Edges of Interpolating Tetrahedron Based Encryption Algorithm for 3D Printing Model

Giao N. Pham[1,2], Son T. Ngo[1], Anh N. Bui[1], Ban Q. Tran[1], Dinh V. Tran[3], and Suk-Hwan Lee[4]
[1] Dept. of Computing Fundamentals, FPT University, Hanoi, Vietnam
[2] Advanced Analytics Center, FPT Software Co., Ltd., Hanoi, Vietnam
[3] Dept. of Computer, University of Freiburg, Freiburg, Germany
[4] Dept. of Information Security, Tongmyong University, Busan, South Korea
Email: {giaopn, sonnt69, anhbn5, bantq3}@fe.edu.vn, Dinh@informatik.uni-freiburg.de, sklee@tu.ac.kr

Abstract² With the increase of 3D printing applications in many areas of life, a large amount of 3D printing models is attacked and stolen by hackers. Moreover, some special models and anti-weapon models in 3D printing must be secured from un-authorized users. Therefore, 3D printing models should be encrypted before being stored and transmitted in order to prevent illegal copying. This paper presents an encryption algorithm for 3D printing models based on the edges of the interpolating tetrahedron. The proposed algorithm is based on encrypting the normal vector of facet and the edges of the interpolating tetrahedron by a secret key after the tetrahedron interpolation process. Each facet of 3D printing model is extracted to interpolate a tetrahedron, and the edges of the interpolating tetrahedron are then encrypted by a secret key. The encrypted edges of the interpolating tetrahedron and the encrypted normal vector are then used 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 also provide a better method and more security than previous methods.

Index Terms² 3D printing security, 3D triangle mesh, tetrahedron, encryption and cryptography

I. Introduction

Recent years, Three Dimension (3D) printing, also known as additive manufacturing is a process of making 3D solid objects from a digital file and widely used in many areas of life [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 user can buy a 3D printer and download 3D printing models from Internet to print out real objects without obtaining any permission from the original providers. Moreover, some special models and anti-weapon models must be secured from un-authorized users. Thus 3D printing models should be encrypted before being stored and transmitted in order to ensure the access and to prevent illegal copying. The purpose of encryption is to make pirates or un-authorized users cannot attack and view the shape or content of 3D printing models, so the encryption techniques must change the entire shape of 3D printing models after the encryption process. Moreover, 3D printing uses some different formats, thus the encryption techniques must be responsive to the various formats of 3D printing model.

For meeting to above requirements, we would like to propose an encryption algorithm for 3D printing models in this paper. The data format of 3D printing is the 3D triangle mesh. Facet is the main component of a 3D triangle mesh. The facets of a 3D tri-angle mesh will be extracted to interpolate tetrahedrons, and the edges of the interpolating tetrahedrons are then encrypted by a secret key. The encrypted edges of the interpolating tetrahedrons are then used to obtain the encrypted 3D printing model. 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 the proposed algorithm will be shown in Sec. 4. Sec. 5 shows the conclusion.

II. Related Work

A. 3D Model Encryption

There are some proposed techniques for secret sharing the content of 3D models or 3D CAD model encryption. E. Esam et al. [3] proposed two secret sharing approaches for 3D models using Blakely, Thien and Lin schemes. The 3D models are separated into parts before sharing and users then reconstruct 3D model from the shared part of that model. Actually, this method is only a secret sharing technique for the secured transmission. It is not an encryption method. E. Maret et al. [4] proposed a method to encrypt 3D objects based on geometry-preserving. This algorithm presented a geometry-preserving paradigm that heavily distorts 3D objects while preserving some intrinsic geometrical property. The key idea of this method only permute some facets of a 3D object. It did not alter the entire shape of a 3D object and it is not effective to the various formats of 3D printing models. Moreover, the reconstruction cannot fully restore the original object from the encrypted 3D objects and the security of this method is very low. Cai et al. [5]-[7] proposed an encryption approach for CAD models, which is based on geometric transformation encryption mechanisms on the features of CAD models. This approach encrypted 3D CAD models

Manuscript received November 25, 2019; revised April 7, 2020.

doi: 10.12720/jait.11.2.84-90

based on an Enhanced Encryption Transformation Matrix, which is characterized parametric, randomized and self-adaptive for features encryption. This method only changes a little the shape of 3D CAD models. Consequently, the previous proposed methods cannot response to the secured storage and transmission for 3D printing models.

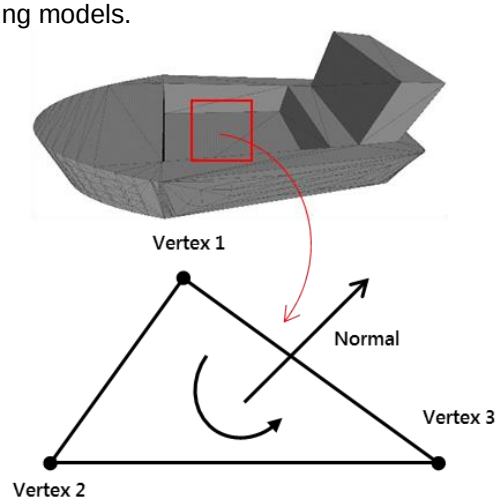

Figure 1. Structure of 3D triangle mesh.

B. 3D Triangle Mesh Based Encryption

Currently, 3D printing technology often uses 3D triangle meshes [8, 9] to print real objects. A 3D triangle mesh is a set of facets. Each facet contains three vertices (a triangle) and a nor-mal vector (see Fig. 1). Each vertex is presented by three coordinates x, y and z. Therefore, facets are the target of the encryption process. So, in order to encrypt a 3D triangle mesh we only extract facets and encryption them by the secret key value to obtain the encrypted 3D triangle mesh.

III. The Proposed Algorithm

A. Overview

The proposed algorithm is described in Fig. 2. Facets are firstly extracted from 3D triangle mesh. Each facet is then used to interpolate the edges of a tetrahedron which is corresponding to that facet. The edges of the interpolating tetrahedron are encrypted by the secret key value. The secret key value $\mathbf{K}$ is generated by a hashing function with user's key input. The normal vector of facet is also encrypted by the secret key value $\mathbf{K}$. After the edges and normal vector encryption process, the encrypted edges and the encrypted normal vector are used to interpolate a new facet. This facet is the encrypted facet. Because it is computed from the encrypted edges and the encrypted normal vector, and this new facet is always different with the original. The encrypted 3D triangle mesh is a set of the encrypted facets.

B. Encryption Process

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. $\mathbf{F}_i = \{v_{i1}, v_{i2}, v_{i3}$ and $\mathbf{n}_i\}$ is indicated the i^{th} facet with three vertices $\{v_{i1}, v_{i2}, v_{i3}\}$ and the normal vector $\mathbf{n}_i = (nx_i, ny_i, nz_i)$. To brief, we define the main notation as the following: $\mathbf{D}_i = \{d_{i1}, d_{i2}, d_{i3} | i \in [1, |\mathbf{M}|]\}$ is the edges of the interpolating tetrahedron that is corresponding to $\mathbf{F}_i$; $\mathbf{E}_{\mathbf{D}_i} = \{ed'_{i1}, ed'_{i2}, ed'_{i3} | i \in [1, |\mathbf{M}|]\}$ is the encrypted edges; $\mathbf{E}_{\mathbf{n}_i} = (nx'_i, ny'_i, nz'_i)$ is the encrypted normal vector; $\mathbf{E}_i = \{e_{i1}, e_{i2}, e_{i3}, (nx'_i, ny'_i, nz'_i) | i \in [1, |\mathbf{M}|]\}$ is the encrypted facet. Finally, $G_E(.)$ and $G_n(.)$ are the edge encryption function, the normal vector encryption function respectively.

The normal vector $\mathbf{n}_i(nx_i, ny_i, nz_i)$ is a vector that has vertex is (nx_i, ny_i, nz_i) and the origin of vector is located on the plane of facet. So, we can consider (nx_i, ny_i, nz_i) as a vertex in 3D space and use it together three vertices $\{v_{i1}, v_{i2}, v_{i3}\}$ to construct a tetrahedron (see Fig. 3a). The edges from the vertex (nx_i, ny_i, nz_i) to three vertices $\{v_{i1}, v_{i2}, v_{i3}\}$ is the edges the interpolating tetrahedron $\mathbf{D}_i = \{d_{i1}, d_{i2}, d_{i3} | i \in [1, |\mathbf{M}|]\}$ and calculate as follow

$$\begin{aligned}\mathbf{D}_i &= (nx_i, ny_i, nz_i) - \{v_{i1}, v_{i2}, v_{i3}\} \\ &= \{(nx_i, ny_i, nz_i) - v_{i1}; (nx_i, ny_i, nz_i) \\ &\quad - v_{i2}; (nx_i, ny_i, nz_i) - v_{i3}\} \\ &= \{d_{i1}, d_{i2}, d_{i3} | i \in [1, |\mathbf{M}|]\}\end{aligned} \quad (1)$$

The edges the interpolating tetrahedron $\mathbf{D}_i$ are then encrypted by the secret key value $\mathbf{K}$. We can use the conventional encryption function as AES, DES or MD5 to encrypt the edges of the interpolating tetrahedron. Here, for simplicity we encrypted the edges of the interpolating tetrahedron by the edge encryption function $G_E(.)$ as shown in Eq. (2).

$$\begin{aligned}\mathbf{E}_{\mathbf{D}_i} &= G_E(\mathbf{D}_i, \mathbf{K}) \\ &= \{\frac{\mathbf{K}}{i+1} \times d_{i1}; \frac{\mathbf{K}}{i+2} \times d_{i2}; \frac{\mathbf{K}}{i+3} \times d_{i3}\} \\ &= \{ed'_{i1}, ed'_{i2}, ed'_{i3} | i \in [1, |\mathbf{M}|]\}\end{aligned} \quad (2)$$

The normal vector $\mathbf{n}_i(nx_i, ny_i, nz_i)$ is also encrypted by the secret key value $\mathbf{K}$. Similarly, we encrypted the normal vector $\mathbf{n}_i(nx_i, ny_i, nz_i)$ by the normal vector encryption function $G_n(.)$ as shown in Eq. (3):

$$\begin{aligned}\mathbf{E}_{\mathbf{n}_i} &= G_n(\mathbf{n}_i, \mathbf{K}) \\ &= \{\frac{i}{\mathbf{K}} \times nx_i; \frac{i}{\mathbf{K}} \times ny_i; \frac{i}{\mathbf{K}} \times nz_i\} \\ &= (nx'_i, ny'_i, nz'_i) | i \in [1, |\mathbf{M}|]\end{aligned} \quad (3)$$

After the encryption process, the encrypted edges $\mathbf{E}_{\mathbf{D}_i}$ and the encrypted $\mathbf{E}_{\mathbf{n}_i}$ are used to compute the encrypted facet $\mathbf{E}_i = \{e_{i1}, e_{i2}, e_{i3}, (nx'_i, ny'_i, nz'_i) | i \in [1, |\mathbf{M}|]\}$. This facet has the normal vector $\mathbf{E}_{\mathbf{n}_i}$ and three encrypted vertices $\{e_{i1}, e_{i2}, e_{i3}\}$ (see Fig. 3b). Three encrypted vertices $\{e_{i1}, e_{i2}, e_{i3}\}$ are calculated as shown in Eq. (4).

$$\begin{aligned}\{e_{i1}, e_{i2}, e_{i3}\} &= (nx'_i, ny'_i, nz'_i) - \mathbf{E}_{\mathbf{D}_i} \\ &= \{(nx'_i, ny'_i, nz'_i) - ed'_{i1}; (nx'_i, ny'_i, nz'_i) \\ &\quad - ed'_{i2}; (nx'_i, ny'_i, nz'_i) - ed'_{i3}\}\end{aligned} \quad (4)$$

$$\mathbf{E}_M = \mathbf{E}_i\{|i \in [1, |\mathbf{M}|]\} \quad (5)$$

The encrypted 3D triangle mesh $\mathbf{E}_M$ is a set of the encrypted facets as shown in Eq. (5). Fig. 3 shows the encryption process for a facet of a 3D triangle mesh.

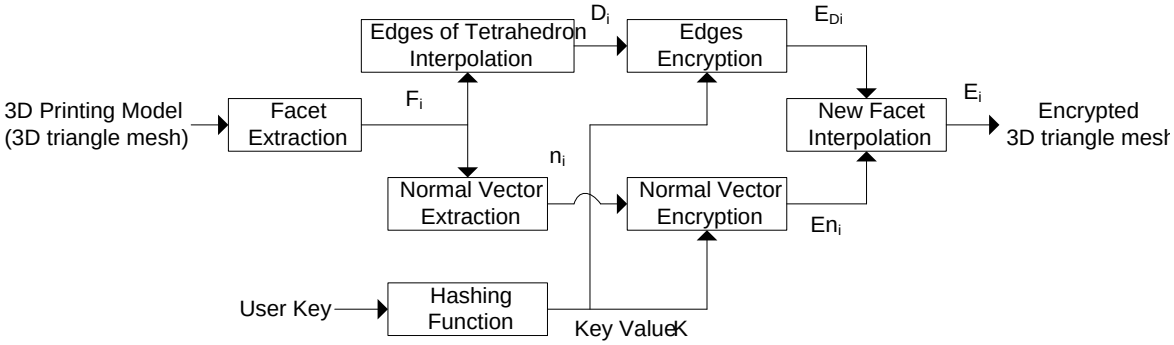

Figure 2. The proposed algorithm.

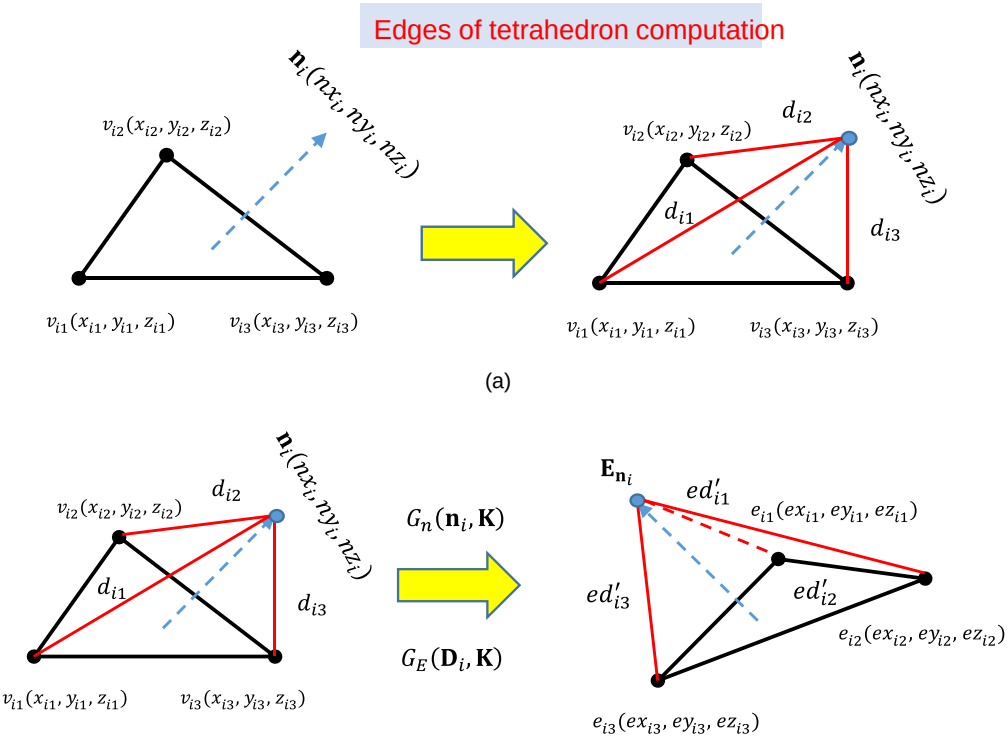

Figure 3. Encryption process for a facet, (a) edges of tetrahedron computation, and (b) edges encryption process.

C. Decryption Process

The decryption process is an inverse process with the encryption process. The encrypted facets are extracted from the encrypted 3D triangle mesh. With the encrypted normal vector $\mathbf{E}_{\mathbf{n}_i}$ and three encrypted vertices $\{e_{i1}, e_{i2}, e_{i3}\}$, we compute the encrypted edges $\mathbf{E}_{\mathbf{D}_i}$. The key value K is used to decrypt the encrypted normal vector $\mathbf{E}_{\mathbf{n}_i}$ and the encrypted edges $\mathbf{E}_{\mathbf{D}_i}$. From the decrypted edges, the decrypted normal vector and Eq. (1), we re-calculate the original facet. The decrypted 3D triangle mesh is a set of the decrypted facet.

IV. Experimental Result

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 [8][9]. The detailed information of test models is shown in Table 1. In order to evaluate the proposed algorithm, we evaluate visualization experiments, the security and computation time of the proposed algorithm. Sec. 4.A shows visualization experiments. Sec. 4.B shows the security evaluation and the computation time of the proposed algorithm is shown in Sec. 4.C.

A. Visualization Experiments

Experimental results are shown in Fig. 4. The number of facets in each model is different. After the encryption process, facets are distorted into small facets (see "Encrypted Mobile Case" and "Encrypted Knife") or big facets (see "Encrypted Gun"), changed location, positioned disorderly (see "Encrypted Wheel"). This leads to the shape of 3D triangle meshes is changed. Consequently, the content of 3D triangle meshes is completely altered after the encryption process. Pirates or un-authorized users cannot extract or view the content of 3D triangle meshes. In Cai's method [5]-[7], the encrypted CAD model is changed a little (see Fig. 5a). Anybody can see the content of the encrypted CAD model. We used the experimented model in Cai's method to experiment with the proposed algorithm. Experimental result is shown in Fig. 5b. We can see that the entire content of model is completely altered and any un-authorized user or pirate cannot view or extract the content of 3D printing model. Comparing with Cai's method, the perceptual results of the proposed method is better than Cai's method.

TABLE I. EXPERIMENTAL RESULTS

1 D P H	) D F H	(Q W U R S \				& R P S X W D W L R Q 7 L		
		Proposed Method	Giao's Method 2	Marc's Method	Cai's Method	Proposed Method	Giao's method 1	Giao's method 2
Gun	1878	20431	20469	10216	10245	36	50	45
Hospital Logo	6396	80872	80910	40441	40470	175	210	220
Mobile Case	6810	86722	86760	43366	43395	189	245	236
Knife	8662	113311	113349	55666	55695	1172	1347	1465
Snowman	25934	380266	380304	190133	190162	1940	2289	2425
Dog	31160	465146	465184	232573	232602	2781	2971	3477
House	108882	1821865	1821903	910933	910962	33427	36421	41783
Wheel	182808	3195486	3195524	1597243	1597272	43016	45243	53770

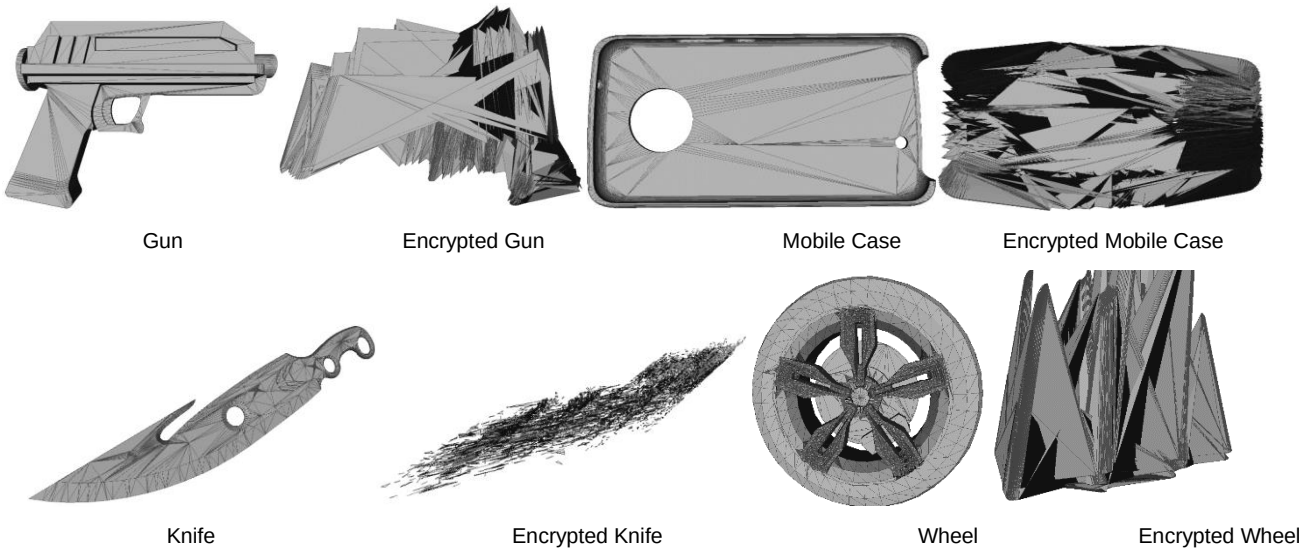

Figure 4. Experimental results with test models.

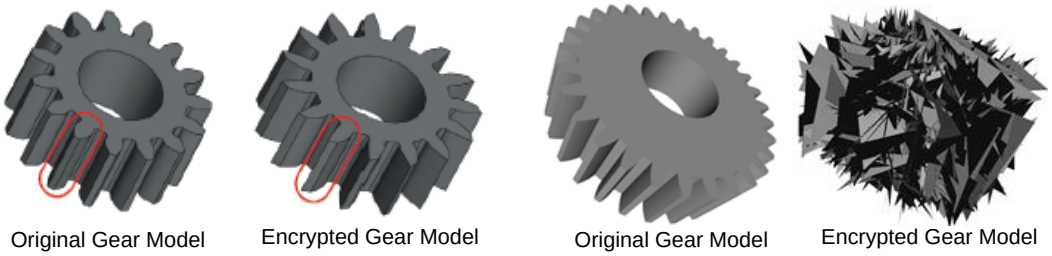

Figure 5. Comparison perceptual results between the proposed algorithm and Cai's method. (a) Results of Cai's method; (b) results of the proposed algorithm.

B. Security Evaluation

In order to decrypt the encrypted 3D triangle mesh, any pirate has to decrypt all the encrypted facets of 3D triangle mesh without knowledge of the keys. In our method, we used the SHA-512 algorithm with a 128 bits salt to generate random keys [10]. The length of the bits salt can be altered to 128, 256 or 512. Thus, if a user uses an English words of length l_k as his password, an attacker has to calculate $l_k \times 2^{128}$ keys to access the encrypted 3D triangle mesh. 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.

The entropy $H(x)$ of a discrete random variable x with a possible value $\{x_1, x_2, \dots, x_n\}$ is defined as

$$H(x) = -\sum_{\tau=1}^{n} p(x_\tau).log_2 p(x_\tau) \quad (6)$$

where $p(x_\tau)$ is the probability density function of on range $\{x_1, x_2, \dots, x_n\}$. From the equations in Sec. 3, we can see that the encrypted 3D triangle mesh is a set of th encrypted 3D triangles, thus the entropy of the encrypte 3D triangle mesh is dependent on both the secret K key and the number of facets $|\mathbf{M}|$. But K and $|\mathbf{M}|$ are random independent variables. So the entropy of the encrypted 3 triangle mesh $H_{\mathbf{M}}$ is the sum of the entropies of variables K and $|\mathbf{M}|$, and determined by Eq.)(7

$$\begin{aligned} H_{\mathbf{M}} &= H(\mathbf{K}) + H(|\mathbf{M}|) \\ &= |\mathbf{K}|.\log_2 |\mathbf{K}|) + |\mathbf{M}|.\log_2 |\mathbf{M}|) \end{aligned} \quad (7)$$

Summary, the entropy of the encrypted 3D triangle mesh is dependent on K and the number of facets $|\mathbf{M}|$. Assume that the secret key is fixed, we can calculate the entropy of the encrypted 3D triangle mesh according to th number of facets $|\mathbf{M}|$ as shown in Table I. The entropy of the encrypted 3D triangle mesh is formed from 20431 dB to 3.19x10^6 dB with $|\mathbf{M}| \in [1878, 182808]$. From Eq. 7) and Table I we can see that if $|\mathbf{M}|$ is high, the entropy will be high.

In Marc's method [4], 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 Table I, the entropy of Marc's method is formed from 10216 dB to 1.59x10^5 dB (see Table I). In Cai's method [7], he encrypted the features of 3D CAD model by a random 3x3 matrix that is generated by a secret key. Thus, we can consider Cai's method 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 3x3 matrix. In experimental results, around 50% of facets are selected as the feature of 3D CAD model. With test models in Table I, the entropy of Cai's method is formed from 10245 dB to 1.597x10^5 dB. In Giao et al. [11] (Giao's method 1), the entropy of this method is dependent on both the number of clusters, the secret key K, and the number of facets $|\mathbf{M}|$. Thus, the entropy of this method is also higher than the proposed method. In Giao et al. [12] (Giao's method 2), beside the distortion process to encrypt the 3D triangle mesh, we also encrypted the interpolating vector, curvature coefficients and control points of the interpolating curve from facet. Thus the entropy of this method is also dependent on the number of the interpolating vector, control points, and curvature coefficients. This leads to the entropy of Giao's method 2 is always higher than the proposed method. In Giao et al. [13] (Giao's method 3), the entropy of this method is only dependent on the secret key K and the number of facets $|\mathbf{M}|$, thus the entropy of this method is equal with the entropy of the proposed method. Fig. 6 shows the entropy of the proposed method with the entropy of previous methods (Cai's method and Marc's method) 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.

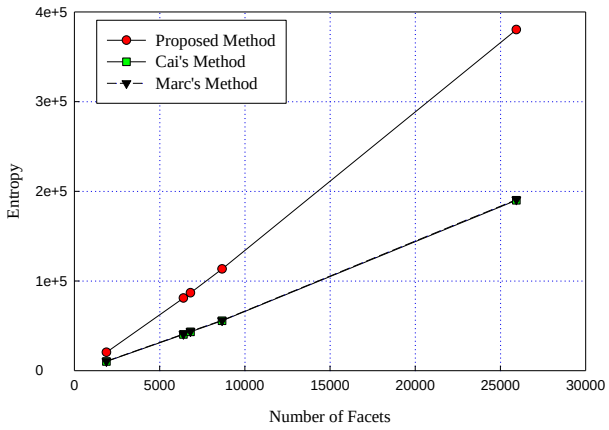

Figure 6. Entropy of the proposed method according to the number of facets.

C. 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 I, the computation time is formed from 36ms to 43016ms with $|\mathbf{M}| \in [1878, 182808]$. From Table I we can conclude that if the number of facets is small, the computation time is 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 process 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. Previously, we (Giao et al. [11], Giao et al. [12] and Giao et al. [13]) also proposed some encryption methods for 3D printing models. Giao's

method 1and Giao's method 3 are selective encryption methods in the frequency of discrete cosine transformation (DCT). The computation time of these methods is dependent on the K-mean clustering algorithm [14] and the processing time in DCT domain. Thus, the computation time of these methods is more expensive than the proposed method. In Giao's method 2, there are many steps in the encryption process, thus the computation time of this method is also more expensive than the computation of the proposed method (see Table I). Fig. 7 shows the computation time of the proposed method, Cai's method, Giao's method 1, and Giao's method 2 according to the number of facets. The proposed method is faster than previous proposed methods.

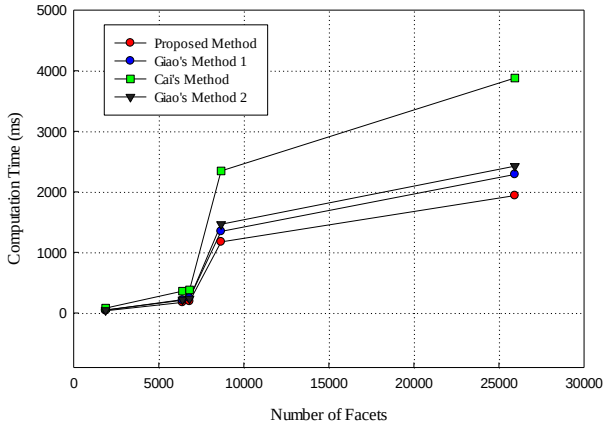

Figure 7. Computation time according to the number of facets.

V. Conclusion

In this paper, we proposed an encryption algorithm for 3D printing models. It is based on encrypting the edges of the interpolating tetrahedron by a secret key. The facet of 3D printing model is used to interpolate the tetrahedron, and the edges of the interpolating tetrahedron and the normal vector of facet are then encrypted by a secret key. The proposed algorithm is more effective than previous methods. It is also responsive to the various formats of 3D printing model because the proposed method only encrypted the geometric features of 3D printing model. It provides a better solution and is more security than previous proposed methods. It can be applied to the secured storage and transmission. Next time, we will improve and apply the proposed algorithm to some storage systems.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

All authors joined steps research; analysis; design; implement; evaluation and completion the entire paper.

Acknowledgment

This research was supported by the FPT Software Co., Ltd., Hanoi, Vietnam; FPT University, Hanoi, Vietnam; and also supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2016R1D1A3B03931003 and NRF-2017R1A2B2012456).

References

[1] 3D Printing Opportunities, Challenges, and Policy Implications of Additive Manufacturing, United States Government Accountability Office, USA, June 2015.
[2] White Paper: How 3D Printing Works, The Vision, Innovation and Technologies behind Inkjet 3D Printing, 3D Systems Circle Rock Hill, Jan. 2012.
[3] E. Esam and A. Ben, "Secret sharing approaches for 3D object encryption," Expert Systems with Applications, vol. 38, pp. 13906-13911, 2011.
[4] E. Marc, Y. Maetz, and D. Gwenaël, "Geometry-preserving Encryption for 3D Meshes," in Proc. Conference: Compression at Representation Signal Audio, Nov. 2013, pp. 7-12.
[5] X. T. Cai, F. Z. He, W. D. Li, X. X. Li, and Y. Q. Wu, "Encryption based partial sharing of CAD MODELS," Integrated Computer-Aided Engineering, vol. 22, pp. 243-260, 2015.
[6] X. T. Cai, W. D. Li, F. Z. He, and X. X. Li, "Customized encryption of computer aided design models for collaboration in cloud manufacturing environment," Journal of Manufacturing Science and Engineering, vol. 137, pp. 1-10, 2015.
[7] X. T. Cai, F. Z. He, W. D. Li, X. X. Li, and Y. Q. Wu, "Parametric and adaptive encryption of feature-based computer-aided design models for cloud-based collaboration," Integrated Computer-Aided Engineering, vol. 24, pp. 129-142, 2017.
[8] STL format in 3D printing. [Online]. Available: https://all3dp.com/what-is-stl-file-format-extension-3d-printing/
[9] VRML Format Document, The VRML Consortium Incorporated, 1997.
[10] Password-Based Cryptography Standard, RSA Lab., Oct. 2006.
[11] P. N. Giao, S. H. Lee, E. J. Lee, and K.R. Kwon, "Selective encryption algorithm for 3D Printing model based on clustering and DCT domain," Journal of Computing Science and Engineering, vol. 11, no. 4, pp. 152-159, 2017.
[12] N. P. Giao, S. H. Lee, and K. R. Kwon, "Interpolating spline curve-based perceptual encryption for 3D printing models," Applied Sciences, vol. 8, no. 2, p242, 2018.
[13] N. P. Giao, K. S. Moon, S. H. Lee, and K. R. Kwon, "An effective encryption algorithm for 3D printing model based on discrete cosine transform," Journal of Korea Multimedia Society, vol. 21, no. 1, pp. 61-68, 2018.
[14] J. M. Queen, "Some methods for classification and analysis of multivariate observations," in Proc. the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, 1967, pp. 281-297.

Giao N. Pham received the Degree of Engineering in School of Electronic & Telecommunication in Hanoi University of Science & Technology (HUST) in 2011, Master degree and PhD from Pukyong National University (PKNU), Busan, South Korea in 2014 and 2018 respectively. Currently, He is the IT lecturer at Dept. of Computing Fundamentals, FPT University and AI/Data scientist at FPT Software Co., Ltd. His research interests include digital image processing & application, GIS visualization, multimedia data security, multi system security, smart systems and IoT, machine learning/deep learning, data science and logistic.

Son T. Ngo received BSc in Computing (Field of study: Software Engineering), MSc in Computer Science (Field of Study: Environmental Decision Support System). His interests include Data Mining, Machine Learning, Computer Vision, Optimization, and Intelligent & Adaptive System. Currently, he is an IT Lecturer at FPT University, Hoalac Hightech-park, Thang Long Boulevard, Thach That District, Hanoi, Vietnam.

Anh N. Bui currently is an IT Lecturer at FPT University, Hoalac Hightech-park, Thang Long Boulevard, Thach That District, Hanoi, Vietnam. His interests include Big Data & Mining, Machine Learning, Image Processing, Robotics, and Mobile Computing.

Dinh V. Tran currently is Postdoc at Dept. of Computer Science, University of Freiburg, Freiburg, Germany University, Germany. His interests include Machine learning, Kernel Methods, Deep Learning, and Bioinformatics.

Ban Q. Tran currently works as an IT Lecturer at FPT University, Hoalac Hightech-park, Thang Long Boulevard, Thach That District, Hanoi, Vietnam. His research interests include domains in Software Engineering, Financial Technology, Machine Learning, and Big Data.

Suk-Hwan Lee received a B.S., a M.S., and a PhD Degrees in Electrical Engineering from Kyungpook National University, Korea in 1999, 2001, and 2004 respectively. He worked in Dept of Information Security in Tongmyong University, Busan, South Korea from 2005 to 2020. He is currently a professor in DongA University, Busan, South Korea. His research interests include multimedia security, digital image processing, and computer graphics.