An Effective Encryption Algorithm for 3D Printing Model Based on Discrete Cosine Transform

Ngoc-Giao Pham†, Kwon-Seok Moon**, Suk-Hwan Lee***, Ki-Ryong Kwon****

ABSTRACT

In this paper, we present an effective encryption algorithm for 3D printing models in the frequency domain of discrete cosine transform to prevent illegal copying, access in the secured storage and transmission. Facet data of 3D printing model is extracted to construct a three by three matrix that is then transformed to the frequency domain of discrete cosine transform. The proposed algorithm is based on encrypting the DC coefficients of matrixes of facets in the frequency domain of discrete cosine transform in order to generate the encrypted 3D printing model. Experimental results verified that the proposed algorithm is very effective for 3D printing models. The entire 3D printing model is altered after the encryption process. The proposed algorithm provides a better method and more security than previous methods.

Key words: 3D Printing Data, 3D Printing Security, Selective Encryption, DCT.

1. INTRODUCTION

Recent years, three dimension (3D) printing is widely used in many areas of life as healthcare, industry, automotive and many sectors [1, 2]. Due to the fact that the benefits of 3D printing is enormous in all domain and the price of a 3D printer is not expensive so the individual can buy a 3D printer and download 3D models on Internet to print out real objects without any permission from the original providers. Moreover, some special models and anti-weapon models must be secured from un-authorized users. Thus 3D printing data should be encrypted before being stored and transmitted in order to ensure the access and to prevent illegal copying.

For meeting to issues above, we would like to propose an effective encryption algorithm in the frequency domain of discrete cosine transform for 3D printing models in this paper. The data format of 3D printing is the 3D triangle mesh. The main content of the proposed algorithm is to extract facets from 3D triangle mesh and three vertices of each facet is used to construct a three by three (3×3) dimensional matrix. The proposed algorithm is based on encrypting the DC coefficients of the constructed matrixes in the frequency domain of discrete cosine transformation (DCT) [3] in order...

† Corresponding Author: Ki-Ryong Kwon, Address: (608-737) 45 Yongso-ro, Nam-gu, Busan, Pukyong National University, Daeryon campus, #A12, No. 1315, TEL: +82-51-629-6257, FAX: +82-51-629-6230, E-mail : kkwon@pknu.ac.kr
Received date: Nov. 15, 2017, Revision date: Dec. 15, 2017, Approval date: Jan. 5, 2018
* Dept. of IT Convergence and Application Engineering, Pukyong National University (E-mail: ngocgiacoet@gmail.com)
** Dept. of Electronics Engineering, Pukyong National University (E-mail: ksmoon@pknu.ac.kr)

†† Dept. of Information Security, Tongmyong University (E-mail: skylee@tu.ac.kr)
††† Dept. of IT Convergence and Application Engineering, Pukyong National University
* This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2016R1D1A3B0391003 and NRF-2017R1A2B2012456), Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00225), and also supported by Brain Busan (BB21) project.
to generate the encrypted 3D triangle mesh. To clarify the proposed algorithm, we organize our paper as follow. In Sec. 2, we look into previous encryption techniques for 3D models and explain the relation of 3D triangle mesh to the proposed algorithm. In Sec. 3, we show the proposed algorithm in detail. Experimental results and the evaluation of proposed algorithm will be shown in Sec. 4. Sec. 5 shows the conclusion.

2. RELATED WORKS

2.1 3D Model Security

In fact, the watermarking schemes for 3D content, 3D model [4, 5] are not suitable for the secured storage and transmission. Because the watermarking techniques do not alter the content of 3D printing models. They only embeds watermark data into 3D printing models, and anybody can see the content of 3D printing models and design them again for printing. So, they could not be applied to the secured storage and transmission for 3D printing models.

There are some proposed techniques for secret sharing the content of 3D models or 3D CAD model encryption. E. Marc et al. [6] proposed a method to encrypt 3D objects based on geometry preserving. This algorithm introduces a geometry preserving protection paradigm that heavily distorts 3D objects while preserving some intrinsic geometrical property (e.g. the bounding box or the convex hull), thereby avoiding a global corruption of the whole 3D scene. However, the key idea of this method only permute some facets of a 3D object. It did not alter the entire content of a 3D object and it is not effective to the various formats of 3D printing data. Moreover, reconstruction cannot fully restore the encrypted 3D objects and the security of this method is very low. X.T. Cai et al. [7 - 9] proposed an encryption approach for CAD models, which is based on geometric transformation encryption mechanisms of features of CAD models. The key content of this approach is centered on an Enhanced Encryption Transformation Matrix, which is characterized parametric, randomized and self-adaptive for feature encryption. This method only changes a little the shape of 3D CAD models. Consequently, the previous proposed methods cannot respond to the secured storage and transmission for 3D printing model.

2.2 3D Triangle Mesh Based Encryption

Currently, 3D printing technology often uses 3D triangle mesh [10, 11] to print real objects. A 3D triangle mesh is a set of facets. Each facet contains three vertices (a triangle) and a normal vector (see Fig. 1). Each vertex is presented by three coordinates x, y and z. Triangles are connected together to form a 3D triangle mesh as shown in Fig. 1. Therefore, facets are the target of the encryption process. Due to the fact that the normal vector of a facet only describes the direction of a facet, it does not determine the shape of a 3D triangle mesh. So, in order to encrypt a 3D triangle mesh we only extract facets and encryption all 3D triangles of a

![Fig. 1, Structure of 3D Triangle Mesh.](image-url)
3D triangle mesh by the secret key.

3. THE PROPOSED ALGORITHM

3.1 Overview

The proposed algorithm is described in Fig. 2a. Facets are firstly extracted from 3D triangle mesh and three vertices of each facet is used to construct a 3×3 matrix as shown in Fig. 3. These matrices are then transformed to DCT domain. In DCT domain, DC coefficients are selected and encrypted by a secret key value. The secret value is generated by a hashing function with a user's key input. After the DC coefficients encryption process in DCT domain, DCT coefficients which include the encrypted DC coefficient and AC coefficients, are performed inverse DCT in order to change all coefficients one more time. Finally, the coefficients of the inverse DCT process will be re-arranged to generate the encrypted facet. The vertices re-arrangement process uses the index of 3×3 matrix that is constructed before. The encrypted 3D triangle mesh is a set of the encrypted facets.

3.2 Selective Encryption

As the mention above, a 3D triangle mesh contains a set of facets. Each facet includes three vertices. Each vertex is presented by x, y and z coordinates. We consider a 3D triangle mesh \( \mathbf{M} = (\mathbf{F}_i | i \in [1, |\mathbf{M}|]) \) with \(|\mathbf{M}|\) is the cardinalities of a 3D triangle mesh: \( \mathbf{F}_i = (v_{i1}, v_{i2}, v_{i3}) \) is indicated the \( i^{th} \) facet with three vertices \((v_{i1}, v_{i2}, v_{i3})\) and the normal vector \( \mathbf{n}_i = (n_{x}, n_{y}, n_{z}) \). Each vertex is presented by three coordinates x, y and z. Due to the fact that the normal vector of a facet does not determine the shape of 3D triangle mesh, we briefly consider the facet \( \mathbf{F}_i \) includes three vertices as Eq. (1):

\[
\mathbf{F}_i = \{v_{i1}, v_{i2}, v_{i3}|i \in [1, |\mathbf{M}|]\}
\]

Three vertices of the facet \( \mathbf{F}_i \) is used to construct the matrix \( \mathbf{A}_i \) as shown in Eq. (2). Given \( DCT(.) \), \( IDCT(.) \) are forward DCT function and inverse DCT function respectively; and \( EC(.) \) is the

\[ DCT(F_i) = A \\ IDCT(A) = F_i \]

Fig. 2. The proposed algorithm, (a) Encryption Process and (b) Decryption Process.
encryption function. The matrix $A_l$ is transformed to DCT domain by $DCT(A_l)$ as shown in Eq. (3) with $D_{A_l}$ is the matrix of DCT coefficients.

$$A_l = \begin{bmatrix} x_{l1} & x_{l2} & x_{l3} \\ y_{l1} & y_{l2} & y_{l3} \\ z_{l1} & z_{l2} & z_{l3} \end{bmatrix}$$

$$D_{A_l} = DCT(A_l) = \begin{bmatrix} dx_{l1} \\ dy_{l1} \\ dz_{l1} \\ dx_{l2} \\ dy_{l2} \\ dz_{l2} \\ dx_{l3} \\ dy_{l3} \\ dz_{l3} \end{bmatrix}$$

In DCT domain, the DC coefficient $dx_{l1}$ is encrypted by the key value $K$ as shown in Eq. (4). The key value $K$ is generated by the SHA-512 hashing algorithm [12] that use a user’s key input. The length of each key value is 512 bits. After DC coefficient encryption, we perform inverse DCT to change all DCT coefficients one more time in order to generate the encrypted vertices $D_{A_l}$ as shown in Eq. (5).

$$dx'_{l1} = EC(K, dx_{l1}) = \frac{1}{K} \times dx_{l1}$$

$$D_{A_l} = IDCT(D_{A_l}) = IDCT \left( \begin{bmatrix} dx'_{l1} \\ dy'_{l1} \\ dz'_{l1} \\ dx'_{l2} \\ dy'_{l2} \\ dz'_{l2} \\ dx'_{l3} \\ dy'_{l3} \\ dz'_{l3} \end{bmatrix} \right)$$

$$E_{Fi} = \{e_{l1}, e_{l2}, e_{l3} | i \in [1, |M|]\}$$

$$E_M = \{E_{Fi} | i \in [1, |M|]\}$$

Due to the fact that DC coefficients are changed, after the inverse DCT process all DCT coefficients will be changed one more time. Finally, the encrypted vertices are re-arranged in order to obtain the encrypted facet $E_F$, as shown in Eq. (6). The encrypted facet $E_F$, includes three encrypted vertices $e_{l1}, e_{l2}$ and $e_{l3}$. The encrypted 3D triangle mesh $E_M$ is a set of the encrypted facets as shown in Eq. (7). Fig. 3 show the encryption process in DCT domain of a 3D triangle mesh.

3.3 Decryption Process

Fig. 2b shows the decryption process. The decryption process is an inverse process with the encryption process. The encrypted facet is also extracted from the encrypted 3D triangle mesh and three vertices of each encrypted facet is also used to construct a 3x3 matrix. These matrices are then transformed to DCT domain. In DCT domain, DC coefficients are decrypted by a secret key and all DCT coefficients will be then performed inverse DCT in order to restore the decrypted facet. Finally, the decrypted facets are re-arranged to generate the decrypted 3D triangle mesh.

4. EXPERIMENTAL RESULTS & ANALYSIS

We experimented the proposed algorithm with 3D triangle meshes as shown in Table 1. The format of 3D triangle meshes is STL file, VRML file [10, 11]. The detailed information of models is shown in Tab.1. In order to evaluate the proposed algorithm, we evaluate visualization experiments, the security and computation time of the proposed algorithm. Sec. 4.1 shows visualization experiments. Sec. 4.2 shows the security evaluation and the computation of the proposed algorithm is shown in Sec. 4.3.

4.1 Visualization Experiments

Experimental results are shown in Fig. 4. The
number of 3D triangle meshes is different. Facets are connected together in order to form a 3D shape. After the encryption process, facets are distorted into small facets (see "Encrypted Tower", "Encrypted Knob" and "Encrypted Boat" models) or big facets (see "Encrypted Pawn" and "Encrypted Quad Maxim"), changed location, positioned disorderly and not connected together. This lead to the shape of 3D triangle meshes is changed. Consequently, the content of 3D triangle meshes is completely altered after the selective encryption process. Pirates or un-authorized users cannot extract or view the content of 3D triangle meshes.

4.2 Security Evaluation

To evaluate the security of the proposed method, we will analyze the entropy of the encrypted 3D triangle mesh. If the entropy is high, the security will be high. From the equations in Sec. 3, we can see that the entropy of the encrypted 3D triangle mesh is dependent on both the secret key $K$ and the number of facets $|M|$. But $K$ and $|M|$ are ran-
dom independent variables. So the entropy of the encrypted 3D triangle mesh $H_M$ is the sum of the entropies of variables $K$ and $M$, and determined by Eq. (8).

$$H_M = H(K) + H(G) + H([M]) = |K| \log_2 |K| + |M| \log_2 |M|$$

(8)

Assume that the secret key is fixed, we can calculate the entropy of the encrypted 3D triangle mesh according to the number of facets $|M|$ as shown in Table 1. The entropy of the encrypted 3D triangle mesh is formed from $5281$ dB to $1.21 \times 10^6$ dB with $|M| \in [576, 74830]$. From Eq. (8) and Table 1 we can see that if $|M|$ is high, the entropy will be high.

In Marc’s method [6], he used the secret key $K$ to encrypt and change the location of the vertices of 3D triangle mesh in OXYZ space. Simply, we can understand that Marc’s method encrypted the vertices of 3D triangle mesh by a secret key $K$. But the number of vertices in a 3D triangle mesh is always smaller than the number of facets. Thus the entropy of this method is always lower than the proposed method. With test models in Table 1, the entropy of Marc’s method is formed from $1.456$ dB to $364330$ dB (see Table 1). In Cai’s method [9], he encrypt the features of 3D CAD model by a random $3 \times 3$ matrix that is generated as a secret key. Thus, we can consider that Cai’s method is encrypted 3D CAD models based on features and a random matrix by a secret key $K$. So, the entropy of this method is dependent on both the number of features and the $3 \times 3$ matrix. In experimental results, around 50% of facets are selected as the feature of 3D CAD model. With test models in Table 1, the entropy of Cai’s method is formed from $2381$ dB to $568412$ dB. Fig. 5 show the entropy of the proposed method with the entropy of previous methods according to the number of facets. The entropy of the proposed method is always higher than the entropy of previous methods. Consequently, the proposed method is better and more security than previous methods.

4.3 Computation Time

In our experiments, we used an Intel Core i7 Quad 3.5-GHz, 8 GB of RAM, Windows 7 64-bits, and C++ on Visual Studio 2013. The computation time of the proposed method is dependent on the number of facets. With test models in Table 1, the computation time is formed from $22.3$ ms to $17332$ ms with $|M| \in [576, 74830]$. From Table 1 we can conclude that if the number of groups and the number of facets is small, the computation time is

<table>
<thead>
<tr>
<th>Name</th>
<th># Facets</th>
<th>Proposed Method</th>
<th>Entropy (dB)</th>
<th>Computation Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tower</td>
<td>576</td>
<td>5281</td>
<td>1456</td>
<td>2381</td>
</tr>
<tr>
<td>Knob</td>
<td>850</td>
<td>8271</td>
<td>2307</td>
<td>3739</td>
</tr>
<tr>
<td>Boat</td>
<td>878</td>
<td>8335</td>
<td>2336</td>
<td>3882</td>
</tr>
<tr>
<td>Pawn</td>
<td>7316</td>
<td>93914</td>
<td>27438</td>
<td>43327</td>
</tr>
<tr>
<td>Quad Maxim</td>
<td>32982</td>
<td>495039</td>
<td>147588</td>
<td>231057</td>
</tr>
<tr>
<td>Sheep</td>
<td>74830</td>
<td>1211597</td>
<td>364330</td>
<td>568412</td>
</tr>
</tbody>
</table>
small and otherwise. In Marc's method, he did not show the computation time, so we could not compare Marc's methods with our method. In Cai's method, he only analysis the complexity time. The computation time of Cai's method is dependent on the time of valid check CAD model, time of feature encryption and time of CAD model encryption. He concluded that the enough to meet user's requirements. With the dependent on three processes in Cai's method, we consider and evaluate that the computation time of Cai's method is greater at least two the computation time of our method. Comparing to Cai's method, our method is faster than Cai's method. Fig. 6 show the computation time of the proposed method and Cai's method according to the number of facets.

5. CONCLUSION

In this paper, we proposed an effective encryption algorithm for 3D printing models in DCT domain. Three vertices of facet in 3D printing model is used to construct a 3\times3 matrix. The proposed algorithm is based on transforming matrixes to DCT domain and encrypting selectivity the DC coefficients of matrixes in DCT domain by the secret key in order to generate the encrypted 3D printing model. The proposed algorithm is very more effective than previous methods. It is also responsive to the various formats of 3D printing model. It provides a better solution and is more security than the previous proposed methods. In future, we improve the proposed algorithm and apply it to the secured storage and transmission systems.

REFERENCE


Ngoc-Giao Pham received the Degree of Engineering in School of Electronic & Telecommunication in Hanoi University of Science & Technology (HUST) in 2011, and Master degree from Pukyong National University (PKNU), Busan, South Korea in 2014. Currently, he is Ph.D candidate in Pukyong National University. His research interests include digital image processing & application, GIS visualization, multimedia data security, smart system and IoT.

Kwang-Seok Moon received the B.S., M.S., and Ph.D degrees in Electronics Engineering in Kyungpook National University, Korea in 1979, 1981, and 1989 respectively. He is currently a professor in department of Electronic Engineering at Pukyong National University. His research interests include digital image processing, video watermarking, and multimedia communication.

Suk-Hwan Lee received a B.S., a M.S., and a Ph.D Degrees in Electrical Engineering from Kyungpook National University, Korea in 1999, 2001, and 2004 respectively. He is currently an associate professor in Department of Information Security at Tongmyong University. His research interests include multimedia security, digital image processing, and computer graphics.

Ki-Ryong Kwon received the B.S., M.S., and Ph.D degrees in electronics engineering from Kyungpook National University in 1986, 1990, and 1994 respectively. He worked at Hyundai Motor Company from 1986-1988 and at Pusan University of Foreign Language from 1996-2006. He is currently a professor in Dept. of IT Convergence & Application Engineering at the Pukyong National University. He has researched University of Minnesota in USA on 2000-2002 with Post-Doc, and Colorado State University on 2011-2012 with visiting professor. He was the General President of Korea Multimedia Society from 2015-2016. He is also a director of IEEE R10 Changwon section. His research interests are in the area of digital image processing, multimedia security and watermarking, bioinformatics, weather radar information processing.