure cryptosystem for U.S. government applications. NIST initially selected Rijndael in October 2000 and formal adoption as the AES standard came in December 2001. AES uses an SKC scheme called Rijndael, a block cipher designed by Belgian cryptographers Joan Daemen and Vincent Rijmen. The algorithm can use a variable block length and key length; the latest specification allowed any combination of keys lengths of 128, 192 or 256 bits and blocks of length 128, 192 or 256 bits [28, 29]. FIPS PUB 197 describes a 128-bit block cipher employing a 128-, 192- or 256-bit key. It encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size [30, 31].

6.1.2. Data Encryption Standard (DES)

The most common SKC scheme used today i.e., DES was designed by IBM in the 1970s and adopted by the National Bureau of Standards (NBS) [now the National Institute for Standards and Technology] in 1977 for commercial and unclassified government applications. These days, the DES algorithm is the most broadly used encryption algorithm in the world. The same algorithm and key are used for encryption and decryption, with minor differences. DES accepts an input of 64-bit long plaintext and 56-bitkey (8 bits of parity) and produce output of 64 bit block [32]. DES has a complex set of rules and transformations that were designed specifically to yield fast hardware implementations and slow software implementations, although this latter point is becoming less significant today since the speed of computer processors is several orders of magnitude faster today than twenty years ago. IBM also proposed a 112-bit key for DES, which was rejected at that time by the government, but the use of 112-bit keys was considered in the 1990s. DES was defined in American National Standard X3.92 and three Federal Information Processing Standards (FIPS), all withdrawn in 2005.

Two important variants that strengthen DES are:

- *Triple-DES (3DES):* A variant of DES that employs up to three 56-bit keys and makes three encryption/decryption passes over the block and is the recommended replacement to DES. Triple Data Encryption Algorithm (TDEA or Triple DEA) is a symmetric-key block cipher standard which is similar to DES method but increase encryption level 3 times than DES. As a result, this is slower than other block cipher methods. The block size of 3DES is 64 bit with 192 bits key size [30, 33].
- *DESX:* A variant was devised by Ron Rivest. By combining 64 additional key bits to the plaintext prior to encryption, it effectively increases the key length to 120 bits.

6.1.3. International Data Encryption Algorithm (IDEA)

Secret key cryptosystem was written by Xuejia Lai and James Massey in 1992 and was patented by Ascom. It consisted of a 64-bit SKC block cipher using a 128-bit key. Rivest Ciphers (aka Ron's Code) was named for Ron Rivest and it includes a series of SKC algorithms.

RC1: It was designed on paper but never implemented.

RC2: A 64-bit block cipher was designed using variable-sized keys to replace DES. It's code has not been made public although many companies have licensed RC2 for use in their products.

RC3: It is found to be breakable during development.

RC4: The RC4 (Rivest Cipher 4) is an encryption algorithm designed in 1987 by Ron Rivest for RSA security. It is a variable key-size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation. It is a shared key stream cipher algorithm requiring a secure exchange of a shared key [34, 35]. The RC4 encryption algorithm is used by standards such as IEEE 802.11 within WEP (Wireless Encryption Protocol) using 40 and 128-bit keys. It is widely used in commercial cryptography products. An update to RC4, called Spritz was designed by Rivest and Jacob Schuldt.

Eight to sixteen machine operations are required per output byte and the cipher can be expected to run very quickly in software. RC4 is used in the SSL/TLS (Secure Sockets Layer/Transport Layer Security) standards that have been defined for communication between Web browsers and servers. It is also used in the WEP (Wired Equivalent Privacy) protocol and the newer WiFi Protected Access (WPA) protocol that are part of the IEEE 802.11 wireless LAN standard. RC4 was kept as a trade secret by RSA security. In September 1994, the RC4 algorithm was anonymously posted on the Internet on the cypherpunks anonymous remailers list. A variable-length key of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S, with elements S[0], S[1],..., S[255]. At all times, S contains a permutation of all 8-bit numbers from 0 through 255. For encryption and decryption, a byte *k* is generated from S by selecting one of the 255 entries in a systematic fashion. As each value of *k* is generated, the entries in S are once again permuted.

To generate the key stream, the cipher makes use of a secret internal state which consists of two parts [36]:

1. A permutation of all 256 possible bytes (denoted "S").

2. Two 8-bit index-pointers (denoted "i" and "j").
The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the key-scheduling algorithm (KSA).

RC5: A block-cipher supporting a variety of block sizes (32, 64 or 128 bits), key sizes and number of encryption passes over the data.

RC6: A 128-bit block cipher based upon the improvement over RC5; RC6 was one of the AES Round 2 algorithms.

6.1.4. Blowfish

A symmetric 64-bit block cipher invented by Bruce Schneier and 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. It is in use in a large number of products. Blowfish is better than other algorithms in throughput and power consumption [37, 38].

6.1.5. SEED

A block cipher uses 128-bit blocks and 128-bit keys. It was developed by the Korea Information Security Agency (KISA) and adopted as a national standard encryption algorithm in South Korea.

6.1.6. ARIA

A 128-bit block cipher employs 128-, 192- and 256-bit keys. It was developed by large group of researchers from academic institutions, research institutes and federal agencies in South Korea in 2003 and subsequently named a national standard.

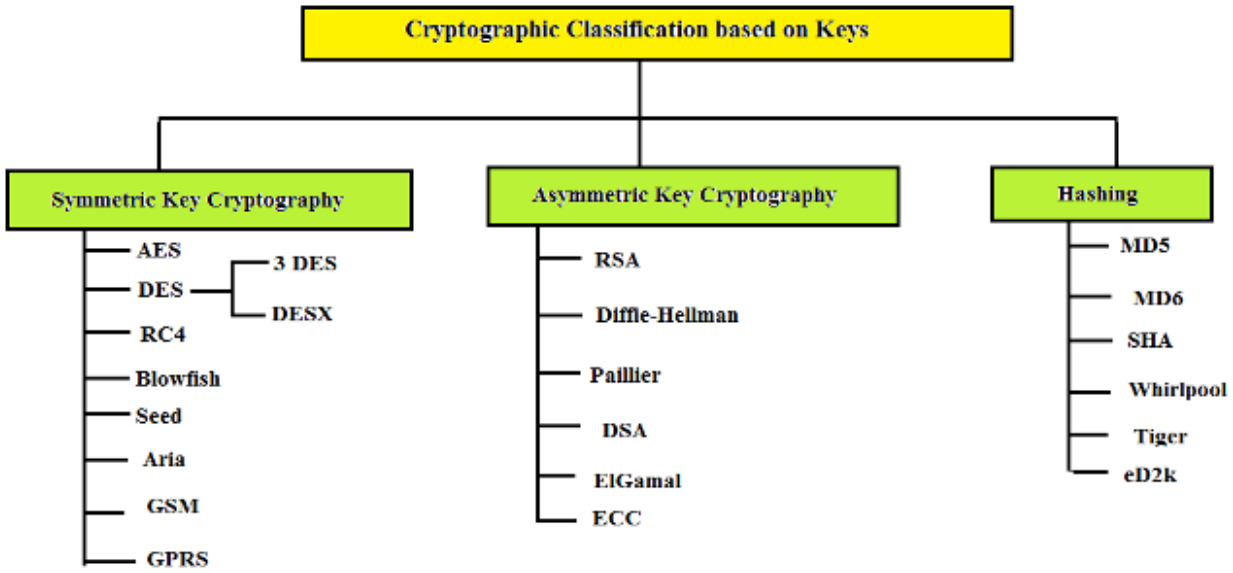

Figure 4. Schematic representation of some widely used Symmetric, Asymmetric and Hashing key cryptographic algorithms

6.1.6. GSM (Global System for Mobile Communications, originally Groupe Spécial Mobile) Encryption

GSM mobile phone systems use several stream ciphers for over-the-air communication privacy. A5/1 was developed in 1987 for use in Europe and the U.S. A5/2, developed in 1989, was a weaker algorithm and intended for use outside of Europe and the U.S. Significant flaws were found in both ciphers after the "secret" specifications were leaked in 1994 and A5/2 has been withdrawn from use. The newest version, A5/3, employs the KASUMI block cipher. Unfortunately, A5/1 has been repeatedly "broken" but this encryption scheme remains in widespread use, even in 3G and 4G mobile phone networks. Use of this scheme is reportedly one of the reasons that the National Security Agency (NSA) can easily decode voice and data calls over mobile phone networks.

6.1.7. GPRS (General Packet Radio Service) Encryption

GSM mobile phone systems use GPRS for data applications and GPRS uses a number of encryption methods, offering different levels of data protection. GEA/0 offers no encryption at all. GEA/1 and GEA/2 are proprietary

stream ciphers, employing a 64-bit key and a 96-bit or 128-bit state, respectively. GEA/1 and GEA/2 are most widely used by network service providers today, although both have been reportedly broken. GEA/3 is a 128-bit block cipher employing a 64-bit key that is used by some carriers. GEA/4 is a 128-bit clock cipher with a 128-bit key, but is not yet deployed.

6.2. Asymmetric (public) Key Cryptography

Public-key cryptography (PKC) has been said to be the most significant new development in cryptography in the last 300-400 years. Modern PKC was first described publicly by Stanford University professor Martin Hellman and graduate student Whitfield Diffie in 1976. Their paper described a two-key crypto system in which two parties could engage in a secure communication over a non-secure communications channel without having to share a secret key. This cryptographic method makes use of two different algorithms for encryption and decryption respectively, a public key for encryption and a private key for decryption. The public key of the sender is used to encrypt the message by the sender. The receiver decrypts the cipher text with the help of a private key. The descriptions of some widely used asymmetric key cryptographic algorithms are given below:

6.2.1. RSA

RSA is the first and still most common, PKC implementation, named for the three MIT mathematicians, who developed it - Ronald Rivest, Adi Shamir and Leonard Adleman. RSA is broadly used asymmetric encryption/decryption algorithm, which involves a public key and a private key (Figure 5). The public key can be informed to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. It secured user data assimilate encryption before the storage, user authentication procedures prior to storage or retrieval and making secure channels for data transmission [39, 40]. 4096 bit key size is used for execution of RSA algorithm. RSA algorithm involves these steps: 1. Key generation, 2. Encryption and 3. Decryption

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. The public key information includes *n* and a derivative of one of the factors of *n*; an attacker cannot determine the prime factors of *n* (and, therefore, the private key) from this information alone and that is what makes the RSA algorithm so secure.

Some descriptions of PKC erroneously state that RSA's safety is due to the difficulty in factoring large prime numbers. In fact, large prime numbers like small prime numbers, only have two factors. The ability for computers to factor large numbers and therefore, attack schemes such as RSA, is rapidly improving and systems today can find the prime factors of numbers with more than 200 digits. Nevertheless, if a large number is created from two prime factors that are roughly the same size, there is no known factorization algorithm that will solve the problem in a reasonable amount of time i.e., a 2005 test to factor a 200-digit number took 1.5 years and over 50 years of compute time. Regardless, one presumed protection of RSA is that users can easily increase the key size to always stay ahead of the computer processing curve. The patent for RSA expired in September 2000, which does not appear to have affected RSA's popularity.

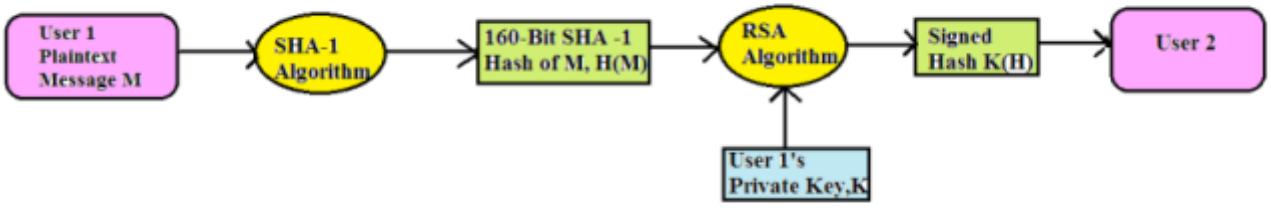

Figure 5. Use of SHA-1 and RSA algorithms for signing nonsecret messages

6.2.2. Diffie-Hellman

After the RSA algorithm was published, this algorithm was first revealed by Whitfield Diffie and Martin Hellman in 1976. D-H is used for secret-key key exchange only and not for authentication or digital signatures. Diffie–Hellman key exchange is a specific method of exchanging cryptographic keys [36]. It permits two parties that have no prior

knowledge of each other to jointly make a shared secret key over an insecure communications channel. This key can then be used to encrypt posterior communications using a symmetric key cipher.

6.2.3. Paillier

The Paillier cryptosystem is an asymmetric algorithm. It has homomorphic property that permits this scheme to do normal addition operations on several encrypted values and achieving the encrypted sum. The encrypted sum can be decrypted later without even knowing the values ever that made up the sum.

6.2.4. Digital Signature Algorithm (DSA)

The algorithm specified in NIST's Digital Signature Standard (DSS), provides digital signature capability for the authentication of messages.

6.2.5. ElGamal

It was designed by Taher Elgamal, a PKC system similar to Diffie-Hellman and used for key exchange.

6.2.6. Elliptic Curve Cryptography (ECC)

PKC algorithm is based upon elliptic curves. ECC can offer levels of security with small keys comparable to RSA and other PKC methods. It was designed for devices with limited computing power and/or memory, such as smart cards and PDAs.

6.3. Hashing Cryptography

Hash functions are fundamental in the field of cryptography and used widely in a broad spectrum of important applications involving message integrity and authentication [41, 42], digital signatures [43], secure time stamping and countless others. Hash functions, also called message digests and one-way encryption, are algorithms that use no key. Instead, a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered. Hash algorithms are typically used to provide a digital fingerprint of a file's contents, often used to ensure that the file has not been altered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. Hash functions, then, provide a measure of the integrity of a file. A hash function H is an efficiently-computable algorithm that takes as input an arbitrary-length message M and potentially a fixed-length key K (considering a keyed hash function), and makes a fixed-length output D called the message digest. H (K, M) = D. The description of some widely used Hashing cryptography algorithms are given below:

6.3.1. Message Digest (MD) algorithms

MD algorithms includes a series of byte-oriented algorithms that produce a 128-bit hash value from an arbitrary-length message.

MD2 (RFC 1319): It was designed for systems with limited memory, such as smart cards. MD2 has been relegated to historical status as per RFC 6149.

MD4 (RFC 1320): It was developed by Rivest, similar to MD2 but designed specifically for fast processing in software. MD4 has been relegated to historical status as per RFC 6150.

MD5 (RFC 1321): MD5 (Message Digest5) is a broadly used cryptographic hash function with a 128-bit hash value. It processes a variable-size message into a fixed-length output of 128 bits [28]. The input message is divided into chunks of 512-bit blocks; then the message is padded for making its length divisible by 512 [44]. In this algorithm, sender use the public key of the receiver to encrypt the message and receiver use its private key to decrypt the message. It was also developed by Rivest after potential weaknesses were reported in MD4; this scheme is similar to MD4 but is slower because more manipulation is made to the original data. MD5 has been implemented in a large number of products although several weaknesses in the algorithm were demonstrated by German cryptographer Hans Dobbertin in 1996.

MD6: The MD6 Message-Digest algorithm makes use of a substantially different tree-based mode of operation that allows for greater parallelism [45]. MD6 may be viewed as a tree-like construction, with a 4-to-1 compression function reducing the overall length of the message at each level [46].

6.3.2. Secure Hash Algorithm (SHA)

Algorithm for NIST's Secure Hash Standard (SHS) was described in FIPS 180-4.

(i) SHA-1 produces a 160-bit hash value and was originally published as FIPS PUB 180-1 and RFC 3174. It was deprecated by NIST as of the end of 2013, although it is still widely used (Figure 5). In October 2015, the SHA-1 Freestart Collision was announced.

(ii) SHA-2: It was originally described in FIPS PUB 180-2 and eventually replaced by FIPS PUB 180-3 (and FIPS PUB 180-4). It comprises five algorithms in the SHS: SHA-1 plus SHA-224, SHA-256, SHA-384 and SHA-512 which can produce hash values that are 224, 256, 384 or 512 bits in length, respectively. The newer and stronger SHA-2 hash function is currently used in a wide variety of applications, including TLS, SSL, SSH and PGP. SHA1 outputs a 160-bit digest of any sized file or input. SHA-256 algorithm produces an almost-unique, fixed size 256-bit (32-byte) hash. This creates it suitable for password validation, challenge hash authentication, anti-tamper and digital signatures. SHA-256 is one of the successor hash functions to SHA-1 and is one of the strongest hash functions available. SHA-256 hash functions is computed with 32-bit words.

SHA-2 recommends use of SHA-1, SHA-224 and SHA-256 for messages less than 264 bits in length and employs a 512 bit block size. SHA-384 and SHA-512 are recommended for messages less than 2128 bits in length and employs a 1,024 bit block size. FIPS PUB 180-4 also introduces the concept of a truncated hash in SHA-512/*t*, a generic name referring to a hash value based upon the SHA-512 algorithm that has been truncated to *t* bits.

(iii) SHA-3 is the current SHS algorithm. Although, there had not been any successful attacks on SHA-2, NIST decided that having an alternative to SHA-2 using a different algorithm would be prudent. In 2007, a SHA-3 competition was launched to find that alternative and a list of submissions can be found at the SHA-3 Zoo. In 2012, NIST announced that after reviewing 64 submissions, the winner was KECCAK (pronounced "catch-ack"), a family of hash algorithms based upon sponge functions. The NIST version can support hash output sizes of 256 and 512 bits.

6.3.3. Whirlpool

This algorithm was designed by V. Rijmen (co-inventor of Rijndael) and P.S.L.M. Barreto. Whirlpool is one of two hash functions endorsed by the New European Schemes for Signatures, Integrity and Encryption (NESSIE) competition (the other being SHA). Whirlpool operates on messages less than 2256 bits in length and produces a message digest of 512 bits. The design of this hash function is very different than that of MD5 and SHA-1, making it immune to the same attacks as on other hashes.

6.3.4. Tiger

It was designed by Ross Anderson and Eli Biham. Tiger is designed to be secure, run efficiently on 64-bit processors and easily replace MD4, MD5, SHA and SHA-1 in other applications. Tiger/192 produces a 192-bit output and is compatible with 64-bit architectures, whereas Tiger/128 and Tiger/160 produce a hash of length 128 and 160 bits, respectively, to provide compatibility with the other hash functions.

6.3.5. eD2k

It was named for the EDonkey2000 Network (eD2K). The eD2k hash is a root hash of an MD4 hash list of a given file. A root hash is used on peer-to-peer file transfer networks, where a file is broken into chunks. Each chunk has its own MD4 hash associated with it and the server maintains a file that contains the hash list of all of the chunks. The root hash is the hash of the hash list file.

VII. CONCLUSION AND PERSPECTIVES

Cryptography is a particularly interesting field because of the amount of work done in secrecy. Regardless of the mathematical theory and statistical tools behind an algorithm, the best algorithms are those that are well-known, well documented and are also well-tested [47]. In fact, time is the only true test of good cryptography and any cryptographic scheme that stays in use year after year is most likely a good one. The strength of cryptography lies in the choice (and management) of the keys, where longer keys will resist attack better than shorter keys. Encryption algorithms keep very important contribution in security of data in cloud environments [48], web technology and communication security. The performance of widely used encryption techniques like AES, DES and RSA algorithms has been compared. Based on the text files used, AES algorithm was reported to consume least encryption and RSA consumed longest encryption time.

REFERENCES

[1] Smith, L.D., "Cryptography: The science of secret writing", New York: Dover Publications. (1943).
[2] Ferguson, N., Schneier, B. and Kohno, T., "Cryptography engineering: Design principles and practical applications. New York: John Wiley & Sons (2010).
[3] Basin, D., Cremers, C., Miyazaki, K., Radomirovic, S. and Watanabe, D., "Improving the security of cryptographic protocol standards", IEEE Security and Privacy 13(3), 24-31 (2015).
[4] Hossain, M.A., Hossain, M.B. Uddin, M.S. and Imitiaz, S.M., "Performance analysis of different cryptographic algorithms", International Journal of Advanced Research in Computer Science and Software Engineering 6(3), 659-665 (2016).
[5] Jangala, Anusha, S.K.M., Vijaykumar, A. and Kavya, M., "Cryptography: The science of secure communication", International Journal of Computer Science and Network Security, 16(4), 129-134 (2016).
[6] Ryan, M.D., "Cloud computing security: The scientific challenge and a survey of solutions", The Journal of Systems and Software 86, 2263-2268 (2013).
[7] Denning, D.E., "Cryptography and data security". Reading, MA ed.: Addison-Wesley (1982).
[8] Krombholz, K., Hobel, H., Huber, M. and Weippl, E., "Advanced social engineering attacks", Journal of Information Security and Applications 22, 113-122 (2015).
[9] Zeng, W., Koutny, M., Watson, P. and Germanos, V., "Formal verification of secure information flow in cloud computing", Journal of Information Security and Applications 27-28, 103-116 (2016).
[10] Bhardwaj, A., Subrahmanyan, G.V.B., Avasti, B. and Sastry, H., "Security algorithms for cloud computing", Procedia Computer Science 85, 535-542 (2016).
[11] Stallings, W., "Cryptography and network security: Principles and practice, 4th ed. Englewood Cliffs, NJ; Prentice Hall (2006).
[12] Potey, M.M., Dhote, C.A. and Sharma, D.H., "Homomorpic encryption for security of cloud data", Procedia Computer Science 79, 175-181 (2016).
[13] Khalifa, O.O., Islam, M.D.R., Khan, S. and Shebani, M.S., "Communication cryptography", RF and Microwave Conference, Subang, Selangor, Malaysia (2004).
[14] Nadeem, A. and Javed M.Y., "A performance comparison of data Encryption Algorithm", Global Telecommunication Workshops, Globe Com Workshops 2004, IEEE. (2004).
[15] Choi, K.J. and Song, J., "Investigation of feasible cryptographic algorithm for wireless sensor network", International conference on ICACT, Feb 20-22, 2006, (2006).
[16] Mellu, P. and Mali, S., "AES: Asymmetric key cryptographic System", International Journal of Information Technology and Knowledge Management, 4, 113-117 (2011).
[17] Vijayalakshmi, Ch., Lavanya, L. and Navya, Ch., "A hybrid encryption algorithm based on AES and RSA. International Journal of Innovative Research in Computer and Communication Engineering, 4(1), 909-917 (2016). DOI: 10.15680/IJIRCCE.2016. 0401064 909.
[18] http://tools.ietf.org/html/rfc5246 IETF, "RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2.":
[19] https://developer.mozilla.org/en-US/docs/Overview of_NSS. Mozilla, Overview "of NSS MDN:
[20] http://www.matasano.com/articles/javascript-cryptography/.Matasano, "Javascript Cryptography Considered Harmful," 2011:
[21] http://hellais.wordpress.com/2011/12/27/how-to improve-java script-cryptography/. "How to improve JavaScript cryptography.":
[22] http://log.nadim.cc/?p=33. N. Kobeissi, "Thoughts on Critiques of JavaScriptCryptography.":
[23] Yassin, A.A., Jin, H., Ibrahim, A., Qiang, W. and Zou, D., "Efficient pass word based two factors authentification in cloud computing", International Journal of Security and its Applications, 6(2), 143-148 (2012).
[24] Rachana, S.C. and Guruprasad, H.S., "Emerging security issues and challenges in cloud computing", International Journal of Engineering Science and Innovative Technology, ISSN: 2319-5967, 3(2) 485-490 (2014).
[25] Khan, S.S. and Tuteja, R.R., "Security in cloud computing using cryptographic algorithms", International Journal of Innovative Research in Compute and Communication Engineering, 3(1), 148-154 (2015).
[26] Arora, R. and Parashar, A., "Secure user data in cloud computing using encryption algorithms", International Journal of Engineering Research and Applications, ISSN: 2248-9622, 3(4), 1922-1926 (2013).
[27] Jasim, O.K., Abbas, S., El-Horbaty El.S.M. and Salem, A.B.M., "Efficiency of modern encryption algorithms in cloud computing", International Journal of Emerging Trends and Technology in Computer Science, ISSN 2278-2286, 2(6), (2013).
[28] Singh, V.K. and Dutta, M., "Analyzing cryptographic algorithms for secure cloud network" International Journal of Advanced Studies in Computer Science and Engineering, 3(6), 1-9 (2014).
[29] Mahajan, P. and Sachdeva, A., "A study of encryption algorithms AES, DES and RSA for security", Global Journal of Computer Science and Technology Network, Web and Security, ISSN: 0975-4350, 13, 15-22 2058-2064 (2013).
[30] Singh, G. and Kinger, S., "Integrating AES, DES, and 3-DES encryption algorithms for enhanced data security", International Journal of Scientific and Engineering Research, 4(7), (2013).
[31] Kaur, G. and Mahajan, M., "Analyzing data security for cloud computing using cryptographic algorithms", International Journal of Engineering Research and Applications, ISSN : 2248-9622, 3(5), 782-786 (2013).
[32] Prashanti. G, Deepthi. S. and Sandhya Rani. K., "A novel approach for data encryption standard algorithm", International Journal of Engineering and Advanced Technology, ISSN: 2249- 8958, 2(5), 264-269 (2013).
[33] Wankhade, N.M., Sahare K.A. and Bhujade, V.G., "Secure cloud simulation using triple DES", International Journal of Research in Advent Technology, 2(1), (2014).
[34] Kumar, Y., Munjal, R. and Sharma, H., "Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures", International Journal of Computer Science and Management Studies, 11(3), (2011).
[35] Ahmad, I. and Khandekar, A., "Homomorphic encryption method applied to cloud computing", International Journal of Information and Computation Technology, ISSN 0974-2239, 4(15), 1519-1530 (2014).
[36] Vijay. G.R. and Rama Mohan Reddy, A., "Data security in cloud based on trusted computing environment", International Journal of Soft Computing and Engineering, ISSN: 2231-2307, 3(1), (2013).
[37] Singh, G., Singla, A. and Sandha, K.S., "Cryptography algorithm comparison for security enhancement in wireless intrusion detection system", International Journal of Multidisciplinary Research,1(4), 143-151 (2011).

[38] Kaur, M., Singh, R., "Implementing encryption algorithms to enhance data security of cloud in cloud computing", International Journal of Computer Applications, ISSN 0975 – 8887, 70(18), 16-21 (2013).
[39] Dharini, A., Devi, R.M. and Chandrasekar, I., "Data security for cloud computing using RSA with magic square algorithm", International Journal of Innovation and Scientific Research, ISSN 2351-8014, 11(2), 439-444 (2014).
[40] Parsi, K. and Sudha, S., "Data Security in cloud computing using RSA algorithm", International Journal of Research in Computer and Communication technology, ISSN 2278-5841, 1(4), 145-152 (2012).
[41] Bellare, M., Canetti, R. and Krawczyk, H., "The HMAC construction", RSA laboratories CryptoBytes, 2(1), 12–15 (1996).
[42] Bellare, M., Canetti, R. and Krawczyk, H., "Keying Hash functions for message authentication", In: Neal Koblitz, ed., CRYPTO, volume 1109 of Lecture Notes in Computer Science, 1–15, (1996).
[43] Bellare, M. and Rogaway, P., "The exact security of digital signatures - how to sign with RSA and Rabin", In: Advances in Cryptology, Maurer, U. (ed.), pp. 399-416 (1996).
[44] Arora, P., Singh, A. and Tyagi, H., "Evaluation and comparison of security issues on cloud computing environment", World Journal of computer Science and Information Technology, ISSN: 2221-0741, 2(5), 179-183 (2012).
[45] Ronald L. Rivest, "The MD6 Hash function" (2008).
[46] Crutchfield, C.Y., "Security proofs for the MD6 Hash function mode of operation", Massachusetts Institute of Technology (2008).
[47] Simion, E., "The relevance of statistical tests in cryptography", IEEE Security and Privacy, 13(1), 66-70 (2015).
[48] Atayero, A.A. and Feyisetan, O., "Security issues in cloud computing: The potentials of homomorphic encryption", Journal of Emerging Trends in Computing and Information Services, 2(10), 546-552 (2011).