Journal of Computer Science & Computational Mathematics, Volume 8, Issue 3, September 2018
DOI: 10.20967/jcscm.2018.03.002

Compression Multi-Level Crypto Stego Security of Texts Utilizing Colored Email Forwarding

Abrar Alsaidi[1], Khulood Al-lehaibi[2], Huda Alzahrani[1], Manal AlGhamdi[3], Adnan Gutub[1*]

[1]Computer Engineering Department, Umm Al-Qura University, Makkah, Saudi Arabia
[2] Programming department; Technical College for Girls in Taif; Technical and Vocational Training Corporation
[3]Computer Sciences Department, Umm Al-Qura University, Makkah, Saudi Arabia
[*]Corresponding author email: aagutub@uqu.edu.sa

***Abstract*:** Encrypted messages will draw the attention of third parties, leading attempts to break the original messages and disclose it. Obscuring facts and communications can be achieved by concealing the message using a technique called steganography, which hides the presence of messages and secrets from the public. Several digital mediums, such as texts, images, audio and video, can be used as cover media for digital steganography, with text steganography being our focus in this research. Utilizing text as a target medium is complex due to the lack of available redundant data-bits within text file. Steganography can be used together with cryptography to offer higher level of privacy and security over the communication channel. This paper presents a multi-level security method that takes the sensitive text and compresses it through the LZW compression algorithm and then encrypts it using the Advanced Encryption Standard (AES) algorithm in order to be stego-hidden in forward email colored platform. Implementation results show that this proposed multi-level scheme is giving motivating security, high-capacity, and practical performance text crypto stego interesting technique showing attractive contribution.

***Keywords*:** steganography, text steganography, LZW compression, capacity, security, color mapping, DHKE, AES.

1. Introduction

With increasing attacks on information exchanged over the Internet, data hiding has become critically necessary, especially private data. Thus, there is a need to have a solution that can protect sensitive data, and data hiding is the process of distributing secret data into media (text, image, sound, and video) and sending these data over a public network. Cryptography and steganography are well-known widely used techniques that manipulate messages in order to cipher or/and hide their existence. Cryptography, the art of secret writing, protects information by transforming it into an unreadable format. It is useful for achieving confidential transmission over public network. The original text or "plaintext" is converted into a ciphertext via cryptography algorithm where only those who have secret key can decrypt the cipher text back into the plaintext [1]. Cryptography systems can be classified into two parts: Symmetric and Asymmetric [2], where both can be implemented separately or merged as hybrid implementation as detailed in [3]. The two crypto classifications can be shown in Figure 1, below:

- Symmetric-key systems: systems that require a single key (i.e., a password) that both the sender and the receiver use.
- Public-key systems: systems that use two keys, a public key known to everyone and a private key known only to the owner.

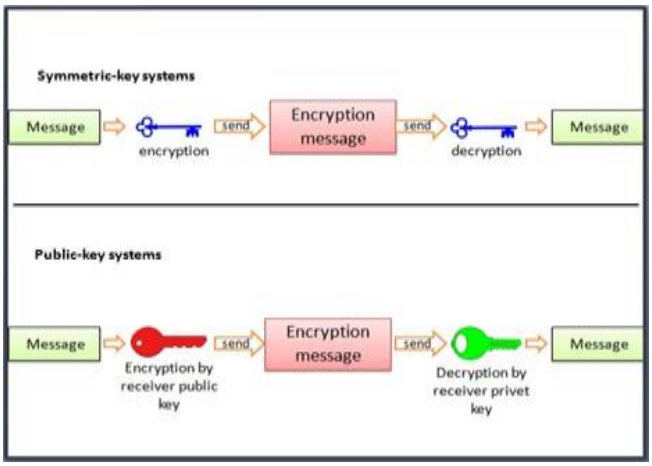

Figure 1. Cryptography systems classification

Cryptography differs from steganography [4], which is the art and science of hiding information by putting secret content in cover media (i.e. text [5], image [6], audio [7], and video [8]), so that its presence cannot be detected. The steganography is not to keep others from knowing the hidden information, but it is to keep others from drawing suspicion that the information even exists [9]. If a steganography method causes someone to suspect that there is secret information in a carrier medium, then the stego method has failed [10].

Although cryptography and steganography are different and distinct [11], these two can be treated as twin sisters of secret communications. The former scrambles a message so it cannot be understood, the later hides the message so it cannot be seen or ever-known.

In text-based stenographic methods, the text is used as cover media to hide the secret data, as shown in Figure 2. Text steganography is one of the hardest areas of data hiding [12] since the human eye is very susceptible to any change between the original and the modified texts (stego-texts) and it can be easily detected [9].

There are several methods for performing text steganography such as:

- Syntactic Method: The method uses syntax or format of the text to hide data. In this method, punctuation is inserted in cover text such as full stop (.), comma (,) etc to hide data [13].
- Semantic Method: It stands for the meaning of something, known synonym of the word for hiding data. A synonym

substitution may hide a single bit or multiple bits of secret information, but sometimes it may alter the actual meaning of the text file [14].

- Text Abbreviation Method: This method is used for hiding data by replacing a target word with its acronym, e.g. replacing 'as soon as possible'with ASAP etc. This technique is used in SMS and social networking applications. Mohammad Shirali-Shahreza and M.Hassan Shirali-Shahreza from Iran proposed a method of substituting a word with its acronym [15]. A much smaller amount of data can be hidden using this method.
- White Spaces Method: This method works by inserting spaces in a cover text to hide the message. White spaces can be inserted at the end of line, paragraph or between words or sentences. The drawback of this method is that the insertion of spaces increases the size of the text file and a much smaller amount of data can be hidden [16].

The main objective of this work is to obtain high capacity and security by increasing the amount of secret data that is hidden in the cover medium by using stego keys and cryptography used in a forward email platform involving email IDs and a cover message, all utilized to hide the secret data. The email message that is chosen contains wishes, messages, jokes, etc. The message is made colorful while hiding the secret data according to the secret data bit stream. Therefore, there is no need to modify the semantics and/or syntax of the cover message content [10].

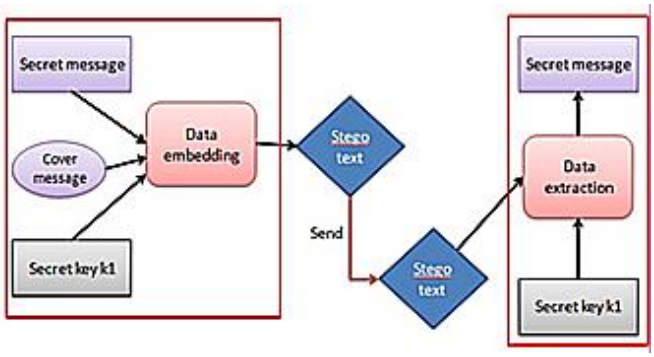

Figure 2. Stenographic methods

In the proposed technique [9] for capacity increment, LZW data compression is used for compressing the secret data. LZW is directly applied to the secret data to get a good compression ratio and to decrease the computational complexity. At first, LZW reads the data and transforms it into fixed length code words. The code words should be as short as possible so as to achieve maximum compression. For the security improvement, stego-keys are employed. The stego keys are divided into two categories, namely the constructed stego key, which is used during the embedding phase, and the previously constructed global stego key, which is shared between the sender and the receiver beforehand.

For hiding the secret data in the email message, the color coding table is used as shown in Figure 3. The table contains a set of colors and each color has a bit value of 0 or 1. The message is colored according to the bit values of the bit stream using color coding table. Thus, the secret data are embedded in the message even without modifying the contents of the message [9].

S.NO	Color Name	Color	Bit Representation
1	Red		0
2	Green		0
3	Blue		0
4	Pink		0
5	Black		1
6	Yellow		1
7	Magenta		1
8	Orange		1

Figure 3. Color table

Our proposed multi-level system will improve this technique by applying encryption to the compressed data to obtain a high level of security, and then stego-keys are employed. The proposed method can be used by many organizations that rely on email to send and receive data, especially confidential data about employees or clients, so that the organization can ensure the confidentiality and privacy of those data. Through this method, confidential data can be sent to patients from their hospitals. Also, banks can use it for more than one purpose, i.e. transferring secret info as well as authenticating customers' accounts, as so many other organizations can.

This paper is organized as follows. Section 2 presents brief definitions and theoretical background of the related works. Section 3 describes the proposed multi-level technique and its improvements. Section 4 includes a comparison of the work to others. Finally, Section 5 concludes the paper.

2. Related Works

To make the paper self-contained, the involved linked algorithms are remarked. The used crypto algorithms: DHKE and AES, are discussed in the following subsection, i.e. Section 2.1. The crypto concepts are followed by the text steganography concerned to our work as discussed in Section 2.2. We start by briefly reviewing the related modulo operations and basic number theory definitions.

Definition 1: Modulo Operation [17] Let $a, r, m \in Z$ (where Z is a set of all integers) and $m \geq 0$.

We write $a \equiv r \ mod \ m$, if m divides r. m is called the modulus and r is called the remainder.

Definition 2: Greatest common divisor [2]. The greatest common divisor of a and b is the largest positive integer that divides both a and b, that is, integer c such that there exists k_1, $k_2 \in Z$: $a = k_1 c$ and $b = k_2 c$. It is denoted by $GCD(a, b)$ or (a, b). If $(a, b) = 1$, a and b are relatively prime or coprime.

Definition 3: Group [18]. A group is a set of elements G together with an operation that combines two elements of G. A group has the following properties:

1. The group operation + is closed. That is, for all $a,b \in G$, it holds that $a + b = c \in G$.
2. The group operation is associative. That is $a + (b + c) = (a + b) + c$ for all $a, b, c \in G$.
3. There is an element $1 \in G$, called the neutral element (or identity element), such that:
   $a + 1 = 1 + a = a$ for all a ∈ G.
   For each a ∈ G there exists an element a-1∈ G, called the inverse of a, such that: a + a-1 = a-1+ a = 1.

4. A group G is abelian (or commutative) if $a + b = b + a$ for all $a, b \in G$.

Definition 4: Field [10] A field F is a set of elements with the following properties:

1. All elements of F form an additive group with the group operation + and the neutral element 0.
2. All elements of F except 0 form a multiplicative group with the group operation × and the neutral element 1.
3. When the two group operations are mixed, the distributivity law holds, i.e., for all $a, b, c \in F$ $a(b+c) = (ab)+(ac)$.

Definition 5: Let p be a prime. The integer ring Z_p is denoted as $GF(p)$ and is referred to as a prime field, or as a Galois field with a prime number of elements [19]. All nonzero elements of $GF(p)$ have an inverse. Arithmetic in $GF(p)$ is done *modulo p*.

Definition 6: Primitive root modulo [2]. Given a prime number p, the group $(Z^*_p,.)$ is always a cyclic group that is generated by a single element. A generator of this cyclic group is called a primitive root modulo p or a primitive element of Z^*_p.

2.1 Cryptography Algorithms

Several crypto algorithms can be utilized for this multi-level security mission [20]. The work preferred to utilize DHKE cryptography as well as AES cryptography, since they both are suitable for adoption in our forward-email-platform text stego system.

DHKE algorithm

The algorithm is shown in Figure 4 and can be summarized as follow:

1. Alice and Bob agree on a prime number p and a base g.
2. Alice chooses a secret integer a, then sends to Bob $A = g^a \ mod \ p$
3. Bob chooses a secret integer b, then sends to Alice $B = g^b \ mod \ p$
4. Alice computes $K_{AB} = B^a \ mod \ p$
5. Bob computes $K_{AB} = A^b \ mod \ p$
6. Alice and Bob now share a secret that can be used as their symmetric AES key.

AES algorithm

The AES procedure operates on a *128-bit* block of information and performs *Nr - 1* loop times. Key length can be *128, 256* or *192 bits* in extent respectively. The first and last rounds differ from the rest in that there is an additional AddRoundKey transformation at the start of the first round and no MixColumns transformation executed in the last round. Each block of data is inputted in the form of a *4×4* matrix called the state.

The round operations (AES core operation) is shown in Figure 5 and summarized as follows:

- Byte substitution transformation. The byte substitution can be viewed as a row of 16 parallel S-Boxes, each with 8 input and output bits. The S-Box is usually realized as a 256-by-8-bit lookup table with fixed entries, so just lookup for the value needed.

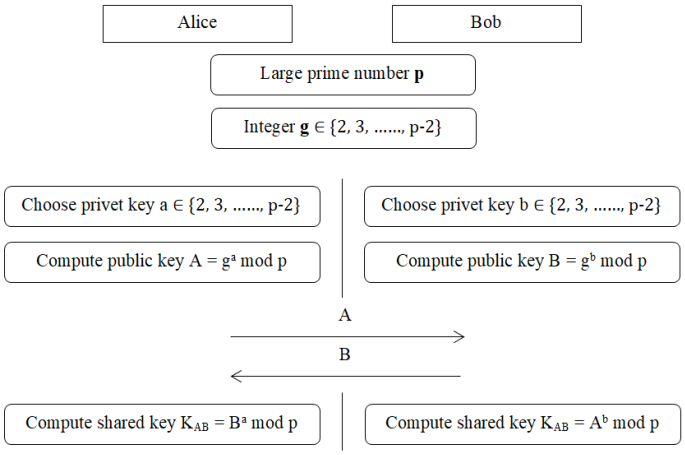

Figure 4. DHKE algorithm

- ShiftRows transformation. Rows of state are cyclically left shifted over dissimilar offsets in ShiftRows transformation. Row 0 is not shifted; row 1 is shifted 1 byte to the left; row 2 is shifted 2 bytes to the left and row 3 is shifted 3 bytes to the left.
- MixColumns transformation. Columns of state are deliberated as polynomials over Galois Field *(GF)* and multiplied by a fixed 4×4 matrix modulo $c(x)$ using a fixed polynomial c(x), given by:
  $c(x) = x^8 + x^4 + x^3 + x + 1$.

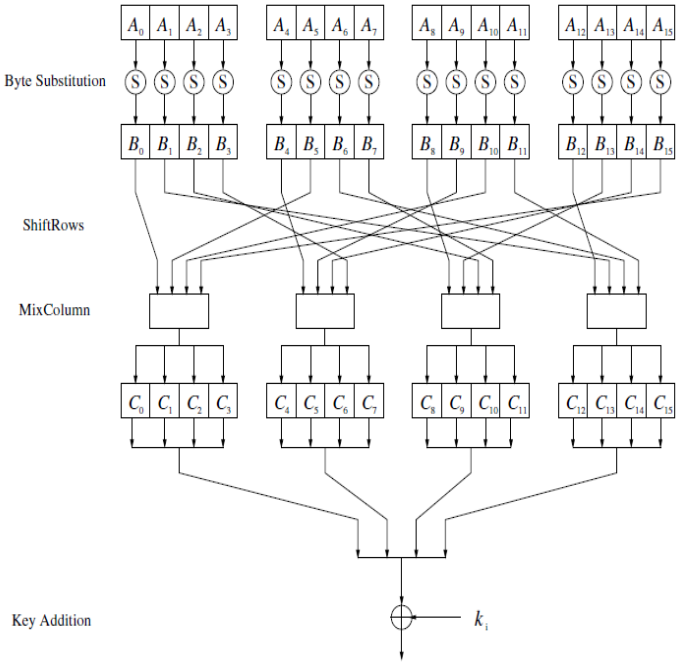

Figure 5. AES core operation [10]

- AddRound Key transformation. Round Key is additional to State and results from the operation of MixColumns transformation through a simple bitwise XOR operation. The Round Key of all rounds is derived from the primary key with Key Expansion procedure.

2.2 Text Steganography Scheme

In [9], capacity and security issues have been considered. The LZW algorithm is applying a directory on the secret text and the obtained bit stream is hidden into email IDs and the

message of the email. The algorithm has two phases: the embedding phase and the extracting phase. The embedding (embedding the secret text into the email page) algorithm works as summarized shown in Figure 6 and as follows:

- The algorithm takes the secret text and compresses it through the LZW compression algorithm.
- The obtained LZW code is converted into a bit stream.
- The number of characters in the cover message without space is counted, and the same number from is extracted from the bit stream.

```
Embedding algorithm ( secret text , cover message )
        SM := secret text
        CM := cover message
        LZW( SM ) : ComM                    //secret text compression
        BitStream := ComM in the binary form
        NC := number of characters in cover message without space.
        Print CM using the color-coded table to color each cover text element in a gradual manner
        C := NC bits from BitStream
        G := rest bits from BitStream
        divide the rest bit stream into 12 bit groups
        N := G.length/12
        For i = 0 .. N do
        G1 := 9 bits from the group
        G2 := 3 bits from the group
        X := (G1)10 / 26
        Y := (G1)10 MOD 26
        Z := (G2)10
        A := Convert the value of X to a first letter in the email id by applying the Latin Square.
        B := Convert the value of Y to a second letter in the email id by applying the Latin Square.
        C := Convert the value of Z to email extension by applying email extension table.
        Ids = A || B || C
        Print Ids
        end for
```

Figure 6. Embedding algorithm

- Using the color-coded table (shared with the receiver) each cover text element is colored using bits extracted from the bit flow. Colors are used in a gradual manner.
- After that, the rest of the bit stream is divided into 12 bit groups (if the bits in the bit stream are not in the multiple of 12 then the required number of zeros are appended to make it the nearest multiple of 12). Each group is partitioned into 9 bits, which is called G1, and 3 bits, which is called G2.
- *X*, *Y*, and *Z* are calculated as follows:
- $X = (G1)_{10} / 26$
- $Y = (G1)_{10}$ *MOD 26*
- $Z = (G2)_{10}$
- The value of *X* and *Y* are converted to letters by applying the Latin Square as shown in Figure 7.

These letters are mapped to one e-mail address as that X is the first character in the email ID and Y is the second character, while *Z* represents the e-mail extension (see Figure 8) (the sender and the receiver have a common set of email IDs and email extensions as they also had a color table).

Rows	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
3	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
4	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
5	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
6	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
7	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
8	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
9	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
10	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
11	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
12	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
13	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
14	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
15	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
16	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
17	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
18	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
19	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
20	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
21	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
22	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
23	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
24	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
25	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
26	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 7. Latin Square table [9]

This previous step is used for all 12-bit group streams to generate all email IDs. Thus, using the cover message and email ISs set (which is called the stego-key), the secret message is transmitted in the form of a forward email platform as shown in Figure 9.

The extracting (extract the secret text from email page) algorithm works as shown in Figure 10 and as follows:

- Secret data bits are extracted from the cover text to obtain C (the first bits of the message), by using the color table.

Email Extinctions	Bit Representation
gmail.com	000
hotmail.com	001
yahoo.com	010
rediffmail.com	011
btinternet.com	100
aol.com	101
msn.com	110
verizon.net	111

Figure 8. Email extensions table

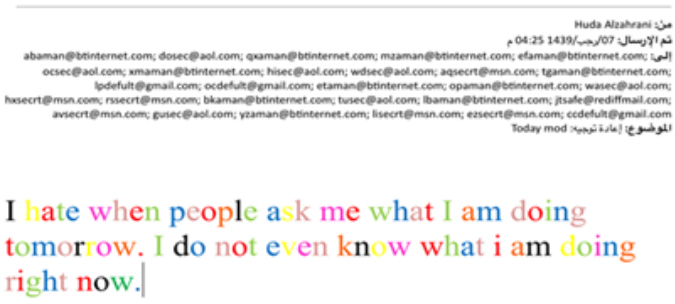

Figure 9. Forward email platform

Extracting algorithm (email ids , colored cover message)
Ids := email ids
CCM := colored cover message
C := secret data bits from colored cover message using the color table
N := number of email ids
For i = 0 .. N
X := Convert the first letter in the email id to its value via Latin Square.
Y := Convert the second letter in the email id to its value via Latin Square.
Z := Convert the email extinction to its value via email extinction table.
G1 := $(X*26+Y)_2$
G2 := $(Z)_2 = (4)_{10}$
G := G1 || G2
end For
BitStream := C || G
ComM := BitStream in String form
LZW (ComM) : SM
Print SM // print secret message

Figure 10. Extracting algorithm

- X, Y and Z are found from the email IDs by employing Latin Square to get X and Y, while Z is obtained from email extensions.
- C and G are concatenated to obtain a compressed bit stream of the secret text.
- The last step involves applying LZW to decompress the bit stream to get the original secret data.

3. Proposed Method

In our proposed multi-level security system, to make the algorithm secure we added the encryption in the embedding phase and decryption in the extracting phase similar in principle to the work presented in [21]. We encrypted the data after it was compressed so that fewer data were assumed sensitive to be encrypted and the encryption was fast. Compression was used to gain capacity, i.e. size reduction, then encryption was used to make it secure. Therefore, the embedding result will be difficult to deduce. We used the DHKE and AES algorithms. DHKE is a one-round-trip key exchange algorithm: the sender chooses his private key, computes his public key, and sends his public key to the receiver. The receiver also chooses his private key, computes his public key, and sends his public key to the sender. Then the sender and receiver compute the shared key separately, similar in principle to Diffie-Hellman key exchange agreement [2]. So, the sender computes the shared key, encrypts and sends the whole lot to the receiver, and the receiver computes the key and decrypts, as we explained earlier. This is compatible with a one-shot communication system, assuming a pre-distribution of the public key, i.e. it works with emails. Therefore, the DHKE algorithm allows two parties to create a shared secret key over an insecure communications channel, and this shared key can be used after that for symmetric encryption.

We used the AES algorithm in symmetric encryption, AES is required by the latest US. and international standards. AES is a secure algorithm (it is less susceptible to cryptanalysis vs. all other symmetric algorithms) and it is fast in terms of both hardware and software, and overall relative simplicity of implementation. The simplicity comes from the fact that the operations depend on XOR, shifting and substitution, as we explained earlier. The proposed algorithm works as shown in Figure 11 and summarized as follows:

- The algorithm takes the secret text and compresses it through the LZW compression algorithm.

- The compression result is encrypted using the AES algorithm.
- The obtained encrypted message is converted into a bit stream.
- The number of characters in the cover message is counted without spaces and the same number is extracted from the bit stream.
- Then the color-coded table is used to color each cover text element by bits extracted from the bit flow, as shown in Figure 12.
- The rest of the bit stream is divided into 12 bit groups (if the bits in the bit stream are not in a multiple of 12 then the required number of zeros is appended to make it the nearest multiple of 12). Each group is partitioned into *9* bits called *G1* and *3* bits called *G2*.
- Calculate *X*, *Y*, and *Z* as follows:
  $X = (G1)_{10} / 26$
  $Y = (G1)_{10}$ *MOD* 26
  $Z = (G2)_{10}$
  Convert the value of *X* and *Y* to the letters by applying the Latin Square. These letters are mapped to one e-mail address as *X* is the first character in the email ID, *Y* is the second character, and *Z* represents the e-mail extension.
  The previous step is used for all 12-bit group streams to generate all email IDs.
- Using the cover message and email IDs set (called a stego key), the secret message is transmitted in the form of a forward mail platform.

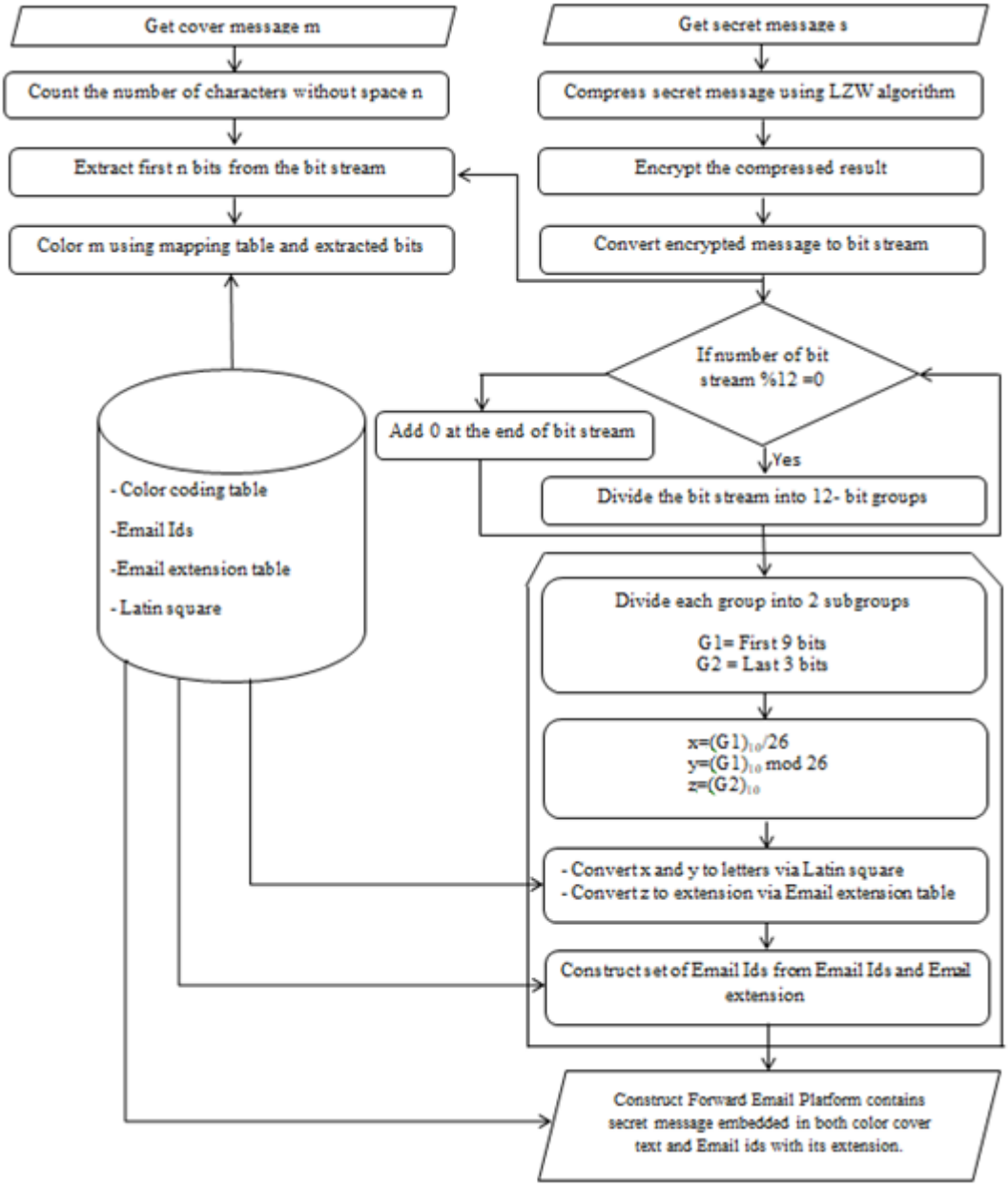

Figure 11. Embedding phase

Figure 12. Email with colors

The extracting (extract the secret text from the email page) algorithm works as shown in Figure 13 and summarized as follows:

1. Secret data bits are extracted from the cover text to obtain C, by using the color table.
2. *X*, *Y* and *Z* are found from the email IDs by employing Latin Square to get *X* and *Y*, and *Z* is obtained from email extensions.
3. *C* and *G* are concatenated to obtain the encrypted bit stream of the secret text.
4. The result is decrypted to obtain the compressed message.
5. The message is decompressed to get the secret message.

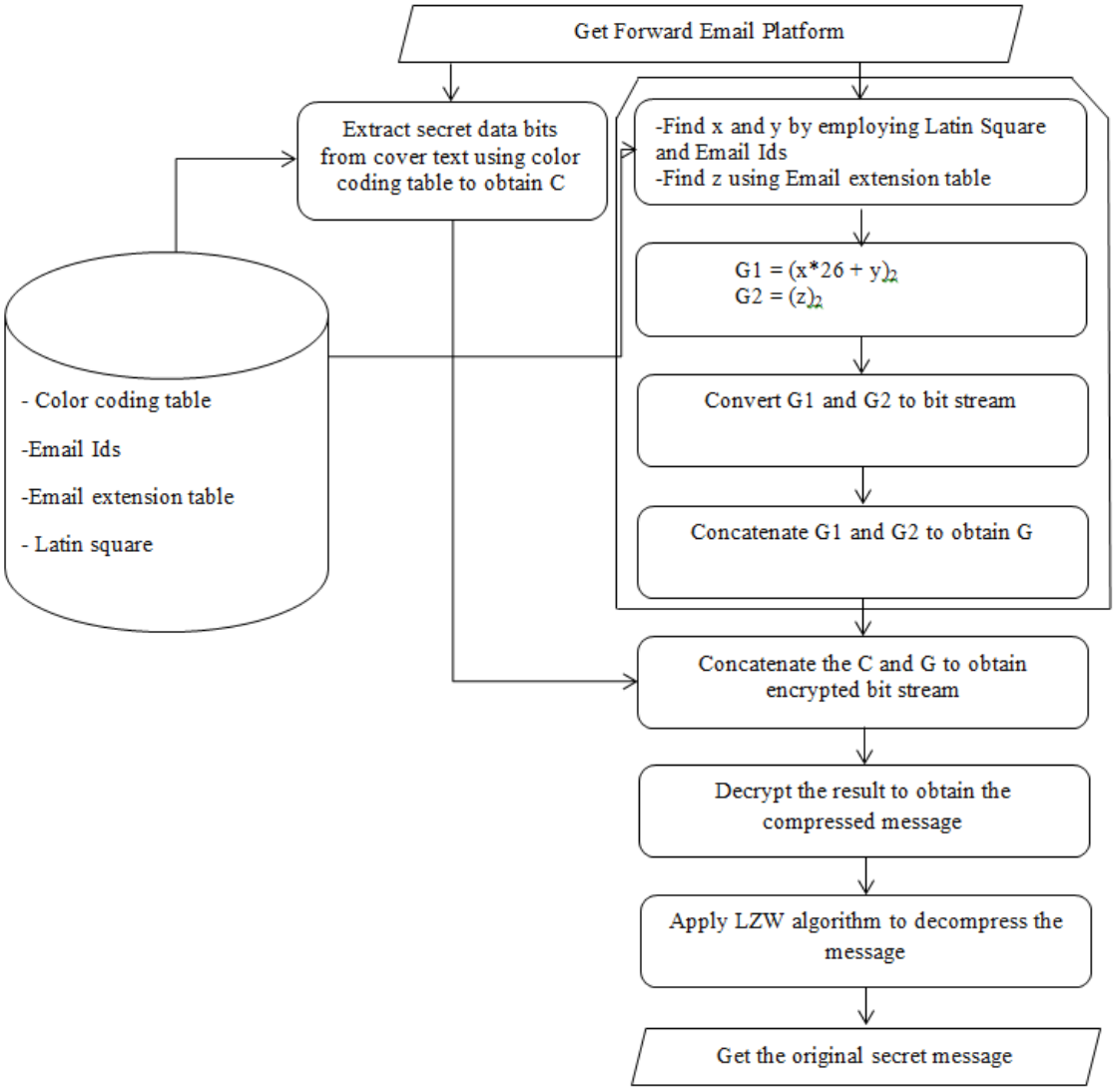

Figure 13. Extracting Phase

We implemented the program using Java language. This is because:

- Java is robust and secure.
- It is simple, object-oriented and familiar.
- It is threaded and dynamic.
- The cryptography implementation is easier by Java.

Also, Java has some features that make it more secure:

1. Protection from security attacks. Tt allows developers to declare classes or methods as FINAL, as any class or method declared as final cannot be overridden, which helps developers to protect code from security attacks, e.g. by creating a subclass and replacing it with the original class and override methods.
2. Bytecode. Every time a user compiles the Java program, the Java compiler creates a class file with Bytecode, which is tested by JVM at the time of program execution for viruses and other malicious files.
3. Access control functionality. Java has access-control functionality on variables, and the methods within the objects provide a secure program by preventing access to the critical objects from the untrusted code.

4. Comparison and Remarks

The proposed multi-level security improvement is compared to the original crypto-stego design. The study focused on security and capacity which can be considered in our scope of motivate concerns in this experimental security systems [12].

Table 1 presents a descriptive comparison between the existing method [9] and the proposed work based on different criteria. In text steganography, the hiding capacity is a major decisive parameter for performance analysis of the algorithm. The hiding capacity is calculated by dividing the number of bits of the secret message with the total number of bits used to construct the entire stego-cover [13]. It is clearly evident from the experimental results that the proposed algorithm achieves better hiding capacity.

The scheme uses color coding to embed the secret data in the body of the email message and additionally in the email IDs. Therefore, it is not possible to distinguish that there is hidden text, i.e. the hidden text is in normal email environment that does not contain noticeable change.

Table 1. Comparison between the original studied work vs. this proposed improvement method

Comparison Parameters	Studied Work	The Improvement
Capacity	Good hiding capacity	
Security	Less Secure	High Secure
Security based on	compression and steganography method	compress, cryptography, and steganography method
Attach required to find out original message	• compression algorithm • steganography method	• compression algorithm • private key of sender or receiver • shared key • steganography method

In encryption, knowing the secret key is the limit in the detection of the confidential message. Security in the proposed method is very high, given the use of encryption. The proposed encryption method does not send the secret key through the channels, it only sends the secret message after the encryption so it is difficult for the attackers to calculate the encryption secret key. The proposed method also includes compression technology as well as the way the information is hidden. These techniques enhance the security of the algorithm.

Figure 14 shows the performances of the existing method before improvement [9] implemented on the same platform to insure fair comparison reference. Figure 15 represents the performances of our proposed method (after improvement) in the same situation. Observe the calculations increase and performance decreases in the proposed method compared to original model by a small percentage. This observation is analyzed and found justified due to the use of encryption, which needs more calculations; different than the previous presented crypto enhancement due to sub-threshold SRAM design [22] as well as the service Hajj smart systems analysis and its security [23] or the Holy Quran (Muslims Religious Holy Book) security authentication [24]. It is clear in this work that the code before optimization used 32% of the CPU time in less than five seconds, three of them for Kernel, and it used for that only 11 threads, while the code after optimization was used more because of the increased computations in the security layer. The code after optimization used 39% of the CPU time, four of them for Kernel, it also did it in less than five seconds and used 10 threads.

5. Conclusions

Steganography is one of the most important topics of the present age. It is used by many people and institutions as well as countries during the exchange of confidential information. There are many studies that have researched how to hide data and send it in a safe way that can only be accessed by the people concerned. In this study a method was designed to hide data in a secure way within the texts in the forward email platform.

The data is compressed and then encrypted with the AES algorithm using the corresponding shared key at the end of the DHKE algorithm. This encrypted data is distributed in the text

of the forwarded e-mail content as well as in the email IDs and then sent via e-mail. At the receiving end, the data is collected and decrypted to obtain confidential data. Even if a hacker could find out how to hide the data, he would get encrypted data that he could not read correctly or benefit from. Despite efforts to improve the security of hidden data, more research is needed to ensure that data is safely hidden during transmission.

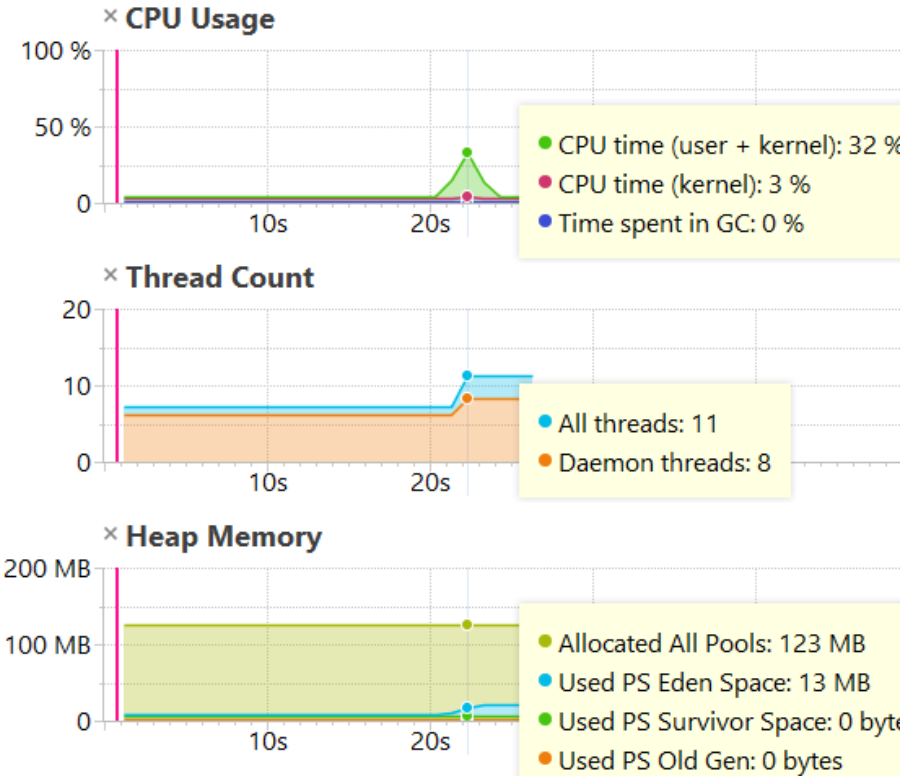

Figure 14. Performance of original model (before improvement)

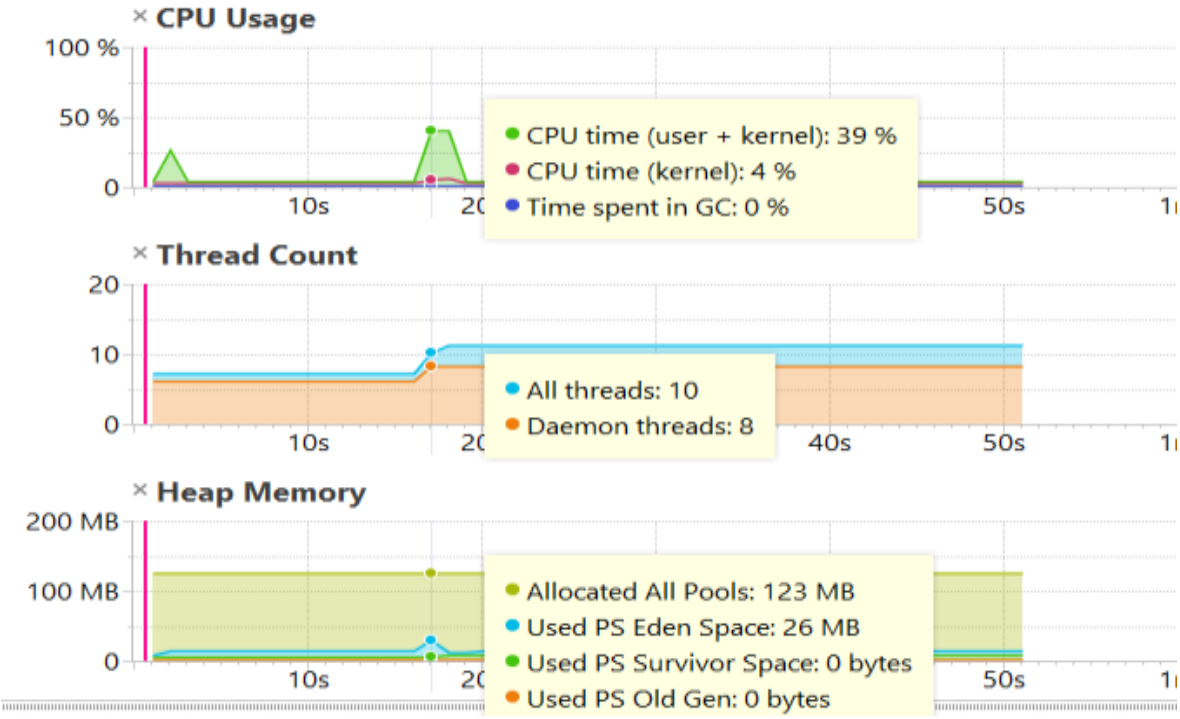

Figure 15. Performance of the proposed design (after improvement)

Acknowledgement

We would like to express our deepest appreciation to all those who provided us with the possibility to complete this research at Umm Al-Qura University (UQU). Special gratitude goes to Prof. Adnan Gutub for his offering us the graduate course of *Advanced Topics in Information Security*, who contributed suggestions, encouraged outstanding innovations strategies and helped us learn research and prepare the paper publications. Also, we would like to offer our special thanks to College of Computers and Information System and UQU for supporting possibility conducting this work.

References

[1] D. Artz, "Digital steganography: hiding data within data", *IEEE Internet Computing*, vol. 5, no. 3, pp. 75–80, May/June 2001.

[2] S. Kallam, "Diffie-Hellman key exchange and public key cryptosystems", *Master degree of Science, Math and Computer Science, Department of India State University*, 2015.

[3] A. Gutub, F. Khan, "Hybrid Crypto Hardware Utilizing Symmetric-Key & Public-Key Cryptosystems", *International Conference on Advanced Computer Science Applications and Technologies – ACSAT2012*, Palace of the Golden Horses, Kuala Lumpur, Malaysia, 2012.

[4] R. Kumar, A. J. Singh, "Understanding steganography over cryptography and various steganography techniques", *International Journal of Computer Science and Mobile Computing*, vol. 4 no. 3, pp. 253–258, 2015.

[5] A. Gutub, M. Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions", *WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE)*, Vienna, Austria, pp. 28-31, 2007.

[6] F. Khan, A. Gutub, "Message Concealment Techniques using Image based Steganography", *The 4th IEEE GCC Conference and Exhibition*, Gulf International Convention Centre, Manamah, Bahrain, 2007.

[7] N. AlAssaf, B. AlKazemi, A. Gutub, "Applicable Light-Weight Cryptography to Secure Medical Data in IoT Systems", *Journal of Research in Engineering and Applied Sciences (JREAS)*, vol. 2, no. 2, pp. 50–58, 2017.

[8] N. Al-Juaid, A. Gutub, E. Khan, "Enhancing PC Data Security via Combining RSA Cryptography and Video Based Steganography", *Journal of Information Security and Cybercrimes Research (JISCR)*, vol. 1, no. 1, Published by Naif Arab University for Security Sciences (NAUSS), 2018.

[9] A. Malik, G. Sikka, H. Verma, "A high-capacity text steganography scheme based on LZW compression and color coding", *Engineering Science and Technology - an International Journal*, vol. 20, no. 1, pp. 72–29, 2017.

[10] C. Paar, J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer-Verlag Berlin Heidelber, 2010.

[11] A. Gutub, N. Al-Juaid, E. Khan, "Counting-Based Secret Sharing Technique for Multimedia Applications", *Multimedia Tools and Applications: An International Journal* – Springer, ISSN 1380-7501, 2017.

[12] A. Gutub, N. Al-Juaid, "Multi-Bits Stego-System For Hiding Text in Multimedia Images Based on User Security Priority", *Journal of Computer Hardware Engineering*, vol. 1, no. 2, p. 9, 2018.

[13] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Technique For Data Hiding", *IBM Systems Journal*, vol. 35, no. 3–4, pp. 316–336, 1996.

[14] K. F. Rafat, "Enhanced text steganography in SMS", *IEEE 2nd International Conference on Computer, Control and Communication*, Karachi, Pakistan, 2009.

[15] M. Shirali-Shahreza, S. Shirali-Shahreza, "Steganography in TeX documents", *IEEE 3rd International Conference on Intelligent System and Knowledge Engineering*, Xiamen, China, 17–19 Nov. 2008.

[16] M. Shirali-Shahreza, M. H. Shirali-Shahreza, "Text Steganography in SMS", *IEEE International Conference on Convergence Information Technology (ICCIT)*, Gyeongju, South Korea, 2007.

[17] A. Gutub, A-R. El-Shafe, M. Aabed, "Implementation of a pipelined modular multiplier architecture for GF(p) elliptic curve cryptography computation", *Kuwait Journal of Science and Engineering (KJSE)*, vol. 38, no. 2B, pp. 125–153, 2011.

[18] A. Gutub, H. Tahhan, "Efficient Adders To Speedup Modular Multiplication For Cryptography", *WoSPA 5th IEEE International Workshop on Signal Processing and its Applications*, University of Sharjah, U.A.E. 2008.

[19] A. Gutub, "Fast 160-Bits GF(p) Elliptic Curve Crypto Hardware of High-Radix Scalable Multipliers", *International Arab Journal of Information Technology (IAJIT)*, vol. 3, no. 4, pp. 342–349, 2006.

[20] A. Gutub, A-A. Tabakh, A. Al-Qahtani, A. Amin, "Serial vs. Parallel Elliptic Curve Crypto Processor Designs", *IADIS International Conference: Applied Computing*, Fort Worth, Texas, pp. 67–74, 2013.

[21] N. Al-Otaibi, A. Gutub, "2-Leyer Security System for Hiding Sensitive Text Data on Personal Computers", *Lecture Notes on Information Theory, Engineering and Technology Publishing*, vol. 2, no. 2, pp. 151-157, 2014.

[22] A. Gutub, "Subthreshold SRAM Designs for Cryptography Security Computations", *ICSECS - 2nd International Conference on Software Engineering and Computer Systems*, Universiti Malaysia Pahang, Kuantan, Malaysia, 2011.

[23] S. Aly, A. Gutub, "Intelligent Recognition System for Identifying Items and Pilgrims", *NED University Journal of Research - Thematic Issue on Advances in Image and Video Processing, ISSN: 2304-716X*, pp. 17–23, 2018.

[24] M. Almazrooie, A. Samsudin, A. Gutub, M. Salleh, M. Omar, S. Hassan, "Integrity verification for digital Holy Quran verses using cryptographic hash function and compression", *Journal of King Saud University - Computer and Information Sciences, Elsevier*, 2018.