Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-018-3354-4

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

# Homomorphic Encryption-Based Reversible Data Hiding for 3D Mesh Models

Mohsin Shah[1] · Weiming Zhang[1] · Honggang Hu[1] · Hang Zhou[1] · Toqeer Mahmood[2]

Received: 22 February 2018 / Accepted: 27 May 2018


**Abstract**
Reversible data hiding in the encrypted domain (RDH-ED) reversibly encode and decode information to an encrypted cover medium without decrypting it. With the rapid development of multimedia applications, 3D models are considered as potential cover media for reversible data hiding due to their intrinsic capacity and potential applications in various areas such as military and medicine. In this paper, we propose a two-tier RDH-ED framework for 3D mesh models using the homomorphic Paillier cryptosystem. Two homomorphic properties of the underlying cryptosystem are utilized to propose a two-tier RDH-ED framework for end-to-end authentication and cloud data management. The proposed framework is successfully implemented on various simple and dense meshes. The performance evaluation of the proposed framework shows high embedding rates. Furthermore, it produces high-quality directly decrypted meshes from which information bits are extracted error-free and the original meshes are recovered losslessly.

**Keywords** 3D mesh models · Encrypted domain · Homomorphic encryption · Paillier cryptosystem · Reversible data hiding

## 1 Introduction

Data hiding methods have received significant attention from the researchers in the last few decades. Various data hiding techniques have been reported in the literature for hiding secret information in a cover medium [1,2]. The classical information hiding methods distort the cover medium severely during the process of hiding the requisite information. This drawback of classical information hiding methods makes them ineffective for sensitive application fields such as military, medicine, remote sensing, and law enforcement, where lossless recovery of the cover medium is also required. Researchers have proposed reversible data hiding (RDH) methods in recent years to address the lossless recovery of cover medium [3–6].

With the rapid development of online storage technologies and swift growth of the Internet, the demand for outsourcing multimedia to the cloud is increasing. Cloud computing provides an infrastructure for storing multimedia and data manipulation. Nowadays, different kinds of multimedia are usually stored in the cloud such as audio, video, images and 3D models. To manage the outsourced multimedia, the cloud server embeds additional information into the multimedia and uses this information for various purposes such as data management, ownership identification and models integrity. However, the cloud server is not allowed to make permanent changes to the outsourced multimedia during information embedding. In this context, RDH methods are used to recover original outsourced multimedia and information extraction.

The classic RDH methods can be classified into three major groups: difference expansion (DE) [3], histogram shifting (HS) [4] and lossless compression [6]. The DE methods embed data into the cover medium by expanding the difference of neighboring pixel values. DE has been extensively investigated and extended to integer transformation (IT) [7–10] and prediction-error expansion(PE) [11,12]. In HS methods, a histogram of the cover image is first generated and then data are embedded based on the zero or minimum points of the histogram. Lossless compression-based methods compress certain portions of the original image that are expected to produce distortion at data embedding and transmit these compressed portions as part of embedding payload. The work proposed in [13] provided equivalency between

✉ Weiming Zhang
zhangwm@ustc.edu.cn

[1] School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China

[2] Department of Computer Science, University of Engineering and Technology Taxila, Taxila 47050, Pakistan

Published online: 08 June 2018

reversible data hiding and lossless compression by estimating the optimal modification probability [14,15].

The demand for protecting the uploaded contents for outsourced storage of cloud service is increasing because multimedia content can be easily copied, manipulated and distributed. The leakage of private photographs of Hollywood actresses from iCloud is an example of the vulnerability of outsourced data to possible attacks [16]. Conventional encryption algorithms are widely used for protecting multimedia content. The multimedia content is encrypted prior to outsourcing. However, the encrypted content requires decryption before any processing/embedding is conducted in the cloud. This scenario is applicable when encryption guarantees the protection of multimedia content against a third party (intruder) while the involved parties (user and the cloud server) trust one another. However, there can be situations in which the involved parties do not trust each other. Let us consider the example of a patient who is under continuous cloud-based health monitoring service in order to get a pre-alert diagnosis for staying healthy. The patient may not trust the service provider that will analyze his/her data and the service provider may not disclose its processing algorithm as it is the basis for its business. It is evident that the encryption of outsourced multimedia contents and its processing in the encrypted domain is required. This triggers the cloud server to use RDH in the encrypted domain (RDH-ED) for managing encrypted outsourced multimedia. Based on the embedding mechanisms, the currently available RDH-ED methods can be categorized into two broad categories: (1) vacating room before encryption (VRBE) in which room for information bits is created before encrypting the cover image [17–19]; and (2) vacating room after encryption (VRAE) in which information embedding is performed after encrypting the cover image [20–24].

The VRBE- and VRAE-based RDH-ED methods [17–24] used private key cryptosystem for image encryption, some RDH-ED methods with homomorphic public key cryptosystem are proposed in [25–29]. In [25], each pixel of the original image is divided into two parts (even integer and bit part) and encrypted both the parts with Paillier cryptosystem. The authors achieved information embedding in encrypted image through the modification of magnitude relationship of bit parts of two neighboring pixels. However, the method proposed in [25] faces pixel overflow problem. In [26], an improved method of [25] is proposed in order to avoid the pixel overflow problem. The authors divided each pixel in the original image into three parts using energy transfer equation and information is embedded by manipulating these energy parts. In [27], Wu et al. proposed two RDH-ED methods using the homomorphic and probabilistic properties of the Paillier cryptosystem for different application scenarios in cloud platform. Zhang et al. [28] proposed a separable RDH-ED with public key cryptosystem using histogram shrinking and Wet Paper Code (WPC). Xiang and Luo [29] proposed separable RDH-ED with public key cryptosystem which embeds data into encrypted image by shifting histogram of the absolute difference of two neighboring pixels.

The research of RDH for digital images has received tremendous attention from the research community and numerous methods have been presented so far. However, RDH for other types of media contents such as audio, video, and 3D models is still in its emerging stage. The rapid development of 3D multimedia applications and its large intrinsic capacity are the provoking factors for researchers to adopt 3D models as the cover medium for RDH. The existing RDH methods for 3D models can be classified into four categories: (1) spatial domain, (2) transform domain, (3) compressed domain, and (4) encrypted domain. Spatial domain RDH methods are low complexity methods which slightly modify vertex positions for data embedding [30–32]. Transform domain RDH methods [33,34] embed data into the coefficients of the transformed 3D model. Compressed domain RDH methods [35,36] use vector quantization (VQ) for compressing the vertices of 3D models and then embed data into compressed mesh stream. The method based on the encrypted domain RDH [16] encrypts the vertices of the cover mesh and embed information bits in the encrypted vertices. The drawback of this method [16] is that it is based on conventional encryption scheme and decryption is required for information extraction at the cloud. Such framework can be used for authentication by a user who has the decryption key, but it cannot be used for data management in the cloud platforms.

To the best of our knowledge and the literature review, RDH for 3D mesh models in the homomorphic-encrypted domain is not studied yet. The presented study proposes a two-tier RDH-ED framework for 3D meshes using the homomorphic Paillier cryptosystem. The proposed two-tier framework provides a solution for end-to-end authentication and cloud data management in the encrypted domain. The proposed framework starts with mapping the signed float values of vertex coordinates of a 3D mesh to positive integers, and thereafter, encrypt these integer coordinates by the homomorphic Paillier cryptosystem. The sender embeds authentication information into the encrypted vertex coordinates through the first-tier method and then the cloud server embeds additional information through the second-tier method. The information embedded by the cloud server can be extracted without decrypting the mesh while the information embedded by the sender can only be extracted with the help of a decryption key. The main contributions of this work are summarized as follows:

1. Solution for end-to-end authentication and cloud data management in the encrypted domain.
2. Both operations of embedding by the sender and cloud server (first-tier and second-tier) are performed in the

encrypted domain. The second-tier scheme can directly extract information from the encrypted mesh which is useful for privacy preserving cloud services.
3. Data are not expanded by both RDH-ED methods.

The remainder of the paper is organized as follows. In Sect. 2, a potential application of the proposed framework will be discussed. Section 3 is devoted to the proposed framework. Section 4 presents the experimental results. Finally, the conclusion is reported in Sect. 5.

## 2 Potential Application of the Proposed Framework

In this section, a potential application of the proposed framework is discussed. In the field of radiography, 3D models, generated from magnetic resonance Imaging (MRI) or Computer Tomography (CT) scan, can be shared through the cloud. The process of information embedding and extraction for a 3D MRI model is illustrated in Fig. 1. While a 3D MRI model can be encrypted for protecting patient's privacy by a doctor, a network administrator may embed authentication information into the encrypted model for cover authentication at the receiver side. While embedding authentication information, the network administrator cannot access the original content of the encrypted model. It is worth noting to mention here that embedding authentication information at this stage is important for authenticating the sender at the recipient side. At this stage, an attacker can use the recipient's public key to encrypt malicious model and embed false information. The network administrator uploads the marked encrypted model to the cloud. The cloud server embeds additional information into the marked encrypted model without knowing the original content of the encrypted model. Information embedded by the server could be used for cloud data management. The cloud server extracts the embedded information in the encrypted domain without distorting the model and sends the model to a legitimate recipient who uses the decryption key to decrypt the model before extracting the hidden information. Having the decrypted model, the recipient extracts the authentication information to first authenticate the cover model and then recovers the model to the original one. During the whole process, the model remains private between the doctor and the recipient without exposing it to the network administrator and the cloud server.

## 3 Proposed Framework

This section presents the proposed two-tier RDH-ED framework for 3D mesh models. The proposed framework is based on different key phases that include: preprocessing, encryption, data embedding, data extraction and mesh recovery. Figure 1 illustrates the key phases of the proposed framework. After preprocessing, the cover mesh is encrypted to generate $E(M)$ with subsequent conduction of data embedding by the sender and the cloud, generating $E(M)_w$ and $E'$, respectively. After the cloud server extracts its data in the encrypted domain, the recipient directly decrypts the mesh to $M_w$, recovering the original mesh $M$ followed by extracting the hidden information. The implementation details of the proposed framework are discussed in the subsequent sections.

### 3.1 Preprocessing

A 3D model represents 3D objects by various geometric entities such as vertices, edges, faces, polygons, surfaces, and triangles. A 3D object can be represented by polygon

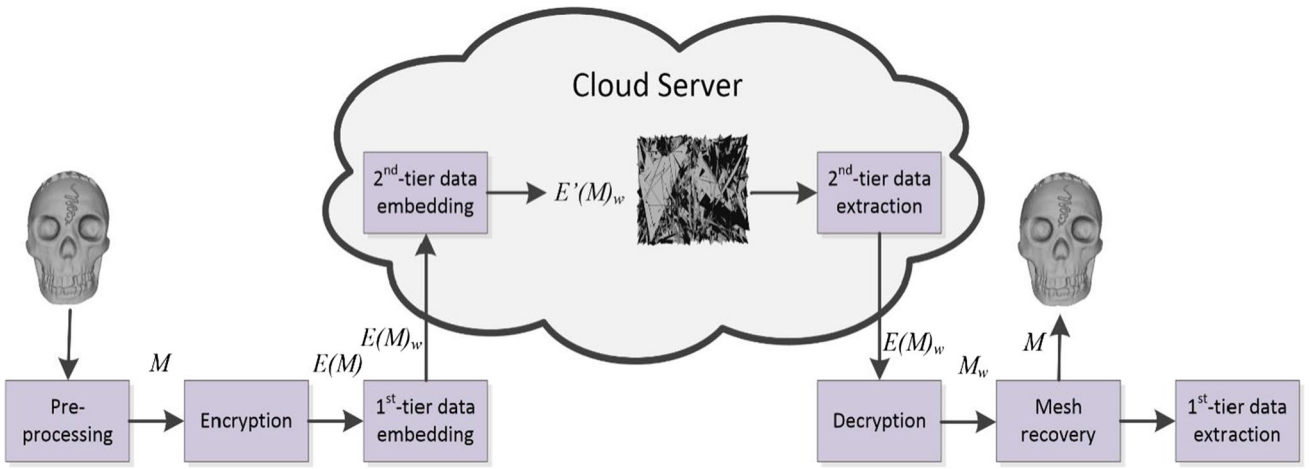


**Fig. 1** Framework of the proposed two-tier RDH-ED

meshes, point clouds, and solid models. Among these three representations, polygon mesh is most commonly used in signal processing, thus we chose polygon meshes as the cover models for the framework proposed in this paper. A polygon mesh consists of two sets of parameters: vertices $V = \{v_1, v_2, \ldots, v_N\}$ and faces $F = \{f_1, f_2, \ldots, f_N\}$. The set $V$ defines the shape of the mesh in three-dimensional space $\Re^3$ and the set $F$ defines the connectivity as described in various files formats $(.off, obj, .ply, .x3d, .vrml)$.

Homomorphic cryptosystem cannot process floating point and fixed point numbers because of their complexity involved in simple mathematical operations such as addition and multiplication [37]. In order to encrypt the set of vertices of a 3D mesh by the Paillier cryptosystem, all vertices should be transformed into integers. Normally, uncompressed vertices of a mesh are specified as 32-bit floating point numbers with a precision of 6 digits. This level of precision is not required by most of the applications, vertex data can be represented by a lossy compressed set as suggested by Deering [38]. Deering normalized the position coordinates in an axis-aligned bounding box and then quantized the coordinates to $k$ bits of precision so that each coordinate can be represented as integers between 0 and $2^k - 1$, where $k \in (1 - 33)$. The quantization of floating point vertices into integer vertices is formulated by Eq. (1).

$$v'_{i,j} = \lfloor v_{i,j} \cdot 10^k \rfloor \tag{1}$$

where $j = x, y, z$, $i$ is the ith vertex, $v_{i,j}$ is the original set of floating point vertices and $v'_{i,j}$ is the new set of integer vertices. After encryption and data embedding, processed mesthe h can be generated by mapping the integer coordinates back to the floating point coordinates using Eq. (2).

$$\bar{v}_{i,j} = v'_{i,j}/10^k \tag{2}$$

Eq. (1) and (2) shows that $k$ is an important parameter in determining whether a mesh can be losslessly recovered or not. The value of $k$ also determines the bit-length of integer coordinates as given in Eq. (3).

$$\text{bit length} = \begin{cases} 8, & 1 \le k \le 2 \\ 16, & 3 \le k \le 4 \\ 32 & 5 \le k \le 9 \\ 64 & 10 \le k \le 33 \end{cases} \tag{3}$$

The message space of Paillier cryptosystem is a set of positive integers $Z_N$. To put it another way, it means that Paillier cryptosystem cannot process negative integers. Therefore, the negative integer vertex coordinates are mapped to positive integer coordinates for further processing by the Paillier cryptosystem. The pseudo-codes for forward and reverse mappings are given in Algorithm 1 and 2, respectively. In both algorithms, $v''_{i,j}$ is the notation used for the transformed integer coordinates and $N$ is the modulus of the cryptosystem. The only requirement for this mapping is that $v'_{i,j}$ should be in the range of $-N/2$ and $N/2$, otherwise reverse mapping will lead to incorrect result. With this mapping, no side information about the sign of the coordinates is transmitted with the hidden data as auxiliary information.

**Algorithm 1** Forward mapping

1: **if** $v'_{i,j} \ge 0$ **do**
2: $\quad v''_{i,j} = v'_{i,j}$
3: **else do**
4: $\quad v''_{i,j} = v'_{i,j} + N$
5: **end if**

**Algorithm 2** Reverse mapping

1: **if** $v''_{i,j} \le N/2$ **do**
2: $\quad v'_{i,j} = v''_{i,j}$
3: **else do**
4: $\quad v'_{i,j} = v''_{i,j} - N$
5: **end if**

### 3.2 Homomorphic Paillier Cryptosystem

Homomorphic cryptosystem is used in the field of secure signal processing as it translates mathematical operations in the encrypted domain to corresponding operations in the plain domain. The idea that certain mathematical operations can be conducted on the encrypted data without decrypting it was first proposed by Rivest et al. [39]. Subsequently, fully homomorphic cryptosystems [40] and partial homomorphic cryptosystems [41–43] are proposed. Homomorphic cryptosystems can be additive or multiplicative homomorphic based on the corresponding operations in the encrypted and plain domain. Moreover, homomorphic cryptosystems are probabilistic whereas conventional cryptosystems are deterministic. Given a plain-text, a deterministic cryptosystem will always produce the same cipher-text output for the same key, making it easy for the attackers to determine the frequency of plain-text encrypted to cipher-text. The idea of probabilistic cryptosystem has been presented in [44]. In a probabilistic cryptosystem, the cipher-text is a function of both plain-text and a random integer which changes for every plain-text value. Consequently, encrypting a plain-text value twice will produce different cipher-text values.

In this paper, we use Paillier cryptosystem for encryption because it possesses both probabilistic and homomorphic properties. Paillier cryptosystem is a homomorphic modular public key encryption method which is based on the computational hardness to decide whether a number is an $N$th residue modulo $N^2$. Let $M$ is the set of plain-texts, $C$ is the set of cipher-texts, $E$ is the notation for encryption, and $D$ is the notation for decryption, then according to the additive homomorphic property we have:

$$\begin{aligned} E(m_1) \cdot E(m_2) &= \left(g^{m_1} \cdot r_1^N \mod N^2\right) \\ &\quad \cdot \left(g^{m_2} \cdot r_2^N \mod N^2\right) \\ E(m_1) \cdot E(m_2) &= g^{m_1+m_2} (r_1 \cdot r_2)^N \mod N^2 \\ E(m_1) \cdot E(m_2) &= E(m_1 + m_2) \end{aligned} \tag{4}$$

and

$$\begin{aligned} E(m)^a &= \left(g^m \cdot r^N \mod N^2\right)^a \\ E(m)^a &= \left(g^m \cdot r^N\right)^a \mod N^2 \\ E(m)^a &= (g^{am} \cdot r^a)^N \mod N^2 \\ E(m)^a &= E(a \cdot m) \end{aligned} \tag{5}$$

where, $m_1$ and $m_2$ are two plain-texts, $a$ is an integer and $r_1, r_2$ are integers randomly selected form $Z_N^*$ where $Z_N^*$ is the set of integers relatively prime to $N$. Paillier cryptosystem is different from other cryptographic systems because it can encrypt a plain-text to many different cipher-texts. In other words, the cipher-text value for a certain plain-text is not unique. This property is termed as self-blinding, which is mathematically formulated in Eq. (6).

$$D\left[\left\{E(m) \cdot P^N \mod N\right\} \mod N^2\right] = m \mod N \tag{6}$$

where $P$ is a random integer relatively prime to $N$. The detailed description of Paillier cryptosystem and its mathematical formulation can be found in [41]. In this paper, only key generation, encryption and decryption of the cryptosystem are discussed.

#### 3.2.1 Key Generation Phase

Let $Z_N$ is the set of integers modulo $N$ and $Z_N^*$ is the set of integers relatively prime to $N$. $p$ and $q$ are two large prime numbers, and $N = p \times q$. Now select $g \in Z_{N^2}^*$ such that $\gcd(g, N) = 1$. Usually, $g = N + 1$ is considered a good choice. The set $(g, N)$ is the public key. At the receiver side, given the values of $p$, $q$, the receiver first computes $\lambda = \text{lcm}(p-1, q-1)$ and then calculates $k = L\left(g^\lambda \mod N^2\right)$, where $L(x) = (x-1)/N$. If $\gcd(k, N) = 1$ then the multiplicative inverse of $k$ is computed using Eq. (7).

$$\mu = k^{-1} \mod N \tag{7}$$

where $(\mu, N)$ is defined as the private key.

#### 3.2.2 Encryption Phase

Let $m \in M$ and $m < N$. Then encryption of $m$ is formulated by Eq. (8).

$$c = E(m, r) = g^m \cdot r^N \mod N^2 \tag{8}$$

where $r$ is integer randomly selected from $Z_N^*$ and $c$ is the cipher-text.

#### 3.2.3 Decryption Phase

Let $c < N^2$, then the original plain-text $m$ is computed using Eq. (9).

$$m = L\left(c^\lambda \mod N^2\right) \cdot \mu \mod N \tag{9}$$

For $i$th integral vertex coordinates of a mesh $\left\{v''_{i,x}, v''_{i,y}, v''_{i,z}\right\} \in Z_N$, encryption is given using Eq. (10).

$$c_{i,j} = g^{v''_{i,j}} \cdot r_{i,j}^N \mod N^2 \tag{10}$$

For the ith encrypted coordinates $\{c_{ix}, c_{iy}, c_{iz}\}$, decryption is given by Eq. (11).

$$v''_{i,j} = L\left(c_{i,j}^\lambda \mod N^2\right) \cdot \mu \mod N \tag{11}$$

### 3.3 Data Embedding

Data embedding is performed on both the sender and cloud sides. The sender embeds authentication information into the encrypted vertex coordinates through the first-tier method and then the cloud server embeds additional information through the second-tier method.

#### 3.3.1 First-Tier Data Embedding

First-tier embedding method uses histogram expansion and shifting for embedding information into the mesh vertices. If $k = 4$ and bit-length $= 16$ in case of a vertex, then each coordinate of the vertex is expanded from $[0, 65, 535]$ to $[0, 131, 070]$ using Eq. (5). This corresponds to take the square of the encrypted values of each coordinate. In plain domain, the expanded histograms are shifted by adding vertex coordinates and the bits of the information to be embedded. In encrypted domain, the histograms are shifted by multiplication of encrypted coordinate values with the encrypted values of bits to be embedded as given in Eq. (4). As the histogram is shifted along the $x$, $y$, and $z$ axis, three bits per vertex are embedded. The histogram expansion and

shifting in the encrypted domain can be combined into one expression as shown in Eq. (12).

$$c'_{i,j} = \left(c^2_{i,j} \mod N^2 \cdot g^{b^1_{i,j}}\right) \mod N^2 \tag{12}$$

where $c'_{i,j}$ is the encrypted value after data embedding, $i = 1, 2, \ldots, M$, $M$ is the number of vertices in the cover 3D mesh and $b^1_{i,j}$ being the 1st-tier embedding bits. The histogram expansion operation expands the range of vertex coordinates from maximum 65,535 (16 bits) to 131,070 to (17 bits). However, the marked encrypted data size would not expand due to mod $N^2$ operation in Eq. (12). In other words, data size of $c'_{i,j}$ and $c_{i,j}$ is equal.

For the correct recovery of the original vertex coordinates and extraction of the embedded bits, $N$ is selected sufficiently large to satisfy Eq. (13).

$$\left(2v_{i,j} + b^1_{i,j}\right) \mod N = 2v_{i,j} + b^1_{i,j} \tag{13}$$

After embedding information into the encrypted 3D mesh, the marked encrypted mesh is outsourced to the cloud.

#### 3.3.2 Second-Tier Data Embedding

Having the information-embedded encrypted mesh, the cloud server can embed additional information using the self-blinding property of Paillier cryptosystem. The self-blinding property allows the cloud server to modify the marked encrypted vertex coordinates without affecting the corresponding plain-text value. Based on the self-blinding property, the cloud server embeds the $b^2_{i,j}$ bits by modifying $c'_{i,j}$ as given in Eq. (14).

$$c''_{i,j} = c'_{i,j} \cdot P^N_{i,j} \mod N^2 \tag{14}$$

where $P_{i,j}$ is a random integer, relatively prime to $N$ and selected to satisfy Eq. (15).

$$\left(c'_{i,j} \cdot P^N_{i,j} \mod N^2\right) \mod 2 = b^2_{i,j} \tag{15}$$

where $b^2_{i,j}$ are the second-tier embedding bits. The coordinates of all the encrypted vertices of a 3D mesh are scanned. The encrypted coordinates will not change when $c'_{i,j} \mod 2 = b^2_{i,j}$, otherwise Eq. (15) is applied to make $c'_{i,j}$ equal to $b^2_{i,j}$.This application is an iterative process and different random integers from $Z^*_N$ can be selected to satisfy Eq. (15).

### 3.4 Data Extraction and Mesh Recovery

The information embedded by the cloud server can be extracted without decrypting the mesh. Whereas, the information embedded by the sender can only be extracted with the help of a decryption key.

#### 3.4.1 Second-Tier Data Extraction

As mentioned in Sect. 3.3.2, the marked encrypted mesh at the cloud server contains both the information embedded by the sender and the management information of the cloud. When a legitimate recipient wants to retrieve the mesh, the cloud server extracts its management information without decrypting the mesh and shares the mesh with the recipient. The legitimate recipient can use his/her decryption key to first decrypt the mesh and then extract the hidden information.

Since the application of Eq. (15) does not change the corresponding plain vertex values, there is no need to recover the original vertex $c'_{i,j}$ coordinate. The cloud server can extract the embedded information from the modified encrypted coordinates using Eq. (16).

$$b^2_{i,j} = c''_{i,j} \mod 2 \tag{16}$$

After extracting the management information from the marked encrypted mesh, the cloud server sends the mesh to the recipient.

#### 3.4.2 First-Tier Data Extraction and Mesh Recovery

In order to obtain the original vertex coordinates $\{\bar{v}_{i,x}, \bar{v}_{i,y}, \bar{v}_{i,z}\}$ and extract the embedded bits at the recipient side, the marked encrypted mesh is decrypted using Eq. (17).

$$D\left(c''_{i,j}\right) = \left(2v''_{i,j} + b^1_{i,j}\right) \mod N \tag{17}$$

where $v''_{i,j}$ are the positive integer vertex coordinates which can be computed by Eq. (18).

$$v''_{i,j} = \left\lfloor \frac{D\left(c''_{i,j}\right)}{2} \right\rfloor \tag{18}$$

The embedded data are extracted using Eq. (19).

$$b^1_{i,j} = D\left(c''_{i,j}\right) - 2v''_{i,j} \tag{19}$$

The positive integer vertex coordinates $v''_{i,j}$ are mapped back to the signed integer coordinates $v'_{i,j}$ using Algorithm 2 and Eq. (2) is further used to acquire the original float vertex coordinates $\bar{v}_{i,j}$.

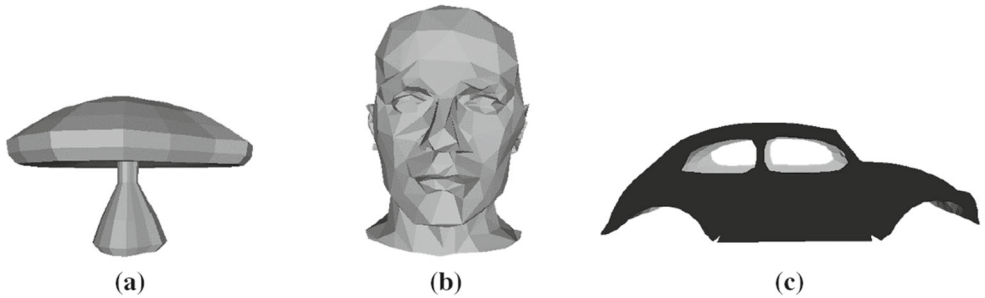


**Fig. 2** Original meshes. **a** Mushroom, **b** Mannequin, **c** Beetle

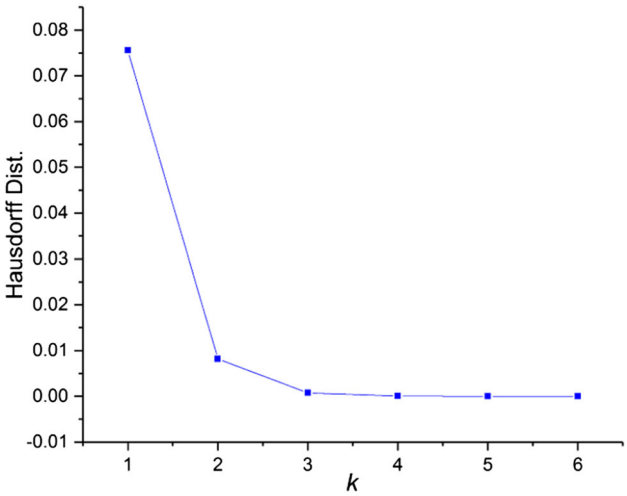


**Fig. 3** Hausdorff distance between the recovered and the original mesh for varying $k$

## 4 Experimental Results and Discussion

The proposed framework is implemented in the experimental environment of MATLAB R2016b under Windows 8.1 Pro. The system configurations are Intel(R) Core i7-7700 CPU 3.60 GHz and RAM 8-GB. To test the performance and effectiveness of the proposed framework, 1,814 mesh models with '$.off$' format from *The Princeton Shape Retrieval and Analysis Group*[1] are considered. Three of these meshes are shown in Fig. 2. The information bits are generated randomly using the *randi* function of MATLAB.

As mentioned in Sect. 3.1, the quality of the recovered mesh is determined by the parameter $k$. In order to investigate the effect of $k$, we encrypt the Mushroom mesh model for the value of $k$ from 1 to 6 and compute the Hausdorff distance between the recovered and original mesh. Figure 3 illustrates that the Hausdorff distance between the recovered mesh and the original mesh decreases with increase in the value of $k$. In [16], it is demonstrated that the value of $k$ is a trade-off between the quality of the recovered mesh and the computational overhead of the process. Therefore, we choose $k = 4$ for experiments in this work which establishes a balance between the quality and the computation overhead of the process. Figure 4 shows experimental results, demonstrating the visual effects of the host meshes at different stages of the proposed framework.

The proposed framework was further tested on dense mesh models, comprising thousands of vertices and triangles. Several dense meshes with '$.ply$' format from *The Standard 3D Scanning Repository*[2] are processed. Two of these are shown in Fig. 5. It is observed that the proposed framework is also applicable to sophisticated dense meshes.

### 4.1 Performance Evaluation

Embedding rate and quality of the directly decrypted meshes are used to evaluate the performance of the proposed framework.

#### 4.1.1 Embedding Capacity

Embedding rate is measured in bits per vertex (bpv) which is a ratio between the total number of embedded bits and the total number of vertices. Table 1 illustrates embedding rate and embedding capacity for the test meshes. Therein, embedding rate is the combined rate of both methods (1st-tier and 2nd-tier). The proposed framework achieves embedding rate of 3 bpv using the first-tier method. In order to keep the Paillier encryption secure against modern factoring methods, $N$ is chosen large enough (1024 bits). The encrypted value of each coordinate of the original mesh is expanded to 1024 bits. As mentioned earlier, the value of $k = 4$ is selected, which corresponds to bit-length of 16 for representing the integer coordinate values. If the vertex coordinate value is represented by a bit-length of 16, then one encrypted coordinate value can carry $1024 - 16 = 1008$ bits. Similarly, the information hiding rate of the second-tier method can be increased by repeatedly applying Eq. (15). Figure 6 shows embedding rate of 3 bpv and 6 bpv using the first-tier method.

[1] http://shape.cs.princeton.edu/benchmark/index.cgi.

[2] http://graphics.stanford.edu/data/3Dscanrep/.

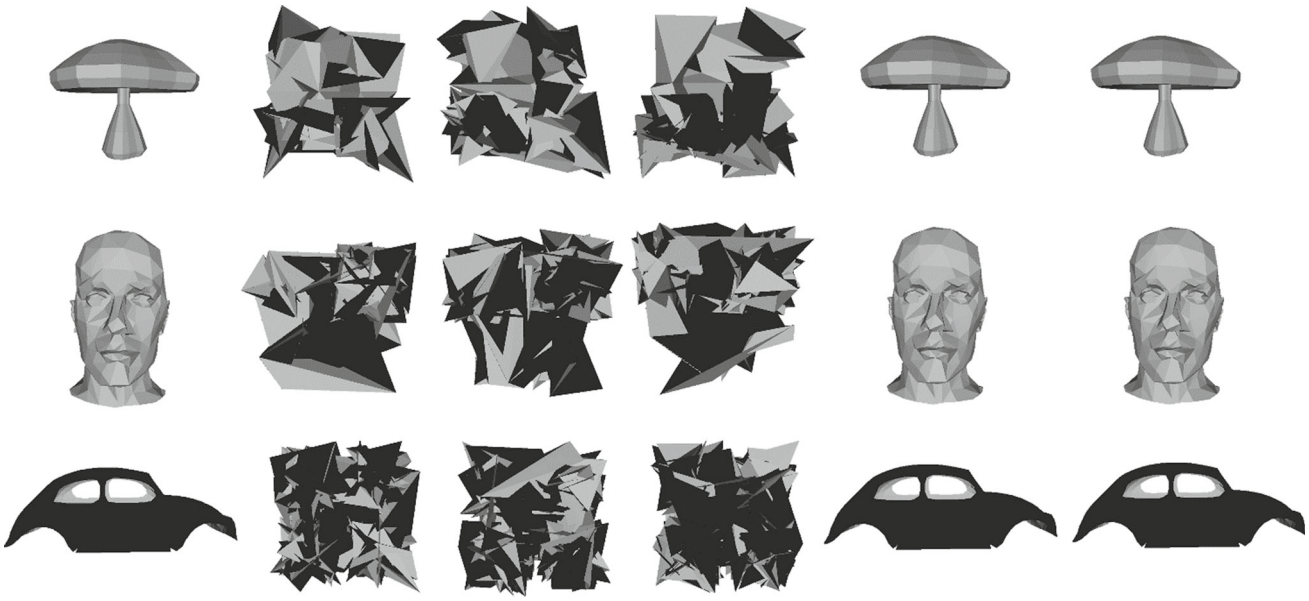

**Fig. 4** Visual effects of test meshes at different stages. From left to right: Original meshes, encrypted meshes, encrypted meshes containing sender's information, encrypted meshes containing sender's and cloud's information, directly decrypted meshes containing sender's information, and recovered meshes

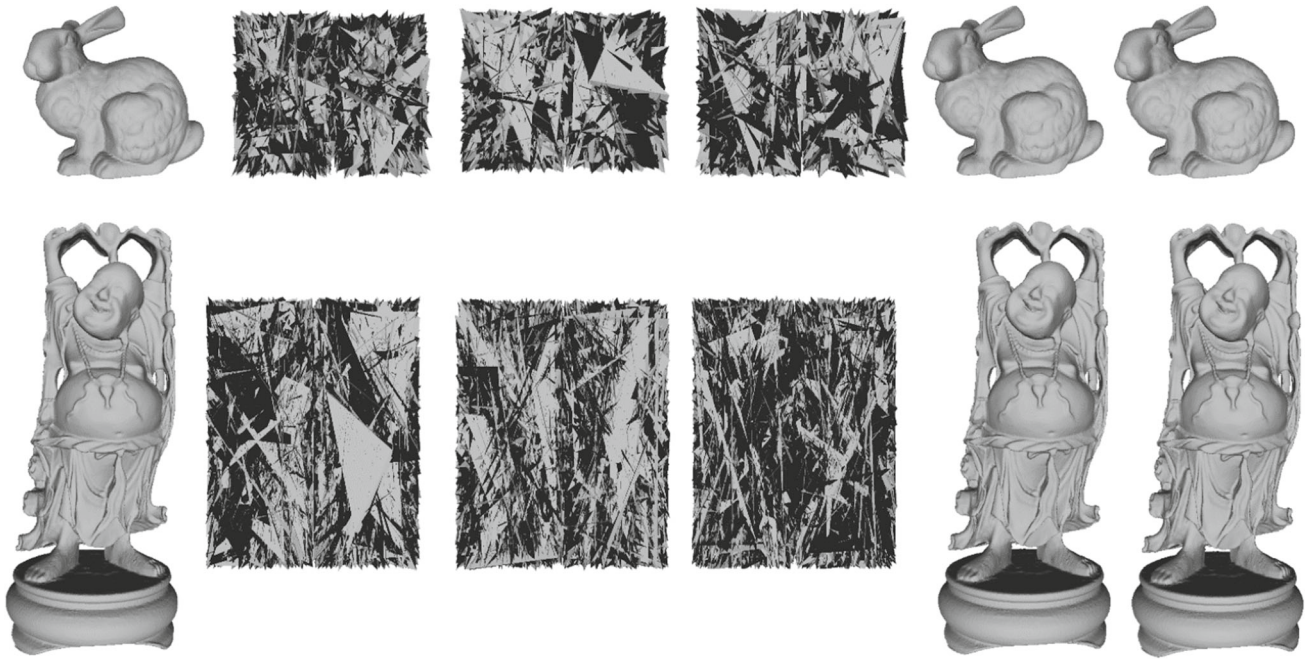

**Fig. 5** Dense meshes: The Stanford Bunny (35,947 vertices, 69,451 triangles) and Happy Bhudda (543,652 vertices, 1,087,716 triangles). From left to right: original meshes, encrypted meshes, encrypted meshes containing sender's information, encrypted meshes containing sender's and cloud's information, directly decrypted meshes containing sender's information, and recovered meshes

#### 4.1.2 Geometric and Visual Quality

The information embedding process adds some distortions to the original cover mesh. These distortions cannot be measured by naked eye inspection which is a complex and time-consuming process. However, mesh processing tools are available which can measure these distortions geometrically or perceptually. In this paper, we propose to use the Hausdorff distance for the geometric measurement and mesh structural distortion measure (MSDM) for the perceptual measurement of distortion between two meshes.

Hausdorff distance is the measure of the maximum distance of a set to the nearest point in the other set. Hausdorff distance between two non-empty sets $A$ and $B$ is defined by

**Table 1** Embedding rate and embedding capacity

| Test models | Number of vertices | Embedding rate (bpv) | Embedding capacity |
|---|---|---|---|
| Mushroom | 226 | 6 | 1356 |
| Mannequin | 428 | 6 | 2568 |
| Beetle | 988 | 6 | 5928 |

Eq. (20).

$$h\left(A, B\right) = \max_{a \in A} \left\{ \min_{b \in B} \left(d\left(a, b\right)\right) \right\} \tag{20}$$

where $d\left(a, b\right)$ is the Euclidean distance between two points $a$ and $b$ of sets $A$ and $B$.

The Hausdorff distance between two meshes does not accurately measure the visual differences [45]; thus, MSDM is used to measure the visual differences between two meshes $M$ and $M'$as given by Eq. (21).

$$d_{\mathrm{MSDM}}\left(M, M'\right) = \left( \frac{1}{n} \sum_{i=1}^{n} d_{\mathrm{LMSDM}}(p_i, q_i)^3 \right)^{1/3} \tag{21}$$

**Table 2** Hausdorff distance and MSDM between original and directly decrypted meshes

| Test models | Hausdorff distance | MSDM |
|---|---|---|
| Mushroom | 0.00040 | 0.00355 |
| Mannequin | 0.00037 | 0.00279 |
| Beetle | 0.000034 | 0.01516 |

where $n$ is the number of vertices in both meshes, $d_{\mathrm{LMSDM}}$ is the local MSDM distances between two meshes and $p$ and $q$ are local windows in both meshes. Therein, $d_{\mathrm{LMSDM}}$ is given in Eq. (22).

$$d_{\mathrm{LMSDM}}\left(p, q\right) = \left(0.4L\left(p, q\right)^3 + 0.4C\left(p, q\right)^3 + 0.2S\left(p, q\right)^3\right)^{1/3} \tag{22}$$

where $L$, $C$ and $S$ are mesh curvature, contrast and structure comparison functions [45], respectively.

The visual quality of directly decrypted meshes is given in Fig. 4. It can be noted that the visual quality of the directly decrypted meshes is identical to the original cover meshes

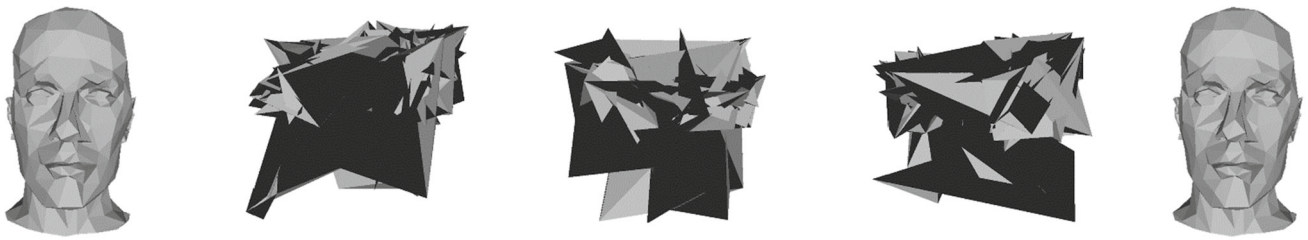

**Fig. 6** First-tier method with 6-bpv. From left to right: original Mannequin mesh, encrypted mesh, encrypted mesh with 3-bpv, encrypted mesh with 6-bpv, and recovered mesh

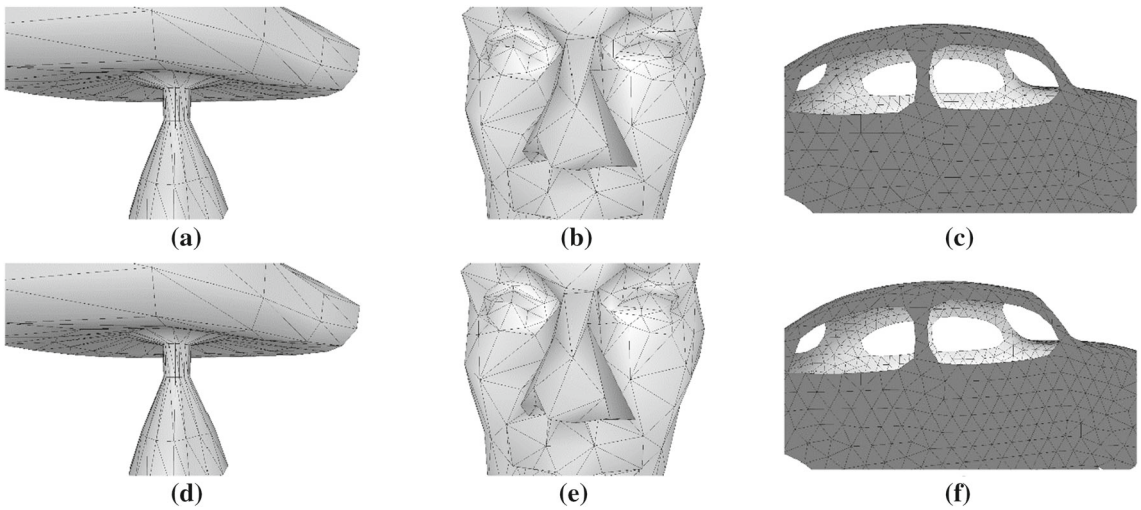


**Fig. 7** Close view of original meshes and directly decrypted meshes. Original meshes (**a**)–(**c**), directly decrypted meshes (**d**)–(**e**)

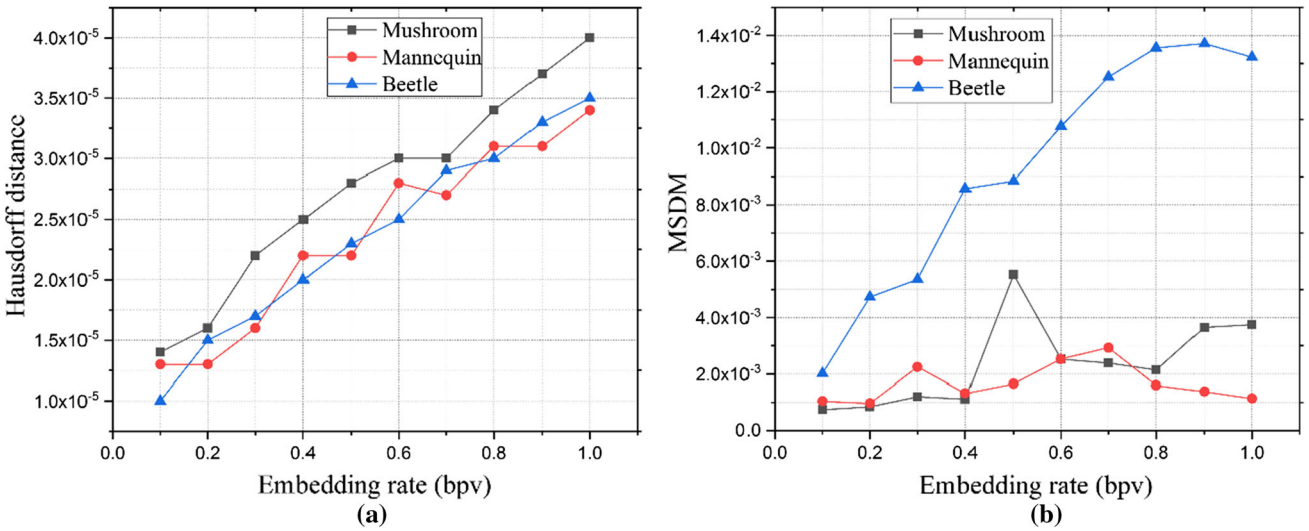


**Fig. 8** Rate distortion curves. **a** Hausdorff distance versus embedding rate, **b** MSDM versus embedding rate

even though the original cover meshes have been modified for information embedding. The naked eye cannot notice the difference between the original and directly decrypted meshes due to the reason that the integer coordinate values of mesh vertices are slightly modified by the addition of information bits to their least significant part. Figure 7 provides a close view of the directly decrypted meshes and original meshes. From Fig. 7, we can observe that no perceptual distortion is introduced by the proposed RDH-ED method. However, these two types of meshes are differentiated geometrically and visually by the Hausdorff distance and MSDM, respectively. Hausdorff distance and MSDM values for the test meshes Mushroom, Mannequin and Beetle are given in Table 2. The rate distortion curves for the test meshes are given in Fig. 8. According to Fig. 8, it can be seen that the Hausdorff distance and MSDM increase with the embedding rate.

#### 4.1.3 Efficiency Performance

The computational efficiency for test meshes is summarized in Table 3. Therein, computational time for encryption, embedding, extraction and decryption processes is given. It is observed that the computational time is proportional to the number of vertices in the host meshes.

#### 4.1.4 Performance Comparison

The performance of the proposed framework is compared with [16].The method proposed in [16] used private key cryptosystem for encryption and is a non-separable method, i.e., hidden information can be extracted after decrypting the mesh. Whereas, the proposed framework uses homomorphic Paillier cryptosystem for encryption. Information can be extracted in the encrypted domain for second-tier method and for the First-tier method, extraction can be achieved in the plain domain after decrypting the mesh. Moreover, the proposed framework expands the vertex values due to the inevitable expansion of the Paillier cryptosystem, whereas the method in [16] does not expand the encrypted vertex values. The embedding rate of [16] is fixed and low due to the fact that embedding depends on the connectivity information of mesh model. The embedding rate reported in [16] for Mushroom is 0.46, for Mannequin is 0.34 and for Beetle is 0.35 bpv. The embedding rate of the proposed framework is much higher than that of [16] which is given in Table 1. Figure 9 compares the geometric quality and Fig. 10 compares the visual quality of the directly decrypted meshes between the proposed framework and [16] based on the parameter $k$. For the proposed framework, we set the embedding rate equal to [16] for the test meshes and computed Hausdorff distance and MSDM for increasing value of $k$ between the directly

**Table 3** Efficiency performance (seconds) of the proposed framework

| Test models | Encryption | 1st-tier | | 2nd-tier | | Decryption |
|---|---|---|---|---|---|---|
| | | Embedding | Extraction | Embedding | Extraction | |
| Mushroom | 3.946 | 1.092 | 1.045 | 43.787 | 0.000284 | 2.933 |
| Mannequin | 8.173 | 1.841 | 1.872 | 163.755 | 0.000510 | 10.638 |
| Beetle | 36.799 | 4.805 | 2.075 | 677.650 | 0.003400 | 33.992 |

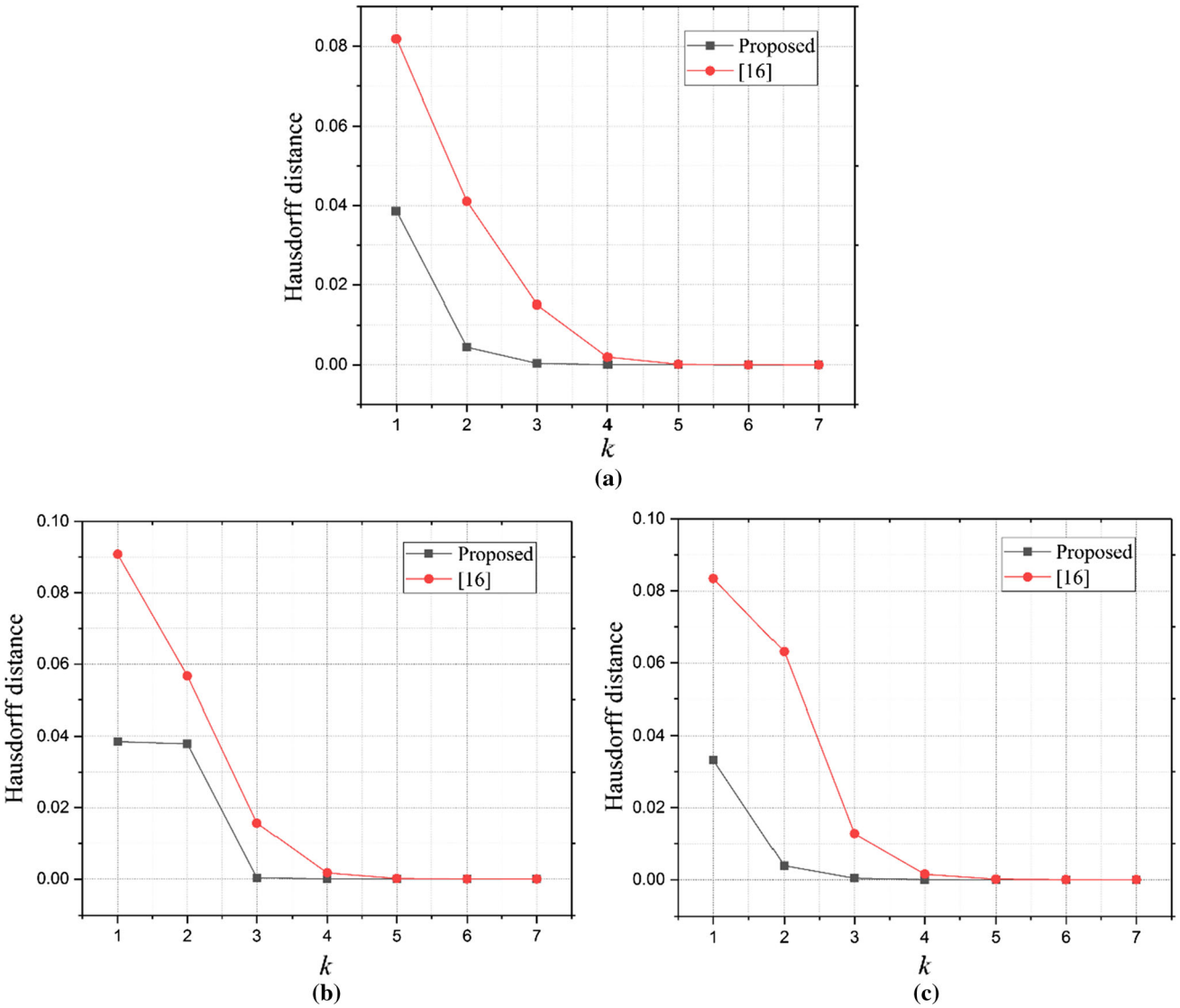


**Fig. 9** Comparison of geometric quality of the directly decrypted meshes between the proposed framework and [16] based on the $k$ parameter. **a** Mushroom, **b** Mannequin, **c** Beetle

decrypted and original meshes. It can be noted that the proposed framework significantly outperforms as compared to the method proposed in [16].

## 5 Conclusion

In this paper, we proposed an effective two-tier RDH-ED framework for 3D mesh models in the homomorphic-encrypted domain. The two-tier RDH-ED framework can be used for end-to-end authentication and cloud data management in the encrypted domain. The vertices of a 3D mesh are first mapped to a form which is suitable for processing by the homomorphic cryptosystem. Paillier encryption is applied to the mapped mesh vertices. The sender first embedded information into the encrypted mesh using the first-tier method and then outsourced the marked mesh to the cloud. Further, the cloud server embedded additional information into the marked encrypted mesh using the second-tier method. Data embedded by the cloud server could be extracted in the encrypted domain directly, whereas, data embedded by the sender could only be extracted in the plain domain. Experimental results demonstrated that 3D meshes can be encrypted, encoded and recovered correctly with embedding rate of 6-bpv. Embedding rate of more than 6-bpv can be achieved by applying the proposed framework iteratively on marked encrypted vertices. Besides, achieving high embedding capacity and reversibility of the mesh models, the proposed RDH-ED framework opens the doorway for complex signal processing in the encrypted domain such as fast

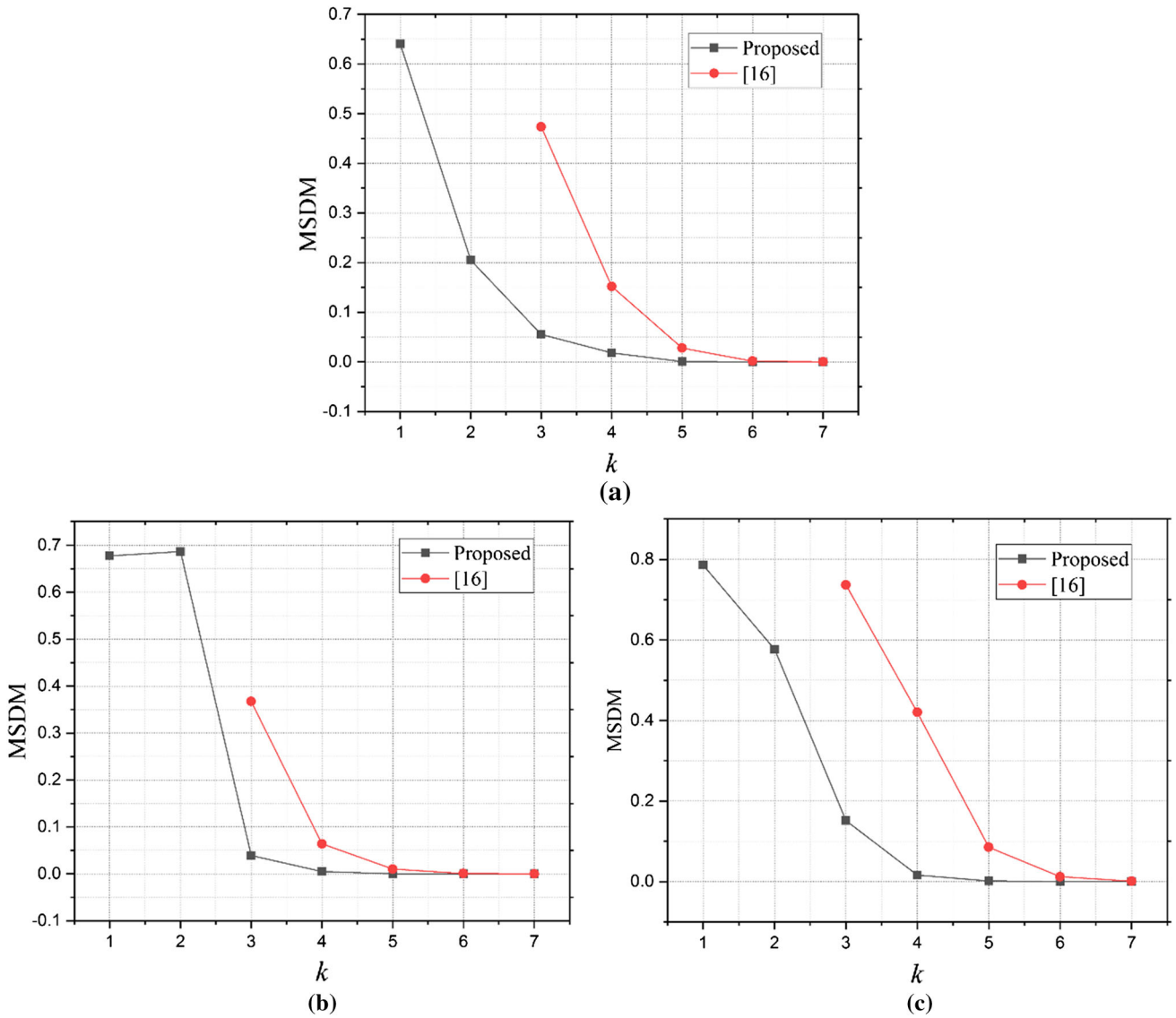


**Fig. 10** Comparison of visual quality of the directly decrypted meshes between the proposed framework and [16] based on the $k$ parameter. **a** Mushroom, **b** Mannequin, **c** Beetle

Fourier transform (FFT), discrete cosine transform (DCT) and discrete wavelet transform (DWT).

**Acknowledgements** This work was supported in part by the Natural Science Foundation of China under Grant U1636201, 61572452 and CAS-TWAS Presidents Fellowship.

## References

1. Cox, I.; Miller, M.; Bloom, J.; Fridrich, J.; Kalker, T.: Digital Watermarking and Steganography. Morgan Kaufmann, Burlington (2007)
2. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, Cambridge (2009)
3. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Technol. **13**(8), 890–896 (2003)
4. Ni, Z.; Shi, Y.-Q.; Ansari, N.; Su, W.: Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. **16**(3), 354–362 (2006)
5. Avci, E.; Tuncer, T.; Avci, D.: A novel reversible data hiding algorithm based on probabilistic XOR secret sharing in wavelet transform domain. Arab. J. Sci. Eng. **41**(8), 3153–3161 (2016)
6. Celik, M.U.; Sharma, G.; Tekalp, A.M.; Saber, E.: Lossless generalized-LSB data embedding. IEEE Trans. Image Process. **14**(2), 253–266 (2005)
7. Weng, S.; Zhao, Y.; Pan, J.-S.; Ni, R.: Reversible watermarking based on invariability and adjustment on pixel pairs. IEEE Signal Process. Lett. **15**, 721–724 (2008)
8. Wang, X.; Li, X.; Yang, B.; Guo, Z.: Efficient generalized integer transform for reversible watermarking. IEEE Signal Process. Lett. **17**(6), 567–570 (2010)

9. Qiu, Y.; Qian, Z.; Yu, L.: Adaptive reversible data hiding by extending the generalized integer transformation. IEEE Signal Process. Lett. **23**(1), 130–134 (2016)
10. Coltuc, D.: Low distortion transform for reversible watermarking. IEEE Trans. Image Process. **21**(1), 412–417 (2012)
11. Thodi, D.M.; Rodriguez, J.J.: Prediction-error based reversible watermarking. In: Image Processing, 2004. ICIP'04. 2004 International Conference on, pp. 1549-1552. IEEE (2004)
12. Thodi, D.M.; Rodríguez, J.J.: Expansion embedding techniques for reversible watermarking. IEEE Trans. Image Process. **16**(3), 721–730 (2007)
13. Zhang, W.; Hu, X.; Li, X.; Yu, N.: Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression. IEEE Trans. Image Process. **22**(7), 2775–2785 (2013)
14. Hu, X.; Zhang, W.; Hu, X.; Yu, N.; Zhao, X.; Li, F.: Fast estimation of optimal marked-signal distribution for reversible data hiding. IEEE Trans. Inf. Forensics Secur. **8**(5), 779–788 (2013)
15. Zhang, W.; Hu, X.; Li, X.; Nenghai, Y.: Optimal transition probability of reversible data hiding for general distortion metrics and its applications. IEEE Trans. Image Process. **24**(1), 294–304 (2015)
16. Jiang, R.; Zhou, H.; Zhang, W.; Yu, N.-H.: Reversible data hiding in encrypted 3D mesh models. IEEE Trans. Multimedia **20**(1), 55–67 (2017)
17. Ma, K.; Zhang, W.; Zhao, X.; Yu, N.; Li, F.: Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans. Inf. Forensics Secur. **8**(3), 553–562 (2013)
18. Zhang, W.; Ma, K.; Yu, N.: Reversibility improved data hiding in encrypted images. Signal Process. **94**, 118–127 (2014)
19. Cao, X.; Du, L.; Wei, X.; Meng, D.; Guo, X.: High capacity reversible data hiding in encrypted images by patch-level sparse representation. IEEE Trans. Cybern. **46**(5), 1132–1143 (2016)
20. Zhang, X.: Reversible data hiding in encrypted image. IEEE Signal Process. Lett. **18**(4), 255–258 (2011)
21. Liao, X.; Shu, C.: Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. J. Vis. Commun. Image Represent. **28**, 21–27 (2015)
22. Zhou, J.; Sun, W.; Dong, L.; Liu, X.; Au, O.C.; Tang, Y.Y.: Secure reversible image data hiding over encrypted domain via key modulation. IEEE Trans. Circuits Syst. Video Technol. **26**(3), 441–452 (2016)
23. Zheng, S.; Li, D.; Hu, D.; Ye, D.; Wang, L.; Wang, J.: Lossless data hiding algorithm for encrypted images with high capacity. Multimedia Tools Appl. **75**(21), 13765–13778 (2016)
24. Qian, Z.; Zhang, X.: Reversible data hiding in encrypted images with distributed source encoding. IEEE Trans. Circuits Syst. Video Technol. **26**(4), 636–646 (2016)
25. Chen, Y.-C.; Shiu, C.-W.; Horng, G.: Encrypted signal-based reversible data hiding with public key cryptosystem. J. Vis. Commun. Image Represent. **25**(5), 1164–1170 (2014)
26. Wu, X.; Chen, B.; Weng, J.: Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer. J. Vis. Commun. Image Represent. **41**, 58–64 (2016)
27. Wu, H.-T.; Cheung, Y.-M.; Huang, J.: Reversible data hiding in Paillier cryptosystem. J. Vis. Commun. Image Represent. **40**, 765–771 (2016)
28. Zhang, X.; Long, J.; Wang, Z.; Cheng, H.: Lossless and reversible data hiding in encrypted images with public-key cryptography. IEEE Trans. Circuits Syst. Video Technol. **26**(9), 1622–1631 (2016)
29. Xiang, S.; Luo, X.: Efficient reversible data hiding in encrypted image with public key cryptosystem. EURASIP J. Adv. Signal Process. **2017**(1), 59 (2017)
30. Chou, D.; Jhou, C.-Y.; Chu, S.-C.: Reversible watermark for 3D vertices based on data hiding in mesh formation. Int. J. Innov. Comput. Inf. Control **5**(7), 1893–1901 (2009)
31. Wu, H.-t.; Dugelay, J.-L.: Reversible watermarking of 3D mesh models by prediction-error expansion. In: Multimedia Signal Processing, 2008 IEEE 10th Workshop on, pp. 797-802. IEEE (2008)
32. Wu, H.-T.; Cheung, Y.-m.: A reversible data hiding approach to mesh authentication. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 774–777. IEEE Computer Society (2005)
33. Luo, H.; Lu, Z.-m.; Pan, J.-s.: A reversible data hiding scheme for 3D point cloud model. In: Signal Processing and Information Technology, 2006 IEEE International Symposium on, pp. 863–867. IEEE (2006)
34. Luo, H.; Pan, J.-S.; Lu, Z.-M.; Huang, H.-C.: Reversible data hiding for 3D point cloud model. In: Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP'06. International Conference on, pp. 487–490. IEEE (2006)
35. Sun, Z.; Lu, Z.-M.; Li, Z.: Reversible data hiding for 3D meshes in the PVQ-compressed domain. In: Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP'06. International Conference on, pp. 593–596. IEEE (2006)
36. Lu, Z.-M.; Li, Z.: High capacity reversible data hiding for 3D meshes in the PVQ domain. In: Shi, Y.Q., Kim, H.J., Katzenbeisser, S. (eds.) Digital Watermarking. IWDW 2007. Lecture Notes in Computer Science, vol. 5041, pp. 233–243. Springer, Berlin, Heidelberg (2008)
37. Bianchi, T.; Piva, A.; Barni, M.: On the implementation of the discrete Fourier transform in the encrypted domain. IEEE Trans. Inf. Forensics Secur. **4**(1), 86–97 (2009)
38. Deering, M.: Geometry compression. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp. 13–20. ACM (1995)
39. Rivest, R.L.; Adleman, L.; Dertouzos, M.L.: On data banks and privacy homomorphisms. Found. Secure Comput. **4**(11), 169–180 (1978)
40. Gentry, C.; Halevi, S.: Implementing gentry's fully-homomorphic encryption scheme. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 129–148. Springer (2011)
41. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 223–238. Springer (1999)
42. Damgård, I.; Jurik, M.: A generalisation, a simpli. cation and some applications of paillier's probabilistic public-key system. In: International Workshop on Public Key Cryptography, pp. 119–136. Springer (2001)
43. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
44. Goldwasser, S.; Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984)
45. Lavoué, G.; Gelasca, E.D.; Dupont, F.; Baskurt, A.; Ebrahimi, T.: Perceptually driven 3D distance metrics with application to watermarking. In: Applications of Digital Image Processing XXIX 2006, p. 63120L. International Society for Optics and Photonics