3D Printing Model Random Encryption Based on Geometric Transformation

Ngoc-Giao Pham, Suk-Hwan Lee, Oh-Heum Kwon and Ki-Ryong Kwon

***Abstract*—Due to the fact that 3D printing has been recently applied in many areas of life, a large amount of 3D printing models have been attacked and stolen by hackers. Moreover, some special models and anti-weapon models used in 3D printing must be secured from unauthorized users. Therefore, 3D printing models must be encrypted before being stored and transmitted in order to prevent illegal copying. In this paper, we present a random encryption algorithm for 3D printing models. The proposed algorithm is based on randomly encrypting the vertices of each facet using a secret key after the geometric transformation process. Each facet of the 3D printing model is distorted by a geometric transformation and the three vertices of each distorted facet are then used to construct a 3×3 matrix. The coefficients of the constructed matrix are randomly encrypted using the random numbers of another matrix in order to generate the encrypted 3D printing model. The experimental results verify that the proposed algorithm is very effective for 3D printing models. The entire 3D triangle mesh is altered after the encryption process. The proposed algorithm is a better method and offers more security than the previously reported methods.**

***Index Terms*—3D printing data, 3D printing security, 3D triangle mesh, geometric transformation and randomization**

I. Introduction

Three-dimensional (3D) printing, also known as additive manufacturing, is a process of making 3D solid objects from a digital file, which is widely used in many areas of life [1], [2]. Due to the fact that the benefits of 3D printing are enormous in all domains and the price of a 3D printer is not expensive, users can buy a 3D printer and easily download 3D printing models from the Internet to print out physical 3D objects without any permission from the original providers. Moreover, some special 3D printing models and anti-3D weapon models must be secured from unauthorized users. Therefore, 3D printing models should be encrypted before being stored and transmitted in order to ensure access and to prevent illegal copying.

In fact, watermarking is not suitable for secured storage and transmission because these techniques do not alter the content of the 3D printing models. They only embed watermark data into the 3D printing models and anybody can see the content of the 3D printing data and design them again for 3D printing [3]-[5]. So, encryption techniques are essential to encrypt 3D printing models before storage and transmission.

To meet the above issues, we would like to propose a random encryption algorithm for 3D printing models in this paper. The data format of 3D printing models is 3D triangle mesh. Each facet of the 3D triangle mesh is distorted by the geometric transformation process and the three vertices of each facet are used to construct a 3×3 dimensional matrix. The coefficients of the constructed matrix were randomly encrypted by the random numbers of another 3×3 matrix to generate the encrypted 3D triangle mesh. To clarify the proposed algorithm, we organize our paper as follows: In Section II, we look into the previously reported encryption techniques used for 3D models and explain the relationship of the 3D triangle mesh to the proposed algorithm. In Section III, we describe the proposed algorithm in detail. The experimental results and evaluation of the proposed algorithm are shown in Section IV. Section V gives the conclusions.

II. Related Work

A. 3D Model Encryption

There are some techniques for 3D CAD model encryption reported in the literature. Marc *et al.* [6] proposed a method to encrypt 3D objects based on geometry-preserving. This algorithm introduces a geometry-preserving paradigm that heavily distorts 3D objects while preserving some intrinsic geometrical properties, thereby avoiding a global corruption of the whole 3D scene. The key idea of this method is to only permute some facets of a 3D object. It did not alter the entire shape of the 3D object and it is not effective in various formats of 3D printing models. Moreover, the reconstruction cannot fully restore the encrypted 3D objects and the security of this method is very low. Cai *et al.* [7]-[9] proposed an encryption approach for CAD models, which is based on geometric transformation encryption mechanisms on the 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 slightly changes the shape of the 3D CAD models. Consequently, the previously reported methods cannot respond to the secured storage and transmission of 3D printing models.

B. 3D Triangle Mesh Based Encryption

The input of 3D printing is a 3D triangle mesh [10], [11]. 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*. Thus, to encrypt a 3D triangle mesh, we only extract the facets and

Manuscript received January 17, 2018; revised March 8, 2018.

Ngoc-Giao Pham, Oh-Heum Kwon, and Ki-Ryong Kwon are with the Dept. of IT Convergence and Application Engineering, Pukyong National University, Busan, South Korea (Corresponding Author: Ki-Ryong Kwon; e-mail: ngocgiaofet@gmail.com, ohkwn@pknu.ac.kr, krkwon@pknu.ac.kr).

Suk-Hwan Lee is with Dept. of Information Security, Tongmyong University, Busan, South Korea (e-mail: skylee@tu.ac.kr).

doi: 10.18178/ijmlc.2018.8.2.685

encrypt all the facets using a secret key. However, 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, we only need to encrypt the triangles of the 3D triangle mesh to generate the encrypted 3D triangle mesh.

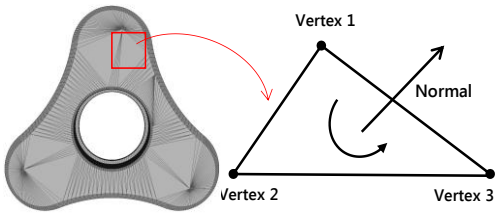


Fig. 1. The structure of 3D triangle mesh.

III. The Proposed Algorithm

A. Overview

The proposed algorithm is described in Fig. 2. The facets are initially extracted from the 3D triangle mesh. Each facet is then distorted by the shearing process that uses the shearing vector. The shearing vector is generated by the index of the facet and the secret key value K. The key value K is generated by a hashing function with user's key input. After the shearing process, the three vertices of the distorted facet are used to construct a 3×3 matrix. This matrix is then randomly encrypted with a matrix of random numbers by the matrix randomization process in order to obtain the encrypted facet. The matrix of random numbers is also generated by the key value **K** and the random distribution function. Finally, the encrypted facets are re-arranged to obtain the encrypted 3D triangle mesh. The encrypted 3D triangle mesh is a set of the encrypted facets.

B. Random Encryption

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 $\mathbf{n}_i\}$ indicates the i^{th} facet with three vertices$\{v_{i1}, v_{i2}, v_{i3}\}$ and the normal vector $\mathbf{n}_i(nx_i, ny_i, nz_i)$. 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):

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

In brief, we define the main notation as the following: $\mathbf{T}_i = \{t_{i1}, t_{i2}, t_{i3} | i \in [1, |\mathbf{M}|]\}$ is the shearing vector corresponding to $\mathbf{F}_i$, $\mathbf{F}_i^{'} = \{v_{i1}^{'}, v_{i2}^{'}, v_{i3}^{'} | i \in [1, |\mathbf{M}|]\}$ is the distorted facets after the shearing process, $\mathbf{A}_i$ is the matrix constructed from the three vertices of $\mathbf{F}_i^{'}$, **R** is the matrix of random numbers, which is used to randomize the matrix $\mathbf{A}_i$ and $\mathbf{E}_i = \{e_{i1}, e_{i2}, e_{i3} | i \in [1, |\mathbf{M}|]\}$ is the encrypted facet. Finally, $G_T(.), G_R(.), S_F(.)$ and $R_F(.)$ are the shearing vector generation function, the random numbers generation function, the facet shearing function and the matrix randomization function, respectively.

To distort the facet $\mathbf{F}_i$, we have to generate the shearing vector $\mathbf{T}_i$ using the index of facet $\mathbf{F}_i$ and the key value **K**, as shown in Eq. (2). The key value **K** is generated by the SHA-512 hashing algorithm [12] that use the user's key input. The length of each key value is 512 bits. The facet $\mathbf{F}_i$ is then distorted into $\mathbf{F}_i^{'}$ by the shearing function, as shown in Eq. (3).

$$\begin{aligned} \mathbf{T}_i &= G_T(\mathbf{K}, i) \\ &= \{\tfrac{i+1}{\mathbf{K}}, \tfrac{i+2}{\mathbf{K}}, \tfrac{i+3}{\mathbf{K}} | i \in [1, |\mathbf{M}|]\} \\ &= \{t_{i1}, t_{i2}, t_{i3} | i \in [1, |\mathbf{M}|]\} \end{aligned} \quad (2)$$

$$\begin{aligned} \mathbf{F}_i^{'} &= S_F(\mathbf{T}_i, \mathbf{F}_i) \\ &= \{(v_{ij} + 1) \times t_{ij} | i \in [1, |\mathbf{M}|], j \in [1,3]\} \\ &= \{v_{i1}^{'}, v_{i2}^{'}, v_{i3}^{'} | i \in [1, |\mathbf{M}|]\} \end{aligned} \quad (3)$$

$v_{i1}(x_{i1}^{'}, y_{i1}^{'}, z_{i1}^{'})$, $v_{i1}(x_{i2}^{'}, y_{i2}^{'}, z_{i2}^{'})$ and $v_{i1}(x_{i3}^{'}, y_{i3}^{'}, z_{i3}^{'})$ are the coordinates of the three vertices of the distorted facets $\mathbf{F}_i^{'}$. The coordinates of three vertices of the encrypted facet $\mathbf{F}_i^{'}$ were used to construct the 3×3 matrix $\mathbf{A}_i$, as shown in Eq. (4). This matrix is then randomly encrypted using the matrix of random numbers **R**. The matrix **R** is generated by the random number generation function that uses the key value **K** as the seed input, as shown in Eq. (5). The randomization of the matrix $\mathbf{A}_i$ is shown in Eq. (6).

$$\mathbf{A}_i = \begin{bmatrix} x_{i1}^{'} & x_{i2}^{'} & x_{i3}^{'} \\ y_{i1}^{'} & y_{i2}^{'} & y_{i3}^{'} \\ z_{i1}^{'} & z_{i2}^{'} & z_{i3}^{'} \end{bmatrix} \quad (4)$$

$$\begin{aligned} \mathbf{R} &= G_R(\mathbf{K}) \\ &= \begin{bmatrix} rx_1 & rx_2 & rx_3 \\ ry_1 & ry_2 & ry_3 \\ rz_1 & rz_2 & rz_3 \end{bmatrix} \end{aligned} \quad (5)$$

$$\begin{aligned} \mathbf{A}_i^{'} &= R_F(\mathbf{A}_i, \mathbf{R}) \\ &= \begin{bmatrix} x_{i1}^{'} \times rx_1 & x_{i2}^{'} \times rx_2 & x_{i3}^{'} \times rx_3 \\ y_{i1}^{'} \times ry_1 & y_{i2}^{'} \times ry_2 & y_{i3}^{'} \times ry_3 \\ z_{i1}^{'} \times rz_1 & z_{i2}^{'} \times rz_2 & z_{i3}^{'} \times rz_3 \end{bmatrix} \\ &= \begin{bmatrix} ex_{i1} & ex_{i2} & ex_{i3} \\ ey_{i1} & ey_{i2} & ey_{i3} \\ ez_{i1} & ez_{i2} & ez_{i3} \end{bmatrix} \end{aligned} \quad (6)$$

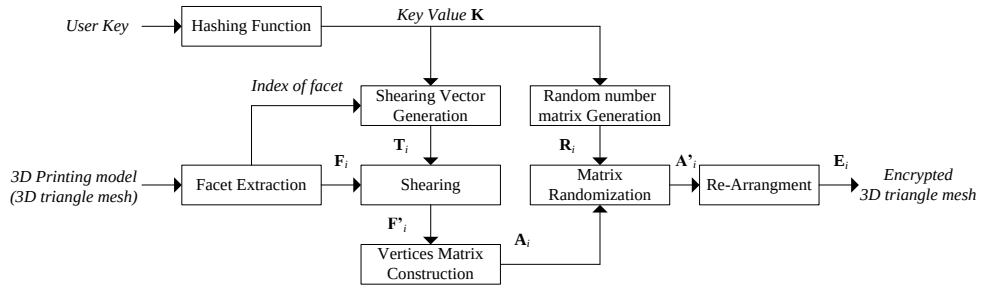


Fig. 2. The proposed algorithm.

After the matrix randomization process, all the coefficients of matrix $\mathbf{A}_i$ are altered into the encrypted matrix $\mathbf{A}_i^{'}$. Finally,

all the coefficients of matrix $\mathbf{A}_i^{'}$ will be re-arranged in order to obtain the encrypted facet $\mathbf{E}_{\mathbf{F}i}$, as shown in Eq. (7). The encrypted facet $\mathbf{E}_{\mathbf{F}i}$ includes three encrypted vertices $e_{i1}(ex_{i1}, ey_{i1}, ez_{i1}), e_{i2}(ex_{i2}, ey_{i2}, ez_{i2}), e_{i3}(ex_{i3}, ey_{i3}, ez_{i3})$. The encrypted 3D triangle mesh $\mathbf{E}_M$ is a set of the encrypted facets, as shown in Eq. (8). Fig. 3 shows the random encryption process for a facet of a 3D triangle mesh.

$$\mathbf{E}_i = \{e_{i1}, e_{i2}, e_{i3} | i \in [1, |\mathbf{M}|]\} \quad (7)$$

$$\mathbf{E}_M = \{\mathbf{E}_i | i \in [1, |\mathbf{M}|]\} \quad (8)$$

C. Decryption Process

The decryption process is the inverse process of the encryption process. The encrypted facet is also extracted from the encrypted 3D triangle mesh in order to construct a matrix and perform the re-randomization process. The matrix of random numbers is also generated by the random number generation function with the key seed value **K**. The shearing vector is also similarly generated using the key value **K** as in the encryption process. If, in the encryption process, the coefficients are randomized by multiplying the coefficients with random numbers, in the decryption process we only need to divide the encrypted coefficients for random numbers. The re-shearing process is also an inverse process with the shearing process, which is shown in Eq. (3). Finally, the decrypted facets are re-arranged to restore the original 3D triangle mesh.

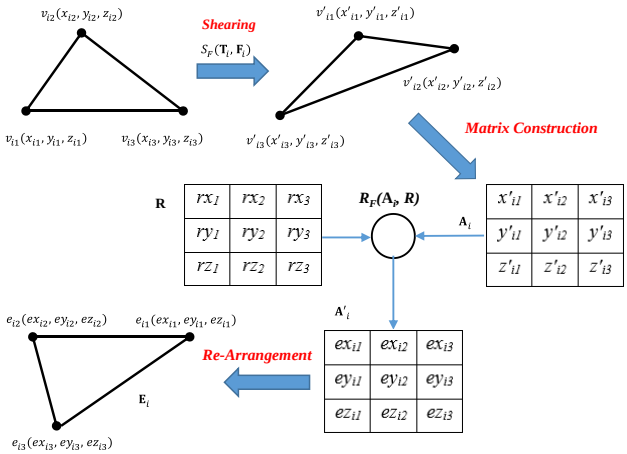


Fig. 3. The random encryption process used for each facet.

TABLE I: EXPERIMENTAL RESULTS

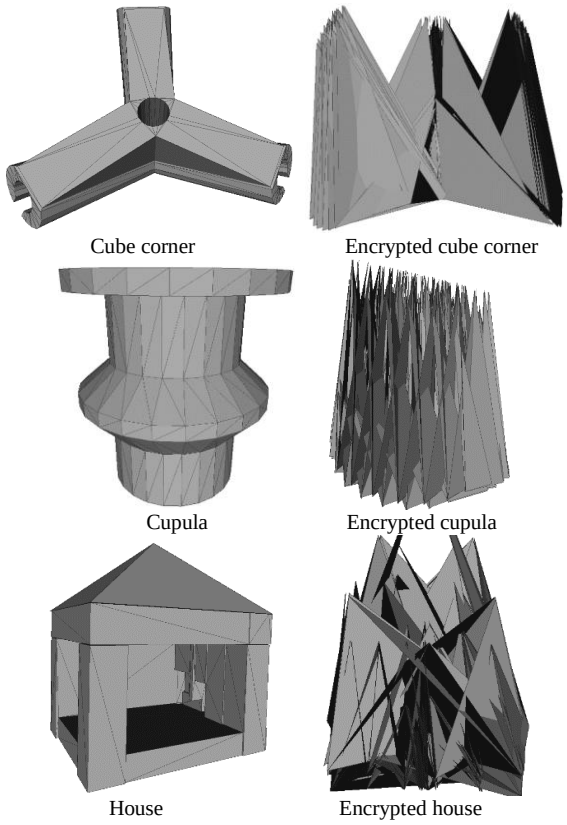
Name	# Facets	Entropy (dB)			Computation time (ms)
		Proposed method	Marc method	Cai method	
Cube Corner	548	5018	2237	2247	22.6
Cupula	678	6057	2869	2877	24.0
House	866	8483	3812	3821	36.3
Blade Holder	6524	82702	38098	38101	175.3
Pikachu	6870	87598	40374	40376	186.9
Horse	28662	424426	194879	197894	2329.0
Bear	48552	755853	353426	353663	6456.6
Shapy	289958	5261461	2485772	2485780	32318.6

IV. EXPERIMENTAL RESULTS

We implemented the proposed algorithm with 3D triangle meshes, as shown in Table I. The format of the 3D triangle meshes was a STL file and VRML file [10], [11]. Detailed information regarding the models is shown in Table I. In order to evaluate the proposed algorithm, we evaluated the visualization experiments, security, and computation time of the proposed algorithm. Section IV. A shows the visualization experiments. Section IV. B shows the security evaluation and the computation time of the proposed algorithm is shown in Section IV. C.

A. Visualization Experiments

The experimental results are shown in Fig. 4. The number of facets in each model is different. After the encryption process, the facets are distorted into small facets (see "Encrypted cupula" and "Encrypted blade holder") or big facets (see "Encrypted cube corner" and "Encrypted horse"), changed location and positioned disorderly (see "Encrypted bear" and so on"). This leads to the shape of the 3D triangle meshes being changed. Consequently, the content of the 3D triangle meshes was completely altered after the random encryption process. Pirates or unauthorized users cannot extract or view the content in the 3D triangle meshes. In Cai's method [7]-[9], the encrypted CAD model was slightly changed. Anybody can see the content of the encrypted CAD model. When compared with the Cai method, the perceptual results of the proposed method are improved.



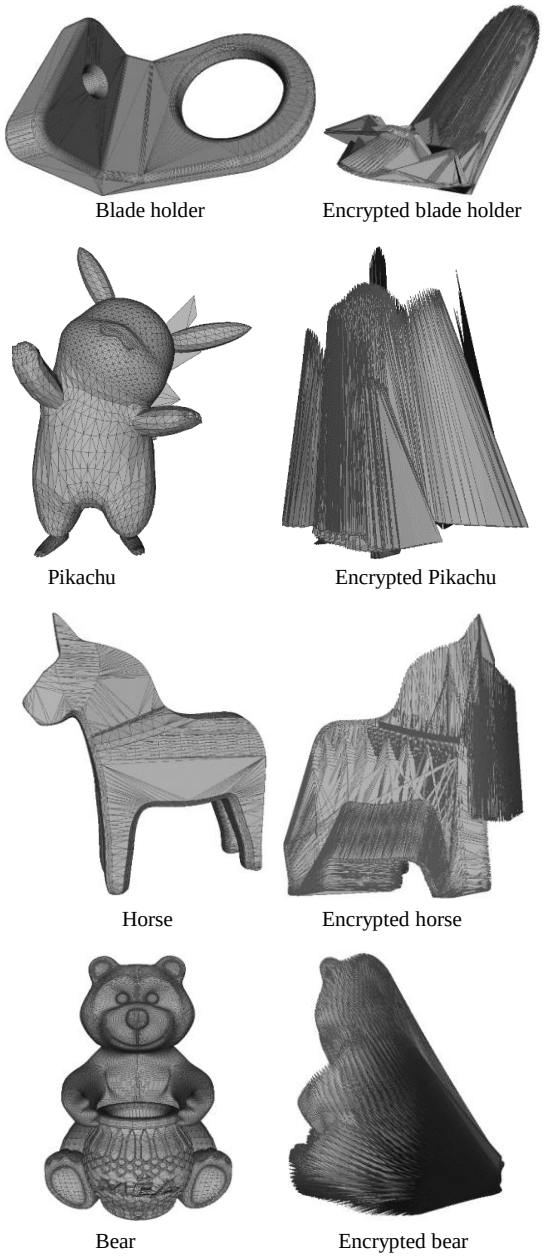


Fig. 4. The experimental results obtained for the test models.

B. 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 Section III, we can see that the entropy of the encrypted 3D triangle mesh is dependent on the secret key **K**, the shearing vector $\mathbf{T}_i$, the matrix of random numbers **R** and the number of facets |**M**|. However, **K**, $\mathbf{T}_i$, **R** and |**M**| are random independent variables. So the entropy of the encrypted 3D triangle mesh $H_{\mathbf{M}}$ is the sum of the entropies of the variables **K,** $\mathbf{T}_i$, **R** and |**M**|, and determined using Eq. (9).

$$\begin{aligned} H_{\mathbf{M}} &= H(\mathbf{K}) + H(\mathbf{R}) + H(\mathbf{T}_i) + H(|\mathbf{M}|) \\ &= |\mathbf{K}|.\log_2|\mathbf{K}|) + |\mathbf{M}|.\log_2|\mathbf{M}|) + |\mathbf{T}_i|.\log_2|\mathbf{T}_i| \\ &\quad + |\mathbf{R}|.\log_2|\mathbf{R}| \end{aligned} \tag{9}$$

However, |**R**| = 9 and $|\mathbf{T}_i|$ is dependent on |**M**|. Consequently, the entropy of the encrypted 3D triangle mesh is dependent on **K** and the parameters of the 3D triangle mesh. Assuming 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 I. The entropy of the encrypted 3D triangle mesh is formed from 5018 dB to 5.26 $\times$ 10^6 dB with $|\mathbf{M}| \in [576, 74830]$. Based on Eq. (9) and Table I 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 the 3D triangle mesh in OXYZ space. We can understand that Marc's method encrypted the vertices of 3D triangle mesh using secret key **K.** However, 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 the test models shown in Table I, the entropy of Marc's method is formed from 2237 dB to 2.48 $\times$ 10^6 dB (see Table I). In Cai's method [9], he encrypted the features of a 3D CAD model using a random 3 $\times$ 3 matrix that was generated from the a secret key. Thus, we can consider that Cai's method encrypted 3D CAD models are based on features and a random matrix using secret key **K**. So, the entropy of this method is dependent on both the number of features and the 3 $\times$ 3 matrix. In the experimental results of Cai's method, around 50% of the facets are selected as the feature of the 3D CAD model. With the test models shown in Table I, the entropy of Cai's method is formed from 2247 dB to 2.48 $\times$ 10^6 dB. Fig. 5 shows the entropy of the proposed method with the entropy of the previous methods according to the number of facets. The entropy of the proposed method is always higher than the entropy of the previous methods. Consequently, the proposed method is better and offers more security than the previous methods.

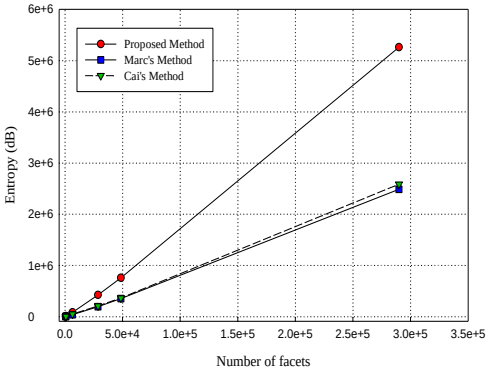


Fig. 5. The entropy of the proposed method according to the number of facets.

C. Computation Time

In our experiments, we used an Intel Core i7 Quad 3.5 GHz processor, 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 the test models shown in Table I, the computation time is observed from 22.6 ms to 32318 ms with $|\mathbf{M}| \in [548, 289958]$. From Table I, we 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 method with ours. In Cai's method, he only analyzed 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 it was sufficient enough to meet the user's requirements. With the dependence on three

processes in Cai's method, we considered and evaluated that the computation time of Cai's method is greater by at least two times the computation time of our method. When compared to the Cai method, our method is faster. Fig. 6 shows the computation time of the proposed method and Cai's method according to the number of facets.

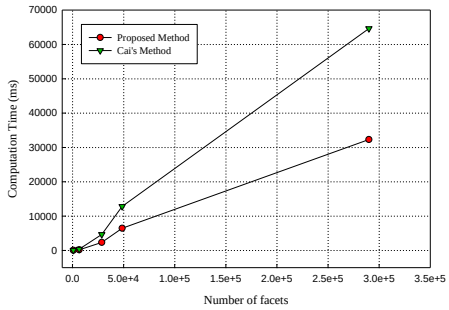


Fig. 6. The computation time according to the number of facets.

V. Conclusions

In this paper, we proposed a random encryption algorithm for 3D printing models. It is based on distorting the facets of the 3D printing model by geometric transformation and then encrypting the vertices of the distorted facets by randomizing the vertices of the distorted facet with the matrix of random numbers. The proposed method is more effective than the previously reported methods. It is also responsive to the various formats of 3D printing models. It provides a better solution and more security than the previously reported methods. It can be applied to secured storage and transmission.

Acknowledgments

This research was 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-NRF-2016R1D1A3B03931003 and NRF-2017R1A2B2012456), Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00225).

References

[1] United States Government Accountability Office, *3D Printing Opportunities, Challenges, and Policy Implications of Additive Manufacturing*, USA, June, 2015.
[2] 3D Systems Circle Rock Hill, "White Paper: How 3D Printing Works, The Vision, Innovation and Technologies behind Inkjet 3D Printing, Jan., 2012.
[3] J. U. Ho, D. G. Kim, S. H. Choi, and H. K. Lee, "3D print-scan resilient watermarking using a histogram-based circular shift coding structure," in *Proc. the 3rd ACM Workshop on Information Hiding and Multimedia Security*, 2015, pp. 115-121.
[4] S. Yamazaki, K. Satoshi, and M. Masaaki, "Extracting watermark from 3D Prints," in *Proc. the 22nd International Conference on Pattern Recognition*, 2014, pp. 4576-4581.
[5] M. Suzuki, S. Piyarat, U. Kazutake, U. Hiroshi, and Y. Takashima, "Copyright protection for 3D printing by embedding information inside real fabricated objects," in *Proc. the 10th International Conference on Computer Vision Theory and Applications*, 2015, pp. 180-185.
[6] E. Marc, Y. Maetz, and D. Gwenael, "Geometry-preserving Encryption for 3D Meshes," in *Proc. Conference: Compression at Representation Signal Audio*, 2013, pp. 7-12.
[7] 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.
[8] 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.
[9] 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.
[10] STL format in 3D printing. (2017). [Online]. Available: https://all3dp.com/what-is-stl-file-format-extension-3d-printing/
[11] The VRML Consortium Incorporated, "VRML format document," 1997.
[12] RSA Lab., Password-Based Cryptography Standard, Oct. 2006.

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

Suk-Hwan Lee received his B.S., M.S. and Ph.D. in electrical engineering from Kyungpook National University**,** Korea in 1999, 2001 and 2004, respectively. He is currently an Associate Professor in the Department of Information Security at Tongmyong University. His research interests include multimedia security, digital image processing, and computer graphics.

Oh-Heum Kwon received his B.S. degree in computer engineering from Seoul National University in 1988 and M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology in 1991 and 1996, respectively. He is currently a Professor in the Department of IT Convergence and Application Engineering at the Pukyong National University. His research interests include the design and analysis of algorithms, combinatorial optimization, mathematical optimization, wireless sensor networks, graph theory, computational geometry, location based service, sensor network localization, and wireless networks.

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