DOI: http://dx.doi.org/10.21123/bsj.2022.19.3.0619

# Text Multilevel Encryption Using New Key Exchange Protocol

*Zaid Nidhal Khudhair*[1] *Ahmed Nidhal*[2] *Nidhal K. El Abbadi* [3*]

[1] Faculty of Information Technology, Imam Ja'afar Al-Sadiq University, Al-Najaf El-Ashraf, Iraq.
[2] Electrical Engineering Dep., Faculty of Engineering, University of Kufa, Al-Najaf El-Ashraf, Iraq.
[3] Computer Science Dep., Faculty of Education, University of Kufa, Al Najaf El-Ashraf, Iraq.
[*]Corresponding author: zaidnidhal88@gmail.com, ahmed.elabadi@uokufa.edu.iq, nidhal.elabbadi@uokufa.edu.iq [*]
[*]ORCID ID: https://orcid.org/0000-0002-4890-9403, https://orcid.org/0000-0001-7573-6248, https://orcid.org/0000-0001-7178-5667 [*]

Received 4/11/2020, Accepted 20/4/2021, Published Online First 20/11/2021 Published 1/6/2022

## Abstract:

The technological development in the field of information and communication has been accompanied by the emergence of security challenges related to the transmission of information. Encryption is a good solution. An encryption process is one of the traditional methods to protect the plain text, by converting it into inarticulate form. Encryption implemented can be occurred by using some substitute techniques, shifting techniques, or mathematical operations. This paper proposed a method with two branches to encrypt text. The first branch is a new mathematical model to create and exchange keys, the proposed key exchange method is the development of Diffie-Hellman. It is a new mathematical operations model to exchange keys based on prime numbers and the possibility of using integer numbers. While the second branch of the proposal is the multi-key encryption algorithm. The current algorithm provides the ability to use more than two keys. Keys can be any kind of integer number (at least the last key is a prime number), not necessarily to be of the same length. The Encryption process is based on converting the text characters to suggested integer numbers, and these numbers are converted to other numbers by using a multilevel mathematical model many times (a multilevel process depending on the number of keys used), while the decryption process is a one-level process using just one key as the main key, while the other keys used as secondary keys. The messages are encoded before encryption (coded by ASCII or any suggested system). The algorithm can use an unlimited number of keys with a very large size (more than 7500 bytes), at least one of them a prime number. Exponentiation is also used for keys to increase complexity. The experiments proved the robustness of the key exchange protocol and the encryption algorithm in addition to the security. Comparing the suggested method with other methods ensures that the suggested method is more secure and flexible and easy to implement.

**Keywords**: Decryption, Hybrid encryption, Private key, Public key, Text encryption.

## Introduction:

Digital information transferred over communications channels increased rapidly with the development of computers, and this may be monitored by electronic eavesdroppers, where in general it is the ideal communication pass from the sender to the receiver directly without an interpreter from the third party. There are many applications of cryptography with various degrees of security [1].

In general, cryptography aims to achieve many goals. It is possible to achieve all these goals or some of them at the same time in one application. The first goal is authentication; it means that the systems or users can prove their own identities to other parties who don’t have personal knowledge of their identities [2]. The second goal is confidentiality, which is the most important goal. Confidentiality ensures that nobody other than the authorized person can decrypt the message. The third goal is data integrity, which confirms that the received message has not been modified intentionally or accidentally. The fourth goal is non-repudiation; this is the technique used to ensure that the message is sent and received by the authorized party. The last goal is access control, which controls the persons that can access the resources [3].

Public key cryptography was introduced in 1976 by Diffie and Hellman (DH) as one of the greatest contributions to solving the problem of key exchange [4]. There are many applications of public-key encryption schemes; one of them allows secure communication between two parties over public channels without needing to know each other to establish a shared secret key [5].

The public key is also called (asymmetric-key). It is one of two schemes of encryption algorithms, while the other scheme is a private key called an asymmetric key [6].

The symmetric scheme uses one key for encryption and decryption; it is called a private key. Anyone possessing the private key will be able to encrypt and decrypt the message. Thus, the private key must be kept secret all the time. This might cause some disadvantages and give way for attacks on algorithm [7].

The asymmetric scheme uses pairs of keys (public and private); one for encryption, while the other for decryption. The public key is used for encryption of the message and the private key for the decryption of the message. Anyone can use the public key which is works in conjunction with its equivalent private key. Asymmetric key Cryptography needs more processing since encryption and decryption use different keys [8].

Hellman's algorithm and the RSA algorithm fail to find a simple way of deriving the cryptographic keys. Existing encryption algorithms fail to find and implement easy and inexpensive ways in software and hardware for producing asymmetric cryptography. The current encryption methods which fail to find a secure method can use integer numbers (not prime numbers) in the cryptographic process [9, 10].

The present method for attacking Diffie-Hellman (DH) depends on compromising one of the private keys by calculating the discrete log of the corresponding public value in the DH group [10].

This attack takes advantage of the fact that an adversary can perform a single huge computation to crack a particular prime and then easily break any individual connection that uses that prime. The attacker can start with the phase of the attack which depends only on the prime. This can be done in advance, which leaves only the second phase to be done on the fly for any particular connection.

Multi-level Data Encryption describes the improvement of data encryption complexity due to multiple operations of single-phase encryption techniques in cryptography. Better security can be achieved by multiple encryptions because even if some component ciphers are exposed or some of the secret keys are detected, the confidentiality of the original data can still be maintained by the multiple encryptions [11].

Many algorithms are used for protecting the information and keeping it secured. These algorithms vary in their ability to resist and avoid attacks. Popularly used algorithms include DES (key size 56 bits), Rijndael (AES) (key size 128, 192, 256 bits), Triple-DES (key size 112, 168 bits), SEAL (key size 160 bit), Blowfish (key size 128 bits) [12], RC2 (key size 128 bits), RC4 (variable key size), Twofish (key size 128, 192, 256 bits), RSA (the key size of an RSA key (1024 bits) specifies the number of bits in the modulus) [13], and HiSea (the key size 1 –4096 set of integers) [14]. Many researchers work in this field, such as:

Mahalakshmi and Kuppusamy suggested a way used asymmetric key encryption. The authors used multiple keys for encryption. They generate one hybrid key from improved cipher block chaining encryption and the second one from a genetic algorithm. This algorithm is used for plain text and images [15].

Madhvi and Gagandeep tried to get the best optimization for improving cloud security by introducing a hybrid algorithm that combines flower pollination and DNA cryptography. The first algorithm stemmed from nature to find the best solution, while the second algorithm helps to encrypt big data by using a very small amount of DNA[16].

Lipi Nski proposed a key exchange method based on octonion algebra. The authors suggested three protocols for key exchange used for cryptography. The authors generalize the RSA algorithm to the octonion arithmetics by using the totient function defined for integral octonions [17].
David Adrian and colleagues showed that, in practice, applications that use Diffie-Hellman tend to choose a universally fixed prime p (and fixed g). For example, several SSH servers and IPsec VPNs use a fixed universal 1,024-bit prime p. Same thing for HTTPS Web servers, despite a less extent. The authors proved that the 1024 bit prime p is not safe to use everywhere. This is due to two steps of the precomputation attack on the discrete-log problem modulo a prime. In the first step, and before attacking the victim, the attacker works on creating a specific table based on the fixed p and g. this table will be used in the second step to compute the discrete log and break session [18].

The rest of the paper is organized as follows: Section two presents the contribution of this paper, followed by the summary of the proposed method which includes key exchange, encryption process, and decryption process. Where section four is focused on the experimental results. Finally, the conclusion comes out with section five.

## Contributions:

Our contribution is constructing a service application to provide security as a service to its users when encrypting any type of data. Two main new algorithms are suggested in this work. A new key exchange algorithm is suggested to offer a highly secure encryption process, this algorithm is an improved algorithm of Diff-Helman's algorithm. The secure keys are determined in a different way which is more secure.

The second proposed algorithm is a new encryption algorithm based on multiple keys (two or more keys) which increases the complexity. The key size can be more than 7500 bytes. For more complexity, the encryption process is modified by raising the key's values to specific exponents. The novelty of this paper is the use of a hybrid scheme (symmetric and asymmetric schemes), where several public keys are used for encryption, and one private key is used for decryption, at the same time all the keys used to calculate the base value which is used in encryption and decryption.

## Summary of the Proposed Algorithms:

The current proposal provides the ability to use more than two keys. Keys can be prime numbers and any kind of integer numbers, not necessarily just prime numbers and not necessarily of the same length (whole keys' values may be positive, negative, or a combination of positive and negative values). Two of the keys' values must be greater than one or less than (-1), or at least one of the key's values should be greater than two or less than (-2). The whole suggested keys except the last one are used for encryption, while the last key is used for decryption in addition to the other keys. Some of the keys are used as public keys while the others are used as private keys. The last key must be a private key and prime number. The proposed algorithm suggests many encryption levels based on the number of keys.

The current suggestion consists of setting up a process and three main functions that process data in various ways. In the setup process of the preferred embodiment, the sender and receiver agree upon a set of initial public parameters and the authorities generate an escrowing public key and corresponding private keys. The functions are key generation, encryption process, and decryption process.

### Key generation

This algorithm uses arithmetic modulus as the basis of its calculation. Thus, establishing a shared secret over an unsecured communication channel.

The first protocol suggested by Diffie and Hellman works by choosing a large prime number $p$ and determining certain exponentiations reminder of this prime number. The protocol will be secure if at the very least difficult solving the discrete-log problem modulo the prime number $p$. This problem is quite easy to state: fix a large prime $p$, and an integer $0 < g < p$ (a generator). Next, select an integer $0 < x < p$ and determine the remainder (h) of dividing $g^x$ on $p$.

The problem of discrete-log is to determine (x) when we have p, g, and h only. the Diffie-Hellman protocol regards as insecure for selected (p, g) when this problem for most (h) will be efficiently solved[18].

In this paper, the suggested key exchange algorithm is based on the agreement between the sender and the receiver on a key, in such a way that an eavesdropper cannot obtain the key.

Steps of the suggested protocol (Fig. 1):

1. The sender and receiver agree on an integer number (g) and a base number (n), whether they are prime or not.
2. The sender selects a private random natural number (a) and sends ($g^a$ mod n) to the receiver. do
3. The receiver selects a secret number (b) and sends ($g^b$ mod n) to the sender.
4. The sender computes ((($g^b$ mod n) × $g^{a-1}$) mod n).
5. The receiver computes ((($g^a$ mod n) × $g^{b-1}$) mod n).
6. Both sender and receiver can use this number as their key.

Notice that (g) and (n) do not need to be protected.

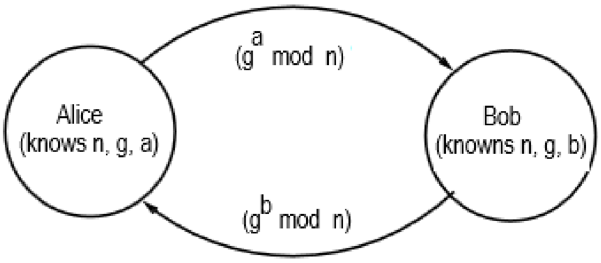

**Figure 1. Suggested key exchange process.**

Details of the suggested protocol are as follow:
The sender and receiver agree on two numbers (g and base number n).
Sender picks random numbers (a), and:
The sender determines (X)

$$X = g^a \, mod \, n \qquad (1)$$

The receiver picks random numbers (b), and:
The receiver determines (Y)

$$Y = g^{b} \bmod n \qquad (2)$$

The protocol just exchanges these numbers:
Sender → Receiver: X (3)
Receiver → Sender: Y (4)
Sender calculates private key

$$x = (Y \times g^{a-1)} \bmod n \qquad (5)$$

The receiver calculates the private key

$$y = (X \times g^{b-1}) \bmod n \qquad (6)$$

Both (x, y) are equal.

For proving this suggestion:

$$x = (Y \times g^{a-1}) \bmod n = (g^{b} \times g^{a-1}) \bmod n \qquad (7)$$
$$= (g^{b+a-1}) \bmod n$$
$$y = (X \times g^{b-1}) \bmod n = (g^{a} \times g^{b-1}) \bmod n \qquad (8)$$
$$= (g^{a+b-1}) \bmod n$$

So, x = y

### Encryption process

In this method, the first step is encoding the messages before the encryption process, by a mapping technique that converts the plain text into ASCII values or any vector of suggested numbers. The data encrypting is based on finding the base value. The base value is calculated by multiplying the key values ($k_1$, $k_2$, $k_3$, …, $k_n$) and then subtracting one from the product or adding one to the product (when the base value is a negative value) as in Fig. 2. The keys ($k_1$, $k_2$, $k_3$, …, $k_n$) are generated using the key generated process or they are picked randomly. At least one of these keys should be a prime number.

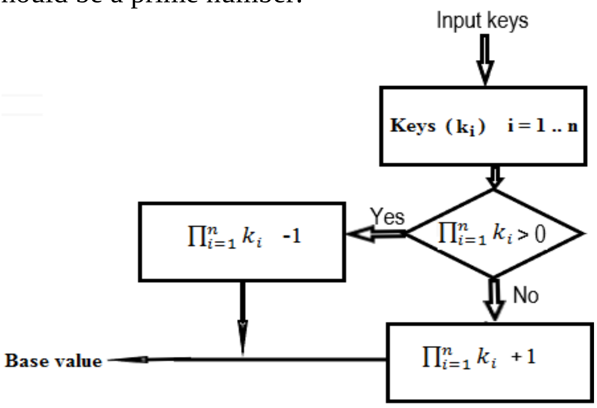

**Figure 2. A process of determining the base value.**

The value of the message to encrypt may be any integer numbers positive or negative (text can be represented by numbers, either ASCII numbers or suggested numbers for each character). The range of the value to encrypt (message) is related to the sign and value of the base value. When the base value is positive then the range of message value must be from (- the base value minus one to the base value minus one), while in the case of a negative base value then the range of the message must be from (the base value minus one to absolute value of the base value minus one).

$$-B < M < B \qquad \textit{When (base value (B) >0)} \qquad (9)$$
$$B < M < |B| \qquad \textit{When (base value (B) < 0)} \qquad (10)$$

The value to be encrypted is multiplied by the first key-value, then the remainder of dividing the product on the base values is determined by the modulus, which represents the first sub-encrypted value.

The first sub-encrypted value is multiplied again with the second key-value, the remainder of dividing the product on the base value is determined by a modulus operation, which represents the second sub-encrypted value. This process is repeated; every time the result from the previous step is encrypted with a new key until all keys have been processed ($k_1 \ldots k_{n-1}$) (the last key specified for decryption). The result after encrypting the sub-encrypted value with key ($k_{n-1}$) is the final encrypted message. The sign of the encrypted message must be the same as the sign of the message. So, when the sign of the message is negative the sign of the encrypted message changes to be negative. The Encrypted message is sent to the receiver (Fig. 3).

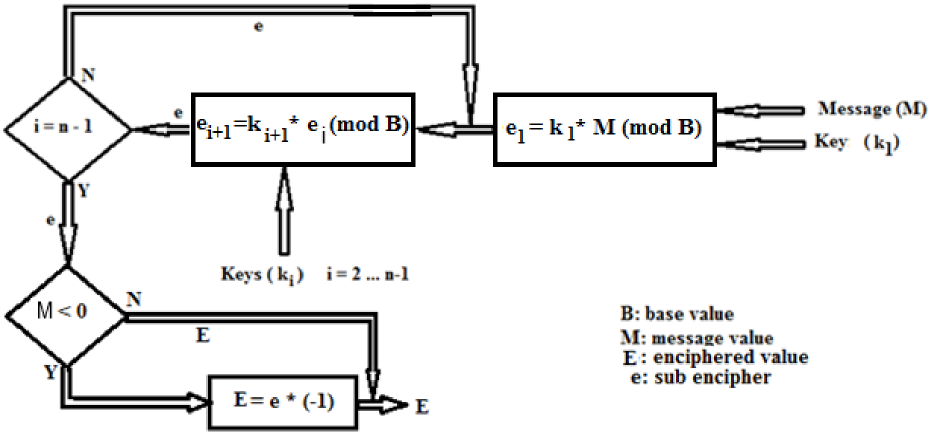
e
N
i = n - 1
e
e<sub>i+1</sub>=k<sub>i+1</sub>* e<sub>i</sub> (mod B)
e<sub>1</sub> = k<sub>1</sub>* M (mod B)
Message (M)
Key (k<sub>1</sub>)
Y
e
Keys ( k<sub>i</sub> ) i = 2 ... n-1
M < 0
N
E
Y
E = e * (-1)
E
B: base value
M: message value
E : enciphered value
e: sub encipher

Figure 3. Proposed encryption process.

**Modified method:**

A modified method for finding the encryption value is implemented by choosing an exponent value, the range of exponent value is the entire positive integers; raising each key-value ($k_1 \ldots k_{n-1}$) to the exponent value and then multiplying each key by the message or sub enciphered value as explained previously (Fig. 4). The result is a newly enciphered message.

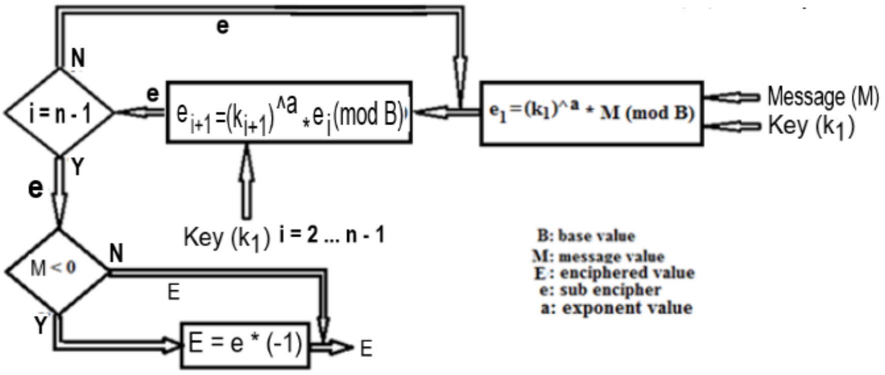

Figure 4. Steps of the modified encryption process.

**Decryption process:**

To decrypt the encrypted message, the receiver should calculate the base value. The base value is calculated by multiplying the keys' values ($k_1$, $k_2$, $k_3$, …, $k_n$) and then subtracting one from the product. The encrypted value can be decrypted by multiplying it by the last key value ($k_n$). Then, the remainder of dividing the result of multiplication on the base value is the decrypted value. Check the sign of encrypted value. If it was a negative value, then the sign of the decrypted value must be changed to a negative value. Figure 5 clarifies the decryption process.

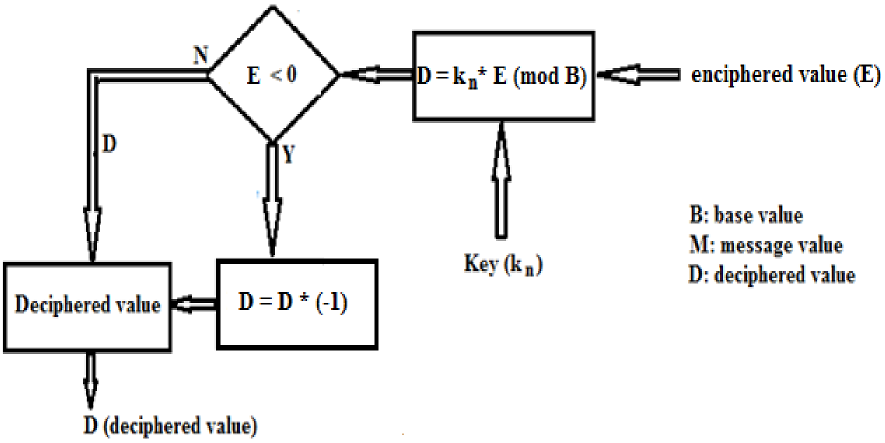

Figure 5. Steps of the decryption process.

If the encryption process uses an exponent value, then the same exponent value is selected in the decryption process; raise the last key ($k_n$) to the exponent value and then multiply it by the encrypted value. Then the remainder of dividing the result of multiplication on the base value is the decrypted value as in Fig. 6.

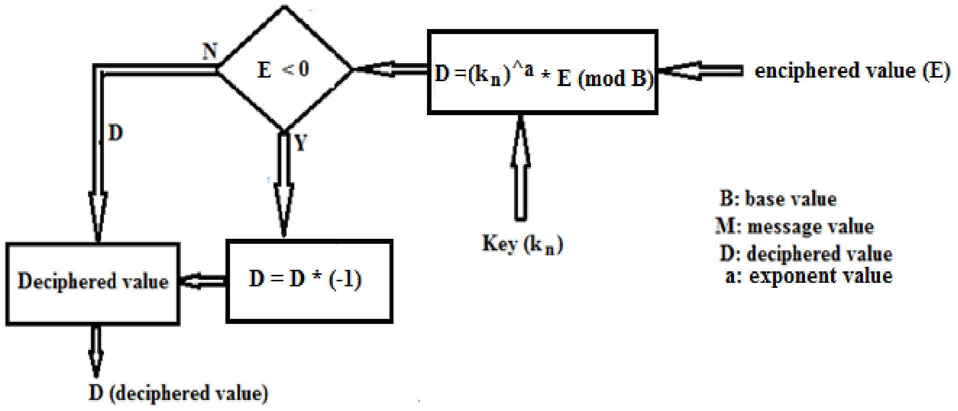

**Figure 6. Decryption steps for modified encrypted value.**

## The Results:

Many examples used to exhibit the power of the suggested algorithm as follow:

***Example 1: key exchange***

Sender and receiver agree for

(g = 1218, and n = 29013432)

The sender selects a random prime number (a = 9103)

The receiver selects a random prime number (b = 417)

Sender calculates (X) where

$X = g^a \bmod n$

$X = (1218)^{9103} \bmod 29013432 = 13598256$

The receiver calculates (Y) where

$Y = g^b \bmod n$

$Y = (1218)^{417} \bmod 29013432 = 496608$

The sender send (X) to Receiver

The receiver sends (Y) to Bob

The sender calculates the private key

$x = Y \times g^{a-1} \bmod n$

$x = 496608 \times (1218)^{9103-1} \bmod 29013432$

x = 8774472

The receiver calculates the private key

$y = X \times g^{b-1} \bmod n$

$y = 13598256 \times (1218)^{417-1} \bmod 29013432$

y = 8774472

***Example 2***

This process step-by-step is:

Picking three numbers (keys): k1=10, k2 = 8, and k3 = 11

Given the message (M) = 330

**Encryption**

While (k1*k2*k3 > 0) then

Base value (B) = k1 * k2 * k3 – 1

B = 10 * 8 * 11 – 1 = 879

e1 = k1 * M mod B

e1 = 10 * 330 mod 879 = 663

e2 = k2 * e1 mod B

e2 = 8 * 663 mod 879 = 30

**Decryption**

Base value (B) = k1 * k2 * k3 – 1

B = 10 * 8 * 11 – 1 = 879

e = 30

Decrypted value (D)= k3 * e mod B

D = 11 * 30mod 879 = 330

***Example 3 (Using negative keys)***

Pick three negative numbers (keys): k1 = -5, k2 = -9, k3 = -23

Given the message (M) = 234

**Encryption**

While (k1*k2*k3 <0) then

Base value (B) = k1 * k2 * k3 + 1

B = -5 * -9 * -23 + 1 = -1034

e1 = k1 * M mod B

e1 = -5 * 234 mod -1036 = 136

e2 = k2 * e1 mod B

e2 = -9 * 136 mod -1034 = 190

**Decryption**

Base value (B) = k1 * k2 * k3 + 1

B = -5 * -9 * -23+ 1 = -1034

e = 190

Decrypted value (D) = k3 * e mod B

D = -23 * 190 mod -1036 = 234

***Example 4 (negative message)***

Picking three numbers (keys): k1 = -12, k2= -7, k3= 17

Given the message (M) = -76

**Encryption**

While (k1*k2*k3 > 0) then

Base value (B) = k1 * k2 * k3 – 1

B = -12 * -7 * 17 – 1 = 1427

e1 = k1 * M mod B

e1 = -12 * -76 mod 1427= 912

e2 = k2 * e1 mod B

e2 = -7 * 912 mod 1427= 751
if (M < 0) then e2 = e2 * (-1)
e2 = -751

**Decryption**

Base value (B) = k1 * k2 * k3 – 1
B = -12 * -7 * 17 – 1 = 1427
e = -751
Decrypted value (D) = k3 * e mod B
D = 15 * (-751) mod 1427= 76
If (e < 0) then D = D * (-1)
D = -76

*Example 5*

This example clarifies different cases for the signs of keys, messages, and the base. The calculating of encrypted and decrypted values are summarized in Table 1.

**Table 1. determine values of encrypted and decrypted values for different cases.**

| K1 | K2 | K3 | M | B | E1 | E2 | E2 send | D | Final D | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 5 | 7 | 10 | 104 | 30 | 46 | 46 | 10 | 10 | The sign of (E2) changed according to the sign of (M), and the sign of (D) changed according to the sign of (E2). |
| -3 | -5 | 7 | -10 | 104 | 30 | 58 | -58 | 10 | -10 | |
| -3 | 5 | -7 | 10 | 104 | 74 | 58 | 58 | 10 | 10 | |
| 3 | -5 | -7 | -10 | 104 | 74 | 46 | -46 | 10 | -10 | |
| -3 | 5 | 7 | 10 | -104 | 30 | 58 | 58 | 10 | 10 | |
| 3 | -5 | 7 | -10 | -104 | 30 | 46 | -46 | 10 | -10 | |
| 3 | 5 | -7 | 10 | -104 | 74 | 46 | 46 | 10 | 10 | |
| -3 | -5 | -7 | -10 | -104 | 74 | 58 | -58 | 10 | -10 | |

*Example 6 (using five keys)*

In this example picking 5 numbers (keys): k1=22, k2 = 122, k3 = 9, k4 = 4001, and k5 = 1163
Given message (M) = 123

**Encryption**

While (k1*k2*k3 > 0) then
base value (B) = k1 * k2 * k3 * k4 * k5 - 1
B = 22 * 122 * 9 * 4001 * 1163 – 1
B = 112401805427
e1 = k1 * M mod B = 22 * 123 mod 112401805427 = 2706
e2 = k2 * e1 mod B = 122 * 2706 mod 112401805427 = 330132
e3 = k3 * e2 mod B = 9 * 330132 mod 112401805427 = 2971188
e4 = k4 * e3 mod B = 4001 * 2971188 mod 112401805427 = 11887723188

**Decryption**

base value (B) = k1 * k2 * k3 * k4 * k5 – 1
e = 11896636752
B = 22 * 122 * 9 * 4001 * 1163 – 1
B = 112401805427
Decrypted value (D) = k5 * e mod B
D = 1163 * 11887723188 mod 112401805427 = 123

*Example 7 (using keys with exponent)*

The same numbers in example 6 are used in this example, with selecting exponent value = 3 picking 5 numbers (keys): k1=22, k2 = 122, k3 = 9, k4 = 4001, and k5 = 1163
Given message (M) = 123
exponent value (ex) = 3

**Encryption**

base value (B) = k1 * k2 * k3 * k4 * k5 - 1
B = 22 * 122 * 9 * 4001 * 1163 – 1
B = 112401805427
e1 = k1^3 * M mod B = 22^3 * 123 mod 112401805427 = 1309704
e2 = k2^3 * e1 mod B = 122^3 * 1309704 mod 112401805427 = 17785475025
e3 = k3^3 * e2 mod B = 9^3 * 9535307390 mod 112401805427 = 39403669120
e4 = k4^3 * e3 mod B = 4001^3 * 10083506413 mod 112401805427 =28129774537

**Decryption**

base value (B) = k1 * k2 * k3 * k4 * k5 – 1
e = 28129774537
B = 22 * 122 * 9 * 4001 * 1163 – 1
B = 112401805427
Decrypted value (D) = k5 * e mod B
D = 1163 * 28129774537 mod 112401805427= 123

*Example 8 (large number keys)*

This process step-by-step is:
Picking three large numbers (keys):
k1= 1234567*$(10)^{300}$
k2 = 2468*$(10)^{300}$

k3 = $1122334455*(10)^{290}$

Given the message (M) = 1122334455667788

**Encryption**

Base value (B) = k1 * k2 * k3 – 1

B = $(1234567*(10)^{300} * 2468*(10)^{300} * 1122334455*(10)^{290}) – 1 = 3.41965359616957095988 * (10)^{908}$

e1 = k1 * M mod B

e1 = $1234567*(10)^{300} * 1122334455667788 \pmod{3.41965359616957095988 * (10)^{908}} = 1.385597081930414030*(10)^{321}$

e2 = k2 * e1 mod B

e2 = $2468*(10)^{300} * 1.385597081930414030*(10)^{321} \pmod{3.41965359616957095988 * (10)^{908}} = 3.4196535982042617986*(10)^{624}$

**Decryption**

Base value (B) = k1 * k2 * k3 – 1

B = $(1234567*(10)^{300} * 2468*(10)^{300} * 1122334455*(10)^{290}) – 1 = 3.41965359616957095988 * (10)^{908}$

e = $3.4196535982042617986*(10)^{624}$

Decrypted value (D) = k3 * e mod B

D = $1122334455*(10)^{290}* 3.4196535982042617986*(10)^{624} \pmod{3.41965359616957095988 * (10)^{908}} = 1122334455667788$

The above examples showed the ability of the suggested algorithm and how it is flexible to work with a different number of keys, different lengths of keys, and values, also prove the possibility to work with positive and negative numbers, prime and non-prime numbers.

The suggested algorithm is compared with other encryption algorithms. The suggested algorithm has several advantages over many other encryption algorithms as shown in Table 2.

**Table 2. comparing different encryption algorithms**.

| Methods | Authors | Key Size bits | Structure | Features |
|---|---|---|---|---|
| DES | IBM | 64 | Festial | Not Strong Enough |
| 3DES | IBM | 112 or 168 | Festial | Adequate Security |
| AES | Joan Daemen & incent Rijmen | 128, 192, 256 | Substitution Permutation | Excellent Security |
| Blowfish | Bruce Schneier | 32-448 | Festial | Excellent Security |
| RC4 | Ron Rivest | Variable | Festial Stream | Fast Cipher in SSL |
| RC2 | Ron Rivest | 8-128<br>64 by default | Festial | Stream Cipher |
| Twofish | Bruce Schneier | 128- 256 | Festial | Good Security |
| Serpent | Anderson, Lars Knudsen | 128- 256 | Substitution permutation | Good Security |
| IDEA | James Massey | 128 | Substitution Permutation | Not Strong Enough |
| RC6 | Ron Rivest, Matt Robshaw | 128 to 256 | Festial | Good Security |
| RSA | Rivest, Shamir, Adleman | 1,024 to 4,096 | Public Key algorithm | Excellent Security, low speed |
| Diffie-Hellman | Whitfield Diffie Hellman | 1024 to 4096 | Asymmetric algorithm | Many attacks |
| HiSea | Sapiee Jamel | 1 –4096 set of integers | Substitution-Permutation | Highly secure |
| **NAZ** | **Proposed algorithm** | **1-6000 set of integers** | **Asymmetric algorithm** | **Highly secure** |

Also, the suggested key exchange algorithm compared with two of the most popular key exchange algorithms, it is clear that each algorithm has its advantage, and all of them are robust against attacks. The three key exchange algorithms are shown in Figs. 7, 8, and 9 allow the reader to compare between them. Where in Fig. 7 Diffe-Helmin algorithms are shown, and the ID-KEX protocol is shown in Fig. 8. and finally, the proposed key exchange (Nidhal algorithm) is shown in Fig. 9.

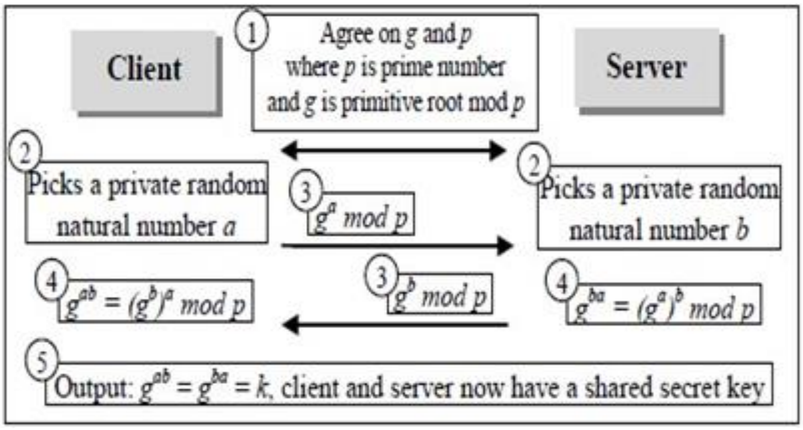
Client
1
Agree on $g$ and $p$ where $p$ is prime number and $g$ is primitive root mod $p$
Server
2
Picks a private random natural number $a$
3
$g^a$ mod $p$
2
Picks a private random natural number $b$
4
$g^{ab} = (g^b)^a$ mod $p$
3
$g^b$ mod $p$
4
$g^{ba} = (g^a)^b$ mod $p$
5
Output: $g^{ab} = g^{ba} = k$, client and server now have a shared secret key

**Figure 7. Diffie-Hellman key exchange (DH-KEX) (18).**

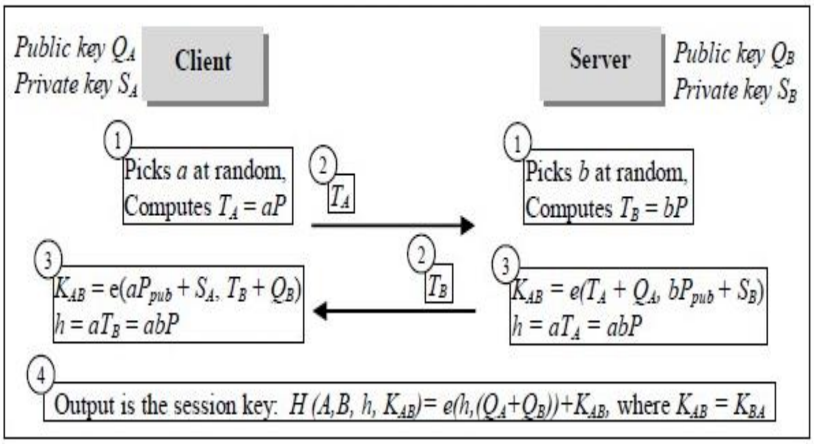
Public key $Q_A$
Private key $S_A$
Client
Server
Public key $Q_B$
Private key $S_B$
1
Picks $a$ at random, Computes $T_A = aP$
2
$T_A$
1
Picks $b$ at random, Computes $T_B = bP$
3
$K_{AB} = e(aP_{pub} + S_A, T_B + Q_B)$
$h = aT_B = abP$
2
$T_B$
3
$K_{AB} = e(T_A + Q_A, bP_{pub} + S_B)$
$h = aT_A = abP$
4
Output is the session key: $H(A,B,h,K_{AB}) = e(h,(Q_A+Q_B))+K_{AB}$, where $K_{AB} = K_{BA}$

**Figure 8. ID-based key exchange (ID-KEX) (18).**

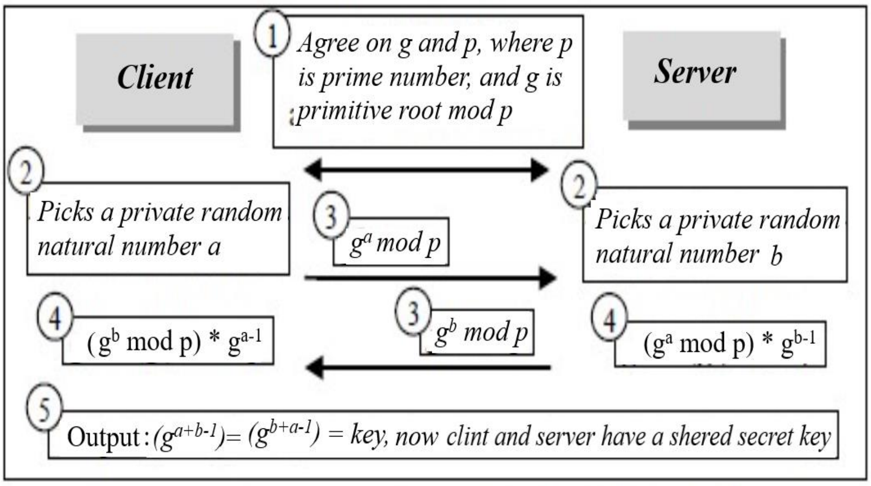
Client
1
Agree on g and p, where p is prime number, and g is primitive root mod p
Server
2
Picks a private random natural number a
3
$g^a$ mod p
2
Picks a private random natural number b
4
$(g^b \text{ mod } p) * g^{a-1}$
3
$g^b$ mod p
4
$(g^a \text{ mod } p) * g^{b-1}$
5
Output: $(g^{a+b-1}) = (g^{b+a-1})$ = key, now clint and server have a shered secret key

**Figure 9: Nidhal key exchange (NAZ-KEX).**

One important advantage of this proposed algorithm is the robustness to attack. The supercomputer with about $10^{17}$ FLOPS needs more than $10^{800}$ years to find the key value in the simple form**.**

Finally, the performance of the suggested algorithm compared with other algorithms suggested by other authors is shown in Table 3. The comparing results prove that the suggested algorithm has promised and dependable results.

**Table 3. comparing crypto time using Crypto ++ (19).**

| S. No | Algorithm | Megabytes (2^20 bytes) processed | Time Taken | MB/Second |
|---|---|---|---|---|
| 1 | Blowfish | 256 | 3.976 | 64.386 |
| 2 | Rijndael (128-bit key) | 256 | 4.196 | 61.010 |
| 3 | Rijndael (192-bit key) | 256 | 4.817 | 53.145 |
| 4 | Rijndael (256-bit key) | 256 | 5.308 | 48.229 |
| 5 | Rijndael (128) CTR | 256 | 4.436 | 57.710 |
| 6 | Rijndael (128) OFB | 256 | 4.837 | 52.925 |
| 7 | Rijndael (128) CFB | 256 | 5.378 | 47.601 |
| 8 | Rijndael (128) CBC | 256 | 4.617 | 55.447 |
| 9 | DES | 128 | 5.998 | 21.340 |
| 10 | (3DES) DES-XEX3 | 128 | 6.159 | 20.783 |
| 11 | (3DES) DES-EDE3 | 64 | 6.499 | 9.848 |
| 12 | Proposed Algorithm | 256 | 4.023 | 63.634 |

## Conclusion:

This type of encryption scheme is more effective than a single encryption scheme.

Better security provided is one of the multi-level encryption advantages. If someone breaks some of the cipher components or disclosed keys, the origin data will remain confident by multi-level encryption.

The current proposal tries to encrypt the message by using multiple keys. The text message can be represented as a number, there are many methods used to convert text to a number, it is easier and more complex to convert text to a new form of numbers before encrypting them by a suggested method. It is highly recommending to include the location of each word in the text in the process of converting text to numbers. On the other hand, this algorithm needs some modification to become suitable for the encryption of the images.

There are many contributions and advantages presented in this work.

The contributions of this paper are the use of integer numbers in addition to the prime numbers in key exchange and key generation. Using integer numbers reduces the probability of attacking prime numbers as happened with Diffie-Hellman’s algorithm. The number of keys is unlimited with a different key sizeable to exceed 7000 bytes.

In this method, public and private keys are used for encryption while the decryption is based on the private key (prime number) and the public keys for determining the base value (may integer and prime numbers).

Attacking for this method will focus on the values exchanged between two parties which represent part of the base value, cracking these values will explore part of the base values or some of the keys, but not all the keys were used in the encryption and may need a very long time to crack the keys.

For more complexity, the encryption process is modified by using the exponent value (e) which represents the entire positive integer values.

Encryption is a multilevel encryption process, while decryption is a single-level process. All the keys will be used in the encryption and decryption process in different ways.

The current proposal shows a hybrid scheme (asymmetric and symmetric schemes) of cryptography. The base algorithm produces encryption in a cheap way and without complications for encryption achievement and use. This hybrid scheme provides excellent security and speed in one algorithm. It is enabled by the integer numbering system disclosed herein.

Huge key size and huge module number (base value) are disclosed herein. Key size reached (for all the keys) with a personal computer (Core (TM) i7, CPU 2.4 GH) in the present algorithm is more than 7000 bytes, the same as for module number (base value).

The suggested algorithm introduces an easy way and can be implemented in a cheap computer for asymmetric encryption which reduces the complexity of current methods in making private use of prime numbers and exponentiation. also, this algorithm can be used for symmetric encryption.

Using one multiplication in the suggested encryption leads to an increase the security and a decrease in complications and cost.

## Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine. Besides, the Figures and images, which are not mine ours, have been given permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at the University of Kufa.

## Authors' contributions statement:

Zaid Nidhal contributed to design, conceptualization, methodology, software, data curation, analysis of results, writing of the manuscript.

Ahmed Nidhal contributed to the design, methodology, analysis of results, validation, investigation, writing of the manuscript.

Nidhal El Abbadi contributed to design, conceptualization, visualization, supervision, project administration, and funding acquisition, review, and editing of the manuscript.

## References:

1. Nentawe Y. Goshwe. Data encryption, and decryption using RSA algorithm in a network environment. IJCSNS. 2013; 13(7): 9-13.
2. Adil J Z, Momeen K. General Summary Cryptography. Int J Eng Res Appl. 2018; 8(2): 68-71: DOI: 10.9790/9622-080206871.
3. Kumari S. A research paper on cryptography encryption and compression techniques. IJECS. 2017; 6(4): 20915-9: DOI: 10.18535/ijecs/v6i4.20.
4. Nidhal K. El Abbadi, Abaas S T, Abd Alaziz A. New image encryption algorithm based on Diffie-Hellman and singular value decomposition. IJARCCE. 2016; 5(1): 197-201: DOI 10.17148/IJARCCE.2016.5147.
5. Abdullah M J, Azman S. A New public-key encryption scheme based on non-expansion visual cryptography and Boolean operation. IJCSI. 2010; 7(4): 1-10.
6. Mahajan P, Sachdeva A. A Study of encryption algorithms AES, DES, and RSA for security. GJCST, Web & Security. 2013; XIII(XV): 15-22.
7. Tumati G, Rajesh Y, Manogna T, Ram K J. A New Encryption Algorithm Using Symmetric Key Cryptography. IJET. 2018; 7(2.32): 436 –8: DOI: 10.14419/ijet.v7i2.32.15734.
8. Sowmiya M, Prabavathi S. Symmetric and Asymmetric Encryption Algorithms in Cryptography. Int J Recent Technol Eng. 2019; 8(1S2): 355-7.
9. Almeida J. Asymmetric cryptography using shadow numbers. US Patent 8811606 B2, 2014;
10. David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, et. al. Imperfect forward secrecy: How Diffie-Hellman Fails in practice. (CACM) Communications of the ACM. 2019; 62(1): 106-14: DOI: http://dx.doi.org/10.1145/3292035.
11. Gupta H, Sharma V. K. Multiphase encryption: A New concept in modern cryptography. Int. J. Comput. Theory Eng. 2013; 5(4): 638-40: DOI: 10.7763/IJCTE.2013.V5.765.
12. Omar G A, Shawkat G. A Survey on Cryptography Algorithms. IJSRP. 2018; 8(7): 495-516: DOI: 10.29322/IJSRP.8.7.2018.p7978.
13. Al-Hazaimeh O. A New approach for complex encrypting and decrypting data. IJCNC. 2013; 5(2): 95-103: DOI: 10. 5121/ijcnc.2013.5208.
14. Mushtaq MF, Jamel S, Disina AH, Pindar ZA, Shakir NS, Deris MM. A Survey on the Cryptographic Encryption Algorithms. IJACSA. 2017; 8(11): 333-44.
15. Mahalakshmi J, Kuppusamy K. Hybridization of ICBC and Genetic Algorithm for Optimizing Encryption Process in Cloud Computing Application Service. Fundam. Inform. 2018; 157 (1-2): 79–109: DOI 10.3233/FI-2018-1619.
16. Popli M. DNA Cryptography: A Novel Approach for Data Security Using Flower Pollination Algorithm, Proceedings of International Conference on Sustainable Computing in Science, Technology and Management, Amity University Rajasthan, India. 2019; 26(28): 2069-76
17. Lipinski Z. Symmetric and Asymmetric cryptographic key exchange protocols in the octonion algebra. Springer. AAECC. 2021; 32:81–96; DOI.org/10.1007/s00200-019-00402-1.
18. Mokhtarnameh R, Muthuvelu N, Ho SB, Chai I. A Comparison Study on Key Exchange-Authentication protocol. Int. J. Comput. Appl. 2010 Sep;7(5):5-11.DOI:10.5120/1161-1459.
19. Thirupalu U, Kesavulu R E. Performance Analysis of Cryptographic Algorithms in the Information Security. IJERT. NCISIOT - 2019 Conference Proceedings. 2019; 8(2): 64-9.

# تشفير النص متعدد المستويات باستخدام برتوكول جديد لتبادل المفاتيح

زيد نضال خضير[1] احمد نضال خضير[2] نضال خضير العبادي[3]

[1] كلية تكنلوجيا المعلومات، جامعة الامام جعفر الصادق، النجف الاشرف، العراق.
[2] قسم الهندسة الكهربائية، كلية الهندسة، جامعة الكوفة، النجف الاشرف، العراق.
[3] قسم الحاسبات، كلية التربية، جامعة الكوفة، النجف الاشرف، العراق.

## الخلاصة:

رافق التطور التكنولوجي في مجال المعلومات والاتصالات ظهور تحديات أمنية تتعلق بنقل المعلومات. التشفير هو حل جيد. عملية التشفير هي إحدى الطرق التقليدية لحماية النص العادي، وذلك بتحويلها الى صيغ غير مفهومة. التشفير من الممكن ان ينفذ باستخدام تقنيات التعويض، تقنيات التحويل، او العمليات الرياضية. اقترحت هذه الورقة طريقة تتكون من فرعين لتشفير النص. الفرع الأول هو نموذج رياضي جديد لانشاء وتبادل المفاتيح، طريقة تبادل المفاتيح المقترحة هي تطوير لطريقة دايف-هلمن. وهو نموذج لعمليات رياضية جديدة لتبادل المفاتيح بالاعتماد على الاعداد الأولية وإمكانية استخدام الأرقام الصحيحة. بينما الفرع الثاني لهذا المقترح هو خوارزمية تشفير متعددة المفاتيح. توفر الخوارزمية الحالية القدرة على استخدام اكثر من مفتاحين. من الممكن ان تكون المفاتيح أي نوع من الأرقام الصحيحة (على الأقل المفتاح الأخير يكون رقم اولي)، وليس بالضرورة ان تكون المفاتيح بنفس الطول. تعتمد عملية التشفير على تحويل احرف النصوص الى ارقام صحيحة مقترحة، ويتم تحويل هذه الأرقام الى ارقام أخرى باستخدام نموذج رياضي متعدد المستويات عدد من المرات (عدد المستويات يعتمد على عدد المفاتيح المستخدمة)، في حين ان عملية كسر التشفير هي عملية من مستوى واحد يستخدم فيها مفتاح واحد كمفتاح رئيس، بينما المفاتيح الأخرى تستخدم كمفاتيح ثانوية. يتم تغير قيم الرسالة قبل عملية التشفير (من الممكن باستخدام شفرة الاسكي او استخدام نظام مقترح). من الممكن ان تستخدم الخوارزمية المقترحة عدد غير محدود من المفاتيح ذات حجوم كبيرة جدا (اكثر من 7500 بايت)، وعلى الأقل واحد من هذه المفاتيح يكون عدد اولي. يستخدم الاس أيضا للمفاتيح لزيادة التعقيد.

**الكلمات المفتاحية:** كسر التشفير، التشفير الهجين، المفتاح الخاص، المفتاح العام، تشفير النصوص.

,