CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter describes a literature review of the research scope that covers the field of cryptography which focuses on the lightweight block cipher. Cryptographic projects on lightweight block cipher were observed in order to identify the security evaluation criteria. The construction of existing algorithms is studied for benchmarking purposes in terms of security and performance. Finally, the theoretical definitions of cryptanalysis tests to evaluate cryptographic algorithms are discussed.

2.2 Cryptography

Cryptology is a science of the design, processing, and study of cryptographic algorithms to provide information security in communication. In general, the cryptology concept is inclusive of cryptography and cryptanalysis as shown in Figure 2.1. Designing secure algorithms to keep messages secret is studied under cryptography while analyzing the security of algorithms falls under cryptanalysis.

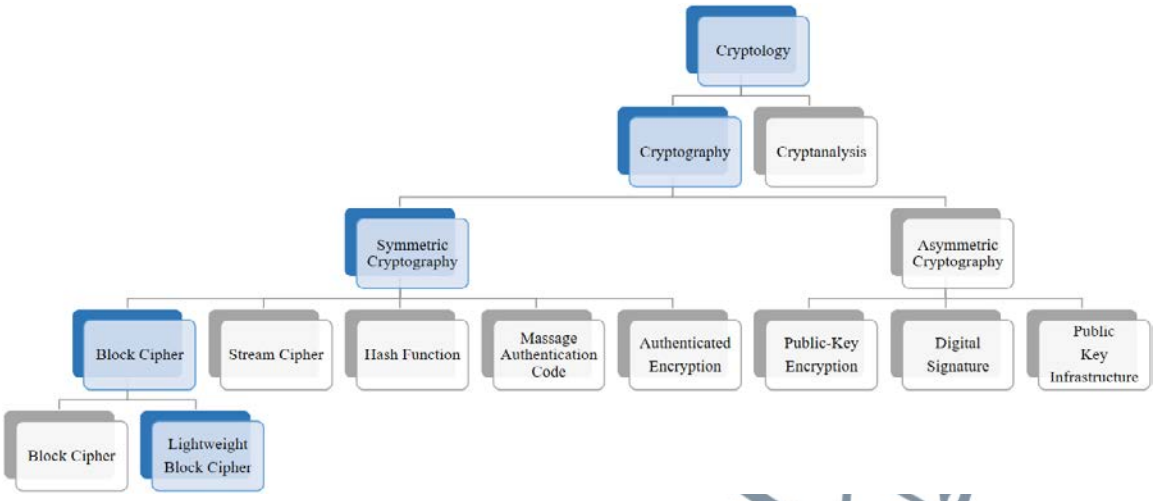

Figure 2.1: Overview of Cryptology

Cryptography has a long history and is evolving to this day. In terms of applications and academic research topics, cryptography is considered a massive research field. Without realizing it, cryptographic tools and devices are brought into every single day of our lives. Cryptographic algorithms are used in many areas from a simple website to security devices to protect information.

In a standard communication, there exist two parties, namely the sender and receiver. The sender applies cryptographic algorithms to make a message unavailable to third parties (Verma et al., 2019). Plaintext P is the message, while ciphertext C is the unreadable message. The secret key K is used in the encryption and decryption process. The transformation of P into C is called encryption. An encryption algorithm E_K contains the master key K is described as $E_K(P) = C$. On the other hand, the transformation of C into P is called decryption. E_K^{-1} denotes the decryption algorithm described as $E_K^{-1}(C) = P$.

Modern cryptography includes symmetric and asymmetric cryptography. Symmetric cryptography which is also known as secret-key cryptography makes use of the same key to encrypt and decrypt information as shown in Figure 2.2 (Pritchard et al., 2018). There are five categories that fall under this type of cryptography including block ciphers, stream ciphers, authenticated ciphers, hash functions, and message authentication codes.

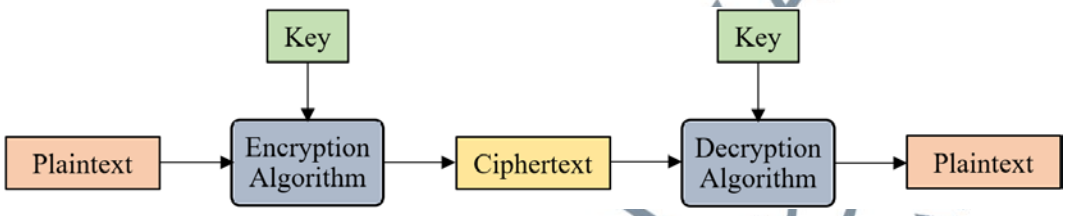

Figure 2.2: Symmetric Cryptography

On the other hand, asymmetric cryptography (Krishna et al., 2018) or known as public-key cryptography is a study of cryptographic systems that involve a public key and a private key as shown in Figure 2.3. A pair of mathematically-related keys are required to be generated such that it is computationally infeasible in order to determine the private key from the public key. The private key must be kept secret, meanwhile, the public key can be distributed to other parties. There are three well-known applications of asymmetric cryptography that include public-key encryption, public-key infrastructure (PKI), and digital signatures as described in Table 2.1.

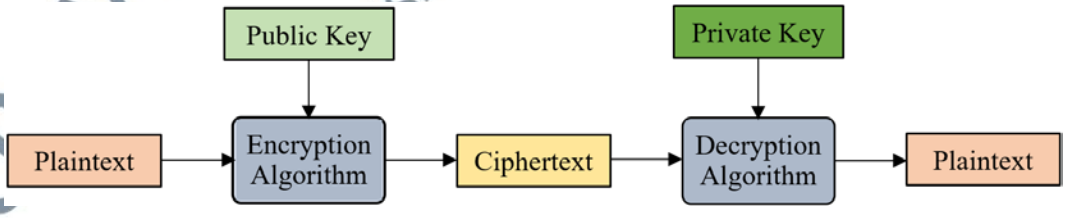

Figure 2.3: Asymmetric Cryptography

Table 2.1: Classification of Cryptography

Category	Primitive	Method
Symmetric Cryptography	Block Cipher	Operates in a plaintext block of bits or bytes to generate the corresponding ciphertext block (e.g., AES, Camellia, and CLEFIA).
	Stream Cipher	Encrypts each plaintext bit using a keystream to generate the ciphertext (e.g., ChaCha20, KCipher, and Rabbit).
	Hash Function	Transforms an arbitrary length of data into a fixed-length digest (e.g., SHA2, SHA3, & SHAKE).
	Message Authentication Code	Takes a key and an arbitrary plaintext to generate a fixed-length message authentication tag.
	Authenticated Encryption	Combines confidentiality mode with an authentication mode to simultaneously provide confidentiality, integrity, and authenticity.
Asymmetric Cryptography	Public-Key Encryption	Uses the recipient's public key to encrypt the plaintext and his private key to decrypt the ciphertext (e.g., RSA and NTRU).
	Digital Signature	Encrypts a plaintext using the sender's private key to generate a signature (e.g., DSA and ECDSA).
	Public Key Infrastructure	Certifies the ownership of a key pair by a trusted third party through a public-key certificate.

Both symmetric and asymmetric cryptography have their respective advantages. Asymmetric cryptography is used to establish a secure communication channel and provides non-repudiation using public-key cryptography. On the other hand, symmetric cryptography is preferred for confidentiality as it is faster than asymmetric cryptography (D'souza & Panchal, 2018). In addition, symmetric cryptography uses smaller keys than asymmetric cryptography. Since this research is focusing on solving the security issue in mobile applications, block cipher is the most suitable symmetric cryptography primitive to be focused on.

2.3 Lightweight Block Cipher

This section highlights the requirements of lightweight block ciphers to answer *Research Question 1*. As an introduction, the block cipher is an important primitive in secret-key cryptography to provide confidentiality for data transmission in an insecure environment. Block cipher is also used in information security, computer security, network security, and other security applications. A block cipher is an algorithm that transforms a fixed-length data block which is called a plaintext block, into another data block called a ciphertext block that is controlled by a secret key.

There are two variants of block cipher namely the conventional block cipher and lightweight block cipher with their design requirements as shown in Table 2.2. The main difference between the variants is the size of the cipher block, where the lightweight block cipher should have a smaller block size than the conventional block cipher (Cazorla et al., 2013). For the key size, it is recommended for a lightweight block cipher to have a smaller key size than the conventional block cipher. The lightweight block cipher relies on simple operations such as binary XOR and AND that lead to the increasing number of cipher rounds to achieve the required security strength. Although conventional block cipher has a lower number of rounds, its complex operations consume high memory requirements.

Table 2.2: Comparison of Block Cipher Requirements

Criteria	Block Cipher	Lightweight Block Cipher
Block Size	Bigger block size	Smaller block size
Key Size	Bigger key size	Smaller key size
No. of Rounds	More number of rounds	Less number of rounds
Encryption	Complex operations	Simple operations
Decryption	Complex operations	Simple operations
Key Schedule	Complex operations	Simple operations

The emergence of embedded systems has increased the need for new cryptographic instruments. Hence, the subclasses of cryptography become diversified and one of them is lightweight cryptography. Lightweight cryptography focuses on optimizing the trade-off between security and efficiency (Pawar & Pattanshetti, 2018). Lightweight cryptography research has emerged to address the constraints and limitations of conventional cryptography (Li et al., 2019). Therefore, lightweight block cipher has become the main scope of this research.

2.3.1 Lightweight Block Cipher Projects

This section highlights the importance of lightweight block ciphers to answer *Research Question 2*. The employment of mobile phone devices has increased rapidly. Thus, information security awareness among users and developers makes cryptography a crucial component to be used in those devices. However, conventional cryptography does not fit into the limited resources of constrained environments. Lightweight cryptography is needed to overcome this situation. Consequently, international standardization bodies such as the International Organization for Standards (ISO) and the International Electrotechnical Commission (IEC) have issued lightweight cryptography standards as documented in the ISO/IEC 29192-2:2019 Information security - Lightweight cryptography - Part 2: Block Ciphers (Malutan et al., 2019). In the past years, several lightweight block cipher projects have been conducted to develop guidelines, recommendations, and standards for lightweight algorithms, which detail their design, security, and performance criteria. The following subsections discuss the existing cryptography projects for the lightweight block cipher.

2.3.1.1 NESSIE Project

New European Schemes for Signatures, Integrity, and Encryption (NESSIE) is a European project developed between 2000 and 2003 to search for cryptographic primitives including block cipher, stream cipher, public-key encryption, message authentication code algorithm, hash function, identification schemes, and digital signature scheme (Preneel, 2002). The list of successful algorithms was not adopted by any European government or commission but was recommended to the relevant standardization bodies within Europe. NESSIE project did not specify the lightweight block cipher. However, submitters were encouraged to propose algorithms with a smaller block size of 64 bits which might be crucial in the next 10 to 15 years after the project started.

2.3.1.2 NIST Lightweight Cryptography Project

The National Institute of Standards and Technology (NIST) has been exploring the need for lightweight algorithms in constrained environments. Thereby, the NIST initiated the lightweight cryptography project in 2015 where the algorithm proposals submission ended in April, and the first workshop was held in July of the same year (Turan et al., 2021). The final selection of the successful proposals was completed in 2021. The NIST project is in search of cryptographic primitives and modes including block cipher, stream cipher, authenticated encryption scheme, cryptographic permutation, message authentication code, and hash function.

2.3.1.3 MySEAL Project

National Trusted Cryptographic Algorithm List (MySEAL) is a project designed to build a portfolio for the Malaysian national trusted cryptographic algorithms (CSM, 2021). This project focused on providing a list of cryptographic algorithms to be implemented by Malaysians in their security products to support the National Cryptography Policy (NCP). While NCP plays the role of a guideline to achieve national cryptographic sovereignty, MySEAL becomes the reference in the field of cryptography and cryptanalysis.

For a lightweight block cipher to be listed in MySEAL, the algorithm has to follow certain evaluation criteria. The criteria were established based on international standards (i.e., NIST-FIPS, ISO/IEC, and IEEE) and projects (i.e., CRYPTREC, NESSIE, and eSTREAM). There are two types of cryptographic algorithms lists namely AKSA (Existing Cryptographic Algorithm) and AKBA (New Cryptographic Algorithm). Candidate algorithms were filtered and extensively assessed based on the evaluation criteria before being announced as the selected algorithm.

2.3.1.4 Security Evaluation Criteria

The aim of the research is to develop a new lightweight block cipher, where meeting the security evaluation criteria is the key requirement. Referring to the criteria from the NESSIE (Preneel, 2002), NIST (Turan et al., 2021), and MySEAL (CSM, 2021) projects in previous sections, a comparison of the security evaluation has been carried out as shown in Table 2.3 to differentiate each of their requirements. The cryptographic projects defined three components of the security criteria that must be complied with that include key size, block size, and analysis.

Table 2.3: Comparison of Security Evaluation Criteria of Cryptographic Projects

Criteria		Requirement		
		NESSIE	**NIST**	**MySEAL**
Security	Key Size	At least 128 bits	At least 112 bits	At least 80 bits
	Block Size	64 bits	64 bits	At least 64 bits
	Analysis	Generic attacks	Fault attacks	Linear cryptanalysis
		Side-channel attacks	Side-channel attacks	Differential cryptanalysis
			Related-key attacks	NIST statistical tests

For the evaluation process in this research work, the security evaluation criteria are identified based on the generalization of criteria implemented by each cryptographic project whose members are comprised of expert panelists in the field of cryptography from around the world. The target of this research work is to develop an algorithm that meets the requirements of any cryptographic project. By meeting the criteria set by the projects, the new algorithm should be eligible for any project submission in the future. Therefore, the security evaluation criteria to be implemented in the research are presented in Table 2.4.

Table 2.4: Security Evaluation Criteria

Criteria	Requirement
Key Size	At least 128 bits
Block Size	64 bits
Security Analysis	The cryptanalysis tests must include at least two reports from the list: i) Linear cryptanalysis ii) Differential cryptanalysis iii) NIST statistical tests iv) Side-channel attacks v) Fault attacks vi) Related-key attacks vii) Generic attacks

2.3.2 Previous Work on Lightweight Block Cipher

Lightweight cryptography algorithm is developed based on the design structure such as Substitution-Permutation Network (SPN) (Bogdanov et al., 2007), Feistel Network (FN) (Schneier, 1993), Generalized Feistel Network (GFN) (Suzaki et al., 2011), Nonlinear Feedback Shift Register (NLFSR) (De Cannière et al., 2009), Addition, Rotation, and XOR (ARX) (Lai & Massey, 1991), and Hybrid (Engels et al., 2010). The algorithm design structure along with the block size, key size, and round number would influence the performance of the lightweight block cipher.

Hardware and software efficiency is becoming critical in highly constrained environments, thus highlighting lightweight cryptography as important ongoing research. The goal of hardware design is to reduce the number of required logic gates. Meanwhile, software design cipher considers the standard microprocessor implementation platform, intending to save extra storage space, minimize computation, and optimize performance.

The efficient implementation of lightweight cryptography in electronic devices such as mobile phones is challenging since the performance is influenced by other metrics such as memory footprint and code size. Moreover, lightweight cryptography should withstand any form of cryptographic attack to ensure data protection in the devices.

For comparison and understanding of the construction of lightweight block cipher, several well-known cryptographic algorithms are discussed in the following subsections. The algorithms are HIGHT, PRESENT, and MISTY which are listed in several lightweight block cipher projects including NESSIE and MySEAL. A high-performance algorithm namely RECTANGLE is also discussed to highlight its design structure.

2.3.2.1 MISTY

MISTY is a lightweight block cipher with a 64 bits block size and a 128-bit key (Matsui, 1997). The algorithm structure followed the Feistel network with a variable number of rounds in a multiple of four, with 8-round encryption is recommended. MISTY is listed in the NESSIE project and the cipher functions are presented in Table 2.5. MISTY is applicable for various systems such as identity cards and ATM networks. One of the strengths of the algorithm is its security against differential and linear cryptanalysis.

Table 2.5: MISTY Algorithm

Encryption Algorithm	Key Schedule Algorithm
FL Function: The cipher state is split into halves. Both halves need to go through AND/OR and XOR operations separately before being combined.	FI Function: The key state is split into halves. Both halves need to go through S-box, append, and XOR operations separately before being combined.
FI Function: The cipher state is split into halves. Both halves need to go through S-box, append, and XOR operations separately before being combined.	
FO Function: The cipher state is split into halves. The first half of the cipher state is XOR with FI function output. The resulting XOR output is XOR with the second half of the cipher state before being combined.	

2.3.2.2 HIGHT

HIGHT lightweight block cipher applied 32 rounds of Feistel network structure for encryption. The cipher consists of 64 bits block size and 128 bits key size (Hong et al., 2006). The algorithm provided a low-resource implementation in computing devices such as sensors and RFID tags. HIGHT is listed in the MySEAL project for its performance and efficiency in hardware implementation. The encryption and key schedule algorithms are described in Table 2.6.

Table 2.6: HIGHT Algorithm

Encryption Algorithm	Key Schedule Algorithm
Initial Transformation: The cipher state needs to go through XOR and addition modulo operations with the whitening key.	Whitening Key Generation: Two whitening keys are generated by choosing eight bytes from the master key to be used in the initial and final transformation.
Round Function: The cipher state needs to go through byte rotation, XOR, and addition modulo operations with the round keys.	Round Key Generation: 128 round keys are generated through the constant generation function produced by LFSR and XOR operations.
Final Transformation: The cipher state needs to go through XOR and addition modulo operations with the whitening key.	

2.3.2.3 PRESENT

PRESENT is a 31 rounds substitution-permutation network lightweight block cipher. The block size is 64 bits which supports key sizes of 80 and 128 bits (Bogdanov et al., 2007). PRESENT was designed for low-power consumption and high chip efficiency due to its simplicity as shown in Table 2.7. The algorithm is known for its various applications on IoT devices. PRESENT algorithm is listed in the MySEAL project and become the benchmark for evaluation criteria for future lightweight block cipher submissions.

Table 2.7: PRESENT Algorithm

Encryption Algorithm	Key Schedule Algorithm
addRoundKey: The cipher state is XOR with the round key.	The key state needs to go through rotation, S-box, and XOR operations.
sBoxLayer: The cipher state is substituted with the S-box.	
pLayer: The cipher state is rearranged with the permutation table.	

2.3.2.4 RECTANGLE

RECTANGLE is a 64 bits lightweight block cipher and accepts 80 or 128 bits keys (Zhang et al., 2015). The algorithm implemented the substitution-permutation network structure with 25 encryption rounds. The components of RECTANGLE design are described in Table 2.8. RECTANGLE enabled lightweight and fast implementations using its bit-slice method. Hence, the algorithm provided excellent performance in software and hardware environments which offers sufficient flexibility for multiple application platforms such as the FPGA (Field Programmable Gate Arrays) semiconductor devices. Therefore, RECTANGLE design is used as the reference for the construction of the newly developed lightweight block cipher.

Table 2.8: RECTANGLE Algorithm

Encryption Algorithm	Key Schedule Algorithm
Add Round Key: The cipher state is XOR with the round key.	Sub Column: The key state is substituted with the S-box.
Sub Column: The cipher state is substituted with the S-box.	Feistel Transformation: The key state is shifted to the left, XOR, and rearranged.
Shift Row: The cipher state is shifted to the left and rotated.	Round Constants: The key state is XOR with the generated round constants.

2.3.2.5 Performance of Lightweight Block Ciphers

Comparisons of the performance of lightweight block ciphers are presented in Table 2.9 and Table 2.10. Existing high-performance algorithms such as RECTANGLE (Zhang et al., 2015), SIMON (Beaulieu et al., 2013), LED (Guo et al., 2011), Piccolo (Shibutani et al., 2011), PRESENT (Bogdanov et al., 2007), and LBlock (Wu & Zhang, 2011) are included in this comparison to observe their performance. The throughput is calculated in bits per second or number of cycles per byte as displayed in Table 2.9 to show the efficiency of throughput with the utilization of block cipher in limited-resource hardware. Overall, the result illustrates that RECTANGLE achieved the highest throughput among the compared algorithms.

Table 2.9: Throughput Comparison

Reference	(Zhang et al., 2015)	(Li et al., 2016)	(Hatzivasilis et al., 2018)	(Thakor et al., 2021)
	Throughput (Kb/s)	**Throughput (Kb/s)**	**Throughput (Kb/s)**	**Throughput (Kb/s)**
RECTANGLE	246.00	246.00	246.00	246.00
SIMON	-	-	22.90	15.80
LED	5.10	3.40	133.33	5.10
Piccolo	237.00	237.04	193.94	237.04
PRESENT	200.00	200.00	206	200.00
LBlock	-	200.00	-	200.00

On the other hand, Table 2.10 compares the speed of existing block ciphers in terms of their execution time, cycles, and latency. Overall, RECTANGLE achieved a very competitive software speed among the existing block ciphers. RECTANGLE is constructed based on a simple design, thus the bit-sliced method allows for both low-cost hardware and efficient software implementations. Therefore, this research has set RECTANGLE as the benchmark in the process of developing a new lightweight block cipher.

Table 2.10: Speed Comparison

Reference	(Pehlivanoğlu et al., 2017)	(Omrani et al., 2018)	(Omrani et al., 2018)	(Zhang et al., 2015)
	Time (Cycles)	**Time (ms)**	**Clock Cycle (ms)**	**Cycles per Block**
RECTANGLE	36,121	0.296	4,736	26
SIMON	-	0.720	11,520	-
LED	594,453	-	-	1,248
Piccolo	294,478	-	-	27
PRESENT	274,463	65.276	1,044,416	32
LBlock	-	0.320	5,120	-

2.4 Cryptanalysis

Lightweight block cipher evaluation criteria are divided into three major categories which are cryptanalysis and implementation, as well as cost and performance as shown in Figure 2.4 (Hatzivasilis et al., 2018). Implementation criteria observe the flexibility, suitability, and design simplicity of the cipher. On the other hand, cost and performance criteria evaluate energy consumption, computational efficiency, and memory requirements. In lightweight block cipher, the implementation, cost and performance categories are focused on the implementation of the algorithm in hardware devices. However, due to constraints and limitations of this research work, performance tests are conducted on software applications which include the execution speed and throughput evaluations in order to measure the efficiency of the lightweight block cipher.

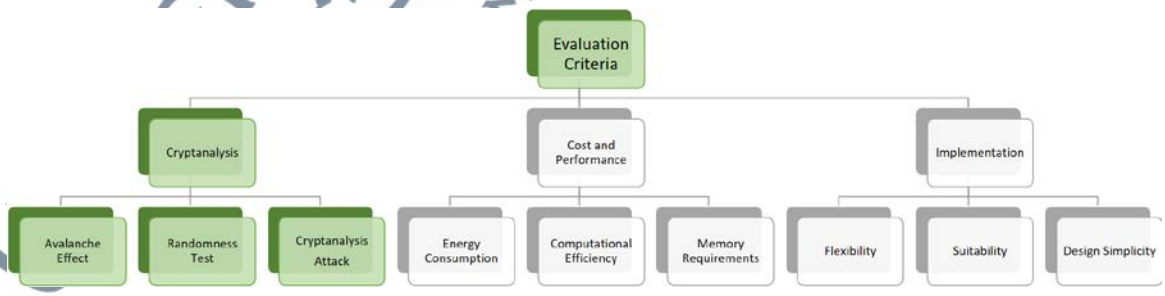

Figure 2.4: Lightweight Block Cipher Evaluation Criteria

Out of the three categories, cryptanalysis is the most important criterion of the lightweight block cipher. Therefore, this research focused on cryptanalysis by implementing the security evaluation criteria as mentioned in Section 2.3.1.4. In addition to the security evaluation criteria, this research also included the avalanche effect analysis that consists of the correlation coefficient, bit error rate, and key sensitivity tests to further evaluate the new algorithm that is discussed in the following subsections.

2.4.1 Avalanche Effect

Strict avalanche criterion (SAC) is a method in which the robustness of cryptographic components is measured (Javeed et al., 2020). SAC observes changes to the output caused by a modification in the input. Ideally, a change in 1-bit input would affect half of the output bits. The avalanche effect can be measured by dividing the number of flipped ciphertext bits by the total ciphertext bits.

A secure encryption algorithm must be able to create a complex relationship between the key, plaintext, and ciphertext, also known as confusion. Also, the algorithm must be able to scatter the changes made on the plaintext over the whole ciphertext, known as diffusion properties (Shannon, 1949). Confusion is accomplished by applying substitution operations (S-boxes) and the use of permutations enables diffusion. Confusion and diffusion create a degree of randomness in the cryptographic algorithm so that no ciphertext pattern is recognizable.

Therefore, to evaluate the avalanche effect property of lightweight block cipher in this research, three experiments are conducted including correlation coefficient, bit error rate, and key sensitivity tests. Correlation coefficient can determine the confusion effect of the block cipher (Sallam et al., 2017). Bit error rate measures the differences of ciphertext caused by a change in the plaintext (Salam et al., 2019). Meanwhile, key sensitivity test observes the ciphertext affected by a secret key modification (Jallouli et al., 2016).

2.4.2 Randomness Tests

There exist many approaches to evaluate cryptographic algorithms, and among these, randomness testing has been identified to be very important. When designing a cryptographic algorithm, it is vital to make sure that the ciphertext produced from this algorithm must be random. In order to determine whether the tested algorithm fulfils this requirement, randomness testing using statistical analysis can be applied.

There are variants of statistical analysis packages that can be used to evaluate the randomness of an algorithm. Table 2.11 lists all 15 tests included in the NIST statistical test (Rukhin et al., 2010), Donald Knuth's statistical test (Knuth, 1998), DIEHARD statistical test (Marsaglia, 1995), and Crypt-XS statistical test (Caelli et al., 1992) packages. The NIST Statistical Test Suite is the most popular statistical analysis application that has been implemented by researchers. On top of that, the test suite was used to validate the candidates for the NIST AES competition (Do Nascimento & Xexeo, 2017) and also being implemented by the MySEAL project to evaluate the security of algorithms. Therefore, the NIST Statistical Test Suite application is used in this research.

Table 2.11: Statistical Tests Application Packages

No.	NIST	DieHard	Donald Knuth	Crypt-XS
1.	Runs	Runs	Run	Runs
2.	Serial	Craps	Gap	Frequency
3.	Frequency	Squeeze	Serial	Change Point
4.	Spectral DFT	Parking Lot	Poker	Binary Derivative
5.	Block Frequency	3D Spheres	Collision	Linear Complexity
6.	Cumulative Sums	Birthday Spacing	Frequency	Sequence Complexity
7.	Linear Complexity	Overlapping Sums	Permutation	
8.	Random Excursion	Minimum Distance	Maximum-of-t	
9.	Maurer's Universal	Ranks of 6x8 Matrices	Serial Correlation	
10.	Binary Matrix Rank	Overlapping Permutations	Birthday Spacings	
11.	Approximate Entropy	Count the 1's in Specific Bytes	Coupon Collector's	
12.	Longest Runs of Ones	Monkey Tests on 20-Bit Words		
13.	Overlapping Templates	Monkey Tests OPSO, OQSO, DNA		
14.	Random Excursion Variant	Count the 1's in a Stream of Bytes		
15.	Non-Overlapping Templates	Ranks of 31x31 and 32x32 Matrices		

The NIST Statistical Test Suite statistical package focuses on identifying various characteristics of non-randomness that may occur in the ciphertext. Table 2.12 specifies the objective of each statistical test in the NIST Statistical Test Suite (Rukhin et al., 2010). For a cryptographic algorithm to be considered random, it must pass all of the applicable tests.

Table 2.12: Objectives of NIST Statistical Tests

Statistical Test	Objective
Runs	To determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence.
Serial	To determine whether the number of occurrences of the 2^m m-bit overlapping patterns is approximately the same as expected for a random sequence.
Frequency	To determine whether the number of ones and zeros in a sequence is approximately the same as would be expected for a random sequence.
Spectral DFT	To determine whether the spectral frequency of the binary sequence agrees with what would be expected for a random sequence.
Block Frequency	To determine whether the frequency of m-bit blocks in a sequence appears as often as would be expected for a random sequence.
Cumulative Sums	To determine whether the maximum of the cumulative sums in a sequence is too large or too small; indicative of too many ones or zeros in the early or late stages.
Linear Complexity	To determine whether or not the sequence is complex enough to be considered random.
Random Excursion	To examine the number of cycles within a sequence and determine whether the number of visits to a given state, between -4 to 4, exceeds the expected for a random sequence.
Maurer's Universal	To determine whether a binary sequence does not compress beyond what is expected of a random sequence.
Binary Matrix Rank	To determine whether the distribution of the rank of 32 x 32 bits matrices agrees with the theoretical probabilities.
Approximate Entropy	To determine whether a sequence appears more regular than is expected from a random sequence.
Longest Runs of Ones	To determine whether the distribution of long runs of ones agrees with the theoretical probabilities.
Overlapping Templates	To determine whether the number of occurrences for a template of all ones agrees with what is expected for a random sequence.
Random Excursion Variant	To determine if the total number of visits to states, between -9 to 9 exceeds the expected for a random sequence.
Non-Overlapping Templates	To determine whether the number of occurrences for a specified non-periodic template agrees with the number expected for a random sequence.

Each statistical test requires a different number of bits to meet its purpose of testing. Table 2.13 summarizes the NIST input bits recommendation for the complete 15 statistical tests (Rukhin et al., 2010). There could be limitations in the derivation of output bits generated by some cryptographic algorithms, therefore few data categories might not be able to complete all tests (Isa & Z'aba, 2012). The NIST recommendation is referred to determine the availability of the statistical tests.

Table 2.13: Bits Requirements for NIST Statistical Tests

Statistical Test	Required No. of Bits
Runs	$n \geq 100$
Frequency	
Block Frequency	
Cumulative Sums	
Longest Runs of Ones	$n \geq 128$
Spectral DFT	$n \geq 1{,}000$
Binary Matrix Rank	$n \geq 38{,}912$
Maurer's Universal	$n \geq 387{,}480$
Linear Complexity	$n \geq 1{,}000{,}000$
Random Excursion	
Overlapping Templates	
Random Excursion Variant	
Serial	Not specified
Approximate Entropy	
Non-Overlapping Templates	

By default, the analysis used a significance level, $\alpha = 0.01$, as implemented in the NIST Statistical Test Suite that produced *p*-values obtained from the statistical tests. However, the significance level can be changed between 0.001 (0.1%) and 0.01 (1%) depending on the number of samples to be tested. Each *p*-value is compared to the significance level to determine the randomness of a ciphertext sequence. The indicators of the statistical test results are listed in Table 2.14.

Table 2.14: Statistical Test Results Indication

Condition	Result
p-value = 1	Completely random
p-value = 0	Completely non-random
p-value ≥ α	Random
p-value < α	Non-random

2.4.3 Cryptanalysis Attacks

Kerckhoffs' principle mentioned that the security of a cryptographic algorithm should be guaranteed even if all of the details, except the key, are publicly known (Zhou et al., 2018). In order to maintain robust confidentiality, it is important to investigate the security of an algorithm against a variety of cryptanalysis attacks. Cryptanalysis attacks explore how an algorithm can be broken, which can also help to develop secure algorithms. The two most important attacks on block ciphers are differential and linear cryptanalysis. Both types of attacks were included as the NIST requirements for the AES competition (Boura et al., 2019) and also being implemented by the MySEAL project (CSM, 2021) as the evaluation criteria for the block cipher. Therefore, linear and differential cryptanalysis are carried out in this research.

2.4.3.1 Differential Cryptanalysis

Differential cryptanalysis was introduced by (Biham & Shamir, 1991) and is indeed one of the important techniques in block ciphers. It is also known as a chosen-plaintext attack, in which the attacker is capable to select or choose random plaintext to be encrypted and obtain the corresponding ciphertext.

Differential cryptanalysis exploits the high probability occurrences of plaintext and ciphertext differences. Consider input $X = [X_1, X_2, \dots, X_n]$ and output $Y = [Y_1, Y_2, \dots, Y_n]$ of n-bit block cipher, let two inputs X' and X'' with the corresponding outputs Y' and Y''. The input difference is represented by $\triangle X = X' \oplus X''$ of n-bit vectors and hence,

$$\triangle X = [\triangle X_1 \ \triangle X_2 \ \dots \ \triangle X_n] \tag{1}$$

where $\triangle X_i = X_i' \oplus X_i''$ with X_i' and X_i'' are the i^{th} bit of X' and X''.

Similarly, the output difference is represented by $\triangle Y = Y' \oplus Y''$ and

$$\triangle Y = [\triangle Y_1 \triangle Y_2 \dots \triangle Y_n] \quad (2)$$

where $\triangle Y_i = Y_i' \oplus Y_i''$ with Y_i' and Y_i'' are the i^{th} bit of Y' and Y''.

For an ideal block cipher, the probability of output difference $\triangle Y$ occurs given an input difference $\triangle X$ is $\frac{1}{2^n}$. Thus, if the probability is high, the attacker can select inputs and observe its outputs in an attempt to derive the secret key.

2.4.3.2 Linear Cryptanalysis

Linear cryptanalysis was introduced by (Matsui, 1993) who presented an attack on the full DES algorithm. It is also a known-plaintext attack, in which an attacker has a set of plaintext and the corresponding ciphertext. Linear cryptanalysis exploits the high probability occurrences of liner expressions that involve plaintext, ciphertext, and round key bits. This analysis aims to determine expressions with a high or low probability of occurrence. The expression is defined in the following equation:

$$X_{i_1} \oplus X_{i_2} \oplus \dots X_{i_n} \oplus Y_{j_1} \oplus Y_{j_2} \oplus \dots Y_{j_n} = 0 \quad (3)$$

where X_i is the i^{th} bit of the input $X = [X_1, X_2, \dots, X_n]$ and X_j is the j^{th} bit of the output $Y = [Y_1, Y_2, \dots, Y_n]$ of n-bit block cipher.

An ideal block cipher should hold the probability of occurrence that close to $\frac{1}{2}$. Thus, if the probability of occurrence is far higher or lower than $\frac{1}{2}$, the attacker can distinguish a cipher from a randomly chosen function, given a sufficient number of plaintext/ciphertext pairs.

2.5 Chapter Summary

Security products in electronic devices such as mobile phones require fast encryption solutions as provided by symmetric cryptography. Lightweight block cipher is a solution to the problems encountered in the implementation of encryption in mobile phones. Many algorithms have been proposed every year, one of them is the RECTANGLE lightweight block cipher. RECTANGLE provides excellent performance in software and hardware environments, which offers flexibility for multiple platforms.

Meeting the security evaluation criteria is the key requirement for a cryptographic algorithm. Dedicated cipher key size, block size, and security analysis are the criteria components that have been set by lightweight cryptography projects including NESSIE, NIST, and MySEAL. This research has identified the security evaluation criteria based on the mentioned cryptography projects. For cryptanalysis, three evaluation methods have been selected such as avalanche effect tests, randomness tests, and cryptanalysis attacks.

Confusion and diffusion create a degree of randomness in the cryptographic algorithm. Three analyses under the avalanche effect including correlation coefficient, bit error rate, and key sensitivity tests are significant to further assess the confusion and diffusion characteristics of a cryptographic algorithm.

The ciphertext produced from a cryptographic algorithm must be random. In order to determine whether the tested algorithm fulfils this requirement, randomness testing using the NIST Statistical Test Suite is implemented. The reason for such a decision is that the test suite was used to validate the candidates for the NIST AES competition and the test suite was also being implemented by the MySEAL project.

Linear and differential cryptanalysis are the two most important attacks on block ciphers. Successful linear cryptanalysis could distinguish a cipher from a randomly chosen function. On the other hand, successful differential cryptanalysis may derive the secret key by observing the ciphertext output.