International Conference on Engineering Research & Applications (ICERA)
May 17-18, 2017 Istanbul (Turkey)

Comparative Analysis of Cryptographic Algorithms and Smart Traffic Control Systems

Diedon BUJARI[1] and Erke ARIBAS[2]

[1]TUT Faculty of Computing and Electrical Engineering, Tampere/FINLAND
[2]ITU Faculty of Computer and Informatics Engineering, Istanbul/TURKEY

Abstract: *In this paper, the most used and popular cryptographic algorithms are investigated in detail, and compared in terms of security and total run time duration. Throughout history, different approaches have been practiced in order to encrypt confidential data, such as transposition and substitution. In modern cryptography, more advanced algorithms were designed, categorized as symmetric and asymmetric ciphers, and protocols. In this study, among symmetric algorithms, Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are studied, and among asymmetric algorithms, RSA is analyzed as a final step. Afterwards, they are compared in terms of performance and efficiency when implemented in MATLAB. Lastly, it is discussed about encryption issues in traffic control systems in smart cities, and the possible adaptation of the cryptographic algorithms mentioned above in those systems.*

Keywords: *cryptography, encryption, cipher, key, DES, AES, RSA, MATLAB, smart cities, traffic control systems*

1. Introduction

Cryptography is the art and science of encryption [1], which has existed for thousands of years, and it is at the heart of the communication network today. It is a crucial instrument for protecting data, like text, audio, video, image etc., from third parties while communicating. In other words, cryptography is used to initiate and preserve a secure communication in the presence of unauthorized attackers.

The fascinating story of cryptography has started in ancient Egypt, but not as a way of hiding information. Then, it turned out to be of great importance - it has decided wars and many other major events that dealt with information. The main cryptographic solutions to secrecy were transposition and substitution. Transposition is the rearrangement of letters in a word, generating an anagram. On the other hand, one of the first well-known substitution types of encryption, referred as the Caesar Cipher, was used by Julius Caesar around 58 B.C. [2]. In this technique, each letter is replaced by a letter some fixed positions down the alphabet. The Caesar cipher was used for nearly 800 years until it revealed that it can be easily broken using the "Letter-frequency Analysis". In order to flatten the distribution of letter frequencies, the polyalphabetic or Vigenère cipher came across, where each letter is replaced with a letter using multiple substitution alphabets [3]. A more modern cipher, which was used in the 19th century, is the Enigma machine. It was practiced by Germans during World War II for military communication [4].

Nowadays, the field of cryptography is much broader, and a fascinating field to work in. It covers various fields, such as communication technologies, computer security, economics, politics and many more extremely varied fields.

https://doi.org/10.15242/DIRPUB.DIR0517040

2. Encryption Methods

The original idea under cryptography is hiding or encrypting data from attackers. This idea is installed using cryptographic algorithms, also called ciphers, which can be divided into three main groups: asymmetric ciphers, symmetric ciphers, and protocols. A cryptographic protocol is a series of steps, which must be followed to apply cryptography. In other words, it includes the details of how the algorithm should be used. Nevertheless, the focus of this paper is not on protocols. Asymmetric ciphers (also called as public-key ciphers) were introduced by Whitfield Diffie and Martin Hellman, which were influenced by Ralph Markle [5]. These algorithms are designed so that there are two different keys: one for encryption and the other one for decryption process. The encryption key is public (public-key), but only the owner of the decryption key (private-key) can decrypt the data, as illustrated in Fig. 1. Symmetric or conventional ciphers are the oldest and most used group of ciphers. They are categorized into two groups: stream ciphers and block ciphers. Stream ciphers operate on single bits; however, block ciphers operate on fix-sized groups of bits. In this paper, only block ciphers of this category will be introduced. Communication using these kind of algorithms is established as shown in Fig. 2:

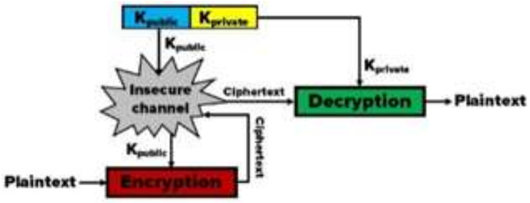

Fig. 1. Set-up for asymmetric algorithms

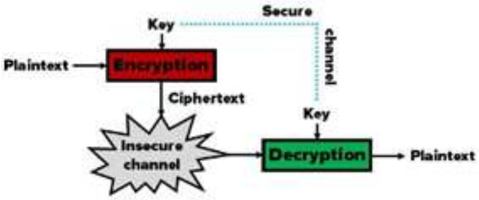

Fig. 2. Set-up for symmetric algorithms

The original data (plaintext) to be sent while communicating over an insecure channel is encrypted using any of the algorithms, yielding a meaningless data (ciphertext). In this way, third parties do not have access to the original data. The receiver, using decryption, which is the inverse procedure of encryption, decrypts the ciphertext, and reads the data. According to the principle postulated by Auguste Kerckhoffs, the encryption and decryption processes should be secure even if the third parties know all the details about the system, except the secret key [6].

2.1. Data Encryption Standard (DES)

The Data Encryption Standard or DES is a symmetric-key algorithm, which was developed by International Business Machines Corporation (IBM) in 1974, under the influence of the U.S. National Security Agency (NSA). In 1977, it was standardized by the National Institute of Standards and Technology (NIST) [7]. Although DES has been the most studied and popular cipher in the world in the last 30 years, it is considered as unsecure today because of its short key length. The algorithm encrypts blocks of 64-bit data, and the key length is restricted to 56 bits. According to Claude Shannon, known as “the father of information theory”, two principles should be considered while building a block cipher: confusion and diffusion [8]. By confusion, the relationship between the plaintext and the encrypted text is obscured; and by diffusion, the influence of each plaintext bit is spread over many cipher text bits. When these two properties are combined many times, an excellent avalanche effect can be reached: a small change in the plaintext can result in a big change in the cipher text.

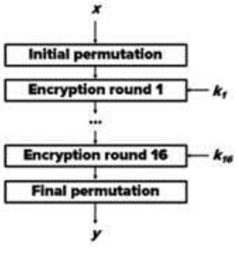

Fig. 3a. External structure of DES

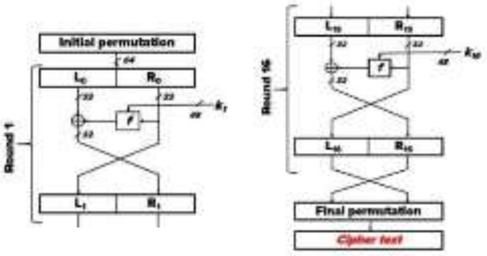

Fig. 3b. Internal structure of DES

The DES algorithm is built using the Feistel structure - a symmetric structure used in the construction of block ciphers. The advantage of this scheme is that encryption and decryption procedures differ only in key schedule. As seen in Fig. 3a, there are 16 rounds, which perform identical operations, and in each round, different subkeys of the main key are used.

a. Initial Permutation (IP): At the beginning of DES, the bits are rearranged according to a specific permutation table. Then, the 64-bit data is split into two 32-bit halves, L_i (left half) and R_i (right half). The inputs of the f-function, which is going to be discussed later, are the 32-bit right half and 48-bit k_i, and the output of this function is XOR-ed with the 32-bit left half. At the end of round 1, left and right halves are swapped and another round begins. The same process, except the initial permutation part, is repeated fifteen times more, and after the 16th round, the left and right halves are swapped, followed by a final permutation.

b. Final Permutation (IP^{-1}): It is the inverse operation of initial permutation. The bits are permuted according to a specific final permutation table. Both IP and IP^{-1}, and the general internal structure of the algorithm is shown in Fig. 3b.

c. The f-function: As it can be seen from the structure of DES, the inputs of the f-function are the 32-bit right half and the 48-bit round key. Firstly, the right half is expanded from 32 bits to 48 bits. This expansion operation provides diffusion. Then, the 48-bit output is XOR-ed with the round key. The output of this operation is divided into eight 6-bit parts, which are given as input to eight different substitution boxes (S-boxes). S-boxes replace 6-bit inputs with 4-bit outputs using specific substitution tables. These substitution operations provide confusion. Finally, there is a final permutation operation of this 32-bit output. At the end of these operations, the plaintext becomes a meaningless text (ciphertext).

Until now, there is no known analytical attack which breaks DES; the algorithm is resistant to differential and linear cryptanalysis. However, with today's technology, it is relatively easy to break it using brute-force attacks. For instance, the machine called Deep Crack, built by Electronic Frontier Foundation (EFF), was able to break the algorithm in just 56 hours, which showed that DES was no longer secure [9].

2.2. Advanced Encryption Standard (AES)

The Advanced Encryption Standard or AES is a symmetric block cipher, which was established in 2001 by the NIST. In the late 1990s, DES was seriously being attacked and became unsecure. By this time, the U.S. government called for a new encryption algorithm. Surprisingly, there were only 15 proposals submitted. One year later, in August 1999, five algorithms were selected for the final stage [10]. Finally, on October 2, 2000, "Rijndael", which was developed by two young Belgium cryptographers, Vincent Rijmen and Joan Daemen, was chosen as the AES.

AES is the most important and widely used cipher in the world at the moment. The fact that it is the first and the only public algorithm used by the NSA for 'top secret' data, is a strong endorsement for the algorithm [11]. AES has 128-bit input and output length, and supports three key lengths: 128/192/256-bit. Compared with other block ciphers, such as DES, it does not have a Feistel structure - all 128-bit data path is encrypted in one round, and each round consists of 4 layers, as illustrated in the internal structure in Fig. 4.

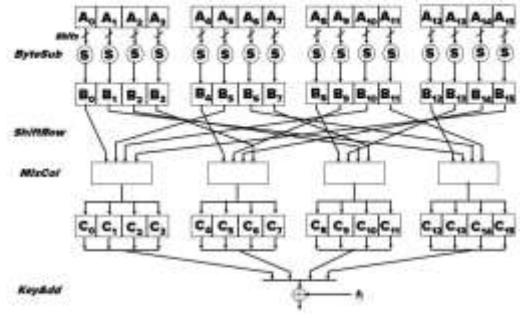
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
Shift
ByteSub
S
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15
ShiftRow
MixCol
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15
KeyAdd
ki

Fig. 4. Internal structure of AES

a. ByteSub layer: In this layer, each byte is substituted by another byte from the substitution table. Because of this, this layer is also called as the "S-box layer". The S-boxes are all identical.

b. ShiftRow layer: The bytes are permuted in a specific order. If you look at the structure of AES, this layer looks very complicated. However, if the state is written in matrix form, it looks very systematic.

c. MixCol layer: Blocks of 4-byte are combined using a linear mixing function. All addition and multiplication operations are done in the Galois Field (2^8).

d. KeyAdd layer: In the last layer, 16-byte data path is XOR-ed with the 16-byte subkey of that round.

This layer completes a single round. The number of rounds depend on the key length: 128-bit key = 10 rounds, 192-bit key = 12 rounds, 256-bit key = 14 rounds. Each round has one subkey, which is derived recursively from the input key. However, the key schedule in AES uses a different structure compared to the key schedule in DES. In order to make the algorithm symmetric, last round does not have the MixCol layer. Moreover, at the beginning and at the end of AES, a subkey is added in order to increase its security.

As it was discussed in DES, confusion and diffusion properties are combined many times in order to build a strong cipher. In AES, ByteSub layer provides confusion, and both ShiftRow and MixCol layers together provide diffusion. When it comes to security, there is currently no attack known. Brute-force attacks are not possible due to 128/192/256-bit key lengths. Several side-channel attacks have come across; however, those attacks were able to attack only the implementation, not the algorithm itself.

2.3. RSA

RSA was designed and firstly published in 1977. Its name comes from the initials of its designers' surnames: Ron Rivest, Adi Shamir and Leonard Adleman. It is the most famous and widely used asymmetric (public-key) algorithm. Every Internet user uses this algorithm or some variant of it, since it can easily be understood and implemented. In symmetric algorithms, there is a need for establishing a secure channel in order to share the key, which is very expensive and does not work well in large networks. However, in RSA, and other asymmetric algorithms, there are two keys: the public-key, which is used for encryption, and the private-key, which is used for decryption. The public-key is shared with everyone, and used for encrypting the plaintext. The resulting ciphertext is also shared; however, only the owner of the private-key can decrypt it in order to have the plaintext.

a. Generating the keys (K_{public}, $K_{private}$): Unlike in symmetric ciphers, the keys in asymmetric ciphers are computed. In order to determine the keys, these steps are followed:

1. *Choose a pair of large prime numbers, p and q, such that $p, q \geq 2512$*
2. *Compute their product: $n = p.q$*
3. *Calculate the value of the Euler's Totient function, which is equal to: $\varphi(n) = (p-1).(q-1)$*
4. *Choose $K_{public} = e$ from the set of $\{1, 2, ..., \varphi(n)-1\}$, such that they don't share any common factor: $gcd(e, \varphi(n)) = 1$*
5. *Compute $K_{private} = d$, which is the inverse of K_{public}, such that $d.e \equiv 1 \bmod \varphi(n)$. It can be calculated using the Euclidean Algorithm.*

=> *$K_{public} = (n, e)$ and $K_{private} = d$*

b. RSA encryption: In order to encrypt a plaintext (*x*) from the set $Z_n = \{0, 1, ..., n-1\}$, the formula is very simple:

$$\text{ciphertext} \equiv y = x^e \bmod n \tag{1}$$

c. RSA decryption: Given the ciphertext (*y*) from the set Z_n, and $K_{private} = d$, the plaintext is extracted using the formula below:

$$\text{plaintext} \equiv x = y^d \bmod n \tag{2}$$

Third parties cannot compute the private-key since they do not know the value of the Euler's Totient Function, particularly, the prime numbers *p* and *q*. Therefore, the security of RSA comes from the difficulty of

https://doi.org/10.15242/DIRPUB.DIR0517040

extracting these large numbers from the *φ(n)* function [12]. The time required for a computer to factor large numbers may take hundreds or thousands of years.

3. Comparison between DES, AES and RSA

The main goal of cryptography is maintaining the security during communication. In other words, the plaintext should be kept as secret against attackers. As far as it is seen, the details of all of these algorithms are public and can be analyzed. However, their security is not based in their details; it is based in the keys used. Because of this, many of today's cryptanalysis is done in order to recover the key of a cryptosystem.

As discussed before, the limited length of the key used in Data Encryption Standard (DES) makes the algorithm unsecure. Today's technological devices with high computing power, make the DES cryptosystem breakable, simply by brute-force attacking the system - checking for the desired key one by one from the key space (2^{56} possible keys). Moreover, as the technology develops, the algorithm will become less and less secure.

Besides the security component, a factor which makes DES still practical is its performance. Although it has a complex structure, including substitution boxes (S-boxes), initial (IP) and final permutation (IP^{-1}) boxes, the algorithm outperforms both AES and RSA in terms of execution. When implemented in MATLAB, it takes just 0.1087 seconds to encrypt a block of 64-bit random data using a random 56-bit key. The decryption process takes roughly the same amount of time: 0.0939 seconds.

Advanced Encryption Standard (AES) and RSA are considered unconditionally, as well as computationally secure today. In AES, there are 2^{128}, 2^{192} or 2^{256} possible keys. Although it was carefully designed to work efficiently in hardware [13], its software implementation is considered inefficient. In MATLAB, the encryption process for a block of 128-bit random data takes 1.2618 seconds, and the decryption process takes 1.2327 seconds, using a key of 128 bits. This inefficiency comes from its complicated structure.

In terms of performance and efficiency, RSA is the worst algorithm compared to DES and AES, since it is based on arithmetic operations, such as the exponentiation operations while encrypting and decrypting. Approximately, its hardware implementation is 1000 times, and its software implementation is 100 times slower than DES [14].

Table I. Comparison of algorithms' run time in MATLAB

Algorithm	**Key size**	**Running time in MATLAB**	
		Encryption	***Decryption***
DES	*56-bit*	*0.1087 s*	*0.0939 s*
AES	*128-bit*	*1.2618 s*	*1.2327 s*
RSA	*32-bit*	*3.5985 s*	*0.0197 s*
	64-bit	*4.1568 s*	*0.0365 s*
	128-bit	*5.3318 s*	*0.0523 s*
	256-bit	*8.1173 s*	*0.2949 s*

4. Smart cities

Since technology has gone beyond its borders and humans have become slaves to modernity, the environment has also incorporated new technologies in order to become smarter. One of the most popular areas arousing interest in recent years is the concept called smart city. A smart city is the one which organizes and manages its assets with the help of advanced technologies, particularly through Information and Communication Technology (ICT) and Internet of Things (IoT), in order to save money, as well as to improve and ease its citizens' lives. Some of the cities which are described as smart are New York, San Francisco, London, Tokyo, Barcelona etc. In these cities, new technologies are adopted in many fields, such as in governance, environment, communication, mobility and commerce. Since the rate of urbanization keeps increasing, related problems and concerns ought to be considered in smarter approaches.

4.1. Smart traffic control systems

The main aim of this paper is to analyze the mobility issues of a smart city, especially smart traffic control systems. Traffic control systems consist of traffic lights and signaling components, which are placed mainly in intersections and pedestrian crossings. These systems are installed in order to control the flow and preserve the security in traffic. When it comes to the question of how these traffic lights work, people mostly think that they are programmed to change their condition under fixed time intervals. This fact is common in nearly all of old signaling devices, which display the same sequence of color change throughout the day; for example, every 30 seconds the lights change.

Present-day technologies have come along with smarter and more progressive signaling systems, composed of three main parts: a detector, a controller and traffic light heads [16]. Detector, which is installed usually above the light heads, registers current traffic conditions and sends information to the controller. Controller uses this information to adjust and enhance the traffic flow based on density demands, and sends signals to traffic lights for changes. However, at the same time, innovations and advancements in these technology-dependent areas constitute new challenges and problems, which are mainly related to cyber security and encryption issues. Traffic control systems are also unguarded to attacks - they can easily be hacked and manipulated since they lack enough security. Primarily, the lights installed in dense streets and intersections, can cause a lot of chaos if attacked. The signals from the controller to the light heads are transferred wireless; therefore, the attacker is very likely to access and change the sequence of lights if the transmission is not encrypted well.

In most countries in the world, the standard time for one sequence of lights (one complete cycle, e.g. red-amber-green) is usually taken as 120 seconds [17]. Totally, there are 24h/120s = 720 cycles in a day. Since in one cycle three signals are sent, it means that about 720 cycles*3 signals = 2160 signals are transmitted from the controller to the traffic heads per day. In order to ensure the security of the communication between these two parts of traffic control systems, and also provide safety for cars and pedestrians in streets, the signals can be effectively encrypted using the cryptographic algorithms described in the first part of the paper.

Table II. Comparison of algorithms' implementation in traffic lights

	DES	AES	RSA
Time for one cycle	*120 s*	*120 s*	*120 s*
# of cycles / day	*720*	*720*	*720*
# of signals / day	*2160*	*2160*	*2160*
Time for encryption	*234.8 s*	*2725.5 s*	*11516.6 s*
Time for decryption	*202.8 s*	*2662.6 s*	*112.9 s*
non-encrypted / encrypted	*99.5 %*	*94.1 %*	*88.1 %*

**Encryption and decryption times are given for 2160 signals. In RSA, they were calculated using 128-bit key length.*

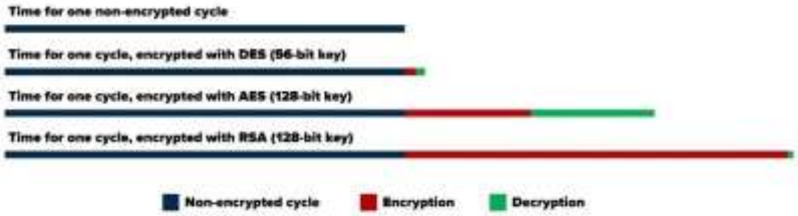

Fig. 5. Graphical representation of the change in time for encrypted signals

5. Conclusion

The current situation of traffic control systems' infrastructure, especially in many developed and modern cities, is concerning and should increase the public consciousness about the security issues in this field. As it is seen in the studies above, although DES has high performance capabilities, it should not be utilized in the encryption procedure of signaling in traffic control systems positioned in critical street networks since it can endanger the traffic safety if hacked. Nevertheless, in local roads, where any manipulation would not cause a lot of chaos, it can be used alternatively. On the other hand, it is possible to claim that AES and especially RSA, should be optimized in time constraints in order to serve faster, for preventing congestions, and in the most secure approach, for averting accidents in traffic.

Our next research will extend over multi-threading, load balancing and pipelining in order to lessen the effects of computational costs.

6. References

[1] Anderson, R. (2001). *Security Engineering*. [Chapter 5: Cryptography]

[2] Singh, S. (1999). *The Code Book*.

[3] Kotas, W. A. (2000). *A Brief History of Cryptography*.

[4] Lycett, A. (2011). *Breaking Germany's Enigma Code*. Available: http://www.bbc.co.uk/history/worldwars/wwtwo/enigma_01.shtml

[5] Schneier, B. (1996). *Applied Cryptography*.

[6] Kahn, D. (1996). *The Codebreakers: The Story of Secret Writing*.

[7] National Institute of Standards and Technology. (1993). *Announcing the Data Encryption Standard (DES)*. FIPS PUB 46-2. Available: http://iaic.csie.tku.edu.tw/netintro/Crypto/fips46-2.pdf

[8] Shannon, C. (1949). *Communication Theory of Secrecy Systems*. Available:
https://doi.org/10.1002/j.1538-7305.1949.tb00928.x
http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf [Page 708]

[9] Electronic Frontier Foundation. (1998). *Frequently asked questions about the Electronic Frontier Foundation's "DES Cracker" Machine*. Available: https://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_des_faq.html#howsitwork

[10] National Institute of Standards and Technology. (2000). *Report on the Development of the Advanced Encryption Standard (AES)*. Available: http://csrc.nist.gov/archive/aes/

[11] Mangard, S., Oswald E. & Popp, Th. (2007). *Power Analysis Attacks*. [Page 296]

[12] Rivest, R. L., Shamir, A. & Adleman, L. (1977). *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Available: http://people.csail.mit.edu/rivest/Rsapaper.pdf

[13] Park, J. S. (n. d.). *Analysis of AES Hardware Implementations*. Available: http://www.cs.ucsb.edu/~koc/cren/project/pp/park.pdf

[14] Skiena, S. S. (1997). *The Algorithm Design Manual*.

[15] Aloul, F. et al. (2012). *Smart Grid Security: Threats, Vulnerabilities and Solutions*. Available:
https://doi.org/10.12720/sgce.1.1.1-6
http://www.ijsgce.com/uploadfile/2012/1011/20121011121836539.pdf

[16] Sanderson Associates (Consulting Engineers) Ltd. (n. d.). *How do Traffic Signals Work*. Available: http://www.traffic-signal-design.com/how_do_traffic_signals_work.htm

[17] Green Signals Consulting Ltd. (n. d.). *UTC/SCOOT and Pedestrian Pushbuttons*. Available: http://www.greensignals.co.uk/news/utcscoot-and-pedestrian-pushbuttons/