ARTICLE

JOURNAL OF UNIVERSITY OF BABYLON

For Pure and Applied Sciences (JUBPAS)

Vol.31; No.2.| 2023

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | ISSN: 2312-8135 | Print ISSN: 1992-0652

An Overview of Parallel Symmetric Cipher of Messages

Hawraa Jabir [1*], Ahmed Fanfakh [2]

1College of Science for Women, University of Babylon, hawraajabir@gmail.com, Babel , Iraq.

2College of Science for Women, University of Babylon, ahmed.fanfakh@uobabylon.edu.iq, Babel , Iraq.

*Corresponding author email: hawraa.hamiza.gsci106@student.uobabylon.edu.iq

نظرة عامة على التشفير المتوازي المتماثل للرسائل

حوراء جابر حمزه*[1]، احمد بدري مسلم [2]

1كلية العلوم للبنات ، جامعة بابل ، hawraajabir@gmail.com، بابل ، العراق

2 كلية العلوم للبنات ، جامعة بابل , ahmed.fanfakh@uobablyon.edu.iq، بابل ,العراق

Received:	15/1/2023	Accepted:	20/4/2023	Published:	30/6/2023

Abstract

Background:

Despite significant developments in communications and technology, data protection has established itself as one of the biggest concerns. The data must be encrypted in order to link securely, quickly through web-based technological data transmission. Transforming plain text into ciphered text that cannot be read or changed by malicious people is the process of encryption.

Materials and Methods:

In order to maintain the required degree of security, both the cryptanalysis and decryption operations took a significant amount of time. However, in order to cut down on the amount of time required for the encryption and decryption operations to be completed, several researchers implemented the cryptography method in a parallel fashion. The research that has been done on the problem has uncovered several potential answers. Researchers used parallelism to improve the throughput of their algorithms, which allowed them to achieve higher performance levels on the encryption algorithm.

Results:

Recent research on parallel encryption techniques has shown that graphics processing units (GPUs) perform better than other parallel platforms when comparing their levels of encryption performance.

Conclusion:

To carry out comparison research on the most significant parallel crypto algorithms in terms of data security efficacy, key length, cost, and speed, among other things. This paper reviews various significant parallel algorithms used for data encryption and decryption in all disciplines. However, other criteria must be considered in order to show the trustworthiness of any encryption. Randomness tests are very important to discover and are highlighted in this study.

Key words:

Information Security, Cryptography, Symmetric key encryption, Lightweight encryption, Parallel Encryption, Randomness test.

الخلاصة

مقدمة:

على الرغم من التطورات الهامة في الاتصالات والتكنولوجيا، فقد أثبتت حماية البيانات نفسها كواحدة من أكبر الاهتمامات. يجب تشفير البيانات من أجل الارتباط بشكل آمن وسريع من خلال نقل البيانات التكنولوجية على شبكة الإنترنت. يمكن تعريف عملية التشفير بانها تحويل النص العادي إلى نص مشفر لا يمكن قراءته أو تغييره بواسطة الأشخاص المؤذيين.

طرق العمل:

من أجل الحفاظ على الدرجة المطلوبة من الأمان ، استغرقت كل من عمليات تحليل التشفير وفك التشفير وقتًا طويلاً. ومع ذلك, من أجل تقليل مقدار الوقت المطلوب لإكمال عمليات التشفير وفك التشفير، طبق العديد من الباحثين طريقة التشفير بطريقة موازية. لقد كشف البحث الذي تم إجراؤه حول المشكلة عن العديد من الإجابات المحتملة. استخدم الباحثون التوازي لتحسين إنتاجية خوارزمياتهم، مما سمح لهم بتحقيق مستويات أداء أعلى في خوارزمية التشفير.

النتائج:

أظهرت الأبحاث الحديثة حول تقنيات التشفير المتوازي أن وحدات معالجة الرسومات (GPUs) تعمل بشكل أفضل من الأنظمة الأساسية المتوازية الأخرى عند مقارنة مستويات أداء التشفير.

الاستنتاجات:

لإجراء بحث مقارنة حول أهم خوارزميات التشفير المتوازية من حيث فعالية أمن البيانات وطول المفتاح والتكلفة والسرعة، من بين أمور أخرى. تستعرض هذه الورقة العديد من الخوارزميات المتوازية الهامة المستخدمة في تشفير البيانات وفك تشفيرها في جميع التخصصات. ومع ذلك، يجب النظر في معايير أخرى لإظهار مصداقية أي تشفير. تعتبر اختبارات العشوائية مهمة جدًا لاكتشافها وتم تسليط الضوء عليها في هذه الدراسة.

الكلمات المفتاحية

أمن المعلومات، التشفير، التشفير المتماثل، التشفير الخفيف، التشفير المتوازي، اختبار العشوائية.

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | ISSN: 2312-8135 | Print ISSN: 1992-0652

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

1. Introduction

Privacy and confidentiality, stability, and origin identification are just a few of the security services that have become the most crucial armor for defending against a variety of attacks [1]. Data encryption solutions are often used to provide these security agencies with the tools they need to overcome and limit such risks. Cyberattacks compromise the system's authentication (resource, customer and machine), stability, and reliability, whereas passive assaults severely compromise the device's confidentiality of data (DC) and anonymity. Furthermore, passive assaults are significantly harder to identify than popular ones because of their nature. An active hacker is capable of adding, removing, or changing data files[2]. Cryptography is a better solution for providing the necessary protection against data intruders. One of the most basic forms of computer safety is cryptography, which uses encryption and decryption techniques to convert data from its original form to one that cannot be read. A variety of cryptographic techniques are being developed to ensure secure communication. There are basically two types of cryptography: symmetric and asymmetric [3].

In symmetric key cryptography, the encryption and decryption codes are identical, and the cryptographic key is generated from the encryption function. It may be split into two categories, namely, block ciphers and stream ciphers. Stream ciphers, on the other hand, encrypt one bit of text at a time, whereas block ciphers require multiple keys, most recently 64 bits, and only encode as a single entity. Additionally, SSL/TLS over the internet employs the AES-256 cipher [4].

The term "asymmetric key cryptography" refers to a cryptographic algorithm that requires two unique keys, one of which is concealed and the other is public. Despite the fact that they are not the same, they are mathematically related. While the private key is used to decode the cipher text, the public key is used to encrypt plain text. When encoding large amounts of information, the asymmetric enciphering procedures used in become impractical because they are approximately 1,000 times slower than symmetric encoding. Additionally, asymmetric algorithms typically require stronger keys than symmetric enciphering steps in order to have comparable security power to the symmetric approach. Figure 1 provides an illustration of the main encoding technique category [5].

In order to satisfy growing demands when networked, where severely restricted software and hardware equipment are anticipated to grow, the topic of lightweight cryptography is expanding quickly. The creation of lightweight cryptography has greatly evolved during the last numerous decades in the global cryptography industry. Some mechanisms (such as an ASIC, FPGA, microcontroller, or microprocessor), a system will frequently outperform its predecessors. On a range of hardware and software systems, lightweight cryptology should be light. The U.S. National Security Administration (NSA) recently released Simon and Speck, two straightforward, lightweight symmetric cryptographic variants that both perform well in both software and hardware **[6],[7]**.

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

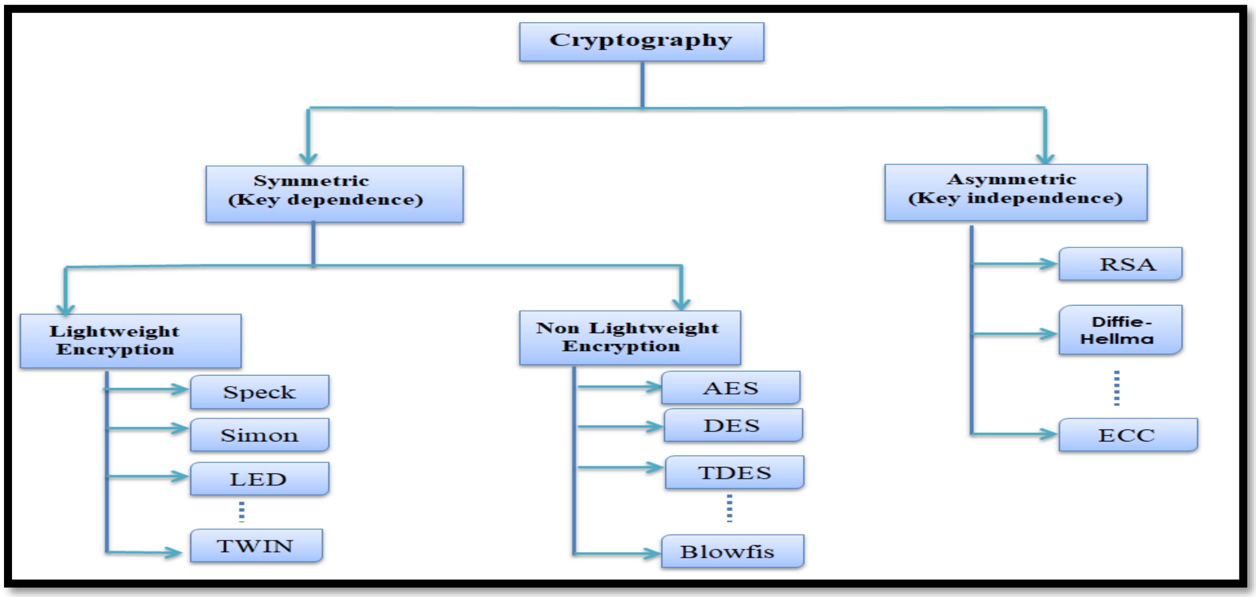

Figure 1 shows the many classifications that may be used in encryption algorithms.

2. Basic concept of symmetric cipher

In a symmetric-key cipher, the same key is used for both encryption and decryption. The symmetric-key cipher encryption function is an objective function that converts the input plaintext message to cipher text. The plaintext message is located in finite message space M, whereas the cipher text message is located in finite cipher text message space C. The keyed encryption function may be written as follows:

$$c = E_e(m); m \in M, e \in K_e \; .... \; (1)$$

Where e represents the encryption key, m represents the input plaintext message, and c represents the output cipher text message. The key e is chosen from the available key space Ke and governs how the function Ek converts plaintext communications to cipher text messages. Similarly, the decryption function is an objective function that converts the input cipher text to a plaintext message. The decryption function may be represented as follows:

$$m = D_d(c); c \in C, d \in K_d \; .... \; (2)$$

Where d represents the decryption key, c represents the input cipher text message, and m represents the output plaintext message. The decryption function with the appropriate decryption key d is the inverse of the encryption function with the appropriate encryption key e.

$$m = D_d(c) \; = E - 1(c). K_d \, K_e .... \; (3)$$

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

Table (1): Explain some security primitives used in cryptograph

Term	Explain
Key generation	It is the method of creating key using procedures. To encrypt and decode data, information, and communication, keys are employed. A tool or software that creates keys is known as a key generator, or key gen.
Rounds implementation	It is specified by each cryptosystem and normally entails a variety of building blocks assembled to produce a function that is executed repeatedly. The cipher text includes a variety of processing actions that include substitution, transposition, and mixing of the input plaintext to transform it into the final output of cipher text.
replacement-box	It is a fundamental substitution-based part of symmetric cryptography. They are frequently employed in block ciphers to ensure Shannon's principle of surprise by masking the connection between the keys and the encrypted data.
Permutation box (p-box)	It is a bit-shuffling process often used to transfer or process complete bytes over parameters in the p-box while staying with blurring. P-boxes are used to make the relationship between the plaintext and the cipher text difficult to understand.

The cryptographic or cipher is determined by combine these appropriate key with encryption and decryption functions. The encryption key in a symmetric-key cipher may simply be turned into the decryption key and vice versa. The encryption and decryption keys are usually the same, e = d, which explains the usage of the term "symmetric key." There are various types of symmetric-key encryption systems, such as block ciphers and stream ciphers. Block ciphers encrypt fixed-size plaintext messages into equal-size cipher text messages. In contrast, stream ciphers encrypt communications one bit or byte at a time. Stream ciphers often create a pseudorandom key stream depending on the encryption key being used, which is then "XORed" with the encrypted text message and the original text message [8]. The basic concepts of cryptography will be based on some security primitives that are used in cryptography, as shown in Table 1.

2-1 Symmetric cipher

In the symmetrical encryption major approaches has the same architecture which is utilized for cryptography and decoding. This key is known as a secret key. Both sides are aware of the secret key (the sender and the receiver). AES, DES, 3DES, Blowfish, IDEA, RC4, and TEA are examples of symmetric algorithms. Symmetric ciphers are still commonly used, particularly for data encryption, decryption, and message integrity checks. There are two kinds of symmetric encryption algorithms: block ciphers and stream ciphers [9], as shown in the figure 2. Symmetric cryptography is quicker than asymmetric cryptography and has a lower computational cost than public key cryptography since the keys used are significantly shorter. Symmetric cryptosystems also employ password authentication to confirm the receiver's identity. Despite these benefits, symmetric encryption has significant drawbacks, which are as follows: The degree of security is lower than with asymmetric encryption, and key transit issues exist. The secret key must be

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

provided to the receiving system before the actual message is delivered, but it is not possible to produce digital signatures that cannot be revoked.

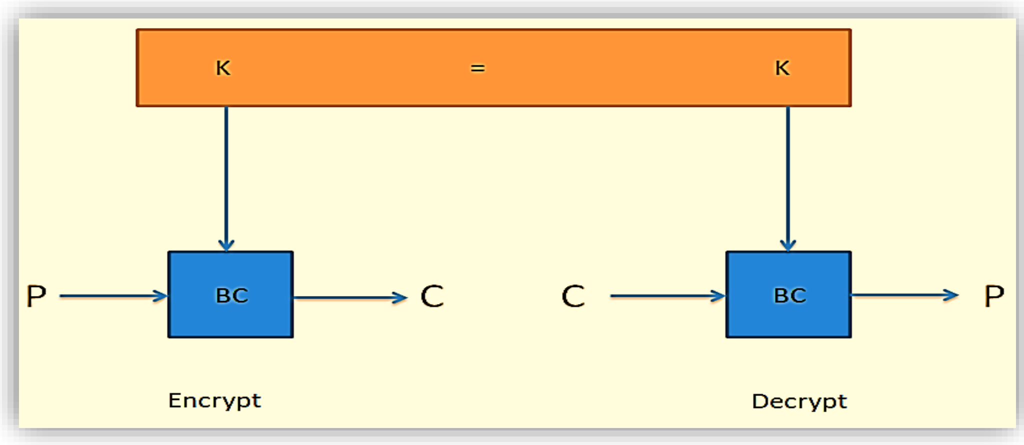

Figure 2: Symmetric encryption

2-2 Asymmetric cipher

Asymmetric encryption methods, often known as "public key" algorithms, use two keys: the public key, which is accessible to everyone, and the private key, which is generally a secret known only to the owner. These two keys are used in conjunction. A person who possesses one of these keys, however, cannot produce the other since it is mathematically impossible. Asymmetric encryption techniques are slower and have a higher computational cost than symmetric algorithms, but they are more secure and difficult to decrypt. Their performance, however, is quite low when compared to symmetric algorithms. In asymmetric algorithms, each participant has a key pair. A person's private key is just for his personal use and should not be shared with anyone else. The encrypted communication can only be opened by the receiver with his private key. Elgamal, RSA, ECC, Diffie-Hellman, and DSA are examples of public-key cryptography algorithms [9] , as shown in the figure (3).

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | ISSN: 2312-8135 | Print ISSN: 1992-0652

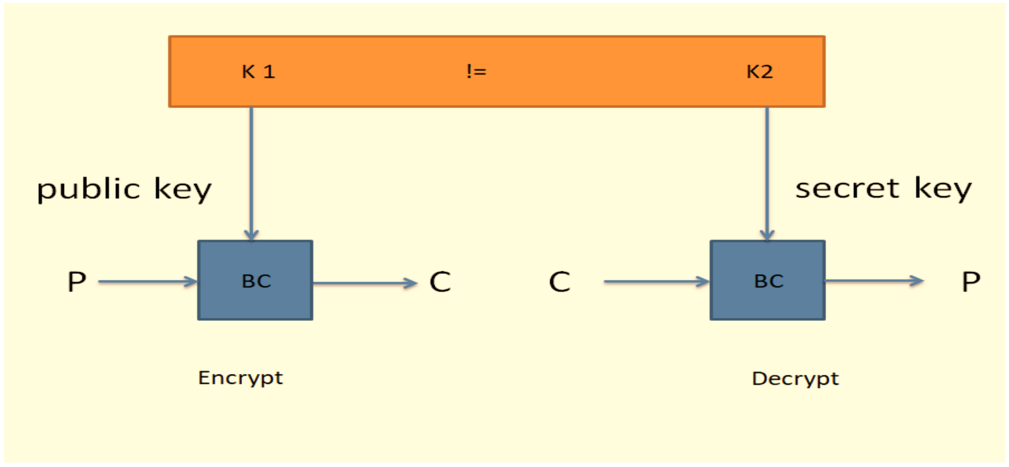

Figure 3 : Asymmetric encryption

3. Related works

In this section, we show some related works that concern some encryption methods executed on different parallel platforms with the goal of increasing throughput. Throughput means manipulating bytes per second to maximize efficiency, with the main goal of increasing the speed of previous encryption methods while keeping the security level as low as possible. Researchers in **[10]** introduced, utilized the dynamic resource technique for a new encryption named "ORSCA" that only needed one round. The suggested cryptosystem was created taking the GPU's peculiarities into consideration. This work included a key stream with one round that is suitable for large-scale applications**.** Productivity results show that it can process more data than other approaches, with a capacity of about 5 terabits per second on a Tesla A100 GPU**.** The presented cryptography surpasses the strongest GPU copies of AES, Simons, and Speck, which makes it more suitable for real-world applications. Our work in [11] presented AES, the Advanced Encryption Standard, which is one of the most commonly used strategies for secure private key exchange in sensing devices. The replacement lookup box list, on the other hand, is a time-consuming procedure in AES. As a result, their research aims to increase AES's running time and throughput by presenting a unique way of partitioning the Cryptography S-box and then performing simultaneous multithreaded replacement of two bytes at once. Other encryption methods utilized in sensor networks, including RC5, Blowfish, and Skipjack, have been compared to their proposed Enhanced AES (EAES) algorithm. When compared to the original AES method, the EAES algorithm performs better against a variety of lengths of the base processes, key lengths, and rounding. As a result, the proposed augmented EAES algorithm enhances the WSN's network lifetime and throughput. Another research paper [12] has proposed a SIMON-based lightweight cryptography algorithm for use in a system powered by the IoT. The emphasis is on speed improvement through the program, which distinguishes it from other experiments that primarily reflect equipment configurations. Furthermore, it implies extra effort toward the fundamental SIMON cryptology in order to accelerate encryption while maintaining a realistic balance of security and efficiency. The proposed work was compared to the Cryptography Standards (AES) and the basic SIMON cryptosystem schemes in terms of execution time and memory consumption. Researchers in [13],[14] presented FPGAs are a type of reconfigurable digital circuits that can be

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

used to implement various encryption algorithms, including the Advanced Encryption Standard (AES). Pipelined AES encryption systems are designed to process multiple blocks of data in parallel, while parallel AES encryption systems are designed to process different parts of a single block of data in parallel. Another author in [15],[16] has focused on hardware and lightweight cryptography, and other authors have implemented AES, SIMON, SPECK, PRESENT, LED, and TWINE, which are six block ciphers that may be implemented both in software and hardware utilizing the proprietary programmable microprocessor. These architectures may be directly compared in terms of bandwidth, location, throughput-to-area (TP/A), voltage, and cost since they are implemented on similar Xilinx Kintex-7 FPGAs. Our work in [17] is presented the parallel long messages Encryption Scheme (PLMES) is a cryptographic technique that is used to encrypt large data sets or long messages in parallel. The idea behind PLMES is to divide a long message into multiple smaller blocks, encrypt each block independently in parallel, and then concatenate the encrypted blocks to form the final encrypted message. The authors of [18],[19] proposed Speck-R, a new Speck version. Speck has been one of the most successful lightweight cryptography approaches. Dynamically replacing layers based on adaptive keys are added to Speck-R. A dynamic key is used to build the S-boxes, which are then modified according to the number of iterations. The main difference between the fixed renditions of the earlier suggestions for a decreased Speck and the intended Speck-R is this. They used the 96-bit Speck version, CTR mode, and 64-bit block. The main accomplishment of the study is the reduction of the Speck round count from 26 to 7, while maintaining a high level of security. Execution times will be shortened by lowering the number of repetitions. Another study in [20] proposed offering the blowfish method's performance assessment in a parallel environment. The experiments are carried out on the IMAN1 machine, and the method is constructed using the MPI standard. The experimental findings demonstrate that as the number of processors grows, the blowfish system's run time falls and its speed rises. For a plaintext length of 160 MB, it performs optimally with 32 processors. Simultaneous effectiveness performs best with 2, 4, or 8 cpus, and it reaches up to 99%, 98%, and 66%, respectively, with 16, 32, 64, or 128 microprocessors. Another researcher in [2] has proposed "ESSENCE," a lightweight stream cipher scheme based on a dynamic key approach that combines two different pseudo-random number generators (PRNGs). The technique delivers a high level of protection with less latency and necessary support when compared to current encryption standards like AES. Furthermore, on the GPU, the recommended flexible key-dependent cipher method outperforms all currently available AES implementations. With a capacity of more than 115 Gigabyte on a Tiger X Processor and far more than 372 Gigabyte on a Tiger V100 NVidia, the indicated encryption is quite effective. Because of its high level of unpredictability (the Big Crushing of TestU01), frequency, and key responsiveness, ESSENCE is a viable stream cipher option.

4. Overview of the existing ciphers' randomness tests

In this section, we present some statistical randomization tests that agree or disagree with the randomness of the encrypted output generated by some cryptographic algorithms. These random tests are essential to the crypt's ability to withstand cryptographic analysis and attacks. Testing cryptographic keychain randomness is a valuable and critical activity to determine the quality of implementation. These tests may also determine whether the generated key streaming meets the necessary randomized and consistency requirements. Randomness is the result of a probabilistic process that generates independent, uniformly dispersed, unexpected qualities that are impossible

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

to consistently repeat. Random sequences, on the other hand, might have distinct statistical qualities that can be assessed using various statistical methods. TestU01, Practrand, Diehard, ENT, Die Harder, NIST STS, and others are among these tools [21]. These random tests are described as follows:

- **PractRand (Practically Random):** is a written package in C that contains RNGs (pseudo-randomly generated) and statistical measures for RNGs. Practrand is the only test suite that supports tests with functionally limitless durations. It takes extra data to discover skew compared to all other test scripts. As well, if cross is available, the highest expedite is usually restricted to roughly 3x. Most development platforms come with random number generators that have major statistical flaws, are a bit slow, and/or have inconvenient interfaces.
- **TestU01** is an extensive software application that includes tools, a series of empirical tests taken from the research literature on RNGs, and examples of PRNGs. A collection of tools for statistical testing of homogeneous generators for random numbers is provided by TestU01, which is written in the computer language ANSI C (RNGs). Specific tests suites for either sequences of uniform random numbers in [0,1] or bit sequences are also available.
- **Gjrand** refers to the creation of pseudo-random variables for use in computer games, Monte-Carlo integration, simulations, and other applications. For such reasons, it is desirable that the statistical characteristics be quite excellent. Gjrand, however, cannot be used as a high-security random number generator for cryptography, the state lottery, or other similar applications. At the moment, gjrand offers a library for C programmers that have generators for uniform integers and normal distribution, a test suite for uniform bits, etc.
- **RaBiGeTe** may examine a random number or word producer to determine if it has the attributes of a real unique number. It works well on LCGs in general. It is more efficient per bit and has a graphical user interface (GUI) for seeing the distribution of results from several samples if required.
- **Dieharder** is a testing suite for random number generators (rng). It is meant to evaluate generators rather than files of potentially random numbers, since the latter represents a faulty understanding of what it means to be random. It goes beyond just cleaning up the steadfast tests and giving the native C code a lovely GPL source face. Both new tests created by RGB and tests from the Statistical Test Suite (STS) created by the National Organization of Standards and Techniques (NIST) are being used.
- **Diehard** are programming tests that evaluate the randomization of quasi-numeric generation as well as their readability and deformation. Their work yields a statistical p-value, which can be used to argue against an opinion. A sequence of 256-bit digits must be provided by the PRNG under test as an input for the tests (a secure minimum is several million integers).
- **NIST STS** every 60 seconds, the implementation produces full-entropy bit strings and uploads them in blocks of 512 bits. Each of these values contains a sequence number, a timestamp, a signature, and the preceding value's hash to link the series of data together and prohibit anybody, not even the source, from secretly changing an output package in the past. As a source of open randomness, NIST also manages the NIST Randomness Beacon. Multiple independent, publicly accessible sources of randomization are included in the service.
- **ENT** presents the results of different tests performed on byte sequences saved in files. The tool is helpful for testing compression techniques, pseudorandom number generators for use in statistical sampling and encryption, as well as other applications where a file's information density is important.

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

5 Comparative study of parallel encryption methods

A comparison of several measures highlights both symmetric algorithms' weaknesses and virtues. In the image below, a performance assessment takes into consideration parameters such as battery consumption, time consumption, block size, round, key size, rounds, throughput, security test type, structure, attack kinds, and hardware and software implementation. The method that is suited for particular implementation contexts may be determined based on the intended usage. Table 3 compares the security level of randomization test. Table 3, compares all of the existing parallel encryption techniques previously mentioned in this study. The goal of the comparison is to briefly describe all of the parallel encryption algorithms that have been evaluated and tested in this work using secure random tests (TestU01 and Practrand), where these tests agree or disagree on the randomness of the encrypted output in order to validate the proposal made. Correlation, probability density functions, and many other tests are used to determine the cipher text's unpredictability. It can ensure and maintain the required level of randomness in the ciphers responsible for protecting various types of exchanged data from various types of attacks. In addition, these algorithms were compared in Table 3 in terms of security level, based on some of the security basics used in encryption, in order to improve the encryption methods used in this research by increasing the level of security, increasing efficiency, and increasing productivity, In addition, discussions about this subject have shown that the size of the key has an impact on the total period of time as well as the quantity of battery that is used. The use of more complicated cryptographic procedures and keys has the potential to extend the amount of time required for an encryption method to accomplish its intended purpose. Yet, in order to build an encryption method that is both quick and safe, it is necessary to strike a balance between the number of security primitives that are needed and the amount of processing power that is available. In order to maintain a high degree of security, some symmetric ciphers make use of several rounds. Nevertheless, the increase in the total number of rounds will also lead to an increase in the computing cost. The primary objective of this study is to present a cryptography that has a high level of security while also having minimal compute needs. Also, the suggested solution will be implemented in parallel so that its throughput may be increased. The results additionally show that the ORSCA approach is better in terms of flexibility, encryption performance, encryption speed, and vulnerability to assaults, making it the best choice overall. This makes the ORSCA method the best alternative. It is the most effective approach compared to others that have been tried.

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | ISSN: 2312-8135 | Print ISSN: 1992-0652

Table(2):Comparison of security level randomize test

Algorithm Name	Test name	failure points	Defends Against Attack	Key size
ORSCA	**TestU01, Practrand**	**Non failued**	**Statistical, differential, linear, brute force attacks**	**512 bits**
ESSENCE	**TestU01**	**Non failued**	**Statistical, differential, Known blain text, brute force attacks.**	**512 bits**
TDES	**TestU01, Practrand**	**Non failued**	**Brute force, known Plaintext attacks**	**168, 112, 56 Bits**
Blowfish	**TestU01, Practrand**	**Non failued**	**Side channel attack**	**32 – 448 bits**
Towfish	**TestU01, Practrand**	**Non failued**	**Impossible differential attack**	**128, 192, 256 Bits**
RC4	**TestU01**	**Practrand**	**Fluhrer Mantin, Shamir Attack**	**40-256 bits**
PRESENT	**TestU01, Practrand**	**Non failued**	**Truncated differential Side-channel attacks.**	**80, 128 bits**
TEA	**TestU01, Practrand**	**Non failued**	**Related key attack**	**128 bits**
TWINE	**TestU01, Practrand**	**Non failued**	**Meet-in-the middle Saturation Attack**	**80, 128 bits**
Speck-R	**Uniformity, Correlation, Entropy tests**	**Non failued**	**Statistical, differential, brute force attacks**	**64 bits**

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

Table(3):Comparison between parallel encryption techniques

Ref	Algorithm	Structure	Key Size	Block Size	Rounds	Security Level	HW& SW	architecture parallel	Throughput
[10]	ORSCA	Festiel Stream	512 bits	256 bits	One	High	Both	GPU	greater than 5 Terabits/s
[2]	SSENCE	Festiel Stream	512 bits	128 bits	One	High	Both	GPU	more than 115 GB/s
[11]	EAES	SPN	128,192, 256 bits	128 bits	10,12, 14	High	Both	Multi core processors	It increases with the number of processors
[14]	AES	SPN	128,192, 256 bits	128 bits	10,12, 14	High	Both	Multi processors platform, Pipeline	Increases with an increase in the degree of improvement 98%
[20]	Blowfish	FN	32-448 bits	64 bits	16,18	Excellent security	Hardware	Multi core processors	run time is decreased as the number of processors increases
[22]	RC4	FN	Variable	40-2048 bits	256	Enough secured	Both	Multi core processors	_
[22]	RC5	FN	128 bits	34,64,128 bits	1 to225	Good	Both	Multi core processors	_
[18,19]	Speck-R	SPN	96 bits	64 bits	26	High	Hardware	Multi core platform	304 µsec
[23,24]	Speck	FN	128 bits	128 bits	32	Good	Hardware	Multi core processors	Very high on ASCI and FPGA
[23]	Simon	FN	128 bits	128 bits	64	Good	Software	Multi core processors	Very high on ASCI and FPGA
[23]	PRESENT	SPN	80/128 Bits	64 bits	32	Good	Hardware	Multi core processors	Very high
[23]	TEA	FN	128 bits	64 bits	64	Security enhanced	Software	Multi core processors	Very high
[23]	TWINE	FN	80, 128 bits	64 bits	36	Enough secured	Hardware	Multi core processors	Very high

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

6. Discussion of Results

Table 2 presents a comparison of the results of the parallel encryption approaches that were described earlier in this study. Based on these findings, it is clear that the performance of these algorithms is much improved when they are implemented on graphics processing units (GPUs), as opposed to when they are implemented on other parallel platforms. The parallel encryption algorithms that were evaluated and examined for this research were also put through secure random tests. The findings of these tests are presented in Table 3, and they show that the presence of randomness has an impact on the level of protection offered by an encryption method.

7. Conclusion

This paper provides an overview of the most important cryptographic algorithms implemented in parallel. According to the state of the art, researchers investigate and analyze these cryptographic algorithms with the goal of improving the performance of existing cryptographic methods. Their results show that the techniques used can be used for real-time encryption. The features and types of randomness tests are also presented in this work. However, this study discusses numerous significant parallel techniques utilized in all domains of encryption and decryption. The main goal of this paper is to conduct a comparative study between them according to effectiveness and period, difficulty, key length, and data protection, among other things. However, the level of security is determined by the use of some critical precautionary measures. The presence of randomness affects the level of security in cryptography, as mentioned earlier in this research.

Acknowledgments:

The researchers would like to thank the University of Babylon, College of Science for Women, for their support of this work.

Conflict of interests

There are non-conflicts of interest.

References

[1] K. B. Logunleko, O. D. Adeniji, and A. Logunleko, “A Comparative Study of Symmetric Cryptography Mechanism on DES , AES and EB64 for Information Security A Comparative Study of Symmetric Cryptography Mechanism on DES , AES and EB64 for Information Security,” no. June, 2020.

[2] R. Couturier, H. N. Noura, and A. Chehab, “ESSENCE: GPU-based and dynamic key-dependent efficient stream cipher for multimedia contents,” *Multimed. Tools Appl.*, vol. 79, no. 19–20, pp. 13559–13579, 2020, doi: 10.1007/s11042-020-08613-2.

[3] V. Hemamalini, G. Zayaraz, V. Susmitha, M. Gayathri, and M. Dhanam, “A Survey on Elementary , Symmetric and Asymmetric Key Cryptographic Techniques,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 1, pp. 11–26, 2016, [Online]. Available: http://www.meacse.org/ijcar

[4] M. Farik, S. Pandey, and M. Farik, “Best Symmetric Key Encryption -A Review A Review on Cloud Comput ing Securit y Best Symmetric Key Encryption - A Review”.

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

[5] O. G. ABood and S. K. . Guirguis, "A Survey on Cryptography Algorithms," *Int. J. Sci. Res. Publ.*, vol. 8, no. 7, 2018, doi: 10.29322/ijsrp.8.7.2018.p7978.

[6] S. Surendran, A. Nassef, and B. D. Beheshti, "A survey of cryptographic algorithms for IoT devices," *2018 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2018*, pp. 1–8, 2018, doi: 10.1109/LISAT.2018.8378034.

[7] R. F. Atiyah and I. Al-mejibli, "Lightweight secure Approach for IOT Devices," vol. 13, no. 3, pp. 4069–4078, 2022.

[8] R. M. N. Aldahdooh and A. Y. Mahmoud, "Parallel Implementation and Analysis of Encryption Algorithms," no. April, 2018.

[9] E. Science, *Iconst est'21*. 2021.

[10] A. Fanfakh, H. Noura, and R. Couturier, "ORSCA-GPU: one round stream cipher algorithm for GPU implementation," *J. Supercomput.*, vol. 78, no. 9, pp. 11744–11767, 2022, doi: 10.1007/s11227-022-04335-4.

[11] M. Gupta and A. Sinha, "Enhanced-AES encryption mechanism with S-box splitting for wireless sensor networks," *Int. J. Inf. Technol.*, vol. 13, no. 3, pp. 933–941, 2021, doi: 10.1007/s41870-021-00626-w.

[12] N. Alassaf, A. Gutub, S. A. Parah, and M. Al Ghamdi, "Enhancing speed of SIMON: A light-weight-cryptographic algorithm for IoT applications," *Multimed. Tools Appl.*, vol. 78, no. 23, pp. 32633–32657, 2019, doi: 10.1007/s11042-018-6801-z.

[13] C. Arul Murugan, P. Karthigaikumar, and S. Sathya Priya, "FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-box techniques for compact applications," *Automatika*, vol. 61, no. 4, pp. 682–693, 2020, doi: 10.1080/00051144.2020.1816388.

[14] M. Nabil, A. A. M. Khalaf, and S. M. Hassan, "Design and implementation of pipelined and parallel AES encryption systems using FPGA," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 1, pp. 287–299, 2020, doi: 10.11591/ijeecs.v20.i1.pp287-299.

[15] W. El Hadj Youssef, A. Abdelli, F. Dridi, and M. Machhout, "Hardware implementation of secure lightweight cryptographic designs for IoT applications," *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8860598.

[16] I. K. Dutta, B. Ghosh, and M. Bayoumi, "Lightweight cryptography for internet of insecure things: A survey," *2019 IEEE 9th Annu. Comput. Commun. Work. Conf. CCWC 2019*, pp. 475–481, 2019, doi: 10.1109/CCWC.2019.8666557.

[17] X. Wu, Y. Han, M. Zhang, and S. Zhu, "Parallel Long Messages Encryption Scheme Based on Certificateless Cryptosystem for Big Data," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10726 LNCS, pp. 211–222, 2018, doi: 10.1007/978-3-319-75160-3_14.

[18] L. Sleem, "Design and implementation of lightweight and secure cryptographic algorithms for embedded devices Lama Sleem To cite this version : HAL Id : tel-03101356," 2021.

[19] L. Sleem and R. Couturier, "Speck-R: An ultra light-weight cryptographic scheme for Internet of Things," *Multimed. Tools Appl.*, vol. 80, no. 11, pp. 17067–17102, 2021, doi: 10.1007/s11042-020-09625-8.

[20] M. R. Asassfeh, M. Qatawneh, and F. M. Al Azzeh, "Performance evaluation of blowfish algorithm on supercomputer IMAN1," *Int. J. Comput. Networks Commun.*, vol. 10, no. 2, pp. 43–53, 2018, doi: 10.5121/ijcnc.2018.10205.

[21] L. Sleem and R. Couturier, "TestU01 and Practrand: Tools for a randomness evaluation for famous multimedia ciphers," *Multimed. Tools Appl.*, vol. 79, no. 33–34, pp. 24075–24088, 2020, doi: 10.1007/s11042-020-09108-w.

[22] M. Agrawal and P. Mishra, "A comparative survey on symmetric key encryption techniques,"

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

Intern. J. Comput. Sci. Eng., vol. 4, no. 5, pp. 877–882, 2012, [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=82397469&site=ehost-live.

[23] J.Jebrane and S.Lazaar, "A performance comparison of lightweight cryptographic algorithms suitable for IoT transmissions,". General Letters in Mathematics, 10(2), 46–53. Hendi, A. Y., Dwairi, M. O., Al-Qadi, Z. A., & Soliman, M. S. (2019). A novel simple and highly secure method for data encryption-decryption. International Journal of Communication Networks and Information Security, 11(1), 232–238. https://doi.org/10.17762/ijcnis.v11i1.3999.

[24] A. Project, "Performance Analysis of the Speck Cryptography Algorithm Supervisors Certification I certify that this Project entitled ' Performance Analysis of the Speck cryptography algorithm ' was done by (Dunay Abd Al- Hussain," 2022.

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية