

Key Derivation Functions and cryptographic Feistel Cipher

Hasan Kadhim A. Alsuwaiedi¹, Abdul Monem S. Rahma²

¹Iraqi Commission for Computers & Informatics, Informatics Institute for Postgraduate Studies, Baghdad, Iraq,

²Computer Science Department Al-Maarif University College, Iraq,

Date of Submission: 28-03-2023

Date of Acceptance: 07-04-2023

ABSTRACT

Because of its significance in the formation of the core of encryption/decryption algorithms, the key derivation function (KDF) or pseudorandom number generator (PRNG) plays a major part in the cryptographic system. Random numbers are required for the usage of encryption and decryption techniques in a variety of network security applications. In this research we Focus onkey generation functions, which are an important aspect of symmetric and asymmetric algorithms. Disgusting the background of PRNG basics and attempting to characterize and evaluate several publications that offer proposals for key generators.

Keywords: Key Derivation Function, Pseudorandom Number Generator (PRNG), Key Generation.

I. INTRODUCTION

The continuous developments in digital information technology, which are made by the evolution in the digital and electronic contents and applications, in addition to, the expansion of the Internet network, have complicated the digital world and necessitates the preservation of critical and sensitive data transmitted. It is also well known that Internet communication devices and tools are very inexpensive around the world, with the only negative being the issue of security, which can lead to the urgent need to invest heavily in the development of encryption methods. In today's digital world, information security has become one of the most crucial criteria [1].

In information security systems, key derivation functions or pseudorandom number generators (PRNGs) are employed in numerous domains of technology. The PRNG can be categorized based on several characteristics, including the implementation technique (software, hardware), immunity to exposure (cryptographically secure, cryptographically insecure), and the algorithms used the one-way

functions, elementary functions, and the shift registers [2].

When a key is strong, it can withstand cryptanalysis, even if the attacker discovers all system information linked to the production or verification of the encryption key. The remainder of this paper is organized as follows: In Section 2, we go through the basics of the infrastructure, as well as the different types of key generation functions and how they're implemented. In Section 3, Tests and measurements for the PRNG Key Generator are explained. In Section 4, the analysis of the comparison table. Finally, in section 5, the Conclusions are stated.

II. BACKGROUND

A. Key derivation function

The Key Derivation Function (KDF) is a function that takes an input value (key and other input data) and returns an additional key that cryptographic algorithms can use. The KDF uses pseudorandom functions. The Key Derivation Function was explained as a process that possibly involves two different steps: 1) extraction of randomness to generate an initial key. 2) Iterative key expansion to generate new keys from the starting key and other inputs.

Inside the KDF, a pseudorandom function is used which is considered as a basic construction block represented as $PRF(s,x)$ where s as index or seed and input variables x . The pseudorandom function's output can be utilized as keying material. [1]. For the derivation, the use of either the keyed-hash Authentication Message (HMAC) or the Message Authentication Message (CMAC), which uses ciphers considered as the pseudorandom function [1][2].

A key derivation function iterates a pseudorandom function several times and concatenates the results until the required length bits of keying material are generated as in Figure 1 below [3].

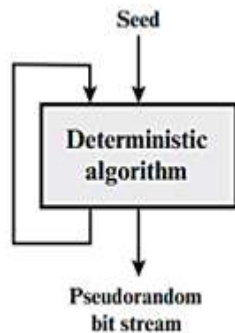


Fig. 1. Pseudorandom Number Generators

The cryptographic applications typically using the algorithmic techniques concerned the generation of the pseudorandom numbers which can be used in, key distribution and mutual authentication schemes, session key generation, RSA public-key algorithm creation, and bitstream generation in symmetric encryption (ex. RC4 algorithm) [4].

There are two types of pseudorandom number generator algorithms: -

- **The linear congruential generator:** the linear congruential method is a widely used technique proposed by Lehmer for pseudorandom generators. Represented by the equation: -

$$X_{n+1} = (aX_n + c) \text{ mod } m$$

m	the modulus	$m > 0$
a	the multiplier	$0 < a < m$
c	the increment	$0 \leq c < m$
X_0	the starting value, or seed	$0 \leq X_0 < m$

The m , a , c and X_0 are integers, then the equation will generate a sequence with integers in the domain $0 \leq X_n \leq m$. the chosen values a , c , and m is crucial in generating a good random number technique [4].

- **Blum BlumShub generator:**

The Blum-Blum-Shub (BBS) generator (writers Lenore Blum, Manuel Blum, and Michael Shub) was first proposed in 1986 and has since become a widely used secure PRNG technique. It's built on the employment of a crypto-secured one-way function that's focused on program implementation. The BBS generator generates a **Bi**-bit sequence using the following algorithm:

$$X_0 = s^2 \text{ mod } n$$

for $i = 1$ to ∞

$$X_i = (X_{i-1})^2 \text{ mod } n$$

$$B_i = X_i \text{ mod } 2$$

s is a random number relatively prime to n . $n = p * q$ (both p and q are prime numbers too). As a result, for each iteration, the least significant bit is taken. as shown in figure 2 [5].

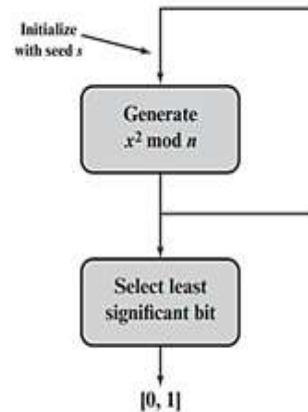


Fig.2. Blum Blum Shub Block Diagram

B. Utilization of Pseudorandom Number Generation

i) Block Cipher

One of the famous approaches to building the PRNG, by using the symmetric block cipher which is considered the core of PRNG mechanisms. Any input clear text (plaintext), produces a symmetric block cipher text as a random output. Therefore, a symmetric block cipher is a good mechanism to construct a pseudorandom number generator.

DES and AES are considered standard block cipher mechanisms therefore the security distinctive of the PRNG can be confirmed and established in many implementations of DES and AES applications.

The counter **CTR**, in every plaintext block, is XORed with an encrypted counter. For each subsequent block, the counter is increased. And the Output Feedback **OFB** when the previous encryption output is utilized as the input to the encryption process, and entire blocks are employed. Both the CTR and OFB represent the

two approaches to using the block ciphers for constructing the PRNG as illustrated in figure 3 [4].

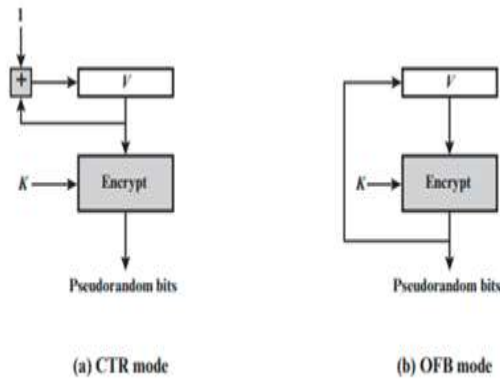


Fig.3. Block Cipher-Based PRNG Mechanisms

ii) Stream ciphers

Stream ciphers encrypt one bit or byte or larger than a byte at one time. Figure 4 depicted the structure of the stream ciphers, which depicted the encryption and decryption process with the combination of the pseudorandom number generation (keystream generator). The stream cipher could be as secure as a block cipher with the same key length. If there is an application that demands encryption and decryption of stream data, over a communication channel or web link/browser, the stream cipher might be better than the block cipher [5].

RC4 is one of the basic stream cipher algorithms designed by Rone Rivest for RSA in 1986 which uses the random permutation. One of its main implementations is the wireless LAN standard IEEE 802.11, of Wi-Fi Protocol Access (WPA) which uses the RC4 algorithm [6].

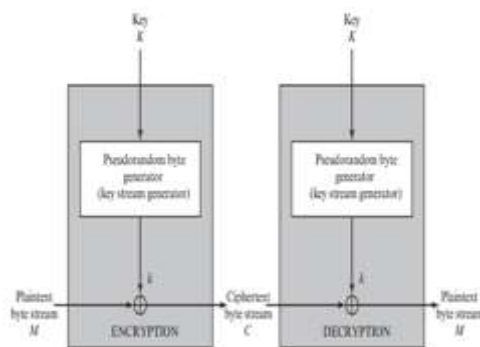


Fig.4. Stream Cipher

III. PRNG KEY GENERATOR TESTS

The output secrecy of PRNG produces specific requirements in the disciplines of randomization, unpredictability, and seed characters arise as a result.

- **Randomness:** although being deterministic, the resulting PRNG bit stream should appear random. Randomness has three characteristics:
 - Uniformity: this means in short, it's equally likely that zero or one will appear, with the probability of each being exactly as $n/2$, where n = sequence length.
 - Scalability: If a chain is random, any subsequence retrieved from it should be random as well.
 - Consistency: A generator's behavior must be consistent across different starting values (seeds).

[4] The NIST standards SP 800-22 registered fifteen statistical analysis recommended tests, beyond the scope of this paper. Therefore we list three important types as follows:-

- Frequency test: The target of this mechanism is to specify whether a sequence's number of ones and zeros are even and the same as it would be predicted for a truly random sequence.
- Runs test: The goal of this method is to check if the number of one-and-zero runs of various lengths is as predicted for a random pattern [7].
- Maurer's test: The aim of this test is the number of bits between the same equivalent patterns (compressed sequence). The test's goal is to see if the matching sequence can be compressed greatly without sacrificing any information [7].

- **Seed requirements:** the seed that works for input to the PRNG must be secure for cryptographic applications. Because the PRNG is a deterministic algorithm, the output can be classified if the attacker can determine the seed. As a result, the seed must be unpredictably unexpected. The seed should, in reality, be a pseudorandom digit [4].

- **Unpredictability:** should be present in a stream of pseudorandom numbers (both forward and backward). If the created bit-stream appears random, it is hard to forecast a specific bit or bit-sequence based on previous bits' capacities. Also, if the bit-sequence looks to be random, there is no way to guess the seed from the bit-sequence [6].

IV. LITERATURE SURVEY

S.M and A.M in this paper [8], proposed a new modification to the Blowfish algorithm which use three generated keys, instead of using an XOR, a multi-state operation is used in the encryption and decryption procedures. Rather than one for

calculating the state table numbers and managing the variable block bit widths (1, 2, 4, and 8 bits). To increase the complexity of the proposed technique, these tables are created from the addition table in a Galois field $GF(2^n)$ based on variable block bit size.

W. W and C. W [9], proposed a security framework model known as The CAM (Adaptive Chosen All Inputs Model) examines the security of KDFs in terms of bit flipping and timing attacks. Existing KDFs are evaluated using this security model. The results reveal that for both the bit-flipping and timing attacks, none of the available KDFs are safe in CAM. The cryptographic keys' security feature is indistinguishable from random strings of identical length. As a result, this study proposes a new KDF that has been shown secure in CAM.

L.T and H.L [10] submitted a modified data encryption algorithm in cloud computing. By analyzing and modifying the advanced encryption standard for data security in cloud computing through a proposing a random disturbance information to improve the data security, by column mix operation and key choreography in AES are improved. Tests and experiments are done on Hadoop, indicate that the proposed solutions ensures that the attribute privacy and algorithm efficiency for data storage in mobile cloud computing.

Talib M and Lahieb Mohammed [11], based on the confusion and diffusion architecture they proposed a new concept of text encryption algorithm. Suggested Pseudo-Random Number Generator (PRNG) approach derived from the input key's SHA-1 digest and plain text using Newton Raphson's method. The suggested PRNG was designed to generate a random integer key sequence that is the same length as the plaintext needed to permute the characters locations in the confusion stage, as well as a starting key for the RC4 random generator. During the diffusion step, the RC4 method is employed to produce a new PRNG key sequence to modify the value of the characters.

Álvarez. R and Martínez. F [12] proposes a novel nonlinear filter design that improved the output sequences of pseudo random generators in terms of complexity. It is built on four seed-dependent substitution-boxes, an evolving internal state register, and a mixture of different sorts of operations with the goal of diffusing nonrandom patterns of input sequence. It is inspired by techniques used in symmetric ciphers. The approach delivers great unpredictability and can even convert a basic counter generator into fully

useable pseudo random sequences, according to test findings. Furthermore, the performance was great while the storage use was minimal, and the implementation used low-power computing platforms.

K. Raghunanda and Aithal Ganesh [13] using the notion of false modulus and the extended Pell's equation, we may improve security. The extended Pell's equation reveals that the public key exponent is dependent on multiple factors, making obtaining the private key parameter a major difficulty. Which reflect on emphasizes the RSA by propose a new method of key generation.

Alz W. and V. Liwandouw [14] the diffusion qualities of the Shannon principle may be achieved by using logistics functions as a random number generator in a cryptography procedure. This research proposes three strategies for providing input key flexibility while employing the domain logistics function's x_0 value. Each method does not provide a sample that is inversely or directly proportional to the value of x_0 and relative error x_0 . As a result, key inputs may be used to determine the existence of logistical functions in creating chaotic numbers. Furthermore, the resulting random numbers are dispersed equally across the chaotic range, which strengthens the process if used as a key in cryptography.

H. Atee, R.Ahmad [15] proposed a sub key generation concept for achieving a secure cryptosystem based on Artificial Neural Network (ANN) algorithm. The input hidden layer weights and data are initialized (in each cycle). The initialization key is based on the topology of the ANN, the activation function, and the seeds for the Pseudo-Random Number Generator (PRNG). The output layer weights are used to create the sub-key in each round.

J.Mcginthy, A. Michaels [16] introduced a pseudorandom number generator (PRNG) based on the Residue Number System (RNS) and the key derivation function (PKDF) that has high initial energy efficiency for IoT devices. Implementations were performed, Subsequent embedded software on a MSP430 and MSP432 (mixed-signal microcontroller) and the key generation function of the Transport Layer Security (TLS) 1.3 hash-based message authentication code (HMAC). Adding an extra layer of security on top of the PRNGs to provide design flexibility for resource-constrained devices.

Baby T. and Sujatha R. [17] Chaotic autonomous oscillator circuits were created in this research to generate a binary sequence that may be utilized as a key generator for cryptographic

methods to ensure data security. The resulting binary sequences are tested for randomness using the NIST standard test and the Dieharder battery test. Furthermore, these random sequences may be utilized for a variety of purposes, including NONCE creation, One-time-pad, and initial password in cryptosystem settings.

V. COMPARISON TABLE

In this table we analyze and compare the chosen papers in the subjects of Algorithm used, Key size, Flexible, Feature and Methods of key Generation

No. Ref	Algorithm	Key size	Flexible	Feature	Methods of key Generation
[8]	New modification of Blowfish algorithm	Variable block bit size	Efficiency	The results demonstrate that the keys are more resistant to attempts to shatter them.	Multi-state tables operation instead of an XOR operations
[9]	CAM is a new security model for analyzing the security of KDFs.	keys cannot be distinguished from random strings of equal length.	Security versus performance efficiency trade-off	In terms of bit-flipping and timing attacks, assess the security of KDFs.	Randomness timing of the alternative KDF
[10]	A Modified Advanced Encryption Standard (AES)	key choreography and column mix operation	yes	In mobile cloud computing, there is excellent security for external data storage.	algorithm has better key sensitivity
[11]	Enhanced RC4 Algorithm	The input key's SHA-1 digest.	high-performance and high-security	Results comes with big secret key size for resisting force-brute attacks, high sensitivity at secret key and plain-text, resisting the entropy attack, and generating uniform histogram, and time fast encryption	Chaotic pseudo-random generator
[12]	A novel nonlinear filter that enhances the output patterns of commonly used pseudo random generators.	Key-stream sequences	Excellent performance	High unpredictability is achieved, and it's even possible to turn a simple counter encoder into perfectly useable pseudo random patterns.	The combination of numerous sorts of operations is based on four seed-dependent substitution-boxes and a developing internal state register.
[13]	Enhanced Algorithm RSA	Public key exponent	The encrypting and	The suggested cryptosystem is difficult in the	Forged modulus was utilized, and Pell's equation

			decryption processes take the same amount of time.	same way that the factorization problem is.	was generalized.
[14]	Designed a scheme capable of providing input key of generation function	/	providing key input flexibility	When used as a key in cryptography, the algorithm's strength.	Key inputs can be used to create logistics functions for creating chaotic numbers.
[15]	Artificial Neural Network (ANN) algorithm	Initialized key depend on ANN topology	Efficiency	The ANN output layer weights are used to create the sub-key in each round	Seeds and activating function for Pseudo Random Number Generator (PRNG)
[16]	Based on a pseudorandom number generator, the Residue Number System (RNS) was developed	The key is determined using the key derivation function (KDF)	Providing flexibility	Implementations were performed, Subsequent embedded software on a MSP430 and MSP432 (mixed-signal microcontroller)	Key derivation function for hash-based message authentication code (HMAC). On top of the PRNGs, an extra layer of protection is added.
[17]	chaotic autonomous oscillator circuits cryptographic algorithms	stream	used for various protocols in crypto systems	The resulting binary sequences are tested for randomness using the NIST standard and Dieharder battery tests.	generate a binary sequences which can be used as key

VI. CONCLUSION

In this research, we talked about a background of the PRNG fundamentals and review at several key derivation functions or pseudorandom number generators (PRNG) for various algorithms across times (2016-2022). writing about key generation security, a literary research review, and a paper by some authors about methods that work with any algorithm. We noticed the efficiency and security of algorithms employed in all forms of algorithms when analyzing the suggested research on key creation in various types. This article literature review gives a brief introduction and analysis of algorithms, as well as some guidance on how to improve encryption systems to ensure network security.

REFERENCES

- [1]. Lily Chen, Draft NIST Special Publication 800-108 Recommendation for Key Derivation Using Pseudorandom Functions, October 2021.
- [2]. FIPS 198-1, The Keyed-Hash Message Authentication Code (HMAC), Revision expected to be published in 2008.
- [3]. Morris Dworkin, NIST SP 800-38B, Recommendation for Block Cipher Modes of Operation – The CMAC Mode for Authentication, INCLUDES UPDATES AS OF 10-06-2016.
- [4]. William Stallings, Cryptography and Network Security Principles and Practice Seventh Edition Global Edition, Pearson Education Limited 2017.

- [5]. Shanshan Yu, Development of Modified Blum-Blum-Shub Pseudorandom Sequence Generator and its use in Education, *Measurement Science Review*, 22, (2022), No. 3, 143-151.
- [6]. Talib M. Jawad, An Enhanced RC4 Algorithm Using An Efficient PRNG generation Method Based on Newton Raphson Method, SHA-1 and Piecewise Chaos Method, College of Information Engineering, Al-Nahrain University, Baghdad, *Journal of Xi'an University of Architecture & Technology Volume XII, Issue IV, 2020 ISSN No : 1006-7930*.
- [7]. Steven D. Galbraith, *Mathematics of Public Key Cryptography*, Cambridge University Press, 2018.
- [8]. Suhad Muhajer Kareema, Abdul Monem S. Rahma, A new multi-level key block cypher based on the Blowfish algorithm, *TELKOMNIKA Telecommunication, Computing, Electronics and Control Vol. 18, No. 2, April 2020, pp. 685~694*
- [9]. Wen WenKoh , Chai Wen Chuah, Robust security framework with bit-flipping attack and timing attack for key derivation functions, *The Institution of Engineering and Technology*, 2020, Vol. 14 Iss. 5, pp. 562-571
- [10]. Lin Teng, Hang Li, A Modified Advanced Encryption Standard for Data Security, *International Journal of Network Security*, First Online June 26, 2019 (VDOI: 1816-3548-2019-00016)
- [11]. Talib M. Jawad, Lahieb Mohammed, An Enhanced RC4 Algorithm Using An Efficient PRNG generation Method Based on Newton Raphson Method, SHA-1 and Piecewise Chaos Method, *Journal of Xi'an University of Architecture & Technology*, Vol XII, Issue IV, 2020 ISSN No : 1006-7930 Page No: 4466
- [12]. Rafael Álvarez, Francisco Martínez, Improving the Statistical Qualities of Pseudo Random Number Generators, Department of Computer Science and Artificial Intelligence (DCCIA), University of Alicante, 03690 Alicante, Spain; *Symmetry* 2022, 14, 269.
- [13]. K. R. Raghunanda , Aithal Ganesh, Key Generation Using Generalized Pell's Equation in Public Key Cryptography Based on the Prime Fake Modulus Principle to Image Encryption and Its Security Analysis, *Cybernetics And Information Technologies*, Vol 20, No 3, Sofia 2020.
- [14]. Alz Danny Wowor and Vania Beatrice Liwandouw, Domain Examination of Chaos Logistics Function As A Key Generator in Cryptography, *International Journal of Electrical and Computer Engineering (IJECE)* Vol. 8, No. 6, December 2018, pp. 4577 – 4583.
- [15]. HayfaaAbdulzahraAtee, Robiah Ahmad, Machine learning based key generating for cryptography, *Journal of Engineering and Applied Science, Medwell Journals*, 2016.
- [16]. Jason M. Mcginthy, Alan J. Michaels, Further Analysis OfPrng-Based Key Derivation Functions, DOI 10.1109/ACCESS.2019.2928768.
- [17]. Baby H. T., Sujatha B. R., Chaotic Key Generators for Data Security, *IOP Conf. Series: Materials Science and Engineering*, Doi:10.1088/1757-899X/1110/1/012013, 2021