JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 183-195 (2023)
DOI: 10.6688/JISE.202301_39(1).0011

Perfect Confidentiality through Unconditionally Secure Homomorphic Encryption Using OTP With a Single Pre-Shared Key

MOSTEFA KARA[1], ABDELKADER LAOUID[1], AHCÈNE BOUNCEUR[2], MOHAMMAD HAMMOUDEH[3] AND MUATH ALSHAIKH[4]
[1]*LIAP Laboratory, University of El Oued, Algeria*
[2]*Lab-STICC UMR CNRS, University of Western Brittany UBO, Brest*
[3]*Information and Computer Science Department*
King Fahd University of Petroleum and Minerals
Dhahran, 31261 Saudi Arabia
[4]*Computer Science Department*
College of Computing and Informatics, Saudi Electronic University
11673 Riyadh, Kingdom of Saudi Arabia
E-mail: karamostefa@univ-eloued.dz; abdelkader-laouid@univ-eloued.dz; Ahcene.Bounceur@univ-brest.fr; mohammad.hammoudeh@kfupm.edu.sa; M.ALSHAIKH@seu.edu.sa

Unconditional security means that knowledge of an encrypted text does not provide any information about the corresponding plaintext; or more, regardless of the number of ciphertexts available to an attacker, no amount of cryptanalysis can break the cipher. Until now, only the One-Time Pad (OTP) method meets this condition with well-defined assumptions. The design of a Homomorphic Encryption scheme that allows operations over the encrypted data is required in current applications to reach the highest possible level of privacy. However, existing symmetric solutions that use OTP have a key management problem; they are not linear encryption, which means that they have high computational complexity, and some of them do not meet all homomorphic properties. This article simulates the OTP taking into consideration these issues and achieving the maximum resistance to cryptanalysis, even when the attacker has great computing power. The first major advantage of the proposed OTP-based method is that it only uses a single pre-shared key. The key is composed of two sections, a fixed number of bits followed by random bits; the size of each section is dependent on the robustness of the system. Analysis of the proposed technique shows that it provides perfect privacy by using a different key for each message to be encrypted.

Keywords: unconditional security, perfect privacy, homomorphic encryption, OTP, confidentiality, asymmetric OTP

1. INTRODUCTION

In 1949, Claude and Shannon [1] published their Theory of Communication Secret Systems which is considered the basis of cryptography. Cryptographic system security can be classified based on the level of security they provide into computationally secure cipher unconditionally secure cipher. The first is concerned with measuring the number of

Received October 1, 2021; revised December 23, 2021; accepted March 14, 2022.
Communicated by Changqiao Xu.

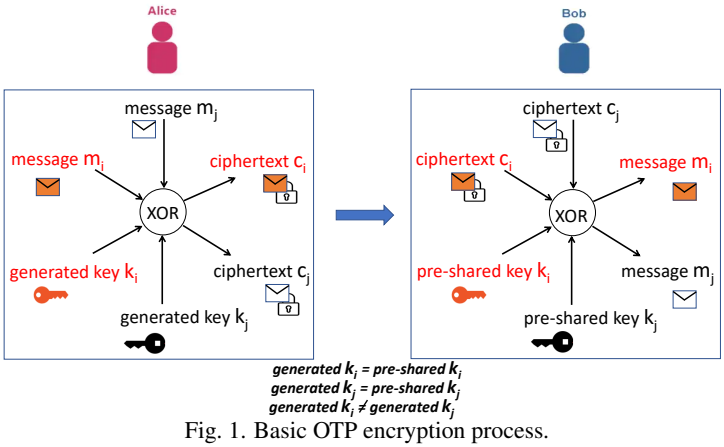

Fig. 1. Basic OTP encryption process.

computation operations necessary to break an encryption system. In complexity theory, a system is safe if it requires a large number of operations to break it where this number is not applicable in practice; therefore, the best-known attack cannot be successfully completed in a reasonable computing time, even with theoretically infinite computing power. The second notion is perfect secrecy [2], where an attacker cannot get any information about the plaintext if he only knows the ciphertext. From these two concepts, the importance of OTP is the only one that can achieve perfect privacy [3, 4] through unconditional security. The standard expands in the first use [5] is an encryption method that theoretically cannot be broken due to the use of a truly random and single-use of a secret key. In original OTP (Fig. 1), each plaintext is associated with a random secret key (single-use block) where the encryption consists of combining the plaintext with the key using a modular addition (XOR). Looking at the definition of basic OTP, we see that it is difficult to implement due to many obstacles, the most important of which are the issues of secure key distribution which make this method impractical for most applications. To solve these problems and put a practice OTP, we propose a Homomorphic Encryption method which uses a single key that is made from two parts, the first is fixed and the second is random. We consider that the first section is the real key that will be shared with the other communicating entity, the second key is generated randomly for the encryption of each message. In this way, we concatenated these two keys to build the system key. So, each message is encrypted using a different key. The core of the proposed method can be represented under the following equation:

$$c = m \times K \tag{1}$$

where c is the ciphertext, m is the plaintext, and K denotes the secret key that is used only once. It should also be noted that this encryption is as light as possible because it consists of a single multiplication operation. We randomly generate a secret key K for each message we want to encrypt.

The contributions of this paper can be summarized as follows:

- A linear symmetric and asymmetric schemes over integer using the OTP method with a single pre-shared secret key have been proposed.

- Realization of a homomorphic addition property for dynamic keys scheme.
- A study of the influence of a random r on the asymmetric encryption security with public keys of the form $pk = k + r \times p$, where k denotes the secret key and p denotes the trapdoor.

2. LITERATURE SURVEY

Cryptography is the only method that allows us to protect our privacy by protecting our sensitive information against attacks. In this section, we present some encryption schemes. Several asymmetric schemes have been proposed [6–9], these techniques use different keys for encryption and decryption. Unlike the asymmetric schemes where Alice shares the public key with anyone, OTP is symmetric encryption, also call- ed secret key encryption when Alice and Bob use the same key to both encrypt and decrypt operations. The OTP encryption appeared in 1917 by Major Joseph Mau- borgne as the improvement of the Vernam Cipher to achieve the perfect security [10].

In [11], a study of OTP encryption was presented and analysis of an application that implements a One-Time Pad (OTP) algorithm was presented. The authors performed tests on document formats doc, Excel file, an image file, and a PDF file. In [12], the OTP encryption was demonstrated by combining two properties of semiconductor lasers; their potential and their ability. To implement OTP, the authors of [13] combined full-phase image encryption and hiding. The plain images are encoded in the phase and encrypted by phase keys loaded by using quantum key distribution; after, producing reference wave and forming interferogram; finally, an encrypted image hiding is achieved by Phase-Shifting Interferometry (PSI). An OTP encryption technique that is based on DES conventional block cipher and MD5 one-way hash function was proposed in [14]. A dynamic key theory was presented and analyzed in [15]. To reduce the cryptanalysis attack risk and improve the security of cryptographic systems, these dynamic keys are one-time used symmetric cryptographic keys.

In [16], the authors proposed a symmetric key cryptography method using dynamic keys. This system performs four rounds of encryption and decryption. In each round, different parts of the dynamic key are used to make it hard against common attacks. To produce random numbers, the authors used Linear Congruential Generator (LCG), where they select a modulus m, a multiplier a, an additive term b, and an initial value y_0 and: $y_1 = (a \times y_0 + b) \mod m$ for every encryption and decryption operation, i.e., a new key y_i is calculated. The exchange of these primitives is performed via public key algorithms like modified Diffie-Hellman protocol [17]. To produce the ciphertext, a subkey of the size equal to 49 bits is applied (the same size as the plaintext block) using XOR operation, this gives a dynamic key of 196 bits. The disadvantage of this technique is that the dynamic keys produced each time are linked to each other and can be hacked by calculating some probabilities.

OTP encryption is a proven secure encryption method but it requires a high key generation rate in practical applications because each bit of key is needed for each bit of message to be encrypted. In this paper, we propose a new homomorphic OTP encryption as a dynamic keys scheme that uses a single shared secret key.

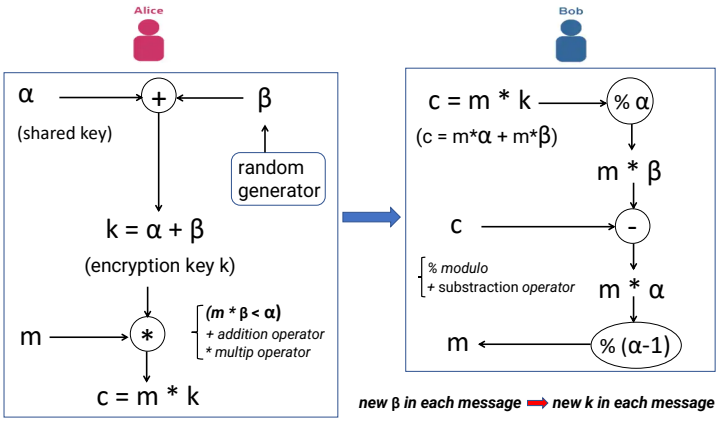

Fig. 2. Proposed OTP encryption process.

3. OTP ENCRYPTION DESIGN

Fig. 2 shows our proposed OTP process where Alice has a real key α which will be shared with Bob for use in the decryption process. For each message m to be encrypted, Alice generates another imaginary key β; finally, the encryption key k is the sum of the real key and the imaginary key. Note that k will be random in each message to be sent. After calculating the encryption key k, Alice computes the ciphertext c as $m \times k$, where m is the message to be encrypted. The decryption operation is also simple; Bob calculates $c \mod \alpha$ to remove the imaginary part ($m \times \beta$); only the real part ($m \times \alpha$) remains. Bob can retrieve the original message m by dividing on the shared key α.

Table 1 designates the used primitives and symbols.

Table 1. List of the used primitives and symbols.

Symbol	Designation
α and β	keys
m and c	plaintext and ciphertext respectively
+ and x	addition and multiplication in $\mathbb{Z}$
M	the plaintext space
C	the ciphertext space
K	the keys space
$E()$	the encryption function where $E_{\alpha,\beta} : M \longrightarrow C$ with $m \times \beta < \alpha$
$D()$	the decryption function where $D_{\alpha} : C \longrightarrow M$
n	a public key where $n = p \times q$ and p and q are two prime numbers
pk	a public key where $pk = \alpha + r \times p$ and r is a random
λ	the security parameter
l	$\|k\| = l$ bits
D	the number of additions

In $\mathbb{Z}$, the core of the proposed OTP can be defined as follow:

- KeyGen: Given one secret key (to be shared and used in decryption) and random keys (to be used only in encryption as an OTP), which are α and β, respectively; for any new message, KeyGen function generates a new β and give us a new encryption key k, $k = \alpha + \beta$ where $+$ is the addition operator.

- $E(m)$: The message m is encrypted using the secret key k, $c = m \times k$ where $\times$ is the multiplication operator; in more detail, $c_i = m_i \times k_i$ with $k_i = \alpha + \beta_i$ and $m_i \times \beta_i < \alpha$.

- $D(c)$: The ciphertext c is decrypted using the secret key α, $m = (c - (c \mod \alpha)) \mod (\alpha - 1)$, or $m = \dfrac{c - (c \mod \alpha)}{\alpha}$.

Keys generation Randomization is very useful and fundamental in all areas of computer science, such as approximation algorithms, counting problems, distributed computing, and most importantly cryptosystems; where randomization is important in cryptography, the keys must be generated randomly, which will allow us to make encryption probabilistic [18]. If Alice wants to send data to Bob through an insecure channel, Alice and Bob originally agree on a shared key k. Firstly, Alice uses k to compute the ciphertext c and sends it. Bob receives the ciphertext c' and computes $m' = D(c')$. So, if $c = c'$, then $m = m'$. To achieve privacy, the adversary does obtain no information about the sent data. Now, if k is a truly random $N - bit$ string, Alice and Bob can obtain perfect encryption of an $N - bit$ data by using the OTP method. On the contrary, if k is not truly random (coming from an imperfect source of randomness [19]), the encryption cannot be securely [20].

Algorithm 1 : KeyGen algorithm

Require: α
Ensure: k_s

```
function KEYGEN
    for each message do
        generate β
        k ← α + β
    end for
    return k_s
end function
```

Encryption To encrypt a message m, we use the following Eq. (2):

$$c = m \times k_s \quad with\ k\ is\ random \tag{2}$$

The encryption process could be defined by the following algorithm:

Algorithm 2 : Encryption algorithm

Require: m_i, k_i
Ensure: $c_i = E(m_i)$

```
    function E
2:      for each i do
            c_i ← m_i × k_i
4:      end for
        return c_i
6:  end function
```

Decryption To decrypt a ciphertext c, we use the following Eq. (3):

$$m = (c - (c \mod \alpha)) \mod (\alpha - 1) \tag{3}$$

The decryption process could be defined by the following algorithm:

Algorithm 3 : Decryption algorithm

Require: c_i, α
Ensure: $m_i = D(c_i)$

```
    function D
        for each i do
3:          m_i ← ((c_i − (c_i mod α)) / α)
        end for
        return m_i
6:  end function
```

4. SECURITY ANALYSIS

The OTP is the only known encryption method that is considered unconditionally secure. If an attacker obtains an encrypted message, he has no way of determining the original message or the corresponding key. If the key is random and of length equal to (or greater) that of the message to be encrypted, there is no information in the encrypted message (such as letter frequency) that the attacker can use to determine the real message or the key.

The principal rule of OTP cryptography is one should never use the same secret keys more than once. Otherwise, the encrypted message will be vulnerable to known ciphertext (ciphertext-only) attacks. The following example shows how the security of the OTP encryption is affected by using the same keys twice: $c = m \bigoplus k$ and $c' = m' \bigoplus k$. Having the two encrypted messages c and c', an attacker is able to break the encryption if he adds them together: $c \bigoplus c' = m \bigoplus k \bigoplus m' \bigoplus k = m \bigoplus m'$, the attacker is able to extract the original plaintexts: $m \bigoplus m' \Longrightarrow m, m'$.

Brute force attack If the attacker wants to apply a brute force attack (BFA), *i.e.*, decrypt the encrypted message with all possible keys, he would have no way of knowing which plaintext is the original plaintext. On the other hand, BFA will produce a large number of potential plaintext which all make sense to this attacker.

Known plaintext attack When the attacker have a plaintext and the corresponding ciphertext; the potential scenario is to extract the corresponding key k where: $k = \frac{c}{m}$, knowing that the one-time secret key $k = \alpha + \beta$ and β is random, it is impossible for the attacker to obtain the key decryption (real key β) because there are $2^{\frac{l}{2}}$ possibilities if $|k| = l$ bits. Assuming that $\alpha > \beta$ (condition), the attacker has $2^{\frac{l}{2}-1}$ possibilities for α; therefore, in our technique, it is impossible to find out the secret key α even if an opponent having a large number of plaintxts and their corresponding ciphertexts.

Chosen plain/cipher text If the attacker has obtained access to the encryption machinery. So, he can choose a plaintext m, and compute the corresponding ciphertext c. In chosen ciphertext, it is the contrary, the attacker has obtained access to the decryption machinery. Then, he can choose a ciphertext c, and compute the corresponding plaintext m. In these two attacks, no information will be discovered because the secret key is random; whatever the number of plain/cipher text generated and whatever combination the attacker has to make, he will always obtain new and different information on the key k.

5. SCHEME PERFORMANCE

If we look at OTP mathematically, it is indeed impossible to break unless the secret keys are discovered. Besides the problem of exchanging these keys, where they can be exposed to eavesdropping, there is a storage problem which leads to increased security and increased space; and so, what happens if the number of messages we send is very large? In basic OTP, the answer to this question is: encryption is not applicable. In the proposed method, we overcome these problems by making the encryption dependent on a single shared key, this key is exploited to produce new (random) encryption values in each message to be encrypted, while this shared key remains valid to decrypt all these randomly encrypted messages.

5.1 Additive Homomorphic Property

Homomorphism is a very important property where one can perform computational operations on encrypted data without having to decrypt it. The results of these operations give the same results as if they were performed on unencrypted data. By carrying out this equation:

$$Decryption(Encryption(m_1) + Encryption(m_2)) = m_1 + m_2. \tag{4}$$

This equation represents the realization of the homomorphic addition property which is provided by the proposed encryption.

Lemma 1. $D(E(m_1) + E(m_2)) = m_1 + m_2$.

Proof. $E(m_1) = m_1 \times k_1 = m_1 \times \alpha + m_1 \times \beta_1$,
$E(m_2) = m_2 \times k_2 = m_2 \times \alpha + m_2 \times \beta_2$,
$C = E(m_1) + E(m_2) = (m_1 + m_2) \times \alpha + (m_1 \times \beta_1 + m_2 \times \beta_2)$,
So, $D(C) = (C - (C \mod \alpha)) \mod (\alpha - 1) = (C - (m_1 \times \beta_1 + m_2 \times \beta_2)) \mod (\alpha - 1)$
$= m_1 + m_2$. Where

$$\sum_{i=1}^{D} m_i \times \beta_i < \alpha \tag{5}$$

with D is the number of additions (depth of operations). □

In this proof, we prove that an unreliable third party 'Eve' can perform a finite number of operations on Alice's encrypted data without knowing the real values of that data.

Lemma 2. *Posing D denotes the number of additions and $|k| = l = l_\alpha + l_\beta$, if $l_\alpha - l_\beta$ increases ($l_\beta \searrow$) then D inceases.*

Proof. We have $S_D = \sum_{i=1}^{D} E(m_i) = (\alpha \times \sum_{i=1}^{D} m_i) + (\sum_{i=1}^{D} m_i \times \beta_i)$, if $Max(m_i) = M \; \forall \; i$, then $S_D = D \times M \times \beta <= \alpha \Rightarrow D = \frac{\alpha}{M \times \beta}$; So, if β decreases, then D increases. □

On the other hand, if the value $l_\alpha - l_\beta$ is smaller, than the encryption is stronger.

Lemma 3. *Posing λ denotes the parameter security and $|k| = l = l_\alpha + l_\beta$, if $l_\alpha - l_\beta$ decreases ($l_\beta \nearrow$) then λ inceases.*

Proof. If λ is the security parameter, we can express it by l_β because the number of operations that an attacker must do to decrypt a message is effectively equal to l_β knowing that α is fixed. Consequently, if we increase the number of possibilities for the encryption keys k_i (notably the β_i), it is obvious that the encryption will be more secure. So,

$$\lambda = 2^{l_\beta}. \tag{6}$$

where λ is the security parameter and $|\beta| = l_\beta$. □

5.2 Computational Complexity

In the proposed technique, we have only one operation to do (according to Algorithm 2) whatever the length of the message to be encrypted; therefore the complexity $C(m+1)$ is equal to the complexity $C(m)$. The same in Algorithm 3, the number of operations does not change when m increases. Table 2 shows the performance of the proposed OTP in terms of complexity compared with other schemes under the same parameters (security level, key length).

5.3 Reduced Size

In the proposed OTP encryption method shown in Eq. (2), the size of ciphertext is a little small.

$$size(Enc(m)) = size(m) + size(k) \approx size(k) \tag{7}$$

Table 2. Encryption and decryption complexity comparison.

Scheme	E()	D()
proposd OTP	$O(1)$	$O(1)$
[21]	$O(\lambda^4)$	$O(\lambda^4)$
[22]	$O(\lambda^5)$	$O(\lambda^{4.8})$
[23]	$O(\lambda^6)$	$O(\lambda^5)$
[24]	$O(\lambda^6)$	$O(\lambda^5)$
[25]	$O(\lambda^{13})$	$O(\lambda^{12})$

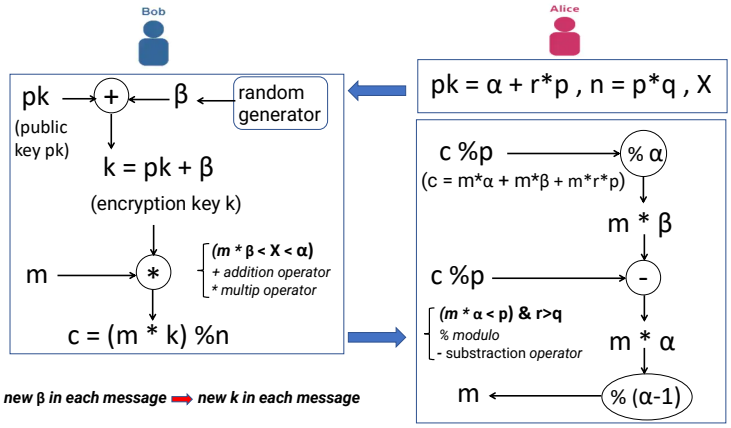

Fig. 3. Asymmetric proposed OTP encryption process.

It is true that the larger the size of the key, the stronger the encryption, but compared to techniques that use exponential, the size of the ciphertext in our technology is relatively small.

5.4 Asymmetric OTP

Fig. 3 shows that we have replaced the shared secret key α with the public key pk where $pk = \alpha + r \times p$. It is true that asymmetric encryption is weak against attacks (notably, it is vulnerable to Known-plaintext attack) compared to symmetric encryption, but in fact, this depends on the length of the primitive $n(n = p \times q)$ and its hardness. Where p and q are two large prime numbers of the form $2 \times h + 1$ and h is a prime number. r must satisfy the following two conditions:

1. $r > q$, if $r < q$ then $c = m \times k < n$. So, the modulo operation has no effect; *i.e.*, we did not hide the ciphertext c effectively.

2. We put $v = pk \mod n$, if r was not chosen in the right way, then $c \mod v = m \times \beta$, despite β is random but it can give to the attacker an information on the secret key α or on the trapdoor p . For example, if we set the following values: p = 1051, q = 401, α = 101, and m = 2; with $1 <= \beta < 10$ we will obtain the results of Table 3.

5.5 Implementation Results

For the encryption of data of 16 bits, Table 4 presents the execution results of the proposed method and the reduced time compared to other schemes. The smallest encryption time in cited schemes [6] is estimated to be ten times our encryption time. As for the fastest decryption operation [6], it is estimated to be over 1700 times. This is due to our lightweight technique.

We notice that the encryption time is equal to the decryption time because this two operations effectively contains three elementary processes; encryption consists of the generation of a random number (β), the sum ($k = \alpha + \beta$), and the multiplication ($c = m \times k$); decryption consists of modulo ($c \mod \alpha$), subtraction ($c - (c \mod \alpha)$), and division ($\frac{c-(c \mod \alpha)}{\alpha}$).

Table 3. The influence of the value of r on the encryption.

	$(c \mod v, \beta)$
$401 < r < 601$ ($r = 402$)	(2, 1); (4, 2); (6, 3); (8, 4); (10, 5)
$601 < r < 802$ ($r = 602$)	(1255, 1); (1257, 2); (1259, 3); (1261, 4); (1263, 5)

Table 4. Execution time (ms).

Scheme	E()	D()
proposed OTP	0.007	0.007
[6]	0.07	11.95
[26]	11.91	17.67
[27]	47	15
[28]	50	10
[29]	255	493
[30]	899	785

6. CONCLUSION

The OTP uses a randomly generated secret key of the same (or more) length as the data. To encrypt data m, it is combined with the random key k using the exclusive-OR operation bit-wise. To avoid OTP unconditional security issues in the original versions such as key storage, accessibility, and confidentiality, we presented a new OTP on $\mathbb{Z}$ using the natural operators in both symmetric and asymmetric versions, the proposed partial homomorphic OTP encryption only uses one secret (respectively, public) key. We have analyzed the proposed technique and studied its performance in the addition property, complexity, size, and execution time. Thanks to our practical OTP, we have obtained a complexity equal to $O(1)$ in both encryption and decryption; thus, we have obtained an execution time equal to 0.007 *ms* in the encryption and decryption operations.

REFERENCES

1. C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, Vol. 28, 1949, pp. 656-715.

2. A. Aloraini and M. Hammoudeh, "A survey on data confidentiality and privacy in cloud computing," in *Proceedings of International Conference on Future Networks and Distributed Systems*, 2017, pp. 1-7.
3. H. Egerton, M. Hammoudeh, D. Unal, and B. Adebisi, "Applying zero trust security principles to defence mechanisms against data exfiltration attacks," *Security and Privacy in the Internet of Things: Architectures, Techniques, and Applications*, 2021, pp. 57-89.
4. W. Rjaibi and M. Hammoudeh, "Enhancing and simplifying data security and privacy for multitiered applications," *Journal of Parallel and Distributed Computing*, Vol. 139, 2020, pp. 53-64.
5. G. S. Vernam, "Cipher printing telegraph systems: For secret wire and radio telegraphic communications," *Journal of the AIEE*, Vol. 45, 1926, pp. 109-115.
6. M. Kara, A. Laouid, R. Euler, M. A. Yagoub, A. Bounceur, M. Hammoudeh, and S. Medileh, "A homomorphic digit fragmentation encryption scheme based on the polynomial reconstruction problem," in *Proceedings of the 4th International Conference on Future Networks and Distributed Systems*, 2020, pp. 1-6.
7. A. Laouid, M. AlShaikh, F. Lalem, A. Bounceur, R. Euler, M. Bezoui, H. Aissaoua, and A. Tari, "A distributed security protocol designed for the context of internet of things," in *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, 2018, pp. 1-5.
8. M. Kara, A. Laouid, M. A. Yagoub, R. Euler, S. Medileh, M. Hammoudeh, A. Eleyan, and A. Bounceur, "A fully homomorphic encryption based on magic number fragmentation and el-gamal encryption: Smart healthcare use case," *Expert Systems*, Vol. 39, 2021, pp. 1-10.
9. S. Medileh, A. Laouid, E. Nagoudi, R. Euler, A. Bounceur, M. Hammoudeh, M. AlShaikh, A. Eleyan, and O. A. Khashan, "A flexible encryption technique for the internet of things environment," *Ad Hoc Networks*, Vol. 106, 2020, p. 102240.
10. R. Munir, "Algoritma enkripsi citra dengan pseudo one-time pad yang menggunakan sistem chaos," *Konferensi Nasional Informatika*, Vol. 4, 2011, pp. 12-16.
11. B. E. Purnama, "An analysis of encryption and decryption application by using one time pad algorithm," *International Journal of Advanced Computer Science and Applications*, Vol. 6, 2015, pp. 292-297.
12. A. Argyris, E. Pikasis, and D. Syvridis, "Gb/s one-time-pad data encryption with synchronized chaos-based true random bit generators," *Journal of Lightwave Technology*, Vol. 34, 2016, pp. 5325-5331.
13. J. Li, J. Xiong, Q. Zhang, L. Zhong, Y. Zhou, J. Li, and X. Lu, "A one-time pad encryption method combining full-phase image encryption and hiding," *Journal of Optics*, Vol. 19, 2017, p. 085701.
14. S. Tang and F. Liu, "A one-time pad encryption algorithm based on one-way hash and conventional block cipher," in *Proceedings of IEEE 2nd International Conference on Consumer Electronics, Communications and Networks*, 2012, pp. 72-74.
15. H. H. Ngo, X. Wu, P. D. Le, C. Wilson, and B. Srinivasan, "Dynamic key cryptography and applications," *International Journal of Network Security*, Vol. 10, 2010, pp. 161-174.

16. Z. Mahmood, J. Rana, and A. Khare, "Symmetric key cryptography using dynamic key and linear congruential generator (lcg)," *International Journal of Computer Applications*, Vol. 50, 2012, pp. 7-11.
17. M. Kara, A. Laouid, M. AlShaikh, A. Bounceur, and M. Hammoudeh, "Secure key exchange against man-in-the-middle attack: Modified diffie-hellman protocol," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, Vol. 7, 2021, pp. 380-387.
18. Y. Dodis and J. Spencer, "On the (non) universality of the one-time pad," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 376-385.
19. J. L. McInnes and B. Pinkas, "On the impossibility of private key cryptography with weakly random keys," in *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, 1990, pp. 421-435.
20. M. Santha and U. V. Vazirani, "Generating quasi-random sequences from semi-random sources," *Journal of Computer and System Sciences*, Vol. 33, 1986, pp. 75-87.
21. K. Gai, M. Qiu, Y. Li, and X.-Y. Liu, "Advanced fully homomorphic encryption scheme over real numbers," in *Proceedings of IEEE 4th International Conference on Cyber Security and Cloud Computing*, 2017, pp. 64-69.
22. V. Biksham and D. Vasumathi, "A lightweight fully homomorphic encryption scheme for cloud security," *International Journal of Information and Computer Security*, Vol. 13, 2020, pp. 357-371.
23. H.-M. Yang, Q. Xia, X.-F. Wang, and D.-H. Tang, "A new somewhat homomorphic encryption scheme over integers," in *Proceedings of IEEE International Conference on Computer Distributed Control and Intelligent Environmental Monitoring*, 2012, pp. 61-64.
24. Y. G. Ramaiah and G. V. Kumari, "Towards practical homomorphic encryption with efficient public key generation," *International Journal on Network Security*, Vol. 3, 2012, p. 10.
25. J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun, "Batch fully homomorphic encryption over the integers," in *Proceedings of Annual International Conference on Theory and Applications of Cryptographic Techniques*, 2013, pp. 315-335.
26. H. Pang and B. Wang, "Privacy-preserving association rule mining using homomorphic encryption in a multikey environment," *IEEE Systems Journal*, Vol. 15, 2020, pp. 3131-3141.
27. M. Thangavel and P. Varalakshmi, "Enhanced dna and elgamal cryptosystem for secure data storage and retrieval in cloud," *Cluster Computing*, Vol. 21, 2018, pp. 1411-1437.
28. J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully homomorphic encryption over the integers with shorter public keys," in *Advances in Cryptology*, 2011, pp. 487-504.
29. S. Dasgupta and S. Pal, "Design of a polynomial ring based symmetric homomorphic encryption scheme," *Perspectives in Science*, Vol. 8, 2016, No. j.pisc.2016.06.061.
30. D. Boer and S. Kramer, "Secure sum outperforms homomorphic encryption in (current) collaborative deep learning," *arXiv Preprint*, 2020, arXiv:2006.02894.

Mostefa Kara received the Eng. degree in Computer Science from the University of Biskra, Algeria, in 2005. He received the Master degree in Artificial Intelligence from the University of Eloued, Algeria, in 2019. He is currently a Ph.D. student at University of Eloued, Algeria. His research interests include cryptography, information security and decentralized algorithms.

Abdelkader Laouid received the MSc. degree in Computer Science from the University of Bejaia, Algeria, in 2011. He received the Ph.D. degree in 2017 from University of Bejaia, Algeria. He is currently an Associate Professor at the University of El-Oued. His research interests include distributed algorithms oriented to limited resource networks.

Ahcene Bounceur is an Associate Professor of Computer Science at the University of Brest (UBO). He is a member of the Lab-STICC Laboratory (MOCS Group). He received a Ph.D. in Micro and Nano Electronics at Grenoble INP, France in 2007. He received his MSc degree from ENSIMAG, Grenoble, France in 2003. His current research activities focus on Tools for physical simulation of WSN, mixed-signal, and RF circuits.

Mohammad Hammoudeh is the Saudi Aramco Chair Professor of Cyber Security in the Information & Computer Science Department at King Fahd University of Petroleum & Minerals. He is the founder and co-Editor in Cheif of ACM's journal of Distributed Ledger Technology: Research & Practice. His research interests are centered around the applications of zero trust security to Internet-connected critical national infrastructures, blockchains, and other complex highly decentralised systems.

Muath AlShaikh had earned a Ph.D. degree in Computer Science at Universit de Bretagne Occidentale, France, in 2016. He received his Master's degree in computer science from Utara University, Malaysia, in 2010 and his BSc in Computer Science from AlBalga University, Jordan, in 2006. Muath is an Assistant Professor at the College of Computing and Informatics, Saudi Electronic University, KSA. His research interests include cryptology, image processing, and IoT.