International Journal of Cryptology Research 9(1): 1 - 18 (2019)

Cryptographic Randomness Analysis on Simon32/64

Isma Norshahila Mohammad Shah[*1], **Hazlin Abdul Rani**[1], **Miza Mumtaz Ahmad**[2], and **Eddie Shahril Ismail**[2]

[1]*Cryptography Development Department, CyberSecurity Malaysia, Kuala Lumpur, Malaysia.*
[2]*School of Mathematics, National University of Malaysia.*

E-mail: isma@cybersecurity.my, hazlin@cybersecurity.my, mmumtaz@ukm.edu.my, esbi@ukm.edu.my
[*]*Corresponding author*

ABSTRACT

Internet of Things (IoT) contains private data that must be protected from irresponsible parties. Conventional cryptography algorithms are not compatible with IoT devices due to its limited resources. A family of lightweight cryptography algorithm, Simon, has been developed to fulfill this constraint. Simon family of lightweight cryptography designed by NSA is efficient for optimal hardware performance. A randomness analysis on Simon32/64 is presented in this paper. Analysis is conducted using NIST Statistical Test Suite to ciphertext sequence generated from nine data categories. From the analysis, it reveals that Simon32/64 failed at least one test from each data categories.

Keywords: Internet of Things, Lightweight Cryptographic Algorithm, Simon, Randomness Test, NIST Statistical Test Suite.

1 INTRODUCTION

Internet of Things (IoT) refers to scenarios where objects or devices interact and transfer data among each other via the internet network. IoT allows individuals, animals and objects to work together to create and use variance of services to achieve common goals. The examples of IoT applications that are widely used today are smart appliances, smart energy meters, connected cars and smart healthcare devices. As the evolution of IoT grows to provide new and exciting experience for end users, it also opens up new opportunities for hackers and organized crime.

It is challenging to apply conventional cryptographic standards to small devices. The tradeoff between security, performance and resource requirements in many conventional cryptographic algorithms such as 3DES (Barker and Mouha, 2017) and AES (FIPS, 2001) are optimized for high end device spectrum such as desktop and servers environment. This makes them difficult or impossible to be implemented in resource-constrained devices where their performance may not be acceptable.

To address this challenge, a new field of cryptography, called lightweight cryptography, was formed to focus on designing cryptographic algorithms and protocols that are appropriate for use in resource-constrained devices such as RFID tags, sensors, contactless smart cards, health-care devices and so-on. Some examples of lightweight cryptography algorithm are CLEFIA (Shirai et al., 2007), PRESENT (Bogdanov et al., 2007) and PICCOLO (Shibutani et al., 2011). This paper will illustrate randomness analysis conducted on Simon32/64, which is the recommended algorithm for implementation in RFID systems and wireless sensor networks. A study on implementation of Simon (Feizi et al., 2014) has proven that Simon is suitable for usage in embedded system. Simon32/64 is chosen for analysis since it has the smallest data length and key length. One of the basic properties of cryptographic algorithms is its indistinguishability from a random mapping. Therefore, the evaluation of the randomness of the outputs from this algorithm is of a great importance.

The organization of this paper is as follows. The second section presents previous works on randomness analysis on cryptographic algorithms. The

third section gives a brief description of Simon32/64. The methodology used to conduct randomness analysis is explained in the fourth section. Results and discussion are explained in the fifth section. Finally, this paper is concluded in the sixth section.

2 RELATED WORK

One of the important characteristics of block cipher is its ability to produce random-looking outputs. Pseudorandom number generator (PRNG) statistical test suite which applies a series of statistical tests to the outputs can be used to evaluate the outputs randomness. Diehard (Marsaglia, 1995), Dieharder (Brown et al., 2013) and NIST Statistical Test Suite (Bassham III et al., 2010) are examples of widely used statistical test suite.

NIST Statistical Test Suite is a general-purpose statistical test suite for evaluating the randomness of binary sequences. This test suite was used to evaluate algorithms submitted for AES competition (Soto and Bassham, 2000). For the evolution of AES competition in 1997, Soto and Soto (1999) used nine different ways to generate large sequences of data from block ciphers and the outputs were tested using the NIST Statistical Test Suite.

Recently, several authors have conducted randomness analysis on various types of cryptographic algorithms, using NIST Statistical Test Suite. Randomness analysis on lightweight block ciphers has been conducted to Katan (Lot et al., 2011), Ktantan (Abdullah et al., 2011), Simon (Mohammad Shah et al., 2015) and Speck (Chew et al., 2015). These analyses have been conducted using NIST Statistical Test Suite to the output sequence generated from nine data categories as explained by Soto (Soto and Soto, 1999). Only several variances from Simon algorithms has been evaluated using NIST Statistical Test Suite before, which are Simon64/96, Simon96/96 and Simon128/128 (Mohammad Shah et al., 2015). Apart from these, randomness analysis using NIST Statistical Test Suite on a stream cipher, Grain-128 (Zawawi et al., 2013) has also been conducted.

3 DESCRIPTION OF ALGORITHM

On 2013, National Security Agency (NSA) had designed two new families of lightweight block cipher, Simon and Speck (Ray et al., 2013). They come in a variety of widths and key sizes. Even though both families can perform well in hardware and software platforms, Simon family is optimized for hardware platforms. Simon algorithm operates in classical Feistel Network. Simon family consists of Simon32/64, Simon48/72, Simon48/96, Simon64/96, Simon64/128, Simon96/96, Simon96/144, Simon128/128, Simon128/192 and Simon128/256. Simon 32/64 is Simon algorithm with 32-bit block and 64-bit key. Each round of Simon32/64 uses three operations on its 16-bit words, which are bitwise XOR, $\oplus$, bitwise AND, & and left circular shift, Sj by j bits. It includes a non-linear and non-invertible function, F. Given $X \in \{0,1\}^{16}, F(X)$ is calculated as follows

$$F(X) = (X <<< 2) \oplus ((X <<< 1) \& (X <<< 8))$$

Simon operates on two 16-bit halves in each round. The output of F is XORed with the right half and round key. Output of this XOR operation is swapped with the left half.

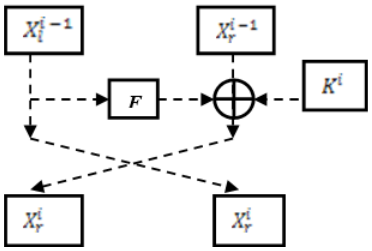

Figure 1: Round Function of Simon.

In the key schedule of Simon32/64, two rotations are performed with shifting to the right by $x >>> 3$ and $x >>> 1$. The results are XORed together with a fixed constants, c and constant sequences, z_0 which are dependent. The value of $z0$ is 0x3479AD88170CA4EF . The number of word for key, K is 4, the first subkeys are driven from K, i.e $K^0, \ldots, K^3$. The subkey for round i

where $4 \leq i \leq r-1$, is calculated as follows

$$m = 4; \quad K^i = K^{i-4} \oplus K^{i-3} \oplus (K^{i-1} >>> 3) \\ \oplus((K^{i-3} \oplus (K^{i-1} >>> 3)) >>> 1) \oplus c \oplus (z_0)_{i-4}$$

where $c = 2^{16} \oplus 3$ is a constant value, $(z_0)_{i-4}$ denotes the i-th bit of z_0 where $4 \leq i \leq r-1$. The value of $i-4$ is modulo by 62.

4 METHODOLOGY

The process of evaluating performing randomness analysis consists of several parts, i.e., taking sample sequences from the algorithm, performing statistical test in NIST Statistical Test Suite and analyzing the results from the conducted statistical randomness tests. Table 1 summarizes the length of output sequence generated in each data category.

DATA CATEGORIES	**LENGTH OF OUTPUT (BITS)**
Strict Key Avalanche	1,001,472
Strict Plaintext Avalanche	1,000,448
Plaintext Ciphertext Correlation	1,000,000
Cipher Block Chaining	1,000,000
Random Plaintext Random Key	1,000,000
Low Density Key	66,592
Low Density Plaintext	16,928
High Density Key	66,592
High Density Plaintext	16,928

Table 1: Length of output sequences generated according to data categories.

4.1 Generating Samples

Output sequence from Simon32/64 is generated using nine data categories, i.e., Strict Key Avalanche, Strict Plaintext Avalanche, Plaintext Ciphertext Correlation, Cipher Block Chaining, Random Plaintext Random Key, Low Density Key, Low Density Plaintext, High Density Key, High Density Plaintext. A total of nine different sets of data are analyzed. Each data set was selected due to its specific function.

4.1.1 Strict Key Avalanche (SKA)

SKA is to examine the sensitivity of individual algorithm to changes in key. 1,000 samples were generated and each sample required a total of 1,001,472 bits binary sequences. The samples were constructed from 489 set of random keys and a set of all zeroes plaintext block. Each block of random key was then used as a base-key. The base-key is encrypted with the all zero-plaintext block to derive a block of base-ciphertext. Then, every bit of the base-key is flipped and encrypted with its respective length of all zero plaintext blocks to get the perturbed-ciphertext. Every block of perturbed-ciphertext is then XORed with the base-ciphertext and concatenated to produce a derived block of at least 106 bits binary output for each sample.

4.1.2 Strict Plaintext Avalanche (SPA)

SPA is to analyze the sensitivity of individual algorithms to changes in plaintext. 1,000 samples were generated and each sample required a total of 1,000,448 bits binary sequences. The samples were constructed from 977 set of random plaintexts and a set of all zeroes key block. Each block of random plaintext was then used as a base-plaintext. The base-plaintext is encrypted with the all zero-key block to derive a block of base-ciphertext. Then, every bit of the base-plaintext is flipped and encrypted with its respective length of all zero key blocks to get the perturbed-ciphertext. Every block of perturbed-ciphertext is then XORed with the base-ciphertext and concatenated to produce a derived block at least 106 bits binary output for each sample.

4.1.3 Plaintext Ciphertext Correlation (PCC)

In order to study the correlation of plaintext/ciphertext pairs, 1,000 samples were generated. Each sequences consisting of at least 106 bits binary. 31,250 blocks of random plaintext and keys for Simon32/64 were used to produce the concatenated ciphertext block of 1,000,000 bits binary of output sequences. The output sequences are constructed by XORed operation between the plaintext block and its corresponding ciphertext block which is computed in ECB mode.

4.1.4 Cipher Block Chaining (CBC)

In this data category, 1,000 samples were generated using 31,250 blocks of random keys for Simon32/64. The derived block of 1,000,000 bit from this data categories are constructed using CBC mode. The first ciphertext block (CT_1) is defined by $CT_1 = E_k(IV \oplus PT)$. Subsequent ciphertext blocks were defined by $CT_{i+1} = E_k(CT_i \oplus PT)$ for $1 \leq i \leq N$ (N is 31,250 blocks of random keys). These derived blocks are then concatenated to produce an output of at least 10^6 bits binary.

4.1.5 Random Plaintext Random Key (RPRK)

In order to examine the randomness of ciphertext, 1,000 samples were generated. Each sample was a result of the concatenation of 31,250 blocks of ciphertext using 31,250 blocks of random plaintexts and random keys for Simon32/64. At least 10^6 bits of sequences are produced.

4.1.6 Low Density Key (LDK)

In this data category, a set of data consisting of 1,000 sequences were generated based on low-density keys block. For each block size, a random plaintext block is used. 2081 ciphertext block were generated for this data category.

The first ciphertext block is obtained using a block of all zeroes keys. The subsequence ciphertext block until ciphertext block number 65 are obtained using a block of keys with a single one in each of the possible bit positions. Then, for the remaining ciphertext blocks it was obtained using a block of keys with two ones and 62 zeroes (the two ones appear in each combination of two bits position within the length of the key). The derived block of ciphertext is concatenated to produce 66,592 bits binary sequence.

4.1.7 Low Density Plaintext (LDP)

In this data category, a set of data consisting of 1,000 sequences are generated. For each block size, a random key block is used. 529 ciphertext block were generated for this data category. The first ciphertext block is obtained using a block of all zeroes plaintext. The subsequence ciphertext blocks until ciphertext block number 33 are calculated using a block of plaintext with a single bit one in each of the possible bit positions. Then, for the remaining ciphertext blocks it was obtained using a block of plaintext with two bit ones and 30 bit zeroes (the two bit ones appear in each combination of two bit position within the length of the plaintext). The derived block of ciphertext is concatenated to produce 16,928 bits binary sequence.

4.1.8 High Density Key (HDK)

In this data category, a set of data consisting of 1,000 sequences are generated based on high-density keys block. A random plaintext block is used. 2081 ciphertext block were generated for this data category. The first ciphertext block is obtained using a block of all ones keys. The subsequence ciphertext block until ciphertext block number 65 are obtained using a block of plaintext with a single zero in each of the possible bit positions. Then, for the remaining ciphertext blocks it was obtained using a block of plaintext with two zeroes and 62 ones (the two zeroes appear in each combination of two bits position within the length of the key). The derived block of ciphertext is concatenated to produce 66,592 bits binary sequence.

4.1.9 High Density Plaintext (HDP)

In this data category, a set of data consisting of 1,000 sequences are generated. For each block size, a random key block is used. 529 ciphertext block were generated for this data category. The first ciphertext block is obtained using a block of all ones plaintext. The subsequence ciphertext blocks until ciphertext block number 33 are calculated using a block of plaintext with a single bit zero in each of the possible bit positions. Then, for the remaining ciphertext blocks it was obtained using a block of plaintext with two bit zeroes and 30 bit ones (the two bit ones appear in each combination of two bit position within the length of the plaintext). The derived block of ciphertext is concatenated to produce 16,928 bits binary sequence.

4.2 Randomness Testing Tools

Randomness analysis is conducted using NIST Statistical Test Suite developed by National Institute of Standards and Technology, USA (NIST) [9]. NIST Statistical Test Suite consists of 15 tests which are divided into two types of categories; parametrized test selection and non-parameterized test selection. There is a recommendation for required minimum bit length for each sample in each test. Table 2 shows the lists of tests available and its description while Table 3 shows the minimum required bit length for each statistical test.

STATISTICAL TEST	DESCRIPTION
Parameterized Test Selection	
Block Frequency Test	To determine whether the number of ones in an M-bit block is approximately M/2 where M is the length of each block.
Non-overlapping Templates Test	To reject sequences that exhibit too many occurrences of a given non- periodic pattern.
Overlapping Template Test	To reject sequences that shows deviations from the expected number of runs of ones of a given length.
Maurer's Universal test	To detect whether the sequence can be significantly compressed without loss of information or not.
Linear Complexity Test	To determine whether the sequence is enough to be random or not.
Serial Test	To determine whether the number of occurrences of m-bit overlapping patterns is approximately the same as would be expected for a random sequence (m-bit is referred to the length in bits of each block).
Approximate Entropy Test	To compare the frequency of overlapping blocks of two consecutive/adjacent lengths (m and m+1) against the expected result for a normally distributed sequence (m-bit is referred to the length of each block).

STATISTICAL TEST	DESCRIPTION
Non-Parameterized Test Selection	
Frequency Test	To determine whether the number of zeroes and ones in a sequence are approximately the same as would be expected for a truly random sequence
Runs Test	To determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence.
Longest Runs of Ones Test	To determine whether the longest run of ones is consistent with the longest run of ones that would be expected in a random sequence.
Binary Matrix Rank Test	To check for linear dependence among fixed length substrings of the original sequence.
Spectral Test	To detect periodic features in the tested sequence that would indicate a deviation from the assumption of randomness.
Cumulative Sums (Forward/ Reverse) Test	To determine whether the sum of the partial sequences occurring in the tested sequence is too large or too small.
Random Excursion Test	To determine if the number of visits to a particular state within a cycle deviates from what one would expect for a random sequence.
Random Excursion Variant Test	To detect deviations from the distributions of the number of visits of a random walk to a certain state.

Table 2: Description of statistical randomness tests.

STATISTICAL TESTS	MINIMUM BIT LENGTH
Parameterized Test Selection	
Block Frequency Test	100
Non-overlapping Templates Test	100
Overlapping Template Test	10^6
Maurer's Universal test	387,840
Linear Complexity Test	10^6
Serial Test	100
Approximate Entropy Test	100
Non-Parameterized Test Selection	
Frequency Test	100
Runs Test	100
Longest Runs of Ones Test	128
Binary Matrix Rank Test	38,912
Spectral Test	1000
Cumulative Sums (Forward/Reverse) Test	100
Random Excursion Test	10^6
Random Excursion Variant Test	10^6

Table 3: Minimum bit lengths for each tests.

User needs to identify parameter value for each test in Parameterized test selection. Table 4 shows the parameter(s) selection characteristics for test in Paramaterized Test Selection as it requires some parameter input.

STATISTICAL TEST	CHARACTERISTICS
Block Frequency Test	User needs to define *M*, block length; *n* = bit sequence, *N* = Partition the input sequence, *n/M* i. $n \geq 100$ ii. $n \geq nM$ iii. $M \geq 100$ iv. $M \geq 0.01n$ v. $N < 100$
Overlapping Template Test	User needs to define *m* = template length; Various values of m may be selected, but for the time being, NIST recommends choosing *m* = 9 or *m* = 10.
Non-Overlapping Templates Test	User needs to define *m* = template length; NIST recommends choosing *m* = 9 or *m* = 10 to obtain the meaningful results.
Serial Test	User needs to define *m* = block length; *n* = bit sequence, $m < [\log_{2n}]-2$
Approximate Entropy test	User needs to define *m* = block length; *n* = bit sequence, $m < [\log_{2n}]-5$
Linear Complexity Test	User needs to define *M* = block length; *n* = bit sequence, *N* = Partition the input sequence, n/M i. $500 \leq M \leq 5000$ ii. $N \geq 200$
Universal Test	User needs to define *L* = block length, and *Q* = number of block; *n* = bit sequence, There is a table in the NIST manual documents for user to identify the parameter of this test. $n \geq 387840$; $L = 6$, $Q = 640$ $n \geq 904960$; $L = 7$, $Q = 1280$ $n \geq 2068480$; $L = 8$, $Q = 2560$

Table 4: Characteristics of Parameterized Test Selection.

4.3 Analysis Framework

The randomness analysis was performed on Simon32/64 based on significance level of 0.1% to the output sequence generated using nine data categories. Table 5 shows the list of input quantity for parameters used in each of test contained in the parameterized test selection.

STATISTICAL TEST	PARAMETER(S)
Block Frequency Test	20,000
Overlapping Templatest Test	10
Non-Overlapping Templates Test	10
Serial Test	2
Approximate Entropy test	2
Linear Complexity Test	2,000
Universal Test	Depends on the length of the bit sequence of the sample

Table 5: Input for the Parameterized Test Selection.

All tests except Cumulative Sums Test, Serial Test, Non-overlapping Templates Test, Random Excursion Test and Random Excursion Variants Test produce one p-value for each sample. Cumulative Sums and Serial Test produce two p-values for each test. Non-overlapping Template Test produces 148 p-values for each sample. For Random Excursions and Random Excursion Variants Test, the p-value produced depends on the sample accepted. Only samples with number of cycles exceeding 500 were evaluated. Samples with insufficient number of cycles were not considered. Note that all p-values were collected and analyzed. The randomness of the algorithm based on data categories are determined by this p-values collected. The sequence for each sample is considered as random if the p-value is more or equal to 0.001. If the p-value

of the sample is less than 0.001, the sample is observed. The formula used to compute the maximum number of failed sample accepted is as follows

$$\gamma = s\left(\alpha + 3\sqrt{\frac{\alpha(1-\alpha)}{s}}\right)$$

where, $\gamma =$ maximum number of failed sample accepted, $s =$ the sample size, $\alpha =$ the significance level.

For example, Non-overlapping Template Test produces 148,000 p-values for each data categories in all algorithms. Therefore, the rejection rate for this test should not exceed 184 sequences. The p-values produced in Random Excursion Test and Random Excursion Variants Test varies as it depends on the sample accepted. The rejection rates for these tests are different according to data categories and algorithm.

As this analysis used 1,000 samples and the significance level is fixed at 0.001, the rejection rate for most of the test in all data categories for all algorithm should not exceed 3 samples.

Random Excursion Test and Random Excursion Variants Test require minimum bit length of 10^6 bits. Therefore, analysis to the output sequence generated from LDK, LDP, HDK and HDP data categories cannot be conducted in these two tests. The rejection rate for Random Excursion Test for SKA data category is 9 while for SPA, PCC, RPRK and CBC data categories, the rejection rate is 11. The rejection rate for SPA data category for Random Excursion Variants Test is 17 while for SPA, PCC, RPRK and CBC data categories, the rejection rate is 21.

5 RESULT AND DISCUSSION

This statistical test was conducted on Simon32/64 algorithms under nine data categories with 1,000 samples. A total of 78,955 binary sequences were evaluated. Five data categories on Simon32/64 that have a sample size of at least 106 are SKA, SPA, PCC, CBC and RPRK. Therefore, sample from these data categories were evaluated using all 16 NIST tests. LDK and HDK are having a

sample size of 66,592 bits sequence while LDP and HDP are having a sample size of 16,928 bits sequence. Therefore, sample from LDK, HDK, LDP and HDP were not evaluated on Overlapping Template Test, Maurers Universal test, Linear Complexity Test, Random Excursion Test and Random Excursion Variant Test. LDP and HDP were also not evaluated on Binary Matrix Rank Test due to lack of minimum required bit lengths for this test.

Table 6 explains a case-by-case description according to data categories. In this Table, only the tests that exceeded the maximum number of rejection rate are shown.

Data Categories	Statistical Test	Maximum Number of Failed Sample Accepted, α	Number of Failed Sample, β	β - α
Strict Key Avalanche	Block Frequency	3	38	35
Strict Plaintext Avalanche	Approximate Entropy	3	4	1
	Random Excursion Variants	17	24	7
Plaintext Ciphertext Correlation	Runs	3	4	1
	Serial (p-value 2)	3	4	1
	Random Excursion Variants	21	23	2
Cipher Block Chaining	Cumulative Sums (Reverse)	3	4	1
	Linear Complexity	3	4	1
Random Plaintext Random Keys	Linear Complexity	3	4	1
	Random Excursion Variants	21	22	1
Low Density Key	Spectral	3	4	1
	Non-Overlapping Template	184	306	122
High Density Keys	Non-Overlapping Template	184	282	98
Low Density Plaintext	Non-Overlapping Template	184	777	593
High Density Plaintext	Non-Overlapping Template	184	717	533

Table 6: Number of sample which exceed the maximum number of failed sample accepted.

According to the results shown in Table 4, we can observe samples that have bigger failed result (more than 100 samples) based on calculations of differences between Number of Failed Samples and Maximum Number of Failed Sample Accepted are Non-Overlapping Templates Test in LDK, LDP and HDP data categories.

6 CONCLUSION

In this research paper, we have examined the randomness analysis on Simon32/64. This analysis was conducted on 1,000 set of samples, for which output sequence were generated using nine data categories. The significance level was set to 0.01% to determine whether the output sequence generated from the algorithm was random or not. From the results of this test analysis, there is at least one statistical test that has exceeded the maximum number of rejection in each data categories. Therefore, it is concluded that based on the output generated from Simon32/64, this algorithm are non-random on 0.1% significance level. As mentioned above, Non-Overlapping Templates Test in LDK, LDP and HDP data categories have larger failed results. Therefore, we advise to avoid using low density and high density values for keys and plaintext for Simon32/64.

REFERENCES

Abdullah, N. A. N., Lot, N. H., Zawawi, A., and Rani, H. A. (2011). Analysis on lightweight block cipher, ktantan. In *2011 7th International Conference on Information Assurance and Security (IAS)*, pages 46–51. IEEE.

Barker, E. and Mouha, N. (2017). Recommendation for the triple data encryption algorithm (TDEA) block cipher. Technical report, National Institute of Standards and Technology.

Bassham III, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E. B., Leigh, S. D., Levenson, M., Vangel, M., Banks, D. L., et al. (2010). Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., Seurin, Y., and Vikkelsoe, C. (2007). PRESENT: An ultra-lightweight block cipher. In *International workshop on cryptographic hardware and embedded systems*, pages 450–466. Springer.

Brown, R. G., Eddelbuettel, D., and Bauer, D. (2013). Dieharder: A random number test suite. *Open Source software library, under development, URL http://www. phy. duke. edu/˜ rgb/General/dieharder. php*.

Chew, L. C. N., Mohammad Shah, I. N., Nik Abdullah, N. A., Ahmad Zawawi, N. H., Rani, H. A., and Zakaria, A. A. (2015). Randomness Analysis on Speck Family Of Lightweight Block Cipher. *International Journal of Cryptology Research*, 5(1):61–77.

Feizi, S., Ahmadi, A., and Nemati, A. (2014). A hardware implementation of SIMON cryptography algorithm. In *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 245–250. IEEE.

FIPS, P. (2001). 197: Advanced encryption standard (AES). *National Institute of Standards and Technology*, 26.

Lot, N. H., Abdullah, N. A. N., and Rani, H. A. (2011). Statistical analysis on katan block cipher. In *2011 International Conference on Research and Innovation in Information Systems*, pages 1–6. IEEE.

Marsaglia, G. (1995). Diehard battery of tests of randomness.

Mohammad Shah, I. N., Chew, L. C. N., Azeala, N., Nik Abdullah, N. A., Ahmad Zawawi, N. H., and Rani, H. A. (2015). Analysis on Lightweight Block Cipher, Simon. *International Journal of Cryptology Research*, 5(2):28–44.

Ray, B., Douglas, S., Jason, S., Stefan, T., Bryan, W., and Louis, W. (2013). The simon and speck families of lightweight block ciphers. *Technical report, Cryptology ePrint Archive, Report./404*.

Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., and Shirai, T. (2011). Piccolo: an ultra-lightweight blockcipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 342–357. Springer.

Shirai, T., Shibutani, K., Akishita, T., Moriai, S., and Iwata, T. (2007). The 128-bit blockcipher CLEFIA. In *International workshop on fast software encryption*, pages 181–195. Springer.

Soto, J. and Bassham, L. (2000). Randomness testing of the advanced encryption standard finalist candidates. Technical report, BOOZ-ALLEN AND HAMILTON INC MCLEAN VA.

Soto, J. and Soto, J. (1999). *Randomness testing of the advanced encryption standard candidate algorithms*. US Department of Commerce, Technology Administration, National Institute of .

Zawawi, N. H. L. A., Seman, K., and Mohd Zaizi, N. J. (2013). Randomness analysis on grain-128 stream cipher. In *AIP Conference Proceedings*, volume 1557, pages 15–20. AIP.