This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

A 3D Object Encryption Scheme Which Maintains Dimensional and Spatial Stability

Alireza Jolfaei, Xin-Wen Wu, and Vallipuram Muthukkumarasamy

***Abstract*— Due to widespread applications of three-dimensional (3D) vision technology, the research into 3D object protection is primarily important. To maintain confidentiality, encryption of 3D objects is essential. However, the requirements and limitations imposed by 3D objects indicate the impropriety of conventional cryptosystems for 3D object encryption. This suggests the necessity of designing new ciphers. In addition, the study of prior works indicates that the majority of problems encountered with encrypting 3D objects are about point cloud protection, dimensional and spatial stability, and robustness against surface reconstruction attacks. To address these problems, this paper proposes a 3D object encryption scheme, based on a series of random permutations and rotations, which deform the geometry of the point cloud. Since the inverse of a permutation and a rotation matrix is its transpose, the decryption implementation is very efficient. Our statistical analyses show that within the cipher point cloud, points are randomly distributed. Furthermore, the proposed cipher leaks no information regarding the geometric structure of the plain point cloud, and is also highly sensitive to the changes of the plaintext and secret key. The theoretical and experimental analyses demonstrate the security, effectiveness and robustness of the proposed cipher against surface reconstruction attacks.**

***Index Terms*—3D object encryption, geometry deformation, permutation, geometric rotation, cryptanalysis, statistical analysis**

I. Introduction

Advances of multimedia computing and networking have unlocked the path for the application of 3D objects in a variety of domains, including virtual reality and augmented reality. The fast growing demand for high definition visualization applications has opened up a number of challenges regarding the confidentiality of 3D objects. Secure communication of 3D objects is a legitimate concern of Intellectual Property (IP) owners, developers, government regulatory bodies and law enforcement agencies. Thus, there is a strong need to protect 3D objects against unauthorized use or other security violations. To maintain confidentiality of 3D objects, encryption is essential.

Since the 1970s, a large number of encryption schemes have been proposed, among which some have been standardized and widely adopted all over the world, such as Data Encryption Standard (DES) [1] and Advanced Encryption Standard (AES) [2]. Thus, it seems natural to use established and tested ciphers to encrypt 3D objects bit by bit. This simple and naïve approach has already been used in many Digital Rights Management (DRM) systems [3], [4]. For instance, in Extensible Markup Language (XML) encryption [5], the content of the XML elements is considered as a binary stream and thus, while keeping the syntax format, it is encrypted by conventional ciphers. This approach provides the same level of security as that of the conventionally used cipher. It also offers granularity. In other words, users can choose to encrypt only a few of many objects in a complicated 3D scene. However, due to special features of 3D objects, naïve encryption may not be a suitable solution for many 3D applications. The problem of 3D object encryption is beyond the application of established and well-known encryption algorithms. This is primarily due to the structure of 3D objects and the way they are used commercially. Unlike data encryption, where a complete bitstream is encrypted, 3D content encryption introduces several challenges.

In comparison with 1D and 2D data, 3D objects imply a higher level representation or semantics. 3D objects have a 3D geometry and in many applications, such as socializing metaverses and games, there is a requirement to display such geometry in a 3D space. Hence, 3D objects should be confined to a virtual space in order to be displayed. If such consideration is not taken into account, then encrypted objects may exceed the viewing screen resolution or they may overlap with other objects of the virtual world. Hence, the encryption outcome may conceal other objects behind its surface. This decreases users' observation capability. Conventional ciphers, such as AES, are oblivious to dimensional and spatial stability of 3D objects and hence, the rendering may spill out the encrypted outcome from the intended size and location, and corrupt the whole 3D scene. In practical terms, conventional ciphers destroy the spatial and dimensional stability. It can be argued that the encrypted content may not require rendering and it would be better to display nothing. This may not be an ideal solution. In many applications, such as in virtual museums and 3D e-commerce, if users have no means of noticing that something is missing, it is then unlikely that they will find that something is missing. In practical terms, the motivation to pay for having access to 3D content would be lacking.

An example for the dimensional and spatial stability requirement is the case of a game production workflow where a crew of level designers, artists and programmers cooperate under the supervision of a producer. The output of the

The authors are with the School of Information and Communication Technology, Griffith University, Gold Coast, QLD 4222, Australia. (e-mail: alireza.jolfaei@griffithuni.edu.au, {x.wu, v.muthu}@griffith.edu.au)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

production workflow is a game program and a series of game assets to be consumed by the game program. These assets include 3D models and images data. The 3D designs are normally updated during the production workflow. For instance, some parts or textures can be added to or added from a particular object. In the production workflow, the design studio may not be willing to give access to the full scene to a particular crew. However, this crew may need to have access to visual cues in order to insert graphical elements without striking existing ones that would be invisible because of the protection framework.

3D objects have different kinds of representations depending on the type of 3D scanning device used. For instance, 3D laser scanners produce a cloud of points while Computed Tomography (CT) scanners create a volumetric model in the form of a 3D cube. In addition, different 3D applications, such as medical applications and Computer-Aided Design (CAD), use different representations. In this paper, we focus on point cloud model. Over the last decade, point clouds have gained more attention [6] and have been used for a multitude of modeling tasks, such as editing [7] and compression [8]. The reason for this popularity is that the point cloud representation offers several advantages compared to other 3D object representations. Firstly, it is an explicit method of demonstrating the 3D raw data captured by 3D sensors. This representation is also simple and flexible, and does not require any information about connectivity or topological consistency. Hence, it can be directly used as a rendering primitive to circumvent the need for difficult surface construction procedures. Moreover, it is very efficient when the available point data density exceeds the viewing screen resolution [9], because there is no need to maintain, store and render the polygons associated with the edge of the off-screen mesh. More importantly, other 3D representations, such as polygon soups and meshes, can be easily converted into point clouds by sampling. Therefore, methods applied on the point cloud representation can be extended to other 3D representations.

In this paper, we propose a chaos-based symmetric encryption scheme for protecting 3D objects. Firstly, due to the large amount of data involved in a 3D object, it requires a huge storage capacity and transmission bandwidth. To provide a better execution performance, we consider symmetric (rather than asymmetric) encryption. Secondly, chaos-based cryptographic primitives have a number of advantages as shown in [10], such as the sensitivity to input data. Chaos has been used to design encryption schemes for multimedia data [11], such as image and video. However, these image and video encryption schemes are not applicable to 3D object encryption due to the 3D geometry and representation, as explained in the above discussion.

The proposed cipher employs random permutation matrices and random geometric rotations to deform the geometry of 3D objects. Permutation and rotation are linear transformations represented by orthogonal matrices. Since the inverse of an orthogonal matrix is its transpose, no extra calculation is required to implement the decryption procedure which is based on the inverse matrix. This remarkable property makes the implementation of decryption very efficient.

An overview of previous studies in the area of 3D object protection demonstrates that this research is mainly focused on 3D digital watermarking and is in an effort to pinpoint the source of leaks (traitor tracing) [12], [13]. However, digital watermarking is a complement to encryption and one can never be used to replace the other. To prevent unauthorized users from accessing valuable 3D content, secure encryption schemes need to be studied. Despite the importance of 3D content encryption, few technological solutions have been given [14], [15], [16], [17], [18], [19]. However, the method proposed in [14] is not applicable to point cloud representation and may leak the point cloud information, and the methods proposed in [15], [18] and [19] are not secure as they leak the point cloud information and cannot resist surface reconstruction attacks. In addition, the method suggested in [16] cannot maintain dimensional and spatial stability, and the method proposed in [17] is not compatible with standard file formats.

Following the above discussion, this study addresses the major shortcomings of the literature, and gives a technical solution to the problem of 3D content encryption, which encrypts 3D point cloud based objects making use of pseudorandom permutations and geometric rotations. The proposed cipher is compatible with standard file formats and maintains the semantic requirements of 3D objects, including the dimensional and spatial stability. We showed that the proposed cipher has a large key space and so is robust against brute-force attack. The rigorous security analysis showed no statistical weaknesses in the cipher and demonstrated no simple method of recovering the secret key. It also confirmed the security of the proposed cipher against known/chosen plaintext attacks and surface reconstruction attacks. The sensitivity analyses indicated that the proposed cipher is highly sensitive to the changes of the plaintext and secret key. Moreover, a spatial randomness test determined that there is no presence of homogenous patterns in the cipher point clouds. In addition to security evaluations, a performance analysis was performed to evaluate the encryption speed of the proposed cipher.

The remainder of the paper is organized as follows: Section 2 reviews the related work in 3D object encryption. Section 3 provides details of the proposed encryption and decryption schemes. Section 4 proves that the proposed scheme ensures the dimensional and spatial stability of the original content. Sections 5 and 6 evaluate the security of the proposed cipher using cryptanalysis and statistical methods, respectively. Section 7 measures the performance of the proposed cipher by calculating its computational complexity and encryption/ decryption time. Finally, Section 8 concludes that the proposed scheme is secure, efficient and feasible.

II. Related Work

To address the confidentiality problem of 3D objects, several researchers proposed a number of initial solutions. These solutions are briefly described as follows. In [14], Koller et al. investigated several possible protection methods for ensuring the security of the high-resolution geometric details of 3D objects in the underlying 3D graphics system. Based on this investigation, they developed a remote rendering

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

system, which included a 3D viewer client and a rendering server, appropriate for secure distribution of 3D objects. The rendering server used a number of defensive approaches, such as monitoring and limiting request streams, to protect the 3D geometry from unauthorized extraction. The Koller et al.'s approach protects geometric detail of 3D objects by rendering a lower resolution version (fewer polygon surfaces) of 3D models to unauthorized users. In this method, what users see is the snapshots of the rendering results in the rendering server rather than the results rendered with the client's graphics pipeline. The advantage of this method is that the transmission of sequence of images, rather than 3D models, reduces startup latency on the client side. Thus, protected objects can be displayed quickly. However, displaying 2D images of low resolution 3D models can locally leak the surface information. Therefore, to increase the scheme's resistance to surface reconstruction methods, such as shape-from-silhouette attacks [20] and shape-from-shading attacks [21], Koller et al. applied particular perturbation and distortion mechanisms, such as perturbing the viewing and lighting parameters, and adding noise to the rendered images. Although these defensive mechanisms can reduce the amount of surface information leaked from 2D images, they cannot completely protect the point clouds' vertex information. For instance, if the server's 3D graphics system ignores the facets rendering, then the vertices of the 3D object will be revealed. This provides enough information to the potential adversary to reconstruct the surface [22], [23]. In addition, as the encryption/decryption process is handled within a remote rendering system, this method cannot be used to protect the 3D data stream in applications running in local computers, such as games.

In [15], Pan et al. proposed an encryption method based on vertex shader programming to protect the transmission of 3D models. In this approach, the coordinates of each point are permuted by a simple comparison with the average of the coordinates. Security of such cipher solely relies on a simple permutation of coordinates, and it is therefore easy to break the cipher and deduce the original 3D data. As well as the mentioned design's shortcoming, this method has a number of drawbacks. Firstly, the proposed distortions may not sufficiently distort the 3D object, and hence the underlying pattern of the encrypted object would still be distinguishable after rendering. This indicates the vulnerability of Pan et al.'s scheme to surface reconstruction attacks. In addition, this method encrypts all 3D objects which may not be necessary.

In [16], a digital rights enabled graphics processing system was proposed by Shi et al. In this system, firstly the privacy-sensitive 3D content primitives, including vertices and textures, are encrypted by the content provider. Hence, only valuable graphics data (not all data) are encrypted. The decryption process is then handled within the graphics pipeline, under the control of licenses. The encryption and decryption are performed using AES in CBC (cipher block chaining) or counter mode. This system renders only 3D objects with multi-resolution representations, which have different levels of detail. By this method, protected and unprotected versions of the 3D object primitives can simultaneously be delivered. In this method, it is difficult to exploit particular properties of a specific 3D object by running software exploits or simple hardware-based tampering. Although Shi et al.'s system is a real progress towards secure 3D environments; it cannot render the protected scenes with other representation formats due to the interoperability issues.

In [17], Phelps described a 3D object protection method by encrypting the privacy sensitive object and representing a dummy object, such as a bounding box, with the same size and the same location in lieu. In this method, the dummy object is stored as non-encrypted data in one file and the protected 3D object as encrypted data in a separate file. Any user may access the non-encrypted data, but only authorized users can access the encrypted data; non-authorized users see the dummy object thereof. This method informs the unauthorized users about the location and boundaries of the encrypted content. Hence, any content modifications that may lead to interference among objects can be avoided. However, this method is not compatible with the standard file formats because it only works with a specific format that stores privacy sensitive objects in two separate files, one file as a non-encrypted data (dummy) and the other one as an encrypted data. Hence, it is only usable by adapted rendering devices, not standard ones. In practical terms, the encrypted data does not respect the syntax of most 3D techniques, and therefore cannot be widely adopted for any rendering devices.

In [18] and [19], the Technicolor researchers proposed a 3D encryption technique based on vertex coordinates shuffling to deform and encrypt 3D content. Similarly to Pan et al.'s encryption scheme, Technicolor's proposal is a permutation-only cipher. However, it permutes the set of individual coordinates of vertices rather than permuting the coordinates of each point. A distinct characteristic of Technicolor's encryption scheme is that it keeps the dimensions of the cipher object bounded within a box with respect to the bounded dimensions of the plain object. Using this approach, the cipher object's dimension cannot exceed the virtual world's dimensions. Thus, the cipher object can be rendered without any undesired interference with the objects of the virtual world. However, Technicolor's encryption scheme is not secure because the secret key can be easily deduced by a known/ chosen plaintext attack (see Section 5 for further explanation).

The study of related works indicate that the major problems in 3D object encryption are about point cloud protection, dimensional and spatial stability, and robustness against surface reconstruction attacks. Therefore, we aim to address these problems by proposing an appropriate methodology in Section 3.

III. Proposed Encryption Scheme

In this section, the encryption and decryption procedures of the proposed cipher will be described. Before the algorithm description, we firstly explain the content-dependent metadata employed to ensure the semantic requirements of 3D point clouds. 3D point clouds must be placed in designated locations and must be confined in limited spaces with defined boundaries. Hence, we assume that every point cloud is distributed in a bounding sphere of radius r with center $\mathbf{G}$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

where, **G** is the barycenter of the point cloud and r is the distance from the farthest vertex to **G**. In certain 3D content applications, such as animation and game production workflow, the unauthorized designers may need to know the size and position of objects in order to insert graphical elements without striking existing ones that would be invisible because of the protection framework. Hence, r and **G** need to be discernible to both authorized and unauthorized users. We therefore keep r and **G** as public information and obfuscate privacy-sensitive information, that is, point cloud vertices, with respect to these public content-dependent metadata.

To elaborate the steps of the encryption algorithm, let $\mathbf{P}^1$, $\mathbf{P}^2$, …, $\mathbf{P}^n$ be points of the plain point cloud, and $\mathbf{C}^1$, $\mathbf{C}^2$, …, $\mathbf{C}^n$ be points of the corresponding cipher point cloud. For any j ($1 \le j \le n$), $\mathbf{P}^j$ and $\mathbf{C}^j$ are defined as follows:

$$\mathbf{P}^j = \begin{bmatrix} p_1^j \\ p_2^j \\ p_3^j \end{bmatrix} \in \mathbb{R}^3,\ p_1^j, p_2^j, p_3^j \in \mathbb{R}, \tag{1}$$

$$\mathbf{C}^j = \begin{bmatrix} c_1^j \\ c_2^j \\ c_3^j \end{bmatrix} \in \mathbb{R}^3,\ c_1^j, c_2^j, c_3^j \in \mathbb{R}. \tag{2}$$

To achieve the confusion and diffusion properties defined by Shannon [24], and to maintain the dimensional and spatial stability, we propose an n-round encryption scheme based on a combination of a confusion-diffusion structure and 3D geometric rotations. As we will analyze later in Section 5, two rounds of encryption is enough to have a secure cipher. More rounds of encryption give a higher level of security at the price of losing efficiency. Without loss of generality, in the following subsections, the components used in the encryption algorithm as well as the detailed description of the two-round encryption algorithm will be given. To illustrate the encryption steps, a block diagram of the proposed encryption algorithm is depicted in Figure 1.

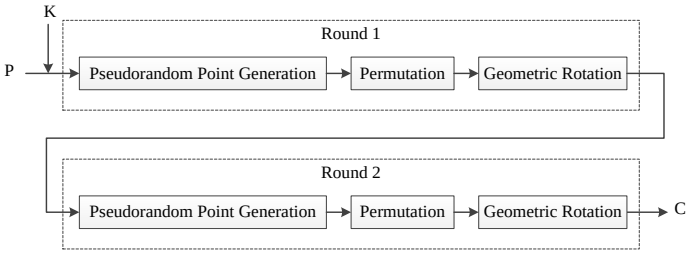

Fig. 1. The structure of the encryption algorithm.

A. Key Scheduling Algorithm

To encrypt a large number of points, we need to iterate the encryption operations several times. Therefore, we first present a key scheduling algorithm based on a Chebyshev map [25] to expand the relatively short secret key to a large expanded key. To avoid simple relationships between the secret key and the expanded key stream and to resist certain types of cryptanalysis, such as related-key attacks and slide attacks, the key schedule algorithm produces different key streams for different encryption rounds [26].

The key scheduling algorithm is described as follows:

$$k_i^j = \cos\left(D \cos^{-1}(k_i^{j-1})\right), \text{ for } 1 \le i \le 6 \text{ and } 1 \le j \le 2n, \tag{3}$$

where D is a constant denoting the degree of the Chebyshev map, $K = \left(k_1^0, k_2^0, k_3^0, k_4^0, k_5^0, k_6^0\right)$ denotes the encryption seed point (secret key), and for any i ($1 \le i \le 6$), $k_i^0 \in [-1, +1]$.

B. Pseudorandom Point Generation Process

To meet the confusion requirements, the relationship between the encryption key and the cipher point cloud should be made as complex as possible. To this end, in each encryption round, n pseudorandom points are generated within the point cloud's sphere of radius $\sqrt{3}r$ with center **G**, as follows:

$$\mathbf{O}_v^j = \mathbf{G} + r \cdot \left[k_1^{j+\lfloor \frac{v}{2} \rfloor n}, k_2^{j+\lfloor \frac{v}{2} \rfloor n}, k_3^{j+\lfloor \frac{v}{2} \rfloor n}\right]^t, 1 \le v \le 2 \text{ and } 1 \le j \le n, \tag{4}$$

where v represents the encryption round, j represents the point index and $\mathbf{A}^t$ denotes the transpose of matrix **A**. (k_1^0, k_2^0, k_3^0) is the seed point for this point generation process.

C. Permutation process

To meet the diffusion requirements, the statistical relationship between the plain and cipher point clouds should be made as complex as possible. To this end, the dimensional coordinates of points of the plain point cloud are shuffled with the coordinates of pseudorandom 3D points. This dissipates any meaningful relationship between the points of the plain point cloud. Permutation of a large number of coordinates, for instance, all of the dimensional coordinates at once, may not be an efficient approach because producing a large-scale permutation matrix requires a considerable amount of computation, time and memory. In order to increase the permutation efficiency of the encryption scheme, permutations are locally employed to reorder subsets of the set of all dimensional coordinates of the plain-points and pseudorandom points specified in the previous subsection. The following definitions are given for the one round of the encryption scheme to elaborate the permutation process.

Definition 1. Let $\pi = \{k_1^1, \ldots, k_1^n, k_2^1, \ldots, k_2^n, k_3^1, \ldots, k_3^n, k_4^1, \ldots, k_4^n, k_5^1, \ldots, k_5^n, k_6^1, \ldots, k_6^n\}$ denote the permutation keystream.

Definition 2. Given n plain-points $\mathbf{P}^j$ and n pseudorandom points $\mathbf{O}^j$, $j = 1, \ldots, n$, the universal set U is defined as the set of all dimensional coordinates of $\mathbf{P}^j$ and $\mathbf{O}^j$, that is,

$$U = \{u_k | u_{i+3(j-1)} = p_i^j \text{ and } u_{i+3(j-1)+3n} = o_i^j, \text{ for } 1 \le i \le n \text{ and } 1 \le j \le n\}. \tag{5}$$

By definition above, the cardinality of the universal set $\#U = 6n$.

Remark 1. Given an input array of size $6n$, there are $(6n)!$ possible permutations for the inputs. To sort this input, any deterministic comparison-based sorting algorithm requires performing $O(n \cdot \log n)$ comparisons in the worst case [27]. To reduce this complexity to $O(n)$ and perform an efficient permutation, the universal set is first partitioned into $\lfloor \frac{n}{8} \rfloor$ small subsets defined as follows:

Definition 3. For any m ($1 \le m \le \lfloor \frac{n}{8} \rfloor$), let X_m denote a subset of the universal set U, which is defined as follows:

$$X_m = \{x_k | x_{i+3(j-1)} = p_i^{j+8(m-1)} \text{ and } x_{i+3(j-1)+24} = o_i^{j+8(m-1)}, \text{ for } 1 \le i \le 3 \text{ and } 1 \le j \le 8\}. \tag{6}$$

If $8 \nmid n$, the last l plain-points and l pseudorandom points remain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

unclassed, where $l = n - 8\lfloor \frac{n}{8} \rfloor$. In this case, these elements are added to the last subset $X_{\lfloor \frac{n}{8} \rfloor}$, which is constructed as follows:

$$X_{\lfloor \frac{n}{8} \rfloor} = \{x_k | x_{i+3(j-1)} = p_i^{j+(n-l)-8} \quad \text{and} \quad x_{i+3(j-1)+24} = o_i^{j+(n-l)-8}, \text{ for } 1 \le i \le 3 \text{ and } 1 \le j \le 8 + l\}, \tag{7}$$

where $l = n - 8\lfloor \frac{n}{8} \rfloor$. For any m ($1 \le m \le \lfloor \frac{n}{8} \rfloor - 1$), by definition the cardinality of each subset $\#(X_m) = 48$. If $8 \mid n$, then the cardinality of the last subset $\#(X_{\lfloor \frac{n}{8} \rfloor}) = 48$; otherwise $\#(X_{\lfloor \frac{n}{8} \rfloor}) = 48 + 6l$. Obviously, by definition the following conditions hold for the subsets X_m:

$$\bigcup_{m=1}^{\lfloor \frac{n}{8} \rfloor} X_m = U, \tag{8}$$

and

$$X_m \bigcap X_n = \varnothing, \text{ for } m \ne n. \tag{9}$$

Remark 2. For any m ($1 \le m \le \lfloor \frac{n}{8} \rfloor$), permutations are performed locally in each subset X_m, where $48 \le \#(X_m) \le 90$. This makes the permutation more efficient because for each subset X_m, there are $\#(X_m)!$ possible permutations, and sorting the elements of each subset X_m requires $O(1)$ comparisons. Therefore, the computational complexity of rearranging the elements in all of the $\lfloor \frac{n}{8} \rfloor$ subsets is $O(n)$. This is more efficient than the permutation of the whole $(6n)$ elements at once.

Remark 3. The design rationale for specifying a particular size for the subsets is to achieve resistance to known/chosen plaintext attacks by two rounds of encryption. To clarify further, see Section 5 for the detail of cryptanalysis.

Utilizing the definitions above, the permutation process is defined as follows. For any m ($1 \le m < \lfloor \frac{n}{8} \rfloor$), $\prod_{\{\pi_i\}_{i=48(m-1)+1}^{48(m-1)+48}}$: $X_m \to X'_m$, where

$$X'_m = \left\{ x'_j | x'_j \in X_m,\ x'_j = \prod_{\{\pi_j\}}(x_j), \text{for } 1 \le j \le 48 \right\}. \tag{10}$$

If $8 \mid n$, then for $m = \lfloor \frac{n}{8} \rfloor$ the same mapping is applied; Otherwise, for l ($1 \le l \le 7$), $8 \mid n - l$, $\prod_{\{\pi_i\}_{i=6(n-l)-47}^{6n}}: X_{\lfloor \frac{n}{8} \rfloor} \to X'_{\lfloor \frac{n}{8} \rfloor}$, where

$$X'_{\lfloor \frac{n}{8} \rfloor} = \left\{ x'_j | x'_j \in X_{\lfloor \frac{n}{8} \rfloor}, x'_j = \prod_{\{\pi_j\}}(x_j), \text{for } 1 \le j \le 48 + l \right\}. \tag{11}$$

The permutation outcome, including the permuted plain point cloud $P' = \{\mathbf{P'^1}, \mathbf{P'^2}, \ldots, \mathbf{P'^n}\}$ and the permuted pseudorandom points $O' = \{\mathbf{O'^1}, \mathbf{O'^2}, \ldots, \mathbf{O'^n}\}$, are obtained as follows. For any m ($1 \le m < \lfloor \frac{n}{8} \rfloor$),

$$\mathbf{P'}^{j+8(m-1)} = \left(x'_{3(j-1)+1}, x'_{3(j-1)+2}, x'_{3(j-1)+3} \right)^t, \text{ for } 1 \le j \le 8, \tag{12}$$

$$\mathbf{O'}^{(j-8)+8(m-1)} = \left(x'_{3(j-1)+1}, x'_{3(j-1)+2}, x'_{3(j-1)+3} \right)^t, \text{for } 9 \le j \le 16. \tag{13}$$

If $8 \mid n$, then the above relations hold for $m = \lfloor \frac{n}{8} \rfloor$. If for l ($1 \le l \le 7$), $8 \mid n - l$, then for $m = \lfloor \frac{n}{8} \rfloor$, the permuted plain point cloud and the permuted pseudorandom points are obtained as follows:

$$\mathbf{P'}^{j+(n-l)-8} = \left(x'_{3(j-1)+1}, x'_{3(j-1)+2}, x'_{3(j-1)+3} \right)^t, \text{ for } 1 \le j \le 8+l, \tag{14}$$

$$\mathbf{O'}^{(j-8)+(n-l)-8} = \left(x'_{3(j-1)+1}, x'_{3(j-1)+2}, x'_{3(j-1)+3} \right)^t, \text{ for } 9+l \le j \le 16+2l. \tag{15}$$

D. Geometric Rotation Process

To maintain the dimensional and spatial stability, and also to safeguard the confusion-diffusion process from chosen-plaintext attacks, such as point re-ordering attacks [12], for any j ($1 \le j \le n$), the j-th point of P' is geometrically rotated about the j-th point of O' with random Euler angles $\left(\alpha_1^j, \alpha_2^j, \alpha_3^j\right)$. As the inverse of a 3D rotation matrix is equal to its transpose, no extra calculation is required to compute the reciprocal matrix. This remarkable property in the design of our cryptographic algorithm makes the implementation of decryption very efficient. The geometric rotation function $\mathcal{R}ot(\cdot)$ is defined as follows:

$$\mathcal{R}ot_K(\mathbf{P'}^j) = \psi \cdot \mathbf{R}^j\left(\alpha_1^j, \alpha_2^j, \alpha_3^j\right) \times [\mathbf{P'}^j - \mathbf{O'}^j] + \mathbf{O'}^j, \text{for } 1 \le j \le n, \tag{16}$$

where $K = \left(k_4^0, k_5^0, k_6^0\right)$ denotes the seed point (secret key), and $\mathbf{R}(\alpha_1, \alpha_2, \alpha_3)$ is the 3D rotation matrix defined as follows:

$$\mathbf{R}(\alpha_1, \alpha_2, \alpha_3) = \begin{bmatrix} \cos(\alpha_2)\cos(\alpha_3) & \sin(\alpha_1)\sin(\alpha_2)\cos(\alpha_3) - \cos(\alpha_1)\sin(\alpha_3) & \cos(\alpha_1)\sin(\alpha_2)\cos(\alpha_3) + \sin(\alpha_1)\sin(\alpha_3) \\ \cos(\alpha_2)\sin(\alpha_3) & \sin(\alpha_1)\sin(\alpha_2)\sin(\alpha_3) + \cos(\alpha_1)\cos(\alpha_3) & \cos(\alpha_1)\sin(\alpha_2)\sin(\alpha_3) - \sin(\alpha_1)\cos(\alpha_3) \\ -\sin(\alpha_2) & \sin(\alpha_1)\cos(\alpha_2) & \cos(\alpha_1)\cos(\alpha_2) \end{bmatrix}, \tag{17}$$

$0 < \psi \le \frac{1}{9}$ is the scaling factor which is used to adjust the size of the rotated point cloud. For any encryption round l ($1 \le l \le 2$), Euler angles are obtained as follows:

$$\alpha_i^{j+\lfloor \frac{l}{2} \rfloor n} = \left\lfloor 180 k_{i+3}^{j+\lfloor \frac{l}{2} \rfloor n} \right\rfloor, \text{ for } 1 \le i \le 3 \text{ and } 1 \le j \le n. \tag{18}$$

E. Encryption and Decryption Algorithms

Details of the proposed encryption and decryption algorithms are described as pseudo-codes in Algorithm 1 and Algorithm 2, respectively. The proposed cipher is based on a series of permutations and rotations. Both permutation and rotation are linear transformations represented by orthogonal matrices. Since the inverse of an orthogonal matrix is its transpose, no extra calculation is required to implement the decryption procedure which is based on the inverse matrix.

Algorithm 1. Pseudo-code of the encryption algorithm

Input: Plain-points $\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n$, content-dependent metadata including r and $\mathbf{G}$, control parameter ψ, and secret key K
Output: Cipher-points $\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^n$
Generate an expanded key stream using the key scheduling algorithm, as explained in Section 3.*A*.
For *Round* = 1 to 2
  Generate $\mathbf{O}^1, \mathbf{O}^2, \ldots, \mathbf{O}^n$ using the pseudorandom point generation algorithm explained in Section 3.*B*.
  For m =1 to $\lfloor \frac{n}{8} \rfloor$
    $\Pi: X_m \to X'_m$, as explained in Section 3.*C*.
    For j =1 to 8
      Compute $\mathbf{C}^j$ by geometric rotation of $\mathbf{P'}^j$ about $\mathbf{O'}^j$, as explained in Section 3.*D*.
      $\mathbf{C}^j \to \mathbf{P}^j$.
    End
  End
End

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

Algorithm 2. Pseudo-code of the decryption algorithm

Input: Cipher-points $\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^n$, content-dependent metadata including r and $\mathbf{G}$, control parameter ψ, and secret key K

Output: Plain-points $\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n$

Generate an expanded key stream using the key scheduling algorithm, as explained in Section 3.*A*.
For *Round* = 2 to 1
  Generate $\mathbf{O}^1, \mathbf{O}^2, \ldots, \mathbf{O}^n$ using the pseudorandom point generation algorithm explained in Section 3.*B*.
  For m =1 to $\left\lfloor \frac{n}{8} \right\rfloor$
    $\Pi: \left\{ \left\{ \mathbf{P}^{j+8(m-1)} \right\}_{j=1}^{8}, \left\{ \mathbf{O}^{j+8(m-1)} \right\}_{j=1}^{8} \right\} \rightarrow \left\{ \left\{ \mathbf{P}'^{j+8(m-1)} \right\}_{j=1}^{8}, \left\{ \mathbf{O}'^{j+8(m-1)} \right\}_{j=1}^{8} \right\}$, as explained in Section 3.*C*. In this step, $\mathbf{O}$ is known and $\mathbf{P}$ is unknown. Using the permutation mapping, the locations, where the dimensional coordinates of $\mathbf{P}$ are mapped into, are determined. Therefore, $\mathbf{P}'$ and $\mathbf{O}'$ are partially determined.
    For j = 1 to 8
      Consider the system of equations generated by the inverse geometric rotation of $\mathbf{C}^{j+8(m-1)}$ about $\mathbf{O}'^{j+8(m-1)}$, and determine the unknown coordinates of $\mathbf{P}'^{j+8(m-1)}$ and $\mathbf{O}'^{j+8(m-1)}$, that are equal to the unknown coordinates of $\mathbf{P}^{j+8(m-1)}$.
      $\mathbf{P}^j \rightarrow \mathbf{C}^j$.
    End
  End
End

IV. Dimensional and Spatial Stability

A principal requirement for 3D content encryption is to ensure the dimensional and spatial stability of the original content. If such consideration is not taken into account, then encrypted objects may exceed the viewing screen resolution or they may collide with other objects of the virtual world. The literature review indicates that previous dimension and space preserving encryption schemes, such as [17], [18] and [19], attempted to provide stability in the form of a bounding box. However, from the cryptographic point of view, providing stability in the form of a bounding box is not an appropriate method for the encryption applications as it discloses the maximum dimensional coordinates of the plain point cloud. Therefore, this paper maintains the stability via bounding spheres, in which the maximum dimensional coordinates of the plain point clouds are not revealed. Given the radius of the bounding sphere, the adversary cannot correctly decompose it into vertex coordinates. Also, the size of the bounding sphere of the encrypted point cloud is adjustable, in that it can be made strictly smaller, but not greater than the size of the bounding sphere of the plain point cloud. This can improve the usability of the encrypted point cloud in 3D applications. In this section, we prove that the proposed encryption scheme maintains the stability of 3D objects. However, before we continue substantiating the stability of our encryption scheme, we firstly establish the notion of dimensional and spatial stability.

To elaborate the stability notion, let $\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n$ be points of the plain point cloud P, and $\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^n$ be points of the corresponding cipher point cloud C. Given a point cloud, denote by $\mathbf{G}$ the barycenter of the point cloud, and by r the distance from the farthest vertex to $\mathbf{G}$. Any point cloud with (r, $\mathbf{G}$) is encapsulated in a bounding sphere of radius r with center $\mathbf{G}$. Therefore, the bounding spheres for the plain point cloud P and cipher point cloud C are characterized by (r_P, $\mathbf{G}_P$) and (r_C, $\mathbf{G}_C$), respectively. To avoid visual inconsistencies and therefore maintain the usability of the point clouds in virtual scenes, the encryption transformation should maintain the cipher-points within the bounding sphere of the plain point cloud; otherwise, the encrypted point cloud may overlap with other objects of the virtual world.

Spatial stability implies that the encrypted point cloud is placed at the same location as the original point cloud. In other words, the center of the bounding sphere of the encrypted point cloud is positioned inside the bounding sphere of the plain point cloud. More precisely, $\|\mathbf{G}_C - \mathbf{G}_P\| \leq r_P$. Dimensional stability implies that the dimensional size of the encrypted point cloud is equal to or smaller than the dimensional size of the plain point cloud. In other words, the radius of the bounding sphere of the encrypted point cloud is less than the radius of the bounding sphere of the plain point cloud. More precisely, $r_C \leq r_P - \|\mathbf{G}_C - \mathbf{G}_P\|$.

The notion of dimensional and spatial stability implies that stability is *inclusive*, that is, the bounding sphere of the encrypted point cloud is included in the bounding sphere of the original point cloud. In addition, if the enclosing sphere of the encrypted point cloud is included in the bounding sphere of the original point cloud, one can easily infer that the requirements for the dimensional and spatial stability are satisfied. Therefore, the dimensional and spatial stability of the cipher point cloud is maintained if and only if the bounding sphere of the encrypted point cloud is included in the bounding sphere of the original point cloud, that is, inclusiveness is equivalent to dimensional and spatial stability.

We use the following lemmas to prove our claim (Theorem1).

Lemma 1. Given n random points $\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n$ within a sphere of radius r with center $\mathbf{G} \in \mathbb{R}^3$, let $\mathbf{P}'^1, \mathbf{P}'^2, \ldots, \mathbf{P}'^n$ be the result of permutation of dimensional coordinates of $\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n$. For any j ($1 \leq j \leq n$), $\|\mathbf{P}'^j - \mathbf{G}\| \leq \sqrt{3}r$.

Proof. Permutation Π: $X \rightarrow X'$ is an injective and surjective mapping that assigns elements of a finite set X of dimensional coordinates of $\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n$ to itself (a finite set X' of dimensional coordinates of $\mathbf{P}'^1, \mathbf{P}'^2, \ldots, \mathbf{P}'^n$). For any j ($1 \leq j \leq n$), $\|\mathbf{P}^j - \mathbf{G}\| \leq r$. Hence, dimensional coordinates of $\mathbf{P}^j$ can have a value between $-r$ to $+r$. If permutation generates 3-tuples with maximal dimensional coordinates, such as ($\pm r$, $\pm r$, $\pm r$), then the maximal distance of $\mathbf{P}'^j$ from $\mathbf{G}$ would be $\sqrt{3}r$. ■

Lemma 2. Given three distinct points $\mathbf{P}^1$, $\mathbf{P}^2$ and $\mathbf{P}^3$ in $\mathbb{R}^3$, let $\mathbf{P}^3$ be the result of rotating $\mathbf{P}^1$ about $\mathbf{P}^2$ with an arbitrary angle.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

If $\mathbf{P}^1$ and $\mathbf{P}^2$ are arbitrary points inside a sphere of radius r with center $\mathbf{G} \in \mathbb{R}^3$, then $\| \mathbf{P}^3 - \mathbf{G}\| \leq 3r$.

Proof. We start the proof by definition of a rotation. A rotation is a distance preserving transformation determined by the rotation center and Euler angles. To let the rotation sweep a larger area, the rotation distance $||\mathbf{P}^1 - \mathbf{P}^2||$ should be maximized. To this end, $\mathbf{P}^1$ and $\mathbf{P}^2$ must be positioned diametrically antipodal on the surface of the sphere. By fixing the rotation distance, the rotation maps $\mathbf{P}^1$ to $\mathbf{P}^3$, which is located on the surface of a larger sphere of radius $2r$ with center $\mathbf{P}^2$. Figure 2(a) depicts the maximal space that $\mathbf{P}^3$ can appear. As shown in the figure, the maximal space is a bounded sphere of radius $2r$ with center $\mathbf{P}^2$. By the triangle inequality, the following relationship is hence true:

$$||\mathbf{P}^3 - \mathbf{G}|| \leq ||\mathbf{P}^2 - \mathbf{G}|| + ||\mathbf{P}^3 - \mathbf{P}^2|| = 3r. \quad (19)$$

■

Theorem 1. Given a plain point cloud $P = \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n\}$ bounded by a sphere of radius r with center $\mathbf{G}$, the proposed encryption scheme maintains the dimensional and spatial stability of the corresponding cipher point cloud $C = \{\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^n\}$.

Proof. To prove this theorem, we need to measure dimensional and spatial deviations of the cipher point cloud from the plain point cloud. The proposed encryption algorithm is a combination of 3 consecutive procedures: a pseudorandom point generation process, a permutation process, and a geometric rotation process. Given n plain-points $\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n$ within a sphere of radius r with center $\mathbf{G}$, the first encryption step generates n pseudorandom points within a sphere S of radius $\sqrt{3}r$ with center $\mathbf{G}$. According to Lemma 1, the permutation process generates a new set of points distributed in a sphere S' of radius $3r$ with center $\mathbf{G}$. By Lemma 2, random geometric rotations disperse the permuted points to a distance no more than thrice the radius of the sphere S′ from the center $\mathbf{G}$. Figure 2(b) depicts the bounded spheres that every point may be mapped into after a random rotation. As shown in the figure, all spheres are located within a bigger sphere S'' whose radius is thrice the radius of the sphere S' and center is $\mathbf{G}$. Thus, the result of the geometric rotation process will be a set of points distributed within the sphere S''. To control the boundary of the encrypted point cloud and therefore to ensure the dimensional stability of the point cloud, a scaling factor is utilized in the geometric rotation process. The result of this analysis is the same for any number of encryption rounds. This proves that given n plain-points $\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^n$ within a sphere of radius r with center $\mathbf{G}$, their cipher-points $\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^n$ will be distributed in a sphere of the same radius with the same center. This meets the requirements for the dimensional and spatial stability, because the bounding sphere of the encrypted point cloud is included in the bounding sphere of the original point cloud. In other words, the bounding sphere of the encrypted point cloud is positioned inside the bounding sphere of the plain point cloud (spatial stability), and the size of the bounding sphere of encrypted point cloud is equal to or smaller than that of the plain point cloud (dimensional stability). ■

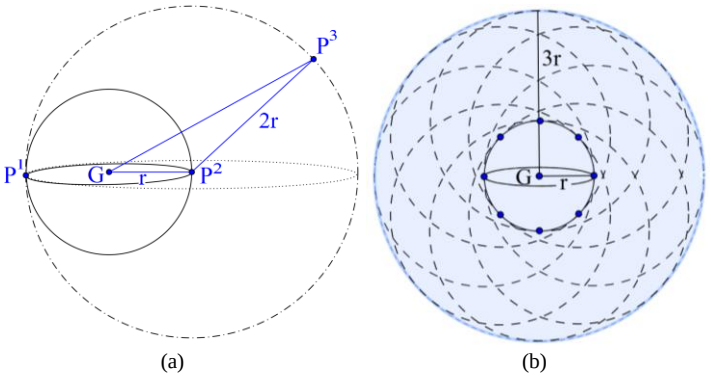

Fig. 2. (a) The maximal space that a random rotation can disperse a point, (b) the boundary of the rotated point cloud.

V. Cryptanalysis

In this section, we evaluate the security of the proposed cipher using cryptanalytic methods. Theoretically, the security level of a cryptosystem is dependent on its key length. In literature, there are various recommendations for the appropriate key length of a particular encryption system [28], [29], [30]. According to the guidelines released by the National Institute for Standards and Technology (NIST) [30], 128 bit key length is an acceptable margin for designing secure symmetric-key encryption algorithms until 2030. The proposed cipher uses six seed points to initiate the Chebyshev map. Considering the double precision 64-bit IEEE 754 format [31], a 64-bit number is represented using 1 bit for the sign bit, 11 bits for the exponent width, and 52 bits for the fraction precision. In this format, the exponent is biased by adding 1023 (= 0b1111111111) before being stored. For any i ($1 \leq i \leq 6$), $|k_i| \leq 1$. Hence, the most significant bit of biased exponent field always remains unchanged (equal to 0). However, the sign and the fraction parts use all their bits. Therefore, the key length of the proposed cipher is 378 (= 6 × 63) bits. Accordingly, the computational complexity of the exhaustive key search (ciphertext-only attack) is 2^{378}. Table I compares the key space of the proposed cipher with a number of well-known 3D object encryption schemes, namely, the schemes by Shi et al. [16] and Technicolor [18], [19]. Compared to Shi et al.'s and Technicolor's encryption schemes, the proposed cipher has a larger key space, which indicates a higher security level against brute-force attacks.

TABLE I
A Comparison of Key Space

Scheme	Key Space Analysis
Shi et al., 2006 [16]	2^{128}
Technicolor, 2012[18] and 2013, [19]	2^{192}
Proposed	2^{378}

To break the cipher, an adversary has temporary access to the encryption and decryption machinery, and so is able to make queries of certain behaviors and observe the corresponding output without the knowledge of the key. This may help the adversary to determine, or at least partially determine, the secret key. In addition, for any i ($1 \leq i \leq 2n$),

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

deducing the expanded keystream, that is, $\{k_1^i, k_2^i, k_3^i, k_4^i, k_5^i, k_6^i\}$, is totally equivalent to finding the secret key whenever different point clouds are encrypted using the same secret key.

Before we continue showing the security of our scheme, we firstly point out why simpler variants on the same idea are insecure, for example, if $\mathcal{E}_K(\mathbf{P}) = \prod_K(\mathbf{P})$, such as encryption schemes of [15], [18] and [19]. Choosing a permutation of large length size can exponentially increase the number of possible permutations $\#(\pi)$ of dimensional coordinates, that is,

$$\#(\pi) = (n!)^{3n}, \tag{20}$$

where n is the number of 3D points. This exponential search space can make the statistical attacks cumbersome by increasing the size of a plain point cloud. However, permutation of a large number of coordinates, for instance, all of the dimensional coordinates at once, may not be an efficient solution because it can take a considerable amount of time to generate a cipher point cloud. In addition, permutation-only schemes are vulnerable to known/chosen plaintext attacks. In practical terms, only one pair of plain/cipher point clouds of n points with non-repeated coordinates is sufficient to uniquely determine the permutation mapping $\prod_K(\cdot)$ of length $3n$. Another example is $\mathcal{E}_K(\mathbf{P}) = \mathcal{R}ot_K(\mathbf{P})$. Given a pair of input/output point clouds ($\mathbf{P}$, $\mathbf{C}$), for any j ($1 \le j \le n$), $\mathbf{C}^j = \mathbf{R}^j \times [\mathbf{P}^j - \mathbf{O}^j] + \mathbf{O}^j$. To determine $\mathbf{R}^j{}_{3\times3}$ and $\mathbf{O}^j{}_{3\times1}$, the adversary needs to solve the following system of equations:

$$\begin{aligned} c_1^j &= r_{11}^j(p_1^j - o_1^j) + r_{12}^j(p_2^j - o_2^j) + r_{13}^j(p_3^j - o_3^j), \\ c_2^j &= r_{21}^j(p_1^j - o_1^j) + r_{22}^j(p_2^j - o_2^j) + r_{23}^j(p_3^j - o_3^j), \\ c_3^j &= r_{31}^j(p_1^j - o_1^j) + r_{32}^j(p_2^j - o_2^j) + r_{33}^j(p_3^j - o_3^j). \end{aligned} \tag{21}$$

For any j ($1 \le j \le n$), the adversary therefore requires only 4 pairs of plain/cipher point clouds to construct a system of 12 nonlinear equations and uniquely determine the unknown matrices, which are $\mathbf{R}^j{}_{3\times3}$ and $\mathbf{O}^j{}_{3\times1}$.

Another variant is a one-round permutation-rotation structure, that is, $\mathcal{E}_K(\mathbf{P}) = \mathcal{R}ot_K(\prod_K(\mathbf{P}, \mathbf{O}))$. To break this variant, for any i ($1 \le i \le 3n$) and j ($1 \le j \le n$), the adversary needs to determine the permutation keystream π_i and the unknown matrices, which are $\mathbf{R}^j{}_{3\times3}$ and $\mathbf{O}^j{}_{3\times1}$. Since permutation of dimensional coordinates is performed within the groups of 8 plain-points and their corresponding pseudorandom points, the adversary needs to observe the 8 consecutive geometric rotations, that is, 8 systems of equations such as equation (21), at once. Due to the permutation process, for l ($1 \le l \le 8$), there are 48! different possible arrangements for the dimensional coordinates of $\mathbf{P}^l$ and $\mathbf{O}^l$. Hence, there are 48! different possibilities for every 8 systems of equations. As explained above, for any j ($1 \le j \le n$), only 4 pairs of plain-points/ cipher-points is sufficient to determine $\mathbf{R}^j{}_{3\times3}$ and $\mathbf{O}^j{}_{3\times1}$. Therefore, the adversary can attack this variant by a known-plaintext attack using 4 pairs of plain/cipher point clouds with the computational complexity of 48! = $2^{202.9}$ encryption. The complexity of this attack is much less than the exhaustive key search, that is 378. To reduce the complexity of this attack, a chosen-plaintext attack can be employed to reduce the search space for determining the permutation keystream. To this end, the adversary can exchange the indices of at least two points of the plain point cloud (point reordering attack [12]) and observe the changes which occur at the cipher point cloud. For any m $(1 \le m < \lfloor \frac{n}{8} \rfloor)$, $\prod_{\{\pi_i\}_{i=48(m-1)+1}^{48(m-1)+48}} : X_m \to X'_m$. In other words, every permutation is performed between dimensional coordinates of 16 points (8 consecutive plain-points and 8 consecutive pseudorandom points) in X_m. Exchanging the indices (positions) of at least two points of the plain point cloud can, at most, affect the coordinates of 6 points of X_m. If these 6 affected points hold different indices, then, in the worst case, every 6 out of 8 points (= 75%) of the cipher point cloud would be affected by geometric rotation. To determine the permutation mapping of whole elements in every X_m, the adversary can repeat the point index shifting 8 times. This reduces the search space of permutation mapping from 48! to 36!. As explained above, only 4 pairs of plain/cipher point clouds are sufficient to determine the geometric rotation. Therefore, this variant is broken by a chosen-plaintext attack with 12 pairs of plain/cipher point clouds and computational complexity of $36! = 2^{138}$ encryption. This attack is more efficient than the known-plaintext attack with complexity 202.9.

Now we analyze the security of the proposed cipher (a two-round permutation-rotation scheme), that is, $\mathcal{E}_K(\mathbf{P}) = \mathcal{R}ot_K(\prod_K(\mathbf{P}, \mathbf{O}))^2$. To break the cipher, for any i ($1 \le i \le 3n$) and j ($1 \le j \le n$), the adversary could employ a similar attack procedure explained for the one-round variant to determine the permutation keystream π_i and the unknown matrices, which are $\mathbf{R}^j{}_{3\times3}$ and $\mathbf{O}^j{}_{3\times1}$. By observing every block of 8 consecutive plain-points separately, there are 8 consecutive geometric rotations of plain-points about pseudorandom points in each block. Mathematically, this is equivalent to 8 systems of 12 non-linear equations. Due to the permutation procedure, these 8 systems of equations can have 48! different arrangements in each round. Therefore, there are $(48!)^2$ different arrangements for every 8 systems of equations after 2 rounds. Also, for any j ($1 \le j \le n$), 4 unknown matrices, which are $\mathbf{R}^j{}_{3\times3}$ and $\mathbf{O}^j{}_{3\times1}$ in the first round, and $\mathbf{R}'^j{}_{3\times3}$, and $\mathbf{O}'^j{}_{3\times1}$ in the second round, need to be determined. As explained above, the adversary needs at least 8 pairs of known plain/cipher point clouds to determine the geometric rotation in both rounds. Therefore, the data complexity of the known-plaintext attack is 8 pairs of plain/cipher point clouds and its computational complexity is $(48!)^2 = 2^{405.8}$ encryption. This attack strategy (known-plaintext attack) is less efficient than the exhaustive key search, and therefore, it is not feasible on the two-round permutation-rotation encryption scheme. To reduce the attack complexity, the adversary may use a point reordering attack (chosen plaintext attack) to reduce the search space for determining the permutation keystream. However, this approach does not work because exchanging the indices of two points of the plain point cloud can change the coordinates of 6 points of X_m by the first round and by the second round, it can change the coordinates of all points of X_m. Hence, the point reordering attack is not feasible on the proposed cipher and compared to the known-plaintext attack, it cannot reduce the search space for determining the permutation keystream. We

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

thus conjecture that the proposed encryption scheme is secure from known-plaintext attacks and chosen-plaintext attacks.

VI. Statistical Analysis

Statistical analysis evaluates the statistical properties of an output stream of a cipher, independent of knowledge of the cipher structure. The failure in statistical tests indicates a bias in the cipher output, and hence, shows that it can be predicted from input. While having good statistical properties alone cannot guarantee the cryptographic security, statistical analyses are a compliment to cryptanalysis. In this section, we perform a number of tests, including a similarity analysis, a plaintext sensitivity analysis, a key sensitivity analysis, and a randomness analysis, to evaluate the statistical properties of the proposed 3D object encryption scheme. The evaluation consisted of both theoretical analysis and practical well-known experimentation. To perform the tests, we have chosen a number of plain point clouds from [32]. We also used different sets of keys to perform the tests: K = (0, 0.2, − 0.5, 0.9, − 0.8, 0.1) as the original key and K' = (0.1, 0.2, − 0.5, 0.9, − 0.8, 0.1) as the slightly modified key.

A. Similarity Analysis

The ability of the adversary to have access to a number of plaintext and ciphertext pairs can help them to perform a similarity analysis between the plaintext and ciphertext, and therefore, obtain additional information about the encryption mapping. This may help the adversary to learn about the plaintext using the ciphertext information, without the knowledge of the secret key. To ensure the security of a 3D content encryption system, cipher objects must leak no information regarding the geometric structure of plain objects. A straightforward similarity analysis is the pairwise Euclidean distance between the points of the point clouds under study. This measure provides the point-wise similarity information of the point clouds. Given a set P of n points $\mathbf{P}^1$, $\mathbf{P}^2$, …, $\mathbf{P}^n$, for any i ($1 \le i \le n$) and j ($1 \le j \le n$), the Euclidean distance (2-norm distance) between $\mathbf{P}^i$ and $\mathbf{P}^j$ is defined as follows [33]:

$$d(\mathbf{P}^i, \mathbf{P}^j) = \left(\sum_{m=1}^{3} |p_m^i - p_m^j|^2\right)^{\frac{1}{2}}. \quad (22)$$

For any i ($1 \le i \le n$), the pairwise Euclidean distance d_p between the corresponding points of two point clouds P and C is

$$d_P^i = d(\mathbf{P}^i, \mathbf{C}^i). \quad (23)$$

Clearly, if this difference is non-zero, then it shows that the encryption procedure scatters the plain-points to different locations. The pairwise Euclidean distance metric is a one-dimensional method and cannot provide any information about the rigid similarity of surfaces (objects). Using such a one-dimensional method for a multi-attribute analysis can lead to making wrong conclusions. Therefore, in addition to the pairwise Euclidean distance, we use the Hausdorff distance [34] to detect similar surfaces with rigid isometries, such as shifting and rotation. Let P and Q be two non-empty subsets representing two rigid surfaces (objects). For any i ($1 \le i \le \#P$) and j ($1 \le j \le \#Q$), the Hausdorff distance is defined as follows [34]:

$$d_H(P, Q) = \max\left\{\sup_{\mathbf{P}^i \in P} \inf_{\mathbf{Q}^j \in Q} d\left(\mathbf{P}^i, \mathbf{Q}^j\right), \sup_{\mathbf{Q}^j \in Q} \inf_{\mathbf{P}^i \in P} d\left(\mathbf{P}^i, \mathbf{Q}^j\right)\right\}, \quad (24)$$

where sup represents the supremum and inf the infimum. The Hausdorff distance determines whether P and Q represent the same rigid surface (object) or not. To measure the dissimilarity between the plain and cipher point clouds, a number of similarity experiments were performed. Figure 3 depicts the results of a similarity analysis for a circular loop of 10000 points with radius 2 and its cipher point cloud. A visual observation indicates that the encryption process pseudo-randomly scatters plain-points into a limited space. It also shows that the encryption result of a circular, ring-shaped point cloud has a different shape (approximately ball-shaped). This observation verifies that the visual information (meaningful pattern) of the plain point cloud is completely damaged and a noisy aspect is observed. The result obtained from the pairwise Euclidean distance between the corresponding points and the Hausdorff distance confirms the visual analysis. As shown in Figure 3(c), all points of the plain point cloud are displaced from their original location. For a better comparison, a sample of the plain point cloud with size 350 is depicted in a 2D plot (Figure 3(d)) along with its corresponding cipher point cloud. Figure 3(d) shows that the Hausdorff distance between the plaintext and ciphertext is the maximum minimum distances

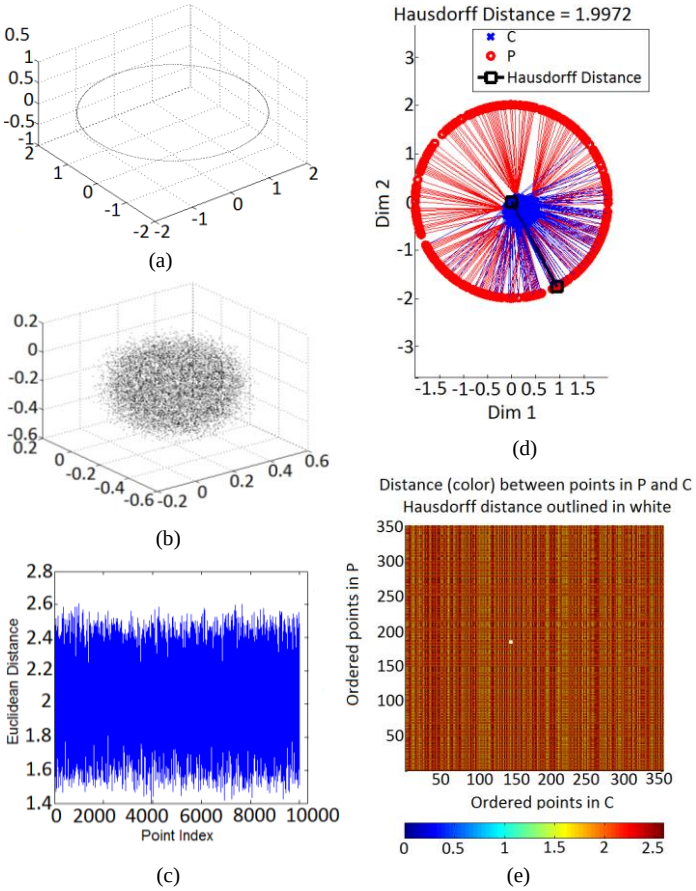

Fig. 3. Similarity analysis: (a) the plain point cloud, (b) the cipher point cloud, (c) the pairwise Euclidean distance between the corresponding points of the plain point cloud and cipher point cloud, (d) the Hausdorff distance between the plain point cloud and cipher point cloud, (e) the heat map of the distance matrix.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

between the plaintext and ciphertext. The Hausdorff distance being non-zero shows that the plaintext and ciphertext clouds are dissimilar. Figure 3(e) depicts the heat map of the distance matrix, where entry (n, m) is the distance of the n-th point in the plain point cloud from the m-th point in the cipher point cloud. The visual summary given by the heat map suggests dissimilarity of the point clouds under study. In comparison with previous dimension and space preserving schemes, such as [15], [17], [18] and [19], our proposal has a better performance with regard to similarity analysis. For instance, given a point cloud of n vertices with zero coordinates, previous schemes ([15], [17], [18] and [19]) render the same output while the proposed scheme generates a completely different point cloud.

B. *Plaintext Sensitivity Analysis*

In general, a desirable property for an encrypted point cloud is being sensitive to minor alternations in the plain point cloud, for instance, modifying only the position of one point. To study the relationship between the plaintext and ciphertext, the adversary may slightly change the position of one point in the point cloud and observe the changes in the encrypted point cloud. By this method, the meaningful relationship between the original point cloud and the encrypted point cloud can be found, which further facilitates in determining the secret key. If a small change in the position of one point in the original point cloud changes the position of a significant number of points in the encrypted point cloud, then the differential attack becomes practically infeasible. In the proposed cipher, each point is rotated randomly with respect to a random rotation reference which is calculated using the point cloud's center of mass. A tiny change in the point cloud's center of mass can affect the result of all random rotations, and therefore, it can result in a completely different cipher point cloud. For instance, if the cryptanalyst changes the position of one point, then the point cloud's center of mass will be changed. This therefore changes the position of other points in the encrypted point cloud. This shows that the proposed cipher is sensitive to changes to the plaintext, and it is robust to differential cryptanalysis. This analysis is also confirmed by our simulation result shown in Figure 4. In this experiment, we have chosen *Michael11*, which contains 52560 points, as the plain point cloud and encrypted it using the original key. We also modified the plain point cloud by displacing 1 point by a distance 0.1% of radius r of the bounding sphere, and encrypted it using the original key. To observe the changes in the encrypted point clouds, scatter plots of the encrypted point clouds are depicted in Figures 4(b) and 4(d). For a better comparison, the pairwise Euclidean distances between the corresponding points of two plain point clouds and two encrypted point clouds are calculated and results are depicted in Figure 5. It can be observed that a small change in the position of even one point in the original point cloud will result in a significant change in the location of cipher-points. In addition to the pairwise Euclidean distance analysis, we also studied the impact of increasing the number of displaced points and the ratio of displacement on the dissimilarity

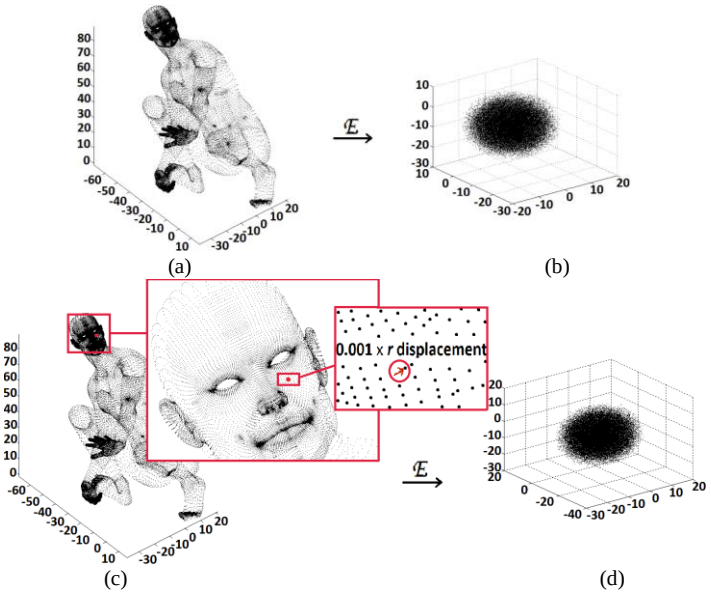

Fig. 4. Plaintext sensitivity test result: (a) original plain point cloud, (b) encrypted point cloud from the original plaintext, (c) slightly changed plain point cloud by displacing 1 point by a distance 0.1% of radius r of the bounding sphere, and (d) encrypted point cloud from the slightly changed plaintext.

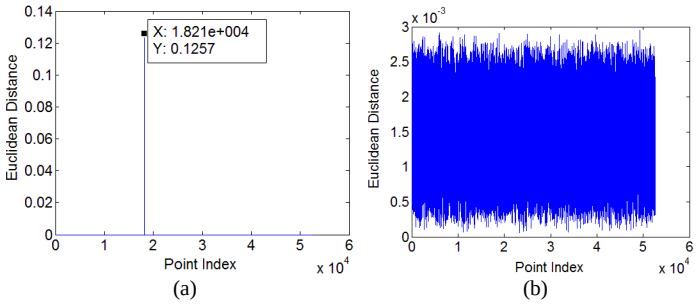

Fig. 5. Pairwise Euclidean distance between the corresponding points of (a) Plain point clouds and (b) Encrypted point clouds.

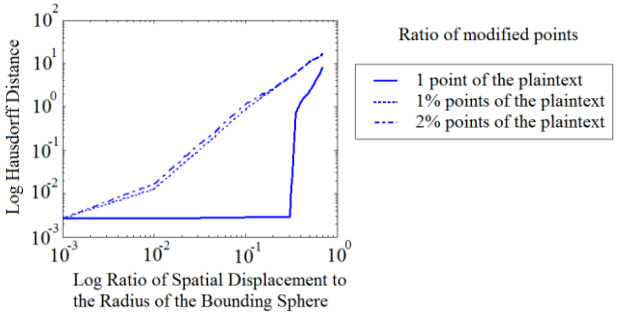

Fig. 6. Log-log plot of the Hausdorff distance against the ratio of the spatial displacement.

between surfaces of the cipher point clouds. To this end, we plotted multiple Hausdorff distance curves (Figure 6) by varying the ratio of modified points in the plain point cloud, that is, 1 point, 1% and 2% of points moved by a distance between 0.1% and 7% of radius r of the cipher object's bounding sphere. The result of this analysis shows that the shape of the cipher point cloud alters by changing the number of altered points and the ratio of spatial displacement. This indicates that the plaintext sensitivity of the proposed scheme is a function of both the ratio of points changed in the plain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

point cloud, and the ratio of displacement. Hence, any plaintext alteration within the plain point cloud not only changes the cipher-points but also changes the shape of the cipher point cloud. This indicates the robustness of our scheme to any differential analysis.

In comparison with previous dimension and space preserving schemes, such as [15], [17], [18] and [19], our proposal is more sensitive with regard to small plaintext alterations. For instance, given *Michael11* as the input point cloud, changing the location of only 1 point will only change one cipher-point in Pan et al.'sencryption scheme [15], and at most 3 cipher-points in Technicolor's encryption scheme [18], [19]. It also has no impact on the rendered bounding box in Phelps's encryption scheme [17]. In addition, such modifications do not change the shape of cipher point clouds by the encryption scheme of [15], [17], [18] and [19], and create similar cipher point clouds with zero Hausdorff distances. As explained in the cryptanalysis section (see Section 5), this can help an adversary to easily track the alterations and deduce the encryption mapping.

C. Key Sensitivity Analysis

A 3D content encryption scheme should be sensitive to changes to the secret key. In other words, a change in a single bit of the secret key should produce a completely different cipher point cloud. To test the key sensitivity of the proposed algorithm, a number of point clouds were encrypted using the original secret key and a slightly modified secret key. As it was not easy to compare the encrypted point clouds by simply observing them, the pairwise Euclidean distance between the corresponding points of two encrypted point clouds were calculated. Figure 7 shows the result of key sensitivity analysis for *Centaur5* [32]. It is observed that two encrypted point clouds with a slightly different key are quite different. This indicates the high sensitivity of the proposed method to changes of the key. In comparison with previous encryption schemes, such as [15], [18] and [19], the proposed encryption scheme is more sensitive to the changes of the secret key. For instance, given a point cloud of n vertices with zero coordinates, changing the secret key creates the same point cloud by the encryption scheme of [15], [18] and [19], while the proposed cipher generates a completely different cipher point cloud.

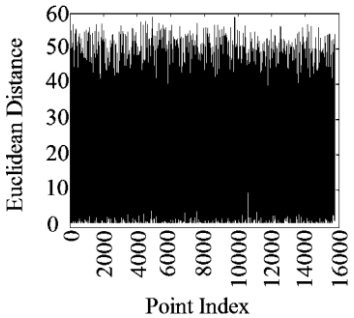

Fig. 7. Key sensitivity test result for *Centaur5* [32].

D. Spatial Randomness

To ensure the security of a 3D encryption system, the 3D cipher must have good probabilistic properties, one of which is random distribution. More specifically, the points of a plain point cloud are required to be dispersed as randomly as possible. This can annihilate any distinguishable patterns or shapes within the original object. This is desirable because the existence of any distinguishable pattern or relationship among the points of a cipher point cloud may lead to data leakage, which may help the adversary partially discover the plain object and hence break the cipher. Any competent adversary would attempt to acquire any knowledge from the point cloud, and therefore, would recognize any hidden pattern to then enable successful reconstruction of the surface. A primary analysis that an adversary may take into consideration is measuring the distance between each pair of points within the ciphertext cloud. The purpose of such analysis would be to cluster any existing pattern inside the ciphertext cloud, such that points in the same cluster have a small distance from one another, while points in different clusters are at a large distance from one another. This may allow partial reconstruction of the surface from an unorganized point cloud. To resist such analysis, the cipher point cloud must be distributed as randomly as possible. Spatial randomness of cipher-points suggests that it would be hard to find any clusters among so many pairs that are all at approximately the same distance.

In a good point cloud encryption scheme, cipher-points are equally likely to occur at any location and the position of any cipher-point is not affected by the position of any other point. In other words, cipher-points occur within a given study volume with no apparent ordering of the distribution. Therefore, in comparison with plain-points, cipher-points are specific point events. This indicates that a good point cloud encryption scheme is synonymous with a homogenous three dimensional Poisson process. In this process, it is easy to verify that the mean and variance of the distance r of a randomly selected point to its nearest neighbor are $0.5540(\rho)^{-\left(\frac{1}{3}\right)}$ and $0.3474(\rho)^{-\left(\frac{2}{3}\right)}$, respectively, where ρ is the expected number of points per unit volume (intensity). For further information about the calculation of the model parameter values, that is, the mean and variance of r, please see [35] and [36].

To investigate the robustness of the proposed encryption scheme to surface reconstruction attacks, we evaluate the spatial randomness of cipher-points using the nearest neighbor method [37]. To this end, we use the Euclidean distance between the nearest neighbors as our statistic to study the spatial distribution of the cipher-point. For any i ($1 \le i \le n$) and j ($1 \le j \le n$), the nearest neighbor distance of point $\mathbf{P}^i$, that is, the distance of point $\mathbf{P}^i$ to its nearest neighbor point in P, is defined as follows:

$$d_i = \min\{d(\mathbf{P}^i, \mathbf{P}^j)\}, \quad (25)$$

where $\mathbf{P}^i$ and $\mathbf{P}^j \in P$, and $i \neq j$. Taking the nearest neighbor distance as the statistic would violate the independence assumption of events in a Poisson process. One can easily observe that for any i ($1 \le i \le n$) and j ($1 \le j \le n$), where $i \neq j$, if $\mathbf{P}^i$ and $\mathbf{P}^j$ are mutual nearest neighbors, then $d_i = d_j$. Thus, $\mathbf{P}^i$ and $\mathbf{P}^j$ are clearly not independent. To resolve this problem, we consider the mean of nearest neighbor distances in a randomly selected subset of a point cloud, as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

$$\bar{d}_j = \frac{1}{j}\sum_{i=1}^{j} d_i. \quad (26)$$

According to the Central Limit Theorem (CLT), under the spatial randomness hypothesis (null hypothesis) and for a sufficiently large sample point cloud with *independently and identically distributed* (*iid*) points, the mean of nearest neighbor distances $\bar{d}_j$ must be approximately normally distributed, with the following mean and variance [38]:

$$\bar{d}_j \sim N\left(0.5540(\rho)^{-\left(\frac{1}{3}\right)}, \frac{0.3474}{j}(\rho)^{-\left(\frac{2}{3}\right)}\right). \quad (27)$$

To construct the spatial randomness test, the sample mean $\bar{d}_j$ needs to be standardized. Under the spatial randomness hypothesis, the standardized sample mean, that is, $Z_j \sim N(0,1)$, is calculated as follows:

$$Z_j = \frac{\bar{d}_j - E(\bar{d}_j)}{\sigma(\bar{d}_j)} = \frac{\bar{d}_j - \left(0.5540\rho^{-\left(\frac{1}{3}\right)}\right)}{\sqrt{\frac{0.3474}{j}\rho^{-\left(\frac{2}{3}\right)}}}. \quad (28)$$

In the standard normal distribution,

$$Z_j \sim N(0,1),\ \Pr(Z \geq z_\alpha) = \alpha. \quad (29)$$

Hence, according to the standardization procedure,

$$\Pr\left(|Z_j| \geq z_{\frac{\alpha}{2}}\right) = \Pr\left[\left(Z_j \leq -z_{\frac{\alpha}{2}}\right) \text{or} \left(z_{\frac{\alpha}{2}} \leq Z_j\right)\right] = \alpha. \quad (30)$$

Equation (30) shows the significance of departure from random expectation. If the null hypothesis is valid, then Z_j should be a sample from $N(0,1)$. The null hypothesis is rejected if and only if $|Z_j| \geq z_{\frac{\alpha}{2}}$. To interpret the test results, the *P*-value can be reported as follows:

$$P - \text{value} = \Pr\left(|Z| \geq z_{\alpha/2}\right) = 2\Phi\left(-\left|z_{\alpha/2}\right|\right), \quad (31)$$

where $\Phi(\cdot)$ denotes the cumulative distribution function. If *P*-value < α, then the spatial randomness hypothesis of the cipher point cloud, that is, the null hypothesis, is rejected. To check the presence of any pattern in the point distribution of the cipher-text point clouds, we have applied the spatial randomness test on the ciphertext result of a number of 3D objects chosen from [32]. According to test results, for all cipher-text point clouds $|Z_j| \leq z_{0.025} = 1.96$ and *P*-value > 0.05, which shows that with the standard significance level of α = 0.05 the spatial randomness is not rejected and the encryption scheme under study passes the statistical test. This indicates a good statistical property of the encryption algorithm which can successfully dissipate any meaningful relationship between the points of the plain point cloud. Figure 8 shows the result of the spatial randomness test on *Michael11*'s cipher point cloud. To draw the histogram of mean nearest neighbor distances, 5000 samples of size j = 100 were selected, and the corresponding *Z*- values were calculated. As shown in the figure, the results of this simulated sampling scheme yield a distribution of *Z*-values that is approximately normal. The mean of this distribution is −1.5258 and its *P*-value is 0.12705. This shows that no clusters can be identified within the cipher point cloud. Results of our analysis for the permutation-only encryption schemes, such as [15], [18] and [19], show that such schemes fail the spatial randomness test and they do not disperse the plain-points randomly into the bounded space.

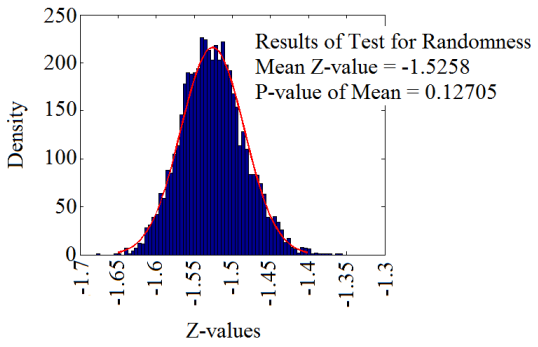

Fig. 8. Sampling distribution of *Z*-values.

VII. Performance Analysis

In addition to security analysis, the encryption performance of 3D content is also an important factor to consider, especially for real-time applications which require a high level of efficiency. Generally, encryption performance hinges upon the structure of the central processing unit, programming language, memory size and the operating system. Since the proposed cipher is a symmetric algorithm, the encryption and decryption performance are the same. In this paper, to evaluate the performance of the proposed cipher, the encryption scheme was implemented using an un-optimized MATLAB code on a machine with Intel Core i5 2.5 GHz processor and 4 GB of installed memory running under Windows 7. In addition, to having an accurate benchmark result, each timing test was executed 10 times and the average time was reported. The results of encryption time for encrypting 100 point clouds of various sizes (10^2 to 10^5 points) are presented in Figure 9. The encryption time is then used to calculate the throughput (encryption speed) of the proposed algorithm, that is, the point cloud size (number of points) divided by the encryption time. This analysis indicates that on average, the proposed cipher encrypts roughly 12196 points per second. Considering the double precision format, each point is stored using 3×64 bits; hence, the average throughput is 292.704 kilobytes per second. The computational complexity of the proposed cipher is the summation of complexities of the key scheduling algorithm and its three components: a pseudorandom point generation, a permutation, and a geometric rotation. The computational complexity of the key scheduling algorithm mainly depends on the implementation of the Chebyshev map. As determined by Brent and Zimmermann [39], the computational complexity of cosine function and its inverse can be calculated from log function and is $O(M(a)\log(a))$, where $M(a)$ is the cost of multiplication and a is the number of digits of precision. As a is determined by the computing system, both a and $M(a)$ are fixed and independent of the input point cloud. Thus, they are constant terms. Accordingly, the computational cost of generating the chaotic keystream for two round encryption is $12M(a)\log(a)\cdot n$. The computational complexity of a pseudorandom point generation is a constant term. Hence, the computational complexity of generating $2n$ points is $d \cdot 2n$, where d is a constant. The computational complexity of each permutation mapping and each geometric rotation is a constant term. As explained in Algorithm 1, the n input points are partitioned into $\left\lfloor \frac{n}{8} \right\rfloor$ subsets of points. The permutation mapping

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

is firstly performed within every set of eight points and their corresponding pseudorandom points, and then all points of each set are geometrically rotated about their corresponding pseudorandom points. In each subset, the computational complexity of local permutations and geometric rotations is $O(1)$. Therefore, the computational complexity of the one-round permutation-rotation process is $O(n)$. Accordingly, the computational complexity of the encryption algorithm is $O(n)$. This complexity estimation is confirmed by our simulation result shown in Figure 9. 3D content protection schemes in [14], [15], [16], [18] and [19], have the same computational complexity as the proposed cipher, that is, $O(n)$, where n is the number of points. However, as discussed earlier in the paper (Section 2 and Section 5), the scheme in [14] is not applicable to the point cloud representation, and schemes in [15], [18] and [19] cannot ensure the security of point cloud vertices.

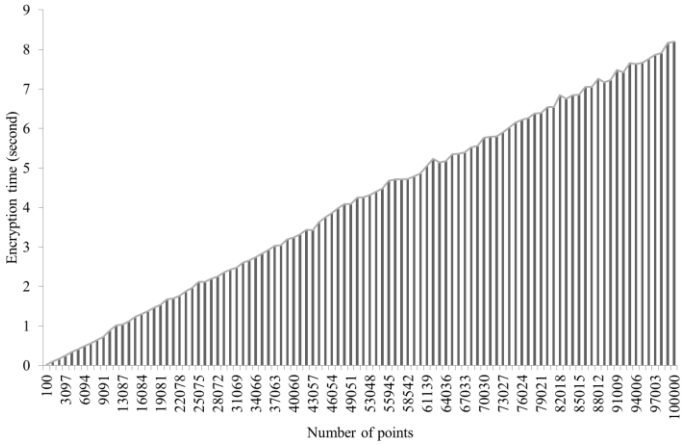

Fig. 9. The encryption time of various sized point clouds.

VIII. Conclusion and Future Work

To overcome the limitations of the current techniques in addressing the confidentiality requirement of 3D objects, this paper proposes a technical solution for encrypting 3D objects. The proposed cipher, which is based on a series of random permutations and rotations, is compatible with standard file formats and maintains the semantic requirements of 3D objects, including the dimensional and spatial stability. Since the inverse of permutation and rotation matrices is their transpose, the implementation of the decryption scheme is very efficient. The cipher displaces the plain-points, and thus, deforms the geometry of the 3D object. This deformation preserves the dimensional and spatial stability of the original object. The security of the proposed cipher was convincingly verified by the evidence obtained from cryptanalytic methods and statistical analyses. The result of rigorous cryptanalysis indicates that the proposed encryption scheme is secure against known-plaintext attacks and chosen-plaintext attacks. The performed statistical tests include similarity analysis, plaintext sensitivity analysis, key sensitivity analysis, and spatial randomness analysis. The result of similarity analysis indicates that the plaintext and ciphertext clouds are not only dissimilar but also have dissimilar surfaces. It is shown that the proposed cipher is sensitive to changes of the plaintext and key, and is robust to differential cryptanalysis. The result of the spatial randomness test indicates that there is no appearance of homogeneous zones in the spatial distribution of the cipher point cloud. This shows that the proposed cipher disperses the plain-points randomly into the bounded space. Results of statistical analyses indicate that the proposed encryption scheme is robust against surface reconstruction attacks. In addition to security analysis, the performance of the proposed cipher was tested, and its efficiency for 3D object encryption was validated. Finally, a comparison with existing protection methods shows that the proposed cipher is more effective and has better security despite having the same level of computational complexity.

The proposed encryption method maintains the location and boundaries of the encrypted content but it is blind to the existence of other nearby objects. Therefore, the plan for future work is to investigate more intelligent methods to control the distortion level of 3D content with respect to its environment. Another direction for further research is to investigate the security of the proposed method against more sophisticated techniques in both cryptography and computer graphics, which address 3D reconstruction of synthetic data under antagonistic conditions. Finally, supplementary methods are being investigated to extend point cloud based encryption methods to other 3D models, while preserving the application requirements of 3D content.

References

[1] United States National Institute of Standards and Technology (NIST), "Announcing the Data Encryption Standard (DES)," Federal Information Processing Standards Publication 46-3, 1999.

[2] United States National Institute of Standards and Technology (NIST), "Announcing the Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication 197, 2001.

[3] B. Zhu, M. Swanson and S. Li, "Encryption and authentication for scalable multimedia: Current state of the art and challenges," *Proceedings of SPIE Internet Multimedia Management Systems V*, vol. 5601, pp. 157-170, 2004.

[4] S. Li, "Multimedia encryption," in *Encyclopedia of Multimedia Technology and Networking*, 2 ed., M. Pagani, Ed., IGI global, 2009, pp. 972-977.

[5] T. Imamura, B. Dillaway and E. Simon, "XML encryption syntax and processing," Technical report, W3C recommendation, 2002.

[6] R. Unnikrishnan, "Statistical approaches to multi-Scale point cloud processing," Doctoral thesis, Carnegie Mellon University, USA, 2008.

[7] R. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1-4, 2011.

[8] V. Morel, S. Orts, M. Cazorla and J. Garcia-Rodriguez, "Geometric 3D point cloud compression," *Patt. Recog. Lett.*, 2014, Article in press, DOI: http://dx.doi.org/10.1016/j.patrec.2014.05.016.

[9] M. Levoy, "The early history of point-based graphics," in *Point-Based Graphics*, M. Gross and H. Pfister, Eds., Burlington, MA: Elsevier, 2007, pp. 9-16.

[10] J. Amigo, L. Kocarev and J. Szczepanski, "Theory and practice of chaotic cryptography," *Phys. Lett. A.* , vol. 366, pp. 211-216, 2007.

[11] S. Li, G. Chen and X. Zheng, "Chaos-based encryption for digital images and videos," in *Multimedia Security Handbook*, B. Furht and D. Kirovski, Eds., Boca Raton, FL, CRC Press, 2004, pp. 133-167.

[12] P. Alface and B. Macq, "From 3D mesh data hiding to 3D shape blind and robust watermarking: A survey," *Trans. Data Hid. Multimedia Sec.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2378146, IEEE Transactions on Information Forensics and Security

2, pp. 91-115, 2007.

[13] W. Kai, G. Lavoue, F. Denis and A. Baskurt, "A comprehensive survey on three-dimensional mesh watermarking," *IEEE Trans. Multim.*, vol. 10, no. 8, pp. 1513-1527, 2008.

[14] D. Koller, M. Turitzin, M. Levoy, M. Tarini, G. Croccia, P. Cignoni and R. Scopigno, "Protected interactive 3D graphics via remote rendering," *ACM Trans. Graph. (TOG)*, vol. 23, no. 3, pp. 695-703, 2004.

[15] Z. Pan, S. Sun, J. Yang and X. Wei, "Protect interactive 3D models via vertex shader programming," *4th International Conference on Entertainment Computing (ICEC)*, no. Sanda, Japan, pp. 59-66, 2005.

[16] W. Shi, H. Lee, R. Yoo and A. Boldyreva, "A digital rights enabled graphics processing system," *The 21st ACM SIGGRAPH, EUROGRAPHICS symposium on Graphics hardware*, pp. 17-26, 2006.

[17] N. Phelps, "Method for exchanging a 3D view between a first and a second user". US Patent 2008/0022408 A1, 24 01 2008.

[18] X. Rolland-Nevière, Y. Maetz, M. Éluard and G. Doerr, "Protecting computer generated 3D graphics," *The security newsletter, Technicolor Security and Content Protection Laboratories*, vol. 21, pp. 8-13, Winter, 2012.

[19] M. Eluard, Y. Maetz and G. Doërr, "Geometry-preserving Encryption for 3D Meshes," *Actes de COmpression et REprésentation des Signaux Audiovisuels*, pp. 7-12, 2013.

[20] G. Haro, "Shape from silhouette consensus," *Patt. Recog.*, vol. 45, no. 9, pp. 3231-3244, 2012.

[21] J.-D. Durou, M. Falcone and M. Sagona, "Numerical methods for shape-from-shading: A new survey with benchmarks," *Comput. Vis. Image Understand.*, vol. 109, no. 1, pp. 22-43, 2008.

[22] C. Poullis, "A framework for automatic modeling from point cloud data," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2563-2575, 2013.

[23] L. Zagorchev and A. Goshtasby, "A curvature-adaptive implicit surface reconstruction for irregularly spaced points," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 9, pp. 1460-1473, 2012.

[24] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656-715, 1949.

[25] N. Liu, "Pseudo-randomness and complexity of binary sequences generated by the chaotic system," *Commun. Nonlin. Sci. Num. Sim.*, vol. 16, pp. 761-768, 2011.

[26] A. Bogdanov, "Analysis and Design of Block Cipher Constructions," Phd thesis, Ruhr University, Bochum, Germany, p. 75, 2009.

[27] T. Cormen, C. Leiserson, R. Rivest and C. Stein, Introduction to Algorithms, 3rd ed., MIT Press, McGraw-Hill, 2009.

[28] A. Lenstra, "Key lengths," in *Handbook of Information Security*, Wiley, 2005, pp. 617-635.

[29] European Network of Excellence in Cryptology II, "Yearly Report on Algorithms and Key Sizes (2010−2011)," 2011.

[30] E. Barker, W. Barker, W. Burr, W. Polk and M. Smid, "NIST Special Publication 800-57. Recommendation for Key Management – Part 1: General (Revision 3)," National Institute of Standards and Technology (NIST), 2011.

[31] IEEE 754, "Standard for Binary Floating-Point Arithmetic".

[32] A. Bronstein, M. Bronstein and R. Kimmel, Numerical Geometry of Non-Rigid Shapes, New York: Springer-Verlag, 2008.

[33] Q. Yong and Y. Jie, "Geodesic distance for support vector machines," *Acta Automatica Sinica*, vol. 31, no. 2, pp. 202-208, 2005.

[34] A. Bronstein, M. Bronstein and R. Kimmel, "Efficient computation of isometry-invariant distances between surfaces," *Siam J. Sci. Comput.*, vol. 28, no. 5, pp. 1812-1836, 2006.

[35] M. Pinsky and S. Karlin, An Introduction to Stochastic Modeling, 4th ed., Academic Press, 2011.

[36] A. Merchán-Pérez, J.-R. Rodríguez, S. González, V. Robles, J. DeFelipe, P. Larrañaga and C. Bielza, "Three-dimensional spatial distribution of synapses in the neocortex: A dual-beam electron microscopy study," *Cereb. Cortex.*, vol. 24, no. 6, pp. 1579-1588, 2014.

[37] P. Diggle, Statistical Analysis of Spatial and Spatio-Temporal Point Patterns, 3rd ed., Boca Raton, Florida, USA: CRC Press, Taylor & Francis, 2013, pp. 60-62.

[38] J. Rice, Mathematical Statistics and Data Analysis, 2nd ed., Duxbury Press, 1995.

[39] R. Brent and P. Zimmermann, "Modern Computer Arithmetic," in *Cambridge Monographs on Computational and Applied Mathematics*, 2010, pp. 135-200.

Alireza Jolfaei is a Ph.D. candidate in Multimedia Security at the Griffith University with research interests primarily in design and development of robust 3D content encryption schemes. His work focuses on investigating innovative solutions for maintaining the dimensional and spatial stability of encrypted content. Alireza received a Bachelor Degree (Hons.) in Biomedical Engineering from Islamic Azad University, Science and Research branch, Tehran, Iran in 2007 and a Master Degree (Hons.) in Telecommunication Engineering from Imam Hossein Comprehensive University, Tehran, Iran in 2010.

Xin-Wen Wu received the Ph.D. degree from the Chinese Academy of Sciences. He was with the Chinese Academy of Sciences, the University of California, San Diego (as a post-doctoral researcher), and the University of Melbourne (as a research fellow). He was affiliated with the School of Information Technology and Mathematical Science, University of Ballarat, Australia. In April 2010, he joined Griffith University, Australia, as a faculty member of the School of Information and Communication Technology. His research interests include cyber and data security, coding techniques, and information theory with applications. He has published over fifty papers and a book in the above-mentioned areas.

Vallipuram Muthukkumarasamy obtained B.Sc. Eng. with 1st Class Hons. from University of Peradeniya, Sri Lanka and obtained Ph.D. from Cambridge University, England. He is currently attached to the School of Information and Communication Technology, Griffith University, Australia as Senior Lecturer. His current research areas include investigation of security issues in wireless networks, sensor networks, trust management in MANETs, key establishment protocols and medical sensor networks. He is currently leading the Network Security research Group at the Institute for Integrated and Intelligent Systems at Griffith University. He has also received a number of best teacher awards.