International Journal of Grid and Distributed Computing
Vol. 10, No. 11 (2017), pp.79-98
http://dx.doi.org/10.14257/ijgdc.2017.10.11.08

Identifying Open Research Problems in Cryptography by Surveying Cryptographic Functions and Operations

Rahul Saha[1], G. Geetha[2], Gulshan Kumar[3] and Hye-Jim Kim[4]

[1,3]*School of Computer Science and Engineering, Lovely Professional University, Punjab, India*
[2]*Division of Research and Development, Lovely Professional University, Punjab, India*
[4]*Business Administration Research Institute, Sungshin W. University, 2 Bomun-ro 34da gil, Seongbuk-gu, Seoul, Republic of Korea*

Abstract

Cryptography has always been a core component of security domain. Different security services such as confidentiality, integrity, availability, authentication, non-repudiation and access control, are provided by a number of cryptographic algorithms including block ciphers, stream ciphers and hash functions. Though the algorithms are public and cryptographic strength depends on the usage of the keys, the ciphertext analysis using different functions and operations used in the algorithms can lead to the path of revealing a key completely or partially. It is hard to find any survey till date which identifies different operations and functions used in cryptography. In this paper, we have categorized our survey of cryptographic functions and operations in the algorithms in three categories: block ciphers, stream ciphers and cryptanalysis attacks which are executable in different parts of the algorithms. This survey will help the budding researchers in the society of crypto for identifying different operations and functions in cryptographic algorithms.

Keywords: *cryptography; block; stream; cipher; plaintext; ciphertext; functions; research problems*

1. Introduction

Cryptography [1] in the previous time was analogous to encryption where the main task was to convert the readable message to an unreadable format. The process was complex as the sender needed to send the decoding technique personally to the receiver. Modern cryptography, in comparison, is easy to manage as it is an intersection of mathematics, computer science and electrical engineering. Different types of transformations are used in modern cryptography. The schemes of modern cryptography are computationally secure because practical breaking down of such cryptosystems is infeasible with any practical means.

The growth of cryptographic technology has emerged a number of legal issues in this digital information era. In our daily life, we often experience the cryptographic measures and methods. From electronic mail to cellular communications, from secure web browsing to digital cash, everywhere the applications of cryptographic algorithms is observed. Therefore, the information systems and the data communication among such information systems are dependable on such cryptographic schemes to make the process enough secure.

Received (July 20, 2017), Review Result (October 30, 2017), Accepted (November 6, 2017)

ISSN: 2005-4262 IJGDC

Different applications [2] of cryptography demand for different types of security services. Cryptography provides the following services [1].

Confidentiality: the data is hidden from the third party.
Authentication: the data is handled by only the legitimate person.
Integrity: the data is not changed in the process of communication.
Non-repudiation*:* the responsibility of sending and receiving the data cannot be avoided by either sender or the receiver.

There are two basic category of cryptography [1, 2, 3] exist in this domain- Symmetric cryptography and Asymmetric cryptography [4]. Both the categories convert the plaintext (original message) to ciphertext (encrypted message) with the help cipher and the vice versa is done with the help of decipher. Symmetric key cryptography [7] deals with the usage of a single shared key between a sender and a receiver. Asymmetric key cryptography [4] deals with two keys for an encryption-decryption process. They key which is public is used for encryption process and another key called as private key, which is mathematically related with the public key, is used to decrypt the encoded message. Both the symmetric key and asymmetric key cryptography have their relative advantages and disadvantages. Symmetric is faster than the asymmetric key process and has more strength with a large size of key. On the other hand, asymmetric key process is slower though it maintains better scalability. It can also provide authentication and non-repudiation which is not provided by the former. Another way of categorization [1] of the cryptographic algorithms depends upon the data it works upon. One of them is called as block ciphers which deal with the predefined size of the block of data on which the cipher is applied. For example, DES algorithm works with 64 bit data block. On the other hand, ciphers like RSA works on data stream where each bit of data is encrypted with the cipher until the total bits of data are encrypted.

A number of algorithms are getting developed each day. Some of them are eventually getting approved by different standard organization and some of them are getting rejected in comparison with others. This selection of the cryptographic algorithms is based upon some metrics which helps to provide a utility for Common Criteria Level of Assurance (LA). These metrics assist in developing a framework for specifying the appropriate measures to design a cryptographic algorithm. Although all the characteristics [2, 3] of the algorithms cannot be quantified, the parameters or the metrics of such algorithms can be objective or subjective. The type of the algorithm is developed is depending upon the key structure used for the algorithm such as symmetric or asymmetric mentioned earlier. But the type of the algorithm leads to the measurement of complexity and time consumption factors of the algorithms. Another subjective metric is plaintext format requirement such as blocks of data or streams of bits. Moreover, the structure of the algorithm also makes an effect on the strength of the algorithms. For example, most of the block ciphers use Feistel structure [8, 9] which emphasizes on permutation of data blocks at the starting of each round and in the completion of all rounds. This attributes the algorithm with security by providing confusion and diffusion. The type of functions used in the algorithm is also an important factor. For example, the algorithm for confidentiality must emphasis on the key for better perspective. But, the algorithm for authentication or integrity must consider the hash functions that are used to develop the algorithm. The security aspect of all the cryptographic algorithms is a function of length of key. Higher the bits, more is security and less bits, less security as the less bit counts of keys are vulnerable to the brute force attacks, factoring attacks or discrete log attacks. Rounds are specifically not having any threshold as per the metric concern. More number of rounds is adopted to generate more confusion and diffusion property [6] in the algorithm which is a desired measurable characteristic of a good cryptographic algorithm. The complexity of a cryptographic algorithm depends upon the setup of encryption, decryption and the key. Basically, the

round functions consist of different bitwise operations, modular arithmetic. This metric is critical because the cryptographic algorithm must not be too complex as the algorithms are used even for resource constraint environments. The main objectivity of this metric is to execute a parallelism in the operations. Cryptography does not deal with only developing encryption or decryption algorithms; the algorithms need to be proved to have strength by the cryptanalysis process [5]. Brute force attack, factoring, linear and differential cryptanalysis are some of the best known attacks that are executed on the cryptographic algorithms to identify the strength of the algorithms. The number of steps performed for the attack, the time requirement and the type plaintext or ciphertext used for the attack are also the parts of this metric.

2. Review of Literature

The review of literature has been categorized in three parts for our research work.

Category 1: Identifying the functions used for the block ciphers

Category 2: Identifying the functions used for the stream ciphers

Category 3: Identifying the cryptanalytic attacks on ciphers that identify the functional relation in the round function.

2.1. Category 1

We have observed 60 block ciphers. Bitwise XOR has been used in all the algorithms as it provides permutation objectives. Apart from XOR, the usage of bitwise AND, OR, NOT has also been seen. Some of the specialized functions have also been used such Fast Fourier Transformation [115], Hadamard Transformation [85], Affine Transformation [155], Self inverse Transformation [165], Linear and non-linear transformations [51, 66, 103, 117, 140]. The summarized table for the utilized functions and operations of all the block ciphers has been shown in Table 1.

Table 1. Literature Review of Block Ciphers

Algorithm	Ref	Year	Block Size	Key Size	Summary of used function
Lucifer	[10]	1971	48, 32 or 128 bits	48, 64 or 128 bits	Uses the linear and nonlinear transformation functions, Arithmetic operations like AND, XOR.
DES	[11]	1975	64 bits	56 bits + 8 parity bits)	Bitwise addition modulo, XOR along with substitution and permutation boxes
DESX	[12]	1984	64 bits	184 bits	Same as DES, but input is XORed with 64 bit key material beforehand and similarly the output is XORed with another 64 bit key part
FEAL	[13]	1987	64 bits	64 bits	XOR operations
RC2	[14]	1987	64 bits	8–1024 bits, in steps of 8 bits; default 64 bits	It uses two's-complement addition, bitwise AND, bitwise XOR operation, bitwise COMPLEMENT, exponentiation operation

					and modulo operation.
Khafre	[15]	1989	64 bits	512 bits	OR and XOR operations
Khufu	[15]	1989	64 bits	512 bits	Key whitening with XOR operation.
FEALNX	[16]	1990	64 bits	128 bits	XOR operations as FEAL
LOKI	[17]	1990	64 bits	64 bits	Non linearity used in S-box and key whitening
Redoc II	[18]	1990	80- bits	160 bits	Only XOR operations
IDEA	[19]	1991	64 bits	128 bits	Key generator uses XOR operation, multiplication modulo $2^{16} + 1$ and addition modulo 2^{16}
Blowfish	[20]	1993	64 bits	32-448 bits	Modulo operations and XOR
Safer K-64	[21]	1993	64 bits	64 bits	Sub keys are added using either addition modulo 256 or XOR. Pseudo hadamard transform is used here as a diffusion layer.
VINO	[22]	1993	64 bits	128 bits	In round scheme bitwise XOR is used. Addition modulo is used.
GOST	[23]	1994	64 bits	256 bits	Addition modulo and left rotation
MacGuffin	[24]	1994	64 bits	128 bits	XOR operations in different stages.
RC5	[25]	1994	32, 64 or 128 bits (64 suggested)	0 to 2040 bits (128 suggested)	Bitwise XOR is used.
TEA	[26]	1994	64 bits	128 bits	Bitwise XOR is used.
Misty1	[27]	1995	64 bits	128 bits	Key Scheduling, input function, output function all of them use the bitwise AND, bitwise inclusive OR, multiplication, quotient, remainder operations
Akelarre	[28]	1996	128 bits	128 bits	Uses the basics of IDEA and RC5.
BEAR	[29]	1996	On the order of 2^{13} to 2^{23} bits or more	160 or 128 bits	Bitwise XOR
CAST128	[30]	1996	64 bits	40 to 128 bits	modular addition and subtraction, and XOR operations, bent function that takes several inputs and gives one output, each of which has two possible values (such as 0 and 1, or true and false)
LION	[29]	1996	On the order of 2^{13} to 2^{23} bits or more	160 or 128 bits	Bitwise XOR

Shark	[31]	1996	64 bits	128 bits	It is a six round Substitution Permutation-network (XOR and rotation) that uses linear and non-linear transformation layers. MDS matrix is used by the linear and the nonlinear layer is composed of eight 8×8-bit S-boxes based on the function $F(x) = x^{-1}$ over $GF(2^8)$.
ICE	[32]	1997	64 bits	64 bits	XOR operations
Square	[33]	1997	128 bits	128 bits	Linear and nonlinear transformations
XMX	[34]	1997	Variable	Variable, equal to block size	the only operations it uses are XORs and modular multiplications
AES	[35]	1998	128 bits	128, 192 or 256 bits	XOR operation in different stages, and multiplication modulo operator, Substitution and permutation process
BKSQ	[20]	1998	96 bits	96, 144, 192 bits	Uniform round transformations use XOR operations.
CAST256	[36]	1998	128 bits	128, 160, 192, 224, 256 bits	Same as CAST128
CS Cipher	[37]	1998	64 bits	128 bits	Round function is based upon Fast Fourier Transformation function
Crypton	[38]	1998	128 bits	128, 192, 256 bits	In linear transformation procedure, bit permutation depends on functions that utilize bitwise AND, XOR, OR and complement too.
DEAL	[39]	1998	128 bits	128, 192, 256 bits	Uses DES as round function.
DFC	[40]	1998	128 bits	128, 192, 256 bits	It uses a round function having a single 6×32-bit S-box, as well as an affine transformation mod $2^{64}+13$.
E2	[41]	1998	128 bits	128, 192, 256 bits	It uses an input transformation and output transformation. Such transformations use modular multiplication. The round function uses XORs and S-box lookups.
Frog	[42]	1998	128 bits	128, 192, 256 bits	All operations are byte-wide and consist of XORs and substitutions.

Hasty Pudding	[43]	1998	variable	Variable	Key expansion uses multiplication, addition and shifting. It also uses bitwise XOR.
LOKI97	[44]	1998	128 bits	128, 192, 256 bits	Round function is dependable on S-boxes which are designed to be highly non-linear and use XORs.
Magenta	[45]	1998	128 bits	128, 192, 256 bits	Different function uses in the whole process of MAGENTA. The functions utilize XOR operation.
Mars	[46]	1998	128 bits	128, 192, 256 bits	Forward core layer and the backward core layer use a combination of S-box lookups, multiplications, data-dependent rotations, additions, and XORs. Addition, subtraction and XOR are used for mixing data and key values.
RC6	[47]	1998	128 bits	128, 192, or 256 bits	Addition, subtraction, XOR, multiplication
Rijndael	[20]	1998	128 bits	128, 192, or 256 bits	It's an AES proposal that uses almost all the operations defined in AES.
Serpent	[48]	1998	128 bits	128, 192, or 256 bits	Mixing operations, S-boxes, linear transformation.
Skipjack	[49]	1998	64 bits	80 bits	Stepping rules use bitwise XOR operation
Twofish	[50]	1998	128 bits	128, 192 or 256 bits	Uses Pseudo Hadamard Transformation
Triple-DES	[51]	1998	64 bits	168, 112 or 56 bits	Same as DES
UES	[52]	1999	128 bits	128, 192 or 256 bits	Two parallel DES.
Khazad	[53]	2000	64 bits	128 bits	Substitution-permutation network with XOR operations for permutations.
Anubis	[54]	2000	128 bits	128 to 320 bits in steps of 32 bits	It uses pseudo-random S-box depends upon XOR operations. The newer version of Anubis is called the "tweaked" version.
Camellia	[55]	2000	128 BITS	128, 192, 256 bits	Bitwise OR, AND, XOR and complement.
DFCv2	[56]	2000	128 bits	128, 192, 256 bits	The round function uses a single 6×32-bit S-box, as well as an affine transformation mod $2^{64}+13$
Grand Cru	[57]	2000	128 bits	128 bits	Based largely on AES

Hierocrypt L1	[58]	2000	64 bits	128 bits	XOR and concatenation used for different functions like whitening, s-box, rounding function
Kasumi	[59]	2000	64 bits	128 bits	Bitwise XOR and Bitwise AND
Nimbus	[60]	2000	64 bits	128 bits	Bitwise XOR
Noekeon	[61]	2000	128 bits	128 bits	Self inverse transformation
NUSH	[62]	2000	64, 128, or 256 bits	128, 192, or 256 bits	No S-box is used; different bitwise operations such as AND, OR, XOR, modular addition, and bit rotation are used here.
Q	[63]	2000	128 bits	128, 192 or 256 Bits	Uses S boxes that depends on XOR
SC2000	[64]	2000	128 bits	128, 192, or 256 bits	combination Substitution Permutation Network and Feistel network
SHACAL	[65]	2000	160/256 bits	128/ 512 bits	Compression function
PRESENT	[66]	2007	64 bits	80 or 128 bits	XOR operation
KATAN and KTANTAN	[67]	2009	32, 48, or 64-bit	80 Bits	two nonlinear Boolean functions that mainly consists of XOR operations
LED	[68]	2012	64 bits	64 /128 bits	XOR operations in different stages
Simon and Speck	[69]	2015	32, 48, 64, 96 or 128 bits	64/72/ 96/ 128/144/192 or 256 bits	Bitwise XOR, AND, Left Circular Shift, Right Circular Shift and modular addition

We have identified five major categories of he functions and operations used in different block ciphers and how much percentage of our observed block ciphers are using those operations have been shown in the pie-chart shown in Figure-1. The categories are as follows.

Arithmetic and bitwise operations: This category includes all the bitwise operators such as AND, OR, XOR, NOT, left shift, right shift, modulo operations, arithmetic addition.

Fast Fourier Transformation: It computes the discrete Fourier Transform (DFT) of a sequence or its inverse. It factorizes the DFT matrix into a product of sparse factors which reduces the complexity from O (N^2) to O (N log N).

Hadamard Transformation: Hadamard transformation is an example of Fourier transforms which is linear and works on 2^m real numbers. It is built out of size-2 discrete Fourier transforms, and is in fact equivalent to a multidimensional DFT of size $2 \times 2 \times \ldots \times 2 \times 2$. It decomposes an arbitrary input vector into a superposition of Walsh functions which take the values of +1 and -1 only with the subintervals of dyadic fraction whose denominator is a power of 2.

Affine Transformation: Affine transformations include translation, scaling, homothety, similarity transformation, reflection, rotation, shear mapping, and compositions of them in any combination and sequence. If X and Y are affine spaces, then every affine transformation $f : X \rightarrow Y$ is of the form $x \rightarrow M x + b_x$, where M is a linear transformation on X and b is a vector in Y. Unlike a purely linear transformation, an affine map need not preserve the zero point in a linear space. Thus, every linear transformation is affine, but not every affine transformation is linear.

Linear and non-linear transformation: Linear transformations deal with linear algebraic maths. The non-linear transformations do not rely on one-to-one linearity, different substitution equations can be made up using such transformations.

Others: In this category we have put such algorithms which do not apply any function or operation related to above categories such self-inverse and compression function.

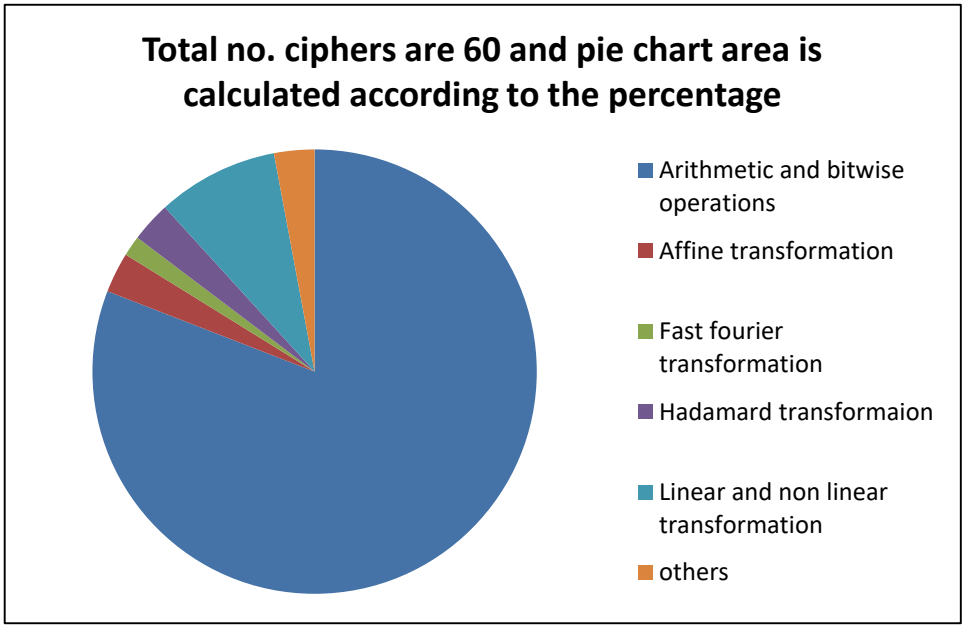

Figure-1. Summary of Work of Operations in Block Ciphers

2.2. Category 2

We have surveyed 32 stream cipher algorithms and also identified the functions in those algorithms.

Table 2. Literature Review of Stream Ciphers

Stream Cipher Name	Reference	Year	Functions and operations
RC4	[70]	1987	Key scheduling and random number generator uses bitwise XOR
A5/1 and A5/2	[71]	1989	linear feedback shift registers with irregular clocking and a non-linear combiner
FISH	[72]	1993	Lagged Fibonacci generators and the shrinking generator
WAKE	[73]	1993	XOR operations for S-boxes
Pike	[74]	1994	Lagged Fibonacci generators
ISAAC	[75]	1996	Uses pseudo random number generator
SEAL	[76]	1997	Pseudo random function family
PANAMA	[77]	1998	Step right up
MUGI	[78]	1998	Linear transformation
E0	[79]	2000	XOR operators
Scream	[80]	2002	The round function is based on the AES-round function, but is narrower, 64 bits instead of 128 bits.
Rabbit	[81]	2003	The mixing function uses a g-function based on arithmetical squaring, and the ARX operations

			including logical XOR, bit-wise rotation with hard-wired rotation amounts, and addition modulo 2^{32}.
Snow	[82]	2003	The cipher consists of a combination of a LFSR and a Finite State Machine (FSM) where the LFSR also feeds the next state function of the FSM.
Sober 128	[83]	2003	32-bit XOR and addition modulo 2^{32}
Turing	[84]	2003	Pseudo hadamard function, Linear Feedback Shift Register (LFSR)
Trivium	[85]	2004	Primarily XOR and AND
Sosemanuk	[86]	2004	It uses a combination of a Linear Feedback Shift Register (LFSR) and a Finite State Machine (FSM) where the LFSR also feeds the next state function of the FSM.
Salsa20	[87]	2004	It is built on a pseudorandom function based on add-rotate-xor (ARX) operations comprises of 32-bit addition, bitwise addition (XOR) and rotation operations.
Py	[88]	2004	XOR, addition modulo 2^{32}
Phelix	[89]	2004	The cipher uses only the operations of addition modulo 2^{32}, exclusive or, and rotation by a fixed number of bits.
HC-256	[90]	2004	Bitwise arithmetic operations
Grain	[91]	2004	Linear Feedback Shift Register and Non Linear Feedback Shift Register
CryptMT	[92]	2005	Linear generator and filter
VEST	[93]	2005	It uses a balanced T-function that can also be described as a bijective nonlinear feedback shift register with parallel feedback (NLPFSR) or as a substitution-permutation network, which is assisted by a non-linear RNS-based counter.
Achterbahn-128/80	[94]	2006	Nonlinear Feedback Shift Registers
Quad	[95]	2006	It works on GF ( ) functions.
WG family	[96]	2008	Primarily XOR operation, besides Linear Feedback Shift Registers and Finite Field operations
Rakaposhi	[97]	2009	Dynamic Linear Feedback Shift Register, Non Linear Feedback Shift Register, Linear Feedback Shift Register
ZUC	[98]	2010	It uses a 16-stage Linear Feedback Shift Register
MICKEY	[99]	2011	Non-linear functions
Spritz	[100]	2014	Arithmetic operations
Espresso	[101]	2015	Non Linear Feedback Shift Register, nonlinear output function

We have identified seven major categories of he functions and operations used in different stream ciphers and how much percentage of our observed stream ciphers are using those operations have been shown in the pie-chart shown in Figure-2. The categories that we have identified in this section are listed below.

Bitwise and arithmetic operations: This category includes all the bitwise operators such as AND, OR, XOR, NOT, left shift, right shift, modulo operations, arithmetic addition.

Pseudorandom operations: It uses pseudorandom number generators.

Linear Feedback Shift Registers (LFSR): It is a shift register whose input bit is controlled by the XOR operation of some previous bits' values of the overall shift register. As the input depends on the previous stage values, it works as a feedback.

Non-Linear Feedback Shift Registers (NLFSR): It is an extension of LFSR but its theory is not completed yet to support the general implementation. The period of non-linearity must be high enough.

Lagged Fibonacci functions: It is a type of pseudorandom number generator which generalizes the basic Fibonacci series concept. The operation used here can be any binary operation. The relation is given as:

$S \equiv S_{n-j} * S_{n-k} \bmod (m), 0 < j < k$,where m is power of 2 , generally 2^{32} or 2^{64}

Finite State Machine (FSM): It is a mathematical model used for digital logic circuits in cryptography. He machine will be in only one state at a time or in a finite number of states.

Others: It includes different functions such as Galois Field functions, Hadamard functions, State-right up functions that do not imply the previous categories mentioned.

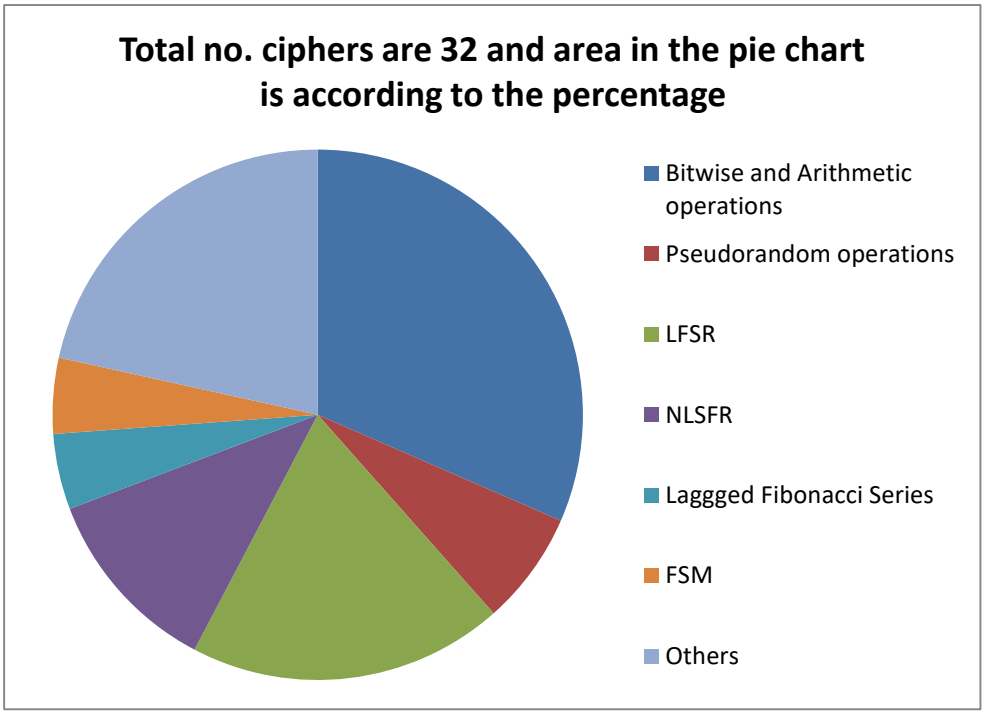

Figure-2. Summary of Work of Operations in Stream Ciphers

2.3. Category 3

We have reviewed 16 cryptanalytic attacks and also identified how these attacks analyse the round functions and operations in those algorithms.

Table-3. Literature Review of Cryptanalytic Attacks on Block Ciphers and Stream Ciphers

Name of the attack [Ref No.]	Year	Description	Attacks on
Slide Attack [102]	1977	The slide attack does not consider number of rounds in a cipher. It works by analysing the key schedule and exploiting weaknesses in it to break the cipher. This attack tries to break down a cipher into multiple rounds of an identical F function to identify cyclic key schedule. The F function must be vulnerable to a known-plaintext attack.	New Data Seal (NDS)
Differential Cryptanalysis Attack [103]	1990	It is a chosen plaintext attack. It calculates a constant difference among pairs of plaintexts; Then the differences of the corresponding ciphertexts are calculated to determine statistical patterns in their distribution.	DES
Linear Cryptanalysis Attack [104]	1992	It identifies the affine approximation	FEAL, DES
Davies' attack [105] [106]	1993	It is a known-plaintext attack. It detects the non-uniform distribution of the outputs of plaintext-ciphertext pairs of adjacent S-boxes.	DES
Timing Attack [107]	1993	It analyses the time taken to execute cryptographic algorithms even for a single logical operation.	Deffie-Hellman, RSA, DSS
Related -key Attack [108]	1994	It can observe the operation of a cipher under several different keys whose values are initially unknown, but where some mathematical relationship connecting the keys is known to the attacker.	Kasumi, WEP
Partitioning Cryptanalysis [109] [110]	1995	This attack is an extended version of linear cryptanalysis where affine transformations are replaced by balanced Boolean functions.	DES, CRYPTON
Side Channel Attack [111]	1995	This attack is based on information gained from the physical implementation of a cipher. It depends on the sound, power, time, electromagnetic fields and many more.	RSA
Integral Cryptanalysis [112]	1997	It is a chosen plaintext attack where some bits of the plaintexts are kept constant and other bits are varied. This will generate the value 0 for an XOR sum, and the XOR sums of the all the corresponding sets of cipher texts reveals the information about the cipher's operation.	SQUARE, IDEA, Camellia, Skipjack, Khazad
Interpolation Attack [113]	1997	It uses simple quadratic, or a polynomial or rational function over a Galois field to represent a S-box. Coefficients of the generated equations are determined by standard Lagrange interpolation techniques.	SNAKE, SHARK

Boomerang Attack [114]	1999	It depends upon differential cryptanalysis. The attack generates a "quartet" structure at a point halfway through the cipher.	COCONUT98, KASUMI
Mod n cryptanalysis [115]	1999	It exploits the differences in how the cipher operates over equivalence classes modulo n. It uses Fermat number concept.	RC5
Amplified boomerang Attack [116]	2001	Same as boomerang but the selection of input output pairs need to be strict to get into the collision to analyse relation among the pairs.	MARS- 11 rounds, SERPENT- 8 rounds
Rectangle Attack [117]	2001	Same base as boomerang but with the modifications of more number of quartets, sorting of piles of wrong beta values to execute the attack to get the quartet values and the gamma dash values instead of gamma for all possible differential characteristics.	SERPENT
XSL Attack [118]	2002	It generates quadratic simultaneous equation system and solves the equations with extended sparse linearization.	SERPENT, AES
Rotational Attack [119], [120]	2010	It depends on ARX operations: Addition, rotation, modulo and XOR.	Threefish

We have identified four major categories of the functions and operations where different cryptanalytic attacks have been executed. In the Figure-3, we have shown a pie-chart depicting how much percentage of our observed cryptanalytic attacks is executed on a particular category of operations.

Attacks on key schedule: These attacks identify the key entirely or partially depending upon the internal values of the operations and functions of plaintext ciphertext combinations.

Attacks on statistical relation between plaintext and ciphertext: These attacks identify the statistical relationship between plaintext and ciphertext by utilizing the round functions as a whole or partially.

ARX operations: These attacks work on addition, rotation and XOR operations in the functions of ciphers.

Equation and transformation analysis: The attacks on this category deals with analysing different linear and non-linear equations to solve the seed values to identify the plaintext or keys according to the possibility of occurrences.

Analysing physical factors: In this category, the attacks deal with different physical factors such as sound, electromagnetism, power level evaluation while executing a particular function.

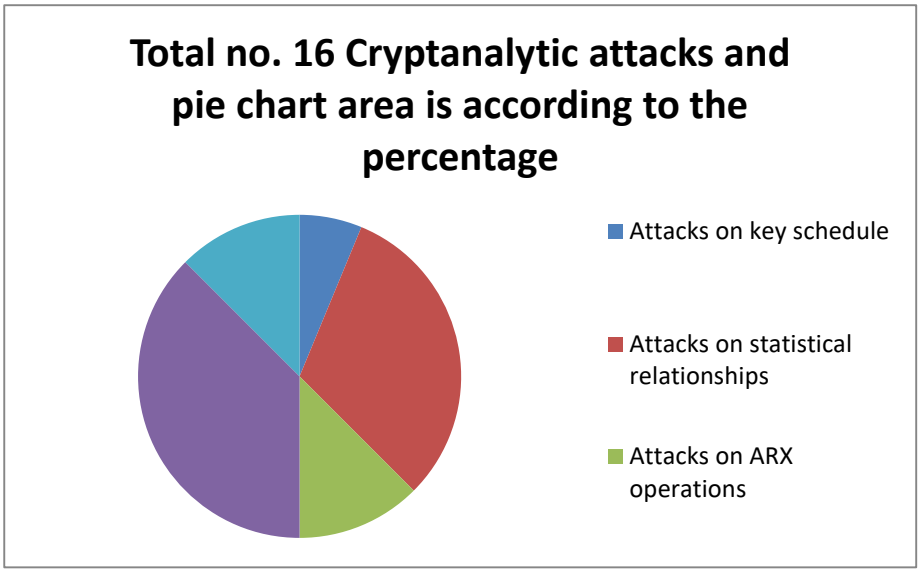

Figure-3. Summary of Existing Attack on Different Operation Categories

3. Timeline Analysis

The trend of the operations and functions in the algorithms and the trend of the attacks on the algorithms have also been analysed according to the timeline. We have categorized the timeline into 1970s, 1980s, 1990s, 2000s and 2010s. The trends are shown in Figure-4, Figure-5 and Figure-6 respectively for block ciphers, stream ciphers and attacks.

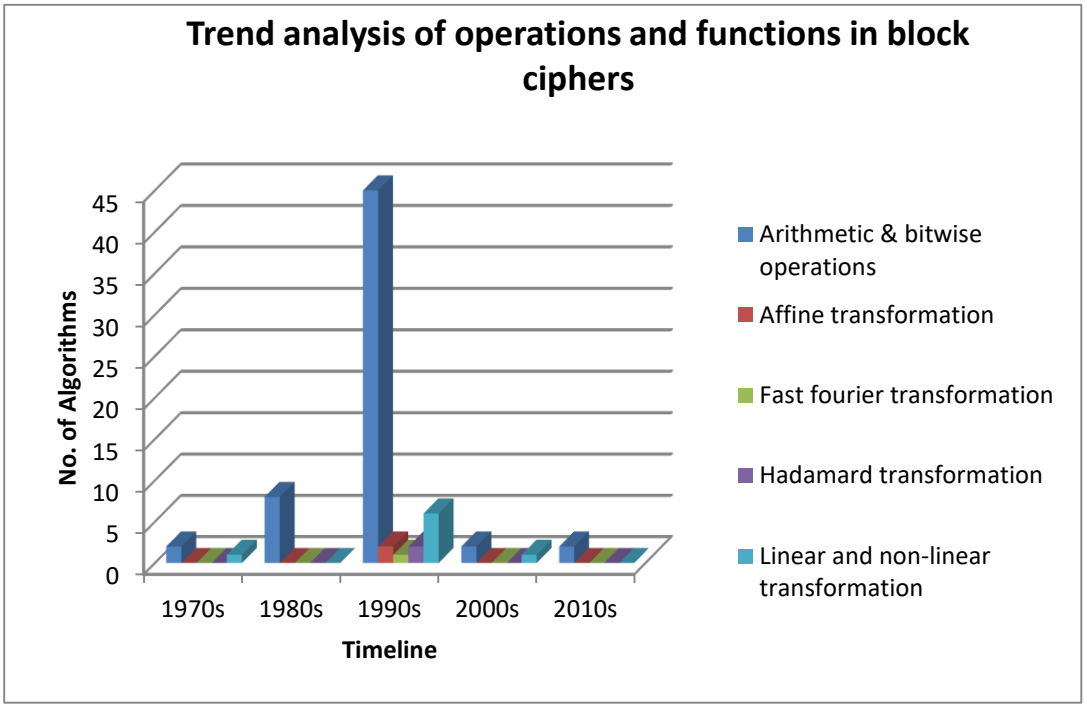

Figure-4. Trend Analysis of Block Cipher Operations and Functions

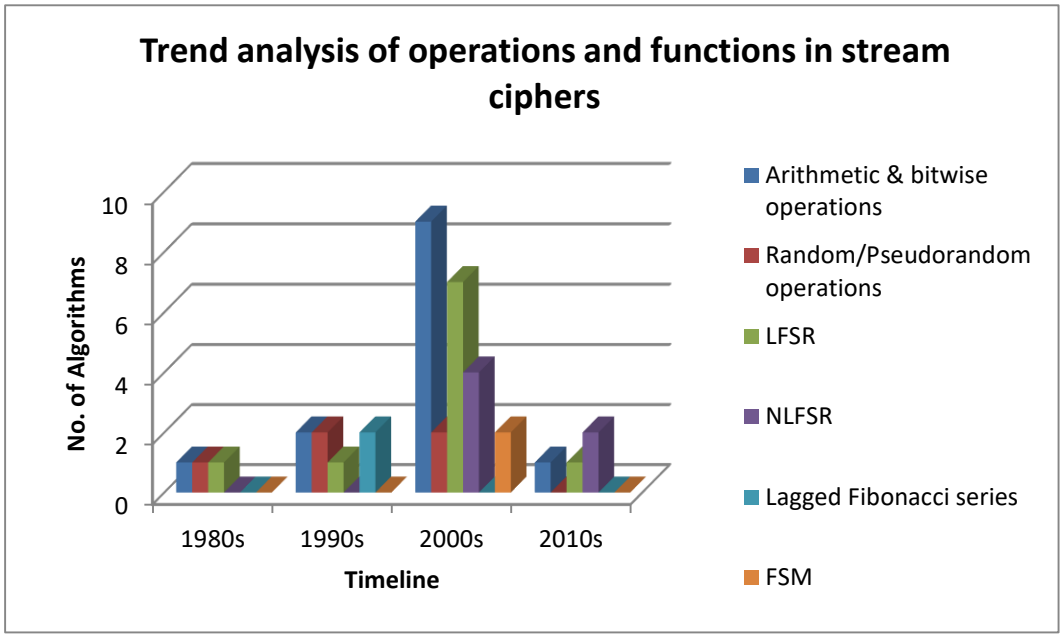

Figure 5. Trend Analysis of Stream Cipher Operations and Functions

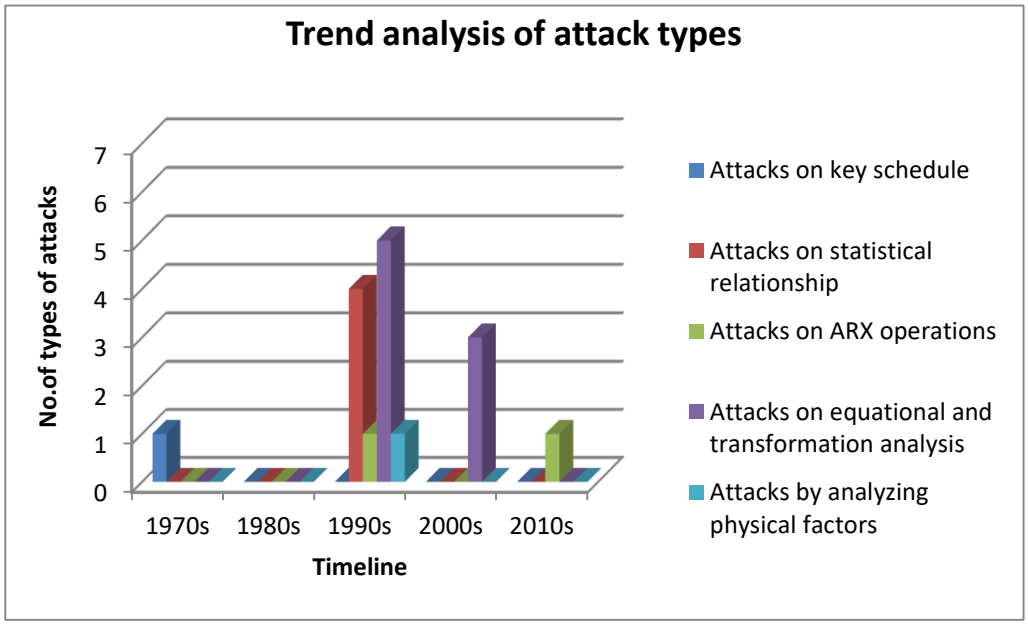

Figure 6. Trend Analysis of Types of Cryptanalytic Attacks

The trend in Figure-4 shows that use of arithmetic and bitwise operations was increasing exponentially till 2000s but has got a slow down further. But as compared to other operations and function types, these are used more. Moreover, the three transformation functions such as Affine, Hadamard and Fast Fourier are only used in 1990s. Similarly for stream ciphers, as shown in Figure-5, the use of arithmetic and bitwise operations, pseudorandom operations and linear shift back registers are significantly high. Figure-6 shows the trend of the attacks. According to the analysed trend, attacks on the bitwise and arithmetic operations are still in process whereas attacks on statistical relationships between plaintext-ciphertext or plaintext-key or ciphertext-key and attacks on equational and transformation analysis is decreasing.

4. Open Research Problems

After organizing the literature review and executing the analysis of the trend, as shown in the previous section, some open research problems have been identified. These research problems can help the upcoming researchers in the field of cryptography. The cryptologists can work further on these given aspects to enrich this crypto world with some valuable research work for future. Some of the future research problems are given below.

New ciphers using different equational or transformations such as Affine, Hadamard, Fast Fourier, Linear-nonlinear need to be explored as attacks are significantly high on such category of operations in ciphers.

The performance of Fibonacci series or pseudorandom operations in block ciphers can be researched as it is only used in stream ciphers till date.

New stream ciphers can be designed with more sophisticated pseudorandom or random operations as less work have been executed in this domain.

The effect of side channel attack on all types of cryptographic algorithms using various factor need to be analysed further.

5. Conclusion

We have observed 60 block ciphers, 32 stream ciphers and 16 cryptographic attacks. The survey identifies which domains of the mathematical functions are primarily dominant in cryptographic procedures. The block and stream ciphers are primarily using the arithmetical and bitwise operations. From the literature review, it is also observed that maximum cryptanalytic attacks have been executed on statistical relationship between ciphertext and plaintext. From the attackers view point, these statistical relationships are inferred from the functional operations in the algorithms stepwise or even from implications of the identical operations. The open research problems identified in the above section will be helpful for further research in the domain of cryptography using different functions and operations in the algorithms.

References

[1] H.C.A. Van Tilborg, "Fundamentals of Cryptology", Kluwer Academic Publishers, New York, **(2000)**.
[2] B. Schneier, "Applied Cryptography", New York: John Wiley & Sons Inc., **(1996)**.
[3] S. V. Kartalopoulos, "A primer on cryptography in communications", IEEE Commun. Mag., vol. 44, no. 4, **(2006)**, pp. 146–151.
[4] W. Diffie, "The First Ten Years of Public-Key Cryptography", Proceedings of the IEEE, vol. 76, no. 5, **(1988)**, pp. 560-577.
[5] H. M. Heys, "A Tutorial on Linear and Differential Cryptanalysis", University of Newfoundland, St. John's, NF, Technical report, Electrical and Computer Engineering, **(2004)**.
[6] C. E. Shannon, "Communication Theory of Secrecy Systems", Bell System Technical Journal, vol. 28, **(1949)**, pp. 656-715.
[7] W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson and M. Weiner M. Blaze, "(Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security. http://www.schneier.com/ paper-keylength.html, **(1996)**.
[8] Substitution-permutation network. [Online]: https://en.wikipedia.org/wiki/Substitutionpermutation_network.
[9] Feistel network operation. [Online]. http://www.wikiwand.com/simple/Feistel_cipher.
[10] J. Smith, "The design of Lucifer: a cryptographic device for data communications", IBM T.J. Watson Research Center, Yorktown Heights, N.Y., USA, Technical report, **(1971)**.
[11] "Data Encryption Standard", National Institute of Standards and Technology, Federal Information Processing Standards Publication, **(1999)**, pp. 46-3.
[12] P. Rogaway and J. Kilan, "How to protect DES against exhaustive key search", Journal of Cryptology, vol. 14, no. 1, **(2001)**, pp. 17-35.
[13] S. Shimizu and Miyaguchi, "Fast Data Encipherment Algorithm FEAL", in Advances in Cryptology – Eurocrypt, vol. 87, **(1987)**.
[14] R. Rivest, "A description of the RC2(r) encryption algorithm", http://www.ietf.org, **(1998)**.

[15] R. Merkle, “Fast software encryption functions”, in Advances in Cryptology - Crypto'90, A. Menezes and S. Vanstone, Ed. Santa Barbara, California, USA: Springer-Verlag, **(1990)**, pp. 476-501.
[16] S. Miyaguchi, “The FEAL cipher family”, in Advances in Cryptology - Crypto'90, A. Menezes and S. Vanstone, Ed. California, USA: Springer-Verlag, **(1990)**, pp. 627-638.
[17] M. Kwan, J. Pieprzyk and J. Seberry L. Brown, “Improving resistance to differential cryptanalysis and the redesign of LOKI”, in Advances in Cryptology - ASIACRYPT'91, R. Rivest, and Matsumoto H. Imai, Ed. Fujiyoshida, Japan: Springer-Verlag, **(1991)**, pp. 36-50.
[18] M. Wood and T. Cusick, “The RedocII cryptosystem”, in Advances in Cryptology - Crypto'90, A. Menezes and S. Vanstone, Ed. California, USA: Springer-Verlag, **(1990)**, pp. 545-563.
[19] X. Lai, J. Massey and S. Murphy, “Markov ciphers and differential cryptanalysis”, in Advances in Cryptology - Eurocrypt'91, D. Davies, Ed. Brighton, UK: Springer-Verlag, **(1991)**, pp. 17-38.
[20] J. Daemen and V. Rijmen, “The Block Cipher Rijndael”, in Smart Card Research and Applications, Bruce Schneier Jean-Jacques Quisquater, Ed. Belgium: Springer-Verlag, **(2000)**, pp. 277-284.
[21] J. Massey, “SAFER-K: a byte-oriented block-ciphering algorithm”, in Fast Software Encryption, R. Anderson, Ed. Cambridge, UK: Springer-Verlag, **(1994)**, pp. 1-17.
[22] W. Wlfovics and A. Di Porto, “VINO: a block cipher including variable permutations”, in Fast Software Encryption, R. Anderson, Ed. Cambridge, UK: Springer-Verlag, **(1994)**, pp. 205-210
[23] Government Committee of the USSR for Standards, GOST - Gosudarstvennyi Standard, Cryptographic protection for data, **(1989)**, pp. 28147-89.
[24] M. Blaze and B. Schneier, “The MacGuffin cipher algorithm”, in ast Software Encryption: Second International Workshop, B. Preneel, Ed. Leuven, Belgium: Springer-Verlag, **(1995)**, pp. 97-110.
[25] R. Rivest, “The RC5 encryption algorithm”, in Fast Software Encryption: Second International Workshop, B. Preneel, Ed. Leuven, Belgium: Springer-Verlag, **(1995)**, pp. 86-96.
[26] D.J. Wheeler and R.M. Needham, “TEA, a Tiny Encryption Algorithm”, in Proc. FSE, **(1994)**, pp. 363-366.
[27] M. Matsui, “New block encryption algorithm MISTY”, in Fast Software Encryption: 4th International Workshop, FSE'97, E. Biham, Ed. Haifa, Israel: Springer-Verlag, **(1997)**, pp. 53-67.
[28] G. Alvarez, D. de la Guia, F. Montoya and A. Peinado, “Akelarre: a new block cipher algorithm”, in SAC'96, Workshop Record, Queen's University, Kingston, Ontario, Canada, **(1996)**, pp. 1-14.
[29] E. Biham and R. Anderson, “Two practical and provably secure block ciphers: BEAR and LION”, in Fast Software Encryption, Dieter Gollmann, Ed. Cambridge, UK: Springer-Verlag, **(1996)**, pp. 99-111.
[30] C. Adams, “Constructing symmetric ciphers using the CAST design procedure”, Designs, Codes and Cryptography, vol. 12, no. 3, **(1997)**, pp. 283-316.
[31] V. Rijmen, J. Daemen, B. Preneel, A. Bossalaers and E.D. Win, “The cipher Shark”, in Fast Software Encryption, D. Gollman, Ed. Cambridge, UK: Springer-Verlag, **(1996)**, pp. 99-111.
[32] M. Kwan, “The design of the ICE encryption algorithm”, in Fast Software Encryption: 4th International Workshop,FSE'97, E. Biham, Ed. Haifa, Israel: Springer-Verlag, **(1997)**, pp. 69-82.
[33] J. Daemen, L.R. Knudsen and V. Rijman, “The block cipher Square”, in Fast Software Encryption, Eli Biham, Ed.: Springer Berlin Heidelberg, **(1997)**, pp. 149-165.
[34] D. M'Raihi, D. Naccache, J. Stern and S. Vaudenay, “XMX: A firmware-oriented block cipher based on modular multiplications”, in Fast Software Encryption, E. Biham, Ed. Haifa, Israel: Springer-Verlag, **(1997)**, pp. 166- 171.
[35] J. Daemen and V. Rijmen, “The Design of Rijndael: AES - The Advanced Encryption Standard”, **(2002)** , Springer.
[36] H. Heys, C. Adams, S. Tavares and M. Wiener, “CAST256: a submission for the Advanced Encryption Standard”, in First AES Candidate Conference (AES1), Ventura, California, USA, **(1998)**, Springer.
[37] S. Vaudenay and J. Stern, “CS-Cipher”, in Fast Software Encryption, S. Vaudenay, Ed. Paris, France: Springer-Verlag, **(1998)**, pp. 189-205.
[38] C. Lim, “Crypton: a new 128-bit block cipher”, in First AES Candidate Conference (AES1), Ventura, California, USA, **(1998)**.
[39] L. Knudsen, “DEAL: a 128-bit block cipher”, in First AES Candidate Conference (AES1), Ventura, California, USA, **(1998)**.
[40] M. Girault, P. Hoogvorst, F. Noilhan, T. Pornin,G. Poupard, J. Stern, S. Vaudenay and H. Gilbert, “Decorrelated fast cipher: an AES candidate”, in First AES Candidate Conference (AES1), California, USA, **(1998)**.
[41] M. Kanda, T. Matsumoto, S. Moriai, K. Ohta, M. Ookubo, Y. Takashima and H. Ueda K. Aoki, “E2
[42] D. Georgoudis, D. Leroux, and B. Chaves, “The “FROG” encryption algorithm”, in First AES Candidate Conference (AES1), Ventura, California, USA, **(1998)**.
[43] R. Schroeppel and H. Orman, “Overview of the hasty pudding cipher”, in First AES Candidate Conference (AES1), Ventura, California, USA, **(1998)**.
[44] J. Pieprzyk, J. Seberry and L. Brown, “Introducing the new LOKI97 block cipher”, in First AES Candidate Conference (AES1), Ventura, California,USA, **(1998)**.
[45] M. Jacobson and K. Huber, “The Magenta block cipher algorithm”, in First AES Candidate Conference (AES1), Ventura, California, USA, **(1998)**.

[46] D. Coppersmith, "Mars - a candidate cipher for AES", in First AES Candidate Conference (AES1), Ventura, California, USA, **(1998)**.
[47] M. Robshaw, R. Sidney, Y. Yin and R. Rivest, "The RC6 block cipher", in First AES Candidate Conference (AES1), Ventura, California, USA, **(1998)**.
[48] E. Biham, R. Anderson and L.R. Knudsen, "Serpent: a new block cipher proposal", in Fast Software Encryption, S. Vaudenay, Ed. Paris, France: Springer-Verlag, **(1998)**, pp. 222-238.
[49] National Institute of Standards and Technology, Skipjack and KEA algorithm specifications. http://csrc.nist.gov/CryptoToolkit/skipjack/skipjack.pdf. [Online], **(1998)**.
[50] B. Schneier, J. Kelsey, D. Whiting, D. Wagner and C. Hall, "Twofish: A 128-bit block cipher", in First AES Candidate Conference (AES1), California, USA, **(1998)**.
[51] "Exhaustive cryptanalysis of the NBS Data Encryption Standard", Computer, vol. 10, no. 6, **(1977)**, pp. 74-84,
[52] S. Vaudenay and H. Handschuh, "A universal encryption standard", in Selected Areas in Cryptography, H. Heys and C. Adams, Ed. Ontario, Canada: Springer-Verlag, **(2000)**, pp. 1-12.
[53] P. Barreto and V. Rijmen, "The Khazad legacy-level block cipher", in First Open NESSIE Workshop, Leuven, Belgium, **(2000)**.
[54] P. Barreto and V. Rijmen, "The Anubis block cipher", http://www.larc.usp.br/ pbarreto/AnubisPage.htm, **(2000)**.
[55] T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, T. Tokita and K. Aoki, "Camellia: a 128-bit block cipher suitable for multiple platforms - design and analysis", in Selected Areas in Cryptography, D. Stinson and S. Tavares, Ed. Ontario, Canada: Springer-Verlag, **(2001)**, pp. 39-56.
[56] P. Nguyen, F. Noilhan, S. Vaudenay and L. Granboulan, "DFCv2", in Selected Areas in Cryptography, D. Stinson and S. Tavares, Ed. Ontario, Canada: Springer-Verlag, **(2001)**, pp. 57-71.
[57] J. Borst, "The block cipher: GrandCru", in First Open NESSIE Workshop, Leuven, Belgium, **(2000)**.
[58] Toshiba Corporation, "Specification on a block cipher: Hierocrypt-L1", in First Open NESSIE Workshop, Leuven, Belgium, **(2000)**.
[59] ETSI/SAGE, "Kasumi Specification", Part of the Specification of the 3GPP Confidentiality and Integrity Algorithms. http://www.etsi.org, **(1999)**.
[60] Machado, "The Nimbus cipher: a proposal for NESSIE", in First Open NESSIE Workshop, Leuven, Belgium, **(2000)**.
[61] M. Peeters, G. Van Assche, V. Rijmen and J. Daemen, "The Noekeon block cipher," in First Open NESSIE Workshop, Leuven, Belgium, **(2000)**.
[62] Volchkov and A. Labedev, "NUSH", in First Open NESSIE Workshop, Leuven, Belgium, **(2000)**.
[63] L. McBride, "Q: a proposal for NESSIE", in First Open NESSIE Workshop, Leuven, Belgium, **(2000)**.
[64] H. Yanami, K. Yokoyama, M. Takenaka, K. Itoh, J. Yajima,N. Torii, H. Tanaka and T. Shimoyama, "Specification and supporting document of the block cipher SC2000", in First Open NESSIEWorkshop, Leuven, Belgium, **(2000)**.
[65] H. Handschuh and D. Naccache, "SHACAL", in First Open NESSIEWorkshop, Leuven, Belgium, **(2000)**.
[66] L.R. Knudsen, "PRESENT: An Ultra-Lightweight Block Cipher", in Cryptographic Hardware and Embedded Systems - CHES 2007.: Springer Berlin Heidelberg, **(2007)**, pp. 450-466.
[67] C. De Canniere, O. Dunkelman and M. Knežević, "KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers", in Cryptographic Hardware and Embedded Systems-CHES 2009, **(2009)**, pp. 272-288.
[68] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, "The LED block cipher", in Cryptographic Hardware and Embedded Systems–CHES 2011, **(2011)**, pp. 326-341
[69] D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers and R. Beaulieu, "The SIMON and SPECK lightweight block ciphers", in 52nd Annual Design Automation Conference (DAC '15). ACM, New York, NY, USA, **(2015)**, pp. 1-6.
[70] R. Basu, S. Ganguly, S. Maitra and G. Paul, "A Complete Characterization of the Evolution of RC4 Pseudo Random Generation Algorithm", Journal of Mathematical Cryptology, vol. 2, no. 3, **(2008)**, pp. 257–289.
[71] M. Madani and S. Chitroub, "Enhancement of A5/1 Stream Cipher Overcoming its Weakneses", In Proceedings of The Tenth International Conference on Wireless and Mobile Communications, **(2014)**.
[72] U. Blocher and M. Dichtl, "Fish: A fast software stream cipher", Fast Software encryption, LNCS, Springer-Verlag, vol. 809, **(1994)**, pp. 41-44.
[73] M. Pudovkina, "Analysis of chosen plaintext attacks on the WAKE stream cipher", Available at: http://citeseer.ist.psu.edu/452427. html, **(2001)**.
[74] "On Fibonacci Keystream Generators≒, RJ Anderson, in Fast Software Encryption, Springer LNCS, vol 1008, **(1994)**, pp. 346–352.
[75] R. J. Jenkins Jr., ISAAC, Fast Software Encryption, **(1996)**, pp. 41–49.
[76] "Software-efficient pseudorandom function and the use thereof for encryption", Patent no: US 5454039 A.
[77] J. Daemen and C. Clapp, "Fast hashing and stream Encryption with PANAMA", Fast Software Encryption, LNCS 1372, S. Vaudenay, Ed., Springer-Verlag, **(1998)**, pp. 60-74.

[78] D. Watanabe, S. Furuya, H. Yoshida and B. Preneel, "A new keystream generator MUGI", In Fast Software Encryption (FSE) 2002, Lecture Notes in Computer Science, Springer, vol. 2365, **(2002)**, 179-194.
[79] O. Y. Shaked, "Cryptanalysis of the Bluetooth E0 cipher", citeseer. ist.psu.edu/744254.html.
[80] D. Coppersmith, S. Halevi and C. Jutla, "Scream: a software-efficient stream cipher", In Fast Software Encryption (FSE) 2002, Lecture Notes in Computer Science, Springer, vol.2365, **(2002)**, pp. 195-209.
[81] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen and O. Scavenius, "Rabbit: A new high-performance stream cipher", In FSE, **(2003)**.
[82] P. Ekdahl and T. Johansson, "SNOW - a new stream cipher", Proceedings of first NESSIE Workshop, Heverlee, Belgium, **(2000)**.
[83] P. Hawkes and G. Rose, "Primitive Specification for SOBER-128", IACR ePrint Archive, http://eprint.iacr.org/2003/81/, **(2003)**.
[84] G. G. Rose and P. Hawkes, "Turing: a Fast Stream Cipher", FSE 2003, LNCS, Springer.
[85] C. De Canni_ere, B. Preneel. Trivium, M.J.B. Robshaw and O. Billet, "New Stream Cipher Designs - The eSTREAM Finalists", LNCS, Springer, vol. 4986, **(2008)**, pp. 244-266.
[86] C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin and H. Sibert, "Sosemanuk, a Fast Software-Oriented Stream Cipher", In New Stream Cipher Designs, volume 4986 of Lecture Notes in Computer Science, Springer-Verlag, **(2008)**, pp.98-118.
[87] D. J. Bernstein, "The Salsa20 Family of Stream Ciphers", In New Stream Cipher Designs, Lecture Notes in Computer Science, Springer-Verlag, vol. 4986, **(2008)**, pp. 84-97.
[88] E. Biham and J. Seberry, "\Py(Roo): a fast and secure stream cipher using rolling arrays", eSTREAM, http://www.ecrypt.eu.org/stream/papersdir/081.pdf.
[89] D. Whiting, B. Schneier, S. Lucks and F. Muller, "Phelix - Fast Encryption and Authentication in a Single Cryptographic Primitive", Report 2005/020, eSTREAM - ECRYPT - Stream Cipher Project, http://www.ecrypt.eu.org/stream, **(2005)**.
[90] H. Wu, "A New Stream Cipher HC-256", in Fast Software Encryption (FSE'04), LNCS 3017, The full version is available at http://eprint.iacr.org/2004/092.pdf, pp. 226-244.
[91] M. Hell, T. Johansson, W. Meier, "Grain — A Stream Cipher for Constrained Environments", eSTREAM submission.
[92] M. Matsumoto, M. Saito, T. Nishimura and M. Hagita, "CryptMT version 2. 0: A Large State Generator with Faster Initialization", eSTREAM report 2006/023 (2006). URL: http://www.ecrypt.eu.org/stream/papers. html.
[93] . O'Neil, B. Gittins and H. Landman, "VEST – Hardware Dedicated Stream Ciphers", Report, 2005/032, eSTREAM - ECRYPT - Stream Cipher Project, http://www.ecrypt.eu.org/stream, **(2005)**.
[94] B. Gammel, R. G¨ottfert and O. Kniffler, "Achterbahn-128/80: Design and analysis", in SASC'2007: Workshop Record of The State of the Art of Stream Ciphers, **(2007)**, pp. 152–165.
[95] C. Berbain, H. Gilbert and J. Patarin, "QUAD: A Practical Stream Cipher with Provable Security", In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS. Springer, Heidelberg, vol. 4004, **(2006)**, pp. 109–128.
[96] Y. Nawaz and G. Gong, "WG: A Family of Stream Ciphers with Designed Randomness Properties", Information Sciences, vol.178, no.7, **(2008)**, pp. 1903-1916.
[97] C. Cid, S. Kiyomoto and J. Kurihara, "The rakaposhi stream cipher", in Proceedings of the 11th International Conference on Information and Communications Security, ICICS'09, Berlin, Heidelberg, 2009, Springer-Verlag, pp. 32-46.
[98] X.-T. Feng, "ZUC Algorithm: 3GPP LTE, International Encryption Standard", Information Security and Communications Privacy, **(2011)**.
[99] S. Babbage and M. Dodd, "The stream cipher MICKEY-128 2.0. eSTREAM", ECRYPT Stream Cipher Project, **(2006)**.
[100] R.L. Rivest and J.C.N. Schuldt, "Spritz—A spongy RC4-like stream cipher and hash function", In Proceedings of the Charles River Crypto Day, Palo Alto, CA, USA, **(2014)**.
[101] E. Dubrova and M. Hell, "Espresso: A Stream Cipher for 5G Wireless Communication Systems", Cryptology ePrint Archive, **(2015)**.
[102] E.K. Grossman and B. Tuckerman, "Analysis of a Feistel-like cipher weakened by having no rotating key", IBM Thomas J. Watson Research Report RC 6375, **(1977)**.
[103] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", Advances in Cryptology — CRYPTO '90. Springer-Verlag, 2–21, **(1990)**.
[104] M. Matsui and A. Yamagishi, "A new method for known plaintext attack of FEAL cipher", Advances in Cryptology - EUROCRYPT, **(1992)**.
[105] D. Davies, "Sean Murphy", Pairs And Triplets Of DES S-Boxes. Journal of Cryptology, ISSN 0933-2790, vol. 8, no. 1, **(1993)**, pp. 1–25
[106] E. Biham and A. Biryukov, "An Improvement of Davies' Attack on DES (gzipped PostScript)", Advances in Cryptology — Eurocrypt '94. Perugia: Springer-Verlag, **(1994)**, pp. 461–467.
[107] R. Lipton and J. F. Naughton, "Clocked adversaries for hashing', Algorithmica, vol. 9, no. 3, **(1993)**, pp. 239–252.

[108], E. Biham, "New types of cryptanalytic attacks using related keys", Journal of Cryptology, vol. 7, no. 4, **(1994)**, pp. 229-246
[109] C. Harpes, G. G. Kramer *and* J.L. Massey, *"*A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma", Advances in Cryptology — Eurocrypt '95, Saint-Malo*:* Springer-Verlag, **(1995*)***, pp. 24–38.
[110] C. Harpes and J. Massey, "Partitioning Cryptanalysis*",* 4th International Workshop in Fast Software Encryption (FSE '97). Haifa*:* Springer-Verlag, **(1997)**, pp. 13–27.
[111] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems", Advances in Cryptology—CRYPTO'96. Lecture Notes in Computer Science, vol. 1109, **(1996)**, pp. 104–113.
[112] J. Daemen, L. Knudsen and V. Rijmen, "The Block Cipher Square (PDF)", 4th International Workshop on Fast Software Encryption (FSE '97), Lecture Notes in Computer Science. Haifa: Springer-Verlag. pp. 149–165. Retrieved 2007-02-15, Vol. 1267, **(1997)**.
[113] T. Jakobsen *and* L. Knudsen, *"*The Interpolation Attack on Block Ciphers*",* 4th International Workshop on Fast Software Encryption (FSE '97), LNCS 1267. Haifa*:* Springer-Verlag, **(1997)**, pp. 28–40.
[114] D. Wagner, "The Boomerang Attack", 6th International Workshop on Fast Software Encryption (FSE '99), Rome: Springer-Verlag, **(1999)**, pp. 156–170.
[115] J. Kelsey, B. Schneier and D. Wagner, "Mod n Cryptanalysis, with Applications against RC5P and M6", (PDF/PostScript). Fast Software Encryption, Sixth International Workshop Proceedings. Rome: Springer-Verlag, **(1999)**, pp. 139–155.
[116] J. Kelsey, T. Kohno and B. Schneier, "Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent", Seventh Fast Software Encryption Workshop, Springer-Verlag, **(2001)**, pp. 75-93.
[117] E. Biham, O. Dunkelman and N. Keller, "The Rectangle Attack – Rectangling the Serpent. Advances in Cryptology", Proceedings of EUROCRYPT 2001, Innsbruck: Springer-Verlag, **(2001)**, pp. 340–357.
[118] N. Courtois and J. Pieprzyk Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. LNCS 250, vol. 1, **(2002)**, pp. 267–287.
[119] D. Khovratovich and I. Nikolić, "Rotational Cryptanalysis of ARX", (PDF). University of Luxembourg, **(2010)**.
[120] B. Schneier, "Schneier on Security: New Attack on Threefish", **(2010)**.

