Computer
& Robotics

Cryptographic Algorithms: A Review of the Literature, Weaknesses and Open Challenges

Yashar Salami[a], Vahid Khajehvand [a,]*, Esmaeil Zeinali[a]

[a]Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Received 07 April 2023; Accepted 31 May 2023

Abstract

Information security has become an important issue in the modern world due to its increasing popularity in Internet commerce and communication technologies such as the Internet of Things. Future media actors are considered a threat to security. Therefore, the need to use different levels of information security in different fields is more needed. Advanced information security methods are vital to prevent this type of threat. Cryptography is a valuable and efficient component for the safe transfer or storage of information in the cyber world. Familiarity with all types of encryption models is an essential need for cybersecurity experts. This paper separates Cryptographic algorithms into symmetric (SYM) and asymmetric (ASYM) categories based on the type of cryptographic structure. SYM algorithms mostly use the Feistel network (FN) structure, Substitution-Permutation Network (SPN), and the ASYM algorithms follow the mathematical structures. Based on this, we examined different encryption methods in terms of performance and detailed comparison of key size, block size, and the number of rounds. In continuation of the weakness of each algorithm against attacks and open challenges in each category, to study more is provided.

Keywords: Cryptography, Security, Algorithms.

1.Introduction

The Internet is one of the most influential new communication technologies that has somehow affected all aspects of human life. These new communication technologies have made people spend much of their time on the Internet and use online services for education, research, banking services, and other activities [1], [2].

The significant progress has made paper, the main carrier of important information, to be regularly replaced by other ways of information exchange. In fact, the paper has the disadvantages of slow transfer and high cost; in addition, it’s archiving results in many problems. Regularly, as the computer networks progress, the traditional trade had moved from the current state toward the electronic one and the exchange of documents has become widespread in this kind of commerce. These documents often contain sensitive information, such as legal contracts, confidential technologies, or financial transactions; however, we cannot exchange sensitive information in the cyberspace in a safe way, because when the context and the essence of traditional life is transforming into a modern model, the social crimes and civil disorders also take the form of modernity.

*Corresponding Author. Email: vahidkhajehvand@gmail.com

Today, jobbers would access the confidential information to hit persons, organizations and governments to access confidential information, and for this purpose, they use different methods, such as viruses [3], worms [4], Trojans [5], backdoors [6], and other available methods to achieve this important and sensitive information. In this case, ensuring unauthorized access to sensitive information is among the most important security challenges in relation to the distribution of information in cyberspace[7]. Various solutions, such as authentication [8] and key exchange systems[9][10], firewall [11]–[13], Intrusion detection systems [14]–[17], encryption of data [18], and the use of various security tools have been provided to maintain network security.

In encryption, the presence of information or message sending is no secret. Still, data storage or message sending is clear, and only the intended persons can restore the original report. In general, the encryption algorithms are divided into SYM and ASYM categories. SYM encryption algorithms use similar encryption keys to encrypt and decrypt the ciphertext, while ASYM algorithms use a pair key for encryption and decryption [19].

Most SYM algorithms have been designed based on the FN structure and SPN. ASYM algorithms are based on Mathematics and have been developed chiefly based on the factorization of prime numbers (FPN) and the discrete logarithm (DL). After searching, we concluded that a comprehensive paper about encryption algorithms is essential. Accordingly, we tried to collect several important cryptographic algorithms in a form with all details; we also compared all the encryption algorithms in terms of structural type, key size, the length of blocks, the number of rounds, weakness against attacks, and the production year of the algorithm and the obtained results were prepared for other researchers to use this paper for advancing, as well as developing the encryption and authentication systems in the environments, such as cloud, IoT, etc.

1.1.Motivation

There are many challenges in the field of research and development to create an intelligent world. Real, digital, and virtual worlds combine to create innovative environments. In the meantime, protecting data and personal privacy is essential with the development of the intelligent world. Since a complete study of cryptographic algorithms' weaknesses and open challenges has not been done so far, we decided to present a detailed analysis of SYM and ASYM in this article.

1.2.Contribution

In this article, we provide a detailed classification of SYM and ASYM cryptographic algorithms, and we will continue to examine the main concepts of cryptography and their structures.

We provide an overview of encryption methods and then analyze encryption methods based on year of production, key size, block size, and structure type.

Encryption methods are examined regarding weaknesses against various attacks, and the open challenges of encryption methods are raised.

1.3.Paper Organization

The rest of this paper is organized as follows: In Sect 2, basic conceptual information about SYM and ASYM encryption systems have been provided. The related work is explained in Sect 3, provided. Sect 4 describes the important indicators which have been discussed for evaluating the encryption algorithms. Sect 5 provides a general overview of the cryptography categories. Sect 6 gives an analysis for observing and examining the cryptography algorithms. In Sect 7, open challenges are explained. Finally, Sect 8 and 9 of the paper ends with the concluding and future work remarks.

2.Basic Concepts

In this section, the definitions used in the paper and the structure of encryption algorithms are explained.

2.1.Definitions and Terms

Given the fact that specific interpretations have been used for cryptography in this paper, a brief description has been presented in Table (1) to the reader for a better and further understanding of the information used in the paper.

Table 1
Describes Important Cryptographic Information

Concept	Details
Plain text	Message and data in the original state, before turning to encryption mode is called plain text, or in brief, message. In this case, the information can be understood by human beings
Code text	A message and information when it converts into a code. The encrypted data cannot be understood by human beings.
Encryption	An operation converts a message into code, using a code key.
Decryption	An operation converts the encrypted message into the original text, using a code key. Mathematically, this algorithm is contrary to the encoding algorithm.
Code key	The code key is generally a numerical information, which is given to the coding algorithm as an input parameter, by which encryption and decryption operations are performed. Different kinds of code keys are defined and used in cryptography.

2.2.Cryptography

It is composed of two words, Kryptos means "confidential," and Graphien means "writing." Cryptography is a mathematical or logical system; on that basis open and generally understandable information and concepts are converted into obscure information in a reversible order. This obscure information is reversible and utilizable by those who know the reverse order and the required parameters. No point shall be kept secret in the cryptographic algorithms and their reverse order; thus, all cryptographic algorithms require a parameter called (cipher key) by which the mysterious nature of the encrypted information would be changed unpredictably.

2.3.SYM Encryption Algorithms

Public key encryption or ASYM encryption is an encryption method in which the key used for encryption differs from the key for decryption (as opposed to SYM cryptography, in which encryption and decryption are performed with a single key) [20]. In ASYM cryptography, the user holds a pair of keys:

1-The public key (used for encrypting the original text)

2-The private key (used for decrypting the ciphertext)

It is clear that the private key remains secret, but the public key may be widely disseminated. The received encoded messages by the user's public key are only readable to themselves, as the user itself holds the private key for decryption. The public key is mathematically correlated, but the private key is not practically measurable from the public key[21]–[23]. The following Fig (1) shows the ASYM encryption.

2.4.SYM Encryption Algorithms

The ASYM key algorithm is a class of encryption methods which uses a similar key for both encryption and decryption of a text. The encryption keys might be similar, or there might be a simple correlation between them. The key, in practice, represents a shared secret between two or more than two parties which can be used for protecting the private information in secret[21], [24]–[26]. Fig (2) shows the SYM encryption.

2.5.Discrete Logarithm

DL functions in Mathematics and Algebra are a set of functions that are similar to conventional logarithm functions and are defined on the numerical groups. The mathematical definition of these functions is simply as follows. Suppose that the ring group G with n member(s) of integers is based on the multiplication with a producer like b. By this assumption, we could write every member of g from group G as $g = bk$, which in this equation, k is an integer, and for each specific g and b, there are different values for k, all of which are members of a modular arithmetic class with n module. On that basis, the DL function in the basis b is a function of G to Zn (the ring of integers with n modulo), which attributes to every g member of the set, G arithmetic class of k with n module. The issue of the DL is identical with solving the equation $ax \equiv b \bmod p$ for x, and for its use in Mathematics, especially in cryptography, it has become a terminology[27]–[31].

Mathematically, solving a discrete algorithm problem (calculating the DL) is considered as solving the problem of integer analysis and they have something in common: either of them is among difficult mathematical problems, such that no quick way has been found for solving them [32]. Every algorithm related to either problem can be converted to a similar algorithm in relation to another problem. Difficulty with solving either problem has been used for designing, as well as developing cryptographic systems, such as the ElGamal Cryptography.

2.6.Integer Analysis

In the number theory, breaking down a complex number and writing it as a product of some integers is called the analysis of natural numbers. No efficient algorithm has been identified for analysis of very large numbers. Many mathematical and computer sciences have been employed to deal with this issue, such as quantum calculations and theory of algebraic numbers. The most difficult state (for the current methods) is the *semi-integer* numbers [33]–[35]. These kinds of numbers are the ones which can be written as the multiplication of two integers. When these two numbers are very great and are randomly selected with a fairly close value, even for the fastest algorithms on the fastest computers it would take such a time which is in fact inefficient. It has caused this difficulty to be used in the body of most cryptographic systems [36]. Some cryptographic systems, like RSA, use this method[37].

2.7.Substitution Permutation Network

Substitution-Permutation encryption is a set of mathematical operators, which is used in key-frame algorithms, such as Advanced Encryption Standard. Such network takes one frame of the main text as an input for the key k, applying multiple layers of substitution and permutation boxes on it to obtain a frame of encrypted text. In fact, substitution and permutation boxes are converting the sub-frames of input bits into output bits. In principle, operators such as XOR efficiently running on the hardware, are used. In each step, a sub-key of the main key is obtained using a key schedule, which is considered the key to that layer. The encryption is done in a simple way, such that the Substitution and Permutation boxes are used conversely, along with sub-keys of each step[38]–[43]. A substitution box converts a small frame of input bits into another frame of bits. This substitution should be done one by one in order to ensure that the encryption is performed. Specifically, the output length of the substitution box is the same as its input length. Fig (3) shows substitution boxes with 4 inputs and 4 outputs, though it is not generalized and there are substitution boxes, which are not same in the length of input and output. A good substitution box should have the property that substituting an input bit make changes in half of the output bits. Also, every output bit of such a box should depend on all input bits. A permutation box provides a permutation of bits. In every layer, this box takes output [bits] of substitution boxes, permuting its bits, and then, gives its output to the substitution boxes of the later step. A good permutation box distributes the output bits of a substitution box of the previous step among several substitutions of the next step, as far as possible. In each step, the step key is permutated with XOR box [44]–[47].

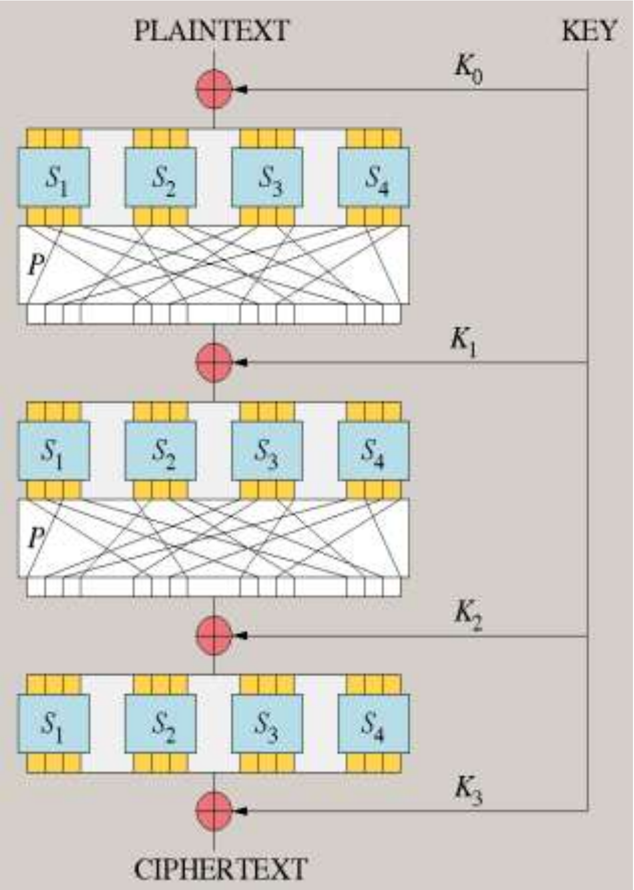

Fig. 3. The Substitution Permutation Network.

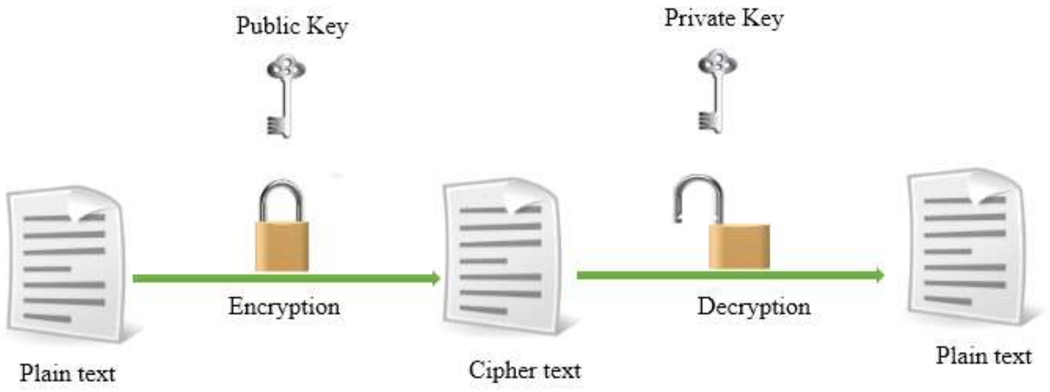

Fig. 1. The ASYM encryption algorithms.

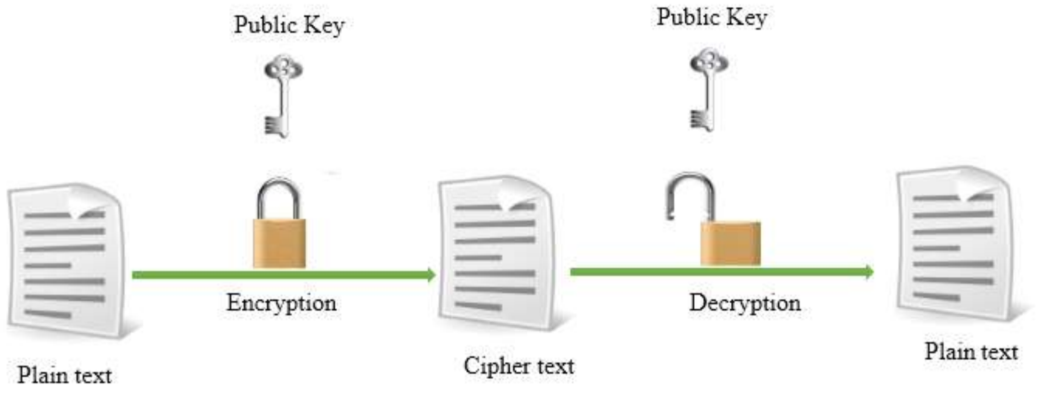

Fig. 2. The SYM encryption algorithms.

2.8.Feistel Network

Based on Shannon's studies and suggestions, Horst Feistel, a researcher from IBM Company, proposed a general pattern for the SYM cryptography, which was significant for nearly 30 years, on which many modern cryptographic methods were designed [48][49]. Fig (4) shows the Feistel architecture for each round of the cryptographic process.

- Cryptography is composed of some repeated and similar steps known as round. In each round, possibly only one round-key (or sub-key) changes, and the nature of the practice is clear and intact in every round.
- The input of each round is divided into left and right halves and in each round, an input semi-thesis remains intact, while the second half becomes round-key based on a very complex and highly non-linear hybrid of the first and second halves.
- To encrypt each round, it is enough to have the intact half and the round key.
- After each round, two halves should be switched (swap) in order that the intact half be included in the next round.
- The cryptography process of the second half of the input should be performed by a highly non-linear and non-algebraic function (Substitution and Permutation, combining the bits of the key, modular calculations). If we call this complex function as f, the strength and speed of the

cryptographic system will depend on the inner support f.

- To perform decryption operation, no reverse function f $(i.e., f^{-1})$ is needed, instead, the decryption is done by re-applying f on the input parameters of each round (including round key, the right half, and the left half). This property makes encryption and decryption algorithms similar, structurally.

The function f is called the Mangler as it combines the data in the bits of the key, when it shedders data in a nonlinear way, forwarding the data to the output after permutation. The function f might be a one-way function; it means that even by having an output key, f^{-1} might not be calculated. There are different kinds of the Feistel methods, including tri-stage, nested, and unbalanced, all of which are generated by slight changes from the original version [50]–[53]. There are other types of the FNs, such as Type-3 FN, Unbalanced FN and the Nested FN which have been designed on this basis.

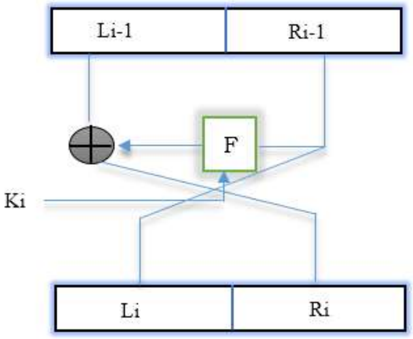

Fig. 4. The Feistel architecture for one round

3.Related Work

M. Ebrahim, S. Khan, and U. Bin Khalida comprehensive comparative analysis of different existing cryptographic algorithms (SYM) [54]. They have been examined on some of the designing factors, such as the key length, the number of rounds, and the type of structure in terms of flexibility and vulnerability against the attacks, and the methodology of the SYM algorithms has been described. However, their study is not a comprehensive cryptographic study. In short, the disadvantages of this method are as follows:

- ASYM algorithms have not been raised.
- It has not been investigated in terms of production decade.
- Most well-known SYM algorithms have not been investigated.
- Open challenges have not been addressed.
- The proposed algorithms have not been considered in terms of the developer type.

G. Singh, have focused on a few Cryptographic Algorithms, which are used for data encryption [55]. They have been examined in some designing factors, such as the key length, the number of rounds, and the type of the structure. They have described only 4 algorithms in the paper. However, their review is not a comprehensive cryptographic study. In short, the disadvantages of this approach are as follows:

- Most ASYM algorithms have not been raised.
- It has not been investigated in terms of production decade.
- Most well-known SYM algorithms have not been investigated. Others have not been considered.
- The proposed algorithms have not been thought-out in terms of vulnerability and flexibility.
- The proposed algorithms have not been considered in terms of the type of developer.
- Open challenges have not been addressed.

T. Gunasundari and K. Elangovan, have offered some important Comparative of the Cryptographic Algorithms in data encryption [56]. They also have examined most designing factors, such as the key length, the number of rounds, and the type of the structure. This paper has proposed only on 4 algorithms developed by solely Mr. Rivet; however, their review is not a comprehensive study on cryptography. In short, the disadvantages of this approach are as follows*:*

- ASYM algorithms have not been raised.
- It has not been investigated in terms of production decade.
- Other SYM algorithms have not been investigated.
- Other proposed algorithms have not been investigated in terms of flexibility.
- Other proposed algorithms have not been investigated in terms of the type of developer.
- Open challenges have not been addressed.

As another survey, M. Agrawal and P. Mishra have offered some important Comparative Symmetric Key Encryption Techniques *[57]. Similar* to other writers, s/he has studied on comparing the designing factors, such as the block length, the key length, the number of rounds, the type of structure, and the type of vulnerability. The author describes only four well-known SYM algorithms in the paper. However, their review is not a comprehensive study on cryptography. In short, the disadvantages of this approach are as follows:

- ASYM algorithms have not been raised.
- Other SYM algorithms have not been investigated.
- Other proposed algorithms have not been investigated in terms of flexibility.
- Other proposed algorithms have not been investigated in terms of the type of developer.
- Open challenges have not been addressed.

Finally, this paper presents by E, Surya; C.Diviya a detailed study of the symmetric encryption techniques over each other.[58]. They have been studied on comparing the designing factors, such as the block length, the number of rounds, the type of structure, and the type of vulnerability. The methodology has been described on four well-known SYM algorithms. However, their review is not a comprehensive study on cryptography. In short, the disadvantages of this approach are as follows:

- ASYM algorithms have not been raised.
- Other SYM algorithms have not been investigated.
- Other proposed algorithms have not been investigated in terms of flexibility.
- Other proposed algorithms have not been investigated in terms of the type of developer.
- Open challenges have not been addressed.

As a result, although the paper mentioned is essential. However, items not covered can be shown in Table (2). In this article, we tried to cover things not covered in the previous paper and provide a detailed assessment of the criteria set out in Sect 4 for the reader to read.

Table 2
Comparison of related work.

Features	[54]	*[55]*	*[56]*	*[59]*	*[58]*
A symmetric	*x*	✓	*x*	*x*	*x*
Symmetric	✓	✓	✓	✓	✓
Flexibility	✓	*x*	*x*	*x*	*x*
Vulnerability	✓	*x*	✓	✓	✓
Developer	*x*	*x*	*x*	*x*	*x*
Decade	*x*	*x*	*x*	✓	✓
Open challenges	*x*	*x*	*x*	*x*	*x*

X indicates not supported; ✓ indicates partially supported.

4.Criteria

The important indicators discussed for evaluating cryptographic algorithms have been described in this Sect. Table (3) to the reader for a better and further understanding of the information used in the paper.

Table 3:
show important indicators discussed for evaluating cryptographic algorithms.

concept	Details
Architecture	It defines that on what structure do the cryptographic algorithms have been designed?
Key size	It defines what key sizes are used for encryption by the cryptographic algorithms?
Round	It defines how many rounds have been used for data encryption by the cryptographic algorithms?
Block size	It defines the size of each encryption block in each cryptographic algorithm.
Developer	It defines by whom the cryptographic algorithms have been generated: Organizations or individuals.
Years	It defines in what year the cryptographic algorithms have been produced?
Flexibility	It defines that the encryption algorithm is capable of tolerating minor user variations.
Attack	It defines that to what types of attacks the encryption algorithms are susceptible.

Since each encryption algorithm is susceptible to one type of attack, not all algorithms can be studied in terms of a single type of attack. For this reason, several attacks in which the studied encryption algorithms are weak have been explained as follows:

Slide Attack:
A slide attack is a form of code analysis designed to counteract the general idea that even weak passwords can become very strong by increasing the number of rounds, confronting with a differential attack. The slide attack acts in such a way that unrelated the number of rounds in a single password, uncovers the flaws to decipher the code. The maximum common cause is the cyclically repeated keys [60][61].

Brute-Force Attack:
A comprehensive search is an attack in which all possible scenarios for obtaining an answer are examined. For each cryptographic model, we can calculate the required time for testing all possible methods for the key. Usually, cryptographic patterns are designed so that the construction of all possible scenarios is impossible or ineffective at any given time. Also, the "inclusive search attack" is a criterion for recognizing password-cracking methods. Testing all possible scenarios is also considered a way of finding the password. Usually, the software blocks the user account several times after entering incorrect passwords or delays the validation process to avoid testing other scenarios [62][63].

Man-in-the-middle attack:
The middle man attack (often abbreviated as MITM and also known as the bucket brigade attack) is a form of eavesdropping in cryptography and computer security in which the attacker establishes independent connections with the victims, redistributing the messages between them in such a way that they are convinced to speak with each other directly through a private connection. At the same time, all conversations are controlled by the attacker. The attacker should be able to eavesdrop on all the messages exchanged between the two victims and create a new message capable of good function in many situations [64][65].

Shor's Algorithm:
Shor’s algorithm, adopted from Mathematician Peter Shor, is a quantum algorithm (an algorithm that runs on quantum computers) for integer factorization. Formulated in 1994, it informally solves the prime factors [66]–[68].

Pohlig–Hellman Algorithm:
In group theory, the Pohlig–Hellman algorithm, sometimes called as the Silver–Pohlig–Hellman algorithm, [69] is an especial algorithm for computing DLs in a finite abelian group whose order is a smooth integer[70].

Related-key attack:

In cryptography, a related-key attack is considered any form of cryptanalysis where the attacker can observe the operation of a cipher under several different keys whose values are initially unknown, but in which some mathematical relationships connecting the keys are known to the attacker[71][72].

Differential cryptanalysis:

Differential cryptanalysis is a general form of cryptanalysis applicable primarily not only to block ciphers but also to stream ciphers and cryptographic hash functions]. In the widest sense, it is the study of how differences in information input can affect the resulted difference in the output. In the case of a block cipher, it refers to a set of techniques for tracing differences through the network of transformation, discovering where the cipher shows non-random behavior, and exploiting such characteristics to recover the secret key [73].

Interpolation Attack:

In cryptography, an interpolation attack is a cryptanalytic attack against block ciphers. In this attack, an algebraic function is employed to represent an S-box which might be a simple quadratic, a polynomial, or rational function over a Galois field. Its coefficients can be determined using standard Lagrange interpolation techniques using known plaintexts such as data points. Alternatively, selected plaintexts can be used to simplify the equations and optimize the attack. In its simplest version, an interpolation attack expresses the ciphertext as a polynomial of the plaintext. If the polynomial has a relatively low number of unknown coefficients, then the polynomial can be reconstructed with a set of plaintext/ciphertext (p/c) pairs. With the reconstructed polynomial, the attacker has a representation of the encryption without the exact knowledge of the secret key [74][75].

Mod *n* Cryptanalysis:

In cryptography, mod n cryptanalysis is an attack that can be applied to block and stream ciphers. It is a form of partitioning cryptanalysis that exploits unevenness in how the cipher operates over equivalence classes (congruence classes) modulo n. The method was first suggested in 1999 by John Kelsey, Bruce Schneier, and David Wagner [76].

Truncated differential cryptanalysis:

In cryptography, truncated differential cryptanalysis is a generalization of the differential cryptanalysis which is an attack against block ciphers. Lars Knudsen developed the technique in 1994. While ordinary differential cryptanalysis analyzes the full difference between two texts, the truncated variant addresses partially determined differences. That is, the attack predicts only some of the bits instead of the full block[77][78].

Impossible Differential Cryptanalysis:

In cryptography, an impossible differential cryptanalysis is a form of differential cryptanalysis for block ciphers. While normal differential cryptanalysis tracks the differences that propagate through the cipher with higher probability than expected. Impossible differential cryptanalysis exploits the differences that are impossible at some intermediate state of the cipher algorithm[79][80].

XSL Attack:

In cryptography, the Extended Sparse Linearization (XSL) attack is a way of cryptanalysis for block ciphers. The attack was first published in 2002 by the researchers, Nicolas Courtois and Josef Pieprzyk. It has created many controversies as it was claimed that it has the potential to break the Advanced Encryption Standard cipher, also known as Rijndael, faster than an exhaustive search[81]. Since AES is already widely used in commerce and government for transmitting secret information, finding a technique that can shorten the time required to retrieve the secret message without having the key, could have wide implications. The method has a high work-factor which means that in the case of lessening, the effort required to break AES is not reduced compared to an exhaustive search. Therefore, it does not affect the real-world security of block ciphers in the near future. Nonetheless, the attack has made some experts to express greater inconvenience at the algebraic simplicity of the current AES. In an overview, the XSL attack first relies on analyzing the internals of a cipher and deriving a system of quadratic simultaneous equations. These equation systems are generally very large; for example, 8,000 equations with 1,600 variables for the 128-bit AES. Several methods are known for solving such systems. In the XSL attack, a specialized algorithm known as Extended Sparse Linearization, is then applied to solve these equations and recover the key. The attack is known since it requires only a handful of known plaintexts for operation; previous methods of cryptanalysis, such as linear and differential cryptanalysis, often require unrealistically large numbers of known or chosen plaintexts[82]–[84].

5.Cryptography Categories

This Sect provides a general overview of the cryptography categories based on the type of algorithm and its structure.

5.1.Based on SYM and ASYM Algorithms

Encryption Algorithms can be divided into two categories structurally: first, the ASYM algorithms which have been designed based on the FN and SPN; second, the ASYM algorithms, in which most cryptographic algorithms have been designed based on the DL or prime number factorization. Fig (5) shows the categorization of cryptographic algorithms.

5.2.Based on SYM Algorithms

In this Sect, we have studied the SYM encryption algorithms, which have been designed based on the FN and SPN structures. Since in this paper many algorithms have been studied, we have described the well-known and influential ones

5.2.1.Algorithms Designed Based on the FN

This Sect describes the various SYM algorithms based on the FN.

DES:

In the early 1970s, the US Federal Government and IBM Company jointly developed a methodology for data encryption, in order to be used as a standard for keeping government documents confidential. This method was called the Data Encryption Standard (DES) [85]. The DES is a SYM key encryption algorithm, which was designed based on the structure of the FN in 1970. In the DES encryption algorithm, both the block sizes and the key sizes are 64 bits, but from a 64-bit key, only 56 bits are used, and from a 64-bit key, only 56 bits are used and the remaining 8 bits are used only for checking the parities [86][87]. The DES algorithm is composed of 16 similar stages, each of which is called round. The text supposed to be encrypted is exposed to an IP (Initial Permutation), and then a series of complex actions related to the key is performed on it, and finally, it is exposed to the Final Permutation (FP). IP and FP are reversed. FP neutralizes the action performed by IP[88]. As shown in Fig (6), each round of encryption i takes the 64-bit block of the previous round i-1 as its input. Then, this block is divided into two 32-bit left and right parts of Li-1and Ri-1, respectively. The right part is directly used as the left part of the next round, or in other words, Li=Ri-1. The difficult part of the work is done by the Mangler F. This function encrypts the 32-bit Ri-1 block with a 48-bit Ki key in order to obtain an encrypted 32-bit block. Then, this block is combined with Li-1 by XOR operator to obtain Rioutput. At first, the Mangler function develops 32-bit Ri-1 block to a 48-bit block, and then divides the output into a 6-bit block by XOR to Ki. Then, each 6-bit block is given to another function known as s-box, which it functions to convert a 6-bit input into a 4-bit output. Eight 4-bit outputs of s-box function are combined and are delivered as function F after being re-permuted. 48-bit Ki keys are produced of the main 56-bit key. First, the main key is permuted and then is divided into two 28-bit parts. For each round i of semi 28-bit, one or two left or right bits are rotated, and 24 bits are extracted from them. At last, these 24-bit blocks are combined and the final 48-bit key is generated [89].

TDES:

Triple Data Encryption Algorithm(TDES) or Triple DES is a SYM key encryption algorithm, which has been designed in 1970 by IBM, based on the structure of the FN [90]. In the TDES encryption algorithm, the blocks have the size of 64 bits, and the key size may include 168, 112, or 56 bits. TDES algorithm contains 48 rounds, as opposed to DES having 16 rounds. The TDES encryption is developed to improve the security of DES [91], [92].

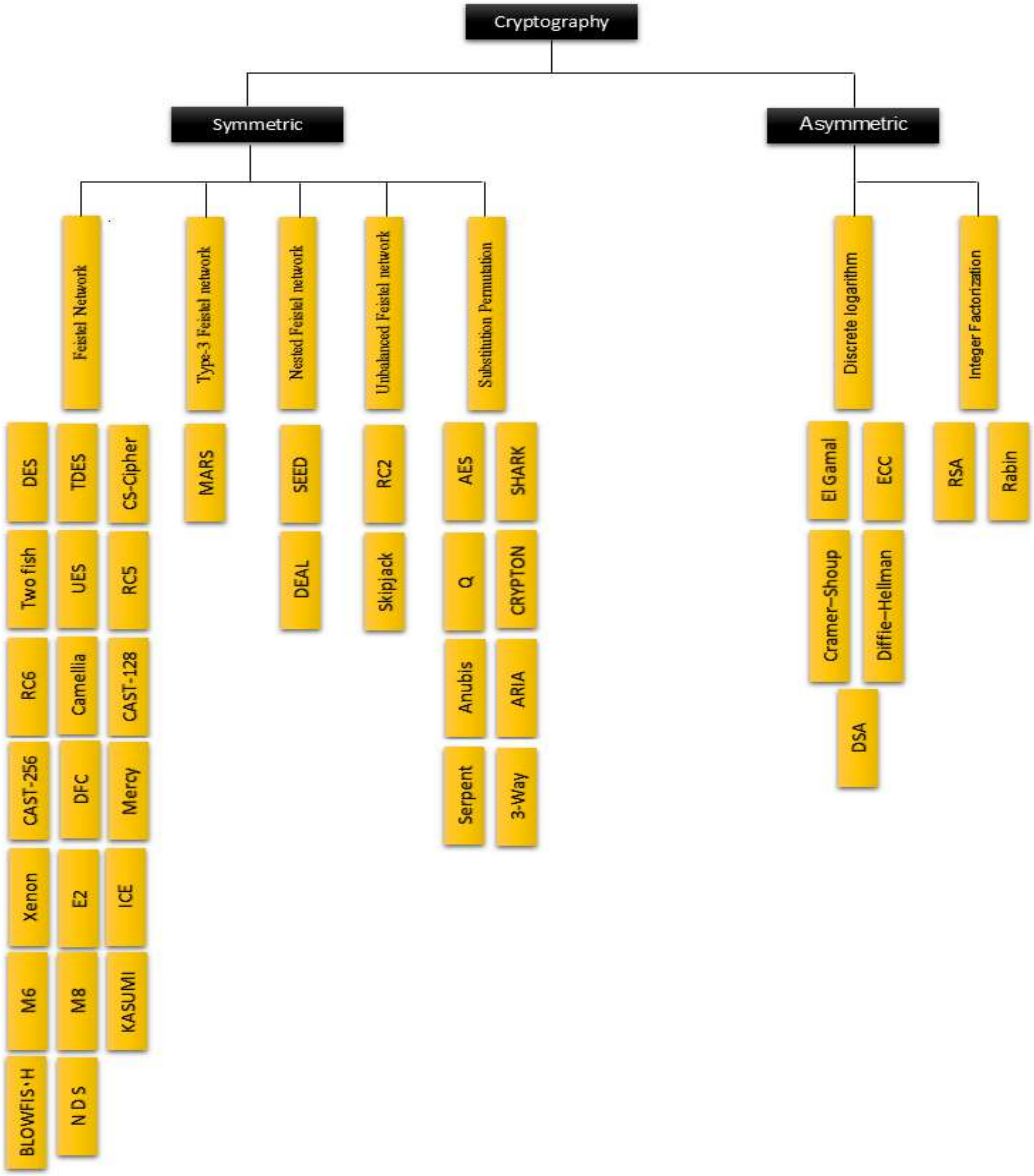

Fig. 5. The categorization of the cryptographic algorithms.

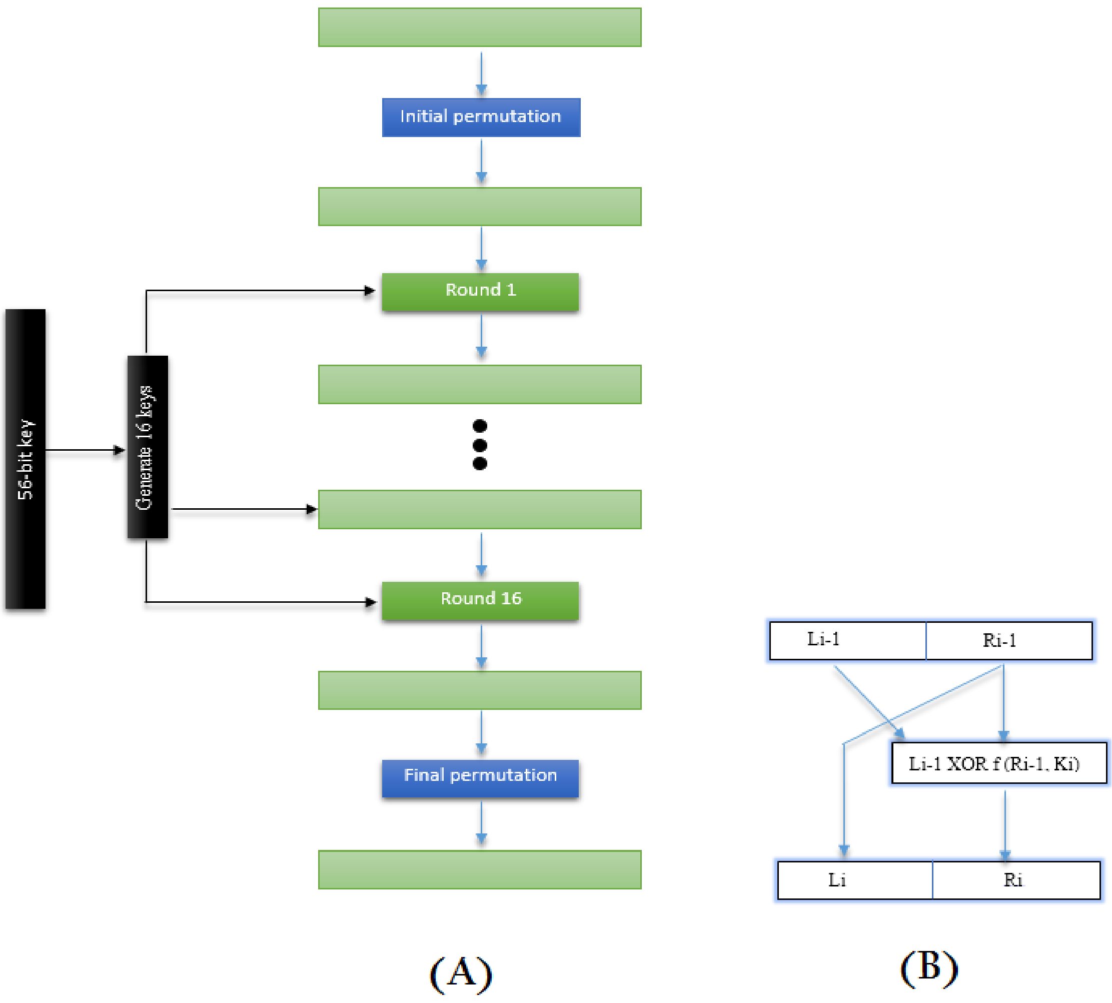

Fig. 6. Shows the principles of DES cryptography (a) and (B) the details of one round of cryptography.

Blowfish:

Blowfish is a SYM key encryption algorithm designed by Bruce Schneier in 1993 based on the FN structure [93][94], [95]. In the Blowfish algorithm, the blocks have the size of 64 bits and the key length may vary from 32 to 488 bits. Like the DES algorithm, the Blowfish uses 16 rounds for encryption. The Blowfish has designed an all-purpose algorithm as a problem-free and limitation-free substitution for communicating with other algorithms[96]. When Blowfish was released, many other schemes were plagued by the Patent Law, or otherwise considered as a trade or state secrets. Schneier announced that Blowfish is not patented and will remain in that way in all countries; hereby, the Blowfish algorithm will be placed in the public proprietorship and can be freely used by anyone. Fig (7) shows the Blowfish encryption structure.

Twofish:

Twofish algorithm is a SYM key encryption algorithm, which has been designed by Bruce Schneier in 1993 based on the FN structure [97]. The Twofish algorithm uses 16 rounds for encryption, and each block has the size of 128 bits, and the key size can be 128, 192 or 256 bits[98], [99]. Fig (8) shows the Twofish algorithm structure. Being in the public proprietorship, the Twofish algorithm has not been patented. As a result, the Twofish algorithm, freely available for everybody, can use it with no limitation. This algorithm is among the few encryptions involving Open-PGP (RFC4880), though it has not been used as widely as Blowfish. The latter managed to gain third place in the AES competition.

RC:

ARC2 or RC2 is a SYM key encryption algorithm designed by Ron Rivest in 1987 based on an unbalanced FN [100], [101]. The number of turns in this algorithm is 16 and each block has the size of 64 bits. It is noteworthy that a key size can vary from 8 to 1024, but it has been considered as having 64 bits by default [102].

RC5:

RC5 is a SYM key encryption algorithm developed by Ron Rivest in 1994 based on the FN and has come into attention due to its simplicity[103]–[105]. In RC5 algorithm, the number of rounds may vary from 1 to 255, and the size of a block may vary between 32, 64, or 128 bits, but the 64-bit size is suggested more frequently [106], [107]. In RC5, the key size may vary from 0 to 2048; however, the 128-bit key size is used more often than others.

RC6:

RC6 or Rivest cipher 6 is a SYM key encryption algorithm inspired by RC5. The RC6 algorithm was designed in 1998 by Ron Rivest, Matt Robshaw, Ray Sidney, Yiqun Lisa Yin based on the FN [108]–[110]. In this encryption system, 20 rounds have been used for encrypting each block. Each block has the size of 128 bits, and its key may have the sizes of 128, 192, or 256. The RC6 algorithm works on word *W*, its value is variable and selected, but it had been considered equal to 32 to compete in AES[111]–[114]. In other words, the words are processes with a length of 4 bites, which is highly useful for 32-bit processors [115]. RC6 couldn't rank a place better than third in the competition for selecting the Advanced Encryption Standard.

Camellia:

Camellia Algorithm is a SYM key encryption system, which based on the FN has been jointly designed in 2000 between *Mitsubishi Electric* and (Nippon Telegraph and Telephone) NTT in Japan for the project *NESSIE* (New European Schemes for Signatures, Integrity and Encryption) and *Cryptography* (Research and Evaluation Committees)[116]–[118]. The number of rounds in Camellia may vary from 18 to 24, and each block is 128 bits in length and the key length can be 128, 192, or 256 [119]–[121]. Camellia algorithm has been designed in such a way that is appropriate for both software and hardware implementation. Also, it has been used in Transport Layer Security (TLS) for the security of network connections[122].

CAST-128:

CAST-128 is a SYM key encryption algorithm of CAST family, which has been designed by Carlisle Adams and Stafford Tavares in 1996 [123], [124]. It has been designed based on the FN. Each block in the CAST-128 encryption algorithm has the size of 64 bits, and the key may vary between 40 to 128 in length [125]–[127]. The number of rounds in the CAST-128 algorithm varies from 12 to 16. CAST-128 has been used in some versions of PGP and GPG by default. The Government of Canada uses the CAST-128 to secure the communications.

CAST-256:

CAST-256 or CAST6 is a part of the SYM key encryption algorithm, which had been proposed for the nomination of *Advanced Encryption Standard* (AES). CAST-256 algorithm has been designed by Carlisle Adams, Stafford Tavares, Howard Heys, Michael Wiener in 1998 based on the FN [128], [129]. CAST-256 encryption algorithm has 48 rounds for each 128-bit block, and the key size can be 128, 160, 192, 224, or 256 bits [129]–[132]. CAST-256 is available for the commercial and non-commercial uses, free of charge and without a license.

SEED:

The SEED is a SYM key encryption algorithm designed by KISA (Korea Information Security Agency) in 1998 [133]. The SEED algorithm uses a 128-bit block and the key length of 128 bits for data encryption. This algorithm has been built based on the nested FN and employs 16 rounds for data encryption [134]–[136]. The SEED encryption system is highly favored in South Korea but rarely used elsewhere.

Skipjack:

Skipjack is a SYM -block encryption algorithm, which has been designed by the National Security Agency (NSA) in 1998 based on the unbalanced FN[137][138]. The Skipjack encryption algorithm uses 32 rounds for encrypting the data blocks with the length of 64 bits and with the key size of 80 bits [139][140]. Skipjack represents one family of encryption algorithms, which was the part of NSA collection of "NSA product types". In designing this algorithm, mostly the combinatory and algebraic techniques have been used.

Xenon:

The Xenon encryption algorithm has been designed in 2000 by Chang-Hee Lee based on the FN. The Xenon is an ASYM key algorithm which uses 64-bit blocks with 16 rounds with 128-, 192-, or 256-bit key lengths.

E2:

The E2 encryption algorithm has been designed by Nippon Telegraph and Telephone Corporation (NTT) in Tokyo, Japan, based on the FN. E2 algorithm is a SYM key encryption system, which has blocks with 128 bits in length and the key length of 128, 192, or 255, with 12 rounds for each block [141]. Nominated for AES competition, E2 failed to win it.

ICE:

The ICE Encryption algorithm (Information Concealment Engine) has been designed in 1997 by Matthew Kwan, whose structure is similar to the DES algorithm [142]. The ICE algorithm is a SYM key algorithm based on the FN. ICE algorithm uses 16 rounds for encrypting 64-bit blocks with the key size of64 bits [143], [144]. This algorithm has never been a patented algorithm and its source code has been open, available to the public.

M6:

The M6 is a SYM key encryption algorithm presented by Hitachi in 1997 for IEEE 1394 FireWire Standard. The design of this algorithm allows for freedom of choice in the code operation to the users. For this reason, the M6 has been considered as a big family of SYM encryption. The M6 encryption algorithm uses 10 rounds for data with 64 bits in the FN. The key sizes are 40 bits by default, but they can be continued up to 64 bits [76].

M8:

TheM8encryption algorithm has been designed by Hitachi in 1999 to improve the security of the M6 and enhance the efficiency in the implementation of hardware and software in 32-bit systems. The M8algorithm uses 10 rounds for encrypting 64-bit blocks with the key length of 256 bits in the FN. The round function may include bit rotations, XORs, and modular addition, but the structure of each function round is determined by the key [145].

KASUMI:

KASUMI algorithm has been designed in 2000 by Mitsubishi Electric [146]–[149]. It is a SYM key algorithm which is mostly used in the mobile communication systems, such as UMTS, GSM, and GPRS[150]–[152]. The KASUMI encryption algorithm has been designed based on the FN . KASUMI uses a key size of 128 bits with 8 rounds of rotation for encrypting 64-bit data [153]–[155].

ND:

NDS or (New Data Seal) is a SYM key encryption algorithm that has been designed by IBM in 1975, based on the Lucifer algorithm. Like DES, the NDS algorithm uses 16 rounds for encryption in the FN and 128-bit blocks with the 2048-bit key length is used for encryption[156][157].

CS-Cipher:

The CS-Cipher encryption algorithm is a SYM key encryption algorithm designed in 1998 by Jacques Stern and Serge Vaudenay for the Project of NESSIE (New European Schemes for Signatures, Integrity, and Encryption) [158], [159]. However, it did not qualify for the Project NESSIE, opening its place for this project. The CS-Cipher algorithm has been designed based on the FN. In the CS-Cipher algorithm, the key length for data encryption may vary from 0 to 128, such that it should be a multiple of 8. The encryption operation is performed at 64-bit blocks with 8 rounds. The round function works based on the Fast Fourier Transformation using the E binary expansion [160], [161].

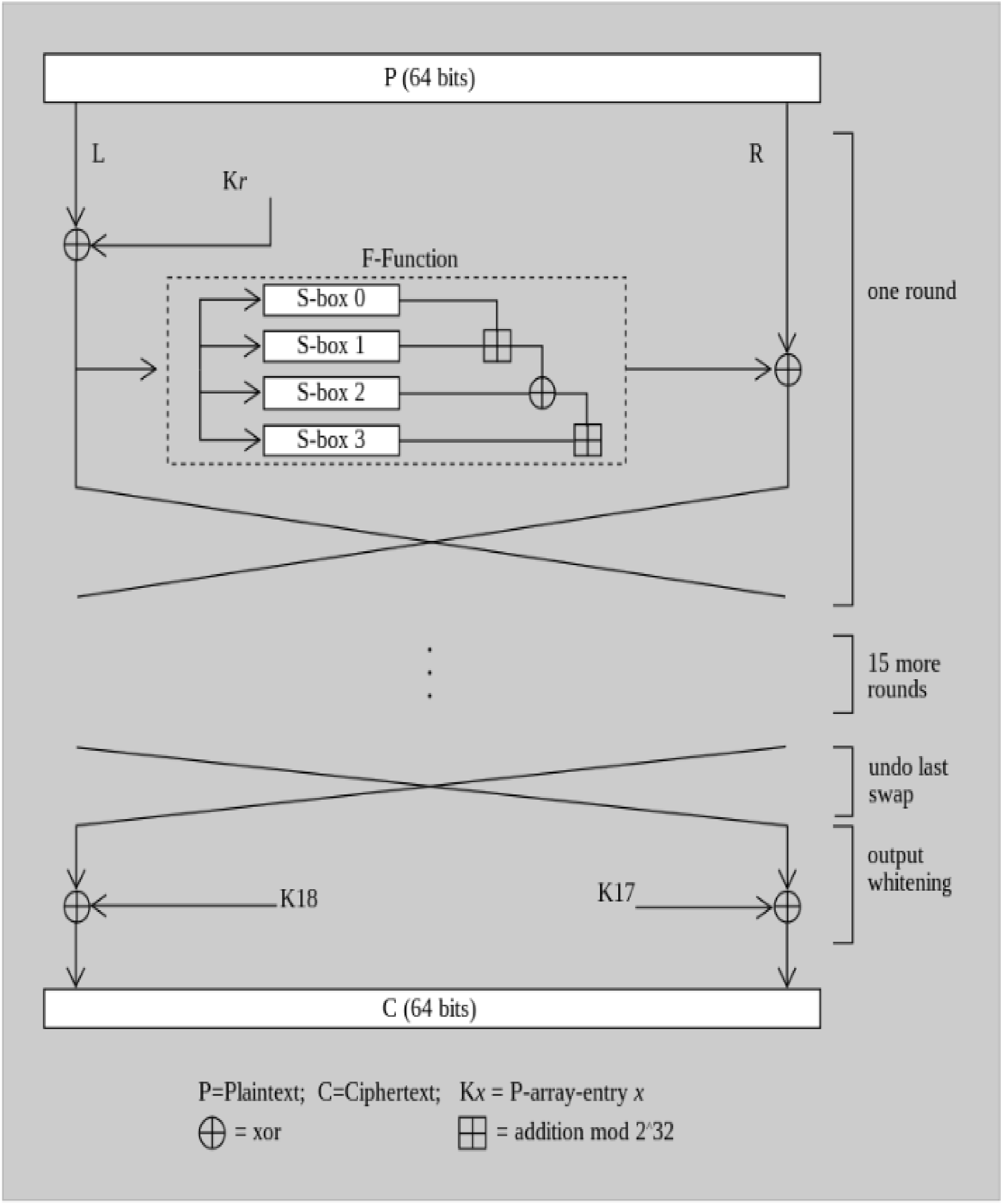

Fig. 7. The structure of the Blowfish cryptography algorithm.

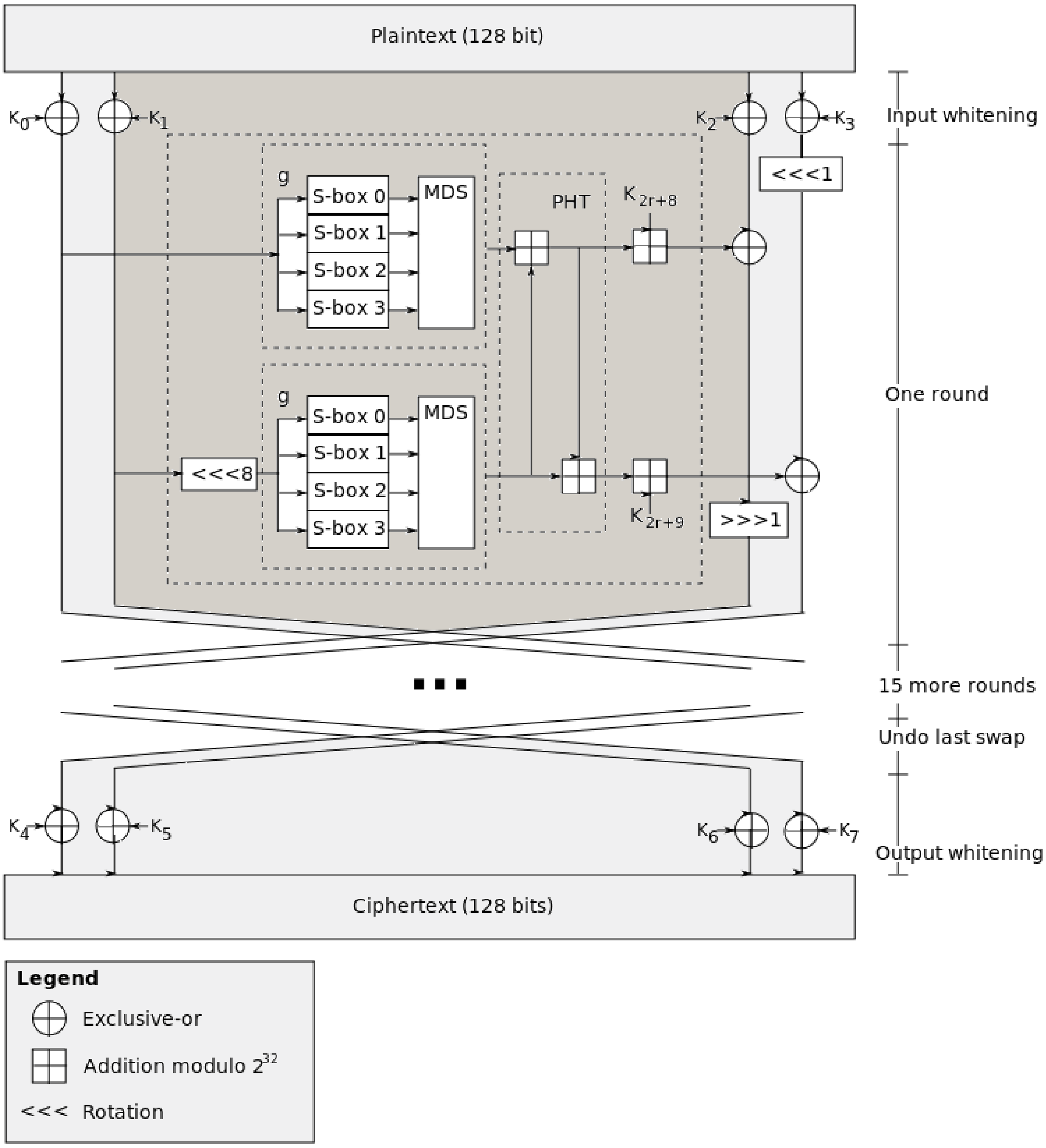

Fig. 8. The structure of the Twofish cryptography algorithm.

MARS:

The MARS encryption is a SYM key algorithm designed by IBM in 1998. MARS managed to enter the five leading finalists in the Conference AES2[164]–[166]. In a report, IBM announced that two MARS and Serpent algorithms are the only suitable algorithms for implementing the advanced security in the network. MARS algorithm uses a 3-stage FN with 16 rounds. The MARS algorithm has a block length of 128 bits, and the key length of 128, 192, or 256 can be used [127], [167], [168].

DFC:

The DFC encryption or (Decorrelated Fast Cipher) is a SYM key algorithm, which was designed in 1998 to compete in the *Advanced Encryption Standard Competition*, but it failed to be placed among the five finalists[169][170]. The DFC algorithm has been designed based on the FN and uses 8 encryption rounds of the blocks with 128 bits in length, which the key size in the DFC can be equal to 128, 192, or 256 [171].

DEAL:

The DEAL (Data Encryption Algorithm with Larger blocks) encryption algorithm is a SYM block algorithm, originated from the DES encryption system. The DEAL algorithm was designed by Lars Knudsen in 1998 [172]. The DEAL encryption algorithm is based on the nested the FN. The DEAL algorithm uses the blocks with 128 bits in length and the key sizes of 128, 192, and 256 bits for encryption. For encryption with the key sizes of 128 to 192, the DEAL algorithm uses 6 rotation rounds, and for encryption with the length of 256, it uses 8 rounds [173], [174]. The DEAL also participates in the AES competition, but it couldn’t manage to be among 5 finalists [175], [176].

UES:

The UES (Universal Encryption Standard) encryption algorithm is SYM key encryption which has been designed by Helena Handschuh and Serge Vaudenay in 1999[177]. The UES has been designed with the same user interface of AES. The UES algorithm uses a 128-bit block with the key lengths of 128, 192, or 256 bits[164]. It has been designed based on the FN and uses 48 rounds for encryption[178].

5.2.2.Algorithms Designed Based on SPN

This Sect discusses the types of algorithms based on SPN.

AES:

The Advanced Encryption Standard (AES), also known as Rijndael, was developed by two Belgian cryptographers, Joan Daemen, and Vincent Rijmen and first released in 1998[179]–[183]. The AES encryption algorithm has replaced the Data Encryption Standard (DES). The AES is a SYM - key algorithm, meaning that the same key is used for encryption and decryption. Unlike DES, the AES does not use the Feistel encryption system, but it has built on a rule called SPN [179]. This has made the AES become faster, both in hardware and in software. The Advanced Encryption Standard is a variant of Rijndael, which has a block size of 128 bits and the block size of 128, 192, 256 bits[184]–[186]. In contrast, the perse characteristic of the Rijndael algorithm is determined with the key and block sizes which can be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits. The Advanced Encryption Standard operates on a 4 * 4 matrix of bytes in column order, called a *state*, though some versions of Rijndael have larger block sizes and more columns in a state[185], [187]–[190]. Most AES calculations are performed in a certain finite field. The key size used in the AES code determines the number of repetitions in the conversion cycles (transformation), which transforms the input, named plain text into the final output, called ciphertext. The number of repetition cycles is as follows:

- 10 rounds for 128-bit keys
- 12 rounds for 192-bit keys
- 14 rounds for 256-bit keys

Each round consists of several processing stages, any of which depends on the encryption key. A series of reverse cycles is used to convert the ciphertext into the plain text using the same encryption key[187], [191]–[193]. Fig (9) shows the flowchart of a 128-bit AES.

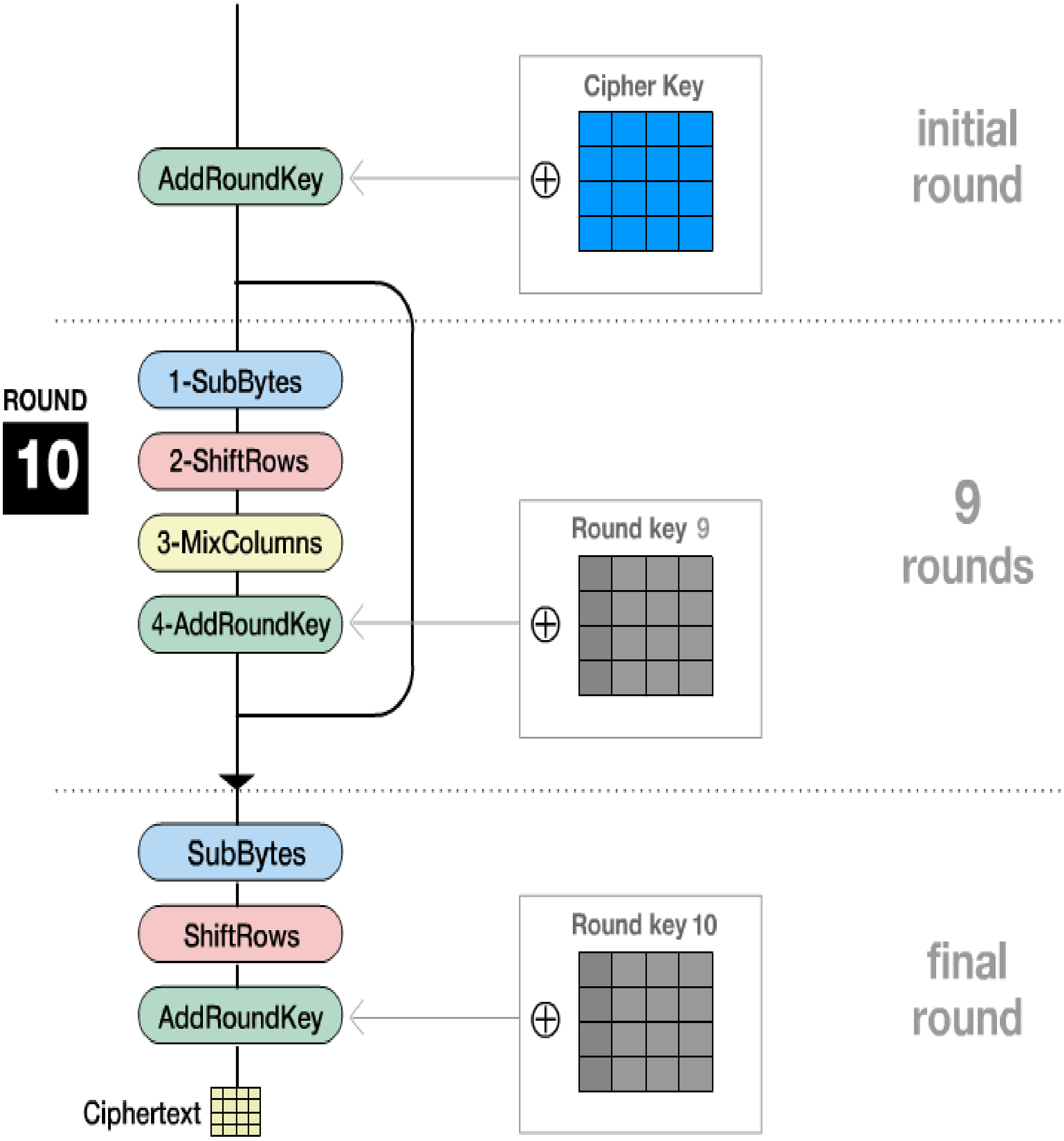

Fig. 9. The AES 128-bit flowchart.

Serpent:

In the Advanced Encryption Standard (AES) which lead to the victory of the Rijndael algorithm, the other proposal called *Serpent* was in the second place. Although the Serpent algorithm failed to become the Standard US Federal government, its strength and power made many great cryptographers to admire it[194][195]. The Serpent is a SYM key encryption algorithm, which has been designed in 1998 by Ross Anderson, Eli Biham, and Lars Knudsen based on the SPN. The Serpent uses 32 rounds to encrypt the 128-bit blocks with the key lengths of 128, 192, and 256[196]–[199]. The Serpent was never patented, and it was freely available to the public. It can be implemented freely on the software and hardware by everyone[200]–[203]. Fig (10) shows the linear diagram block of the Serpent algorithm.

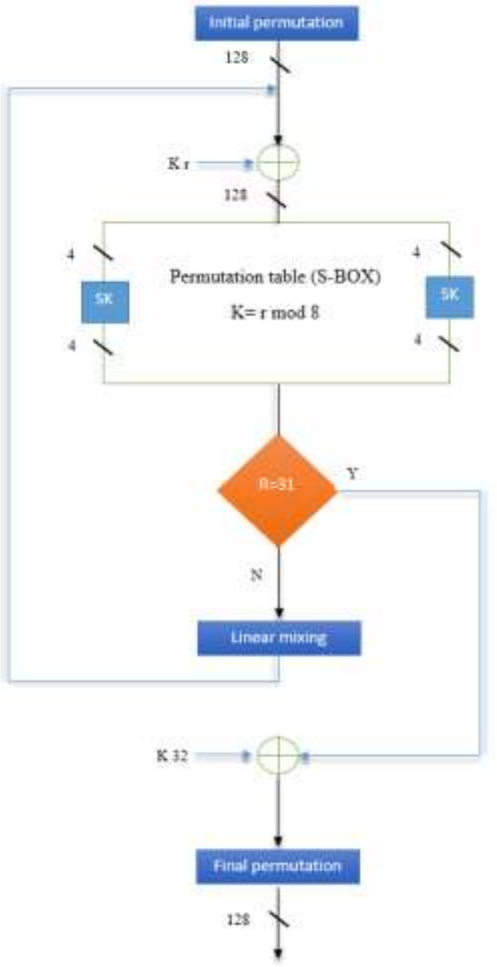

Fig. 10. The linear diagram block of the Serpent algorithm.

ARIA:

The ARIA encryption algorithm is a SYM key encryption algorithm which has been designed by a large group of South Korean researchers in 2003[204]–[207]. In 2004, The Korean Agency for Technology and Standards (KATS) selected the ARIA algorithm as its own encryption system. This algorithm has been designed on the SPN structure based on the AES[208]. The ARIA encryption algorithm uses the block size of 128 bits in length, with the key sizes of 128, 192, 256 bits. The number of rounds can be 12, 14, and 16, depending on the key length[209]–[213].

Way:

The 3-Way encryption algorithm was designed in 1994 by Joan Daemen. It is an ASYM key encryption algorithm developed based on the SPN structure. The 3-Way uses 11 rounds for encrypting 96-bit blocks with a key length of 96 bits[214][215].

Crypton:

The Crypton encryption algorithm is a SYM key algorithm, which was designed based on the SPN in 1998 by Chae Hoon Lim [216]. The Crypton algorithm uses 12 rounds for encrypting 128-bit blocks with the key lengths of 128, 192 or 256 bits [217]–[222].

Anubis:

The Anubis encryption algorithm is a SYM key encryption algorithm, which was designed in 2000 by Vincent Rijmen and Paulo S. L. M. Barreto for the Project of NESSIE (New European Schemes for Signatures, Integrity, and Encryption)[223]. This algorithm has been designed based on the SPN. The Anubis uses 128-bit blocks with the key lengths of 128 to 320 bits for data encryption. The Anubis uses at least 12 rounds for the encryption of data with the key lengths of 128 bits[224][225]. The Anubis encryption algorithm was released freely by its designers for the public uses.

Q:

The Q is a SYM key encryption algorithm which was designed in 2000 for the project NESSIE by Leslie McBride. This encryption algorithm has been designed based on the SPN structure. The Q algorithm uses 8 or 9 rounds for the encryption of 128-bit blocks with the key lengths of 128, 192 or 256[226][227].

Shark:

The Shark is a SYM key encryption algorithm which has been designed in 1996 by Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bosselaers, and Erik De Win. The Shark algorithm has been designed based on the SPN structure. This algorithm uses 6 rounds for encrypting the 64-bit blocks with the key length of 128 bits [75], [228], [229].

5.3.Based on ASYM Algorithms

In this Sect, we have discussed the ASYM encryption algorithms, which have been designed based on DL and FPN. It has been attempted to study those commonly used algorithms.

5.3.1.Algorithms Designed based on the DL

This Sect discusses the types of algorithms based on the DL.

ElGamal:

The ElGamal is a public key encryption algorithm developed based on the Diffie-Hellman (DH) key exchange protocol. It was designed by Taher Elgamal in 1984[230]–[233]. The ElGamal encryption is built based on the DL and can compete with RSA in terms of strength and confidence, though it is much more complicated and slow[234]–[236]. Fig (11) shows the ElGamal encryption algorithm. Dr. Taher El-Jamal did not patent the material and intellectual right for his algorithm. This algorithm progresses as follows:

Assume that Alice wants to choose one public and one private key, to whom others including Bob could forward their messages after encryption using the public key.

1. Alice chooses a very large prime number called(P).
2. Since the ZP collection contains too many generators, Alice chooses one of these numbers, and it is called(g).
3. Alice chooses the number a by the condition of $1 \leq a \leq p-1$, making it as its own private key and keeps it with her.
4. Alice makes the chosen number g as the exponent of her private key, a, and after calculating the residuum with the module cup p, calls it ß according to Eq (1).
5.

$$\beta = G^{A} \text{mod } P \qquad (1)$$

6. Alice gives the three sets (P, g, β) as the public key to the public, while her private key is(P, G, a), in which only a has been kept secret.

Public Key (P, g, β)

Private Key (P, β, a)

Or, assume that Bob wants to send the message M to Alice. Before any work, he should divide his message into i-character blocks and attribute an integer called mi according to a fully arbitrary rule, such that $0 \leq mi \leq p-1$ is held [237]–[239].

1. Bob chooses a completely random and arbitrary number called k, with the condition of $1 \leq X \leq p-2$.
2. He converts every mi block into two numbers, forwarding it to Alice, according to the following Eq (2).

$$g^{Xk}\, mod\, P, mi * \beta^{k}\ mod\ P \qquad (2)$$

3. As the text blocks are encrypted, Bob can change k for the consequential blocks.
4. What Alice receives per encrypted block is a pair of integers(μ, λ), which has been obtained according to the following equations (3), (4) based on what was said previously.

$$\lambda = G^k \, mod \; P \tag{3}$$

$$\mu = mi * \beta^k \; mod \; P \tag{4}$$

5. Alice can decrypt the encrypted data according to the following Eq (5).
6.

$$mi = \lambda^{P-1-A} \; * \mu \, mod \; P \tag{5}$$

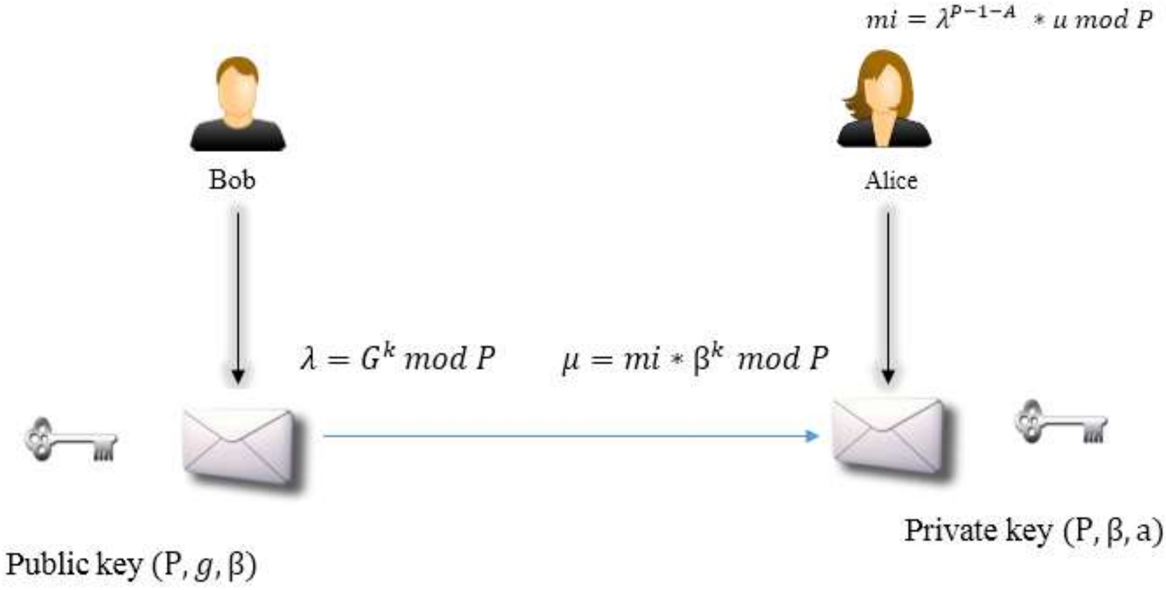
$mi = \lambda^{P-1-A} \; * u \, mod \; P$
Bob
Alice
$\lambda = G^k \, mod \; P$
$\mu = mi * \beta^k \; mod \; P$
Public key (P, g, β)
Private key (P, β, a)

Fig.11. Shows the ElGamal encryption algorithm.

ECC:
The Elliptic Curve Cryptography (ECC) is a public key encryption method, which has been designed based on an algebraic structure of elliptic curves on the finite fields. The use of elliptic curves in the encryption was proposed independently by Neal Koblitz and Victor S. Mille in 1985[240]–[246]. The public key encryption is based on the difficulties in some math problems. Earlier, systems based on the public key were considered safe by assuming the fact that finding two or more prime factors for a large integer was difficult. For elliptic curve-based algorithms, it is assumed that finding the DL from a random element of elliptic curves is impractical, given to a publicly known base point [247]. The size of the elliptic curve determines the difficulty of the problem. The main advantage promised by the ECC was a key with a smaller size, which means reduced storage and the required transport, such that a system of the elliptic curve can provide the same level of security as that of a system based on RSA with large and long key modules[248]–[250]. For today's encryption purposes, the elliptic curve is a flat curve, composed of satisfying points of the Eq (6).

$$y^2 = x^2 + ax + b \tag{6}$$

Cramer–Shoup:
The Cramer–Shoup is an ASYM cipher algorithm developed by Ronald Cramer and Victor Shoup in 1998 [251]–[253] The Cramer-Shoup algorithm has been created based on the premise of DH. In fact, the Cramer-Shoup is an extension of the ElGamal encryption algorithm, which is much more flexible than it. The Cramer–Shoup uses a universal one-way

hash for encryption, which in turn causes the ciphertext of the Cramer-Shoup to become twice that of the ElGamal. The Cramer–Shoup is comprised of three algorithms: the key generator, the encryption algorithm, and the decryption algorithm[254]–[256].

Key Generation:

- Alice generates an effective description of a cyclic group G of order q with two separate, random generators of $g1, g2$.
- Alice selects five random values $(x1, x2, y1, y2, z)$ from $\{0, \ldots\ldots, q-1\}$
- She computes$C = g1^{x1} g2^{x2}$,$d = g1^{y1} g2^{y2}$,$h = g1^{z}$.
- Alice publishes(c, d, h), in addition to the description of $(G, q, q1, g2)$ as her public key. Alice holds $(x1, x2, y1, y2, z)$ as her secret key. The user's groups can share the system between themselves.

Encryption:

Encrypting a message m to Alice under her public key$(G, q, g1, g2. c, d, h)$.

- m is converted into an element of G by Bob.
- Bob selects a random K from $\{0, \ldots\ldots, q-1\}$, and calculates:
- $u1 = g1^{k}$,$u2 = g2^{k}$
- $e = h^{k} m$
- $\Omega = h(u1, u2, e)$, Where H () is a global one-path hash function (or an accident-resistant cryptographic hash function, which is a stronger necessity).
- $v = c^{k} d^{k\Omega}$
- Bob forwards the ciphertext $(u1, u2, e, v)$to Alice.

Decryption

- Decrypting a ciphertext $(u1, u2, e, v)$ with Alice's secret key $(x1, x2, y1, y2, z)$
- $\Omega = h(u1, u2, e)$ Is computed by Alice, verifying that $u1^{x1} u2^{x2} (u1^{Y1}\, u2^{y2})^{\Omega} = v$. If this test was not successful, further decryption is canceled and the output is dismissed.
- Otherwise, the plaintext is computed by Alice as $m = \frac{e}{u1^{z}}$

The decryption step properly decrypts any correctly-formed ciphertext, since $u1^{z} = g1^{kz} = h^{k}$ and $m = \frac{e}{h^{k}}$

If the size of$G$ is smaller than the space of possible messages, then the Cramer–Shoup can be utilized in a hybrid cryptosystem to enhance the effectiveness of the extended messages.

DSA:

The Digital Signature Algorithm (DSA) is a United States Federal Government standard or FIPS for digital signatures. This algorithm was proposed by the National Institute of Standards and Technology (NIST) in August 1991 for the use as Digital Signature Standard (DSA) and was accepted in 1993 by FIPS (Federal Information Processing Standard) [257]–[260]. When a message is sent from an insecure channel, a properly-performed digital signature could be a reason for the message recipient in order to believe the claim made by the sender; in other words, the recipient can ensure that the same sender has signed the letter and it is not fake. Digital signatures are similar to manual signatures in many respects; doing digital signatures properly is much more difficult than a manual signature. The designs of digital signature files are based on ASYM encryption, and they should be done properly to be effective[261]–[263]. Digital signatures can also generate undeniable signatures; it means that the signatory cannot claim that s/he has not signed this letter with his/her signature, as long as the private key has been kept secret. But, when the personal key of a person in the network is exposed, or his/her signature validity period expires, s/he can deny his/her digital signature, though it retains its validity in this situation with its strong structure. Messages signed with digital signatures have the possibility to be presented as bit strings, such as e-mail, contracts or messages that are sent through other encryption rules[264]–[267]. The DSA is another variant of the ElGamal design.

Key Generation:

Key generation includes two steps. The first step includes parameter selection, which can be shared between different users of the system. The second phase calculates the public and private keys for the user.

- Selecting the Algorithm Parameters
  1. Selecting the hash function H
  2. Selecting the key length N, L. The size of N, L is the main criterion for the key encryption resistance.
  3. Select a primary bit N in such a way that $q.N \leq$ the output hash length.
  4. Select a primary bit 1 with module p in such a way that p *-1* is a multiple of q.
  5. Choose a number as $g = H(p-1)/\mathrm{q}$
  6. Algorithm parameters (p, q, g) may be shared among the users. A set of parameters is allocated to the keys per user.

- Allocating the Keys to the Users
  1. A set of parameters is allocated to the key per user. The second stage calculates the public and private keys for a distinct user.
  2. Selecting *x* with random methods.
  3. Selecting the $y = g^x$ residuum.
  4. The public key is (p, q, g, y) and the private key is *x*.

- Signature Generation Algorithm
  1. Generating the random key $k$should be eliminated after using for one time, not employed any more.
  2. Then the ordered signature pair (r, s) is calculated as the following Eq (7).
  3. $$r = (gkmodp)modqs = [k - 1(H(M) + xr)]\, mod\mathrm{q} \quad (7)$$
  4. (r, s) Is joined to the message M and is forwarded.

- The Correct Authentication Signature Algorithm
- The receptor receives $(r, s,)$, M and calculates the values according to following Eq (8), (9), (10) and (11).

$$\mathrm{W} = (\mathrm{s}')^{-1} \bmod \mathrm{q} \quad (8)$$

$$\mathrm{U1} = [\mathrm{H}(\mathrm{M}')\mathrm{w}] \bmod \mathrm{q} \quad (9)$$

$$\mathrm{U2} = [(\mathrm{r}')\mathrm{w}] \bmod \mathrm{q} \quad (10)$$

$$\mathrm{v} = [(\mathrm{g}^{\mathrm{u1}}\mathrm{y}^{\mathrm{u2}}) \bmod \mathrm{p}] \bmod \mathrm{q} \quad (11)$$

If $\mathrm{v} = \mathrm{r}$, the signature is valid.

Diffie–Hellman Key Exchange:

The DH key exchange protocol is an encryption protocol. Using the DH key exchange protocol, two people or two organizations can generate a shared key, not requiring any previous acquaintance, and they can exchange it through an insecure communication path. This protocol is the first practical method for exchanging the key in insecure communication paths, which solves the problem of key exchange in the encryption of SYM keys. This protocol was designed by Whitfield Diffie and Martin Hellman Ralph Merkle in 1976 and was published as a scientific paper. This protocol has been considered an important step in introducing and developing ASYM key encryption [268]–[270]. Fig (12) shows the DH key exchange algorithm. In the early proposed formula of this protocol, the modular arithmetic group of integers with a prime number p and the operator of prime number multiplications has been used. In this numerical group, a primary root is calculated, which is indicated by$g$.

The following steps are also shown in the opposite figure as below:

1. The relationship starter chooses a large given prime number and calls it p; then, the calculated value of g is exchanged between two sides.

2. Either side of the relationship chooses a given integer secretly, keeping them by itself, which are calleda, b.
3. Either side calculates a new value and calls them(A, B) using modular power operation and previous values of p and g, and the secret values of a, b, and forwards them to the other side.
4. The first and second sides calculate new values using p, g, a, B, as well as p , g , b, A values, respectively, and with the same modular power operation, which are same at either side, as indicated by the formula, and that value is the same as the shared code key.

Two issues should be addressed about this protocol are as follows:

- The values of a and b and the calculated shared value never pass directly through the communication channel. Other values p, g, A, and B pass through the communication channel and are available to others.

- The difficulty of solving the DL problem ensures that the values of a and b and the value of shared code key are practically not calculable by having the value of other numbers.

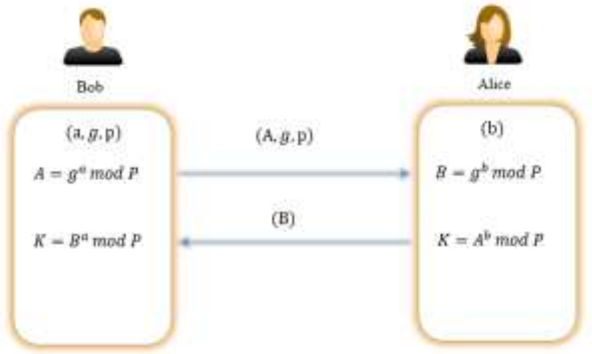

Fig. 12. Shows the stages for the DH key interaction algorithm [271].

5.3.2.Algorithms Designed Based on the FPN

This Sect discusses the types of algorithms based on FPN.

RSA:
The RSA encryption algorithm is among the first public algorithms used for a secure data transfer. RSA algorithm was invented in 1977 by Ron Rivest, A. Shamir, and Leonard Adleman and is widely used[37], [272]–[276]. The security of RSA algorithm results from the fact that no effective way is known for factorizing the prime numbers. It is proven that any number can be written as the product of some prime numbers. For example, the number 4200 can be decomposed into the following prime factors:

$$4200 = 2 * 2 * 2 * 3 * 5 * 5 * 7$$

In the above example, the numbers 2, 3, 5, 7 are prime numbers. In the RSA algorithm, each private and public key is made from two very large prime numbers. Breaking the key in the RSA requires finding these two prime numbers. For centuries, mathematicians have tried to find an effective algorithm for decomposing numbers into prime factors, but so far in no vain [277]–[281]. Generating public and private keys in the RSA is performed in four stages:

- Choose two very large prime numbers, P and$Q$.
- Calculate then$n = p * q$,$z = (p - 1) * (q - 1)$.
- Choose a number like D, such that it is a prime number in respect to$Z$
- Choose E in such a way that $e * d = 1\ mod\ z$

D and E Can be used for decryption and encryption, respectively. With respect to the used algorithm, the number of D can be kept as a private key, while E is exposed to the public. For encrypting the Mimessage, the sender calculates the $Ci = mi^e (mode\ n)$value for each Mi block, forwarding it to the sender. For decrypting the codes of message blocks, the sender is only required to calculate the value of $Ci = mi^d (mode\ n)$[282][283]**.**

Robin:
Robin Algorithm was developed by Michel Robin in 1979 [284]–[286]. The security of this algorithm, like that of the RSA, is based on the factorization of the big numbers. The main drawback of the Robin algorithm is the complexity in detecting a plain text from four possible roots in the decryption process based on the original text. To rightly detect the

message from the four possible roots, a buffer character with an empty space for filling is used before encryption, such that only one message will be the original message from the four possible messages after encryption[287]–[290]. In the following, the Robin method is used for security enhancement, which is performed in such a way that for encryption, a random number is used to secure the intended algorithm [291].
The Robin Algorithm is encrypted as follows:

- Step 1: the public and private keys are produced in the receiver side.
Two p and q numbers are produced using the Eq. (12).

$$4k + 3 \tag{12}$$

- The product of p and q is obtained by the Eq (13).

$$n = p * q \tag{13}$$

- The (n) public key is published generally and the (p, q) private key is kept in secret.

- Step 2: By creating the public and private keys in the side of the receiver and forwarding the public key to the sender, the Robin encryption process in the side of the sender is started.
Encryption of the message M is obtained using the Eq (14).

$$C = M2\, mod n\, 0 < M < n - 1 \tag{14}$$

- Step 3: By forwarding the encoded plaintext from the sender to the receiver, the Robin encryption process is performed by having the public key in the side of the receiver. Decrypting the encoded message C is performed by using the (p, q) private key.
S And R are calculated based on the formula (15), (16).

$$R = cp + 1/4 modp \tag{15}$$

$$S = cq + 1/4 modp \tag{16}$$

a And b are found from the following Eq (17).

$$a * q + b * q = 1 \tag{17}$$

- The main message from the four possible messages M_1, M_2, M_3, and $M_4$is obtained using the Eq (18), (19), (20) and (21).

$$M1 = (aps + bqR)\, modN \tag{18}$$

$$M2 = (N - M1) \tag{19}$$

$$M3 = (aps - bqR) modN \tag{20}$$

$$M4 = (N - M3) \tag{21}$$

- The main Mi (the clear M text), in which $i = 1, 2, 3, 4$ is selected according to the Eq (12).

6.Observations

The algorithm reviewed in the paper has been written in Table (4) based on their production year. Then, the algorithms presented in the Table have been studied in four aspects. The first aspect shows the kind of encryption and the algorithm structure, which can be SYM or ASYM. In the second aspect, the existing algorithms regarding their key length have been studied. The keys in the proposed algorithms have sizes of 64 to 2048 bits, for which we have used the “Variant option” for the ASYM algorithms, in which selecting the key length is in the hands of the algorithm user. The number of rounds has been studied in the third aspect, starting from 1 round to 48 rounds. We use the term “other” for other states. In the fourth aspect, we have studied the existing algorithms in terms of their block size. The blocks of the proposed algorithms vary from 64 to 4096 in size. We used “Variant state” for the

ASYM algorithms, as a selection of the block size is available for the algorithm users.

SYM algorithms are structurally divided into four categories: The first group is the algorithms designed based on the FN. These algorithms often use 128-, 192-, and 256-bit key lengths and 16 rounds and block sizes of 64 and 128 bits. The second group algorithms have been designed based on the type-3 FN. The Mars algorithm has been categorized in this group, which uses the 128-, 192-, and 256-bit key lengths, 32 round numbers, and the block size of 128 bits. The third group, designed based on the unbalanced FN, are RC2 and Skipjack algorithms. The Skipjack algorithm uses a key length of 80 and a block size of 128 bits. However, both algorithms use 16 round numbers. The fourth group of algorithms is those which have been designed based on the Nested FN. The SEED and DEAL algorithms are located within this group, which either uses different key lengths or round numbers but conforms to common block sizes. The fifth group is the algorithms which have been designed based on the SPN

The algorithms designed on that basis often use the 128-, 192-, and 156-bit key lengths and 128-bit block size, but each algorithm is different from others in the number of cycles. ASYM algorithms can be divided into two categories structurally. The first group is the algorithms categorized based on the difficulty in the factorization of integer numbers. Also, the RSA and Rabin algorithms are placed in this group. The second group is those algorithms that have been designed based on the difficulty in the DL, such as the El Gamal, DSA, and ECC. The latter uses linear algebra and DL, so it has been categorized as a DL. In general, due to their high flexibility, the ASYM algorithms allow the algorithm users to select the key length and the size of encryption blocks compared with the SYM algorithms, in which the round number of these algorithms is not more than one.

According to Table (5), before presenting the DES encryption algorithm in 1975, no software or hardware product had been found in the computer world for data encryption. When the DES algorithm was introduced to the world, the proponents of this technology were challenged. It can be said that encryption science started to flourish in that year in an academic, compiled, and purposeful manner. In 1976, the Diffi-Helman key interaction protocol was introduced, by which two individuals or organizations can produce a shared code key, interacting with it through an unsafe communication path. This protocol is the first practical method for interacting with the code key in unsafe communication paths, facilitating the problem of the code key in encrypting the SYM keys. This algorithm can be called the source of generating the ASYM algorithms. One year later, in 1977, Rivest, Adleman, and Shamir proposed the RSA encryption algorithm at the University of MIT. The RSA is the first reliable method among other encryption methods which has been considered one of the greatest advances in the encryption area. The RSA, widely used in electronic interactions, appears completely safe when used adequately with long keys. Between 1979 to 1984, public key algorithms, such as Rabin, were produced with Integer Factorization.

In 1985, Dr. Taher El-Gamal introduced the El Gamal algorithm for encryption of the public key, which was compatible with the RSA in terms of reliability and strength, having higher complexity and lower speed compared with it. In the same year, the ECC encryption algorithm was introduced by Neal Koblitz Victor S. Miller, who had more strength than other public-key algorithms. From 1987 to 1997, most produced algorithms were SYM -type algorithms. The introduction of the DSA algorithm in 1991, which was placed in the family of ASYM algorithms, was a turning point during this period. The DSA algorithm prevented any forgery and manipulation of the signatures or their denial in the legal and commercial documents.

The DSA makes it possible to change many hard-cover documents into electronic (soft copy) ones. The year 1998 can be considered as the apex for the formation of SYM algorithms, as in this year, the encryption fans were encouraged by Advanced Encryption Standard competition to produce alternative algorithms for the DES algorithm, such as MARS, Serpent, Crypton, DEAL, CAST-256, E2, RC6, Twofish, and Rijndael. Among these algorithms, only Rijndael managed to surpass its competitors, winning the competition. Today, it is known as the AES. From 1999 to 2003, various algorithms were introduced, but from 2000 on, most governments and countries have performed many actions to produce specific encryption algorithms. This action shows that the algorithms developed by companies, organizations, and governments are less trusted among the people; this has made the users of encryption algorithms ensure the transparent function of the algorithm.

The algorithms presented in Table (6) have been examined in terms of flexibility and change based on the users' requirements. All of the studied ASYM algorithms provide flexibility and variability for the users, making them determine the length of the algorithm key based on their processing power and required security. Instead, in the SYM algorithms, neither DES, ICE, SEED, MERCY, KASUMI, and Skipjack algorithms designed based on Feistel's structure, nor the Shark.3wayalgorithm which has been designed based on SPN, provide their users with this capability. Most algorithms provide this key length variability to their users, such that the users can use, at best, several types of key lengths and the number of rounds considered by the algorithm designer. The users have no right or authority to interfere with the type of key length and the number of rounds.

Table (7) shows that the SYM and ASYM algorithms were introduced in the 1970s. The algorithms, such as DES and New Data Seal, were introduced in this decade, benefiting from the FN structure. By designing the DH key interaction algorithm in this decade, a new era has been created for ASYM encryption. Then, well-known algorithms, like RSA and Rabin, were produced in these years. Most of the algorithms produced in the 1980s were ASYM, and the most important and influential ASYM algorithms have also been produced in this decade. The El Gamal and ECC algorithms use the discrete algorithm for producing an encryption system in this period. The 1990s can be considered the most brilliant period for SYM encryption, starting with the presentation of the DSA algorithm. Digital signatures have been greatly used in many fields like electronic documents and so on. In this decade, the DES algorithm has attracted much attention for finding an appropriate algorithm to replace the DES algorithm by holding the Advanced Encryption Standard competition. In this decade, we see that most organizations, companies, and governments were trying to produce encryption algorithms for personal and public uses. The 2000s can be considered the most balanced period for encryption. During this period, most of the produced algorithms were symmetric, and the FN and SPN structures have been used in their designs. From a general viewpoint, it can be seen that governments and organizations have produced different encryption algorithms.

With a general look at the history of cryptography, it can be perceived that as the encryption algorithms develop, the methods of breaking these algorithms are also developed and are being developed. All cryptographic algorithms can be cracked using Brute-Force attacks, but this attack is too time-consuming. But many other cryptographic algorithms can be attacked using the mathematical weaknesses employed in designing the encryption algorithm. The last column refers to these attacks. The algorithms produced by organizations and governments have always carried the doubt that there is a master key for breaking the coding algorithm by the algorithm producer.

7.Open Challenges

- The issues related to determining the key size: The efficiency of SYM algorithms compared with ASYM algorithms has made them more attractive for software developers; however, in some situations, developers may want to determine their software security arbitrarily, based on determining the parameter of key size, while the SYM algorithms do not allow this possibility for their users or developers. The inability to determine the key size in the SYM algorithms is considered a fundamental challenge in designing the SYM algorithms.

- The Issues Related to Key Exchange: The key exchange is used for securing an unsecured route between the two exchanging sides; after creating a secure path between the sender and the receiver, they use a cryptographic algorithm to ensure the security between them for sending information to each other. The collection of these activities has initially led the sender and receiver to use one algorithm for the key exchange and another for encryption.

- The Issues Related to Cryptographic Algorithms in a Distributed Environment: In the discussion of distributed systems, an important issue is the processes and their functions. A significant issue raised in the processes is multi-threaded ness, which in the distributed system has many advantages. Multi-threaded ness is used in the customer-service provider relationship, so customers and service providers are fully examined here. Virtualization is proposed in the case of processes running on heterogeneous systems in a distributed environment, and its different types are expressed. A most important feature of the processes in a distributed environment allows migration from one machine to another. However, suppose a process is encrypted for more security in a machine. In that case, it migrates in the encrypted form upon migration to another device, but it requires the source machine key to decrypt data for running the process. In this case, the code key must be shared between the machines. A critical challenge in this regard is the creation of a multi-key encryption algorithm to decrypt the encrypted data upon migration.

- The problem of using a One-Time Password (OTP) in the cryptographic algorithms in databases in the distributed systems: a single-use or disposable password is a code valid only for one login or transaction. Single-use passwords resolve many flaws of old codes (fixed codes). The most critical spot of a single-use password balance is its non-vulnerability in repetitive attacks. Using this method, a potential intruder who manages to acquire a single-use password during access to a service or a transaction is not able to abuse it any longer, saving it in the database because that password has expired; however, a critical problem appears when the user encrypts data and protects it with a single-use key. When the intended key parts are, the user cannot decrypt the encrypted data of the previous key using the new key assigned to it; thus, the presence of OTP is fundamental in cryptography to enhance security; however, the use of OTP for storing information is a real challenge in this area.

- High complexity: one of the main challenges in encryption algorithms is related to the high complexity of these algorithms; the designers of encryption algorithms focus on increasing security to be resistant to various attacks. However, these algorithms consume high energy due to their high complexity. Since the Internet of Things devices have energy limitations, these algorithms are unsuitable for this environment. There is a need to design lightweight encryption algorithms with low complexity that can withstand various attacks.

- Encryption with the ability to exchange keys: considering that encryption methods differ from key exchange methods, and before establishing communication, both parties need to exchange keys and then encrypt their data in the created channel. This study shows that no existing algorithms can exchange keys and are designed for one purpose. Due to the high complexity of these algorithms, it is necessary to create a cryptographic algorithm with key exchange capability to perform encryption and key exchange simultaneously.

7.Conclusion

This paper presents a detailed analysis of the SYM and ASYM algorithms based on various parameters. This study mainly aims to analyze the proposed algorithms in terms of structure, production year, key size, block size, number of rounds, flexibility, and algorithm developer and highlight the attacks that compromise the encryption algorithm. During the analysis, it was specified that all the examined algorithms are vulnerable to Brute-Force attacks. Some algorithms cause the algorithm to be vulnerable in the type of mathematics used in their design. The results of the analysis of the proposed algorithms show that the designing structure of the algorithms, such as the RSA developed by ordinary people with their structure expressed explicitly, allows the algorithm users to use a variable key size and block size, as well as a low number of rounds, along with flexibility, which is highly favorable among the users and these types of algorithms can be used to lightweight authentication in the internet of thing.

9.Future Work

The future works that the authors of the article want to do are as follows:

- Design of encryption algorithms with key exchange capability.
- Designing of low- complexity encryption algorithms.
- Developing cryptographic algorithms resistant to brute-force attacks.

Table 4
Shows the encryption algorithms in terms of their structure type, the key length, the number of rounds, and the block size.

NO	*Algorithm name*	*Structure*									*Key Size*																*Rounds of process*																	*Block Size*						
		Type																																																
		Symmetric	*Asymmetric*	*Feistel network*	*Type-3 Feistel network*	*unbalanced Feistel network*	*Nested Feistel network*	*Substitution-permutation network*	*Integer Factorization*	*Discrete logarithm*	64	40-64	80	96	40-128	128	128-192-256	56-112-168	128-160-192-224-256	256	128 to 320	32-448	8-1024		2048	Variant	1	6	8	8-9	10	11	12	10-11-12	12-14-16	12-16	18-24	16	20	32	48	128	other	8	64	96	128	32-64-128	4096	Variant
1	*DES*	✓		✓							✓																											✓							✓					
2	*New Data Seal*	✓		✓																					✓													✓									✓			
3	*Diffie-Hellman*		✓							✓																✓	✓																							✓
4	*RSA*		✓						✓																	✓	✓																							✓
5	*Rabin*		✓						✓																	✓	✓																							✓
6	*El Gamal*		✓							✓																✓	✓																							✓
7	*ECC*		✓							✓																✓	✓																							✓
8	*RC2*	✓				✓																	✓															✓							✓					
9	*DSA*		✓					✓		✓																✓	✓																							✓
10	*Blowfish*	✓		✓																		✓																✓							✓					
11	*RC5*	✓		✓																				✓																		✓					✓		
12	*3-Way*	✓						✓						✓																		✓														✓				
13	*CAST-128*	✓		✓											✓																					✓									✓					

14	*Shark*	✓						✓								✓												✓																	✓					
15	*ICE*	✓	✓								✓																											✓							✓					
16	*M6*	✓	✓									✓																			✓														✓					
17	*TDES*	✓	✓															✓																							✓				✓					
18	*Twofish*	✓	✓																																			✓									✓			
19	*RC6*	✓	✓													✓																							✓								✓			
20	*CAST-256*	✓	✓																✓																						✓						✓			
21	*SEED*	✓					✓									✓																						✓									✓			
22	*Skipjack*	✓				✓							✓																											✓					✓					
23	*E2*	✓	✓														✓																✓														✓			
24	*CS-Cipher*	✓	✓													✓													✓																✓					
25	*DEAL*	✓					✓																																				✓				✓			
26	*Crypton*	✓						✓																									✓														✓			
27	*Serpent*	✓						✓																																✓							✓			
28	*AES*	✓						✓																										✓													✓			
29	*MARS*	✓			✓																																			✓							✓			
30	*DFC*	✓	✓																																							✓		✓						
31	*Cramer–Shoup*	✓								✓																✓	✓																							✓
32	*M8*	✓	✓														✓			✓											✓														✓					
33	*UES*	✓	✓														✓																								✓						✓			
34	*Camellia*	✓	✓														✓																				✓										✓			
35	*KASUMI*	✓	✓													✓													✓																✓					
36	*Xenon*	✓	✓														✓																					✓									✓			
37	*Mercy*	✓	✓													✓												✓																					✓	
38	*Anubis*	✓						✓													✓																						✓				✓			
39	*Q*	✓						✓									✓													✓																	✓			
40	*ARIA*	✓						✓									✓																		✓												✓			

Table 5

Shows the proposed encryption algorithms in terms of the production year and the algorithm-designing organization.

NO	*Algorithm name*	*Structure*									*Years*																*Developer*	
		Type		Feistel network	Type-3 Feistel network	unbalanced Feistel network	Nested Feistel network	Substitution-permutation network	Integer Factorization	Discrete logarithm	1970	1975	1976	1977	1979	1985	1987	1991	1993	1994	1996	1997	1998	1999	2000	2003	Organization	Personal
		Symmetric	Asymmetric																									
1	DES	✓		✓							✓																✓	
2	New Data Seal	✓		✓								✓															✓	
3	Diffie-Hellman		✓							✓			✓															✓
4	RSA		✓						✓					✓														✓
5	Rabin		✓						✓						✓													
6	El Gamal		✓							✓						✓												✓
7	ECC		✓							✓						✓												✓
8	RC2	✓				✓											✓											✓
9	DSA		✓							✓								✓									✓	
10	Blowfish	✓		✓															✓									✓
11	RC5	✓		✓																✓								✓
12	3-Way	✓						✓												✓								✓
13	CAST-128	✓		✓																	✓							✓
14	Shark	✓						✓													✓							✓
15	ICE	✓		✓																		✓						✓
16	M6	✓		✓																		✓						✓
17	TDES	✓		✓																			✓				✓	
18	Twofish	✓		✓																			✓					✓
19	RC6	✓		✓																			✓					✓
20	CAST-256	✓		✓																			✓					✓
21	SEED	✓					✓																✓				✓	
22	Skipjack	✓				✓																	✓				✓	
23	E2	✓		✓																			✓				✓	
24	CS-Cipher	✓		✓																			✓					✓

25	DEAL	✓					✓																✓					✓
26	Crypton	✓						✓															✓					✓
27	Serpent	✓						✓															✓					✓
28	AES	✓						✓															✓					✓
29	MARS	✓			✓																		✓				✓	
30	DFC	✓		✓																			✓					
31	Cramer–Shoup	✓								✓													✓					
32	M8	✓			✓																			✓				✓
33	UES	✓		✓																				✓				✓
34	Camellia	✓		✓																					✓		✓	
35	KASUMI	✓		✓																					✓		✓	
36	Xenon	✓		✓																					✓			✓
37	Mercy	✓		✓																					✓			✓
38	Anubis	✓						✓																	✓			✓
39	Q	✓						✓																	✓			✓
40	ARIA	✓						✓																		✓		✓

Table 6
Shows the proposed algorithms in terms of flexibility and variability.

	Algorithms	Flexible	Modification	Feature
1	DES	No	None	The structure of DES doesn't support any modifications.
2	New Data Seal	No	None	The structure of New Data Seal doesn't support any modifications.
3	Diffie-Hellman	Yes	Yes	In DH selecting the key length is in the hands of the algorithm user.
4	RSA	Yes	Yes	In RSA selecting the key length is in the hands of the algorithm user.
5	Rabin	Yes	Yes	In Rabin selecting the key length is in the hands of the algorithm user.
6	El Gamal	Yes	Yes	In El Gamal selecting the key length is in the hands of the algorithm user.
7	ECC	Yes	Yes	In ECC selecting the key length is in the hands of the algorithm user.
8	RC2	Yes	Yes	In RC2 key size can vary from 8 to 1024
9	DSA	Yes	Yes	In DSA selecting the key length is in the hands of the algorithm user.
10	Blowfish	Yes	Yes	Blowfish key length must be multiples of 32 bits
11	RC5	Yes	Yes	In RC5 key size can vary from 8 to 1024
12	3-Way	No	None	The structure of 3-Way doesn't support any modifications.
13	CAST-128	Yes	Yes	The number of rounds in the CAST-128 algorithm varies from 12 to 16. key may vary between 40 to 128 in length
14	Shark	No	None	The structure of Shark doesn't support any modifications.
15	ICE	No	None	The structure of ICE doesn't support any modifications.
16	M6	Yes	Yes	In M6 key sizes are 40 bits by default, but they can be continued up to 64 bits.
17	TDES	Yes	Yes	Which the key size in TDES can be 168, 112, or 56 bits
18	Twofish	Yes	Yes	Two fish keys, other than the default sizes, are always padded with "0" bits up to the next default
19	RC6	Yes	Yes	RC6 has a variable key length and can be extended to 2048 bits
20	CAST-256	Yes	Yes	Which the key size in CAST-256 can be 128, 160, 192, 224, or 256 bits
21	SEED	No	None	The structure of SEED doesn't support any modifications.
22	Skipjack	No	None	The structure of Skipjack doesn't support any modifications.
23	E2	Yes	Yes	In the E2 can be of three key-length 128, 192, or 255 modes.
24	CS-Cipher	Yes	Yes	In CS-Cipher, the key length for data encryption may vary between 0 to 128, such that it should be a multiple of 8.
25	DEAL	Yes	Yes	For encryption with the key size of 128 to 192, the DEAL algorithm uses 6 rotation rounds, for encryption with the length of 256 uses 8 rounds.
26	Crypton	Yes	Yes	In the Crypton can be of three key-length 128, 192, or 255 modes.
27	Serpent	Yes	Yes	Serpent keys are always padded to 256 bits. The padding consists of a "1" bit followed by "0" bits.

28	AES	Yes	Yes	In the AES can be of three key-length 128, 192, or 255 modes.
29	MARS	Yes	Yes	MARS operates with variable key lengths, but the key length must be multiples of 32 bits.
30	DFC	Yes	Yes	Which the key size in DFC can be 128, 192, or 256.
31	Cramer–Shoup	Yes	Yes	In Cramer–Shoup selecting the key length is in the hands of the algorithm user.
32	M8	No	None	The structure of M8 doesn’t support any modifications.
33	UES	Yes	Yes	In the UES can be of three key-length 128, 192, or 255 modes.
34	Camellia	Yes	Yes	Which the key size in Camellia can be 128, 192, or 256.
35	KASUMI	No	None	The structure of KASUMI doesn’t support any modifications.
36	Xenon	Yes	Yes	Which the key size in Xenon can be 128, 192, or 256.
37	Mercy	No	None	The structure of Mercy doesn’t support any modifications.
38	Anubis	Yes	Yes	In Anubis key size can vary from 128 to 320.
39	Q	Yes	Yes	Q algorithm uses 8 or 9 rounds for encryption of 128-bit blocks with the key length of 128, 192 or 256
40	ARIA	Yes	Yes	The ARIA uses the key size of 128, 192, 256 bits. The number of rounds can be 12, 14, and 16, depending on the key length.

Table 7
Shows the encryption algorithms in terms of structure, the key length, the number of rounds, the block size, and in terms of the production decade and the algorithm-designing algorithm. It also shows the weakness of the proposed algorithms against.

No	Algorithm name	Created By	Decade	Cipher type	Algorithm Structure	Key Length(bit)	Rounds Of process	Block Size(bit)	Attacks
1	*DES*	*IBM*	*1970*	*Symmetric*	*Feistel network*	*56 bits* *(+8 parity bits)*	*16*	*64 bits*	*Brute Force Attack*
2	*New Data Seal*	*IBM*	*1970*	*Symmetric*	*Feistel network*	*2048 bits*	*16*	*128 bits*	*Slide Attack*
3	*Diffie-Hellman*	*Whitfield Diffie* *Martin Hellman* *Ralph Merkle*	*1970*	*Asymmetric*	*Discrete logarithm*	*Variant*	*1*	*Variant*	*Man-In-The-Middle Attack*
4	*RSA*	*Rivest* *Shamir* *Adleman*	*1970*	*Asymmetric*	*Integer Factorization*	*Variant*	*1*	*Variant*	*Shor's Algorithm*
5	*Rabin*	*Michael Rabin*	*1970*	*Asymmetric*	*Integer Factorization*	*Variant*	*1*	*Variant*	*Brute Force Attack*
6	*El Gamal*	*Taher Elgamal*	*1980*	*Asymmetric*	*Discrete logarithm*	*Variant*	*1*	*Variant*	*Pohlig–Hellman Algorithm*
7	*ECC*	*Neal Koblitz* *Victor S. Miller*	*1980*	*Asymmetric*	*Algebraic structure of elliptic curves*	*Variant*	*1*	*Variant*	*Brute Force Attack*
8	*RC2*	*Ron Rivest*	*1980*	*Symmetric*	*unbalanced Feistel network*	*8–1024 bits, in steps of 8 bits; default 64 bits*	*16 of type MIXING, 2 of type MASHING*	*64 bits*	*Related-Key Attack*
9	*DSA*	*NIST*	*1990*	*Asymmetric*	*Discrete logarithm*	*Variant*	*1*	*Variant*	*Brute Force Attack*
10	*Blowfish*	*Bruce Schneier*	*1990*	*Symmetric*	*Feistel network*	*32–448 bits*	*16*	*64 bits*	*Differential Cryptanalysis*
11	*RC5*	*Ron Rivest*	*1990*	*Symmetric*	*Feistel network*	*0 to 2040 bits (128 suggested)*	*1-255(12suggested originally)*	*32, 64 or 128 bits (64 suggested)*	*Differential Cryptanalysis*
12	*3-Way*	*Joan Daemen*	*1990*	*Symmetric*	*Substitution-permutation network*	*96 bits*	*11*	*96 bits*	*Related-Key Attack*
13	*CAST-128*	*Carlisle Adams and Stafford Tavares*	*1990*	*Symmetric*	*Feistel network*	*40 to 128 bits*	*12 or 16*	*64 bits*	*Man-In-The-Middle Attack*
14	*Shark*	*Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bosselaers, Erik De Win*	*1990*	*Symmetric*	*Substitution permutation network*	*128 bits*	*6*	*64 bits*	*Interpolation Attack*
15	*ICE*	*Matthew Kwan*	*1990*	*Symmetric*	*Feistel network*	*64 bits (ICE),* *64×n bits (ICE-n)*	*16 (ICE), 8 (Thin-ICE), 16×n*	*64 bits*	*Differential Cryptanalysis*

							(ICE-n)		
16	*M6*	*Hitachi*	*1990*	*Symmetric*	*Feistel network*	*40-64 bits*	*10*	*64 bits*	*Mod n Cryptanalysis*
17	*TDES*	*IBM*	*1990*	*Symmetric*	*Feistel network*	*168, 112 or 56 bits (keying option 1, 2, 3 respectively)*	*48*	*64 bits*	*Brute Force Attack*
18	*Twofish*	*Bruce Schneier*	*1990*	*Symmetric*	*Feistel network*	*128, 192 or 256 bits*	*16*	*128 bits*	*Truncated differential cryptanalysis*
19	*RC6*	*Ron Rivest, Matt Robshaw, Ray Sidney, Yiqun Lisa Yin*	*1990*	*Symmetric*	*Feistel network*	*128, 192, or 256 bits*	*20*	*128 bits*	*Brute force Attack*
20	*CAST-256*	*Carlisle Adams, Stafford Tavares, Howard Heys, Michael Wiener*	*1990*	*Symmetric*	*Feistel network*	*128, 160, 192, 224, or 256 bits*	*48*	*128 bits*	*Brute Force Attack*
21	*SEED*	*KISA*	*1990*	*Symmetric*	*Nested Feistel network*	*128 bits*	*16*	*128 bits*	*Brute Force Attack*
22	*Skipjack*	*NSA*	*1990*	*Symmetric*	*unbalanced Feistel network*	*80 bits*	*32*	*64 bits*	*Impossible Differential Cryptanalysis*
23	*E2*	*NTT*	*1990*	*Symmetric*	*Feistel network*	*128, 192, or 256 bits*	*12*	*128 bits*	*Brute Force Attack*
24	*CS-Cipher*	*Jacques Stern and Serge Vaudenay*	*1990*	*Symmetric*	*Feistel network*	*128 bits*	*8*	*64 bits*	*Brute Force Attack*
25	*DEAL*	*Lars Knudsen*	*1990*	*Symmetric*	*Nested Feistel network*	*128, 192 or 256 bits*	*6 (128- and 192-bit) or 8 (256-bit)*	*128 bits*	*Brute Force Attack*
26	*Crypton*	*Chae Hoon Lim*	*1990*	*Symmetric*	*Substitution-permutation network*	*128, 192, or 256 bits*	*12*	*128 bits*	*Brute Force Attack*
27	*Serpent*	*Ross Anderson, Eli Biham, Lars Knudsen*	*1990*	*Symmetric*	*Substitution-permutation network*	*128, 192 or 256 bits*	*32*	*128 bits*	*XSL Attack*
28	*AES*	*Vincent Rijmen, Joan Daemen*	*1990*	*Symmetric*	*Substitution-permutation network*	*128, 192 or 256 bits*	*10, 12 or 14 (depending on the key size)*	*128 bits*	*XSL Attack*
29	*MARS*	*IBM*	*1990*	*Symmetric*	*Type-3 Feistel network*	*128, 192, or 256 bits*	*32*	*128 bits*	*Brute Force Attack*
30	*DFC*	*Jacques Stern, Serge Vaudenay*	*1990*	*Symmetric*	*Feistel network*	*128, 192, or 256 bits*	*128 bits*	*8*	*Differential Cryptanalysis*

31	*Cramer–Shoup*	*Ronald Cramer Victor Shoup*	*1990*	*Asymmetric*	*Discrete logarithm*	*Variant*	*1*	*Variant*	*Brute Force Attack*
32	*M8*	*Hitachi*	*1990*	*Symmetric*	*Feistel network*	*256*	*10*	*64 bits*	*Brute Force Attack*
33	*UES*	*Helena Handschuh, Serge Vaudenay*	*1990*	*Symmetric*	*Feistel network*	*128, 192, or 256 bits*	*48 DES-equivalent rounds*	*128 bits*	*Brute Force Attack*
34	*Camellia*	*Mitsubishi Electric, NTT*	*2000*	*Symmetric*	*Feistel network*	*128, 192 or 256 bits*	*18 or 24*	*128 bits*	*Brute Force Attack*
35	*KASUMI*	*Mitsubishi Electric*	*2000*	*Symmetric*	*Feistel network*	*128 bits*	*8*	*64 bits*	*Impossible Differential Attack*
36	*Xenon*	*Chang-Hyi Lee*	*2000*	*Symmetric*	*Feistel network*	*128, 192, or 256 bits*	*16*	*128 bits*	*Brute Force Attack*
37	*Mercy*	*Paul Crowley*	*2000*	*Symmetric*	*Feistel network*	*128 bits*	*6*	*4096 bits*	*Differential cryptanalysis*
38	*Anubis*	*Vincent Rijmen and Paulo S. L. M. Barreto*	*2000*	*Symmetric*	*Substitution-permutation network*	*128 to 320 bits in steps of 32 bits*	*at least 12 (for 128-bit keys), plus one per additional 32 key bits*	*128 bits*	*Brute Force Attack*
39	*Q*	*Leslie McBride*	*2000*	*Symmetric*	*Substitution-permutation network*	*128, 192, or 256 bits*	*8 or 9*	*128 bits*	*Linear Cryptanalysis*

References

[1] L. D. Jackel, D. Sharman, C. E. Stenard, B. I. Strom, and D. Zuckert, “Optical character recognition for self-service banking,” *T Tech. J.*, vol. 74, no. 4, pp. 16–24, Jul. 1995, doi: 10.1002/j.1538-7305.1995.tb00189.x.

[2] Y. Iano, I. T. Lima, H. J. Loschi, T. C. Lustosa, O. S. Mesquita, and A. Moretti, “Sustainable computing and communications: Internet broadband network of things applied to intelligent education,” in *2015 International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, May 2015, pp. 1–7.

[3] E. Ramírez-Llanos and S. Martínez, “Distributed and Robust Fair Optimization Applied to Virus Diffusion Control,” *IEEE Trans. Netw. Sci. Eng.*, vol. 4, no. 1, pp. 41–54, Jan. 2017, doi: 10.1109/TNSE.2016.2614751.

[4] M. S. Haghighi, S. Wen, Y. Xiang, B. Quinn, and W. Zhou, “On the Race of Worms and Patches: Modeling the Spread of Information in Wireless Sensor Networks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 12, pp. 2854–2865, Dec. 2016, doi: 10.1109/TIFS.2016.2594130.

[5] A. Sengupta, S. Bhadauria, and S. P. Mohanty, “TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling With Optimal Loop Unrolling Factor During High Level Synthesis,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 36, no. 4, pp. 655–668, Apr. 2017, doi: 10.1109/TCAD.2016.2597232.

[6] O. Nasser, S. AlThuhli, M. Mohammed, R. AlMamari, and F. Hajamohideen, “An investigation of backdoors implication to avoid regional security impediment,” in *2015 Global Conference on Communication Technologies (GCCT)*, Apr. 2015, pp. 409–412. doi: 10.1109/GCCT.2015.7342695.

[7] R. Fotohi, Y. Ebazadeh, and M. S. Geshlag, “A New Approach for Improvement Security against DoS Attacks in Vehicular Ad-hoc Network,” *arXiv*, 2020, doi: 10.14569/ijacsa.2016.070702.

[8] Y. Salami, V. Khajehvand, and E. Zeinali, “E3C: A Tool for Evaluating Communication and Computation Costs in Authentication and Key Exchange Protocol,” 2022, doi: 10.48550/ARXIV.2212.03308.

[9] Y. Salami, Y. Ebazadeh, and V. Khajehvand, “CE-SKE: cost-effective secure key exchange scheme in Fog Federation,” *Iran J. Comput. Sci.*, vol. 4, no. 3, pp. 1–13, 2021.

[10] Y. Salami and V. Khajehvand, “LSKE: Lightweight Secure Key Exchange Scheme in Fog Federation,” *Complexity*, vol. 2021, p. 4667586, 2021.

[11] M. D. Grammatikakis *et al.*, “Security in MPSoCs: A NoC Firewall and an Evaluation Framework,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 34, no. 8, pp. 1344–1357, Aug. 2015, doi: 10.1109/TCAD.2015.2448684.

[12] W. Noonan and I. Dubrawsky, *Firewall Fundamentals*. Cisco Press, 2006.

[13] M. Liu, W. Dou, S. Yu, and Z. Zhang, “A Decentralized Cloud Firewall Framework with Resources Provisioning Cost Optimization,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 3, pp. 621–631, Mar. 2015, doi: 10.1109/TPDS.2014.2314672.

[14] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, “Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices,” *ACM Comput. Surv.*, vol. 48, no. 1, pp. 12:1--12:41, Sep. 2015, doi: 10.1145/2808691.

[15] I. Butun, S. D. Morgera, and R. Sankar, “A Survey of Intrusion Detection Systems in Wireless Sensor Networks,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 266–282, 2014, doi: 10.1109/SURV.2013.050113.00191.

[16] E. Kabir, J. Hu, H. Wang, and G. Zhuo, “A novel statistical technique for intrusion detection systems,” *Futur. Gener. Comput. Syst.*, doi: http://dx.doi.org/10.1016/j.future.2017.01.029.

[17] A. L. Buczak and E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[18] A. Sahai, “Computing on Encrypted Data,” in *Information Systems Security: 4th International Conference, ICISS 2008, Hyderabad, India, December 16-20, 2008. Proceedings*, R. Sekar and A. K. Pujari, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 148–153. doi: 10.1007/978-3-540-89862-7_12.

[19] A. J. Menezes, P. C. Van Oorschot, and S. a. Vanstone, *Handbook of Applied Cryptography*, vol. 106. 1997. doi: 10.1.1.99.2838.

[20] L. Yao, C. Yuan, J. Qiang, S. Feng, and S. Nie, “An asymmetric color image encryption method by using deduced gyrator transform,” *Opt. Lasers Eng.*, vol. 89, pp. 72–79, 2017, doi: http://doi.org/10.1016/j.optlaseng.2016.06.006.

[21] E. Fujisaki and T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes," in *Advances in Cryptology --- CRYPTO' 99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15--19, 1999 Proceedings*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 537–554. doi: 10.1007/3-540-48405-1_34.

[22] G. J. Simmons, "Symmetric and Asymmetric Encryption," *ACM Comput. Surv.*, vol. 11, no. 4, pp. 305–330, Dec. 1979, doi: 10.1145/356789.356793.

[23] N. Li, "Asymmetric Encryption," in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, p. 142. doi: 10.1007/978-0-387-39940-9_1485.

[24] N. Jayapandian, A. M. J. M. Z. Rahman, S. Radhikadevi, and M. Koushikaa, "Enhanced cloud security framework to confirm data security on asymmetric and symmetric key encryption," in *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, Feb. 2016, pp. 1–4. doi: 10.1109/STARTUP.2016.7583904.

[25] Ahmad, K. M. R. Alam, H. Rahman, and S. Tamura, "A comparison between symmetric and asymmetric key encryption algorithm based decryption mixnets," in *2015 International Conference on Networking Systems and Security (NSysS)*, Jan. 2015, pp. 1–5. doi: 10.1109/NSysS.2015.7043532.

[26] J. Shimeall and J. M. Spring, "Chapter 8 - Resistance Strategies: Symmetric Encryption," in *Introduction to Information Security*, T. J. Shimeall and J. M. Spring, Eds. Boston: Syngress, 2014, pp. 155–186. doi: http://doi.org/10.1016/B978-1-59749-969-9.00008-0.

[27] L. Adleman, "A subexponential algorithm for the discrete logarithm problem with applications to cryptography," in *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, Oct. 1979, pp. 55–60. doi: 10.1109/SFCS.1979.2.

[28] J. L. Camenisch, J.-M. Piveteau, and M. A. Stadler, "Blind signatures based on the discrete logarithm problem," in *Advances in Cryptology --- EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9--12, 1994 Proceedings*, A. De Santis, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 428–432. doi: 10.1007/BFb0053458.

[29] P. Smart, "The Discrete Logarithm Problem on Elliptic Curves of Trace One," *J. Cryptol.*, vol. 12, no. 3, pp. 193–196, 1999, doi: 10.1007/s001459900052.

[30] H. M. Ying and N. Kunihiro, "Cold Boot Attack Methods for the Discrete Logarithm Problem," in *2016 Fourth International Symposium on Computing and Networking (CANDAR)*, Nov. 2016, pp. 154–160. doi: 10.1109/CANDAR.2016.0037.

[31] M. Lee *et al.*, "Design and implementation of an efficient fair off-line e-cash system based on elliptic curve discrete logarithm problem," *J. Commun. Networks*, vol. 4, no. 2, pp. 81–89, Jun. 2002, doi: 10.1109/JCN.2002.6596898.

[32] K. Nyberg and R. A. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem," in *Advances in Cryptology --- EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9--12, 1994 Proceedings*, A. De Santis, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 182–193. doi: 10.1007/BFb0053434.

[33] A. Obtu\lowicz, "On P Systems with Active Membranes Solving the Integer Factorization Problem in a Polynomial Time," in *Multiset Processing: Mathematical,Computer Science, and Molecular Computing Points of View*, C. S. Calude, G. P\uAun, G. Rozenberg, and A. Salomaa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 267–285. doi: 10.1007/3-540-45523-X_14.

[34] C. P. Sah, K. Jha, and S. Nepal, "Zero-knowledge proofs technique using integer factorization for analyzing robustness in cryptography," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2016, pp. 638–642.

[35] H. M. Elkamchouchi, M. E. Nasr, and R. Esmail, "New public key encryption techniques based on generalized discrete logarithm, integer factorization and double integer factorization problems," in *Proceedings. ICCEA 2004. 2004 3rd International Conference on Computational Electromagnetics and Its Applications, 2004.*, Nov. 2004, pp. 561–564. doi: 10.1109/ICCEA.2004.1459417.

[36] B. Jansen and K. Nakayama, "Neural networks following a binary approach applied to the integer prime-factorization problem," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Jul. 2005, vol. 4, pp. 2577–2582 vol. 4. doi: 10.1109/IJCNN.2005.1556309.

[37] L. Rivest, a Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key

cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978, doi: 10.1145/359340.359342.

[38] A. Biryukov, “Substitution--Permutation (SP) Network,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg, Ed. Boston, MA: Springer US, 2005, p. 602. doi: 10.1007/0-387-23483-7_420.

[39] R. Karri, G. Kuznetsov, and M. Goessel, “Parity-Based Concurrent Error Detection of Substitution-Permutation Network Block Ciphers,” in *Cryptographic Hardware and Embedded Systems - CHES 2003: 5th International Workshop, Cologne, Germany, September 8--10, 2003. Proceedings*, C. D. Walter, Ç. K. Koç, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 113–124. doi: 10.1007/978-3-540-45238-6_10.

[40] H. M. Heys and S. E. Tavares, “Cryptanalysis of tree-structured substitution-permutation networks,” *Electron. Lett.*, vol. 29, no. 1, pp. 40-, Jan. 1993, doi: 10.1049/el:19930026.

[41] C. M. Adams, “A Formal and Practical Design Procedure for Substitution-permutation Network Cryptosystems,” Queen’s University, Kingston, Ont., Canada, Canada, 1990.

[42] H. M. Heys and S. E. Tavares, “Substitution-permutation networks resistant to differential and linear cryptanalysis,” *J. Cryptol.*, vol. 9, no. 1, pp. 1–19, 1996, doi: 10.1007/BF02254789.

[43] E. M. B. Albassal and A. M. A. Wahdan, “Genetic algorithm cryptanalysis of the basic substitution permutation network,” in *2003 46th Midwest Symposium on Circuits and Systems*, Dec. 2003, vol. 1, pp. 471-475 Vol. 1. doi: 10.1109/MWSCAS.2003.1562320.

[44] T. Baignères and S. Vaudenay, “Proving the Security of AES Substitution-Permutation Network,” in *Selected Areas in Cryptography: 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, B. Preneel and S. Tavares, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 65–81. doi: 10.1007/11693383_5.

[45] J. B. Kam and G. I. Davida, “Structured Design of Substitution-Permutation Encryption Networks,” *IEEE Trans. Comput.*, vol. C–28, no. 10, pp. 747–753, Oct. 1979, doi: 10.1109/TC.1979.1675242.

[46] D. Andelman and J. Reeds, “On the cryptanalysis of rotor machines and substitution - permutation networks,” *IEEE Trans. Inf. Theory*, vol. 28, no. 4, pp. 578–584, Jul. 1982, doi: 10.1109/TIT.1982.1056523.

[47] Z.-G. Chen and S. E. Tavares, “Towards Provable Security of Substitution-Permutation Encryption Networks,” in *Selected Areas in Cryptography: 5th Annual International Workshop, SAC’98 Kingston, Ontario, Canada, August 17--18, 1998 Proceedings*, S. Tavares and H. Meijer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 43–56. doi: 10.1007/3-540-48892-8_4.

[48] Y. Dodis and P. Puniya, “Feistel Networks Made Public, and Applications,” in *Advances in Cryptology - EUROCRYPT 2007: 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007. Proceedings*, M. Naor, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 534–554. doi: 10.1007/978-3-540-72540-4_31.

[49] L. Dong, Y. Wang, W. Wu, and J. Zou, “Known-key distinguishers on 15-round 4-branch type-2 generalised Feistel networks with single substitution–permutation functions and near-collision attacks on its hashing modes,” *IET Inf. Secur.*, vol. 9, no. 5, pp. 277–283, 2015, doi: 10.1049/iet-ifs.2014.0402.

[50] A. Bogdanov and K. Shibutani, “Analysis of 3-line generalized Feistel networks with double SD-functions,” *Inf. Process. Lett.*, vol. 111, no. 13, pp. 656–660, 2011, doi: http://dx.doi.org/10.1016/j.ipl.2011.04.002.

[51] O. Kara, “Square reflection cryptanalysis of 5-round Feistel networks with permutations,” *Inf. Process. Lett.*, vol. 113, no. 19–21, pp. 827–831, 2013, doi: http://doi.org/10.1016/j.ipl.2013.08.001.

[52] A. Bogdanov, “On the differential and linear efficiency of balanced Feistel networks,” *Inf. Process. Lett.*, vol. 110, no. 20, pp. 861–866, 2010, doi: http://dx.doi.org/10.1016/j.ipl.2010.07.016.

[53] D. Dachman-Soled, J. Katz, and A. Thiruvengadam, “10-Round Feistel is Indifferentiable from an Ideal Cipher,” in *Advances in Cryptology -- EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, M. Fischlin and J.-S. Coron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 649–678. doi: 10.1007/978-3-662-49896-5_23.

[54] M. Ebrahim, S. Khan, and U. Bin Khalid, “Symmetric Algorithm Survey: A Comparative Analysis,” *Int. J. Comput. Appl.*, vol. 61, no. 20, pp. 975–8887, 2013.

[55] G. Singh, “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security,” *Int. J. Comput. Appl.*, vol. 67, no. 19, pp. 975–8887, 2013.

[56] T. Gunasundari and K. Elangovan, “A Comparative Survey on Symmetric Key Encryption Algorithms,” *Int. J. Comput. Sci. Mob. Appl.*, vol. 2, no. 2, pp. 78–83, 2014.

[57] M. Agrawal and P. Mishra, “A comparative survey on symmetric key encryption techniques,” *Intern. J. Comput. Sci. Eng.*, vol. 4, no. 5, pp. 877–882, 2012.

[58] S. C. D. E, “A Survey on Symmetric Key Encryption Algorithms,” *Int. J. Comput. Sci. Mob. Appl.*, vol. 2, no. 4, pp. 475–477, 2014.

[59] M. Agrawal and P. Mishra, “A comparative survey on symmetric key encryption techniques,” *Int. J. Comput. Sci. Eng.*, vol. 4, no. 5, p. 877, 2012.

[60] A. Biryukov, “Slide Attack,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 1221–1222. doi: 10.1007/978-1-4419-5906-5_617.

[61] S. Furuya, “Slide Attacks with a Known-Plaintext Cryptanalysis,” in *Proceedings of the 4th International Conference Seoul on Information Security and Cryptology*, 2002, pp. 214–225.

[62] L. R. Knudsen and M. J. B. Robshaw, “Brute Force Attacks,” in *The Block Cipher Companion*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 95–108. doi: 10.1007/978-3-642-17342-4_5.

[63] K. Apostol, *Brute-force Attack*. SaluPress, 2012.

[64] M. Tiwari, T. Sharma, P. Sharma, S. Jindal, and Priyanshu, “Prevention of Man in the Middle Attack by Using Honeypot,” in *Proceedings of International Conference on Advances in Computing*, 2012, pp. 593–600.

[65] M. Brooks and B. Yang, “A Man-in-the-Middle Attack Against OpenDayLight SDN Controller,” in *Proceedings of the 4th Annual ACM Conference on Research in Information Technology*, 2015, pp. 45–49. doi: 10.1145/2808062.2808073.

[66] S. M. Hamdi, S. T. Zuhori, F. Mahmud, and B. Pal, “A Compare between Shor’s quantum factoring algorithm and General Number Field Sieve,” in *2014 International Conference on Electrical Engineering and Information Communication Technology*, Apr. 2014, pp. 1–6. doi: 10.1109/ICEEICT.2014.6919115.

[67] F. Li, W. Zhang, C. Xu, and J. Feng, “A Period-Finding Method for Shor?s Algorithm,” in *2007 International Conference on Computational Intelligence and Security (CIS 2007)(CIS)*, 2007, vol. 00, pp. 778–780. doi: 10.1109/CIS.2007.28.

[68] M. J. Nene and G. Upadhyay, “Shor’s Algorithm for Quantum Factoring,” in *Advanced Computing and Communication Technologies*, 2016, pp. 325–331.

[69] J. A. Thiong Ly, “A serial version of the Pohlig-Hellman Algorithm for computing discrete logarithms,” *Appl. Algebr. Eng. Commun. Comput.*, vol. 4, no. 1, pp. 77–80, Mar. 1993, doi: 10.1007/BF01270401.

[70] E. Teske, “The Pohlig---Hellman Method Generalized for Group Structure Computation,” *J. Symb. Comput.*, vol. 27, no. 6, pp. 521–534, Jun. 1999, doi: 10.1006/jsco.1999.0279.

[71] A. Biryukov, “Related Key Attack,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 1040–1041. doi: 10.1007/978-1-4419-5906-5_609.

[72] P. H. Nguyen, M. J. B. Robshaw, and H. Wang, “On Related-Key Attacks and KASUMI: The Case of A5/3,” in *Progress in Cryptology -- INDOCRYPT 2011*, 2011, pp. 146–159.

[73] M. Albrecht and C. Cid, “Algebraic Techniques in Differential Cryptanalysis,” in *Fast Software Encryption*, 2009, pp. 193–208.

[74] C. De Cannière, “Interpolation Attack,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, p. 620. doi: 10.1007/978-1-4419-5906-5_584.

[75] T. Jakobsen and L. R. Knudsen, “The Interpolation Attack on Block Ciphers,” in *In Fast Software Encryption*, 1997, pp. 28–40.

[76] J. Kelsey, B. Schneier, and D. Wagner, “Mod n Cryptanalysis, with Applications against RC5P and M6,” in *Fast Software Encryption: 6th International Workshop, FSE’99 Rome, Italy, March 24--26, 1999 Proceedings*, L. Knudsen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 139–155. doi: 10.1007/3-540-48519-8_11.

[77] S. Moriai, M. Sugita, K. Aoki, and M. Kanda, “Security of E2 against Truncated Differential Cryptanalysis,” in *Selected Areas in Cryptography*, 2000, pp. 106–117.

[78] S. Lee, S. Hong, S. Lee, J. Lim, and S. Yoon, “Truncated Differential Cryptanalysis of Camellia,” in *Proceedings of the 4th International Conference Seoul on Information Security and Cryptology*, 2002, pp. 32–38.

[79] J. Chen, M. Wang, and B. Preneel, “Impossible Differential Cryptanalysis of the Lightweight Block Ciphers TEA, XTEA and HIGHT,” in *Progress in Cryptology - AFRICACRYPT 2012*, 2012, pp. 117–137.

[80] J. Kim, S. Hong, J. Sung, S. Lee, J. Lim, and S. Sung, “Impossible Differential Cryptanalysis for Block Cipher Structures,” in *Progress in Cryptology - INDOCRYPT 2003*, 2003, pp. 82–96.

[81] S. Murphy and M. J. B. Robshaw, “Remarks on security of AES and XSL technique,” *Electron. Lett.*, vol. 39, no. 1, pp. 36–38, Jan. 2003, doi: 10.1049/el:20030015.

[82] C. Cid and G. Leurent, “An Analysis of the XSL Algorithm,” in *Advances in Cryptology - ASIACRYPT 2005*, 2005, pp. 333–352.

[83] C.-W. Lim and K. Khoo, “An Analysis of XSL Applied to BES,” in *Fast Software Encryption*, 2007, pp. 242–253.

[84] C. Diem, “The {XL}-Algorithm and a Conjecture from Commutative Algebra,” *\ifnum\shortbib=1{ASIACRYPT}\else{Advances Cryptol. -- {ASIACRYPT}}\fi~2004*, vol. 3329, no. 1, pp. 323–337, 2004, doi: 10.1007/978-3-540-30539-2_23.

[85] National Bureau Of Standards, “Data Encryption Standard (DES),” 1999.

[86] P. Patil and P. Narayankar, “A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish,” *Procedia Comput. Sci.*, vol. 78, no. December 2015, pp. 617–624, 2016, doi: 10.1016/j.procs.2016.02.108.

[87] N. T. Courtois and G. V Bard, “Algebraic Cryptanalysis of the Data Encryption Standard,” in *Cryptography and Coding: 11th IMA International Conference, Cirencester, UK, December 18-20, 2007. Proceedings*, S. D. Galbraith, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 152–169. doi: 10.1007/978-3-540-77272-9_10.

[88] A. Biryukov and C. De Cannière, “Data Encryption Standard (DES),” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 295–301. doi: 10.1007/978-1-4419-5906-5_568.

[89] R. Bott, *Understanding Cryptography*, no. 1. Springer, 2014. doi: 10.1007/s13398-014-0173-7.2.

[90] H. O. Alanazi, B. B. Zaidan, a. a. Zaidan, H. a. Jalab, M. Shabbir, and Y. Al-Nabhani, “New Comparative Study Between DES, 3DES and AES within Nine Factors,” *J. Comput.*, vol. 2, no. 3, pp. 2151–9617, 2010.

[91] W. C. Barker and E. Barker, “Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher,” *NIST Spec. Publ.*, vol. 1, no. January, pp. 800–67, 2012, doi: 10.6028/NIST.SP.800-67r1.

[92] W. C. Barker and E. B. Barker, “SP 800-67 Rev. 1. Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher,” National Institute of Standards & Technology, Gaithersburg, MD, United States, 2012.

[93] C. De Cannière, “Blowfish,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 157–158. doi: 10.1007/978-1-4419-5906-5_550.

[94] S. P. Priyadharshini and P. G. Scholar, “Implementation of Security in Wireless Sensor Network using Blowfish Algorithm,” pp. 33–37, 2014.

[95] B. Algorithm, “IJESMR I nternational J ournal OF E ngineering S ciences & M anagement R esearch,” vol. 2, no. 10, pp. 45–52, 2015.

[96] O. Kara and C. Manap, “A New Class of Weak Keys for Blowfish,” in *Fast softeare Encryption, 14PthP International Workshop, FSE 2007*, 2007, pp. 167–180.

[97] C. De Cannière, “TWOFISH,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 1339–1340. doi: 10.1007/978-1-4419-5906-5_623.

[98] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, *The Twofish Encryption Algorithm: A 128-bit Block Cipher*. New York, NY, USA: John Wiley &amp; Sons, Inc., 1999.

[99] B. Schneier and D. Whiting, “Twofish on Smart Cards,” in *Smart Card Research and Applications: Third International Conference, CARDIS’98, Louvain-la-Neuve, Belgium, September 14-16, 1998. Proceedings*, J.-J. Quisquater and B. Schneier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 265–276. doi: 10.1007/10721064_25.

[100] L. R. K. V. Rijmen, “On the design and security of RC2,” in *Fast Software Encryption, Fift International Workshop, FSE’98*, 1998, pp. 206–221.

[101] M. Mathur and A. Kesarwani, “COMPARISON BETWEEN DES , 3DES , RC2 , RC6 , BLOWFISH AND AES,” *Proc. Natl. Conf. Horizons IT-NCNHIT*, pp. 143–148, 2013.

[102] R. Rivest, “A Description of the RC2(R) Encryption Algorithm.” RFC Editor, United States, 1998.

[103] B. S. Kaliski Jr. and Y. L. Yin, “On Differential and Linear Crytoanalysis of the RC5 Encryption Algorithm,” in *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, 1995, pp. 171–184.

[104] H. S. Gill, “Selection of Parameter ‘ r ’ in RC5 Algorithm on the basis of Prime Number,” pp. 6–8, 2014.

[105] J. Singh, B. Kumar, and A. Khatri, “Improving stored data security in Cloud using Rc5 algorithm,” *3rd Nirma Univ. Int. Conf. Eng. NUiCONE 2012*, 2012, doi: 10.1109/NUICONE.2012.6493220.

[106] A. R. Bevi, S. S. V Sheshu, and S. Malarvizhi, “FPGA Based Sliding Window Architecture for RC5 Encryption,” in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, 2012, pp. 614–618. doi: 10.1145/2345396.2345496.

[107] A. Biryukov and E. Kushilevitz, “Improved cryptanalysis of RC5,” in *Advances in Cryptology --- EUROCRYPT’98: International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, May 31 -- June 4, 1998 Proceedings*, K. Nyberg, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 85–99. doi: 10.1007/BFb0054119.

[108] G.-H. Kim, J.-N. Kim, and G.-Y. Cho, “An Improved RC6 Algorithm with the Same Structure of Encryption and Decryption,” in *Proceedings of the 11th International Conference on Advanced Communication Technology - Volume 2*, 2009, pp. 1211–1215.

[109] K. Wu and R. Karri, “Idle Cycles Based Concurrent Error Detection of RC6 Encryption,” in *Proceedings of the 16th IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems*, 2001, pp. 200–205.

[110] N. Chandel, S. Mishra, N. Gupta, and a Sinhal, “Creation of secure cloud environment using RC6,” *2013 Int. Conf. Intell. Syst. Signal Process. ISSP 2013*, pp. 317–318, 2013, doi: 10.1109/ISSP.2013.6526926.

[111] K. Aggarwal and A. K. Expansion, “Comparison of RC6 , Modified RC6 & Enhancement of RC6,” pp. 444–449, 2015.

[112] N. Varshney and K. Raghuwanshi, “RC6 Based Data Security and Attack Detection,” in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1*, S. C. Satapathy and S. Das, Eds. Cham: Springer International Publishing, 2017, pp. 3–10. doi: 10.1007/978-3-319-30933-0_1.

[113] J. Borst, B. Preneel, and J. Vandewalle, “Linear Cryptanalysis of RC5 and RC6,” in *Fast Software Encryption: 6th International Workshop, FSE’99 Rome, Italy, March 24--26, 1999 Proceedings*, L. Knudsen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 16–30. doi: 10.1007/3-540-48519-8_2.

[114] J.-L. Beuchat, “FPGA Implementations of the RC6 Block Cipher,” in *Field Programmable Logic and Application: 13th International Conference, FPL 2003, Lisbon, Portugal, September 1-3, 2003 Proceedings*, P. Y. K. Cheung and G. A. Constantinides, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 101–110. doi: 10.1007/978-3-540-45234-8_11.

[115] H. Handschuh, “RC6,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 1033–1034. doi: 10.1007/978-1-4419-5906-5_608.

[116] Y. Wang and N. Wang, “Research on Time-Varying Camellia Encryption Algorithm,” in *Software Engineering and Knowledge Engineering: Theory and Practice: Selected papers from 2012 International Conference on Software Engineering, Knowledge Engineering and Information Engineering (SEKEIE 2012)*, W. Zhang, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 823–828. doi: 10.1007/978-3-642-29455-6_110.

[117] C. De Cannière, “Camellia,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg, Ed. Boston, MA: Springer US, 2005, pp. 61–62. doi: 10.1007/0-387-23483-7_44.

[118] Y. Lu, M. P. O’Neill, and J. V McCanny, “Differential Power Analysis resistance of Camellia and countermeasure strategy on FPGAs,” in *2009 International Conference on Field-Programmable Technology*, Dec. 2009, pp. 183–189. doi: 10.1109/FPT.2009.5377650.

[119] Y. He and S. Qing, “Square Attack on Reduced Camellia Cipher,” in *Information and Communications Security: Third International Conference, ICICS 2001 Xian, China, November 13--16, 2001 Proceedings*, S. Qing, T. Okamoto, and J. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 238–245. doi: 10.1007/3-540-45600-7_27.

[120] J. Lu, Y. Wei, J. Kim, and E. Pasalic, "The higher-order meet-in-the-middle attack and its application to the Camellia block cipher," *Theor. Comput. Sci.*, vol. 527, pp. 102–122, 2014, doi: http://dx.doi.org/10.1016/j.tcs.2014.01.031.

[121] Z. Cica, "Pipelined implementation of Camellia encryption algorithm," in *2016 24th Telecommunications Forum (TELFOR)*, Nov. 2016, pp. 1–4. doi: 10.1109/TELFOR.2016.7818785.

[122] M. S, A. Kato, and M. Kanda, "Addition of Camellia Cipher Suites to Transport Layer Security (TLS)," 2005.

[123] A. Carlisle, "Constructing of Symmetric ciphers using the CAST design Procedure," *Des. Codes, Cryptogr.*, vol. 12, pp. 283–316, 1997, doi: 10.1023/A:1008229029587.

[124] K. G.N, R. V, L. G.H, and A. M.E, "Performance Enhancement of CAST-128 Algorithm by modifying its function," in *Advances in Computer and Information Sciences and Engineering*, T. Sobh, Ed. Dordrecht: Springer Netherlands, 2008, pp. 256–260. doi: 10.1007/978-1-4020-8741-7_46.

[125] G. N. Krishnamurthy, V. Ramaswamy, G. H. Leela, and M. E. Ashalatha, "Performance enhancement of Blowfish and CAST-128 algorithms and Security analysis of improved Blowfish algorithm using Avalanche effect," vol. 8, no. 3, pp. 244–250, 2008.

[126] P. E. Nastou and Y. C. Stamatiou, "Dynamically modifiable ciphers using a reconfigurable CAST-128 based algorithm on AMTEL's FPSLIC rec," in *Proceedings 16th International Parallel and Distributed Processing Symposium*, Apr. 2002, pp. 7 pp-. doi: 10.1109/IPDPS.2002.1016556.

[127] A. Lysyak, "Analysis of gradient statistical attack at block ciphers RC6, MARS, CAST-128," in *2012 XIII International Symposium on Problems of Redundancy in Information and Control Systems*, Sep. 2012, pp. 44–47. doi: 10.1109/RED.2012.6338405.

[128] J.-Y. Zhao, M.-Q. Wang, and L. Wen, "Improved Linear Cryptanalysis of CAST-256," *J. Comput. Sci. Technol.*, vol. 29, no. 6, pp. 1134–1139, 2014, doi: 10.1007/s11390-014-1496-8.

[129] C. Adams, H. M. Heys, S. E. Tavares, and M. Wiener, "An analysis of the CAST-256 cipher," in *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.99TH8411)*, May 1999, vol. 1, pp. 361–366 vol.1. doi: 10.1109/CCECE.1999.807225.

[130] S. Wang, T. Cui, and M. Wang, "Improved Differential Cryptanalysis of CAST-128 and CAST-256," in *Information Security and Cryptology: 12th International Conference, Inscrypt 2016, Beijing, China, November 4-6, 2016, Revised Selected Papers*, K. Chen, D. Lin, and M. Yung, Eds. Cham: Springer International Publishing, 2017, pp. 18–32. doi: 10.1007/978-3-319-54705-3_2.

[131] A. Pestunov, "Differential cryptanalysis of 24-round CAST-256," in *2008 IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering*, Jul. 2008, pp. 46–49. doi: 10.1109/SIBIRCON.2008.4602582.

[132] M. Riaz and H. M. Heys, "The FPGA implementation of the RC6 and CAST-256 encryption algorithms," in *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.99TH8411)*, May 1999, vol. 1, pp. 367–372 vol.1. doi: 10.1109/CCECE.1999.807226.

[133] J. Sung, "Differential cryptanalysis of eight-round SEED," *Inf. Process. Lett.*, vol. 111, no. 10, pp. 474–478, 2011, doi: http://dx.doi.org/10.1016/j.ipl.2011.02.004.

[134] J. Yi, K. Park, J. Park, and W. W. Ro, "Fully Pipelined Hardware Implementation of 128-Bit SEED Block Cipher Algorithm," in *Reconfigurable Computing: Architectures, Tools and Applications: 5th International Workshop, ARC 2009, Karlsruhe, Germany, March 16-18, 2009. Proceedings*, J. Becker, R. Woods, P. Athanas, and F. Morgan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 181–192. doi: 10.1007/978-3-642-00641-8_19.

[135] B. Rogers, S. Chhabra, M. Prvulovic, and Y. Solihin, "Using Address Independent Seed Encryption and Bonsai Merkle Trees to Make Secure Processors OS- and Performance-Friendly," in *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, Dec. 2007, pp. 183–196. doi: 10.1109/MICRO.2007.16.

[136] H. Ko and C. Ramos, "A Study on the Encryption Algorithm for RFID Tag (SEED : 8 Rounds #x0D7; 64 bits block)," in *2008 International Conference on Convergence and Hybrid Information Technology*, Aug. 2008, pp. 672–677. doi: 10.1109/ICHIT.2008.194.

[137] A. Biryukov, "Skipjack," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston,

MA: Springer US, 2011, pp. 1220–1221. doi: 10.1007/978-1-4419-5906-5_616.

[138] R. C.-W. Phan, “Cryptanalysis of full Skipjack block cipher,” *Electron. Lett.*, vol. 38, no. 2, pp. 69–71, Jan. 2002, doi: 10.1049/el:20020051.

[139] E. Eryumaz, I. Erturk, and S. Atmaca, “Implementation of Skipjack cryptology algorithm for WSNs using FPGA,” in *2009 International Conference on Application of Information and Communication Technologies*, Oct. 2009, pp. 1–5. doi: 10.1109/ICAICT.2009.5372531.

[140] J. H. Kong, L. M. Ang, K. P. Seng, and F. T. Ong, “Low-complexity Two Instruction Set Computer architecture for sensor network using Skipjack encryption,” in *The International Conference on Information Networking 2011 (ICOIN2011)*, Jan. 2011, pp. 472–477. doi: 10.1109/ICOIN.2011.5723161.

[141] M. Matsui and T. Tokita, “Cryptanalysis of a reduced version of the block cipher E2,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1636, pp. 71–80, 1999, doi: 10.1007/3-540-48519-8_6.

[142] M. Kwan, “The Design of the ICE Encryption Algorithm,” in *Proceedings of the 4th International Workshop on Fast Software Encryption*, 1997, pp. 69–82.

[143] B. Van Rompay, L. R. Knudsen, and V. Rijmen, “Differential Cryptanalysis of the ICE Encryption Algorithm,” in *Fast Software Encryption: 5th International Workshop, FSE’ 98 Paris, France, March 23--25, 1998 Proceedings*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 270–283. doi: 10.1007/3-540-69710-1_18.

[144] A. P. Fournaris, N. Sklavos, and O. Koufopavlou, “VLSI architecture and FPGA implementation of ICE encryption algorithm,” in *10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003*, Dec. 2003, vol. 1, pp. 88-91 Vol.1. doi: 10.1109/ICECS.2003.1301983.

[145] T. Tokita and T. Matsumoto, “On applicability of differential cryptanalysis, linear cryptanalysis and mod {n} cryptanalysis to an encryption algorithm {M}8 ({ISO}9979-20),” *Ipsj J.*, vol. 42, no. 8, pp. 2098–2105, 2001.

[146] P. Kitsos, M. D. Galanis, and O. Koufopavlou, “High-speed hardware implementations of the KASUMI block cipher,” *2004 IEEE Int. Symp. Circuits Syst. (IEEE Cat. No.04CH37512)*, pp. II-549–52, 2004, doi: 10.1109/ISCAS.2004.1329330.

[147] E. Biham, O. Dunkelman, and N. Keller, “A Related-key Rectangle Attack on the Full KASUMI,” in *Proceedings of the 11th International Conference on Theory and Application of Cryptology and Information Security*, 2005, pp. 443–461. doi: 10.1007/11593447_24.

[148] M. Blunden and A. Escott, “Related key attacks on reduced round kasumi,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2355, pp. 277–285, 2002, doi: 10.1007/3-540-45473-X_23.

[149] R. Fang, Y. Ying-jian, and F. Xiao-bing, “A Small and Efficienct Hardware Implementation of the KASUMI,” in *2009 WASE International Conference on Information Engineering*, Jul. 2009, vol. 2, pp. 377–380. doi: 10.1109/ICIE.2009.187.

[150] O. Dunkelman, N. Keller, and A. Shamir, “A Practical-Time Attack on the A5 / 3 Cryptosystem Used in Third Generation GSM Telephony,” *Int. Assoc. Cryptologic Res.*, vol. 8, no. December 2009, pp. 393–410, 2010, doi: 10.1007/978-3-642-14623-7_21.

[151] I. Sima, D. Ţărmurean, V. Greu, and A. V Diaconu, “XXTEA, an alternative replacement of KASUMI cipher algorithm in A5/3 GSM and f8, f9 UMTS data security functions,” in *2012 9th International Conference on Communications (COMM)*, Jun. 2012, pp. 323–326. doi: 10.1109/ICComm.2012.6262617.

[152] T. Balderas-Contreras and R. A. Cumplido-Parra, “An efficient reuse-based approach to implement the 3GPP KASUMI block cipher,” in *(ICEEE). 1st International Conference on Electrical and Electronics Engineering, 2004.*, Jun. 2004, pp. 113–118. doi: 10.1109/ICEEE.2004.1433860.

[153] K. Jia, C. Rechberger, and X. Wang, “Green Cryptanalysis : Meet-in-the-Middle Key-Recovery for the Full KASUMI Cipher,” pp. 1–18.

[154] K. Jia, L. Li, C. Rechberger, J. Chen, and X. Wang, “Improved Cryptanalysis of the Block Cipher KASUMI,” in *Selected Areas in Cryptography: 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, L. R. Knudsen and H. Wu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 222–233. doi: 10.1007/978-3-642-35999-6_15.

[155] W. Yi and S. Chen, “Multidimensional zero-correlation linear cryptanalysis of the block

cipher KASUMI," *IET Inf. Secur.*, vol. 10, no. 4, pp. 215–221, 2016, doi: 10.1049/iet-ifs.2014.0543.

[156] H. Beker and F. Piper, *Cipher systems: the protection of communications*. Northwood Books, 1982.

[157] D. C. Hankerson *et al.*, *Coding Theory and Cryptography: The Essentials, Second Edition*. Taylor & Francis, 2000.

[158] J. Stern and S. Vaudenay, "CS-Cipher," in *Fast Software Encryption: 5th International Workshop, FSE' 98 Paris, France, March 23--25, 1998 Proceedings*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 189–204. doi: 10.1007/3-540-69710-1_13.

[159] L. Granboulan, G. Martinet, M. Dichtl, P. Serf, and M. Schafheutle, "NESSIE Phase I--Selection of primitives (2001).pdf," 2001.

[160] T. Roche, R. Gillard, and J. L. Roch, "Provable Security against Impossible Differential Cryptanalysis Application to CS-Cipher," *Commun. Comput. Inf. Sci.*, vol. 14, pp. 597–606, 2008, doi: 10.1007/978-3-540-87477-5_63.

[161] S. Vaudenay, "On the Security of CS-Cipher," in *Fast Software Encryption: 6th International Workshop, FSE'99 Rome, Italy, March 24--26, 1999 Proceedings*, L. Knudsen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 260–274. doi: 10.1007/3-540-48519-8_19.

[162] P. Crowley, "Mercy: A Fast Large Block Cipher for Disk Sector Encryption," in *Fast Software Encryption: 7th International Workshop, FSE 2000 New York, NY, USA, April 10--12, 2000 Proceedings*, G. Goos, J. Hartmanis, J. van Leeuwen, and B. Schneier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 49–63. doi: 10.1007/3-540-44706-7_4.

[163] S. R. Fluhrer, "Cryptanalysis of the Mercy Block Cipher," in *Fast Software Encryption: 8th International Workshop, FSE 2001 Yokohama, Japan, April 2--4, 2001 Revised Papers*, M. Matsui, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 28–36. doi: 10.1007/3-540-45473-X_3.

[164] J. Nechvatal *et al.*, "Report on the Development of the Advanced Encryption Standard (AES)," 2000. doi: 10.1.1.106.2169.

[165] C. Burwick *et al.*, "MARS - a candidate cipher for AES," 1999.

[166] W. Xue and X. Lai, "Impossible differential cryptanalysis of MARS-like structures," *IET Inf. Secur.*, vol. 9, no. 4, pp. 219–222, 2015, doi: 10.1049/iet-ifs.2014.0183.

[167] M. Gorski, T. Knapke, E. List, S. Lucks, and J. Wenzel, "Mars Attacks! Revisited," in *Progress in Cryptology -- INDOCRYPT 2011: 12th International Conference on Cryptology in India, Chennai, India, December 11-14, 2011. Proceedings*, D. J. Bernstein and S. Chatterjee, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 94–113. doi: 10.1007/978-3-642-25578-6_9.

[168] M. Gorski, T. Knapke, E. List, S. Lucks, and J. Wenzel, "Mars Attacks! Revisited: Differential Attack on 12 Rounds of the MARS Core and Defeating the Complex MARS Key-schedule," in *Proceedings of the 12th International Conference on Cryptology in India*, 2011, pp. 94–113. doi: 10.1007/978-3-642-25578-6_9.

[169] L. R. Knudsen and V. Rijmen, "On the decorrelated fast cipher (DFC) and its theory," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1636, pp. 81–94, 1999, doi: 10.1007/3-540-48519-8_7.

[170] W. Wu, B. Li, D. Feng, and S. Qing, "On Decorrelated Fast Cipher," *J. Electron.*, vol. 17, no. 1, pp. 94–96, 2000, doi: 10.1007/s11767-000-0028-6.

[171] G. Poupard and S. Vaudenay, "Decorrelated Fast Cipher: An AES Candidate Well Suited for Low Cost Smart Cards Applications," in *Third Smart Card Research and Advanced Applications Conference Proceedings*, p.

[172] L. Knudsen, "DEAL - A 128-bit Block Cipher," 1998.

[173] S. Lucks, "On the Security of the 128-bit Block Cipher DEAL," in *Fast Software Encryption: 6th International Workshop, FSE'99 Rome, Italy, March 24--26, 1999 Proceedings*, L. Knudsen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 60–70. doi: 10.1007/3-540-48519-8_5.

[174] J. Kelsey and B. Schneier, "Key-Schedule Cryptanalysis of DEAL," in *Selected Areas in Cryptography: 6th Annual International Workshop, SAC'99 Kingston, Ontario, Canada, August 9--10, 1999 Proceedings*, H. Heys and C. Adams, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 118–134. doi: 10.1007/3-540-46513-8_9.

[175] E. Biham, "A note on comparing the AES candidates," 1999.

[176] J. Dray and C. S. Division, “Report on the NIST Java^{TM} AES Candidate Algorithm Analysis,” 1999.

[177] H. Handschuh and S. Vaudenay, “A Universal Encryption Standard,” in *Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography*, 2000, pp. 1–12.

[178] J. Soto and L. Bassham, “Randomness Testing of the Advanced Encryption Standard Finalist Candidates Randomness Testing of the Advanced Encryption Standard Finalist Candidates 1,” 2000.

[179] N. Fips, “Announcing the ADVANCED ENCRYPTION STANDARD (AES),” *Byte*, vol. 2009, no. 12, pp. 8–12, 2001, doi: 10.1016/S1353-4858(10)70006-4.

[180] S. Murphy and M. Robshaw, “New Observations on Rijndael,” *Inf. Secur.*, pp. 1–17, 2000.

[181] B. M. P. and K. R. R. Babu, “Secure cloud storage using AES encryption,” in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Sep. 2016, pp. 859–864. doi: 10.1109/ICACDOT.2016.7877709.

[182] X. Yang and W. Wen, “Design of a pre-scheduled data bus for advanced encryption standard encrypted system-on-chips,” in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2017, pp. 506–511. doi: 10.1109/ASPDAC.2017.7858373.

[183] V. Miškovský, H. Kubátová, and M. Novotný, “Influence of fault-tolerant design methods on differential power analysis resistance of AES cipher: Methodics and challenges,” in *2016 5th Mediterranean Conference on Embedded Computing (MECO)*, Jun. 2016, pp. 14–17. doi: 10.1109/MECO.2016.7525685.

[184] M. A. Wright, “The Advanced Encryption Standard,” *Netw. Secur.*, vol. 2001, no. 10, pp. 11–13, 2001, doi: http://dx.doi.org/10.1016/S1353-4858(01)01018-2.

[185] W. Stallings, “The Advanced Encryption Standard,” *Cryptologia*, vol. 26, no. 3, pp. 165–188, Jul. 2002, doi: 10.1080/0161-110291890876.

[186] S. M. Wadi and N. Zainal, “Rapid Encryption Method based on AES Algorithm for Grey Scale HD Image Encryption,” *Procedia Technol.*, vol. 11, pp. 51–56, 2013, doi: http://dx.doi.org/10.1016/j.protcy.2013.12.161.

[187] J. Daemen and V. Rijmen, *AES Proposal: Rijndael*. 2003.

[188] C. Paar and J. Pelzl, “The Advanced Encryption Standard (AES),” in *Understanding Cryptography: A Textbook for Students and Practitioners*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 87–121. doi: 10.1007/978-3-642-04101-3_4.

[189] S. Heron, “Advanced Encryption Standard (AES),” *Netw. Secur.*, vol. 2009, no. 12, pp. 8–12, 2009, doi: http://dx.doi.org/10.1016/S1353-4858(10)70006-4.

[190] H. Trang, “An efficient FPGA implementation of the Advanced Encryption Standard algorithm,” pp. 5–8, 2012.

[191] J. Daemen and V. Rijmen, *The Design of Rijndael*. 2002. doi: 10.1007/978-3-662-04722-4.

[192] S. Murphy, “The Advanced Encryption Standard (AES),” *Inf. Secur. Tech. Rep.*, vol. 4, no. 4, pp. 12–17, 1999, doi: http://dx.doi.org/10.1016/S1363-4127(99)80083-1.

[193] E. P. Nugroho, R. R. J. Putra, and I. M. Ramadhan, “SMS authentication code generated by Advance Encryption Standard (AES) 256 bits modification algorithm and One time Password (OTP) to activate new applicant account,” in *2016 2nd International Conference on Science in Information Technology (ICSITech)*, Oct. 2016, pp. 175–180. doi: 10.1109/ICSITech.2016.7852629.

[194] J. Hartmanis and J. Van Leeuwen, *Lecture Notes in Computer Science*.

[195] R. Anderson, E. Biham, and L. Knudsen, “Serpent : A proposal for the advanced encryption standard,” *NIST AES Propos.*, pp. 1–23, 1998.

[196] R. Anderson, E. Biham, and L. R. Knudsen, “Serpent and smartcards,” *Smart Card Res. …*, pp. 246–253, 2000.

[197] C. De Canni`ere, “Serpent,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg, Ed. Boston, MA: Springer US, 2005, pp. 563–564. doi: 10.1007/0-387-23483-7_386.

[198] W. Wu, D. Feng, and S. Qing, “Power analysis of RC6 and SERPENT,” in *Information Security for Global Information Infrastructures: IFIP TC11 Sixteenth Annual Working Conference on Information Security August 22--24, 2000, Beijing, China*, S. Qing and J. H. P. Eloff, Eds. Boston, MA: Springer US, 2000, pp. 201–209. doi: 10.1007/978-0-387-35515-3_21.

[199] E. Biham, O. Dunkelman, and N. Keller, “Differential-Linear Cryptanalysis of Serpent,” in *Fast Software Encryption: 10th International*

Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003. Revised Papers, T. Johansson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 9–21. doi: 10.1007/978-3-540-39887-5_2.

[200] A. K. Lutz *et al.*, "2Gbit/s Hardware Realizations of RIJNDAEL and SERPENT: A Comparative Analysis," in *Cryptographic Hardware and Embedded Systems - CHES 2002: 4th International Workshop Redwood Shores, CA, USA, August 13--15, 2002 Revised Papers*, B. S. Kaliski, çetin K. Koç, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 144–158. doi: 10.1007/3-540-36400-5_12.

[201] A. J. Elbirt and C. Paar, "An FPGA Implementation and Performance Evaluation of the Serpent Block Cipher," in *Proceedings of the 2000 ACM/SIGDA Eighth International Symposium on Field Programmable Gate Arrays*, 2000, pp. 33–40. doi: 10.1145/329166.329176.

[202] M. Alioto, S. Bongiovanni, G. Scotti, and A. Trifiletti, "Leakage Power Analysis attacks against a bit slice implementation of the Serpent block cipher," in *2014 Proceedings of the 21st International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)*, Jun. 2014, pp. 241–246. doi: 10.1109/MIXDES.2014.6872193.

[203] M. A. Amiri, M. Mahdavi, R. E. Atani, and S. Mirzakuchaki, "QCA Implementation of Serpent Block Cipher," in *2009 Second International Conference on Advances in Circuits, Electronics and Micro-electronics*, Oct. 2009, pp. 16–19. doi: 10.1109/CENICS.2009.18.

[204] D. Kwon *et al.*, "New Block Cipher: ARIA," in *Information Security and Cryptology - ICISC 2003: 6th International Conference, Seoul, Korea, November 27-28, 2003. Revised Papers*, J.-I. Lim and D.-H. Lee, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 432–445. doi: 10.1007/978-3-540-24691-6_32.

[205] Y. Yeom, Y. Cho, and M. Yung, "High-Speed Implementations of Block Cipher ARIA Using Graphics Processing Units," in *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, Apr. 2008, pp. 271–275. doi: 10.1109/MUE.2008.94.

[206] J. Park, Y.-D. Kim, S. Yang, and Y. You, "Low power compact design of ARIA block cipher," in *2006 IEEE International Symposium on Circuits and Systems*, May 2006, pp. 4 pp. – 316. doi: 10.1109/ISCAS.2006.1692585.

[207] L. Xiao, Y. Li, L. Ruan, G. Yao, and D. Li, "High Performance Implementation of ARIA Encryption Algorithm on Graphics Processing Units," in *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, Nov. 2013, pp. 504–510. doi: 10.1109/HPCC.and.EUC.2013.78.

[208] B. Koo, G. Ryu, T. Chang, and S. Lee, "Design and implementation of unified hardware for 128-bit block ciphers ARIA and AES," *ETRI J.*, vol. 29, no. 6, pp. 820–822, 2007, doi: 10.4218/etrij.07.0207.0077.

[209] C. H. Kim, "Differential fault analysis of {ARIA} in multi-byte fault models," *J. Syst. Softw.*, vol. 85, no. 9, pp. 2096–2103, 2012, doi: http://dx.doi.org/10.1016/j.jss.2012.04.009.

[210] J. Ha, C. Kim, S. Moon, I. Park, and H. Yoo, "Differential Power Analysis on Block Cipher ARIA," in *High Performance Computing and Communications: First International Conference, HPCC 2005, Sorrento, Italy, September 21-23, 2005. Proceedings*, L. T. Yang, O. F. Rana, B. Di Martino, and J. Dongarra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 541–548. doi: 10.1007/11557654_63.

[211] Akshima, D. Chang, M. Ghosh, A. Goel, and S. K. Sanadhya, "Improved Meet-in-the-Middle Attacks on 7 and 8-Round ARIA-192 and ARIA-256," in *Proceedings of the 16th International Conference on Progress in Cryptology -- INDOCRYPT 2015 - Volume 9462*, 2015, pp. 198–217. doi: 10.1007/978-3-319-26617-6_11.

[212] X. Tang, B. Sun, R. Li, C. Li, and J. Yin, "A Meet-in-the-middle Attack on Reduced-round ARIA," *J. Syst. Softw.*, vol. 84, no. 10, pp. 1685–1692, Oct. 2011, doi: 10.1016/j.jss.2011.04.053.

[213] Wenling Wu Wentao Zhang and D. Feng, "Impossible Differential Cryptanalysis of ARIA and Camellia." 2006.

[214] J. Kelsey, B. Schneier, and D. Wagner, "Related-key cryptanalysis of 3-WAY, Biham-DES,CAST, DES-X, NewDES, RC2, and TEA," in *Information and Communications Security: First International Conference, ICIS '97 Beijing, China, November 11--14, 1997 Proceedings*, Y. Han, T. Okamoto, and S. Qing, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 233–246. doi: 10.1007/BFb0028479.

[215] J. Daemen and J. V, "A New Approach Towards Block Cipher Design," in *Fast Software Encryption*, pp. 18–33.

[216] C. H. Lim, "CRYPTON: A New 128-bit Block Cipher - Specification and Analysis." 1998.

[217] J. H. Cheon, M. Kim, K. Kim, L. Jung-Yeun, and S. Kang, “Improved Impossible Differential Cryptanalysis of Rijndael and Crypton,” in *Information Security and Cryptology --- ICISC 2001: 4th International Conference Seoul, Korea, December 6--7,2001 Proceedings*, K. Kim, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 39–49. doi: 10.1007/3-540-45861-1_4.

[218] C. H. Lim, “A Revised Version of CRYPTON: CRYPTON V1.0,” in *Fast Software Encryption: 6th International Workshop, FSE’99 Rome, Italy, March 24--26, 1999 Proceedings*, L. Knudsen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 31–45. doi: 10.1007/3-540-48519-8_3.

[219] M. Minier and H. Gilbert, “Stochastic Cryptanalysis of Crypton,” in *Fast Software Encryption: 7th International Workshop, FSE 2000 New York, NY, USA, April 10--12, 2000 Proceedings*, G. Goos, J. Hartmanis, J. van Leeuwen, and B. Schneier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 121–133. doi: 10.1007/3-540-44706-7_9.

[220] H. Wei and B. Wang, “Integral Cryptanalysis of Reduced-Round Crypton Block Cipher,” in *2009 International Symposium on Computer Network and Multimedia Technology*, Jan. 2009, pp. 1–4. doi: 10.1109/CNMT.2009.5374509.

[221] Y. Wei, C. Li, and B. Sun, “Related-key impossible differential cryptanalysis on Crypton and Crypton v1.0,” in *2011 World Congress on Internet Security (WorldCIS-2011)*, Feb. 2011, pp. 227–232.

[222] M. Shakiba, M. Dakhilalian, and H. Mala, “Non-isomorphic biclique cryptanalysis of full-round Crypton,” *Comput. Stand. Interfaces*, vol. 41, pp. 72–78, 2015, doi: http://dx.doi.org/10.1016/j.csi.2015.02.002.

[223] N. Consortium, “New European Schemes for Signatures, Integrity, and Encryption,” 2004.

[224] A. Biryukov, “Analysis of Involutional Ciphers: Khazad and Anubis,” in *Fast Software Encryption: 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003. Revised Papers*, T. Johansson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 45–53. doi: 10.1007/978-3-540-39887-5_5.

[225] B. B. Brumley, “Secure and Fast Implementations of Two Involution Ciphers,” in *Information Security Technology for Applications: 15th Nordic Conference on Secure IT Systems, NordSec 2010, Espoo, Finland, October 27-29, 2010, Revised Selected Papers*, T. Aura, K. Järvinen, and K. Nyberg, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 269–282. doi: 10.1007/978-3-642-27937-9_19.

[226] L. Keliher, H. Meijer, and S. Tavares, “NESSIE security report,” 2001.

[227] E. Biham, V. Furman, M. Misztal, and V. Rijmen, “Differential Cryptanalysis of Q,” in *Fast Software Encryption: 8th International Workshop, FSE 2001 Yokohama, Japan, April 2--4, 2001 Revised Papers*, M. Matsui, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 174–186. doi: 10.1007/3-540-45473-X_15.

[228] V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, and E. De Win, “The cipher SHARK,” in *Fast Software Encryption: Third International Workshop Cambridge, UK, February 21--23 1996 Proceedings*, D. Gollmann, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 99–111. doi: 10.1007/3-540-60865-6_47.

[229] K. Ohkuma, H. Muratani, F. Sano, and S. Kawamura, “The Block Cipher Hierocrypt,” in *Selected Areas in Cryptography: 7th Annual International Workshop, SAC 2000 Waterloo, Ontario, Canada, August 14--15, 2000 Proceedings*, D. R. Stinson and S. Tavares, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 72–88. doi: 10.1007/3-540-44983-3_6.

[230] A. G. Chefranov and A. Y. Mahmoud, “Elgamal Public Key Cryptosystem and Signature Scheme in GU(M,P,N),” in *Proceedings of the 3rd International Conference on Security of Information and Networks*, 2010, pp. 164–167. doi: 10.1145/1854099.1854134.

[231] Y. Desmedt, “ElGamal Public Key Encryption,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, p. 396. doi: 10.1007/978-1-4419-5906-5_318.

[232] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985, doi: 10.1109/TIT.1985.1057074.

[233] C.-C. Lee, M.-S. Hwang, and S.-F. Tzeng, “a New Convertible Authenticated Encryption Scheme Based on the Elgamal Cryptosystem,” *Int. J. Found. Comput. Sci.*, vol. 20, no. 02, p. 351, 2009, doi: 10.1142/S0129054109006607.

[234] K. Gjøsteen, “A New Security Proof for Damg{å}rd’s ElGamal,” in *Topics in Cryptology -- CT-RSA 2006: The Cryptographers’ Track at the RSA Conference 2006, San Jose, CA, USA,*

February 13-17, 2005. Proceedings, D. Pointcheval, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 150–158. doi: 10.1007/11605805_10.

[235] N. M. S. Iswari, "Key generation algorithm design combination of RSA and ElGamal algorithm," in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Oct. 2016, pp. 1–5. doi: 10.1109/ICITEED.2016.7863255.

[236] Y. Tsiounis and M. Yung, "On the security of ElGamal based encryption," in *Public Key Cryptography: First International Workshop on Practice and Theory in Public Key Cryptography, PKC'98 Pacifico Yokohama, Japan, February 5--6, 1998 Proceedings*, H. Imai and Y. Zheng, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 117–134. doi: 10.1007/BFb0054019.

[237] M.-S. Hwang, C.-C. Chang, and K.-F. Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 2, pp. 445–446, Mar. 2002, doi: 10.1109/69.991728.

[238] P. Balasubramaniam and P. Muthukumar, "Synchronization of chaotic systems using feedback controller: An application to Diffie–Hellman key exchange protocol and ElGamal public key cryptosystem," *J. Egypt. Math. Soc.*, vol. 22, no. 3, pp. 365–372, 2014, doi: http://dx.doi.org/10.1016/j.joems.2013.10.003.

[239] Y.-L. Qi, "An Improved Traitors Tracing Scheme Based on ELGamal," *Procedia Environ. Sci.*, vol. 10, pp. 392–395, 2011, doi: http://dx.doi.org/10.1016/j.proenv.2011.09.064.

[240] "Elliptic Curve Cryptography," in *Cryptographic Algorithms on Reconfigurable Hardware*, Boston, MA: Springer US, 2007, pp. 291–328. doi: 10.1007/978-0-387-36682-1_10.

[241] B. Anggorojati, N. R. Prasad, and R. Prasad, "Elliptic curve cryptography based key management for the M2M local cloud platform," in *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2016, pp. 73–78. doi: 10.1109/ICACSIS.2016.7872754.

[242] M. M. Chauhan, "An implemented of hybrid cryptography using elliptic curve cryptosystem (ECC) and MD5," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, Aug. 2016, vol. 3, pp. 1–6. doi: 10.1109/INVENTIVE.2016.7830092.

[243] G. Seroussi, "Elliptic curve cryptography," in *1999 Information Theory and Networking Workshop (Cat. No.99EX371)*, 1999, pp. 41-. doi: 10.1109/ITNW.1999.814351.

[244] M. Rosing, *Implementing Elliptic Curve Cryptography*. Greenwich, CT, USA: Manning Publications Co., 1999.

[245] N. Gura *et al.*, "An End-to-End Systems Approach to Elliptic Curve Cryptography," in *Cryptographic Hardware and Embedded Systems - CHES 2002: 4th International Workshop Redwood Shores, CA, USA, August 13--15, 2002 Revised Papers*, B. S. Kaliski, çetin K. Koç, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 349–365. doi: 10.1007/3-540-36400-5_26.

[246] D. He and S. Zeadally, "An Analysis of RFID Authentication Schemes for Internet of Things in Healthcare Environment Using Elliptic Curve Cryptography," *IEEE Internet Things J.*, vol. 2, no. 1, pp. 72–83, Feb. 2015, doi: 10.1109/JIOT.2014.2360121.

[247] Y. Salami, V. Khajehvand, and E. Zeinali, "SAIFC: A Secure Authentication Scheme for IOV Based on Fog-Cloud Federation," *Secur. Commun. Networks*, vol. 2023, 2023.

[248] S. R. Singh, A. K. Khan, and T. S. Singh, "A critical review on Elliptic Curve Cryptography," in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Sep. 2016, pp. 13–18. doi: 10.1109/ICACDOT.2016.7877543.

[249] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs," in *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, M. Joye and J.-J. Quisquater, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 119–132. doi: 10.1007/978-3-540-28632-5_9.

[250] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[251] D. Boneh, "Cramer--Shoup Public Key System," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg, Ed. Boston, MA: Springer US, 2005, pp. 108–109. doi: 10.1007/0-387-23483-7_84.

[252] S. Raju and Y. M. Sirajudeen, "Data security in Cloud Computing using Cramer - Shoup cryptosystem," in *2014 International*

Conference on Contemporary Computing and Informatics (IC3I), Nov. 2014, pp. 343–346. doi: 10.1109/IC3I.2014.7019773.

[253] X. Sun, B. Li, and X. Lu, “Cramer-Shoup Like Chosen Ciphertext Security from LPN,” in *Information Security Practice and Experience: 11th International Conference, ISPEC 2015, Beijing, China, May 5-8, 2015, Proceedings*, J. Lopez and Y. Wu, Eds. Cham: Springer International Publishing, 2015, pp. 79–95. doi: 10.1007/978-3-319-17533-1_6.

[254] J. Chang and R. Xue, “KDM-CCA security of the Cramer-Shoup cryptosystem, revisited,” in *2014 11th International Conference on Security and Cryptography (SECRYPT)*, Aug. 2014, pp. 1–8.

[255] H. Zhu, L. Chan, and X. Deng, “Variation of Cramer-Shoup public key scheme,” *Electron. Lett.*, vol. 35, no. 14, pp. 1150-, Jul. 1999, doi: 10.1049/el:19990701.

[256] S. Lucks, “A Variant of the Cramer-Shoup Cryptosystem for Groups of Unknown Order,” in *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, 2002, pp. 27–45.

[257] F. Publication, “DIGITAL SIGNATURE STANDARD,” 1998.

[258] P. Number *et al.*, “DIGITAL SIGNATURE STANDARD,” 2013.

[259] C. NIST, “The Digital Signature Standard,” *Commun. ACM*, vol. 35, no. 7, pp. 36–40, Jul. 1992, doi: 10.1145/129902.129904.

[260] D. Boneh, “Digital Signature Standard,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, p. 347. doi: 10.1007/978-1-4419-5906-5_145.

[261] J. Buchmann, E. Dahmen, and M. Szydlo, “Hash-based Digital Signature Schemes,” in *Post-Quantum Cryptography*, D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 35–93. doi: 10.1007/978-3-540-88702-7_3.

[262] E. Noroozi, S. M. Daud, A. Sabouhi, and H. Abas, “A New Dynamic Hash Algorithm in Digital Signature,” in *Advanced Machine Learning Technologies and Applications: First International Conference, AMLTA 2012, Cairo, Egypt, December 8-10, 2012. Proceedings*, A. E. Hassanien, A.-B. M. Salem, R. Ramadan, and T. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 583–589. doi: 10.1007/978-3-642-35326-0_58.

[263] B. A. Priya and A. Sheshasaayee, “Effective design of a parametrical security model for digital signatures using cryptography,” in *2016 International Conference on Communication and Electronics Systems (ICCES)*, Oct. 2016, pp. 1–3. doi: 10.1109/CESYS.2016.7889959.

[264] I. F. Blake and T. Garefalakis, “On the Security of the Digital Signature Algorithm,” *Des. Codes Cryptogr.*, vol. 26, no. 1–3, pp. 87–96, Jun. 2002, doi: 10.1023/A:1016549024113.

[265] S. Zu-hua, “Public-key cryptosystem and digital-signature schemes based on linear algebra over a local ring,” *IEE Proc. E - Comput. Digit. Tech.*, vol. 134, no. 5, pp. 254–256, Sep. 1987, doi: 10.1049/ip-e.1987.0043.

[266] S. G. Aki, “Digital signatures: A tutorial survey,” *Computer (Long. Beach. Calif).*, vol. 16, no. 2, pp. 15–24, Feb. 1983, doi: 10.1109/MC.1983.1654294.

[267] S. R. Subramanya and B. K. Yi, “Digital signatures,” *IEEE Potentials*, vol. 25, no. 2, pp. 5–8, Mar. 2006, doi: 10.1109/MP.2006.1649003.

[268] P. Joshi, M. Verma, and P. R. Verma, “Secure authentication approach using Diffie-Hellman key exchange algorithm for WSN,” in *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Dec. 2015, pp. 527–532. doi: 10.1109/ICCICCT.2015.7475336.

[269] M. N. Mejri, N. Achir, and M. Hamdi, “A new group Diffie-Hellman key generation proposal for secure VANET communications,” in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2016, pp. 992–995. doi: 10.1109/CCNC.2016.7444925.

[270] E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater, “Provably Authenticated Group Diffie-Hellman Key Exchange,” in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, 2001, pp. 255–264. doi: 10.1145/501983.502018.

[271] Y. Salami and V. Khajehvand, “SMAK-IOV: Secure Mutual Authentication Scheme and Key Exchange Protocol in Fog Based IoV,” *J. Comput. Robot.*, vol. 13, no. 1, pp. 11–20, 2020.

[272] H. Bob and H. Bob, “Math237 — Mathematics IIC Semester 1 , 2003 A worked example of RSA public key encryption,” pp. 1–3, 2003.

[273] T. Davis, “RSA Encryption,” *October*, pp. 1–6, 2003, doi: 10.1109/FPGA.2003.1227282.

[274] X. Zhou and X. Tang, “Research and implementation of RSA algorithm for encryption and decryption,” in *Proceedings of 2011 6th International Forum on Strategic Technology*, Aug. 2011, vol. 2, pp. 1118–1121. doi: 10.1109/IFOST.2011.6021216.

[275] H. A. Yajam, Y. K. Ahmadabadi, and M. Akhaee, “Deniable Encryption based on Standard RSA with OAEP,” in *2016 8th International Symposium on Telecommunications (IST)*, Sep. 2016, pp. 84–88. doi: 10.1109/ISTEL.2016.7881788.

[276] A. Fiat, “Batch RSA,” in *Advances in Cryptology --- CRYPTO’ 89 Proceedings*, G. Brassard, Ed. New York, NY: Springer New York, 1990, pp. 175–185. doi: 10.1007/0-387-34805-0_17.

[277] S. A. Kakvi, E. Kiltz, and A. May, “Certifying RSA,” in *Advances in Cryptology -- ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, X. Wang and K. Sako, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 404–414. doi: 10.1007/978-3-642-34961-4_25.

[278] M. Calderbank, “The RSA Cryptosystem : History , Algorithm , Primes,” 2007.

[279] D. Aggarwal and U. Maurer, “Breaking RSA Generically Is Equivalent to Factoring,” *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6251–6259, Nov. 2016, doi: 10.1109/TIT.2016.2594197.

[280] M. Klinkowski, M. Żotkiewicz, K. Walkowiak, M. Pióro, M. Ruiz, and L. Velasco, “Solving large instances of the RSA problem in flexgrid elastic optical networks,” *IEEE/OSA J. Opt. Commun. Netw.*, vol. 8, no. 5, pp. 320–330, May 2016, doi: 10.1364/JOCN.8.000320.

[281] S. Sarkar, “Revisiting Prime Power {RSA},” *Discret. Appl. Math.*, vol. 203, pp. 127–133, 2016, doi: http://dx.doi.org/10.1016/j.dam.2015.10.003.

[282] X.-J. Lin, L. Sun, and H. Qu, “An efficient RSA-based certificateless public key encryption scheme,” *Discret. Appl. Math.*, p., 2017, doi: http://dx.doi.org/10.1016/j.dam.2017.02.019.

[283] E. Lüy, Z. Y. Karatas, and H. Ergin, “Comment on ‘An Enhanced and Secured {RSA} Key Generation Scheme (ESRKGS),’” *J. Inf. Secur. Appl.*, vol. 30, pp. 1–2, 2016, doi: http://dx.doi.org/10.1016/j.jisa.2016.03.006.

[284] D. Pointcheval, “Rabin Cryptosystem,” in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg, Ed. Boston, MA: Springer US, 2005, pp. 501–502. doi: 10.1007/0-387-23483-7_339.

[285] M. Elia, M. Piva, and D. Schipani, “The Rabin cryptosystem revisited,” *Appl. Algebr. Eng. Commun. Comput.*, vol. 26, no. 3, pp. 251–275, 2015, doi: 10.1007/s00200-014-0237-0.

[286] R. Amin and G. P. Biswas, “Remote Access Control Mechanism Using Rabin Public Key Cryptosystem,” in *Information Systems Design and Intelligent Applications: Proceedings of Second International Conference INDIA 2015, Volume 1*, J. K. Mandal, S. C. Satapathy, M. Kumar Sanyal, P. P. Sarkar, and A. Mukhopadhyay, Eds. New Delhi: Springer India, 2015, pp. 525–533. doi: 10.1007/978-81-322-2250-7_52.

[287] W. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr, “RSA/Rabin Bits are 1/2 + 1 / Poly (Log N) Secure,” in *25th Annual Symposium onFoundations of Computer Science, 1984.*, Oct. 1984, pp. 449–457. doi: 10.1109/SFCS.1984.715947.

[288] M. Kaminaga, H. Yoshikawa, A. Shikoda, and T. Suzuki, “Crashing Modulus Attack on Modular Squaring for Rabin Cryptosystem,” *IEEE Trans. Dependable Secur. Comput.*, vol. PP, no. 99, p. 1, 2016, doi: 10.1109/TDSC.2016.2602352.

[289] L. Harn and T. Kiesler, “Improved Rabin’s scheme with high efficiency,” *Electron. Lett.*, vol. 25, no. 11, pp. 726–728, May 1989, doi: 10.1049/el:19890492.

[290] K. S. Selvi and T. Vaishnavi, “Rabin PublicKey Cryptosystem for mobile authentication,” in *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, Mar. 2012, pp. 854–860.

[291] D. Boneh, “Simplified OAEP for the RSA and Rabin Functions,” in *Advances in Cryptology --- CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19--23, 2001 Proceedings*, J. Kilian, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 275–291. doi: 10.1007/3-540-44647-8_17.