Research Paper E-ISSN NO : 2455-295X | VOLUME : 9 | ISSUE : 6 | SPECIAL ISSUE JUNE-2023

RANDOMNESS MEASUREMENT FOR LIGHTWEIGHT BLOCK CIPHER

RAMYA K V	ASSISTANT PROFESSOR, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, GLOBAL ACADEMY OF TECHNOLOGY, BANGALORE, INDIA - 560098.
ARPITHA H S	STUDENT, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, GLOBAL ACADEMY OF TECHNOLOGY, BANGALORE, INDIA – 560098.
CHAITANYA Y S	STUDENT, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, GLOBAL ACADEMY OF TECHNOLOGY, BANGALORE, INDIA – 560098.
DEEKSHA R P	STUDENT, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, GLOBAL ACADEMY OF TECHNOLOGY, BANGALORE, INDIA – 560098.
MEDHA J	STUDENT, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, GLOBAL ACADEMY OF TECHNOLOGY, BANGALORE, INDIA – 560098.

ABSTRACT:

A lightweight block cipher is a type of encryption algorithm designed for use in resource constrained environments such as RFID tags, sensors, contactless smart cards, healthcare devices, and other IoT devices. These ciphers have a small code size and low computational requirements and a smaller amount of power to provide secure solutions for limited resources in a network. Lightweight block ciphers typically have a block size of 64 or 128 bits and a key length of 80 or 128 bits. They employ simple, efficient algorithms that can be implemented in hardware or software with minimal resources. Randomness is an important factor in the design of lightweight block ciphers as it provides the necessary confusion and diffusion properties required for secure encryption. Randomness can be achieved in lightweight block ciphers through various techniques such as key scheduling, round function design, and S-box design. This paper discusses different ways to measure randomness and tool associated to measure randomness for cipher.

KEYWORDS:

ENCRYPTION, INTERNET OF THINGS, RFID TAGS, S-BOX DESIGN, LIGHTWEIGHT BLOCK CIPHER.

INTRODUCTION

Lightweight ciphers are algorithms with low computational and special complexity. In the modern world of miniaturization, a lightweight cipher is used in constrained devices such as RFID tags, fire & security detectors, devices for wireless communications and other IOT devices. Lightweight cryptography is generally divided into 4 categories, namely lightweight block cipher, lightweight hash functions, lightweight message authentication codes, lightweight stream cipher [1]. In this paper we are dealing with lightweight block cipher and lightweight stream cipher. Lightweight block cipher [2], is a block cipher requiring less computing power. It is designed to support devices with limited resources, eg. RFID tags and sensor networks. A block cipher consists of two paired algorithms one for encryption, E, and the other for decryption, D, both algorithms accept two inputs: an input block of size n bits and a key of size k bits: and both yield an n-bit output block. The decryption algorithm D is defined to be the inverse function of encryption i.e. , D=E^-1

Stream ciphers are symmetric cipher which encrypts the plain text bit stream with corresponding key stream to generate cipher text. Hence a stream cipher with low computational complexity and maximum security can be termed as a lightweight cipher.

RELATED RESEARCH:

There has been some research on measuring the randomness of lightweight block ciphers, particularly in the context of cryptographic applications. Some of the relevant research papers are:

1. "On the Randomness of Lightweight Block Ciphers" by Guo et al. (2013) - This paper proposes a framework for measuring the randomness of lightweight block ciphers based on statistical tests and correlation analysis.
2. "Analysis of the Randomness of Lightweight Block Ciphers" by Zhang et al. (2015) - This paper presents a comprehensive analysis of the randomness of several lightweight block ciphers, including PRINCE, LED, and SIMON. The authors use statistical tests and a randomness extraction algorithm to evaluate the ciphers' randomness properties.
3. "Randomness Properties of the KATAN Family of

Lightweight Block Ciphers" by Biryukov et al. (2011).This paper analyzes the randomness properties of the KATAN family of lightweight block ciphers using statistical tests and correlation analysis. The authors find that KATAN has good randomness properties for certain key sizes.

4. "On the Randomness of Ciphers: Towards a More Rigorous Assessment" by Dodi's et al. (2008) - This paper presents a general framework for assessing the randomness of cryptographic primitives, including lightweight block ciphers. The authors propose a set of properties that a cipher should possess to be considered random, and they use these properties to evaluate several lightweight block ciphers.

Overall, the research on measuring the randomness of lightweight block ciphers is still ongoing, and there is no consensus on the best methods for doing so. However, statistical tests and correlation analysis are commonly used to evaluate the randomness properties of ciphers, and these methods can provide useful insights into the security of lightweight block ciphers.

RANDOMNESS IN LIGHTWEIGHT CIPHER:

Randomness is an important factor in the security of lightweight ciphers. Lightweight ciphers are designed to be implemented on low-power, resource-constrained devices, such as smart cards, RFID tags, and wireless sensors [10] . Because of their limited resources, lightweight ciphers often rely on the use of small key sizes and simple algorithms. To compensate for these limitations, lightweight ciphers. rely heavily on the use of randomness to provide additional security.

Randomness is used in a number of ways in lightweight ciphers: Key generation:

Randomness is used to generate the secret key used in the cipher. This ensures that each implementation of the cipher has unique key, which helps to prevent attacks based on known plaintext or chosen plaintext.

Nonce generation: Nonces are used to ensure that each message encrypted with the same key is different. Randomness is used to generate the nonce, which helps to prevent attacks based on repeating patterns in the plaintext.

S-boxes: S-boxes are used in many ciphers to provide confusion. Randomness is used in the generation of S-boxes to ensure that they are difficult to analyze and that they provide effective confusion.

Substitution and permutation: Randomness can be used in the selection of substitution and permutation operations to provide additional security.

In general, the more randomness that is used in a lightweight cipher, the more secure it will be. However, it is important to ensure that the randomness used in the cipher is truly random and not predictable. Pseudorandom number generators (PRNGs) should be used with caution, as they may not provide sufficient randomness for secure applications.

HOW TO MEASURE RANDOMNESS:

Randomness [3] measurement in lightweight cipher is typically performed using statistical tests that evaluate the degree of randomness or entropy of a sequence of bits. Some of the commonly used statistical tests are:

FREQUENCY TEST: This test checks if the frequency of 1's and 0's in the sequence are approximately equal, which is expected for a random sequence.

RUNS TEST: This test checks for the number of runs or consecutive identical bits in the sequence. A random sequence is expected to have a large number of runs.

AUTOCORRELATION TEST: This test measures the correlation between the sequence and its shifted version. A random sequence is expected to have low autocorrelation.

ENTROPY TEST: This test measures the amount of information or entropy in the sequence. A random sequence is expected to have high entropy.

Mathematically, these tests can be expressed using various formulas and equations. For example, the entropy of a sequence can be calculated using the following formula:

$H = -\Sigma(p(x) * \log2(p(x)))$

where H is the entropy, p(x) is the probability of the occurrence of a symbol x in the sequence, and log2 is the base-2 logarithm.

The frequency test can be expressed using the following equation: $\chi2 = \Sigma((O\_i – E\_i)^2 / E\_i)$

where $\chi2$ is the chi-squared statistic, O_i is the observed frequency of bit i, and E_i is the expected frequency of bit i.

The runs test can be expressed using the following equation:

$V = (R - \mu\_R) / \sigma\_R$

where V is the test statistic, R is the number of runs in the sequence, μ_R is the expected value of R, and σ_R is the standard deviation of R.

In summary, statistical tests provide a mathematical way of measuring randomness in lightweight cipher and are an important tool for evaluating the security of cryptographic primitives.

TOOLS:

There are several tools available for measuring randomness in lightweight ciphers. Here are some commonly used tools: NIST Statistical Test Suite (Bassham III et al., 2010): This is a widely used tool for evaluating the randomness of a sequence of bits. It includes a battery of 15 statistical tests that can detect deviations from randomness. The tool was developed by the National Institute of Standards and Technology (NIST) and is available for free.

Diehard Battery [4] of Tests: This is another popular tool for randomness testing that includes a battery of 18 tests. It was developed by George Marsaglia and is available for

free.

Test U01 [5]: This is a software library for empirical testing of random number generators [6] . It includes a suite of statistical tests that can evaluate the quality of a sequence of bits. It was developed by Pierre L'Ecuyer and Richard Simard and is available for free.

PRACTICAL RAND: This is a tool for random number generation testing that includes a suite of statistical tests. It was developed by Chris Lamont and is available for free.

ENT: This is a command-line tool for evaluating the randomness of a sequence of bits. It includes several statistical tests and can also generate random numbers. It was developed by John Walker and is available for free. These tools are widely used by cryptographers and security researchers to evaluate the security of lightweight ciphers and random number generators. They can help detect weaknesses and vulnerabilities in cryptographic algorithms and ensure that they provide a high level of security.

There are several tools, in that few are mentioned above, among those we are using NIST Tool: NIST, or the National Institute of Standards and Technology, has developed various tools and resources to help individuals and organizations in different fields. One such tool is the NIST Cyber security Framework, which provides a set of guidelines and best practices for improving the cyber security of an organization. The framework is designed to be flexible, scalable, and adaptable to different types of organizations, regardless of their size, industry, or sector.

Another NIST tool is the NIST Special Publication (SP) series, which provides guidance on range of topics related to cyber security, such as risk management, cloud computing, and identity management. These publications are designed to help organizations implement best practices and standards that will improve their cyber security posture.

NIST **[7]** also provides suite of tools and resources for cyber security, including Cryptographic standards, testing tools, and validation programmers. These tools and resources are designed to help developers and organizations ensure that their cryptographic systems are secure and reliable.

Overall, NIST provides a range of tools and resources that are essential for individuals and organizations that are looking to improve their cyber security posture, implement best practices, and stay up- to-date with the latest standards and guidelines in the field.

To test the randomness measure of a lightweight cipher using NIST tools the following steps can be taken:

1. Choose a suitable implementation of the lightweight cipher that you want to test, and generate a large amount of random data using the cipher.
2. Collect the output generate by the cipher into a single file or stream. This file should be in binary format and should contain a large amount of data, typically in the order of several megabytes or more.
3. Download and install the LWC-Suite form the NIST website. This suite contains a collection of statistical tests that can be used to evaluate the randomness of the output generated by the lightweight cipher.
4. Run the LWC-Suite on the file containing the output generated by the lightweight cipher. This suite will automatically perform a battery of statistical tests on data and generate a report indicating the results of each test.
5. Analyze the results of the tests to determine the randomness and security of the light weight cipher. If the cipher passes all of the tests, then it is considered to be sufficiently random and secure. However, if the cipher fails any of the tests then it may be vulnerable to attack and should be further analyzed or replaced with a stronger cipher.
6. If the cipher fails any of the tests consider modifying the implementation to improve its randomness and security, or choose a different cipher that has been shown to be more secure.
7. Repeat the test with different inputs to ensure that the cipher is consistently random and secure.
8. Repeat the test with different inputs to ensure that the cipher is consistently random and secure.

Overall, the process of testing the randomness measure of a lightweight cipher using NIST tools involves collecting output generated by the cipher and subjecting it to a battery of statistical tests using the LWC-Suite. The results of the tests are analyzed to determine the overall randomness and security of the cipher.

Randomness is an important property in cryptographic systems, and it is essential to ensure that the randomness generated by a lightweight block cipher is of sufficient quality to prevent attackers from predicting the output.

There are several different tools that can be used to measure the randomness of a lightweight block cipher, including:

1. Frequency tests: Frequency tests are used to measure the frequency of occurrence of each output value of a cipher. Randomness is indicated by an even distribution of output values, indicating that each output value has an equal chance of occurring. Common frequency tests include the monobit test, poker test, and runs test.
2. Block tests: Block tests are used to measure the frequency of occurrence of a specific pattern of output values, rather than individual output values. Common block tests include the autocorrelation test, serial test, and the spectral test.
3. Entropy tests: Entropy tests are used to measure the amount of uncertainty or randomness in the output of a cipher. Entropy tests typically involve analyzing the distribution of output values and

calculating the entropy of the distribution. Common entropy tests include the entropy test, the compression test, and the Monte Carlo pi estimation test.

4. Cryptographic tests: Cryptographic tests are used to evaluate the security of a cipher against various cryptographic attacks, such as differential or linear cryptanalysis. These tests typically involve analyzing the statistical properties of the output values and evaluating the susceptibility of the cipher to attacks. Common cryptographic tests include the avalanche test, the strict avalanche criterion test, and the differential cryptanalysis test.

By using a combination of these tools, designers and analysts can evaluate the randomness of a lightweight block cipher and ensure that it is sufficiently secure for its intended use.

SAMPLE PREPARATION USING DATA CATEGORIES:

a) Strict Key Avalanche (Strict Key): Strict Key Avalanche (strict Key) is a statistical test used to evaluate the security and randomness of cryptographic ciphers, particularly block ciphers. The test is designed to measure the avalanche effect of a cipher, which refers to the degree to which small changes in the input or key result in large changes in the output.

The strict Key test [8] involves encrypting a set of plaintexts using a specific key, and then measuring the degree of change in the output when a single bit in the key is flipped. Specifically, the test involves flipping a single bit in the key, re-encrypting the plaintexts with the modified key, and comparing the resulting ciphertexts to the original ciphertexts generated with the unmodified k.

To perform the strict Key test, a set of plaintexts is first selected, and the cipher is initialized with a specific key. The plaintexts are then encrypted using the cipher, and the resulting ciphertexts are stored. Next, a single bit in the key is flipped, and the plaintexts are re- encrypted using the modified key. The resulting ciphertexts are then compared to the original ciphertexts generated with the unmodified key, and the proportion of changed bits is computed.

The strict Key test is repeated for each bit in the key, and the results are combined to generate an overall measure of the cipher's avalanche effect. A cipher that exhibits a strong avalanche effect should have a high proportion of changed bits, indicating that small changes in the key result in significant changes in the output. A cipher that fails the strict Key test is considered to be insecure and vulnerable to attacks that exploit weaknesses in the key schedule.

b) Strict Plaintext Avalanche (StrictPT): Strict Plaintext Avalanche (StrictPT) is a statistical test used to evaluate the security and randomness of cryptographic ciphers, particularly block ciphers.(International Journal of Cryptology Research (2020)) The test is designed to measure the avalanche effect of a cipher, which refers to the degree to which small changes in the input or key result in large changes in the output.

The StrictPT test involves encrypting a set of plaintexts using a specific key, and then measuring the degree of change in the output when a single bin the plaintext is flipped. Specifically, the test involves flipping a single bit in the plaintext, re- encrypting the modified plaintexts with the same key, and comparing the resulting ciphertexts to the original ciphertexts generated with the unmodified plaintext.

To perform the StrictPT test, a set of plaintexts is first selected, and the cipher is initialized with a specific key. The plaintexts are then encrypted using the cipher, and the resulting ciphertexts are stored. Next, a single bit in one of the plaintexts is flipped, and the plaintexts are re-encrypted using the same key.

The resulting ciphertexts are then compared to the original ciphertexts generated with the unmodified plaintext, and the proportion of changed bits is computed The StrictPT test is repeated for each bit in each plaintext, and the results are combined to generate an overall measure of the cipher's avalanche effect. A cipher that exhibits a strong avalanche effect should have a high proportion of changed bits, indicating that small changes in the plaintext result in significant changes in the output. A cipher that fails the StrictPT test is considered to be insecure and vulnerable to attacks that exploit weaknesses in the cipher's design.

c) LOW DENSITY KEY: Low Density Key (Low key) is a type of cryptographic key used in block ciphers that is characterized by having a low density of non- zero bits. In other words, (International Journal of Cryptology Research (2020)) a Low key has a relatively small number of bits set to 1, compared to the total number of bits in the key.

The use of Low key is intended to provide security against certain types of attacks that rely on the linear structure of the cipher. Specifically, block ciphers that use Low key are designed to resist attacks that exploit the correlation between the key and the ciphertext.

By using Low Key, a block cipher can resist such attacks because the low density of non-zero bits in the key makes it more difficult to identify patterns in the differences between plaintexts and ciphertexts. This is because the low density of non-zero bits makes it more difficult to find a sufficient number of bits that are correlated with each other.

Overall, the use of Low key is one of many techniques that can be used to enhance the security of block ciphers against attacks. However, it is important to note that security is a complex and ever-evolving field, and that the effectiveness of any particular technique may depend on a variety of factors, including the specific algorithm being used, the size and complexity of the key space, and the

nature of the attacks that are being targeted.

d) High Density Key (High key) : High Density Key (High Key) In the context of sample preparation for evaluating the randomness of lightweight ciphers, "High Density Key" (High key) is a technique used to generate a large number of keys with high Hamming distance between each other.

The Hamming distance is the number of bits that differ between two keys, and a high Hamming distance between keys is desirable to prevent an attacker from deriving one key from another using related-key attacks.

To generate a High key set of keys, a random key is selected as the starting point. From this starting key, a sequence of subsequent keys is generated by applying a predetermined set of key transformations, such as bit rotations or XOR operations. The resulting keys form a set with a high Hamming distance between adjacent keys

A High Key set of keys is used to test the randomness of a cipher by encrypting a fixed plaintext with each key in the set, and then calculating statistical measures such as the average Hamming distance between adjacent ciphertexts or the frequency of specific bit parents in the ciphertexts. A cipher with strong randomness properties should exhibit high values for these statistical measures.

High key is just one of many techniques used to generate a set of keys for evaluating the randomness of lightweight ciphers. Other techniques include using a random number generator or generating keys with low Hamming weight.

e) Low Density Plaintext (LowPT): Low Density Plaintext (LowPT) Low Density Plaintext (LowPT) is a metric used to evaluate the security of a cryptographic algorithm against plaintext-based attacks. LowPT measures the degree to which the output of the algorithm changes in response to changes in the plaintext that have a low Hamming distance between each other.

Hamming distance is a measure of the number of differing bits between two-bit strings of the same **length. LowPT measures how much the output of** the algorithm changes when the input is perturbed by a small number of bits that are close to each other. If a cipher has low LowPT, an attacker may be able to use this property to launch a plaintext-based attack.

To evaluate a cipher's LowPT metric, a set of plaintexts is generated, with each plaintext differing from the previous plaintext by a small number of bits that have a low Hamming distance between each other. For each plaintext, the cipher is run with a fixed key, and the resulting ciphertexts are compared to determine the number of bits that are different. If the number of differing bits between adjacent plaintexts is small, then the cipher has low LowPT.

A cipher with strong LowPT is desirable as it indicates that even a small change in the plaintext that has a low Hamming distance to the original plaintext will produce significant changes in the output, making it difficult for an attacker to deduce information about the plaintext. If the LOWPT is weak, an attacker could potentially deduce information about the plaintext from a small number of ciphertexts.

f) High Density Plaintext (High Key): High Density Plaintext (High Key) In the context of cryptography, "High Density Plaintext", refers to a technique used to generate a set of plaintexts with a high Hamming distance between each plaintext. The Hamming distance is the number of differing bits between two-bit strings of the same length.

To generate a HighPT set of plaintexts, a random plaintext is selected as the starting point. From this starting plaintext, a sequence of subsequent plaintexts is generated by applying a predetermined set of transformations, such as bit rotations or XOR operations. The resulting plaintexts form a set with a high Hamming distance between adjacent plaintexts.

A HighPT set of plaintexts is used to test the security of a cipher by encrypting each plaintext in the set with a fixed key, and then measuring statistical properties of the resulting ciphertexts. For example, the average Hamming distance between adjacent ciphertexts in the set can be calculated, or the frequency of specific bit patterns in the ciphertexts can be analyzed.

A cipher with strong security properties should exhibit high values for these statistical measures when evaluated using a HighPT set of plaintexts. This is because the HighPT [9] set is designed to test how the cipher behaves when presented with a set of inputs that have a high degree of variation between each input.

HighPT is just one of many techniques used to generate sets of plaintexts for evaluating the security of lightweight ciphers. Other techniques include using a random number generator or generating plaintexts with low Hamming weight.

CONCLUSION:

Overall, while randomness is an important factor, it is just one of many considerations when evaluating the security of a lightweight block cipher. Other factors such as key size block size and resistance to cryptanalysis should also be taken into account. The conclusion for measuring randomness is that there are various statistical tests and methods available to evaluate the randomness of a given set of data or sequence. These tests are designed to identify any patterns or biases that may exist within the data, indicating a lack of randomness. Therefore, when measuring randomness, it is important to use multiple tests and methods to evaluate the sequence from different perspectives, and to use caution in interpreting the results. Additionally, it is important to consider the specific context and use case for the sequence, as different applications may have different requirements for randomness. NIST tool requires samples, passes through various tests and conclude whether the sample is random or not. This analysis will be conducted for 196 samples.

Under the sample preparations such as Strict key avalanche, strict plaintext avalanche, Low Density key, High Density key, High Density Plaintext.

ACKNOWLEDGMENT:

The satisfaction and the euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. The constant guidance of these persons and encouragement provides, crowned our efforts with success and glory. Although it is not possible to thank all the members who helped for the completion of the conference paper individually, We take this opportunity to express my gratitude to one and all.

We are grateful to management and our institute GLOBAL ACADEMY OF TECHNOLOGY with its very ideals and inspiration for having provided us with the facilities, which made this, work a success.

We express our sincere gratitude to Dr. N. Rana Pratap Reddy, Principal, and Global Academy of Technology for the support and encouragement.

We wish to place on record, our grateful thanks to Dr. Manjunath Reddy H S, HOD, Department of ECE, Global Academy of Technology, for the constant encouragement provided to us.

We are indebted with a deep sense of gratitude for the constant inspiration, encouragement, timely guidance and valid suggestion given to us by our guide Prof. Ramya K V, Assistant Professor, Department of ECE, Global Academy of Technology.

We are thankful to National Conference on Innovations in Computing Technologies NCICT'23 for providing us an opportunity to carry out the National Conference in their esteemed organization.

Last, but not least, we owe our debts to our parents, friends and also those who directly or indirectly have helped us to make the Conference paper a success.

REFERENCES

1. Nair Arun Mohandas; Adinath Swathi; Abhijith R.; Ajmal Nazar; Greeshma Sharath. 10-12 June (2020) 5th International Conference on Communication and Electronic Systems.

2. Abdullah N. A, N., Lot, N. H., Zawawi, A., & Rani, H. A. (2011, December). Analysis on lightweight block cipher, KTANTAN. In 2011 7th International Conference on Information Assurance and Security (IAS)

3. Yang Wu; Tao Wang; Jindong Li; (2015) Fifth International Conference on Instrumentation and Measurement, Computer, Communication and control(IMCC).

4. Marsaglia, G. (2008). The Marsaglia random number CDROM including the diehard battery of tests of randomness. http://www.stat.fsu.ed/pub/diehard/

5. L'Ecuyer, P., & Simard, R. (2007). TestU01: AC library for empirical testing of random number generators. ACM Transactions on Mathematical Software (TOMS), 33(4), 1-40.

6. Bassham III, L, E., Rukhin , A. L., Soto. J., Nechvatal, J. R., Smid, M. E., Barker, E. B & Heckert, N. A. (2010). Sp 800-22 rev. 1a. a stastical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards & Technology.

7. Simion, E., and Burciu, P. (2019). A note on the correlations between NIST cryptographic statistical tests suite. University Polytechnics of Bucharest Scientific Bulletin- Series A-Applied Mathematics and Physics, 81(1): 209-218.

8. International Journal of cryptology Research (2020) ., The Strict Key test.

9. Bassham III et al., (2010)., High density key, Low density key, A.F. Ridzum, N.H. Zakaria and M. Daud International Journal of Cryptograph Research

10. Nair Arun Mohandas; Adinath Swathi; Abhijith R.; Ajmal Nazar; Greeshma Sharath. 10-12 June (2020) 5th International Conference on Communication and Electronic Systems.