


Article

Two-Dimensional (2D) Slices Encryption-Based Security Solution for Three-Dimensional (3D) Printing Industry

Giao N. Pham ¹ , Suk-Hwan Lee ², Oh-Heum Kwon ¹ and Ki-Ryong Kwon ^{1,*}

¹ Department of IT Convergence & Application Engineering, Pukyong National University, Busan 608-737, Korea; ngocgiaofet@gmail.com (G.N.P.); ohkwn@pknu.ac.kr (O.-H.K.)

² Department of Information Security, Tongmyong University, Busan 608-711, Korea; skylee@tu.ac.kr

* Correspondence: krkwon@pknu.ac.kr; Tel.: +82-051-629-6257

Received: 22 April 2018; Accepted: 2 May 2018; Published: 7 May 2018



Abstract: Nowadays, three-dimensional (3D) printing technology is applied to many areas of life and changes the world based on the creation of complex structures and shapes that were not feasible in the past. But, the data of 3D printing is often attacked in the storage and transmission processes. Therefore, 3D printing must be ensured security in the manufacturing process, especially the data of 3D printing to prevent attacks from hackers. This paper presents a security solution for 3D printing based on two-dimensional (2D) slices encryption. The 2D slices of 3D printing data is encrypted in the frequency domain or in the spatial domain by the secret key to generate the encrypted data of 3D printing. We implemented the proposed solution in both the frequency domain based on the Discrete Cosine Transform and the spatial domain based on geometric transform. The entire 2D slices of 3D printing data is altered and secured after the encryption process. The proposed solution is responsive to the security requirements for the secured storage and transmission. Experimental results also verified that the proposed solution is effective to 3D printing data and is independent on the format of 3D printing models. When compared to the conventional works, the security and performance of the proposed solution is also better.

Keywords: 3D printing; 3D printing data; 3D printing security; discrete cosine transform; geometric transformation

1. Introduction

The three-dimensional printing (3D printing) also known as additive manufacturing, is a method of converting a virtual 3D model into a physical 3D object using additive processes [1–4]. In an additive process, a physical 3D object is created by laying down successive layers of material until the object is created. Thus, to print a physical 3D object from a 3D printing model by a 3D printer, the 3D printing model must be cut into a set of 2D slices from bottom to top [4]. This set of two-dimensional (2D) slices is stored in a file that is the input data of a 3D printer to print physical 3D objects, as described in Figure 1. Thus, these 2D slices are also considered as 3D printing data.

Since 3D printing is applied in a variety of industries including jewelry, footwear, industrial design, architecture, engineering and construction, automotive, aerospace, medical and healthcare industries, education, and consumer products [5–7], the data of 3D printing can be attacked by hackers [8–10]. As shown in Figure 2, attackers can attack on 3D printing models by attacking the database of 3D printing models or faking user to illegally copy or destroy 3D printing models when they are transmitted to user via the Internet (see Figure 2a). Besides, to prevent the duplication or the modification of 3D printing models by users, the original providers can cut 3D printing model into 2D

slices and then store them in a database. Therefore, hackers can also attack the 3D printing process by attacking on files that contain 2D slices when they are stored in a database or transmitted to users the Internet, as shown in Figure 2b.

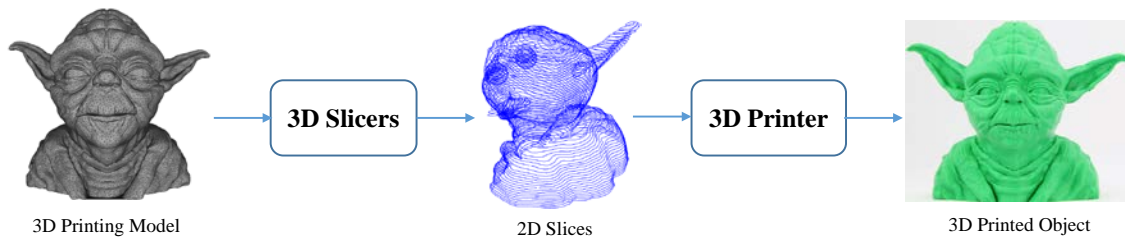


Figure 1. General 3D Printing Process.

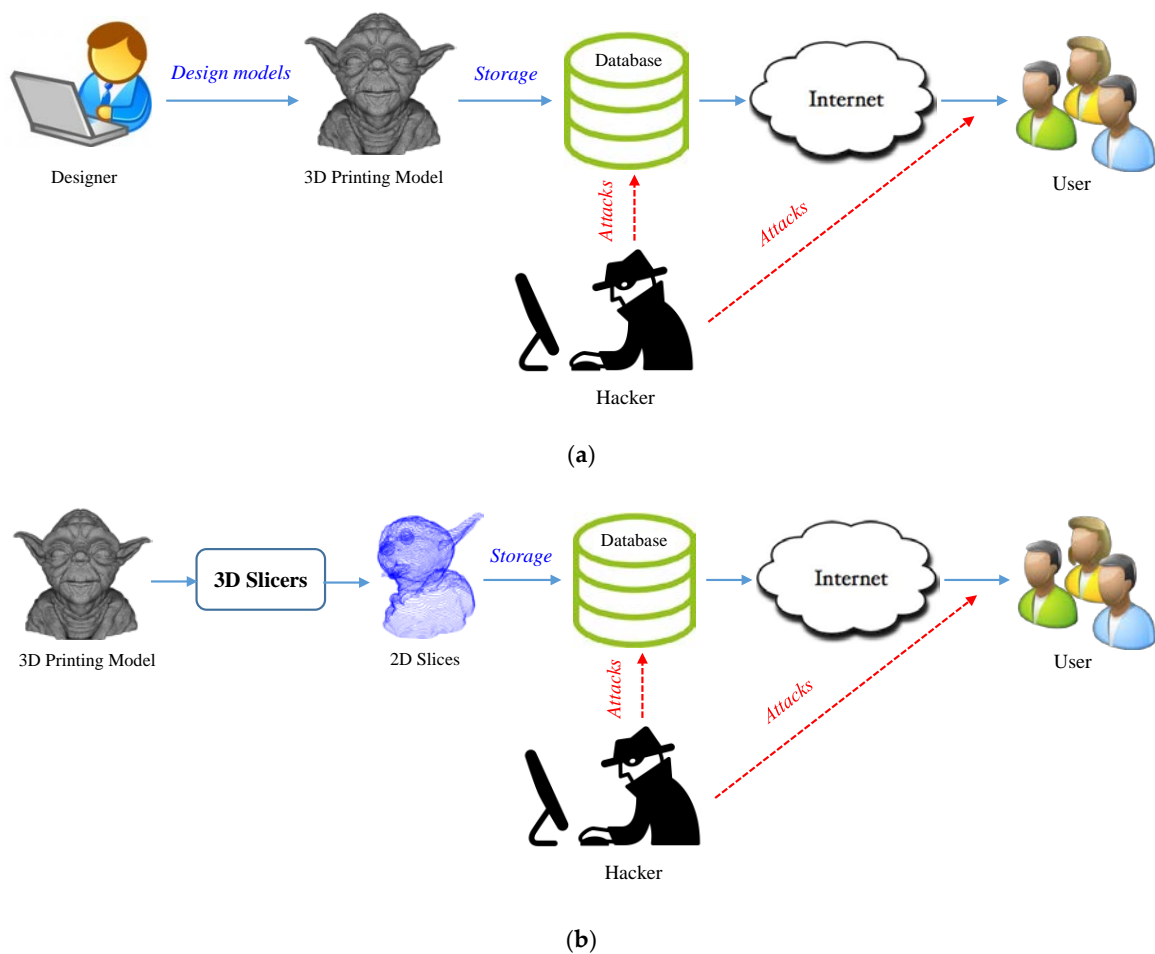


Figure 2. (a) Attack to three-dimensional (3D) printing model, and (b) attack to 3D printing data (two-dimensional (2D) Slice).

To prevent attacks to 3D printing model and the file of 2D slices, encryption solutions for 3D printing model and the file of 2D slices are necessary and suitable. To prevent attacks on 3D printing model, we previously proposed a perceptual encryption method 3D printing model that is based on encrypting the features of the interpolating spline curve of degree 2 in 3D space [11], and a selective encryption for 3D printing model based on K-mean clustering and the discrete cosine transform [12]. The main content of these methods is based on the geometric features of 3D printing model in 3D

space or is based on selectivity encrypting in the frequency domain of the discrete cosine transform. Thus, they could not be applied to 2D slices. Because the data of 2D slices is represented in 2D space.

In order to respond to above issue, we would like to propose an encryption solution for 2D slices in 3D printing. The main idea of the proposed solution is to encrypt the 2D slices of 3D printing, both in the frequency domain and the spatial domain. 2D slices are extracted from the file of 3D printing and are then encrypted by the secret key in the frequency domain or in the spatial domain. In this paper, we performed and experimented the proposed solution with both the frequency and spatial domains. To clarify the proposed solution, we organize our paper as follow. In Section 2, we would like to explain the related works as data encryption, 3D model encryption, and 2D slices-based the proposed solution. In Section 3, we present the proposed solution in detail. Experimental results and the evaluation of the proposed solution will be described in Section 4. Section 5 shows the conclusion.

2. Related Works

2.1. Data Encryption

Data encryption is a process of altering the original data to new data that is different with the original data. The conventional work of data encryption is to convert the original data to bits stream and then encrypt it by the encryption standards as DES (Data Encryption Standard), AES (Advanced Encryption Standard), MD5 (Message-Digest Algorithm 5), or Tri-DES, as shown in Figure 3. This way is the simplest method and applied to normal data as text, image or video [13–16]. If we apply this way to the file of 2D slices in 3D printing, we have to convert the file of 2D slices into bits-stream and then encrypt that bits-stream by the data encryption standards as DES, AES or others. This means that we will encrypt some unnecessary contents as text, notation, and header, while the target of encryption is a set of 2D slices. Moreover, the accuracy of this method is not high and the conversation between 3D printing data and bits-stream is the main cause of the error in the decryption process. So, this way is unsuitable for 2D slices encryption.

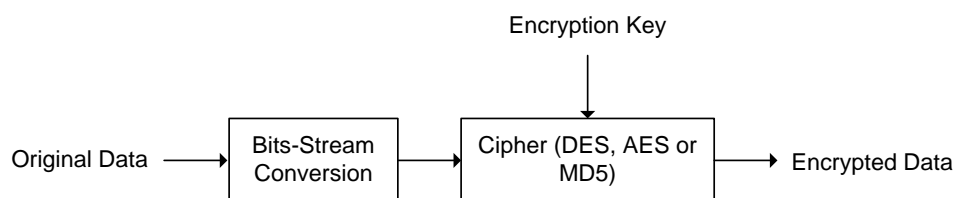


Figure 3. Conventional work of data encryption.

2.2. 3D Model Encryption

Currently, there are some proposed techniques for 3D models encryption [11,12,17,18]. Overall, the main content of these methods is to encrypt 3D model that is based on encrypting the features of 3D model in 3D space. These methods work on 3D data in 3D space to obtain the encrypted 3D model. Thus, they could not be applied to 2D slices. Because 2D slices are 2D data in 2D space. Consequently, the encryption methods for 3D model is unsuitable for the 2D slices of 3D printing.

2.3. 2D Slices-Based Encryption

3D printing technology uses 3D printing models [19,20] to print physical 3D objects. In order to print physical 3D objects, 3D printing models must be cut by a cutting plane along the Z axis from bottom to top via the 3D slicing process. The 3D slicing process is performed by a 3D slicer [21,22]. The output of the 3D slicing process is a set of 2D slices, as shown in Figure 4, and a 3D printer will print physical 3D objects from a set of 2D slices (see Figure 1). The 2D slices that were cut from a 3D printing model, are stored in G-code format [23]. So, in order to encrypt the 2D slices of a 3D print model, we only need to extract 2D slices and then encrypt them by the secret key.

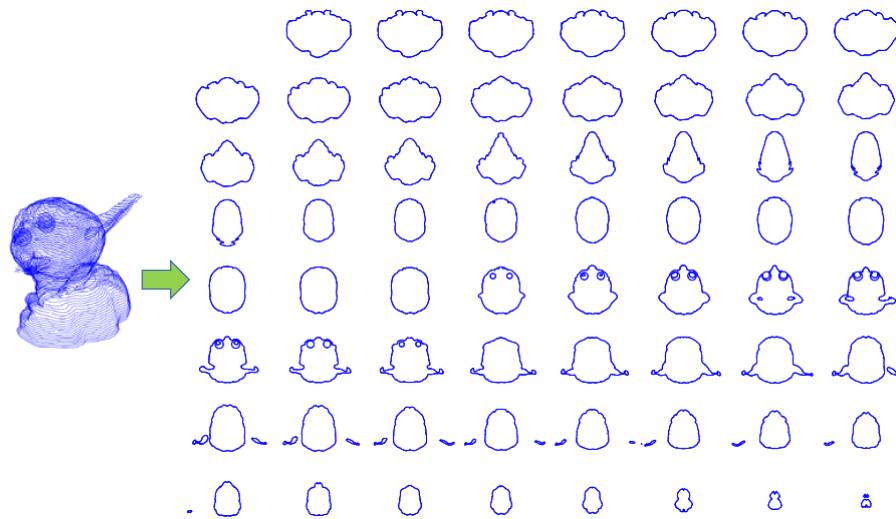


Figure 4. The 2D slices of a 3D printing model in 2D space.

3. The Proposed Solution

3.1. Overview

Overview, the proposed solution is described in Figure 5. 3D printing model is firstly cut into a set of 2D slices via the 3D slicing process. 2D slices are then extracted and are encrypted by the secret key value K to obtain the encrypted 2D slices. The secret key value K is generated by a hashing function with a user’s key input. For the 2D slices encryption process, we would like to propose two methods. First method is the encryption method in the frequency domain. Second method is the encryption method in the spatial domain. The detail of each method will be presented in Sections 3.2 and 3.3.

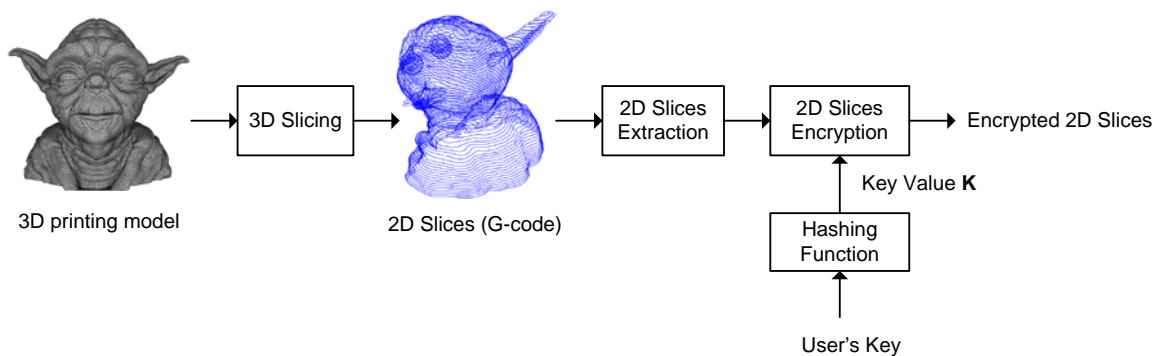


Figure 5. The proposed solution.

To print a physical 3D object, a 3D printing model must be cut into a set of slices by the 3D slicing process. To brief, we define main notation, as follows. Assume that a 3D printing model is cut into a set of 2D slices, $S = \{S_i | i \in [1, N]\}$ where N is the number of slices. Each slice is a set of the intersected points between 3D printing model and the cutting plane, $S_i = \{p_{i,j} | j \in [1, |S_i|]\}$ where $|S_i|$ is the number of points in each slice, and $p_{i,j}$ is the j th point in the i th slice and it is presented by a pair of coordinates $(x_{i,j}, y_{i,j})$.

3.2. Encryption Method in Frequency Domain

In this section, we will present an overview encryption method for 2D slices in the frequency domain, as described in Figure 6a. This method is applied to the “2D slice encryption” step in the

proposed solution. As the mention in Section 3.1, each slice is a set of the intersected points between 3D printing model and the cutting plane. To encrypt 2D slices in the frequency domain, we have to arrange each slice into a matrix, as described in Figure 6b, and then transform this matrix to the frequency domain. In the frequency domain, we selectively encrypt the direct current (DC) coefficient of matrix in the frequency domain by the key value K , and then perform inverse transformation back the spatial domain to obtain the encrypted 2D slices. Since the purpose of the proposed solution is to propose a solution for developers or researchers, they can use some transformations in the frequency domain as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Discrete Fourier Transform (DFT).

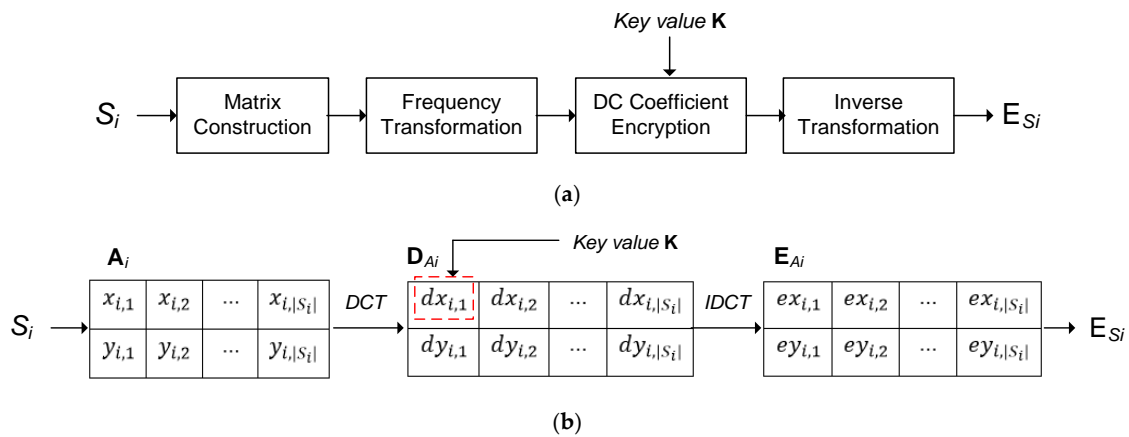


Figure 6. (a) Overview encryption method in the frequency domain, (b) The encryption process of 2D slices in Discrete Cosine Transform (DCT) domain.

Here, we implemented and experimented the proposed method in the DCT domain. Firstly, all the points in the slice S_i are arranged in the matrix A_i , as shown in Equation (1). The matrix A_i is then transformed into the matrix D_{A_i} in DCT domain by the DCT function, as shown in Equation (2). In the DCT domain, the DC coefficient $dx_{i,1}$ of the matrix D_{A_i} is encrypted by the secret key value K as described in Equation (3) with $E_K(\cdot)$ is the encryption function. Finally, the encrypted DC coefficient and DCT coefficients of the matrix D_{A_i} are performed the inverse DCT, as shown in Equation (4) to generate the encrypted matrix E_{A_i} , and this matrix is used to construct the encrypted slice $E_{S_i} = \{e_{i,j} | j \in [1, |S_i|]\}$ with $e_{i,j}(ex_{i,j}, ey_{i,j})$, as shown in Figure 6b. The 2D slices encryption process is described in Figure 6b. We chose DCT for the encryption process because the impact energy of the DCT is massed at DC values. After the DC value encryption and the inverse DCT process, the value of all the coefficients are extensively altered.

$$A_i = \begin{bmatrix} x_{i,1} & x_{i,2} & \dots & x_{i,|S_i|} \\ y_{i,1} & y_{i,2} & \dots & y_{i,|S_i|} \end{bmatrix} \tag{1}$$

$$D_{A_i} = DCT(A_i) = \begin{bmatrix} dx_{i,1} & dx_{i,2} & \dots & dx_{i,|S_i|} \\ dy_{i,1} & dy_{i,2} & \dots & dy_{i,|S_i|} \end{bmatrix} \tag{2}$$

$$dx_{i,1} = E_K(dx_{i,1}, K) = \frac{K}{i + |S_i|} \times dx_{i,1} \tag{3}$$

$$E_{A_i} = IDCT\left(\begin{bmatrix} dx_{i,1} & dx_{i,2} & \dots & dx_{i,|S_i|} \\ dy_{i,1} & dy_{i,2} & \dots & dy_{i,|S_i|} \end{bmatrix}\right) = \begin{bmatrix} ex_{i,1} & ex_{i,2} & \dots & ex_{i,|S_i|} \\ ey_{i,1} & ey_{i,2} & \dots & ey_{i,|S_i|} \end{bmatrix} \tag{4}$$

3.3. Encryption Method in Spatial Domain

In this section, we would like to present an encryption method for 2D slices in the spatial domain. The function of the encryption method in the spatial domain is also similar the function of the encryption method in the frequency domain. It is applied to alter the shape of 2D slices in 3D printing. Due to the fact that the purpose of the encryption in spatial is to alter the shape of 2D slices, we can use geometric transforms for the encryption. But, geometric transforms as rotation, translation, and scaling (RST) only change the spatial location or the size of 2D slice. They did not change the shape of 2D slices. So, they are unsuitable for the encryption process in the spatial domain.

To respond to the purpose of the encryption in the spatial domain and replace the risks of RST, we apply geometric distortion that is also a geometric transform for the 2D slices encryption in the spatial domain. Geometric distortion is a transformation that is used to distort geometric objects [24]. To encrypt the slice S_i by the distortion transform, we have to generate the shearing vector $T_i = \{t_{i,j} | j \in [1, |S_i|]\}$ by the secret key value K with $t_{i,j}$ is computed by Equation (5):

$$t_{i,j} = \frac{K}{|S_i|} \times (i + j) \tag{5}$$

$$E_{S_i} = Distortion(S_i, T_i) = \{e_{i,j} | j \in [1, |S_i|]\} \text{ with } e_{i,j}(ex_{i,j}, ey_{i,j}) = (x_{i,j} + t_{i,j} \cdot y_{i,j}, x_{i,j} \cdot t_{i,j} + y_{i,j}) \tag{6}$$

After that, the slice S_i will be encrypted by the distortion process to generate the encrypted slice E_{S_i} , as shown in Equation (6). The distortion process uses the shearing vector T_i to distort the shape of 2D slices. Figure 7 describes the encryption process of 2D slices in the spatial domain by the key value K and the distortion process.

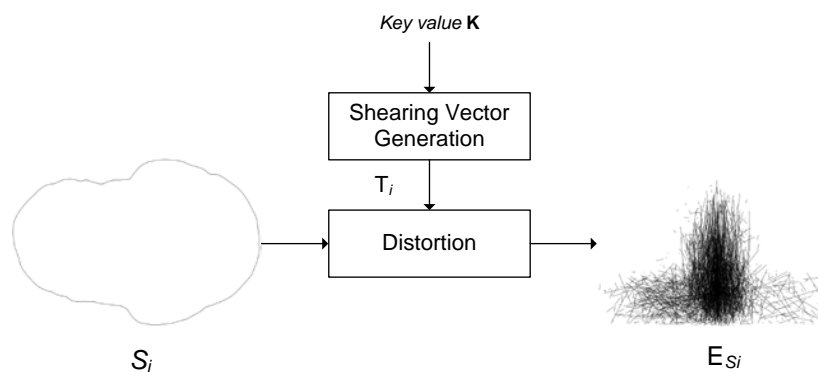


Figure 7. The encryption process of 2D slices in spatial domain.

3.4. Decryption Process

The decryption process is an inverse process with the encryption process. The encrypted 2D slices are extracted from the file that contained the encrypted 2D slices and then they are decrypted by the secret key value K that is used in the encryption process. With the decryption method in the frequency domain, we only arrange the encrypted points of the encrypted 2D slice into a matrix and then transform this matrix into DCT domain. In the DCT domain, we decrypt the encrypted DC coefficient by the secret key value K based on Equation (3), and it then perform the inverse DCT to get the decrypted 2D slice, as shown in Figure 8a. For the decryption method in the spatial domain, we use the secret key value K to compute the shearing vector T_i , as described in Equation (5) and perform geometric re-distortion based on Equation (6) to get the decrypted 2D slice, as shown in Figure 8b.

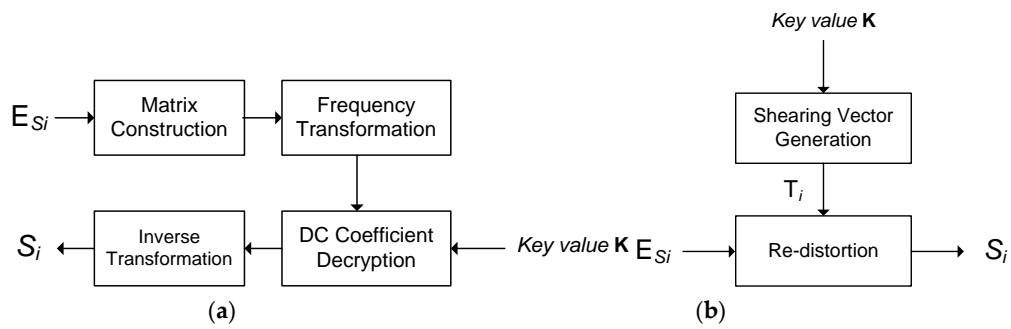


Figure 8. (a) Decryption in the frequency domain, and (b) Decryption in the spatial domain.

4. Experimental Results and Analysis

We experimented the proposed solution with test models in Figure 9. The format of 3D printing models is STL files, VRML files [19,20]. The detailed information of each test model is shown in Table 1. Each test models are cut into a set of 2D slices. The number of 2D slices of each 3D printing model is dependent on both the Z-axis height of that model and the thickness of each slice. The thickness of slice is flexible and determined by user. In experiments, we defined the thickness of slice of 1 mm. In order to evaluate the proposed solution, we evaluated the perceptual encryption result of 2D slices, analyze the security of the encrypted 2D slices and the performance of the proposed solution. Section 4.1 shows the visualization experiments of the encryption process both in the frequency domain and the spatial domain. The security analysis of the encryption methods in the frequency and the spatial domains is explained in Section 4.2. The performance of the proposed solution is shown in Section 4.3.

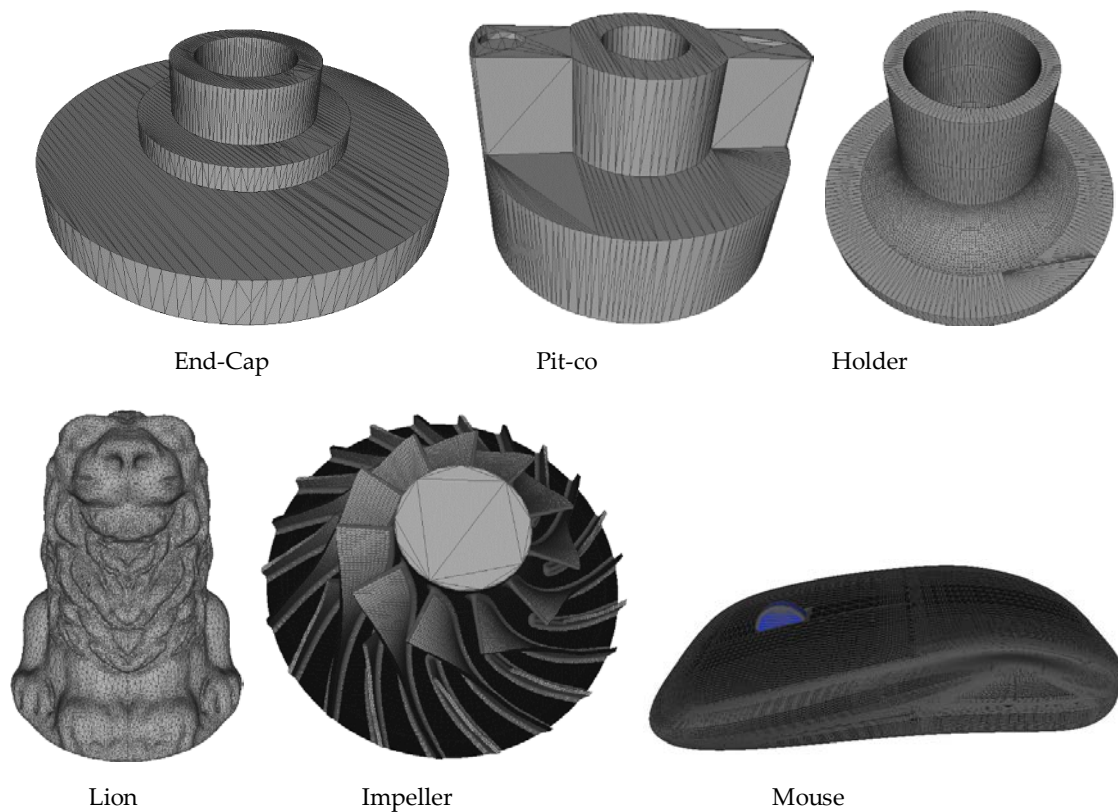


Figure 9. Cont.

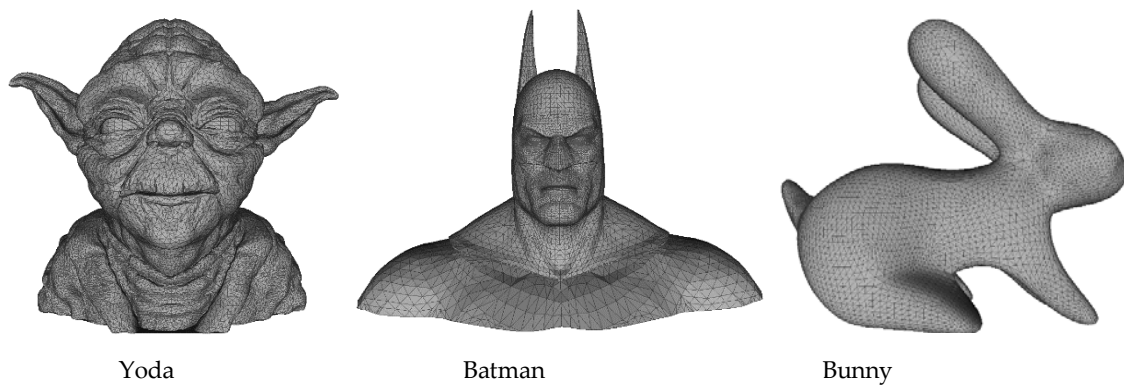


Figure 9. Test Models.

Table 1. Experimental results.

Name	Number of Slices	Entropy (dB)			Computation Time (ms)	
		Conventional Work	Method in DCT Domain	Method in Spatial Domain	Method in DCT Domain	Method in Spatial Domain
End-Cap	6	4608	4623.5	4639.0	43	18
Pit-co	19	4608	4688.7	4769.4	68	35
Holder	23	4608	4712.0	4816.0	72	37
Lion	45	4608	4855.0	5102.0	136	59
Impeller	119	4608	5428.5	6249.0	240	147
Mouse	129	4608	5512.5	6417.0	253	151
Yoda	630	4608	10,466.5	16,325.0	594	322
Batman	730	4608	11,551.5	18,495.1	671	381
Bunny	830	4608	12,656.5	20,705.0	729	425

4.1. Visualization Experiments

We experimented the encryption for the 2D slices of each test model in DCT domain and in the spatial domain. Figure 10 shows the perceptual encryption results with the 2D slices of Bunny model in the frequency domain (DCT domain) and the spatial domain (distortion). After the encryption in the DCT domain, the shape of 2D slices is altered. All the points in each 2D slices are disorderly connected and the segments between two points are broken and not connected (see Figure 10b). In the spatial domain, after the distortion process, the shape of 2D slices also altered completely. All of the points in each 2D slice are disorderly connected or are not connected together. The shape of 2D slices is altered to serrations, as shown in Figure 10c. Consequently, the shape of 2D slices is altered after the encryption process both in the DCT domain and the spatial domain. This means hackers or un-authorized users cannot view the content of 2D slices and cannot use them for 3D printing. We tested the encrypted 2D slices with XYZ 3D Printer Pro 3 in 1 [25], the 3D printer could not print the encrypted 2D slices into a physical 3D object, as shown in Figure 11.

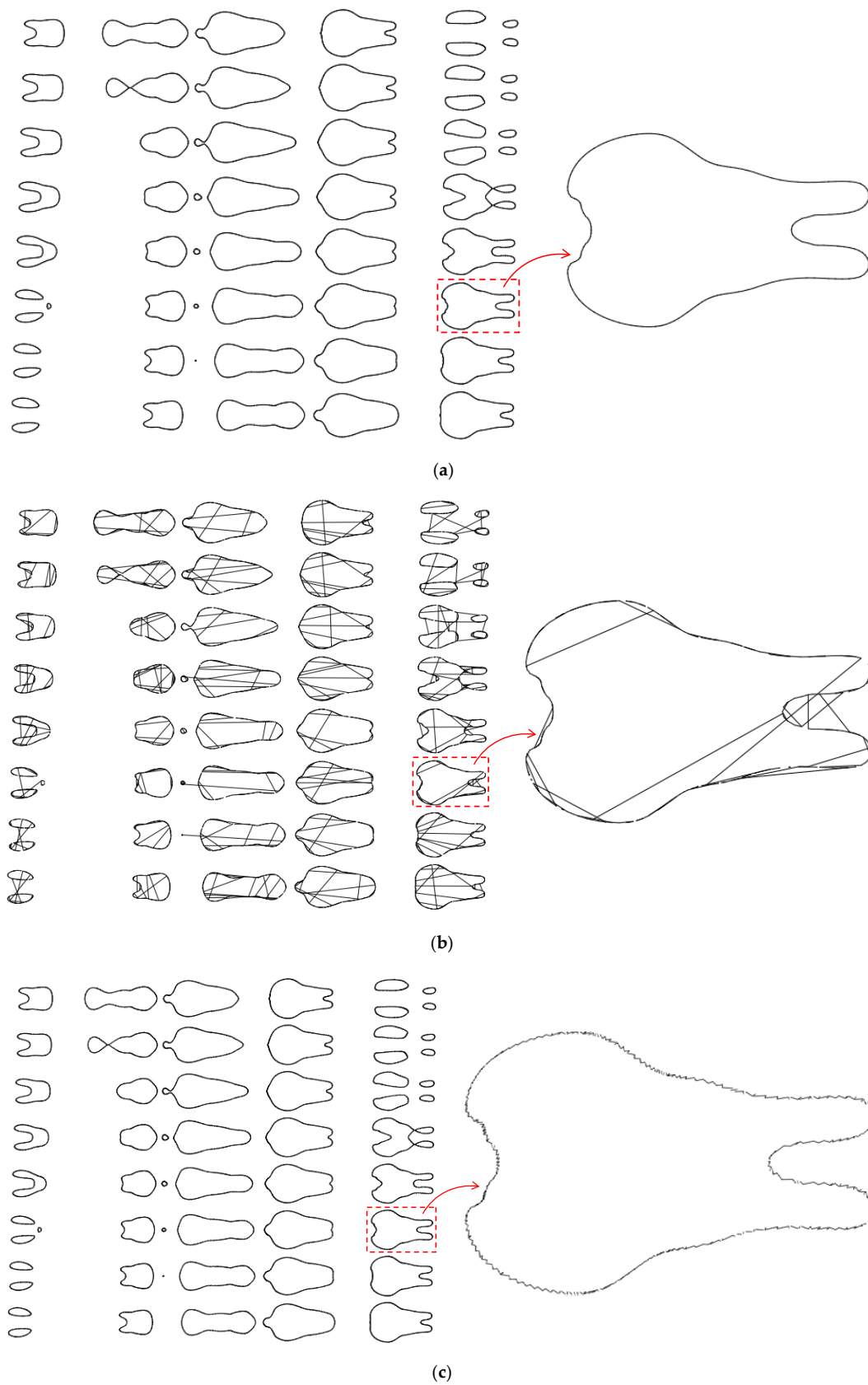


Figure 10. Experimental results 2D slices of Bunny model, (a) Original 2D slices, (b) Encrypted 2D slices in DCT domain, and (c) Encrypted 2D slices in spatial domain.

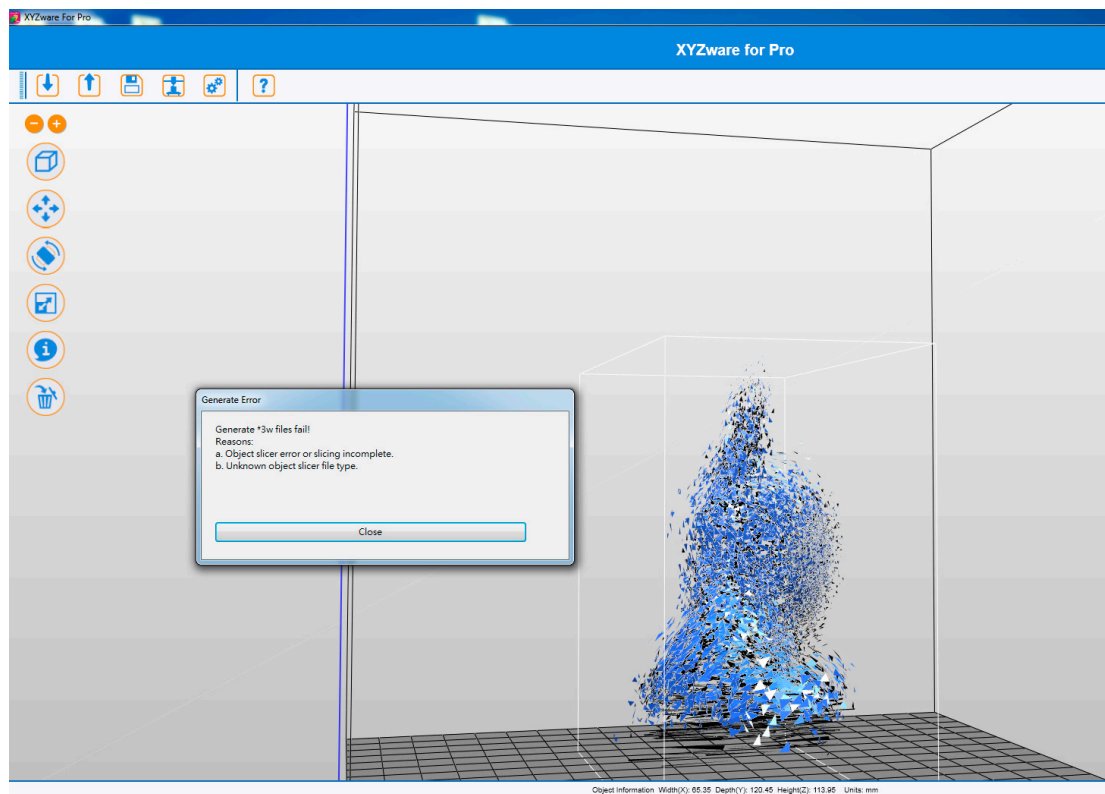


Figure 11. 3D printer cannot print the encrypted 2D slices.

4.2. Security Analysis

To evaluate the security of the proposed solution, we will evaluate the length of the secret key \mathbf{K} and the entropy of the encrypted 2D slices. If the length of the secret key \mathbf{K} is long and the entropy of the encrypted 2D slices is high, the security of the proposed solution will be high. In order to decrypt the encrypted 2D slices, any hacker has to hack the secret key \mathbf{K} . In our solution, we used the SHA-512 algorithm [26] with a 512 bits salt to generate random keys. Thus, if a user uses an English words of length L_k as his password, then an attacker has to calculate $L_k \times 2^{512}$ keys to access the encrypted 3D triangle mesh. The length of each key is 512 bits.

Based on equations in Section 3.2, we can see that the entropy of the encrypted 2D slices in DCT domain is dependent on both the secret key \mathbf{K} and the number of 2D slices N . But, \mathbf{K} and N are random independent variables. So, the entropy of the encrypted 2D slices in DCT domain (frequency domain) H_S^E is the sum of the entropies of variables \mathbf{K} and N , and is determined by Equation (7).

$$H_S^E = H(\mathbf{K}) + H(N) = |\mathbf{K}| \cdot \log_2 |\mathbf{K}| + |N| \cdot \log_2 |N| \quad (7)$$

With the encryption method in the spatial domain, based on equations in Section 3.3, we can see that the entropy of the encrypted 2D slices in the spatial domain is dependent on both the secret key \mathbf{K} , the number of elements in the shearing vector T_i and the number of 2D slices N . But, the number of elements in the shearing vector T_i is equal the number of 2D slices N . So the entropy of the encrypted 2D slices in the spatial domain H_S^S is also the sum of the entropies of variables \mathbf{K} and N , and is determined by Equation (8).

$$H_S^S = H(\mathbf{K}) + H(T_i) + H(N) = |\mathbf{K}| \cdot \log_2 |\mathbf{K}| + |T_i| \cdot \log_2 |T_i| + |N| \cdot \log_2 |N| \quad (8)$$

Due to the fact that the length of the secret key \mathbf{K} is fixed 512 bits, the entropy of the encrypted 2D slices is increased according to the number of 2D slices N , as shown in Table 1. With the encryption

method in the DCT domain, the entropy of the encrypted 2D slices is formed from 4623.5 dB to 12,656.5 dB with $|N| \in [6, 830]$. With the encryption method in the spatial domain, the entropy of the encrypted 2D slices is formed from 4639 dB to 20,705 dB with $|N| \in [6, 830]$. From Equation (7), Equation (8), and Table 1, we concluded that if $|N|$ is high, the entropy will be high.

As shown in Figure 3 in Section 2.1, the conventional work uses only the secret key K to encrypt the bits-stream of 2D slices. Thus, with test models in Table 1 the entropy of the conventional work is always fixed at 4608 dB. This means the entropy of the conventional work is always smaller than the entropy of the proposed solution both in DCT domain and the spatial domain (see Table 1). Based on Equations (7) and (8), we concluded that the entropy of the encryption method in the spatial domain is always higher than the entropy of the encryption method in DCT domain. This means the encryption method in the spatial domain is more security than the encryption method in DCT domain. Figure 12 shows the entropy of the proposed methods with the entropy of the conventional works, according to the number of 2D slices. The entropy of the proposed methods is always higher than the entropy of the conventional work. Consequently, the proposed solution is better and more security than the conventional work.

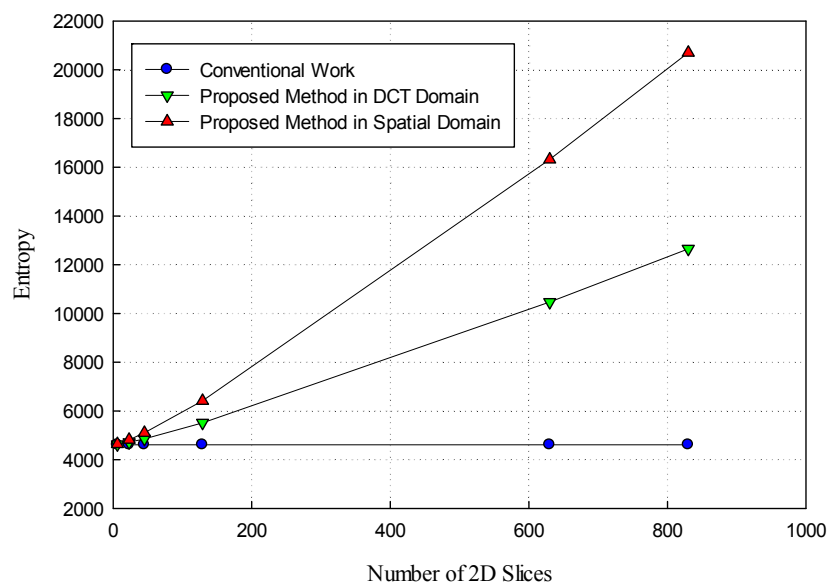


Figure 12. Entropy of the proposed methods according to the number of 2D slices.

4.3. Performance Evaluation

In our experiments, we used an Intel Core i7 Quad 3.5 GHz, 8 GB of RAM, Windows 7 64-bits, and on Visual C# 2013. The computation time of the proposed solution is dependent on the number of 2D slices. With test models in Table 1, the computation time of the encryption method in DCT domain is formed from 43 ms to 729 ms with $|N| \in [6, 830]$, and the computation time of the encryption method in the spatial domain is formed from 18 ms to 425 ms, with $|N| \in [6, 830]$. The cause of this difference is the encryption process in DCT domain be dependent on the computation time of DCT and inverse DCT processes so much. So, the computation time of the encryption method in the DCT domain is longer than the computation time of the encryption method in the spatial domain. This means the encryption process in the spatial domain is faster than the encryption process in DCT domain. Based on Equation (7), Equation (8), and Table 1, we concluded that if the number of 2D slices is small, the computation time will be small and otherwise. We also use the AES standard to encrypted files that contain 2D slices. These files include 2D slices, text, header and other data. The computation time of the conventional work used the AES standard is longer than the computation time of our solution. With test models in Table 1, the computation time of the conventional work is formed from

70 ms to 1372 ms. So, we concluded that our method is better and faster than the conventional work. Figure 13 shows the computation time of methods, according to the number of 2D slices.

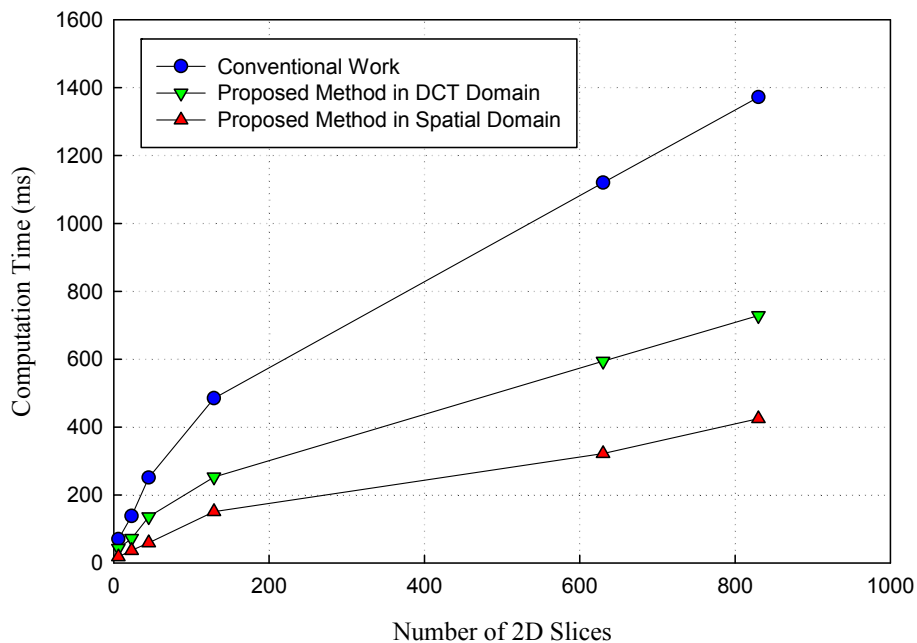


Figure 13. Performance of the proposed algorithm comparing to other methods.

5. Conclusions

In this paper, we proposed a security solution for 3D printing industry based on 2D slices encryption. 2D slices in 3D printing file are encrypted before being storage or transmission to prevent attacks from hackers. We experimented the proposed solution by the encryption processes in the DCT domain and in the spatial domain. Experimental results proved that the proposed solution is effective to 2D slices and independent on the format of 2D slices. The encrypted 2D slices could not printed by 3D printers. The proposed solution provides a better solution and more security than the conventional work. So, we concluded that hackers could not attack the database of 2D slices or fake user to destroy 2D slices when they are transmitted to user via the Internet. The proposed solution is also flexible to developers or researchers. Because, in this paper, we proposed only an overview solution and experimented the 2D slices encryption process by two simple methods in the DCT domain and in the spatial domain. Dependent on the purpose of each application, developers or researchers can replace the content of 2D slices encryption process by their complex methods. This is an advantage of our solution. In the future, we improve the proposed encryption methods and apply them to the secured storage and transmission systems.

Author Contributions: Methodology, G.N.P.; Supervision, K.-R.K.; Validation, S.-H.L. and O.-H.K. In this research activity, all of the authors joined and researched in the data analysis and pre-processing phases, the simulation, the results analysis and discussion, and the manuscript's preparation. All of the authors have approved the submitted manuscript. All of the authors equally contributed to the writing of the paper.

Acknowledgments: This work is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. 2016R1D1A3B03931003, No. 2017R1A2B2012456), and the MSIP (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2017-2016-0-00318) supervised by the IITP (Institute for Information & communications Technology Promotion), the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2015-0-00225) and the Brain Busan 21 (BB21) project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. How 3D Printing Works: The Vision, Innovation and Technologies Behind Inkjet 3D Printing. 3D Systems: Rock Hill, CA, USA. 2012. Available online: http://www.officeproductnews.net/sites/default/files/3dWP_0.pdf (accessed on 14 April 2018).
2. How Paper-Based 3D Printing Works: The Technology and Advantages. Mcor Technologies Ltd. 2013. Available online: <http://rapid3dparts.co.za/how-paper-based-3d-printing-works.pdf> (accessed on 14 March 2018).
3. Lidia, H.A.; Paul, A.J.; Jose, R.J.; Will, H.; Vincent, C.A. White Paper: 3D Printing. Atos: Irving, TX, USA. 2014. Available online: <https://atos.net/wp-content/uploads/2016/06/01052014-AscentWhitePaper-3dPrinting-1.pdf> (accessed on 30 April 2018).
4. What Is 3D Printing & How Do 3D Printers Work? Available online: <https://3dprint.com/82272/what-3d-printing-works/> (accessed on 30 April 2018).
5. Ramya, A.; Vanapalli, S. 3D Printing Technology in Various Applications. *Int. J. Mech. Eng. Technol.* **2016**, *7*, 396–409.
6. Rulania, T. Impact and Applications of 3D Printing Technology. *SSRG Int. J. Comput. Sci. Eng.* **2016**, *3*, 79–82.
7. Helena, D. Applications of 3D printing in healthcare. *Kardiochir. i Torakochirurgia Polska* **2016**, *13*, 283–293.
8. Centralized Command & Control IP, 3D Printers and Users. Available online: <http://secured3d.com/how-it-works> (accessed on 30 April 2018).
9. Security Attack to 3D Printing. Available online: <https://www.claudxiao.net/Attack3DPrinting-Claud-en.pdf> (accessed on 30 April 2018).
10. Ben, B.; Eric, W.; Alex, J.H. Replication Prohibited: Attacking Restricted Keyways with 3D Printing. Available online: <https://www.usenix.org/system/files/conference/woot15/woot15-paper-burgess.pdf> (accessed on 30 April 2018).
11. Pham, G.N.; Lee, S.-H.; Kwon, K.-R. Interpolating Spline Curve-Based Perceptual Encryption for 3D Printing Models. *Appl. Sci.* **2018**, *8*, 242. [CrossRef]
12. Pham, G.N.; Lee, S.-H.; Lee, E.J.; Kwon, K.-R. Selective Encryption Algorithm for 3D Printing Model Based on Clustering and DCT Domain. *J. Comput. Sci. Eng.* **2017**, *11*, 152–159. [CrossRef]
13. Paar, C.; Pelzl, J. Chapter 2 Stream Ciphers. In *Understanding Cryptography*; Springer: Berlin, Germany, 2010; pp. 29–54.
14. Majid, B.; Maarof, M.A. An Efficient Stream Cipher Algorithm for Data Encryption. *Int. J. Comput. Sci. Issues* **2011**, *8*. Available online: <http://www.ijcsi.org/papers/IJCSI-8-3-1-247-253.pdf> (accessed on 30 April 2018).
15. Bhukya, S.; Malathi, N.; Rao, D.S. Welch–Gong 128 Bit Stream Cipher for Encryption and Decryption Algorithm. *Int. J. Emerg. Eng. Res. Technol.* **2015**, *3*, 137–144.
16. Kumari, S. A research Paper on Cryptography Encryption and Compression Techniques. *Int. J. Eng. Comput. Sci.* **2017**, *6*, 20915–20919. [CrossRef]
17. Marc, E.; Maetz, Y.; Gwenael, D. Geometry-Preserving Encryption for 3D Meshes. In Proceedings of the Conference on Compression at Representation Signal Audio, Le Creusot, France, 28–29 November 2013; pp. 7–12.
18. Cai, X.T.; He, F.Z.; Li, W.D.; Li, X.X.; Wu, Y.Q. Parametric and Adaptive Encryption of Feature-Based Computer-Aided Design Models for Cloud-based Collaboration. *Integr. Comput. Aided Eng.* **2017**, *24*, 129–142. [CrossRef]
19. STL Format in 3D Printing. Available online: <https://all3dp.com/what-is-stl-file-format-extension-3d-printing/> (accessed on 14 April 2018).
20. The Virtual Reality Modeling Language. Available online: <http://www.cacr.caltech.edu/~slombey/ascii/vrml/> (accessed on 14 April 2018).
21. 3D Slicer. Available online: <https://www.slicer.org/> (accessed on 14 April 2018).
22. KISS Slicer. Available online: <http://www.kisslicer.com/> (accessed on 14 April 2018).
23. G-Code Tutorial. Available online: <https://www.simplify3d.com/support/articles/3d-printing-gcode-tutorial/> (accessed on 14 April 2018).
24. 2D Geometric Transformations. Available online: <https://www.cs.tau.ac.il/~dcor/Graphics/cg-slides/geom2d.pdf> (accessed on 30 April 2018).

25. XYZ Pro 3 in 1 Printer. Available online: <https://www.xyzprinting.com/en-US/product/da-vinci-1-0-pro-3-in-1> (accessed on 14 April 2018).
26. RSA Lab. *Password-Based Cryptography Standard*; RSA Lab: Bedford, MA, USA, 2006.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).