[Egyptian Informatics Journal 21 (2020) 83–90](https://doi.org/10.1016/j.eij.2019.11.002)

Contents lists available at [ScienceDirect](http://www.sciencedirect.com/science/journal/11108665)

Egyptian Informatics Journal

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com/)

[](http://crossmark.crossref.org/dialog/?doi=10.1016/j.eij.2019.11.002&domain=pdf)A novel crypto technique based ciphertext shifting

Farah R. Shareef

*Designation, Affiliation, Ministry of Education, Iraq*

# a r t i c l e i n f o

*Article history:*

Received 18 February 2019

Revised 14 June 2019

Accepted 12 November 2019

Available online 20 November 2019

*Keywords:* Cryptography Ciphertext Shifting algorithm HMAC

AES

# a b s t r a c t

One of the significant issues in information security areas is a hidden exchange of data. There are several techniques for this purpose such as cryptography, steganography, etc. Generally, in cryptography, the secret message content is scrambled. In another hand in steganography, the secret message is embedded inside the cover medium. In this paper, a new crypto technique-based ciphertext shifting algorithm has been designed to improve the security for our previous work that combining cryptographic and stegano- graphic. The improvement in the security of the secret message is done by changing the ciphertext value. The proposed shifting algorithm is used to rearranges the location of each character of ciphertext based on key value, in which the final ciphertext length is equivalent to encryption value but it different in value. The key strength of this method is two side one is trick the attacker from notice any change in ciphertext which is the same length as the original, so when used the common cryptanalysis will not get anything due to the original ciphertext has been changed. The second key strength of this method is that the shifting value is variable and dependent on key length. This method is inspected to be a very strong technique that can prevent common cryptography attacks such as a dictionary or brute-force attacks, etc.

© 2019 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo

University. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/](http://creativecommons.org/licenses/by-nc-nd/4.0/)

[licenses/by-nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

1. Introduction

Cryptography is the technology that uses the mathematical algorithms to encrypt and decrypt information to keep messages secure by modifying plaintext form (intelligible data) to ciphertext form (unintelligible data). The cryptography allows store the sensi- tive data or transfer it across unsafe networks (like the Internet) to ensure that it cannot be identified by anyone except the desired receiver. Steganography is the technology of hiding a message within another message while not drawing any suspicion there is any secret information exist, while only the intended recipient can decode it and got the original message [[1]](#_bookmark10).

Any cryptosystem involves Key, plaintext, ciphertext, and the encryption/decryption algorithm. Plaintext is the data or message that are is normally in a readable form (not encrypted). The

*E-mail address:* [sabahaliraq@gmail.com](mailto:sabahaliraq@gmail.com)

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

**Production and hosting by Elsevier**

encryption is the method of transforming the plaintext into cipher- text by using the key. The ciphertext outcome from the encryption process by making use of the encryption key on the plaintext. In another hand, the decryption is the method of retrieving the orig- inal plaintext back from the ciphertext form. The Key is employed information to control the cryptosystem, and it just is known by both receiver & sender [[2,3]](#_bookmark10). While the cryptography is highly effective for securing information; the cryptanalysts may possibly success to break the ciphertext by investigating the contents of ciphertext to return the plaintext [[4]](#_bookmark11). The cryptographic systems are usually categorized to a three independent dimensions, that are: Type of Operation on Plaintext, which it took place on plain- text to convert plaintext to ciphertext, the second method is ‘‘Number of Used Keys that is dependent on key utilized to be sym- metric once single key used and be asymmetric when several keys (public and private key) used, the third method is ”The Way where the plaintext is processed‘‘ in which the stream cipher runs on every plaintext element continually, and generates one element at a time, since it goes along [[2]](#_bookmark10).

In another hand, the steganography techniques are also classi- fied into three types, ‘‘Pure Steganography”, that is an approach basically uses the steganography approach simply without merg- ing other methods. It is functioning on hiding information inside cover carrier; the second type is ‘‘Secret Key Steganography” which

<https://doi.org/10.1016/j.eij.2019.11.002>

1110-8665/© 2019 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

utilizes the hybrid of the secret key cryptography approach and the steganography approach. The strategy of this type is to encrypt the information by the secret key approach and then conceal the encrypted data inside cover carrier; the third type is the ‘‘Public Key Steganography”, which is the hybrid steganography approach with the public key cryptography. The strategy of this type is to encrypt the secret information by using the public key method and then conceal the encrypted information inside the cover car- rier [[5]](#_bookmark11).

In this paper, we focused on strengthening the cryptosys- tem side of Secret Key Steganography approach. As in our previous works in [[6]](#_bookmark11) where combined text steganography with AES-HMAC cryptography, we designed a new method to increase the cryptographic strength (the cryptographic strength is measured according to the time and resources which are needing to retrieve the plaintext) via changed the value of ciphertext by used shifting algorithm that rearranged the ciphertext characters depending on key value. The result is very strong cryptography, where the ciphertext that gener- ated is very difficult to decipher without possession of the right decoding tool.

1. Previous works

In this part, we take into consideration some previous works that can assist the improvement of the proposed system and com- pared the effects when using the proposed method as an encryp- tion method than slandered methods.

For hybrid cryptography and steganography technique, we have relied on our previous strategy in [[6]](#_bookmark11), which based on hybrid cryp- tography based on AES- HMAC method. The HMAC (keyed-hash message authentication code) represents a \ method utilized to cal- culate the combination of MAC and hash function type SHA256 with a secret cryptographic key. In another hand, we utilized AES (Advanced Encryption Standard) type AES-256. Then HMAC. The proposed method has advantages of improving security rather than common standard encryption methods. In this work, we used HMAC as an encryption method and we then applied the proposed shifting algorithm on the ciphertext resultant from the HMAC method.

For text steganography strategy we have investigated some techniques concern uses of Arabic text to hiding data. The choice is to select some good techniques in hiding, from [[7], [8] and [9]](#_bookmark11).

The first selected stego method is MSCUKAT (Maximizing Steganography Capacity Using ‘‘Kashida” in Arabic Text). This work is done by Gutub and Al-Nazer [[7]](#_bookmark11), in 2010. This method is utilized as an extension character ‘‘Kashida” to hide the secret data within an Arabic text message. Their work succeeds to maximize the capacity within Arabic text cover via increased the ‘‘Kashida” within the cover which leads to a reduction in file size, without effect on the accuracy of visibility when compared than other ‘‘Kashida” methods.

The second selected text steganographic method is ‘‘Modified Fatha” that is our previous steganographic method in [[8]](#_bookmark11), that is based on the previous work ‘‘Reverse Fatha” that done by Mujtaba and Asadullah in 2016 [[10]](#_bookmark11). This method based on hiding a secret message in the text by utilized ‘‘Fatha” which is a short vowel, diagonal stroke that is written above the consonant which pre- cedes it in pronunciation). The modified ‘‘Fatha” is same symbol ‘‘Fatha” that is in the same direction of original Fatha but it is ori- ented some degree than original one, thus it very similar to original one and it hard to detect any difference, thus, it will not give any suspicious from the observer.

The Third selected text steganographic method is ‘‘Blood Group” that is our previous novel approach for text steganography

called ‘‘Blood Group” [[9]](#_bookmark11), this approach is based on the behavior of blood group to distribute ‘‘Kashida” in order to hide secret data. This method has superiority over other previous methods in many aspects involves time complexity, hiding capacity, similarity, visi- bility, and robustness.

1. Theoretical background

There are different techniques is used to give information secu- rity either by using encryption or steganography or hybrid between these techniques. AES approach, which is also called Rijn- dael, is a type of symmetric-key block cipher [[11]](#_bookmark11). In contrast to the DES method, AES technique is a nonFeistel cipher which encrypts and decrypts an information block of 128bits, by using 10, 12, or 14 rounds. The size of the key can be 128bits, 192bits, or 256bits, and the number of rounds is depending upon the key size since it enables the secret key to be extended to produce subkey for every round [[12]](#_bookmark11). In AES technique, the input and output sequences are in the same length. Regarding the AES technique, the mixing col- umn, substitution byte, key adding steps and shift rows are carried out in each encryption round to encrypt the message, however, the Mixing Column step does not involve in the last round. During the decryption, the 4 steps are used in a reverse way. As well, the inverse of mixing column step does not involve in the latest round of the decryption [[4]](#_bookmark11).

HMAC is an algorithm for cryptographic authentication, the ‘‘Keyed-Hash Message Authentication Code,” highly utilized in combination with the SHA256 cryptography. The operation is when sender and receiver sharing a message m they also share secret key k. In sender side, the sender device (such as a computer) has s = HMAC (k; m) and added (s) to (m). In another side, the receiver computes have s0 = HMAC (k; m) and then it verifies that s0 = s. In theory, any third party will never know k and as a result, cannot compute s. So, the receiver can infer that message m actu- ally came from the sender [[13]](#_bookmark11). As described in our work in [[6]](#_bookmark11), we had used AES-256 and then HMAC SHA-256, a two-step Encrypt then MAC which requires additional keys and additional overhead. The approach action takes the keys and the secret message string, in addition to an optional nonsecret payload then return back then authenticated encrypted string in addition prepended to the non- secret data with a 256bit keys randomly produced. Furthermore, it has a helper method that utilized a string password for keys creation.

1. The proposed approaches

Based on key concept pointed in [[6]](#_bookmark11), that intended to design a highly secure method to protects the secret message by combined two techniques: text steganography and AES-HMAC to get a hybrid method that can combine the many advantages from these two techniques. The main goal of the proposed system is improving the security of the crypto side by using a shifting algorithm. The ciphertext generated from AES-HMAC will be passed to shifting algorithm to rearrange ciphertext and generate shifting ciphertext and then it will be passed to stego part. [Fig. 1](#_bookmark0) illustrated the pro- posed crypto-stego system structure.

The process (shown in [Fig. 1](#_bookmark0)) is done by passing the secret message to the encryption process that used AES 256bit algo- rithm, after putting crypto key it will be generated ciphertext then the ciphertext characters will be shifting via shifting algo- rithm which changes the string value to another based on algorithm, after that the message will be embedded within stego cover through the encoding process by select one of three stego method and then got the final result which is stego message

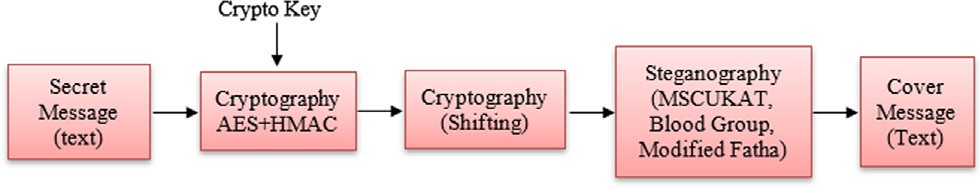


Fig. 1. Crypto-Stego system Structure.

Shifting algorithm

Input Encrypted Message (Ciphertext)

Output Encrypted Message (Shifted Ciphertext)

Step 1 Start.

Step 2 Define a fixed length Key string

Step 3 Find a total of ASCII values of characters in the key string.

Step 4 Find the length of the key string

Step 5 Right circular shift total (from step 2) by number of bits equal to the length (from step 3)

Step 6 The final value in step 4 is shifts value.

Step 7 Take the last digit of shift value.

Step 8 For the characters at odd places in the input string, take the character on the right side (circular shift) after a number of shifts equal to last digit value in step 6.

Step 9 For the characters at even places in the input string, take the character on the left side (circular shift) after a number of shifts equal to last digit value in step 6.

Step 10 Increment shift value by the number of current characters to be encrypted.

Step 11 Follow steps 6–9 till the end of the string is reached.

Step 12 Store position of each space and number of spaces at the end of the string.

Step 13 End.

The decryption process will be run in opposite to encryption/shift- ing process, the user should be entering crypto key, then the process will be first rearranged the ciphertext to its actual value then decrypt ciphertext to get the original secret message, the decryption algorithm is in follows:

Decryption algorithm

Input Encrypted shifted Secret Message. Output Secret Message.

Step 1 Start.

Step 2 Define a fixed length Key string (It should be the same as used for Encryption).

Step 3 The last character in the string is a number of spaces. From the end of the string, remove characters equal to a number of spaces.

Step 4 Put the spaces at the correct location from trimmed characters in step 7.

Step 5 Find a total of ASCII values of characters in the key string.

Step 6 Find the length of the key string

Step 7 Right circular shift total (from step 2) by number of bits equal to the length (from step 3)

Step 8 The final value in step 6 is shifts value.

Step 9 Take the last digit of shift value.

Step 10 For the characters at odd places in the input string, take the character on the left side (circular shift)

after a number of shifts equal to last digit value in step 8.

Step 11 For the characters at even places in the input string, take the character on the right side (circular shift) after a number of shifts equal to last digit value in step 8.

Step 12 Increment shift value by the number of current characters to be encrypted.

Step 13 Follow steps 8–11 till the end of the string is reached.

Step 14 Get Ciphertext value

Step 15 Get Key value

Step 16 Decrypt ciphertext by Rijndael algorithm

Step 17 End.

The algorithm can be explained in the following example: Let Key String is ‘‘Baghdad2018”, and the Secret message= ‘‘I love Iraq”, then the total of ASCII values of characters in key string = 874, as explained in [Table 1](#_bookmark1).

So, the right circular shift total by a number of bits equal to the length (Shift bits in 874 to the right side by 11), Shifting Binary value of 741(001101101010) to right will get the new binary value which is (011011010100) and it corresponding to (1748). The last digit value (L) is 8, so the shifting result can be explained in [Table 2](#_bookmark2).

The de-shifting process is run in opposite, The shifted string is ‘‘A lpth nlhi”, and the Key String used in the encryption process is ‘‘Baghdad2018”; The number of spaces = 2 and the position of space is 1,6, and the Total of ASCII values of characters in key string = 874, and the length of key string is11; thus, the Right cir- cular shift total by number of bits equal to length: Shift bits in 874 to right side by 11 (New value = 1748), so the last digit is 8. The De- shifting process result is shown in [Table 3](#_bookmark3).

Table 1

Calculating the ASCII values of characters in the key string.

|  |  |
| --- | --- |
| Char | ASCII |
| B | 66 |
| a | 97 |
| g | 103 |
| h | 104 |
| d | 100 |
| a | 97 |
| D | 104 |
| 2 | 50 |
| 0 | 48 |
| 1 | 49 |
| 8 | 56 |
| Total | 874 |

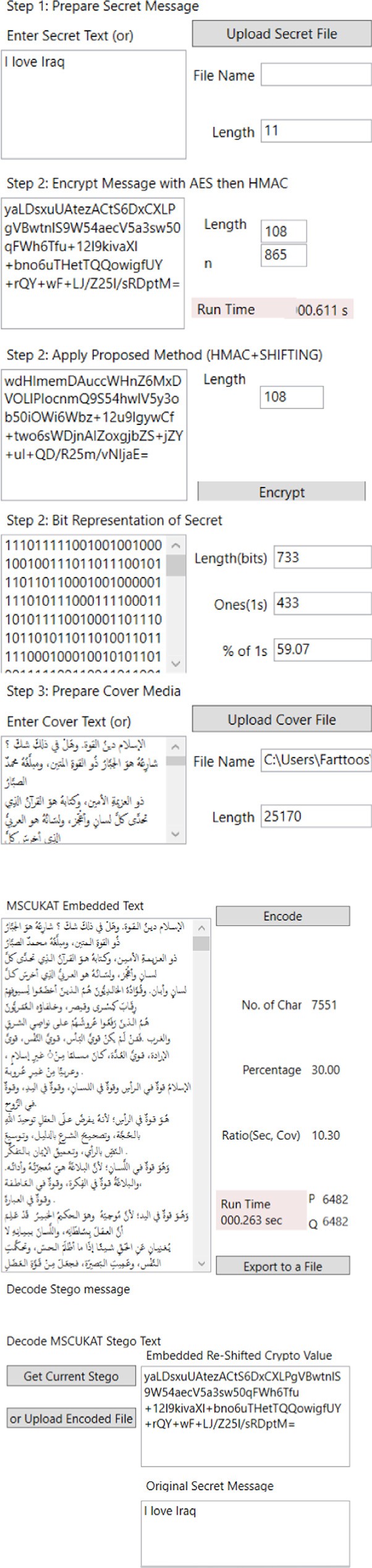
Binary value of 874 = 001101101010. Length of key string = 11.

Table 2

Shifting result for a secret message.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Encrypted Character | Last digit | Shift value | Position | Character |
| A | 8 | 1748 | Even | I |
|  | 9 | 1749 | Odd | (Space) |
| l | 0 | 1750 | even | l |
| p | 1 | 1751 | odd | o |
| t | 2 | 1752 | even | v |
| h | 3 | 1753 | odd | e |
|  | 4 | 1754 | even | (Space) |
| n | 5 | 1755 | odd | I |
| l | 6 | 1756 | even | r |
| h | 7 | 1757 | odd | a |
| i | 8 | 1758 | even | q |

Table 3



(a)

De-shifting result in a shifted secret message.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decrypted character | Last digit | Shift value | Position | Encrypted Character |
| I | 8 | 1748 | Even | A |
|  | 9 | 1749 | Odd | (Space) |
| L | 0 | 1750 | Even | L |
| o | 1 | 1751 | odd | p |
| v | 2 | 1752 | even | t |
| e | 3 | 1753 | odd | h |
|  | 4 | 1754 | even | (Space) |
| I | 5 | 1755 | odd | n |
| r | 6 | 1756 | even | l |
| a | 7 | 1757 | odd | h |
| q | 8 | 1758 | even | i |

As shown in [Table 3](#_bookmark3), the de-shifted process will return the orig- inal secret message which is I love Iraq. The process can be applied also with ciphertext generated from cryptography, in that case, we shifted characters as described in previous, with a condition that the generated shifted ciphertext must be the same length as orig- inal ciphertext were the program continue generated ciphertext and shifted cipher text until condition achieved. After that the gen- erated shifted ciphertext has been embedded within the cover message by using the required stego method. [Fig. 2](#_bookmark4) shows the GUI of software and the example of the process.

1. Results

In this section, we test the proposed approaches with five secret messages (three English one Arabic and one Persian) with a differ- ent size, and we used two Arabic text messages as stego covers. [Table 2](#_bookmark2) shows the cover and secret messages information.:

At first, we have tested the time required for encryption and decryption for both the new proposed method (AES-HMAC + SHIFT ING) and (AES-HMAC) to figured out the speed of each method. We have tested the time of encryption for five types of secret message (which has different languages and sizes as shown in [Table 4](#_bookmark5)). The testing results are shown in [Table 5](#_bookmark5).

As shown from [Table 5](#_bookmark5), the time required to encrypt a secret message by a new method (AES-HMAC + SHIFTING) is more than the previous method (AES-HMAC), this because the time required to check the length of original ciphertext and shifted ciphertext in order to achieve the condition of symmetrical character length. It also required more time as the secret message be larger. How- ever, the encryption time of proposed method (AES-HMAC + SHIFT ING) is relatively fast and it close to our previous method (AES- HMAC), so it can be no concern.

In second part we have tested the embedding of new method by used three stego method of our previous work which is (MSCUKAT, Blood Groupe and Modified Fatha) and compared it with the previ- ous method in order to determine their bit size prior to encryption, bit size after encryption, cover text size which is needed for embedding and time required for embedding process.

There are many equations can be used in the analysis the main aspects for the proposed system, which are [[7]](#_bookmark11):

* 1. Percentage Capacity: to give the percentage of cover media that has been used to hide the encrypted secret message.

## Percentage Capacity = Real Used of Cover

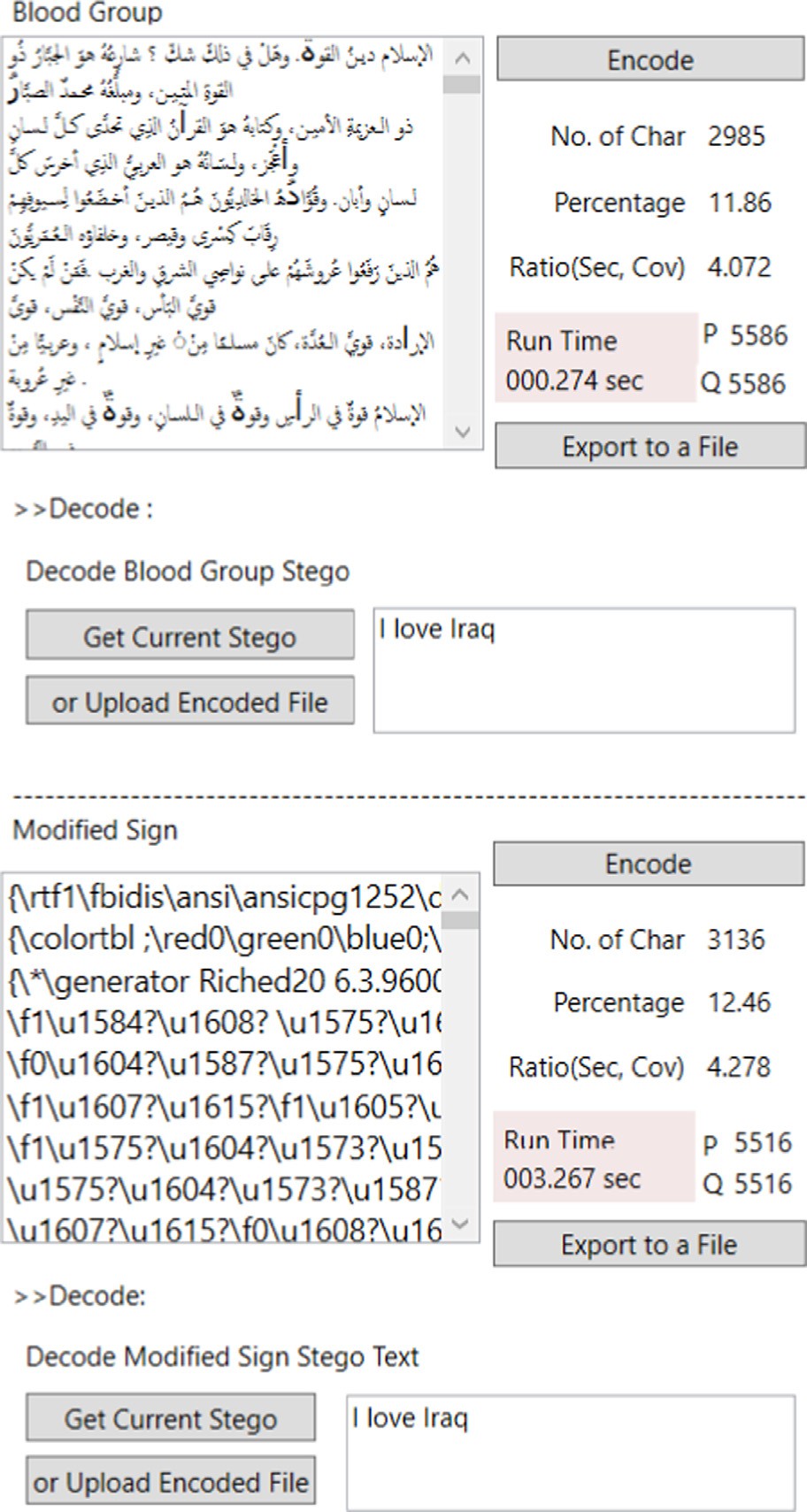
(After Encrypt + Compression) \* 100/Length of cover (1)

* 1. Hiding Capacity: to give the percentage of real use of cover

(bytes) that has been used to hide the encrypted secret mes- sage(bits).

(b-1)

Fig. 2. Crypto-Stego Process. (a) Added secret message and applying HMAC encryption and shifting process. (b-1) Text steganography type MSCUKAT, Upper, embedded the encrypted/shifted message in the cover message, lower is the decoded results (b-2) Text steganography types Blood Group & Modified sign.



(b-2)

Fig. 2 (*continued*)

## Hiding Capacity = Sec ret (bits)/Real Used of Cover (bytes) (2)

* 1. Ratio (secret/cover): It helps to know the ratio between a number of encrypted secret bits to be hidden in a number of characters in the cover media that are enough to hide such encrypted secret bits.

## Ratio (sec, cov) = real used of cover/secret(bits) (3)

Table 4

The information for secret messages, cover, and key.

|  |  |  |  |
| --- | --- | --- | --- |
| Secret/Cover/Key | Language | Characters Number | Length of the Secret Message (in a bit) |
| Secret message (s1) | English | 28 | 240 |
| Secret message (s2) | English | 346 | 1856 |
| Secret message (s3) | Persian | 966 | 4568 |
| Secret message (s4) | Arabic | 1340 | 7920 |
| Secret message (s5) | English | 3212 | 13,032 |
| Cover Message (C1) | Arabic | 187,782 | – |
| Cover Message (C2) | Arabic | 258,107 | – |
| Encryption Key | English | Baghdad2018 | – |

Table 5

The Time Required for Encryption with (AES-HMAC) and time required for the proposed method (AES-HMAC + SHIFTING).

Secret Message Time Required for Encryption (sec)

|  |  |  |  |
| --- | --- | --- | --- |
|  | (AES-HMAC) | (AES-HMAC) +SHIFTING |  |
| Secret message (s1) | 0.672 | 01.413 |  |
| Secret message (s2) | 0.683 | 01.215 |  |
| Secret message (s3) | 0.865 | 01.938 |  |
| Secret message (s4) | 0.697 | 01.590 |  |
| Secret message (s5) | 0.538 | 01.067 |  |

The testing results are shown in [Table 6–8](#_bookmark6).

1. Result when using secret message (S1) with cover (C1)
2. Result when using secret message (S2) with cover (C1)
3. Result when using small secret message (S3) with cover (C1).
4. Result when using small secret message (S4) with cover (C1).
5. Result when using small secret message (S5) with cover (C1).
6. Result when using secret message (S1) with cover (C2)
7. Result when using secret message (S2) with cover (C2)
8. Result when using small secret message (S3) with cover (C2).
9. Result when using small secret message (S4) with cover (C2).
10. Result when using small secret message (S5) with cover (C2).

As shown from applying text steganographic which is results details in tables (6–15). The number of characters of cover message needs for embedding encrypted secret message (i.e. cover size) that generated by used (AES-HMAC + SHIFTING) is same or very close to previous method (AES-HMAC) for all three stego methods (MSCUKAT, blood Group and Modified Fatha) where the cover per- centage capacity and hiding capacity are relatively the same for all secret messages. This is due to that the new method modified the AES-HMAC ciphertext and the length of output is the same as the input. For embedding time, the new method is relatively slower than the previous one, this is related to shifting process and

Table 6

The real used of characters, hiding capacity and time for the secret message (S1) with cover (C1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.016 | 15.61 | 1.99 | 6.40 | 3746 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.018 | 15.61 | 1.99 | 6.40 | 3746 | (AES-HMAC) |  |
| 0.029 | 20.383 | 2.61 | 4.90 | 4892 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.021 | 20.383 | 2.61 | 4.90 | 4892 | (AES-HMAC) |  |
| 289.979 | 18.888 | 2.41 | 5.29 | 4533 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 283.460 | 18.888 | 2.41 | 5.29 | 4533 | (AES-HMAC) |  |

Table 7

The real used of characters, hiding capacity and time for the secret message (S2) with cover (C1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.023 | 7.26 | 7.18 | 13.77 | 13,478 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.022 | 7.28 | 7.19 | 13.73 | 13,509 | (AES-HMAC) |  |
| 0.031 | 9.426 | 9.32 | 10.60 | 17,494 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.035 | 9.445 | 9.33 | 10.58 | 17,529 | (AES-HMAC) |  |
| 283.713 | 8.801 | 8.70 | 11.36 | 16,334 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 274.633 | 8.817 | 8.71 | 11.34 | 16,365 | (AES-HMAC) |  |

Table 8

The real used of characters, hiding capacity and time for the secret message (S3) with cover (C1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.054 | 11.34 | 27.59 | 8.820 | 51,790 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.056 | 11.33 | 27.56 | 8.825 | 51,758 | (AES-HMAC) |  |
| 0.091 | 14.705 | 35.77 | 6.8004 | 67,172 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.091 | 14.700 | 35.76 | 6.802 | 67,149 | (AES-HMAC) |  |
| 287.078 | 13.745 | 33.44 | 7.275 | 62,785 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 266.934 | 13.738 | 33.42 | 7.278 | 62,757 | (AES-HMAC) |  |

Table 9

The real used of characters, hiding capacity and time for the secret message (S4) with cover (C1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.076 | 9.16 | 38.64 | 10.481 | 75,562 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.080 | 9.17 | 38.69 | 10.902 | 72,646 | (AES-HMAC) |  |
| 0.133 | 11.884 | 50.12 | 8.414 | 94,124 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.114 | 11.900 | 50.19 | 8.403 | 94,250 | (AES-HMAC) |  |
| 897.900 | 11.106 | 46.84 | 9.003 | 87,963 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 275.237 | 11.122 | 46.91 | 8.99 | 88,089 | (AES-HMAC) |  |

Table 10

The real used of characters, hiding capacity and time for the secret message (S5).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.091 | 7.39 | 51.27 | 13.53 | 96,277 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.095 | 7.38 | 51.21 | 13.55 | 96,158 | (AES-HMAC) |  |
| 0.141 | 9.583 | 66.51 | 10.43 | 124,885 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.157 | 9.571 | 66.42 | 10.44 | 124,729 | (AES-HMAC) |  |
| 322.797 | 8.957 | 62.16 | 11.16 | 116,722 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 308.238 | 8.947 | 62.09 | 11.17 | 116,595 | (AES-HMAC) |  |

Table 11

The real used of characters, hiding capacity and time for the secret message (S1) with cover (C2).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.024 | 15.19 | 1.41 | 6.58 | 3646 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.021 | 15.19 | 1.41 | 6.58 | 3646 | (AES-HMAC) |  |
| 0.030 | 19.329 | 1.80 | 5.17 | 4639 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.028 | 19.329 | 1.80 | 5.17 | 4639 | (AES-HMAC) |  |
| 512.699 | 21.729 | 2.02 | 4.602 | 5215 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 490.830 | 21.729 | 2.02 | 4.602 | 5215 | (AES-HMAC) |  |

comparing length strategy (the strategy that makes encryption repeated until the length of encryption text is equal to shifting one). However, the encryption time for both are very fast and it not exceed 2 s ([Tables 9–14](#_bookmark7)).

* Time required after Encryption ((AES-HMAC) &(AES-HMAC + SHIFTING)) and steganography for 5-secret messages and 2-

covers

For [Table 15](#_bookmark8), the overall time needed for encryption and decod- ing of proposed method (AES-HMAC + SHIFTING) is very close to the previous method (AES-HMAC) for all three stego methods (MSCUKAT, blood Group and Modified Fatha). In addition, the pro- posed method is run fast and no increases in cover size, thus, this method can be used to have strong encryption with relatively same processing time and embedding size which is the big advantage offer improvement to our previous method ([Table 16](#_bookmark9)).

Table 12

The real used of characters, hiding capacity and time for the secret message (S2) with cover (C2).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.028 | 7.08 | 5.09 | 14.119 | 13,145 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.036 | 7.11 | 5.11 | 14.062 | 13,198 | (AES-HMAC) |  |
| 0.039 | 8.922 | 6.42 | 11.208 | 16,559 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.034 | 8.958 | 6.44 | 11.163 | 16,626 | (AES-HMAC) |  |
| 491.556 | 9.968 | 7.17 | 10.032 | 18,500 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 484.882 | 10.008 | 7.2 | 9.991 | 18,575 | (AES-HMAC) |  |

Table 13

The real used of characters, hiding capacity and time for the secret message (S3) with cover (C2).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.077 | 11.1 | 19.65 | 9.005 | 50,723 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.097 | 11.1 | 19.65 | 9.005 | 50,723 | (AES-HMAC) |  |
| 0.095 | 13.991 | 24.76 | 7.147 | 63,909 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.131 | 13.991 | 24.76 | 7.147 | 63,909 | (AES-HMAC) |  |
| 543.786 | 15.577 | 27.57 | 6.419 | 71,155 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 593.592 | 15.577 | 27.57 | 6.149 | 71,155 | (AES-HMAC) |  |

Table 14

The real used of characters, hiding capacity and time for the secret message (S4) with cover (C2).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.096 | 9 | 27.61 | 11.113 | 71,264 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.114 | 9 | 27.61 | 11.113 | 71,264 | (AES-HMAC) |  |
| 0.127 | 11.328 | 34.76 | 8.827 | 89,721 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.138 | 11.328 | 34.76 | 8.827 | 89,721 | (AES-HMAC) |  |
| 538.06 | 12.619 | 38.72 | 7.924 | 99,942 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 523.809 | 12.619 | 38.72 | 7.924 | 99,942 | (AES-HMAC) |  |

Table 15

The real used of characters, hiding capacity and time for the secret message (S5) with cover (C2).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time(sec) | Ratio (sec, cov.) | Cover percentage Capacity | Hiding capacity | No. of char (real used) | Encryption Method | Stego Method |
| 0.099 | 7.24 | 36.57 | 13.806 | 94,387 | (AES-HMAC) + SHIFTING | MSCUKAT |
| 0.109 | 7.24 | 36.57 | 13.806 | 94,387 | (AES-HMAC) |  |
| 0.182 | 9.123 | 46.07 | 10.96 | 118,897 | (AES-HMAC) + SHIFTING | Blood Group |
| 0.178 | 9.123 | 46.07 | 10.96 | 118,897 | (AES-HMAC) |  |
| 561.468 | 10.161 | 51.3 | 9.842 | 132,412 | (AES-HMAC) + SHIFTING | Modified Fatha |
| 549.54 | 10.161 | 51.3 | 9.842 | 132,412 | (AES-HMAC) |  |

Table 16

The real used of characters, hiding capacity and time for the secret message (S5) with cover (C2).

Secret messages Covers AES-HMAC (AES-HMAC) + SHIFTING S1 C1 0.018 ~ 283.460 0.016 ~ 289.979

C2 0.021 ~ 490.830 0.024 ~ 512.699

S2 C1 0.022 ~ 274.633 0.023 ~ 283.713

C2 0.036 ~ 484.882 0.028 ~ 491.556

S3 C1 0.056 ~ 266.934 0.54 ~ 287.078

C2 0.097 ~ 593.592 0.77 ~ 543.786

S4 C1 0.080 ~ 275.237 0.076 ~ 897.900

C2 0.114 ~ 523.809 0.096 ~ 538.060

S5 C1 0.095 ~ 308.238 0.091 ~ 322.797

C2 0.109 ~ 549.540 0.099 ~ 561.468

1. Conclusions

In this work, we have proposed a new crypto method that used AES-HMAC algorithm then shifting the ciphertext by used shifting algorithm in order to be embedded within the cover text so that the attacker cannot crack it by used cryptanalysis. For cryptogra- phy, we have been used AES256-HMAC algorithm bit then applied

a shifting algorithm. The ciphertext generated has passed to shift- ing algorithm to rearrange ciphertext and generate shifting chip- per. The shifted cipher then passes to stego part. several conclusions can be attracted to this work; the mainly important points are:

1. The crypto-stego combination techniques satisfy the require- ments such as high security and robustness between sender and receiver.
2. The main advantage of this system is that the method used a combination of AES256-HMAC then shifting the ciphertext by used shifting algorithm which is very secure and the stego tech- niques which are very hard to detect.
3. Because of the key length is (256bits) and the number of cycles

(14) it takes a long time for attacker malware to do a dictionary attack. Furthermore, the use of shifting algorithm that shifting the ciphertext to different characters it makes impossible for the attacker to decrypt message because it needs to rearrange to its original ciphertext.

1. This new method can be run at the low computational require- ment, but it will be fast as the computer be stronger. In this work, the test is done by a relatively lower level laptop which

has a Core i3 CPU of speed (1.8GH) and it has (4 GB) of RAM. The encryption process that used combination of AES-HMAC encryption and shifting has need more time which needs (0.016–289.979) second for short message and about (0.023– 283.713) second for medium message, and (0.091–322.797) for large message while for previous method it required (0.018–283.460), (0.022–274.633), and (0.095–308.238)

respectively, but it still relatively lower in time.

REFERENCES

1. Al-Gailani MF. ‘‘Advanced Cryptographic System: Design, Architecture and FPGA Implementation”, Ph.D. Thesis, School of Electrical, Electronic and Computer Engineering, Newcastle University, England, United Kingdom, 2012, pp.
2. [Babu KR, Kumar SU, Babu AV. A survey on cryptography and steganography](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0010) [methods for information security. Int J Comput Appl 2010;12(2):13–7](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0010).
3. Madhuravani B, Bhaskara RP, Lalith SRP. ‘‘Steganography Techniques: Study & Comparative Analysis”, International Journal of Advanced Scientific and Technical Research Issue 4 Vol.2, ISSN 2249-9954, 2014.
4. Rabah K. Theory and implementation of data encryption standard a review. Inf Technol J 2005;4(4):307–25. doi: <https://doi.org/10.3923/itj.2005.307.325>.
5. [Saleh ME, Aly AA, Omara FA. Data security using cryptography and](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0025) [steganography techniques. Int J Adv Comput Sci Appl (IJACSA) 2016;7](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0025) [(6):390–7](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0025).
6. [Malalla S, Shareef FR. Improving hiding security of arabic text steganography](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0030) [by hybrid aes cryptography and text steganography. Int J Eng Res Appl 2016;6](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0030) [(6):60–9](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0030).
7. [Gutub AA, Al-Nazer AA. High Capacity Steganography Tool for Arabic Text](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0035) [Using ‘‘Kashida”.](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0035) [The ISC Int’l J Inf Secur 2010;2(2):107–18](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0035).
8. [Malalla S, Shareef FR. A new modified fatha method for arabic text](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0040) [steganography hybrid with aes encryption. IOSR 2016;18(5):37–45](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0040).
9. [Malalla S, Shareef FR. Novel approach for arabic text steganography based on](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0045) [the ‘‘bloodgroup”](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0045) [text hiding method. Eng Technol Appl Sci Res 2017;7](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0045) [(2):1482–5](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0045).
10. [Mujtaba SM, Asadullah S. A Novel Text Steganography Technique to Arabic](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0050) [Language Using Reverse Fatha”.](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0050) [PJETS 2011;1(2):106–13](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0050).
11. [Alanazi HO, Zaidan BB, Zaidan AA, Jalab HA, Shabbir M, Al-Nabhani Y. New](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0055) [comparative study between DES, 3DES, and](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0055) AES””[. J Comput 2010;2(3):152–7](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0055).
12. [Mushtaq MF, Jamel S, Disina AH, Pindar ZA, Shakir NSA, Deris MM. A Survey on](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0060) [the Cryptographic Encryption Algorithms”.](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0060) [Int J Adv Comput Sci Appl (IJACSA)](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0060) [2017;8(11):333–44](http://refhub.elsevier.com/S1110-8665(19)30034-9/h0060).
13. Beringer L, Petcher A, Ye KQ, Appel AW. Verified Correctness and Security of OpenSSL HMAC”, 24th Usenix Security Symposium, August 2015, pp.1–15.