Journal of Network and Information Security
7 (2), December 2019, 28-34
http://www.publishingindia.com/jnis

Modern Cryptographic Schemes: Applications and Comparative Study

Julius O. Olwenyi[1], Aby Tino Thomas[2] and Ayad Barsoum[3*]
[1]St. Mary's University, San Antonio, TX, USA. Email: jolwenyi@mail.stmarytx.edu
[2]St. Mary's University, San Antonio, TX, USA. Email: athomas15@stmarytx.edu
[3]St. Mary's University, San Antonio, TX, USA. Email: abarsoum@stmarytx.edu
*Corresponding Author

Abstract: Cryptography and encryption have been used for secure communication. In the modern world, cryptography is a very important tool for protecting information in computer systems. With the invention of the World Wide Web or Internet, computer systems are highly interconnected and accessible from any part of the world. As more systems get interconnected, more threat actors try to gain access to critical information stored on the network. It is the responsibility of data owners or organizations to keep this data securely and encryption is the main tool used to secure information. In this paper, we will focus on different techniques and the modern application of cryptography. We will study different cryptographic schemes: symmetric, asymmetric (sometimes referred to as public key), and hybrid systems. The paper will present a comparative study for these schemes and their applications in network protocols. Moreover, we will highlight the concept of Quantum Cryptography, which takes advantage of quantum physics at the physical layer.

Keywords: Cryptography, Data security, Decryption, Encryption, Hybrid encryption.

I. Introduction

Back in the days, cryptography was not all about hiding messages or secret communication, but in ancient Egypt, where it began; it was carved into the walls of tombs to portray sarcastic stories, mysteries, intrigue, or used to amuse the onlooker as they passed by those tombs. Atbash [1] was a Hebrew encryption method that simply flipped the alphabet as shown below:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ZYXWVUTSRQPONMLKJIHGFEDCBA

With Atbash encryption, the cipher text for the plain text "CRYPTOGRAPHY" will be "XIBKGLTIZKSB".

Later on in 400 BC, Spartans developed scytale cipher. They carefully wrote messages on papyrus and wrap around a wooden rod. The message will only be readable if it is wrapped around the right wooden rod size and wrapped correctly.

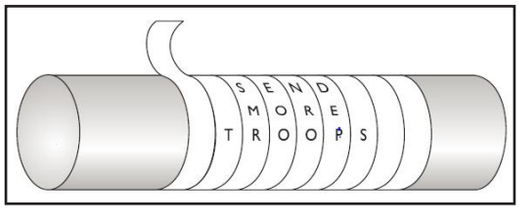

Fig. 1: The Scytale Cipher Used by Spartans

About 100 BC to 400 BC, Roman Emperor Julius Caesar developed a monolithic substitution method where a letter in plaintext is simply shifted three places down the alphabet [4, 5].

ABCDEFGHIJKLMNOPQRSTUVWXYZ

DEFGHIJKLMNOPQRSTUVWXYZABC

The ciphertext of the plaintext "CRYPTOGRAPHY" will be "FUBSWRJUASLB" in a Caesar cipher.

More recent derivative of Caesar cipher is Rot13 which shifts 13 places down the alphabet instead of 3. Rot13 was not all about data protection, but it was used on online forums where members could share inappropriate language or nasty jokes without necessarily being offensive as it will take those interested in those "jokes' to shift characters 13 spaces to read the message and if not interested you do not need to go through the hassle of converting the cipher.

In the 16th century, the French cryptographer Blaise de Vigenere [4, 5] developed the first polyalphabetic substitution basically based on Caesar cipher, but more difficult to crack the cipher text.

For a plaintext P and a key K, Vigenere computes cipher text C by [6]:

$C = (P+K)\ mod 26$

For plain "CRYPTOGRAPHY" with the key "KEY", the cipher text will be:

P	2	17	24	15	19	14	6	17	0	15	7	24
K	10	4	24	10	4	24	10	4	24	10	4	24
C	12	21	22	25	23	12	16	21	24	25	11	22

Fig. 2: A Simple Table Showing Vigenere Cipher

Note that each character in the plaintext and the key is substituted by an integer representing its position in the alphabet (i.e., A = 0, B = 1, C = 2, and so on). From Fig. 2 above, the ciphertext C is "MVWZXMQVYZLW".

Vernam Cipher (One-Time Pad) [7]: All of the above schemes can be exploited by cryptanalysis. One obvious way to attack mono-alphabetic schemes such as Caesar cipher is to use frequency analysis attacks. Attackers can analyze the most frequent letter in the ciphertext and deduce its corresponding plaintext letter. In English language text, for example, the letter 'e' appears at 12.702% relative frequency compared to other alphabets.

One-time pad developed by Gilbert Vernam is unbreakable if implemented properly. The key (pad) must be used only once and discarded and it must be of the same length or longer than the message. Moreover, the pad must be securely distributed to the destination and must be generated by true random numbers. Vernam Cipher involves exclusive-OR (XOR) operation. The plaintext that has to be encoded is converted into bits and then XOR-ed with keystream bits.

Plain Text: 10110011

Keystream: 01010101

Cipher Text: 11100110

II. Cryptanalysis

Cryptanalysis is the combination of procedures, processes, and methods used to translate or interpret secret writings or communication as codes and ciphers for which the key is unknown [8]. Normally, we consider cryptanalysis as exploring the weakness of the underlying mathematics of the cryptographic system or the implementation of that algorithm.

The primary goal of doing cryptanalysis can be one or more reasons: (i) total breakdown and finding the secret key, (ii) global deduction – finding a functionally equivalent algorithm for encryption and decryption that does not require knowledge of the secret key, (iii) information deduction – gaining some information about plaintexts or ciphertexts that was not previously known; and (iv) distinguishing algorithm – the attacker has the ability to distinguish the output of the encryption (ciphertext) from a random permutation of bits.

For any encryption algorithm, the only secret information is the key. The encryption algorithm and its implementation are shared with the external world so that more researchers can look deep into all aspects of that encryption system to identify any potential weaknesses. In the early stages of an encryption method, researchers might discover methods to fully decrypt the message without knowing the key. This helps system developers to improve the system to make it resilient to this kind of attacks.

III. Key Definitions and Concepts

Plaintext: The data or message to be sent, in a clear form, where anyone can read.

Ciphertext: The data in encrypted form.

Bit: Binary digit, the basic unit of information stored in a computer. Any letter or number can be encoded as a string of 8 bits.

Algorithm: The method used to encrypt and decrypt data.

Key: A crucial and secret parameter in the algorithm.

Initialization Vectors (IV): These are random values that are used together with encryption algorithms to ensure that two plaintext messages using the same key do not produce the same ciphertext.

Hash: A fingerprint for a digital file.

*Encryption***:** The process of encoding a message or information in such a way that only authorized parties can access it.

*Decryption***:** The reverse process of encryption. It is the process of decoding the data which has been encrypted into a secret format.

IV. Types of Data Encryption

Classification of the encryption techniques is mainly based on the number of keys that are involved in the cryptographic process [4, 5, 19]. Algorithms that use a single shared key for both encryption and decryption are called symmetric key cryptography, whereas algorithms that use two different keys (one for encryption and the other for decryption) are called asymmetric key cryptography.

A. Symmetric Key Cryptography

Symmetric encryption (also known as single key encryption, shared key encryption, or secret key cryptography) was the only type of encryption that was used until late 1970s. This type of algorithm, as the name suggests, uses a single key for both encryption and decoding of data. The symmetric key algorithm can either be block cipher (act on block of data of a specific size) or stream cipher (act on bit by bit or byte by byte) [4, 5]. Examples of block ciphers are: Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES) [9]. Block ciphers are mostly used for bulk data encryption because of its processing speed and the size of plaintext is always the same as ciphertext. Strong

cipher contains two main attributes: confusion and diffusion. Confusion simply means that it is impossible to deduce the key from the resulting ciphertext, whereas diffusion means that a single bit change in the plaintext will yield a completely different ciphertext. Stream cipher is mostly used with equipment that has low computing power such as GSM devices. Example of stream cipher algorithm is RC4 [10].

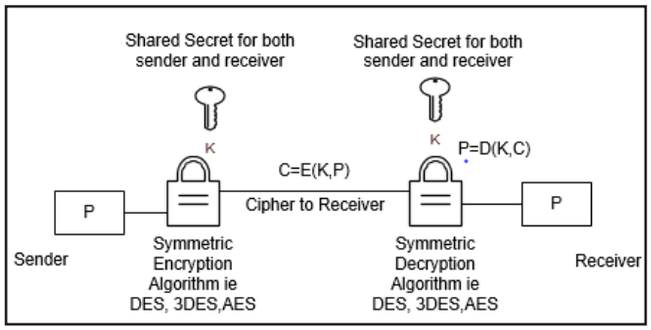

Fig. 3: Symmetric Encryption Algorithm

i) Data Encryption Standard (DES)

This is one of the oldest 'modern' encryption algorithms. It uses a 56 bit key and 64 bit block of plaintext [11]. It was developed in the 1970s based on Feistel algorithm. The blocks of data are put through 16 rounds of transposition and substitution with each round with a different 48 bit key. DES is symmetric key algorithm since the same key used for both encryption and decryption processes.

The biggest shortcoming of DES is the key size [11]. A brute force attack on DES only involves 2^{56} tries (about 7.2×10^{16} keys). With the advancement of modern computing power, the brute force attack on the DES algorithm can only be done in a couple of hours (if not minutes). For that reason, DES was dropped by NIST as the de facto standard for data encryption algorithm in the late 1970s. Another flavor of DES, Triple DES (3DES) was an improvement of DES, which basically applies the DES algorithm three times when encrypting data. Even though 3DES is a secure algorithm, it is not widely used because of its performance (i.e., resource intensive algorithm and very slow especially if implemented on software platforms).

ii) Advanced Encryption Standard (AES)

In 1997, National Institute of Standards and Technology (NIST) requested for the new algorithm that would overcome the limitations that DES had. Several cryptographers applied, and in 2001 Rijndael algorithm developed by two Belgium cryptographers, Vincent Rijmen and Joan Daemen was selected as the winning algorithm. Unlike DES which is based on Feistel structure, in AES, arithmetic operations (such as addition, multiplication, and division) are performed over a finite or Galois field *GF* (2^8) [12]. The algorithm uses 128 bits block of plain text and key length of 128, 192, or 256 bits. The algorithm can either be AES-128, AES-192, OR AES-256 depending on key size. A 4×4 square matrix (state) is fed unto the encryption or decryption algorithm. The key is also represented as a matrix. The AES algorithm consists of 10, 12, and 14 rounds of data transformation. AES-128 goes through 10 rounds, AES-192 goes through 12 rounds and AES-256 goes through 14 rounds. Each round has four different stages:

- Substitute Byte: This stage uses the S-box table to perform byte by byte substitution.
- Shift Rows: Row by row permutation.
- Mix Columns: Does arithmetic operation based on $GF(2^8)$.
- Add Round Key: Uses a simple XOR operation on current data block and part of the expanded key.

Unlike DES, it is practically impossible to attack the AES algorithm using brute force. For AES-128 which uses a key length of 128 bits, the total number of keys is $2^{128} \approx 3.4 \times 10^{38}$ keys, which will take about 5.3×10^{21} years to break using brute force attack. With AES-192 and AES-256, which use larger key sizes, brute forcing operation becomes even more complicated. At the time of this paper, there are no known attacks on AES algorithm, though unsafe block cipher mode of operations such as an electronic code book (ECB) mode can enable an attacker to deduce plaintext or the key from the ciphertext. In ECB, each block of plaintext is encoded independently. Two similar plain text messages encoded with the same key will produce a similar cipher message.

B. Asymmetric Key Cryptography

The modern symmetric algorithms such as AES are highly secure, but the major drawback of those algorithms is key distribution. Public key algorithms also called asymmetric key algorithms are based on complex mathematical functions rather than bit operations as in symmetric algorithms [4, 5]. Asymmetric key cryptography involves the use of two keys: private key and public key. The two keys are different but mathematically related to each other and it is computationally infeasible to derive the private key from the public key. Either key can be used in encryption or decryption algorithm. In fact, whatever one key does, the other key will reverse. Fig. 4 below gives a high level overview of asymmetric algorithm

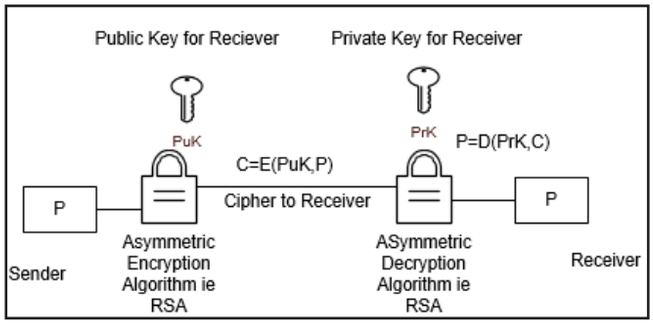

Fig. 4: Asymmetric Encryption Algorithm

The private key is kept secret and never shared with any communicating entity and the public key is published and every entity has access to it. Depending on what security objective (confidentiality or authentication), a user wants to achieve, we can either use public or private key to encode the data. Confidentiality can be achieved when using public key of the receiver to encrypt the data. It is only the receiver that has a corresponding key to decode the encrypted data.

To achieve data integrity or authentication, the private key can be used to encode the data, we will for sure know that the data was encrypted by a sender's private key because it is the sender's public key that can be used to decode the cipher message. The most widely used asymmetric algorithms are: RSA, Diffie-Hellman, El Gamal, and Elliptic curve.

i) RSA

Developed in 1977 by three cryptographers (Ron Rivest, Adi Shamir, and Len Adleman) [13], RSA is the most widely used public-key encryption. The algorithm is based on a large number factorization and the difficulty of getting the factors of very large prime numbers. The key length of the RSA algorithm is 1024, 2048 or 4096 bits [13]. The large key size makes the algorithm unbreakable. The following are the three main services provided by RSA:

*Encryption***:** The two keys involved in RSA (private key and public key) can be used for either encryption or decryption of data to be exchanged between the two communicating entities.

*Key Exchange***:** RSA is used for secure key exchange. The cryptosystems generate asymmetric key that can be used by either DES/AES algorithms. The symmetric key is encrypted by the receiver's public key and then sent to the receiver. The receiver then recovers the symmetric key by using his/her private key. The symmetric key is then fed into a symmetric algorithm.

*Digital Signature***:** RSA can be used to generate digital signatures [14]. The private key of the initiator is used as an input into DSA (Digital Signature Algorithm) and transforms data or its hash value. Then, the sender's public key will be used to verify the authenticity of the message.

ii) Diffie-Hellman Key Exchange

Diffie-Hellman (DH) [15] is not an encryption scheme, but a protocol or algorithm that enables two communicating entities to calculate and have a shared key then use that key in symmetric algorithms such as AES. Diffie-Hellman is based on discrete logarithms. The major drawback of DH is man-in-the-middle attack. An attacker can 'sit' between two legitimate communicating hosts and successful calculate the shared key with each host. The attacker will be able to read/modify the messages and resend to them without being noticed.

iii) El Gamal Cryptography

The El Gamal algorithm [4, 5, 20] is a public key scheme that is based on the difficulty of calculating discrete logarithms in a finite field. The algorithm is used in digital signature, data encryption and exchanging keys. This algorithm is based on DH key exchange but it does provide more services. El Gamal cryptography is not widely used because of its performance compared to other asymmetric algorithm (El Gamal is usually slower).

iv) Elliptic Curve Cryptography

RSA is of no doubt the 'enigma' in the public key cryptography. The majority of products that have implemented asymmetric key encryption system use RSA. Standards such as elliptic curve cryptography (ECC) [16] are recently gaining popularity because they can provide as much security as RSA but with smaller keys. To achieve a high security index with RSA (like some other public encryption schemes) the used keys have to be of large size. With key size as large as 4096 bits, the overhead processing for an application that uses RSA is greatly increased. This can cause slow transaction especially with ecommerce sites that needs to perform faster and secure transactions. ECC will achieve the same security level as RSA with much smaller key size. Table I compares the key sizes of ECC with RSA to achieve the same security level.

Table I: ECC vs RSA Key Sizes

RSA (Size of Key in Bits)	ECC (Size of Key in Bits)
1024	160 – 223
2048	224 – 255
3072	256 – 383
7680	384 – 511
15360	512+

V. Cryptography in the Modern World

A. Secure Hash Algorithm 3 (SHA-3)

SHA3 [17] is a recently standardized hashing algorithm. Some of the most widely used hash algorithms such as MD4, MD5 and SHA-0 have been found to have substantial cryptanalysis weaknesses [4, 5]. Since 2005, SHA-1 was deemed theoretically unsecure. In 2017, it was finally found to be vulnerable to collisions. Google and CWI Amsterdam performed a collision attacks against SHA-1 with two different PDF files producing the same SHA-1 hash and since then all major web browser vendors do not accept SHA-1 SSL certificates. SHA-2 and SHA-1 were invented around the same time and their algorithms are related, so with attacks on SHA-1 NIST decided to put up a competition for SHA-3. In October 2008, there was about 64

algorithm submission and 5 were chosen in December 2010. On October 2012, Keccak algorithm was selected as SHA-3 [17].

NIST SHA-3 requirements:

- Algorithms that will accept arbitrary length of data with 4 possible output lengths of 224, 256,384, and 512 bits.
- The attack resistance for SHA3-224 is (3DES equivalent resistance).
- The attack resistance for SHA3-256 is 2^{112} (AES-128 equivalent resistance).
- The attack resistance for SHA3-384 is 2^{128} (AES-1192 equivalent resistance).
- The attack resistance for SHA3-512 is 2^{192} (AES-256 equivalent resistance).

SHA-3 is based on a scheme called sponge construction. The sponge function has absorption phase that takes an input message X_i and partitions it into fixed-size block. Each block Y_i is processed and then fed into a next iteration for 24 rounds until the fixed size output is produced at the end of the process. Just like any other cryptographic algorithm, each round will involve series of substitution and permutation operations.

In SHA-3 algorithm, the block size (state) is 1600 bits. Table II gives a high-level summary of SHA-3 algorithms.

TABLE II: A HIGH-LEVEL SUMMARY OF SHA-3

Algorithm	Output	State	Block Size (r)	Capacity (c)
SHA3-224	224	1600	1152	448
SHA3-256	256	1600	1088	512
SHA3-384	384	1600	832	768
SHA3-512	512	1600	576	1024

B. Hybrid Encryption Method/Digital Envelopes

To increase the speed and security, a new mode of encryption has been introduced in the recent years, which merges two or more encryption systems. This incorporates a combination of symmetric and asymmetric encryption to share information. The hybrid encryption [2] combines the flexibility of an asymmetric encryption with the efficiency of the symmetrical encryption.

The slow process of asymmetric key encryption is due to the additional complexity of the mathematical formulas used to achieve the public and private key encryptions. In symmetric algorithm, as we use the same key for both encryption and decryption, complexity of private key encryption is much less which makes it much faster compared to asymmetric encryption. An abstract view of a hybrid encryption system is shown in Fig. 5.

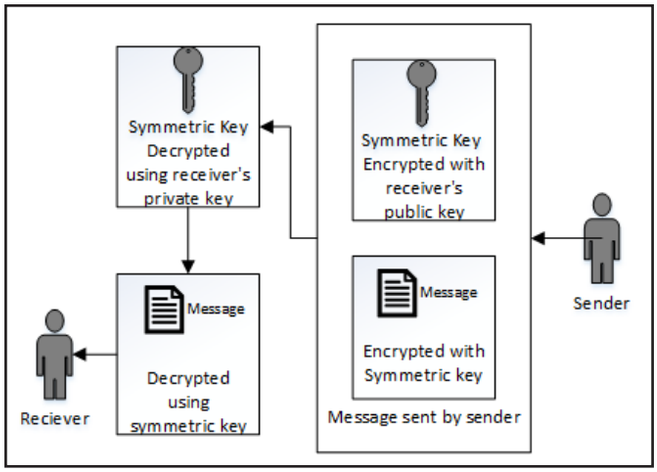

Fig. 5: Hybrid Encryption System

In hybrid encryption, the two keys generated by an asymmetric algorithm for protecting encryption keys and key distribution, and a secret key generated by the symmetric algorithm for protecting the actual message. As each encryption method has its own pros and cons, using them together like this can be the best of both worlds. Hybrid encryption is considered as an extremely secure type of encryption as long as the keys used for asymmetric algorithms are completely secure.

Thus, the hybrid scheme is getting the benefits of both symmetric and asymmetric systems. The key distribution issue is relaxed through the use of the pubic key scheme. Moreover, the improved performance of the hybrid system is achieved through using symmetric encryption scheme to carry out the actual encryption of the data.

C. Digital Cash

Digital cash [18] (also known as e-currency, e-money, electronic cash, electronic currency, digital money, digital currency, and cyber currency) is defined as a system in which any person can securely pay for goods or services electronically without necessarily involving a bank to mediate the transaction. Electronic money is broadly considered as an electronic store of monetary value on a technical device that may be widely used for making payments to undertakings (without necessarily involving bank accounts in the transaction). The electronic money products largely use cryptography to authenticate transactions and to protect the confidentiality and integrity of the transaction.

The key characteristics of electronic cash are [18] highly secured against unauthorized use, anonymous use by the consumers at both the ends of the transaction, portable so that it can be used in any location over the network, and transferable among consumers without a need to refer to a banking system. Some electronic cash systems allow the divisibility feature where electronic cash can be divided into small denominations. Fig. 6 shows an electronic cash process among a customer, a merchant and a bank.

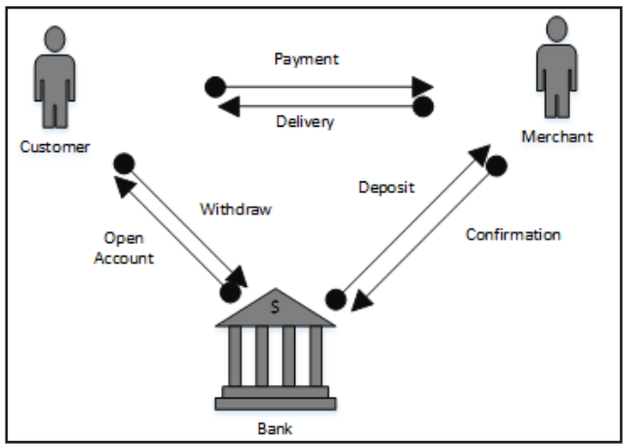

Fig. 6: Electronic Cash Process

The features of authenticity, anonymity, and multiple-spender exposure are achieved most conveniently using public-key cryptography. Electronic cash also uses hybrid encryption as a digital envelope to avoid duplicate spending by the consumers. Electronic cash also uses secure hashing to prevent digital forgery.

D. Message Integrity

Data in any state is subject to intentional or unintentional modifications. These modifications can be triggered by authorized or unauthorized person. Parity bits and cyclic redundancy check functions are used in many protocols to detect this kind of unintended modification. As parity bits do have the limitations of identifying intruders who modify the message along with the parity bit. For all these scenarios, one-way hash algorithms are required to identify all kinds of modifications.

One-way hash is a mathematical function that takes a message in any size and provides a fixed length value called a hash value [4, 5]. The hashing algorithm is not a secret and it is publicly known. The key secrecy is that the function runs only in one direction. For hashing, it is always true that the process cannot be reversed to extract the message.

If an intruder alters the message, it is also possible that he recreates the hash value and attaches the new hash with the message. Message Authentication Code (MAC) [4, 5] is a function that can be used to achieve additional security. A MAC function is an authentication scheme derived by applying a secret key to a message in some form.

There are three types of available MAC functions [4, 5]:

- Hash MAC (HMAC): A symmetric key is concatenated along with the message and hash is generated for the new message. As symmetric key is shared between the sender and receiver, the receiver should be able to recalculate the hash to make sure that the message is not altered.
- Cipher Block Chaining MAC (CBC-MAC): The message is encrypted with a symmetric block cipher in CBC mode and the output of the final block is used as the MAC. The message is shared as a plaintext and the receiver will do the same calculation to regenerate the MAC and compare to validate the integrity.
- Cipher Based MAC (CMAC): It is a new version of CBC-MAC with some additional mathematical security. This new version works with AES and 3DES algorithm. The symmetric algorithms are used to create symmetric key and that key is used to generate sub-keys. The sub-keys are used to encrypt the blocks individually and a MAC is generated at the end.

E. Digital Signature

Digital signature [4, 5] is a technique when a public key encryption is used for message authentication. In this case, the MAC is encrypted using the private key of the sender. At the receiver's side, the MAC is decrypted using the sender's public key.

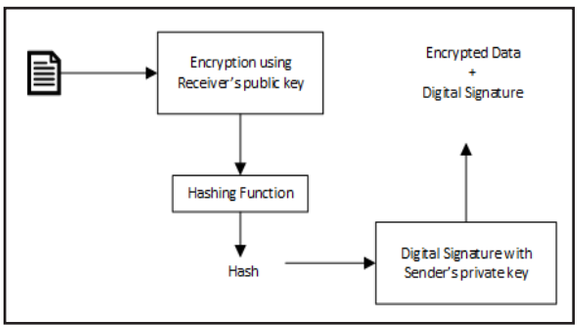

Fig. 7: Digital Signature at the Sender's Side.

F. Quantum Cryptography

Quantum cryptography [3] is the method/science of encrypting information by taking advantage of quantum physics at the physical layer. The principle of quantum cryptography is the implementation of BB84 protocol. In quantum cryptography, the information is encoded in non-orthogonal quantum states in such a way that photons are sent in any of the four predefined directions (vertical, horizontal, $+45^{o}$ or -45^{o}). These photons are used for key distribution between the sender and receiver to have a secured communication. The detailed description of this method is beyond the scope of this paper.

At this time, the quantum cryptography is still an evolving method for secure communication. Researchers are still researching to identify a reliable way to use this in day-to-day communication networks. Even though quantum cryptography has been experimentally proven feasible, the most important technological challenge remains the development of better photon counters, whose noise actually limits the transmission

distance below a few hundreds of kilometers. After transmitting initial few hundreds of kilometers, the photon strength will become too dim to be received and after that, the receiving system cannot decrypt or retransmit it.

Quantum cryptography promises to modernize secure communication by providing security based on the essential laws of physics, instead of the current state of mathematical algorithms or computing technology.

VI. Conclusion

Cryptography can be used in multiple ways to achieve all fundamental principles of security for a message communication. A message can be encrypted (symmetric or asymmetric key) to provide confidentiality. A message can be hashed to verify its integrity. A message can be digitally signed to enforce authentication, nonrepudiation and integrity. A message can be encrypted and digitally signed to provide confidentiality, authentication and integrity. It is important to understand that not all algorithms are defined to provide all security services. In this paper, we have focused on different techniques and its modern application of cryptography.

We have seen that the performance of the symmetric schemes is much higher than that of the asymmetric schemes. This is due to the complex mathematical operations behind the asymmetric systems and the larger key sizes for the same security level. For example, for an 80 bit security index, a symmetric encryption scheme can use a key of size 80 bits, while RSA (asymmetric system) needs a key of size 1024 bits.

Hybrid systems have been proposed to combine the benefits of both schemes (symmetric and asymmetric). The asymmetric part will be used to encrypt symmetric keys, which will be used to carry out the actual data encryption. This way, the key distribution issue of the symmetric schemes will be relaxed while achieving a higher performance.

References

[1] T. Morkel, and J. H. P. Eloff, "Encryption techniques: A timeline approach," *Information and Computer Security Architecture (ICSA) Research Group*, 2004.

[2] P. Kuppuswamy, and S. Q. Y. Al-Khalidi, "Hybrid encryption/decryption technique using new public key and symmetric key algorithm," *MIS Review: An International Journal*, vol. 19, no. 2, 2014.

[3] H. Zbinden, H. Bechmann-Pasquinucci, N. Gisin, and G. Ribordy, "Quantum cryptography," *Applied Physics B: Lasers & Optics*, vol. 67, no. 6, 1998.

[4] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 6th ed., p. 287, p. 292, p. 295, p. 339, 2014.

[5] W. Stallings, and L. Brown, *Computer Security: Principles and Practice*, 4th ed., p. 635, 2018.

[6] S. D. Nasution, G. L. Ginting, Md. Syahrizal, and R. Rahim, "Data security using vigenere cipher and goldbach codes algorithm," *International Journal of Engineering Research & Technology (IJERT),* vol. 6, no. 1, pp. 360-363, 2017.

[7] T. Chatterjee, T. Das, S. Dey, J. Nath, and A. Nath, "Symmetric key cryptography using two-way updated-generalized vernam cipher method: TTSJA algorithm," *International Journal of Computer Applications*, vol. 42, no. 1, pp. 39-42, 2012.

[8] L. Knudsen, and D. Wagner, "Integral cryptanalysis," *International Workshop on Fast Software Encryption*, vol. 2365, pp. 112-127, 2002.

[9] E. Fujisaki, and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," *Journal of Cryptology*, vol. 26, no. 1, pp. 80-101, 2013.

[10] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," *International Workshop on Selected Areas in Cryptography*, vol. 2259, pp. 1-24, 2001.

[11] E. F. Schaefer, "A simplified data encryption standard algorithm," *Cryptologia*, vol. 20, no. 1, pp. 77-84, 1996.

[12] H. Simon, "Advanced encryption standard (AES)," *Network Security*, vol. 12, pp. 8-12, 2009.

[13] S. Burnett, and S. Paine, *RSA Security's Official Guide to Cryptography,* McGraw-Hill, Inc., 2001.

[14] D. W. Davies, "Applying the RSA digital signature to electronic mail," *Computer*, vol. 16, no. 2, pp. 55-62, 1983.

[15] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," *CCS 1996*: *Proc. of 3rd ACM Conf. on Computer and Communications Security*, pp. 31-37, 1996.

[16] D. Hankerson, and A. Menezes, *Elliptic Curve Cryptography,* Springer US, 2011.

[17] J. P. Aumasson, L. Henzen, W. Meier, and R. C. W. Phan, "Sha-3 proposal blake," *Submission to NIST 92*, 2008.

[18] B. M. Jakobsson, and A. Juels, "Executable digital cash for electronic commerce," U.S. Patent 6,157, 920, Dec. 5, 2000.

[19] G. Saju, and M. P. Deepika, "DNA cryptography: New field of cryptography," *Journal of Network and Information Security*, vol. 6, no. 1, pp. 30-33, 2018.

[20] M. P. K. Kishore, and S. Budhiraju, "Hybrid cryptosystem using cellular automata transformations on graphs," *Journal of Network and Information Security*, vol. 4, no. 2, pp. 11-18, 2016.