Secret Keys Extraction Using Light Weight Schemes for Data Ciphering

Sahar Najah Hussein¹, Saif Mahmood Al-Alak²

¹² Department of Computer Science, College of Science for Women, University of Babylon, Hillah, Iraq.
E-mail¹: asssdali44@gmail.com
E-mail²: saif.mahmood@uobabylon.edu.iq

Abstract: One of the most essential methods used to provide security services is encryption. One key is used for encoding in symmetric encryption. The symmetric encryption depends on the encryption block, switching, and replacing. Therefore, it is a problem if the received secret keys from protocols are frequent in some states or they have less randomness. In this paper, a Light Weight Multiple Key Generating (LWM) is proposed to generate the secret keys which are using Light Weight Schemes (LWS). In this work, six experiments are implemented. Three LWA are utilized, which are Xtea, RC5, and Tea algorithms. The SHA2 hash function is used to merge the chains. The diehard test is used in all experiments to determine the randomness of the secret key produced. The entropy is a measure of the uncertainty of a random variable.

Keywords: XTEA, TEA, RC5, Randomness, Lightweight, SHA2, Hash function.

1. Introduction

Cryptography is the analysis of mathematical techniques that provide definite protection services for databases. It is the science of translating "plaintext" data into encrypted data "ciphertext". The latter is subsequently recovered as the original information when the decryption of such data occurs. Therefore, cryptography has been carried out by applying specific tools. Accordingly, in computer-based encryption, the ability to maintain secret encoded data don’t depend only upon algorithms but is also relied on secret information called a secret key. This parameter is a cryptographic operation input that should be used in conjunction with algorithms to generate the encrypted records requested.

Encryption is considered a cryptographic approach that ensures confidentiality by using different keys. As a result, there are two group entities namely "symmetric and asymmetric" encryptions [1]. The former depends on the encryption block, switching and replacing databases. However, asymmetric key encryption requires certain mathematical applications such as discrete logarithm, integer factorization, as well as elliptic curve discrete logarithm [2]. For both encryptions schemes, there are several advantages and drawbacks among which symmetric encryption is faster than asymmetric cryptosystem and uses less power, but it requires a mutual key exchange. Since asymmetric algorithms provide safe transitions
without requiring the exchange of keys, they are used in a wide range of implementations [3]. It is necessary for any organization to adopt a material protection strategy, referred to as safeguarding information and information systems. This potential security is acted against unauthorized access, abuse, and unidentified users along with malicious interruption of information or even other resources alteration and destruction; to afford the triad of security objectives which are “confidentiality, integrity, and availability”.

The security information triad help to guide cybersecurity policies. The confidentiality concept is to prevent the detection of the information by the attacker of systems, at which it is essential to keep the privacy of people whose system maintains their personal information. Another fundamental principle is termed integrity, which means the data cannot change or modify without authorization and secure during transmission. The third object “availability” guarantees that the information must be available to users when needed; aiming to stay available at all times [4]. Therefore, it is a problem that the received secret keys from protocols are frequent in some states or they have less randomness. The authors in [5] proposed a scheme that uses different asymmetric methods to generate a series of secret keys (RSA, ECC, ALJammal) with XOR & sha2. In [6] Seed-based Min–Max Composite (S2MC) is proposed, which takes advantage of the properties of random pre-distribution while requiring less memory and being more resilient. The S2MC scheme is based on the organization of secret material that uses a small number of lightweight pre-distributed seeds in a specific range to generate keys during communication. In [7], to increase network lifetime while preserving WSN protection, a Secret Key Generation (SKG) protocol is proposed. In [8] PHYSEC, which uses channel reciprocity to produce a secret key and is also known as secret key generation (SKG) schemes, was investigated by the authors. centered on efforts to create a simple SKG scheme by removing the information reconciliation stage to lower the high computational and communication costs. In [1] ULEA is a new ultra-lightweight encryption algorithm proposed by the developers. ULEA assumes minor encryption rounds with simpler transformations and functions to complex the cipher, in addition to data diffusion and uncertainty. The security analysis and experimental results indicate that the ULEA algorithm is a suitable, low-storage-space, energy-efficient, and reliable encryption method for WSNs. The researchers tried to provide security by generating chain secret keys; however, the randomness of the keys produced is low, so this paper proposes a Light Weight Multiple Key Generating (LWM) method to improve randomness and improve the system's confidentiality. The Diehard test is used to evaluate the key chains that are produced through six experiments. The objective of this research study is to improving security through increasing randomness of generated secret keys by using an LWM (Light Weight Method) scheme. The following is the rest of the paper: In Section 1, cryptography is explained. Section 2 explains the lightweight algorithms. Section 3 explains the proposed work. Section 4 shows the Measures Metrics and the result. And, Section 5 concludes the paper.

2. The lightweight algorithms

Xtea
It is an Extended Tiny Encryption Algorithm. It's a plain, light-weight cryptographic algorithm with small code size. As a result, it is appropriate for small applications. XTEA has a data size of 128 bits and a block size of 64 bits. For N rounds, it also used the Feistel network. The number N is usually 32. XTEA's operations are as follows: i) Exclusive OR ii) Additions & iii) Shifts [9]. The extended TEA (XTEA) has a basic Feistel structure and is the fastest and most powerful light-weight Cryptographic Block Cipher (LCBC) with 64-bit block data and 128-bit key size and 64 rounds of operations. It is used in the majority of real-time cryptographic applications [10]. The key schedule is clear, and keys for the rounds are dynamically scheduled to minimize memory consumption, which is suitable for microcontrollers with limited memory [11]. When combined with key shuffling operations, the XTEA encryption algorithm was developed as an extension of the original TEA by the same scholars, it was described as a valuable and advanced alternative for enhancing security. Although XTEA is widely regarded as one of the most significant lightweight algorithms, it suffers from a low-round security vulnerability, and high-security applications should be able to handle 32 rounds. By splitting a 64-bit block into two 32-bit halves, v0 and v1, and feeding them into a 32-round algorithmic routine (N = 32), the XTEA encrypts data. In XTEA, the main scheduling has been modified to represent various trends for continuously mixing the data and key during each round, which introduces a lot of uncertainty. Modulo $2^{32}$ is used for easy addition and subtraction, and there are only four subkeys, each with a 32-bit length. The logic shifts are a logical left shift by four and a logical right shift by five, in addition to a simple 32-bit XOR logic operation. The permutation functions are written as $f(x) = (x_4 \ll 5 \oplus x_5) + x$, while the subkey generation functions are written as $sum + key\{sum \land 3\}$ and $sum + key\{sum \land 11 \land 3\}$ respectively. Sum acts as a selector between the four subkeys k0, k1, k2, and k3, which are determined by the sum's bits 0 and 1 or bits 11 and 12. $v_0$ and $v_1$ are XORed and ADDed with the result of the permutation function and the generated subkey. It's worth remembering that before the calculation starts, the value of the number is set to zero, and the value of delta is set to 0x9E3779B9. Table 1 shows both encryption and decryption pseudocode [12].

### Table 1. Pseudocode for XTEA encryption and decryption.

| Algorithm 1: pseudo code XTEA Encryption | Algorithm 2: pseudo code XTEA Decryption |
|------------------------------------------|------------------------------------------|
| Sum=0; delta=0x9E3779B9                  | Sum=0; delta=0x9E3779B9 |
| For i=0 to N do                         | For i=0 to N do |
| $v_0=$((v1<<4) xor (v1>>5) +v1) xor (sum+key {sum & 3}) | $v_1=$((v0<<4) xor (v1>>5) +v0) xor (sum+key {sum >> 11 & 3}) |
| sum+=delta                             | sum+=delta |
| $v_1=$((v0<<4) xor (v0>>5) +v0) (sum+key {sum >> 11 & 3}) | $v_0=$((v1<<4) xor (v1>>5) +v1) (sum+key {sum & 3}) |
| end for                                 | end for |

**TEA**

TEA is a good choice for cryptography because it is a block cipher that can handle only a few lines of code. In a lightweight cryptographic environment, reducing space and time complexity is desirable, which is why TEA uses 128-bit keys and works in 32 rounds, with a block size of 64 bits [9,13,14]. Since TEA uses the same key for both encryption and decryption, the algorithm's security is dependent on the key. All
can encrypt and decrypt the file once the key has been revealed [15]. The main goal of TEA is to use less memory and run faster [16]. In each cycle, the key materials are mixed in the same way. Various golden constants are inserted in every round based on the round's symmetry, which aids in the avoidance of various attacks [17]. The ease of implementation, lack of S-Box and P-Box, and high speed are the most notable features of the TEA [18].

RC5
One of the symmetric key algorithms is the RC5 (Rivest Cipher 5) algorithm. This algorithm is used in several application environments, such as image encryption and ad hoc networks, and its simple structure makes it ideal for implementing modifications. More than that, the algorithm uses basic logical operations and has a simple structure that makes it easy to understand. RC5 can be implemented with basic hardware and software, just like other symmetric key algorithms. The RC5 is derived from the RC4 algorithm and forms the basis for the actual RC6 algorithm. RC5 has a variable block size of 32, 64, and 128 bits, with a variable key length of between 0 and 2040 bits. Figure 1 illustrates the simple RC5 encryption method used to enhance the security level of this symmetric key algorithm without reducing its performance [19]. It has a variable block size, a variable number of rounds, and a variable-length secret key, allowing for performance versatility, security, and lightweight. The heavy use of data-based rotations in RC5 is a novel feature that encourages cryptographic strength evaluation [20].
3. Light Weight Multiple Key Generator (LWM)

The objective of this study is to improve security through increasing the randomness of generated secret keys by using an LWM (Light Weight Method) scheme. The proposed approach here in this study is generating the secret keys. At first, the sink should send four values to Node [initial value1, initial value2, secret key1, secret key2]. The sink and node generate keys from (key1, key2, ......., keyn) and the sink uses public key of Node to encrypt the four values then it sends them to node as shown in figure 2.

Figure 1. Basic RC5 encryption method.
In encryption, the original data is divided into blocks, the blocks are encrypted into cipher blocks by using generated secret keys. In decryption, the ciphered blocks are deciphered by generated keys. The encryption and decryption are shown in figure 3.

Figure 2. Sink generate a public key to the nodes.

Figure 3. The encryption and decryption.
The LWM method generates different key chains based on the initial values (v1, v2). On the left side, initial v1 with the first algorithm (LWM1), are used to get the first element of set list key (LS1). LS1 with the algorithm LWM1 are used to get (LS2), LS2 with the algorithm LWM1 are used to generate next element of the list and so on until it reaches LSn. On the right end, take Initial v2 with the second algorithm (LWM2), to get the right set key (RS1) take RS1 with the algorithm LWM2 to get (RS2), and take RS2 with the algorithm LWM2 and so on until it reaches RSn. Hush function between LS1 and RSn to get the first key k1 then a Hush function between LS2 and RSn-1 to get the second key k2 and it continues generating the keys and stops when the hush function works LSn and RS1 to get the key kn as shown in figure 4.

![Diagram of the suggested LWM method generates chains of keys.](image)

The LWM algorithms are used to generate the keys in each chain, as shown in Eqs. 1 and 2:

\[ \text{LS}[i+1] = \text{LWM1}(\text{LS}[i]) \quad \text{where} \quad 1 \leq i \leq n \quad \text{.........}(1) \]

\[ \text{RS}[i+1] = \text{LWM2}(\text{RS}[i]) \quad \text{where} \quad 1 \leq i \leq n \quad \text{.........}(2) \]

Where:

- LWM1 = The first algorithm, LWM2 = The second algorithm, LS = The left set key, RS = The right set key
As seen in the following, the keys are computed using the hash function ($f$).

\[
\text{Key}[j]= \text{Hash}_f(\text{LS}[j], \text{RS}[n-j+1]) \quad \text{where} \quad 1 < j < n
\]

$\text{Hash}_f$ represents the SHA2 hash function.

---

**LWM Algorithm**

**Input**: initial_v1, initial_v2, n: number_keys

**Output**: Key1, ...., Keyn

**Begin**

1. LS[0] = initial_v1
2. RS[0] = initial_v2
3. For \( i = 0 \) to \( n-1 \)
   4. \( \text{Begin} \_ \text{For} \)
   5. \( \text{LS}[i+1] = \text{LWM1}(\text{LS}[i]) \)
   6. \( \text{RS}[i+1] = \text{LWM2}(\text{RS}[i]) \)
   7. \( \text{End} \_ \text{For} \)
4. \( \text{For} \ j = 1 \) to \( n \)
   8. \( \text{Begin} \_ \text{For} \)
   9. \( \text{Key}[j] = \text{Hash}_f(\text{LS}[j], \text{RS}[n-j+1]) \)
   10. \( \text{End} \_ \text{For} \)
9. \( \text{End} \)

---

**4. Measured Metrics**

This section shows how to evaluate the proposed LWM approach using some main metrics.

**4.1 Randomness test**

The randomness of an algorithm's performance is one of the most crucial factors in determining its security. The diehard tests are carried out by the quest. Diehard is a collection of statistical tests created by Marsaglia to measure the quality of a random number generator. These include 15 Statistical tests. These tests use p-values to divide the set into three categories: safe, doubt, and failure. The following regions can be used to describe these areas: In the region of Doubt, there are 0.1 p-values 0.25 or 0.75 p > 0.9. 0 < p-value ≤0.1 or 0.9 ≤ p-value ≤ 1 in the failure area. In a safe area, a p-value of 0.25 to 0.75 [5].
4.2 Entropy test

Entropy is a measure of the uncertainty of a random variable. The entropy of a random variable is determined using the variable's probability distribution and is a good predictor of randomness or uncertainty. The entropy of a system increases as the randomness or disorder of the system increases, while the entropy decreases as the randomness or disorder of the system decrease.

4.3 Result

4.3.1 Randomness test

The test was carried out on the six LWM method experiments listed in Table 2.

| EXP. NO. | LWA1 | LWA2 | FUNCTION |
|----------|------|------|----------|
| 1        | tea  | xtea | teaxtea  |
| 2        | Xtea | RC5  | XteaRC5  |
| 3        | tea  | RC5  | teaRC5   |
| 4        | tea  | tea  | tea& tea |
| 5        | xtea | xtea | xtea&xtea|
| 6        | RC5  | RC5  | RC5&RC5  |

Table 2. The Lightweight Algorithms’ six experiments.

To measure the effectiveness and strength of the secret keys generated by the proposed method, the work calculates randomness. The first test of randomness is performed by using a Diehard statistical test, which consists of 15 tests. The result of statistical tests, called p-values are computed between [0,1) depending on distributing for the random variable. The area divided into three parts (safe, doubt, fail). The p-values in the fail area should diminish, and the p-values should increase in the safe area, to get better randomness and to increase the security. The Diehard statistical test is the first test used to calculate the randomness of the proposed LWM method to test whether provided high randomness or not to check the confidentiality of the proposed method. The experiments of the proposed LWM method are run to measure the randomness of the secret keys. In each experiment, the secret keys are generated and save in a separate file. All files used in the experiments are binary with size (15) MB. Failure in the area was the highest failure rate in algorithm teaxtea and lower failure rate in algorithm teaRC5 and the area of doubt was the highest rate of doubt in algorithm xtea & xtea and the lowest rate of doubt in algorithm RC5 & RC5 and the safe area, it was the highest safety in algorithm teaRC5 and the lowest rate of safety in algorithm teaxtea. Figures 5,6,7,8,9,10 depict the randomness of the six experiments:
Fig (5) the randomness of RC5 & RC5.

Fig (6) the randomness of tea & tea.

Fig (7) the randomness of xtea & xtea.

Fig (8) the randomness of teaxtea.

Fig (9) the randomness of xteaRc5.

Fig (10) the randomness of teaRc5.
4.3.2 Entropy

The measure is used to evaluate the quantity of randomness. The results were as a measure of entropy, where the highest rate was the algorithm xtea & xtea and the lowest rate was the algorithm xteaRc5. The entropy of the six experiments is illustrated in Table 3.

![Figure 11. The p-value of six experiments.](image-url)
Table 3. Entropy test of six experiments.

| Experiment      | Entropy Value   |
|-----------------|-----------------|
| RC5 & RC5       | 7.99998637498559 |
| tea & tea       | 7.99998866622257 |
| xtea & xtea     | 7.99998924397882 |
| teaxtea         | 7.99998868436613 |
| xteaRc5         | 7.9999857044095  |
| teaRc5          | 7.99998702829434 |

5. Conclusion

One of the most difficult problems is security. The LWM method has been proposed in this study to produce secret keys using a Light Weight Multiple Key Generating. These chains were generated by combining two algorithms with the Hash function (SHA2) and then Diehard Testing them. The results showed that using SHA2 increased randomness, suggesting that the process is safe and confidential.

References

[1] Hayouni, H. & Hamdi, M. A novel energy-efficient encryption algorithm for secure data in WSNs. J. Supercomput. (2020) doi:10.1007/s11227-020-03465-x.

[2] Karki, “Elliptic Curve Cryptography-A new approach to Public Key Cryptography,” Int. J. Eng. Sci. Res. Technol., vol. 3, no. 9, pp. 69–73, 2014.

[3] S. N. Panda, “A PROPORTIONAL ANALYSIS ON CRYPTOGRAPHY TECHNIQUES, FUNCTIONS, AND RELATIVE PERFORMANCE ISSUES,” Journal Glob. Res. Comput. Sci., vol. 2, no. 6, pp. 130–136, 2011.

[4] M. Mahfouz and A. Adjei-Quaye, “Information Security in an Organization,” Int. J. Comput., pp. 1–116, 2017.

[5] Hamza, A. H. & Al-Alak, C. Evaluation key generator of Multiple Asymmetric methods in Wireless Sensor Network (WSNs) Evaluation key generator of Multiple Asymmetric methods in Wireless Sensor Network (WSNs). (2021) doi:10.1088/1742-6596/1804/1/012096.

[6] Geetha, R., Madhusudhan, V., Padmavathy, T. & Lallithasree, A. A Light Weight Secure Communication Scheme for Wireless Sensor Networks. Wirel. Pers. Commun. 108, 1957–1976 (2019).

[7] Bashaa, M. H., Al-Alak, S. M. & Idrees, A. K. Secret key generation in wireless sensor network using public-key encryption. ACM Int. Conf. Proceeding Ser. 106–112 (2019) doi:10.1145/3321289.3321320.

[8] Yuliana, M., Wirawan & Suwadi. A simple secret key generation by using a combination of the pre-processing method with a multilevel quantization. Entropy 21, (2019).

[9] Bhatt, D. P., Raja, L. & Sharma, S. Light-weighted cryptographic algorithms for energy-efficient applications. J. Discret. Math. Sci. Cryptogr. 23, 643–650 (2020).

[10] Anusha, R. & Veena Devi Shastrimath, V. LCBC-XTEA: High Throughput Lightweight Cryptographic Block Cipher Model for Low-Cost RFID Systems. Advances in Intelligent Systems and Computing vol.
986 (Springer International Publishing, 2019).

[11] Maitra, S., Richards, D., Abdelgawad, A. & Yelamarthi, K. Performance Evaluation of IoT Encryption Algorithms: Memory, Timing, and Energy. SAS 2019 - 2019 IEEE Sensors Appl. Symp. Conf. Proc. 1–6 (2019) doi:10.1109/SAS.2019.8706017.

[12] Meer, M. H. Al. Programmable SoC for an XTEA Encryption Algorithm Using a Co-Design Environment Replication Performance Approach. J. Comput. Commun. 05, 40–59 (2017).

[13] Li, R. & Jin, C. Meet-in-the-middle attacks on reduced-round QARMa-64/128. Comput. J. 61, 1158–1165 (2018).

[14] Science, C. a Crypt Analysis of the Tiny Encryption Algorithm in Key Generation. Int. J. Commun. Comput. Technol. 1, 123–128 (2019).

[15] Rachmawati, D., Sharif, A., Jaysilen & Budiman, M. A. Hybrid Cryptosystem Using Tiny Encryption Algorithm and LUC Algorithm. IOP Conf. Ser. Mater. Sci. Eng. 300, (2018).

[16] Rajesh, S., Paul, V., Menon, V. G. & Khosravi, M. R. A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. Symmetry (Basel). 11, (2019).

[17] Chatterjee, Runa, and Rajdeep Chakraborty. "A Modified Lightweight PRESENT Cipher For IoT Security.” In 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1-6. IEEE, 2020.

[18] Novelan, M. S., Husein, A. M., Harahap, M. & Aisyah, S. SMS Security System on Mobile Devices Using Tiny Encryption Algorithm. J. Phys. Conf. Ser. 1007, (2018).

[19] Cevallos Salas, D. F. Enhanced RC5 Algorithm using Parallel Computing for Communication Networks. Ing. y Cienc. 15, 103–125 (2019).

[20] Raj, V. K., Ankitha, H., Ankitha, N. G. & Kanthi Hegde, L. S. Honey encryption based hybrid cryptographic algorithm: A fusion ensuring enhanced Security. Proc. 5th Int. Conf. Commun. Electron. Syst. ICCES 2020 490–494 (2020) doi:10.1109/ICCES48766.2020.09137849.

[21] H. Haran, N., & A.Yousif, Q. (2019). Fabrication of the Titanium dioxide nanowires on the FTO substrate and evaluating the efficiency power conversion in the dye-sensitized solar cell. Al-Qadisiyah Journal Of Pure Science, 24(2).