A simple flexible cryptosystem for meshed 3D objects and images

Article *in* Journal of King Saud University - Computer and Information Sciences · March 2019

DOI: 10.1016/j.jksuci.2019.03.008

CITATIONS
4

READS
154

4 authors:

Manal Abd Al-Jabbar Ahmad Mizher
Universiti Kebangsaan Malaysia
8 PUBLICATIONS **16** CITATIONS
SEE PROFILE

Riza Sulaiman
Universiti Kebangsaan Malaysia
164 PUBLICATIONS **650** CITATIONS
SEE PROFILE

Ayman M. Abdalla
Al-Zaytoonah University of Jordan
53 PUBLICATIONS **317** CITATIONS
SEE PROFILE

Manar Mizher
Amman Arab University
14 PUBLICATIONS **31** CITATIONS
SEE PROFILE

Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A simple flexible cryptosystem for meshed 3D objects and images

Manal Abd Al-Jabbar Ahmad Mizher [a,*], Riza Sulaiman [a], Ayman Mahmoud Aref Abdalla [b], Manar Abduljabbar Ahmad Mizher [a]

[a] Institute of Visual Informatics, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
[b] Faculty of Science & I.T, Al-Zaytoonah University of Jordan, P.O. Box 130, Amman 11733, Jordan

ARTICLE INFO

Article history:
Received 28 November 2018
Revised 18 February 2019
Accepted 12 March 2019
Available online xxxx

Keywords:
Cellular automata
Grayscale
RGB
Meshed 3D object

ABSTRACT

In cryptography, the previous research generally appears ambiguous and complex to the novice researcher due to the use of complex mathematical equations and rules of cryptography. Moreover, many research approaches lack flexibility in their key length or in their level of encryption. Consequently, combining simplicity, flexibility, and reliability is not easily obtainable in a cryptosystem, especially for larger and more complex data items. Therefore, a new system, called Flexible cryptosystem based on Cellular Automata (FcCA), is proposed here as a novel simplified flexible cryptosystem based on cellular automata (CA). FcCA presents simplified techniques for making CA reversible while creating a robust flexible cryptosystem that performs lossless encryption of three-dimensional (3D) objects and images of different types. It uses pure random CA as a diffusion technique, and it employs a modified existing confusion technique by substituting the static start point with proposed multi-dynamic intersected start points. In addition, FcCA novelty includes using a combination of aspects: random configuration with open boundary conditions, g-th order memory independent-cell technique, and classification of two parts of the encryption key into subkeys. The length and complexity of FcCA subkeys can be controlled easily because the subkeys depend on flexible parameters. Testing and validation of FcCA scrambling level were performed with several criteria including correlation, entropy, peak signal to noise ratio, and value difference degree. Experimental results showed that FcCA has high flexibility, a high level of scrambling, and higher robustness of keys compared to other methods of encryption. In addition, sensitivity analysis showed FcCA to be highly sensitive to changes in the encryption key and encrypted images and objects. Overall, the properties of FcCA demonstrated its effectiveness as a cryptosystem for images and 3D objects.

1. Introduction

Encryption is a powerful tool of cryptography that performs the operation of encoding data, so that only authorized sides can understand and read it. Manoj (2010) presented an overview of cryptography, its importance, and some of its major applications. Encryption does not preclude unauthorized access, but it decreases the probability of comprehending encrypted data obtained from that access. Therefore, plain data are often encrypted using an encryption algorithm to transform them into scrawled cipher data. This is commonly done with an encryption key that defines how the data are to be confused (Diffie and Hellman, 1976a,b). The schemas of encryption must be invertible for the original data to be recovered from the encrypted data, where increased complexity of data structures and objects increase the difficulty of this task. For example, encrypting images generally requires more specialized methods than encrypting text. Therefore, lossless encryption methods, such as (Yahya and Abdalla, 2009; Abdalla and Tamimi, 2013; Tamimi and Abdalla, 2015, 2017), were developed to encrypt images of different types. Other methods, such as (Tamimi and Abdalla, 2014) were developed for audio encryption. However, such methods were not suited for encrypting three-dimensional (3D) objects. Furthermore, the shuffling methods used by these algorithms had limited flexibility in terms of changing the level of encryption.

* Corresponding author.
E-mail addresses: manal@siswa.ukm.edu.my (Manal Abd Al-Jabbar Ahmad Mizher), a.riza@ukm.edu.my (R. Sulaiman), ayman@zuj.edu.jo (A.M.A. Abdalla), manar@siswa.ukm.edu.my (M.A.A. Mizher).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2019.03.008
1319-1578/

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

Consequently, reversible cellular automata (CA) methods were used in cryptography algorithms (Nandi et al., 1994), especially for visual data such as images and 3D objects (Abu Dalhoum et al., 2012). CA was presented by Stani-slaw Ulam and John von Neumann in the 1940s, as a self-organizing discrete dynamic system, with a finite set of cells where every cell is updated by discipline rules dependent on a certain number of neighbouring cells (Alvy, 1969; Gutowitz, 1991). CA can use the same representation components used by the majority of visual data types, such as pixels, voxels, and axes. In addition, CA may be used as a single encryption key or as a subkey for further encryption strength and security (Hanis and Amutha, 2018). However, CA often requires the grid to be finite in all dimensions, so some boundary conditions have to be defined. Traditionally, two types of boundary condition were defined: periodic (or cyclic) and static (or closed).

Visual information and its representations, such as 3D objects, are currently trending in the world. 3D objects have various applications in 3D graphics, video games, engineering, medicine, military industry, and many other areas. Most users of these applications require privacy and security. Therefore, numerous researches focused on encrypting 3D objects.

3D objects may be classified into two main types, depending on how they are read and stored. The first type is 3D objects defined by voxels where the 3D object of dimensions ($n \times n \times n$) can be represented with a 3D matrix. Each cell is 1 if the analogous position in the solid object is occupied and 0 if the position is empty. This is illustrated by the example in Fig. 1(b) for the object in Fig. 1(a). The second type of 3D objects, illustrated in Fig. 1(c) through (e), is defined by the general features of 3D object file formats (McHenry and Bajcsy, 2008), such as 3D object geometry, appearance, scene, and animations. The most popular formats used by this type include STL, OFF, OBJ, FBX, and COLLADA. Files with the OFF format can be found in the Princeton Shape Benchmark (PSB) (Shilane et al., 2004), provided by Princeton University since 2003. It has 1814 polygonal objects collected from the internet and classified manually depending on their functionality and form. In addition, the geometry of objects is obtained by assigning the polygons of each object's surface. The polygons may have a massive number of faces and vertices.

For 3D objects, this paper will focus on the second type, described above, because it is currently more applied than the first one. Therefore, the proposed encryption method will use only surface geometry features of the object. Every 3D object has a unique surface geometry. The surface is a basic feature of the 3D object, where there is a variety of existing methods for encoding surface geometry. This paper will pay special attention to object encoding by the Object File Format (OFF) approximate mesh method. Further information and explanations of this method, among others, can be found in (McHenry and Bajcsy, 2008). OFF Objects can be easily represented on computers with two 2D matrices: Face and Vertex. Thus, a 3D object can be encrypted as two 2D matrices instead of one 3D matrix. This should save system resources such as CPU and memory.

The rest of this paper is organized as follows. Previous work is discussed in Section 2. In Section 3, the definition and characteristics of the proposed FcCA system will be introduced. The cryptography method of FcCA will be explained in Section 4. In Section 5, experimental results will be shown where the cryptosystem security analysis will be presented in Section 6. Finally, the conclusion and future work will be presented in Section 7.

2. Related work

There is some scarcity of cryptosystems for 3D objects. Some efforts were made to encrypt 3D objects, such as those defined by holograms. For example, (Alfalou and Brosseau, 2013; Li et al., 2013, 2014; Wen and Xudong, 2013; Yang and Zhang, 2016) used different techniques, including CA. Another effort encrypted 3D objects using the Blakely scheme with Thien and Lin schemes where 3D models were compressed before the encryption (Elsheh and Ben Hamza, 2011). Furthermore, there are some limited encryption methods for 3D objects defined by voxels, such as (Martín del Rey, 2015, 2017; Martín del Rey et al., 2016), who used 3D CA and split the object before encryption. Some models were defined by point clouds (Jolfaei et al., 2015; Jin et al., 2016), and others were defined by meshes without texture (Éluard et al., 2013) where they shuffled vertices and their coordinates by applying key-seeded pseudo-random permutations. Alternatively, (Jin et al., 2017) focussed on texture, where a 3D Lu-chaotic mapping was used for mesh encryption. These proposals are described briefly and compared in Table 1.

Overall, most traditional cryptosystems use fixed sizes for their key spaces and fixed scrambling levels. Moreover, most previous researches depended on traditional complex encryption rules or maps, or on use of 3D cellular automata.

A method for scrambling digital images was proposed by (Abu Dalhoum et al., 2012) based on a special kind of two-dimensional (2D) CA called the Game of Life (GOL). They used GOL rules to design the diffusion component where at each generation, GOL produces cells of 1's and 0's depending on the neighbours' values. The 0's cells are ignored while the 1's cells are used to configure the new pixel coordinates for the scrambled image. This is the confusion component. At each generation in this component of GOL, each pixel in the plain (original, unencrypted) matrix is paired with the pixel in CA that has the same location (index). Then, depending on the indices of 1's in the GOL configuration, the paired pixels in the plain matrix will change their locations. See the example in Fig. 2.

Chai et al. (2018) devised a memristive chaotic system that employs elementary cellular automata and compressive sensing (CS). At first, the wavelet coefficients of the plain image are scrambled by a zigzag path and elementary CA to achieve a higher scrambling degree where a hashing method is utilized to get some parameters used in the encryption process. Next, CS is applied to compress and encrypt the scrambled image, producing the final cipher image. For CS, a circular measurement matrix produced by

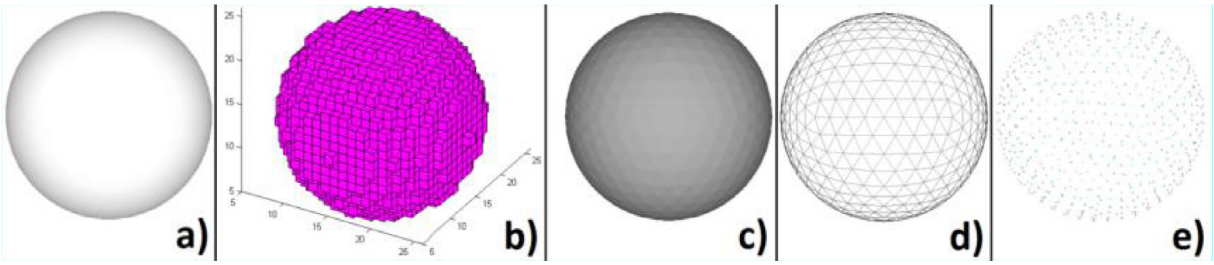


Fig. 1. (a) Solid 3D object (b) Object defined by voxels (c, d, e) Object defined by a mesh.

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

Table 1
A brief comparison between some 3D models of encryption.

Reference	Encrypted Data	Technique	Preprocess	Flexible key	Flexible Scrambling level
Elsheh and Ben Hamza, 2011	3D object (mesh)	Blakely combined with Thien and Lin	Compress object	Yes (Low flexibility)	No
Éluard et al., 2013	3D object (mesh)	Key-seeded pseudo random permutation	None	No	No
Martín del Rey, 2015	3Dobject (voxels)	3D cellular automata	Split object	No	Yes
Jolfaei et al., 2015	3D object (point cloud)	Series of random permutations and rotations	Explain the content-dependent metadata in a special way	No	Yes
Martín del Rey et al., 2016	3D object (voxels)	3D cellular automata	None	No	Yes
Jin et al., 2016	3D object (point cloud)	3 random sequences are generated by the logistic chaotic mapping	None	No	No
Martín del Rey, 2017	3D object (voxels)	3D cellular automata	None	No	Yes
Jin et al., 2017	3D object (mesh) and Images	3D Lu chaotic mapping	None	No	No

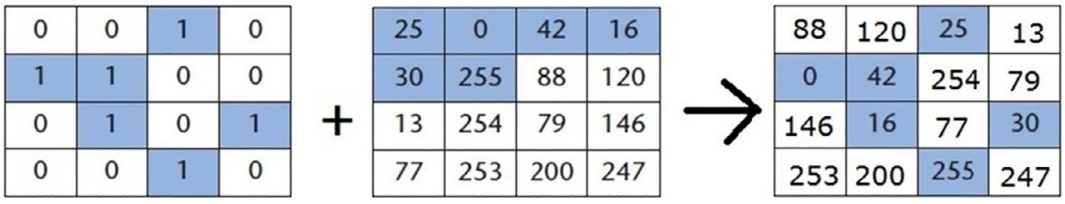


Fig. 2. A confusion technique showing CA, plain data, and scrambled data matrices (Abu Dalhoum et al., 2012).

a magnetic-controlled memristive chaotic system is adopted. This decreases the energy consumption of data transmission, generates a large key space, and makes the encryption algorithm more resistant to a brute force attack.

The encryption system by (Hosseini and Kamel, 2018) was based on intertwining logistic map and cellular automata. In the first step, an intertwining logistic map is used as the primary key, and in the second step, the attractors of cellular automata are used as the second key. In the third and final step, the encryption key is generated by combining the first and second keys.

In the system of (Noshadian et al., 2018), the image is confused using logistic map as a chaos function and diffused by a modified Knuth shuffling algorithm. Then, Teaching-Learning-Based Optimization and Gravitational Search evolutionary algorithms are harnessed to speed up the optimization process of these parameters.

The encryption system of (Ping et al., 2018) consisted of two processes: permutation and substitution. In the permutation process, a two-dimensional Logistic-adjusted-Sine map (2D-LASM) with excellent properties is adopted to shuffle the pixel positions. In the substitution process, a second-order life-like CA with balanced rules is employed. The balanced rules gradually make the distribution of 0's and 1's in the life-like CA remain in equilibrium during iteration. Second-order CA can preserve the result of CA after each iteration to maintain the reversibility. Furthermore, to resist various plaintext attacks, the algorithm controls the initial conditions of 2D-LASM with the key and the weighted histogram of the plain-image.

Liu et al. (2018) used selective permutation as a self-adaptive method to replace the traditional confusion method, so this method can be used for permutation rule generation. Wu et al. (2018) generated a 2D chaotic map by connecting Henon and Sine maps, while (Ye et al., 2018) presented a pixel-level image encryption algorithm using chaotic maps.

3. Features of the FcCA cryptosystem

The proposed cryptosystem, called Flexible cryptosystem based on Cellular Automata (FcCA), performs encryption of images and 3D objects with flexible techniques that employ CA. The flexibility of FcCA includes allowing change to the sizes of encryption subkeys. This provides easier control of the robustness of encryption and decryption processes and allows fitting huge objects and variant images of different dimensions and types. This flexibility has some advantages such as easier programming and lower memory requirements. The following subsections explain FcCA is in detail.

3.1. Configuration

The flexible size of CA in FcCA does not depend on the size of the plain data matrices. This is due to adding an expansion property to CA, called expanded CA (expCA). If the size of the plain data matrix A is $W \times H$, then the size of expCA is $(W + N_1) \times (H + N_2)$, where N_1, $N_2 \in [0, 1000]$ are natural numbers. The upper limit of 1000 for N_1 and N_2 was chosen since it can sufficiently strengthen the cryptosystem within reasonable execution time. More details will be presented in Subsection 6.2.

The initial configuration for CA for the diffusion component in FcCA is constituted randomly. Then, creating more generations is done by filling the CA with 0's and 1's randomly, without any neighbourhood rules. The confusion component is a modification of the approach by Abu Dalhoum et al. (2012). They used one static start point, while FcCA uses multiple dynamic intersected start points (iDL). Therefore, instead of pairing each pixel in the plain data matrix with a pixel in CA that has the same location, FcCA pairs matrix locations with those locations that are determined after shifting the plain data matrix, depending on predefined values (iDL). The expCA matrix has more rows and columns than the matrix of plain data. The latter matrix can flexibly lay on any position in the cells area in the expCA matrix.

Consider the example in Fig. 3. In this example, let A be a plain 4×4 matrix, and E be a 6×6 expCA matrix, as seen in the figure. Then, the starting point of the plain matrix, in this example, will lie on the point (3, 2) in matrix E, and iDL will have the coordinates of this starting point which equal (3, 2). Then, the matrix A will be encrypted depending on the 1's and 0's of the E matrix, which are located by starting at point (3, 2), not (0, 0). Recall that laying

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

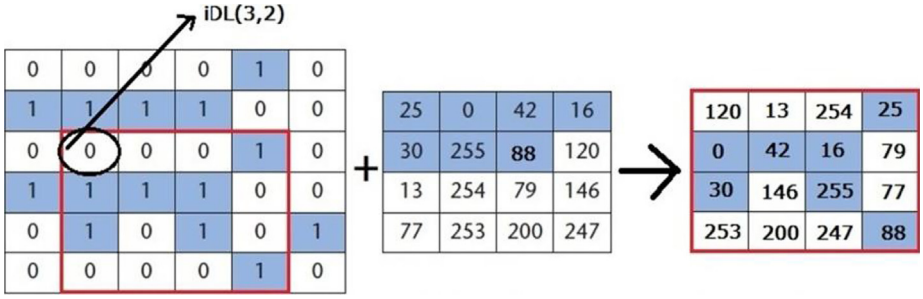


Fig. 3. Example of FcCA showing expCA, plain data, and scrambled data matrices.

position and the iDL point can be changed easily within expCA boundaries.

Moreover, if plain data have more than one matrix, each matrix will be encrypted using the same expCA, but with a different iDL point for each matrix without exceeding expCA boundaries. As shown in the example in Fig. 4, Plain Matrix 1 has iDL = (6, 5), while Plain Matrix 2 has iDL = (3, 7). Consequently, the main goal of the confusion component of FcCA is to permute positions of points in the plain image or in the plain 3D object.

3.2. Boundaries

In the FcCA system, the open-boundary condition is used. For example, if the size of a plain data matrix A is W × H, then, in the simplest case, the size of CA will be W × H. All cells in CA have their values included in the initial randomisation and at all generations. Consequently, FcCA does not need to define any boundary conditions for the cells' values at the subsequent generations since they are completely independent of the cells' values of previous generations.

3.3. Reversibility

Two-dimensional CA is reversible if its inverse progression is possible to compute. However, reversibility is undecidable by default for CA of two or more dimensions. Therefore, a g-th order memory independent-cell technique is proposed to make it decidable. In this technique, the states of each cell in a CA generation are binary and they are saved sequentially in an analogous position in Hybrid Cellular Automata (HCA), where an HCA is another 2D CA with the same size of the expCA in the cryptosystem. HCA will provide the first subkey in the FcCA cryptosystem. Fig. 5 shows an example of composing values of HCA from expCA generations.

This proposed reversibility technique has three main advantages. Firstly, it significantly reduces the number of stored states. For example, when the CA grid size is 8 × 8, only 2^6 states are needed instead of 2^{64}. Secondly, it eliminates ancestor configurations with no predecessors in traditional cellular automata. Thirdly, it eliminates the need to remember the states of neighbour cells from the preceding steps.

3.4. Flexibility

The proposed FcCA cryptosystem has high flexibility, especially for encryption of meshed 3D objects. These objects have high contrast between the number of rows and the number of columns in their Face and Vertex matrices. The flexibility of FcCA results mainly from two proposed techniques: the expansion of HCA and the insertion into HCA.

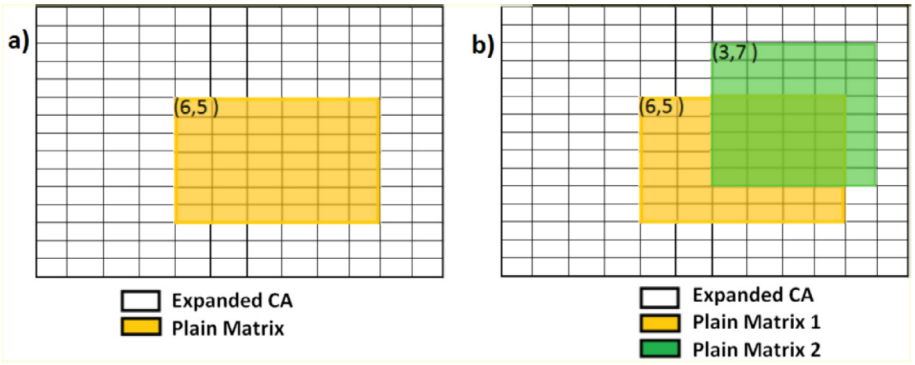


Fig. 4. iDL for plain data matrix a) with one plain matrix b) with two plain matrices.

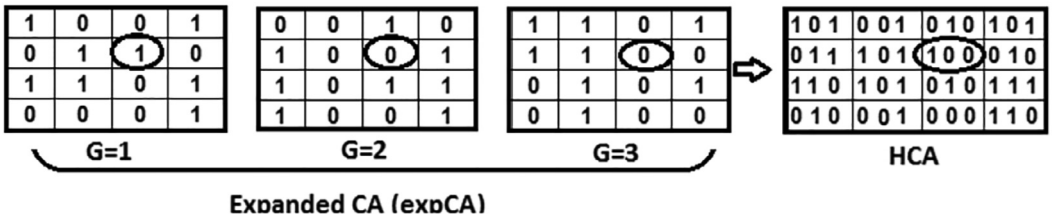


Fig. 5. Composing values in HCA from expCA generations.

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

Expansion of the HCA matrix is performed by simultaneously expanding the expCA matrices. This causes the first subkey to become larger and harder to be broken. While HCA is separately recording states for each cell in expCA, as sequential binary digits, it also can insert, in these binary series, bogus random binary bits in specific positions called (Ps). Bogus bits can be added at the beginning, end, or any given indices.

For example, let the original HCA state at index (3, 5) be '100010101', and the HCA state after adding Ps (bold bits) at the same index be '**01**100**0**0**11**0101**01**0**0101**', so Ps = {1, 2, 5, 8, 9, 10, 14, 15, 18, 19, 20, 21}. These positions can be changed easily for another object or image. The essential advantage of these bogus bits is to mislead attackers and to make finding the original CA configurations more difficult. For another example, see Fig. 6 where Ps are underlined.

The FcCA key consists of two parts. The first part is the HCA, and the second part is a collection of Ps and the coordinates of iDL points. Recall that HCA is the most important part of the key because it has information that may lead attackers to original data. On the other hand, the second part of the key, which is iDL and Ps, is insignificant by itself because it does not contain any information about the original data. Therefore, the main aim of FcCA is to secure and protect HCA. Consequently, the second part of the key has been proposed. FcCA disregards protecting the second part to reduce the complexity of the system, but it is recommended to store and transmit each part of the key individually to increase overall security.

4. The FcCA cryptography method

The FcCA system consists of two components: cipher and decipher. The cipher component has three phases. These phases, illustrated in Fig. 7, are the pre-processing (input/extract matrices) phase, the pre-encryption phase, and the encryption/output phase.

As illustrated in Fig. 7, FcCA takes a plain image or a plain 3D object as input and then transforms it into its corresponding matrix or matrices. After that, it creates the expCA and HCA matrices and assigns values of iDL points, Ps and number of generations. Finally, expCA will be filled randomly by ones and zeros, and HCA will be composed, and then the data enter the diffusion and confusion phases (for a required number generations) with a secret key to produce the scrambled output. The steps of the cipher component, illustrated in the flowchart in Fig. 8, consists of the three phases described in the following subsections.

4.1. The pre-processing phase

This phase takes plain data, in the form of an image or a 3D object, as input and stores them in a suitable format. The data are extracted and stored in a 2D matrix or matrices depending on the nature of these plain data. For example, a greyscale image will be stored as a 2D matrix with the number of rows and columns equal to the number of pixels in the width and height of the image, respectively. Similarly, a colour image in Red-Green-Blue (RGB) format will be stored in three 2D matrices, where each matrix represents one colour component of the RGB image with the same dimensions as of that component. A 3D object will be stored as two 2D data matrices, one for its faces and the other for its vertices. The dimensions of these two matrices usually differ from each other, depending on the numbers of faces and vertices in the 3D object.

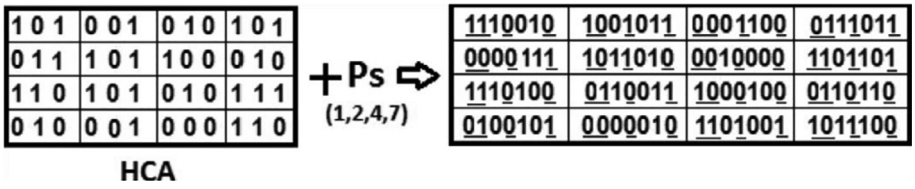


Fig. 6. Example of inserting bogus bits (underlined) into HCA.

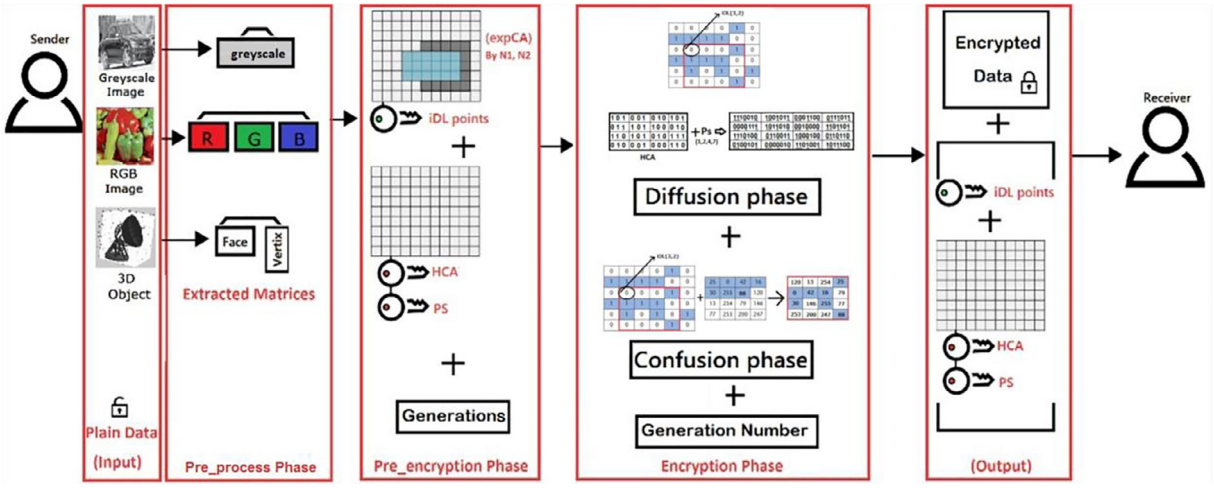


Fig. 7. A sketch of FcCA phases with different types of input.

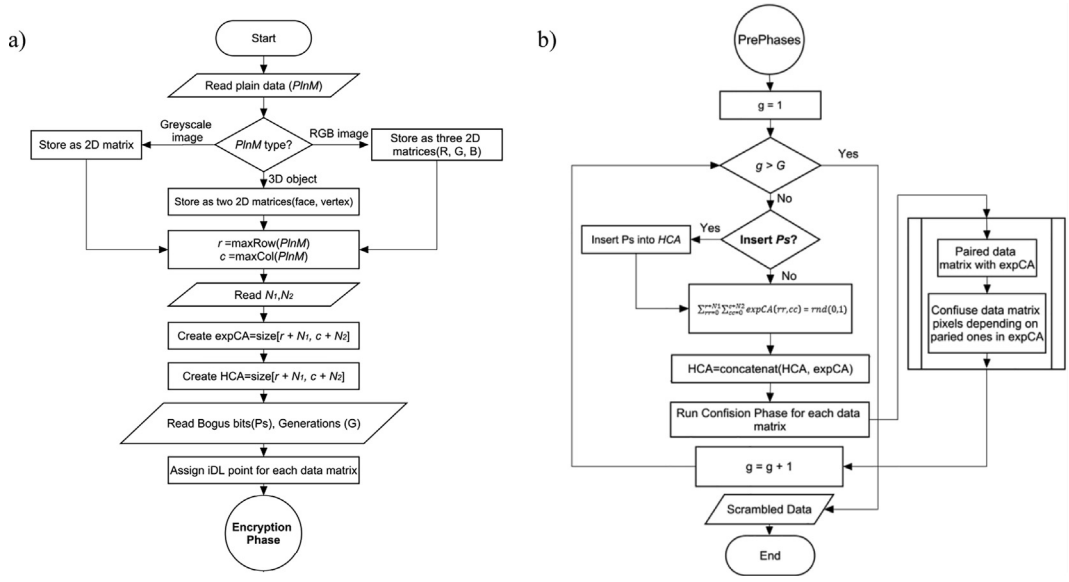


Fig. 8. A flowchart of the cipher component (a) Pre-processing and pre-encryption phases (b) Encryption phase.

4.2. The pre-encryption phase

This phase determines the parameters: inserted bogus numbers (Ps), parameters of expansion to matrix dimensions (N_1, N_2), and the number of CA generations (G). Then, it creates the empty expanded cellular automata (expCA) and the empty hybrid cellular automata (HCA) matrices. These two matrices have the same dimensions, which are determined based on the sizes of plain data matrices as follows.

Let Mtrx be a 2D matrix, and plnM be the set of extracted matrices. Then, ∀ Mtrx: Mtrx ∈ plnM:

$$maxR = max(rowsCount(Mtrx)) \quad (1)$$

$$maxC = \max(columnsCount(Mtrx)) \quad (2)$$

$$expCA = size(maxR + N_1, maxC + N_2) \quad (3)$$

$$size(HCA) = size(expCA) \quad (4)$$

For example, suppose the size of the faces matrix of a 3D object is 100×3, while the size of its vertices matrix is 112×3. Then, *maxR* = 112, and *maxC* = 3.

Each plain data matrix has an iDL(x, y) point with the condition of not exceeding the boundaries of expCA and HCA, where ∀ *Mtrx: Mtrx ∈ plnM:*

$$rowsCount(Mtrx) + iDL(x) \leq rowsCount(expCA) \quad (5)$$

$$columnsCount(Mtrx) + iDL(y) \leq columnsCount(expCA) \quad (6)$$

A flowchart of the pre-processing and pre-encryption phases is shown in Fig. 8(a).

4.3. The encryption phase

The encryption phase is where diffusion and confusion take place, as outlined by the following algorithm. Note that the number of matrices required to represent the input is one for a greyscale image, three for an RGB image, and two for a 3D object.

4.3.1. The diffusion sub-phase

In every generation, the configuration of expCA is constituted randomly without any neighbourhood or boundary rules. In addition, bogus bits (Ps) are inserted into HCA when needed, and the expCA cells' values are copied and concatenated into HCA cells' values. The pseudocode of the Diffusion Algorithm is given below.

Diffusion Algorithm:

```
for row = 1:1: Rows_OF_expCA
  for col = 1:1: Columns_OF_expCA
    if current_generation is an element in Ps
      BogusBit = String(mod(int64(rand.*10),2));
      HCA(row,col) = String Concatenation(HCA(row,col),
  BogusBit);
    End
    TempValue = mod(int64(rand.*10),2);
    if TempValue == 1
      expCA(row,col) = 1;
      HCA(row,col) = String Concatenation(HCA(row,col),'1');
    else
      expCA(row,col) = 0;
      HCA(row,col) = String Concatenation(HCA(row,col),'0');
    end
  end
end
```

4.3.2. The confusion sub-phase

After filling expCA with random 1's and 0's, and concatenating expCA values with bogus bits (if needed) to HCA, FcCA uses multiple dynamic intersected start points (iDL) to pair each plain data matrix with its corresponding cells (subMatrix) in expCA. Then, depending on the indices of 1's in the expCA configuration, the paired pixels in the plain matrix will change their locations as it was shown in Fig. 2. After that, the encryption phase is repeated until all generations are completed. Finally, the scrambled data are produced as output. The pseudocode of the Confusion Algorithm is given below.

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

```
Confusion Algorithm
r=1; c=1;
for row = 1:1: Rows_OF_plainData
  for col = 1:1: Columns_OF_plainData
    paired_X_coordinate= row+iDL(x);
    paired_Y_coordinate= col+iDL(y);
    if expCA(paired_X_coordinate, paired_Y_coordinate)==1
      if c < Columns_OF_plainData
        Picked_element=plainData(r,c);
        c++;
      else
        r++;
        c=1;
        Picked_element=plainData(r,c);
      end
      encryptedData(row,col)= Picked_element;
    end
  end
end
for row = 1:1: Rows_OF_encryptedData
  for col = 1:1: Columns_OF_encryptedData
    if isempty(encryptedData(row, col))
      if c < Columns_OF_plainData
        encryptedData(row, col)= plainData(r,c);
        c++;
      else
        r++;
        c=1;
        encryptedData(row, col)= plainData(r,c);
      end
    end
  end
end
```

A flowchart of the encryption phase with its two sub-phases is shown in Fig. 8(b).

4.4. The decipher component

The decipher component takes as input the scrambled data with the related secret key (HCA, iDL points, and Ps). Then, the decryption steps of this component perform the inverse of the cipher component. Finally, it yields the recovered original image or original object as output without any loss of information.

5. Experimental results

This section demonstrates the effectiveness of FcCA through practical experiments by testing the effects of different numbers of CA generations, G, on three test sets, numbered I, II, and III, shown in each Part (a) of Tables 2, 3(i), 3(ii), 3(iii) and 4. The CA has open boundaries. The implementation in Table 2 used parameter values $N_1 = N_2 = 10$, iDL = (3, 3), and Ps = {3, 4, 7, 9, 11, 15, 16, 17}. In Table 3(i), random iDL points were used with $N_1 = N_2 = 0$ in Table 3(i) and N1 = N_2 = 100 in Table 3(iii). The implementations in Tables 2 and 3(i), 3(ii), 3(iii) were applied with values of G = 1 and 17.

The first set, Set I (shown in Table 2), consists of four greyscale images, each with a size of 256 × 256 pixels: Car, Mountains, Tree, and Von Neumann. This was the test set used by (Abu Dalhoum et al., 2012) with a confusion component. The second set, Set II (shown in Table 3(i)), has fourteen RGB images, commonly used in encryption, with dimensions of 256 × 256 pixels each: Deaa, Jet, Baboon, House, Tiffany, Boat, Mountain, Barbara, Lena, Peppers, Lake, Splash, Gold hill, and Zelda. The last set, Set III (shown in Table 4), is a subset of the Princeton Shape Benchmark (Shilane et al., 2004). It contains ten 3D objects of different sizes: Airplane, Castle, Chair, Flowers, Gun, Man face, Satellite, Skeleton, Spider, and Tank. This will be the validation set. An existing graph toolbox package by (Peyre, 2004) in MATLAB R2013a was used to read 3D objects as matrices.

Tables 2, 3(i), 3(ii), 3(iii) and 4 show Sets I, II, and III consisting of greyscale images, RGB images, and 3D objects, respectively. The tables show when CA was played with one or 17 generations to encrypt these files. As seen in these tables, the encrypted data become visually unrecognisable with only one CA generation, but 17 generations provide better encryption that makes the encrypted data appear random. The decipher component of the cryptosystem was able to correctly decrypt all encrypted images and objects to recover the original plain ones by using the correct secret keys. The decrypted images and objects were exactly the same as the originals, without any loss of information or precision.

6. Analyses and discussion

The analyses of the security aspects of the cryptosystem using different methods will be shown in this section. The analyses were performed by examining the implementation results of the FcCA cryptosystem obtained for Sets I and III, in addition to some other benchmark images. The analyses criteria are statistical analysis, key space, benefits of random boundaries, and sensitivity analysis. To compare some results of FcCA to previous systems, it was also tested with images used by those systems, such as Lena, Baboon, Peppers, Boat, and Cameraman where all of these were 256 × 256-pixel greyscale images. Since these systems were not all tested with all of these images, only available data are reported and compared in this section.

6.1. Statistical analysis

6.1.1. Neighbours' correlation analysis

Value Difference Degree (VDD) can be used to show how the number of CA generations affects the encryption level. This encryption level can be tested by computing the neighbours' correlation of all neighbouring pixels in the scrambled image or scrambled 3D object. To compute VDD, start with computing Value Difference (VD) of each pixel value $P(i, j)$, except for pixels at the boundaries, for the pre-processing matrices described in Section 3.1. VD for pixel $P(i, j)$ is computed by (7).

$$VD(i,\ j) = 1/4 \sum i'j' \left[P(i,\ j) - P(i',j')\right]^2 \tag{7}$$

where the neighbourhood of the pixel at position(i, j) is (i', j') = {$(i-1, j)$, $(i+1, j)$, $(i, j-1)$, $(i, j+1)$}, and $P(i, j)$ is the grey value at position (i, j) of the image matrix, or the coordinate value at position (i, j) of the object matrix. After computing VD, compute the Average Value Difference (AVD) for the whole image or object. Finally, VDD can be obtained by computing the difference of AVD values for plain input and scrambled output, divided by their sum, as in (8). The value of VDD should be a number between −1 and 1 where a value near 1 indicates better encryption.

$$VDD\ = (AVD_{original} - AVD_{encrypted}) \ /\ (AVD_{original} + \ AVD_{encrypted}) \tag{8}$$

The implemented algorithm was run 10 times for various G values, and with different initial states of the CA for encrypting each image and each object. It obtained 10 VDD values for each image and object, and 10 values for each G value, and computed their averages. Tables 5 and 6 show the results of VDD for the scrambled

Table 2
Set I (a) Original images (b) Scrambling with g = 1 (c) Scrambling with g = 17.

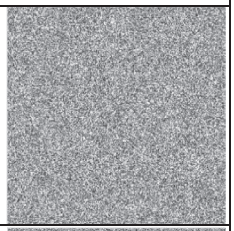
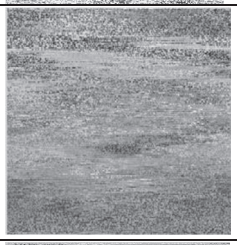
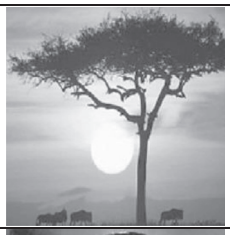
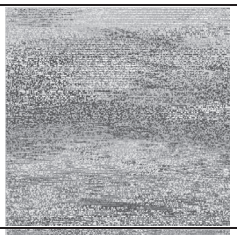
Set I	a	b	c
Car			
Mountains			
Tree			
Von Neumann			

Table 3a
Set II Images used in FcCA experimental tests. In order, left to right: Deaa, Jet, Baboon, House, Tiffany, Boat, Mountain, Barbara, Lena, Peppers, Lake, Splash, Gold hill, Zelda.

images of Set I and Set II. For all images in Set I, the confusion effect of the proposed FcCA cryptosystem was clearly larger (i.e., better) than those of (Abu Dalhoum et al., 2012). In addition, VDD for the images in Set II increased as the number of GA generations increased, with the exception of a small relapse when the number of generations was 15.

Tables 7 and 8 present the numerical results of VDD obtained from the matrices of Set III objects. It can be observed that the values were affected by the number of CA generations, where VDD increased as it increased, except for a few incidences of a slight decrease (≈0.01). The results for all objects in the two tables show that FcCA was highly effective even for a small number of generations, but it was more effective with $G \geq 5$.

6.1.2. Balancedness of cellular automata

In every generation, the generated CA needs nearly the same number of ones and zeros for it to be exhibited in a balanced way. Fig. 9 illustrates the ones' percentage in each generation over ten experiments on Sets I, II, and III. The figure shows that Set I and Set II images, and Set III objects all have ones' and zeros' ratios of approximately 50.1% and 49.9% of the total number of ones and zeros, respectively. That is, the number of ones is approximately equal to the number of zeros in these sets.

6.1.3. Propagation of coordinates across generations

Observations were made of the propagation of cell values in matrices representing images and objects over different genera-

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

Table 3b
Set II with $N_1 = 0$, $N_2 = 0$ (a) Scrambling with G = 1 (b) Scrambling with G = 17.

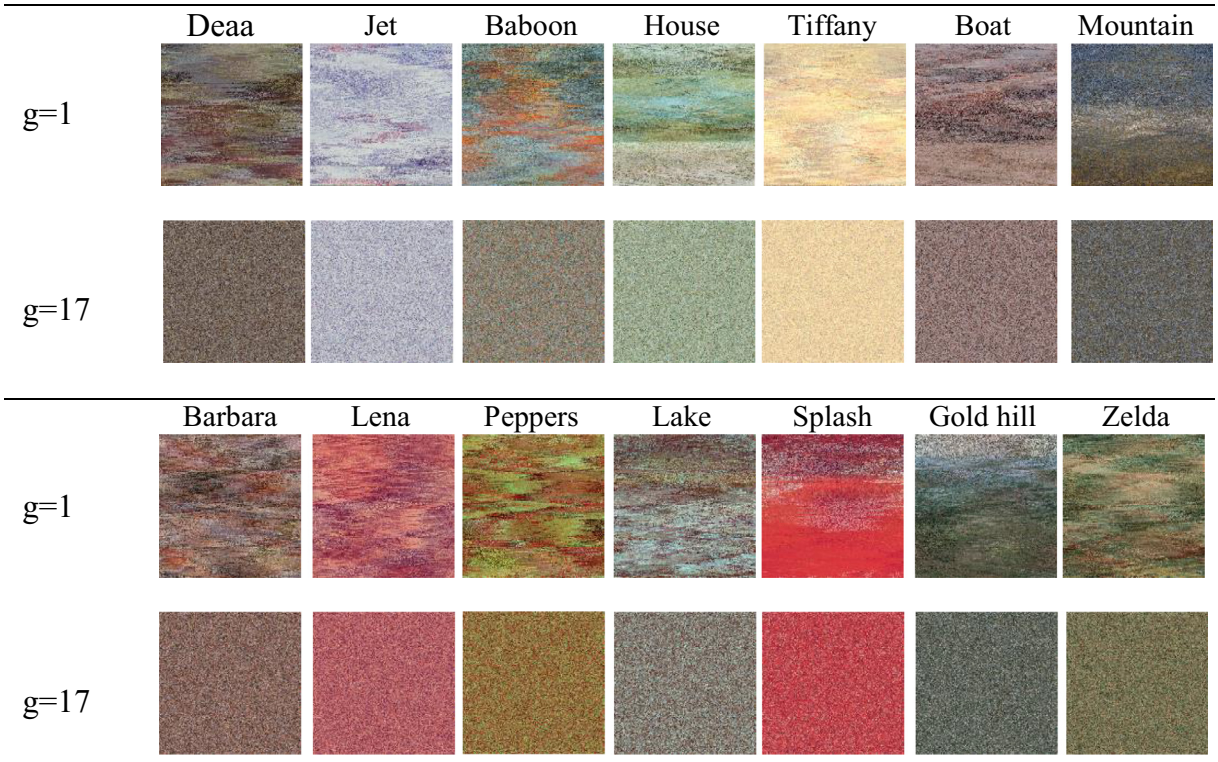
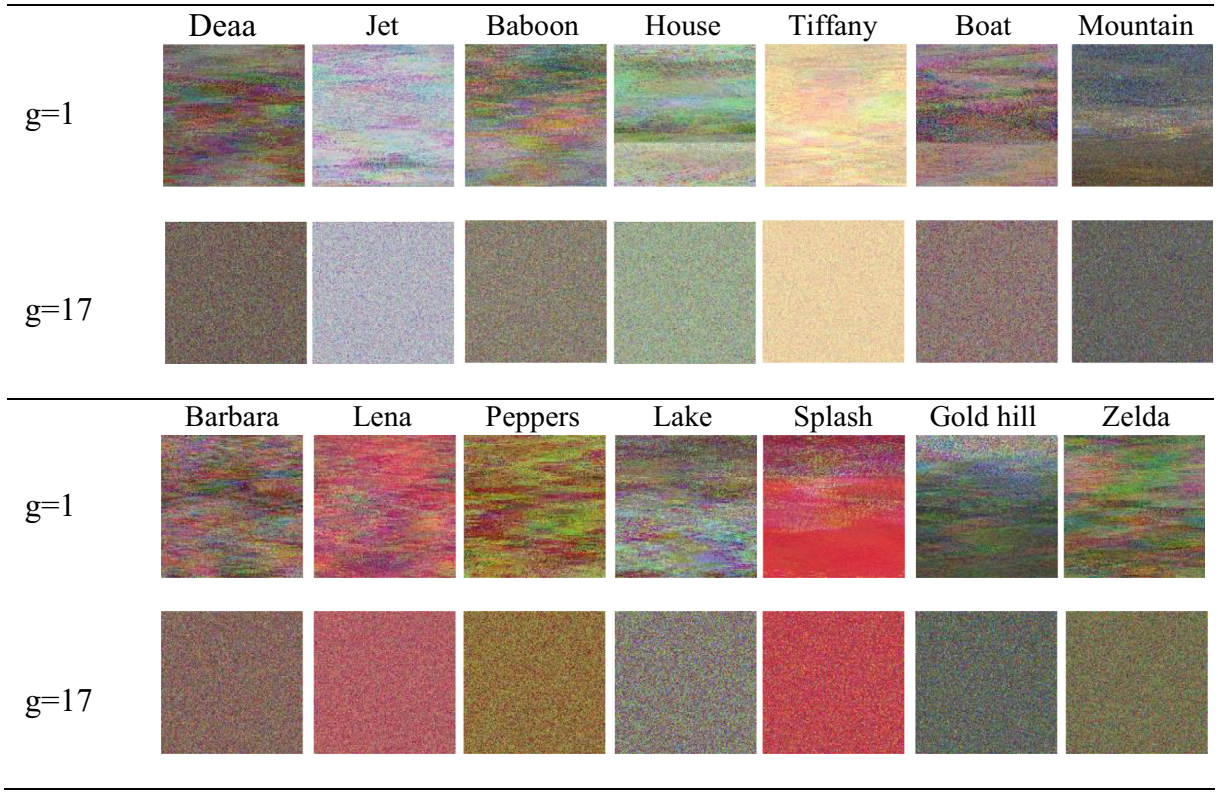


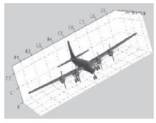
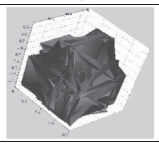
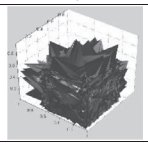
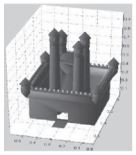
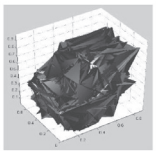
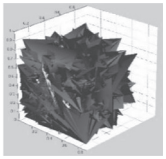
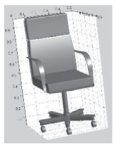
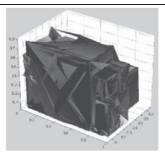
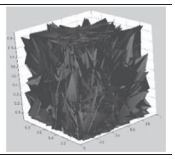
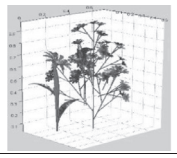
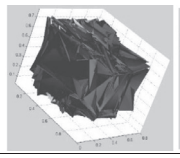
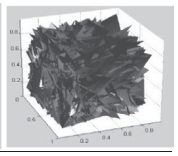
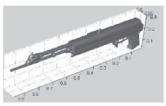
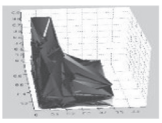
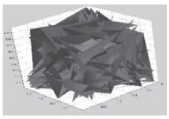
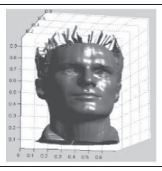
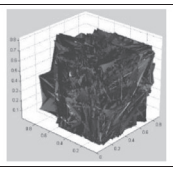
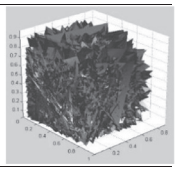
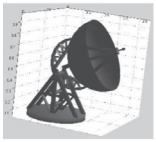
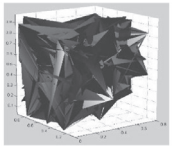
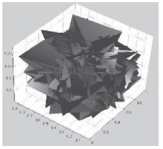
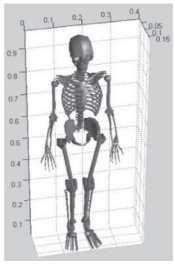
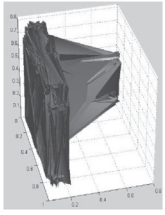
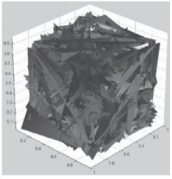
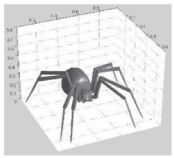
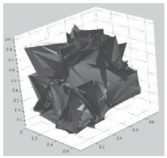
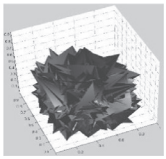
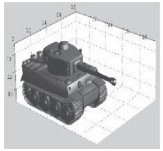
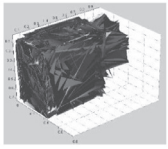
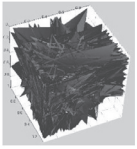
Table 3c
Set II with $N_1 = 100$, $N_2 = 100$ (a) Scrambling with G = 1 (b) Scrambling with G = 17.



tions. This was performed by comparing each cell's numeric value in the plain image or object with the cell that has the same index in the cipher image or object. Then, the average of dissimilarity was computed over all generations for all ten experiments. As an example, Table 9 shows that the matrices associated with the Car image and the Airplane object, over all generations and experiments,

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

Table 4
Set III (a) Original 3D objects (b) Encryption with G = 1 (c) Encryption with G = 17.

Set III	a	b	c
Airplane: Face [3,24430] Vertex [3,13575]			
Castle: Face [3,4460] Vertex [3,2328]			
Chair: Face [3,16268] Vertex [3,8188]			
Flowers: Face [3,24673] Vertex [3,15492]			
Gun: Face [3,2538] Vertex [3,1414]			
Man face: Face [3,316498] Vertex [3,160940]			
Satellite: Face [3 ,1472] Vertex [3,773]			
Skeleton: Face [3,11092] Vertex [3,6100]			
Spider: Face [3,3042] Vertex [3,1624]			
Tank: Face [3,19743] Vertex [3,10241]			

were extremely different from their corresponding encrypted matrices.

6.1.4. Peak signal to noise ratio

Peak Signal to Noise Ratio (PSNR), measured in decibels, is usually used to judge the quality of a reconstructed image after image processing. However, it may be used to show the difference between encrypted and plain images. A lower PSNR value is desired for encrypted images since it indicates more noise and, therefore, more resistance to attacks. As seen in Tables 10 and 11, PSNR values obtained by FcCA are relatively low, indicating an acceptable level of encryption. Since FcCA decryption recovers the exact original plain images, their PSNR values were infinite.

6.1.5. Entropy

To measure the randomness of pixel values in an image, entropy may be used. The entropy value of an encrypted image must be higher than the entropy value of its original image to indicate increased randomness. The increase in randomness can be expressed by the entropy ratio, give in (9).

$$entropy\ ratio = (entropy(scrambled\ image) - entropy(plain\ image))\ /block\ size \quad (9)$$

Table 12 lists entropy ratios obtained by FcCA and other methods for different images where block size = 8. The high entropy ratios indicate more randomness, and consequently, higher resistance to statistical attacks. As seen in the table, FcCA produced acceptable entropy ratios and outperformed other algorithms, except for (Wu et al., 2018).

6.1.6. Correlation

Correlation is a measure of resemblance between images. The statistical correlation between similar images can be used to extract information, such as component displacement. A lower correlation value between a plain image and its encrypted image indicates less resemblance between them and more resistance to attacks. The correlation values for different images, listed in Table 13, compare the results of FcCA to previous methods. It was computed in the diagonal, horizontal, and vertical directions. It can be noted from the table that FcCA produced relatively similar results to other algorithms, as it outperformed them with some images and underperformed them in others. Overall, the low correlation values produced by FcCA show an acceptable level of encryption.

6.2. Key space

In the encryption scheme, key space should be sufficiently extensive to resist brute-force attacks. Therefore, the secret key used in the FcCA cryptosystem is partitioned into two flexible subkeys: HCA and a numeric set containing the Ps series and iDL coordinates. Contrary to most previous encryption schemes with traditional CA, this partitioning provides added flexibility to freely and easily change the sizes of these two subkeys, depending on the desired level of security. The dimensions of CA and the length of generated keys roughly depend on the dimensions of the plain image or object. On the other hand, the lengths of subkeys depend on several parameters; namely the size of HCA, the (G + Ps)-bit length of each cell in HCA, the number of elements in the Ps series, and the coordinates of iDL points, as seen in (10) through (13). In these equations, *space*(*key*) indicates the number of possible values that *key* may have.

$$space(HCA\_bits\_key) = 2^{(rowsCount(HCA))\times(columnsCount(HCA))\times G} \quad (10)$$

Table 5
VDD for Set I matrices.

	G = 1	G = 5	G = 9	G = 13	G = 17	Average
Car	0.885227	0.900852	0.903988	0.905109	0.905419	0.900118861
Mountains	0.930358	0.940873	0.942642	0.943063	0.943188	0.940025056
Tree	0.937378	0.945716	0.946958	0.947241	0.947482	0.944955064
Von Neumann	0.9584	0.961762	0.963448	0.963874	0.964004	0.962297787

Table 6
VDD for Set II matrices when $N_1 = 0$, $N_2 = 0$.

	G = 1	G = 5	G = 9	G = 13	G = 17
Deaa	0.4692	0.4890	0.4974	0.4974	0.4978
Jet	0.3700	0.3853	0.3935	0.3951	0.3943
Baboon	0.0719	0.0785	0.0795	0.0797	0.0793
House	0.2846	0.3157	0.3186	0.3195	0.3187
Tiffany	0.3889	0.4126	0.4179	0.4188	0.4188
Boat	0.3003	0.3264	0.3306	0.3307	0.3304
Mountain	0.1335	0.1499	0.1508	0.1510	0.1519
Barbara	0.2183	0.2364	0.2386	0.2390	0.2388
Lena	0.3381	0.3503	0.3524	0.3542	0.3541
Peppers	0.3073	0.3308	0.3347	0.3348	0.3351
Lake	0.2695	0.2839	0.2868	0.2872	0.2871
Splash	0.5424	0.5784	0.5829	0.5837	0.5837
Gold hill	0.2295	0.2459	0.2508	0.2517	0.2518
Zelda	0.3940	0.4184	0.4201	0.4205	0.4206

Table 7
VDD for Set III face matrices.

	G = 1	G = 5	G = 9	G = 13	G = 17
Airplane	0.7676714	0.7925326	0.7752839	0.7809158	0.7925828
Castle	0.9979806	0.9983408	0.9984034	0.9984323	0.9984374
Chair	0.9997098	0.9997616	0.9997706	0.9997726	0.9997735
Flowers	0.9990866	0.999248	0.9992996	0.9993102	0.9993202
Gun	0.7689505	0.8401231	0.8534815	0.8554567	0.8582629
Man	0.9998992	0.9999183	0.9999216	0.9999225	0.9999228
Satellite	0.9848066	0.9865657	0.9872409	0.9873442	0.987536
Skeleton	0.9997648	0.9998088	0.999816	0.999817	0.9998174
Spider	0.9984149	0.9986867	0.9987417	0.998745	0.9987545
Tank	0.8215543	0.9249282	0.9331278	0.9344721	0.9353561

Table 8
VDD for Set III vertex matrices.

	G = 1	G = 5	G = 9	G = 13	G = 17
Airplane	−0.1446775	0.0061477	0.0039949	0.0170072	0.0259561
Castle	−0.0053184	0.1701983	0.1713969	0.1333991	0.1478647
Chair	−0.1597918	0.0178647	0.0216309	0.0377627	0.0447795
Flowers	−0.226577	−0.0126695	0.0212567	0.0245574	0.0207282
Gun	0.6214571	0.4973528	0.4855446	0.4968812	0.5052729
Man	0.1876535	0.2998939	0.3180451	0.319114	0.3191007
Satellite	0.335882	0.3744447	0.4158016	0.4041317	0.386403
Skeleton	−0.2955229	−0.0225172	−0.0192128	0.0179739	0.028027
Spider	−0.0102544	0.1484157	0.1676528	0.1555408	0.1613567
Tank	0.2225347	0.3131175	0.3006308	0.2536257	0.2767594

$$space(Bogus\_bits\_key) = 2^{(Number\_Of\_element(Ps))\times G} \tag{11}$$

$$space(iDL\_key) = 10^{(N_1\times N_2)\times(\ldots)\times(N_1\times N_2)} < \left(2^4\right)^{(N_1\times N_2)\times(\ldots)\times(N_1\times N_2)} \tag{12}$$

$$space(Total\_Key) = 2^{(space(HCA\_bits\_key)\ +\ space(Bogus\_bits\_key)\ +\ space(iDL\_key))} \tag{13}$$

For example, consider the encryption of the Satellite object. Satellite has two matrices: Face (3 × 1472) and Vertex (3 × 773). Let the number of generations G = 17, with Ps = {0, 1, 2, 4, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19} and $N_1 = N_2 = 1000$. Then, the HCA matrix will have size = (1472 + 1000) × (3 + 1000) = 2,479,416 cells, and the number of possibilities for HCA_bits_key will be $2^{(2{,}479{,}416\ \times\ 17)}$. This huge number makes it improbable to predict all true bits in the first subkey. Additionally, the second subkey has two parts. The first part is for determining which bits in the cells of the bit-series are true and which are bogus. In this example, the number of possibilities for Bogus_bits_key is 2^{32}. The second part is for determining the true coordinates of start points; iDL. The number of possibilities for iDL_key coordinates in this example is computed as in (14).

$$\begin{aligned} &size(face(iDL(x))) \times size(face(iDL(y))) \times size(vertex(iDL(x))) \\ &\quad \times size(vertex(iDL(y))) \\ &= 1000 \times 1000 \times 1000 \times 1000 = 10^{12} < \left(2^4\right)^{12} = 2^{48} \end{aligned} \tag{14}$$

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

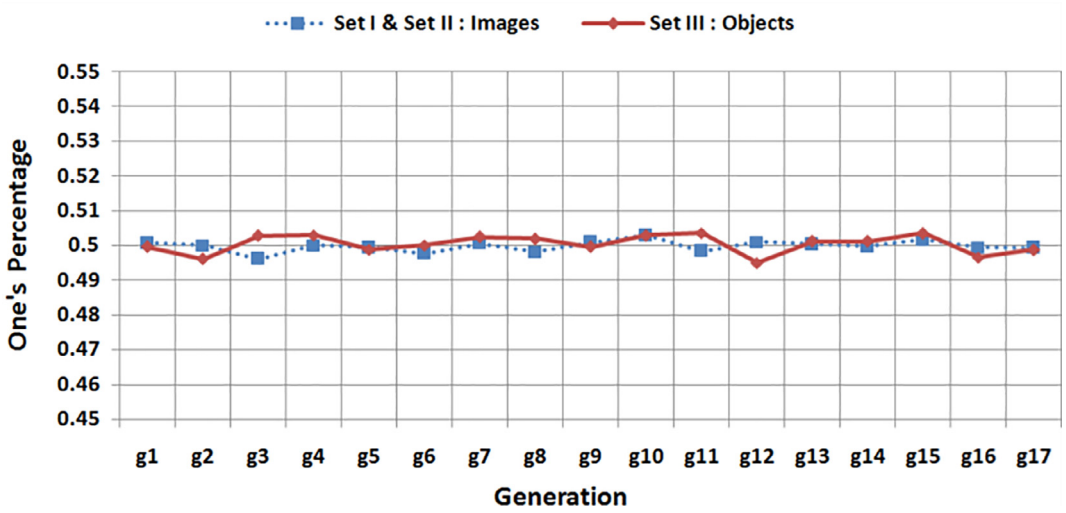


Fig. 9. Balancedness of CA.

Table 9
Example of average dissimilarity for the propagation of coordinates.

	G = 1	G = 5	G = 9	G = 13	G = 17	Average
Car	0.992885	0.994516	0.994951	0.995238	0.994911	0.99450012
Airplane Face	0.999921	0.999917	0.999921	0.99991	0.999913	0.99991622
Airplane Vertex	0.999128	0.999676	0.999607	0.999649	0.999575	0.99952707

Table 10
PSNR values for Set II images encrypted with FcCA.

Deaa	Jet	Baboon	House	Tiffany	Boat	Mountain
11.08888	12.16794	10.03442	11.23088	15.72784	10.04729	11.10458
Barbara	Lena	Peppers	Lake	Splash	Gold hill	Zelda
9.62995	11.70098	10.33371	8.31383	11.46793	11.36029	10.94501

Table 11
PSNR values for different images encrypted with FcCA and other schemes.

Paper	Lena	Baboon	Peppers	Boat	Cameraman
(Chai et al., 2018)	26.06	–	–	–	25.23
(Liu et al., 2018)	9.2089	9.5608	9.0299	–	–
FcCA	10.60396	12.17228	8.86827	10.49501	9.15323

Table 12
Entropy ratio for different images encrypted with FcCA and other schemes.

Paper	Lena	Baboon	Peppers	Boat	Cameraman
(Hosseini and Kamel, 2018)	0.0536625	–	–	–	–
(Liu et al., 2018)	0.0530875	0.0823625	0.05905	–	0.123425
(Noshadian et al., 2018)	0.053525	0.0822125	0.0587375	–	–
(Wu et al., 2018)	0.5453625	0.555875	0.522025	0.588125	0.5618875
(Ye et al., 2018)	0.053595875	–	0.059286375	–	–
FcCA	0.239398	0.0840250	0.2208725	0.1980595	0.2391740

So, the number of possibilities for Total_key will be $2^{(2,479,416 \times 17) + 32 + 48}$. As seen in Table 14, when encrypting a grey-scale 256 × 256 image, this key space is significantly larger than those of previous methods.

Consequently, the subkeys' spaces are very flexible. The keys can be easily changed to become smaller or larger, and less or more complex. To show how key space can greatly affect the key's complexity, the FcCA cryptosystem was implemented on a laptop with Intel Core i5-4200U CPU @1.60 GHz, 2.30 GHz, x64-based processor, RAM size = 4 GB, and 64-bit Windows 8.1. As an example, consider encrypting and decrypting the Satellite object with 17 generations and various HCA expansions. When N_1 and N_2 values were 1000, it required 30,357 h to try just one possibility of 10^{12} possibilities. However, it required 1.495 h with $N_1 = N_2 = 100$, and only 0.21 h with $N_1 = N_2 = 10$.

6.3. Benefit of random boundaries and configurations

The initial configuration of expCA was constituted randomly, and randomisation was used in creating the generations of expCA,

Table 13
Correlation values for different images encrypted with FcCA and other schemes.

Reference	Direction	Lena	Baboon	Peppers	Boat	Cameraman
(Hosseini and Kamel, 2018)	Diagonal	0.0035	–	–	–	–
	Horizontal	−0.0019	–	–	–	–
	Vertical	−0.0016	–	–	–	–
(Liu et al., 2018)	Diagonal	0.0074	−0.004750	0.003140	–	0.009544
	Horizontal	0.0000	0.001650	0.004779	–	0.001472
	Vertical	−0.0011	−0.001467	0.006250	–	−0.000367
(Noshadian et al., 2018)	Diagonal	0.0019	–	–	–	–
	Horizontal	−0.0012	–	–	–	–
	Vertical	0.0030	−0.0026	0.0033	–	–
(Ping et al., 2018)	Diagonal	0.000661	–	–	–	−0.008626
	Horizontal	−0.001664	–	–	–	0.008362
	Vertical	0.002254	–	–	–	−0.007951
(Wu et al., 2018)	Diagonal	0.0032	0.0052	0.0034	0.0015	0.0098
	Horizontal	0.0056	0.0026	0.0016	0.0001	0.0024
	Vertical	0.0037	0.0009	0.0059	0.0031	0.0013
FcCA	Diagonal	0.001484	0.005895	0.001238	0.001181	0.004272
	Horizontal	0.003394	0.005895	−0.002668	0.003106	0.000057
	Vertical	0.003627	−0.004385	0.005793	0.007313	0.0056326

Table 14
Key space for a 256 × 256 greyscale image.

Reference	Approximate Key Space
(Chai et al., 2018)	2^{232}
(Hosseini and Kamel, 2018)	2^{392}
(Noshadian et al., 2018)	2^{256}
(Ping et al., 2018)	2^{512}
(Wu et al., 2018)	2^{100}
(Ye et al., 2018)	2^{224}
FcCA	$2^{26,818,168}$

so the configuration of each generation is completely independent of the other generations. This increases the security of expCA because revealing a configuration would not easily allow calculating other generations. In most traditional techniques that used CA, a brute-force attacker only needs to predicate the initial configuration of CA, while in FcCA, however, such attacker needs to predicate every bit in all generations of CA. In addition, if HCA was hacked, the hacker will be incapable of decrypting ciphered data due to the use of iDLs and Ps, especially when expCA is in its maximum extension. Similarly, hacking iDL points or Ps is not sufficient for decrypting the ciphered data. Moreover, randomization also raises the level of encryption by decreasing cell correlation in the configuration of expCA generations.

6.4. Sensitivity analysis

6.4.1. Key sensitivity

An important feature of encryption systems is key sensitivity. That is, a minor change in the secret key should result in a major mutation in the encrypted image or object. Implementation results of the FcCA cryptosystem showed it to be sensitive to the secret key, where a decryption attempt with a minor modification of the key produced a significantly different decrypted image or object. It was also noticed that such decrypted images or objects remained unrecognisable, providing no clues to the appearance of the originals. The mean square error (MSE) can provide an indication of key sensitivity, where MSE between the plain image and decrypted image is computed by (15).

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{N} \sum_{j=1}^{M} |I(i,j) - D(i,j)|^2 \quad (15)$$

where I(i, j) and D(i, j) are the plain image and its corresponding decrypted image, respectively. The decryption procedure is applied by changing the values of one parameter while keeping the other parameters constant.

Consider the Satellite object as an example. The appearance of the decrypted Satellite object with the modified keys, shown in Fig. 10, is very different from the original object. The objects illustrated in the figure resulted when the following parameters were modified:

a) X Coordinate of (iDL) in the Face matrix was modified by adding one, changing its value from 367 to 368.
b) Sequentially, X and Y coordinates of (iDL) in the Face matrix were modified by adding one to each of them, so X changed from 367 to 368 and Y from 193 to 192.
c) Deleted one true index instead of a bogus index from the Ps list.
d) Randomly changed just 10 bits from: only face indices in HCA matrix (Fig. 10(d-1)); only Vertex indices in HCA matrix (Fig. 10(d-2)); or both Face and Vertex indices in HCA matrix (Fig. 10(d-3)).

6.4.2. Pixel sensitivity

One of the main features of an encryption algorithm is the sensitivity of the cipher image to trivial changes in the plain image. More sensitivity of the cipher image will create more resistance to differential attacks since it would indicate no meaningful relationship between the plain image and the cipher image. In order to measure the effect of making a limited change to the plain image on all pixels of the cipher image, three common measures were used: Mean Absolute Error (MAE), Unified Average Changing Intensity (UACI), and Number of Pixels Change Rate (NPCR).

MAE may be used to compare the plain image with its encrypted image or with its decrypted image where a higher MAE value indicates more dissimilarity between the compared images. As seen in Tables 15 and 16, MAE was over 0.63 for encrypted images in Set I and mostly over 0.65 for Set II, which indicates their high dissimilarity to the plain images. MAE for all decrypted image was 0, which means they were exactly the same as their respective plain images.

NPCR values are commonly used to evaluate the strength of image encryption algorithms with respect to differential attacks where higher values indicate higher resistance to these attacks. NPCR values obtained by FcCA and other schemes are listed in Table 17. It can be observed from this table that NPCR values obtained by FcCA were generally higher than those by other algorithms.

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

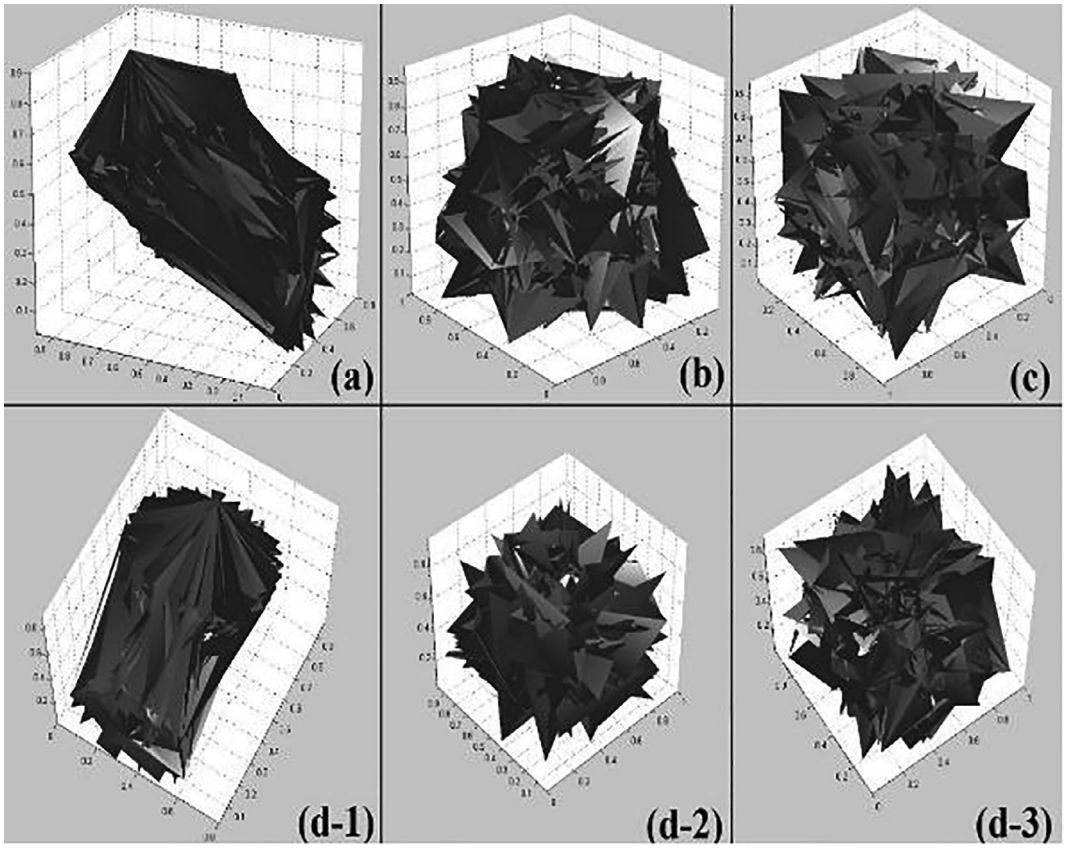


Fig. 10. Sensitivity example: (a) Modify X in iDL (b) Modify X and Y in iDL (c) Delete true index from Ps (d) Randomly change only 10 bits: (d-1) from Face, (d-2) from Vertex, and (d-3) from Face and Vertex.

Table 15
MAE for different images encrypted with FcCA.

	Lena	Baboon	Peppers	Boat	Cameraman
MAE	0.6858	0.7289	0.6340	0.7255	0.6448

Table 16
MAE for Set II images encrypted with FcCA when g = 17.

Deaa	Jet	Baboon	House	Tiffany	Boat	Mountain
0.8708	0.6509	0.9763	0.8555	0.4651	0.9135	0.8362
Barbara	Lena	Peppers	Lake	Splash	Gold hill	Zelda
1.0071	0.7752	0.9129	1.117	0.7128	0.9495	0.8657

For further evaluation of resistance against differential attacks, UACI was used to compute the average intensity difference for the encrypted images. As with NPCR, higher UACI values indicate higher resistance to differential attacks. The UACI scores for FcCA are shown in Table 18. At first look, these values indicate some resistance to differential attacks. However, (Yue et al., 2011) provided theoretical boundaries for desired UACI values depending on image type and size. For grayscale images of 256 × 256 pixels, the UACI values should be between 0.331594 and 0.337677. The values in Table 18 fall below this lower bound. Consequently, according to the theoretical UACI metrics, FcCA performance needs further improvement.

To evaluate FcCA resistance to statistical attacks, histogram analysis was used. The histograms of Set II plain images and their respective ciphered images are shown in Table 19. As seen in the table, the histograms of the ciphered images were more uniform and generally different from those of the plain images. If the histograms do not show enough change after encryption, the diffusion strength of FcCA may be enhanced by increasing the number of GA generations, expanding HCA, and manipulating the iDL values.

6.4.3. Occlusion attack analysis

Some encrypted data may be lost during storage or transmission, so the encryption algorithm should still be able to recover most of the original image. To test FcCA robustness against data

Table 17
NPCR for different images encrypted with FcCA and other schemes.

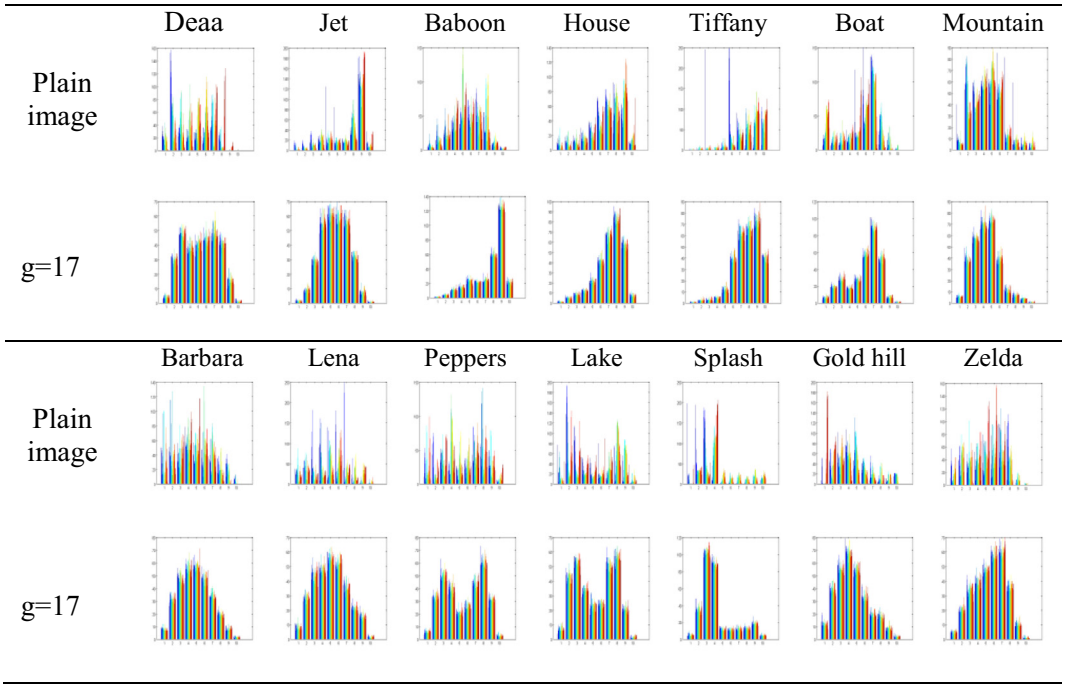
Reference	Lena	Baboon	Peppers	Boat	Cameraman
(Chai et al., 2018)	0.9678				
(Hosseini and Kamel, 2018)	0.996643	–	–	–	–
(Liu et al., 2018)	–	0.996006	0.996008	–	0.9960345
(Noshadian et al., 2018)	0.996446	0.996201	0.996109	–	–
(Ping et al., 2018)	0.996188	–	–	–	0.995959
(Wu et al., 2018)	0.996200	0.995925	0.996078	0.996170	0.996139
(Ye et al., 2018)	0.99620056	–	0.99599838	0.99603653	–
FcCA	0.99987	0.999984741	0.9999237	0.990341187	0.990600586

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

Table 18
Comparison of UACI for different encryption schemes.

Paper	Lena	Baboon	Peppers	Boat	Cameraman
(Hosseini and Kamel, 2018)	0.3043	–	–	–	–
(Liu et al., 2018)	–	0.3345	0.3347	–	0.3347
(Noshadian et al., 2018)	0.3414	0.3356	0.3317	–	–
(Ping et al., 2018)	0.3351	–	–	–	0.3347
(Wu et al., 2018)	0.3341	0.3338	0.3349	0.3366	0.3353
(Ye et al., 2018)	0.3347	–	0.3346	0.3339	–
FcCA	0.2394	0.1991	0.2930	0.2243	0.2672

Table 19
Histogram for Set II images and for encrypted images when g = 17.



loss, encrypted images were partially occluded and then decrypted. As seen in the example in Fig. 11, occluding 25%, 50%, and 75% of the encrypted image caused distortion to the deciphered image, but it did not prevent recovering a meaningful image.

6.4.4. Noise attack analysis

The robustness of the proposed encryption algorithm was also tested against noise attacks. Different types of noise were added to sample encrypted images, and then they were decoded. The noise types included zero-mean Gaussian noise with different intensities, salt and pepper noise, and Speckle noise. Table 20 shows decrypted Lena images with noise intensity values $\sigma = 0.01$ and 0.10. As seen in the example in this table, the results demonstrated that FcCA was able to recover meaningful images, especially when noise intensity was less than 10%. Hence, it was effective in resisting noise attacks.

6.5. Time complexity analysis

In FcCA, the main techniques that consume time are diffusion and confusion. In diffusion, the time complexity depends on two factors: the size of extended cellular automata (expCA), and the number of generations (G). In confusion, the time complexity depends on two factors: G and the size of plain data matrices (PlnM). Let the time taken by a given procedure or algorithm be computed in terms of the number of elementary operations, as a function of the size of input. Then, the time required by FcCA can be approximated as detailed in (16) through (23).

$$Diffusion\_time = size(expCA) \times G \tag{16}$$

$$Confusion\_time = size(PlnM) \times G \tag{17}$$

$$FcCA\_time \approx Diffusion\_time + Confusion\_time \tag{18}$$

Here, G is a small constant, and in the worst case, the size of expCA is larger than that of plain data matrices by an increment of $L \approx 1000 \times 1000 = 10^6$. Therefore, the diffusion and confusion times can be computed as follows:

$$Diffusion\_time = (size(PlnM) + L) \times G \tag{19}$$

$$Confusion\_time = size(PlnM) \times G \tag{20}$$

$$FcCA\_time \approx ((size(PlnM) + L) \times G) + (size(PlnM) \times G) \tag{21}$$

$$FcCA\_time \approx (size(PlnM) \times G) + (L \times G) + (size(PlnM) \times G) \tag{22}$$

$$FcCA\_time \approx G \times (2 \times size(PlnM)) + L) \tag{23}$$

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

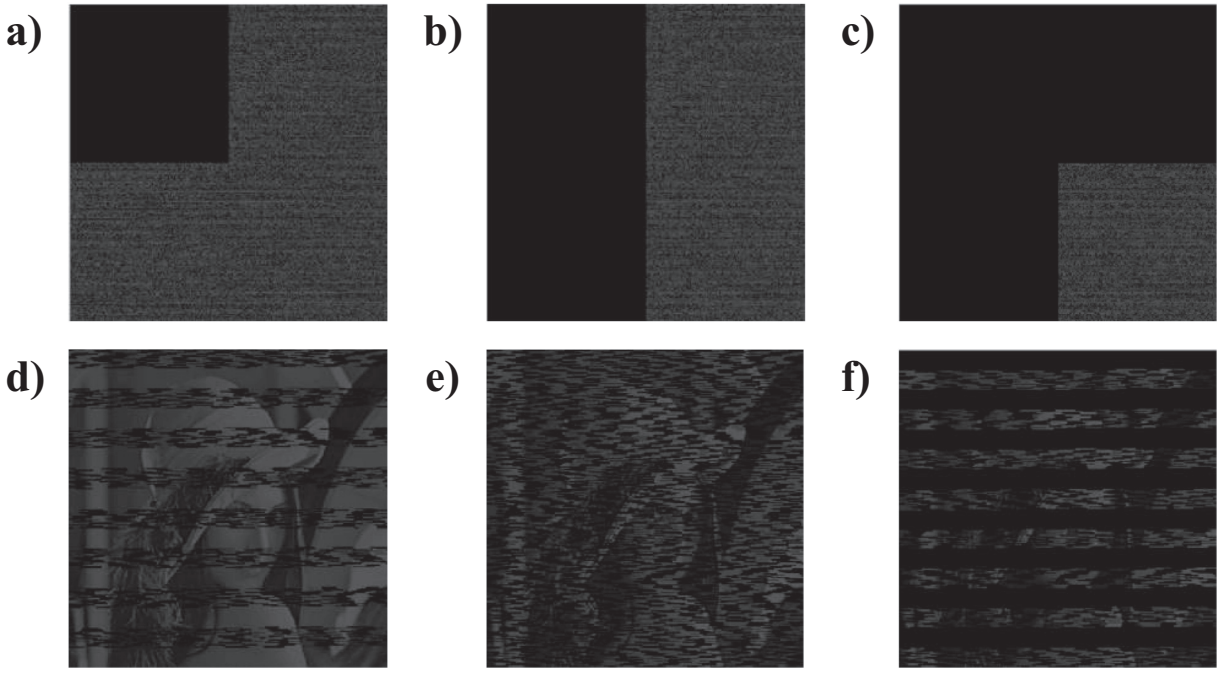
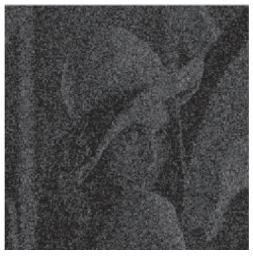
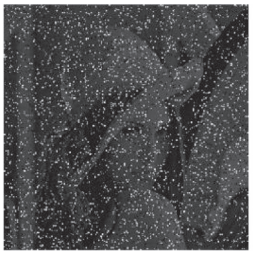
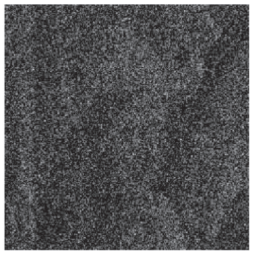


Fig. 11. Occlusion attack example. Parts (a), (b), and (c) show encrypted Lena image occluded by 25%, 50%, and 75%, respectively. Their respective decrypted images are shown in Parts (d), (e), and (f).

Table 20
Decrypted images with noise of different types and intensities.

Intensity	Speckle	Peppers & Salt	Gaussian
0.01			
0.10			

Since the size of PlnM is a constant multiple of the size of input, and both L and G are constants, the time complexity of FcCA is a linear function of the size of input. In addition, a small number of extended matrices is required by FcCA, where the size of each matrix is a constant multiple of the size of an input matrix. This bounds the space complexity of the algorithm to a linear function of the size of input.

Consequently, when using FcCA for a larger key space in conventional cryptography environments, it is more efficient than other cryptosystems. Furthermore, using one CA (expCA) for encrypting all plain data matrices instead of multiple CA (different CA for each plain data matrix) reduces the time and space complexity of the whole system by a constant factor.

6.6. Overall discussion

The proposed FcCA system has three distinct advantages compared to previous systems such as those listed in Table 1. Firstly, the FcCA scheme outperforms (Elsheh and Ben Hamza, 2011; Éluard et al., 2013; Jin et al., 2017), because FcCA can control the number of times the cryptosystem can repeat the shuffling and permutation mechanisms, while others applied these mechanisms only once during encryption. This means FcCA can increase its encryption level compared to others who have fixed encryption levels. Furthermore, the algorithm by (Éluard et al., 2013) is not completely random, so it always produces the same encrypted object for each encryption of the plain object.

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

Secondly, most traditional cryptosystems have fixed sizes for their key spaces while FcCA has extendable and resizable keys. This makes it suitable for more types of data that can be represented with 2D matrices, and it allows changing the security level to suit different applications. Hence, FcCA combines key flexibility with encryption level flexibility to introduce a very flexible system.

Thirdly, FcCA applies novel simplified rules for reverse 2D CA and for encryption. Many previous algorithms depended on traditional complex encryption rules or maps, or they employed heavy 3D cellular automata. Other previous algorithms used encryption with Arnold's Cat Map for its simplicity, but this has the drawback of insufficient security due to its periodic nature and work with square images.

Lastly, the FcCA key space is much larger than the key space proposed by (Abu Dalhoum et al., 2012) where their key space for total_key is ($2 \times$ PlainMatrix_width $\times$ PlainMatrix _length $\times$ G). Furthermore, the FcCA key space can be extended to be larger than the key space by (Jin et al., 2017). Their key space for an object without texture was 10^{90}, which is between $2^{4 \times 90} = 2^{360}$ and $2^{5 \times 90} = 2^{450}$. In addition, FcCA keys are sensitive to minor change. Therefore, FcCA is sufficiently robust in resisting different brute-force and statistical attacks.

7. Conclusion and future work

A new system, called FcCA, was introduced with novel techniques for building an effective cryptosystem based on pure random cellular automata. It can perform encryption of different types of data including images and 3D objects defined by a mesh. Implementation results and analyses revealed FcCA characteristics that outperform previous algorithms, such as flexibility, simplicity, high level of encryption, and high robustness against brute-force, differential, and statistical attacks. For future work, FcCA could use some improvements to its randomness technique to increase its effectiveness. In addition, new cryptography algorithms may be developed for encrypting other formats of 3D objects, and for handling more 3D properties such as light and animation.

Conflict of interest

None.

References

Abdalla, A., Tamimi, A., 2013. Algorithm for image mixing and encryption. Int. J. Multimedia Appl. 5 (2), 15–21.

Abu Dalhoum, A., Mahafzah, B.A., Awwad, A.A., Aldhamari, I., Ortega, A., Alfonseca, M., 2012. Digital image scrambling using 2D cellular automata. IEEE Multi Media 19 (4), 28–36. https://doi.org/10.1109/MMUL.2011.54.

Alfalou, A., Brosseau, C., 2013. Implementing compression and encryption of phase-shifting digital holograms for three-dimensional object reconstruction. Opt. Commun. 307, 67–72. https://doi.org/10.1016/j.optcom.2013.05.053.

Alvy, S., 1969. Cellular Automata Theory. Tech. Rep. 2. Stanford Electronic Lab., Stanford Univ.. Accession Number: AD0707442.

Chai, X., Zheng, X., Gan, Z., Han, D., Chen, Y., 2018. An image encryption algorithm based on chaotic system and compressive sensing. Signal Process. 148, 124–144. https://doi.org/10.1016/j.sigpro.2018.02.007.

Diffie, W., Hellman, M.E., 1976a. Multi-user cryptographic techniques. In: AFIPS '76 Proceedings of the June 7–10, 1976, national computer conference and exposition, pp. 109–112. https://doi.org/10.1145/1499799.1499815.

Diffie, W., Hellman, M.E., 1976b. New directions in cryptography. IEEE Trans. Inf. Theory 22 (6), 644–654. https://doi.org/10.1109/tit.1976.1055638.

Elsheh, E., Ben Hamza, A., 2011. Secret sharing approaches for 3D object encryption. J. Expert Syst. Appl. 38 (11), 13906–13911. https://doi.org/10.1016/j.eswa.2011.04.197.

Éluard, M., Maetz, Y., Doërr, G., 2013. Geometry-preserving encryption for 3D meshes. In: Proceedings of Compression et Représentation des Signaux Audiovisuels (CORESA), Le Creusot, pp 7-12. https://doi.org/10.13140/RG.2.1.3925.6165.

Gutowitz, H., 1991. Cellular Automata: Theory and Experiment. Elsevier science publisher. ISBN: 0262570866-9780262570862.

Hanis, S., Amutha, R., 2018. Double image compression and encryption scheme using logistic mapped convolution and cellular automata. Multimedia Tools Appl. 77 (6), 6897–6912. https://doi.org/10.1007/s11042-017-4606-0.

Hosseini, S.A., Kamel, S.R., 2018. Increasing the security of image encryption based on intertwining logistic map and elementary cellular automata. Rev. Publ. 5 (15), 18–36.

Jin, X., Wu, Z.X., Song, C., Zhang, C., Li, X., 2016. 3D point cloud encryption through chaotic mapping. In: Proceedings of the 17th Pacific Rim Conference on Multimedia Information Processing, Xi'an. pp. 119–129.

Jin, X., Zhu, S.Y., Xiao, C.E., Sun, H., Li, X., Zhao, G., Ge, S., 2017. 3D textured model encryption via 3D Lu chaotic mapping. Sci. China Inf. Sci. 60,. https://doi.org/10.1007/s11432-017-9266-1 122107.

Jolfaei, A., Wu, X.W., Muthukkumarasamy, V., 2015. A 3D object encryption scheme which maintains dimensional and spatial stability. IEEE Trans. Inf. Forensics Secur. 10 (2), 409–422.

Li, S., Xiao, Y.-L., Wang, X., 2013. Three-dimensional information security combined fringe projection with double random phase encoding. Opt. Commun. 296, 35–40. https://doi.org/10.1016/j.optcom.2012.12.046.

Li, X.W., Cho, S.J., Tae Kim, S.T., 2014. A 3D image encryption technique using computer-generated integral imaging and cellular automata transform. J. Optik 125 (13), 2983–2990. https://doi.org/10.1016/j.ijleo.2013.12.036.

Liu, D.-D., Zhang, W., Yu, H., Zhu, Z.-L., 2018. An image encryption scheme using self-adaptive selective permutation and inter intra-block feedback diffusion. Signal Process. 151, 130–143. https://doi.org/10.1016/j.sigpro.2018.05.008.

Manoj, S., 2010. Cryptography and steganography. Int. J. Comput. Appl. 1 (12), 61–65. https://doi.org/10.5120/257-414.

Martín del Rey, Á., 2015. A method to encrypt 3D solid objects based on three-dimensional cellular automata. In: Onieva, E., Santos, I., Osaba, E., Quintián, H., Corchado, E. (Eds.), Hybrid Artificial Intelligent Systems. HAIS 2015. Lecture Notes in Computer Science, vol 9121. Springer, Cham. 10.1007/978-3-319-19644-2_36.

Martín del Rey, Á., 2017. Chaotic Encryption of 3D Objects. In: Ferrández, Vicente J., Álvarez-Sánchez, J., de la Paz López, F., Toledo, Moreo J., Adeli, H. (Eds.), Biomedical Applications Based on Natural and Artificial Computing. IWINAC 2017. Lecture Notes in Computer Science, vol. 10338. Springer, Cham. 10.1007/978-3-319-59773-7_47.

Martín del Rey, Á., Pastora, J.L.H., Sánchez, G.R., 2016. 3D medical data security protection. Expert Syst. Appl. 54, 379–386. https://doi.org/10.1016/j.eswa.2016.02.001.

McHenry, K., Bajcsy, P., 2008. An overview of 3D data content, file formats and viewers Technical Report: isda08-002. Image Spatial Data Analysis Group National Center for Supercomputing Applications.

Nandi, S., Kar, B.K., Chaudhuri, P.P., 1994. Theory and applications of cellular automata in cryptography. IEEE Trans. Comput. 43 (12), 1346–1357. https://doi.org/10.1109/12.338094.

Noshadian, S., Ebrahimzade, A., Kazemitabar, S.J., 2018. Optimizing chaos based image encryption. Multimedia Tools Appl. 77 (19), 25569–25590. https://doi.org/10.1007/s11042-018-5807-x.

Peyre, G., 2004. Gabriel Peyre Copyright in MATLAB.

Ping, P., Wu, J., Mao, Y., Xu, F., Fan, J., 2018. Design of image cipher using life-like cellular automata and chaotic map. Signal Process. 150, 233–247. https://doi.org/10.1016/j.sigpro.2018.04.018.

Shilane, P., Min, P., Kazhdan, M., Funkhouser, T., 2004. The Princeton Shape Benchmark. In: Proceedings Shape Modeling Applications, 2004, Genova, Italy, pp. 167–178. https://doi.org/10.1109/SMI.2004.1314504.

Tamimi, A., Abdalla, A., 2017. A variable circular-shift image-encryption algorithm. In: Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV '17), Las Vegas, NV, USA, pp. 33–37.

Tamimi, A., Abdalla, A., 2014. An audio shuffle-encryption algorithm. In: International Conference on Internet and Multimedia Technologies (ICIMT '14), San Francisco, CA, USA, in Proceedings of the World Congress on Engineering and Computer Science, vol. I, October 2014, pp. 409–412.

Tamimi, A., Abdalla, A., 2015. An image encryption algorithm with XOR and S-box. In: 19th International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV '15), Las Vegas, NV, USA, pp. 166–169.

Wen, C., Xudong, C., 2013. Optical image encryption based on multiple-region plaintext and phase retrieval in three-dimensional space. Opt. Lasers Eng. 51 (2), 128–133. https://doi.org/10.1016/j.optlaseng.2012.09.002.

Wu, J., Liao, X., Yang, B., 2018. Image encryption using 2D Hénon-Sine map and DNA approach. Signal Process. 153, 11–23. https://doi.org/10.1016/j.sigpro.2018.06.008.

Yahya, A., Abdalla, A., 2009. An AES-based encryption algorithm with shuffling. In: Arabnia, H.R., Daimi, K. (Eds.), 2009 International Conference on Security and Management (SAM '09), Las Vegas, NV, USA, in Security and Management. CSREA Press, pp. 113–116.

Yang, X., Zhang, H., 2016. Encryption of 3D point cloud object with deformed fringe. Advances in Optical Technologies, 9. https://doi.org/10.1155/2016/4601462.

Ye, G., Pan, C., Huang, X., Mei, Q., 2018. An efficient pixel-level chaotic image encryption algorithm. Nonlinear Dyn. 94 (1), 745–756. https://doi.org/10.1007/s11071-018-4391-y.

Yue W., Noonan J. P., Agaian S., 2011. NPCR and UACI Randomness Tests for Image Encryption. Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), April Edition, 2011.

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008

Mrs. Manal A. Mizher received her Master degree (MSc) from the University of Jordan in 2009, and her (BSc) in Computer Science from Al-Zaytoonah University of Jordan in 2005. She has a seven-years experience of teaching IT subjects and maintenance the local and global computer networks. She registered at University Kebangsaan Malaysia (UKM) in 2014 for (PhD) in Visual Informatics, specialising in encryption data based on cellular automata, and three dimensional visual data and objects. Her study interests are in the areas of image processing, also in wireless sensor networks and bandwidth utilization. She has many published papers in these fields.

Prof. Dr. Riza Sulaiman is a Senior Research Fellow in the Institute of Visual Informatics, and an academic staff in the Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia (UKM). He received his (BSc) (Eng.) from University of Sunderland, UK. And his (MSc) (Eng.) from University of Portsmouth, UK. And his (PhD) degree from University of Canterbury. He is with expertise in Human-computer Interaction, Networks, and in the orientation histogram, standard deviation over the entropy value and number of circles is used to determine the number of bins in each area. His research interests include: Visualises and CADCAM.

Dr. Ayman M. Abdalla is an associate professor of computer science with experience in research and teaching in the USA and Jordan in addition to working in software development in a company in the USA. He has been a member of the Faculty of Science and Information Technology at Al-Zaytoonah University of Jordan since 2001, where he held different positions including the founding Chair of the Department of Multimedia Systems. He received his Ph.D. in computer science from the University of Central Florida, FL, USA; and his Master's and Bachelor's degrees in computer science from Montclair State University, NJ, USA. His research interests are focused on image and video encryption, with some interest in parallel and distributed computing.

Dr. Manar A. Mizher received her MSc from the University of Jordan in 2010, her BSc in Computer Science from Al-Zaytoonah University of Jordan in 2005, and her Ph.D in 2018 form the Institute of Visual Informatics, , Universiti Kebangsaan Malaysia (UKM). She has a ten-years experience of teaching IT subjects and maintenance the local and global computer networks. Currently, she is a research assistant in a university. She is specialising in video summarisation and authentication, encryption, and networking. Her study interests are using machine/deep learning algorithms and IoT cyber-security.

Please cite this article as: M.-A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla et al., A simple flexible cryptosystem for meshed 3D objects and images, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.008