3D Object Watermarking from Data Hiding in the Homomorphic Encrypted Domain

Bianca Jansen van Rensburg, Pauline Puteaux, William Puech, Jean-Pierre Pedeboy

3D Object Watermarking from Data Hiding in the Homomorphic Encrypted Domain

BIANCA JANSEN VAN RENSBURG, LIRMM, Université de Montpellier, CNRS, France
PAULINE PUTEAUX, CRIStAL, CNRS, Univ. Lille, Centrale Lille, France
WILLIAM PUECH, LIRMM, Université de Montpellier, CNRS, France
JEAN-PIERRE PEDEBOY, Stratégies, France

For over a decade, 3D objects are an increasingly popular form of media. It has become necessary and urgent to secure them during their transmission or archiving. In this paper, we propose a new method to obtain a watermarked 3D object from high-capacity data hiding in the encrypted domain. Based on the homomorphic properties of the Paillier cryptosystem, our proposed method allows us to embed several secret messages in the encrypted domain with a high-capacity. These messages can be extracted in the plain-text domain after the 3D object decryption. To the best of our knowledge, we are the first to propose a data hiding method in the encrypted domain where the high-capacity watermark is conserved in the plain-text domain after the 3D object is decrypted. The encryption and the data hiding in the encrypted domain are format compliant and without size expansion, despite the use of the Paillier cryptosystem. Each time a new message is embedded in the encrypted domain, flags are added in order to indicate which blocks are still available for the embedding of additional messages. After the decryption of a watermarked encrypted 3D object, our method produces a watermarked 3D object which is visually very similar to the original 3D object. From the decrypted watermarked 3D object, we can then extract all the embedded messages directly in the plain-text domain, without the need for an auxiliary file. Moreover, large keys are used, rending our method secure for real life applications.

CCS Concepts: • **Security and privacy → Management and querying of encrypted data**; **Management and querying of encrypted data**; • **Computing methodologies → Computer graphics**.

Additional Key Words and Phrases: Multimedia security, high-capacity data hiding, 3D object security, Paillier homomorphic encryption, signal processing in the encrypted domain, format compliant.

ACM Reference Format:
Bianca Jansen van Rensburg, Pauline Puteaux, William Puech, and Jean-Pierre Pedeboy. 2022. 3D Object Watermarking from Data Hiding in the Homomorphic Encrypted Domain. *J. ACM* 37, 4, Article 111 (August 2022), 21 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Over the last decade, the cloud has become a popular way of storing and transferring multimedia such as images, videos and 3D objects. Therefore, the need for multimedia security has become very important. Various 3D security methods have been proposed, for example, for 3D watermarking [5, 29] or for 3D object sharing [2]. Encryption methods serve to secure the multimedia file by

Authors' addresses: Bianca Jansen van Rensburg, LIRMM, Université de Montpellier, CNRS, Montpellier, France, 34095, bianca.jansen-van-rensburg@lirmm.fr; Pauline Puteaux, CRIStAL, CNRS, Univ. Lille, Centrale Lille, Lille, France, 59000, pauline.puteaux@cnrs.fr; William Puech, LIRMM, Université de Montpellier, CNRS, Montpellier, France, 34095, william.puech@lirmm.fr; Jean-Pierre Pedeboy, Stratégies, Rungis, France, 94510, jp.pedeboy@cadwin.com.

0004-5411/2022/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

converting its content to unintelligible ciphertext. Once the media is encrypted and located in the cloud, a user, whether it be the original owner of the media or a third party, may wish to embed data in the encrypted media. Some uses for embedding data in encrypted media include annotating, tracing or authenticating the media.

The advantage of data hiding in the encrypted domain (DH-ED) is that it allows third party users to embed data into the cover media, without knowledge of the original content and therefore without the need to compromise the confidentiality of the cover media. However, traditional data hiding methods usually have a tendency to distort the cover media [7]. This renders these methods ineffective in domains such as 3D object manufacturing or healthcare, where very high quality recovery of the original data is a necessity. To address this issue, high-capacity data hiding (HCDH) methods have been proposed for high quality recovery of the original multimedia content [9, 13].

In recent years, more and more high-capacity data hiding methods in the encrypted domain (HCDH-ED) have been proposed [20, 37]. While the literature contains many methods that have been developed for images, very few methods have been developed for 3D objects [12, 26, 34]. Over the last decade, the popularity of 3D objects has greatly increased and with it, the need to secure 3D objects during their transmission over networks or their archiving to the cloud. 3D objects are now used in many domains, notably the manufacturing, healthcare and entertainment industry, among others. It is therefore an increasingly urgent necessity to secure and process these 3D objects. Despite the development of applications for 3D data hiding in the encrypted domain, it remains a relatively unexplored research area.

In this paper, we propose an efficient HCDH-ED method based on the Paillier cryptosystem for 3D objects. Based on the homomorphic properties of the Paillier cryptosystem, our method allows us to obtain a watermarked 3D object in the plain-text domain. From this decrypted watermarked 3D object, the embedded messages can then be extracted without the need for an auxiliary file. Moreover, our proposed method is format compliant since it preserves the original format of the 3D object and there is no size expansion in the encrypted domain. In order to have a large key size, vertices are grouped into blocks without reducing the payload. We note that the payload corresponds to the number of bits that can be embedded per vertex. The objectives of the method proposed in this paper are very different from those described in our previous work [25]. Indeed, in this paper, we suggest to embed several messages. What was indicated as a second tier message in [25], we have now adapted to serve as flags. These flags allow us to synchronize a large number of messages, which allows us to find out which blocks are marked and to clearly separate each message. Indeed, with these flags each time a new message is embedded in the encrypted 3D object, the location of this message is highlighted by adding flags. These flags indicate where additional messages can be placed in the 3D object, and allow the messages to be retrieved during the decryption process. This flag embedding process is fully reversible and has no impact on the decoding phase or on the decrypted 3D object. With our proposed method, the visual quality of the watermarked 3D object in the plain-text domain is very high when compared to the corresponding original 3D object. When compared to state of the art methods, our proposal is the only one to avoid size expansion, an auxiliary file and data errors which refer to errors in the retrieved message. Moreover, our method is able to generate a watermarked 3D object in the plain-text domain.

In this paper, the proposed method is based on encrypting only the vertices of the 3D object and therefore the 3D object's point cloud. We note that 3D encryption is very different to 2D image encryption. Indeed, while 2D images are composed of an ordered matrix of 8-bit pixels, 3D

objects consist of vertices composed of unordered 32-bit floating coordinates. Therefore, 3D object watermarking presents additional challenges.

The main contributions of this paper are summarized as follows:

(1) The proposed method allows us to obtain a high-capacity watermarked 3D object in the plain-text domain for which messages have been embedded in the encrypted domain. To the best of our knowledge, we are the only method that can achieve this;
(2) The method is format compliant and there is no size expansion in the encrypted domain;
(3) Very large key sizes can be used by grouping vertices into blocks;
(4) Several messages can be embedded in the same encrypted 3D object. This process has no impact on the reconstruction and no auxiliary information is required for message extraction in the plain-text domain.

This paper is organised as follows. In Section 2, we describe other current state-of-the-art methods including data hiding methods in the encrypted domain, homomorphic cryptosystems, and specific DH-ED methods for 3D objects. In Section 3, we present in detail our proposed HCDH-ED method based on the Paillier cryptosystem for 3D objects. In Section 4, we develop experimental results and comparisons with previous work. Finally, in Section 5, we conclude our paper and present perspectives for future work.

2 PREVIOUS WORK

In this section, we present previous work as well as the Paillier cryptosystem. First, in Section 2.1 we describe data hiding in the encrypted domain (DH-ED), then we present homomorphic cryptosystems, in particular the Paillier cryptosystem in Section 2.2. Finally, in Section 2.3 we detail recent data hiding methods in the encrypted domain applied to 3D objects.

2.1 Data Hiding in the Encrypted Domain

DH-ED allows data to be embedded in the support without revealing information about the content of the original support and therefore ensuring its visual confidentiality.

DH-ED methods can be broken down into two main categories: Reserving Room Before Encryption (RRBE) [4, 16, 20, 21, 23], and Vacating Room After Encryption (VRAE) [10, 19, 35]. In RRBE methods, the content owner liberates space for the data in the media in a preprocessing step. While in VRAE methods, the media is first encrypted by the owner and the data hider can then embed the data by modifying the encrypted media.

Several methods based on public key homomorphic cryptosystems have been proposed [6, 17, 22, 28, 30–32, 36–39]. These methods are based on either the Paillier cryptosystem [18] or cryptosystems involving the learning with errors (LWE) problem [24]. Chen *et al.* were the first to propose a data hiding scheme based on the Paillier cryptosystem [6]. Shiu *et al.* [28] then improved the method of Chen *et al.* [6] by integrating difference expansion. Zhang *et al.* proposed a reversible and a lossless data hiding method [36]. Zhou *et al.* proposed a method based on a two-class SVM classifier which allows us to distinguish encrypted and non-encrypted image patches in order to perfectly reconstruct the embedded message and the original image [38]. Wu *et al.* proposed two methods [30] and [31] based on the Paillier cryptosystem. Xiang and Luo described a method where an image is divided into sections for self-embedding before encryption [32]. Zheng *et al.* described a lossless, high-capacity data hiding method based on efficient mapping and use of expanded pixel values [37]. Malik *et al.* suggested a data hiding method using interpolation [17]. Zhou *et al.* proposed a separable reversible data hiding scheme based on NTRU [39]. Puteaux *et al.* proposed a high-capacity data hiding scheme in images that is based on least significant bit (LSB)

substitution [22]. In fact, in this paper, Puteaux *et al.* perform a histogram shrinking function so that the pixel values are in the range $[0, n-1]$, where $n-1$ is the product of two integers. This is done in order to avoid pixel value overflows. Once the image is encrypted, there is a size expansion of 2 [22].

2.2 Homomorphic cryptosystems

Homomorphic cryptosystems are beneficial in signal processing as they translate a mathematical operation in the plain-text domain to another operation in the encrypted domain:

$$\mathcal{D}(\mathcal{E}(m_1) \oplus \mathcal{E}(m_2)) = \mathcal{D}(\mathcal{E}(m_1 \otimes m_2)), \tag{1}$$

where $\mathcal{E}(\cdot)$ is a homomorphic encryption function, $\mathcal{D}(\cdot)$ is a homomorphic decryption function, and m_1 and m_2 are the two plaintexts to be encrypted.

Homomorphic cryptosystems allow a third party to modify content in the plain-text domain without the need to decrypt the content and therefore without compromising security. What is more is that unlike non homomorphic cryptosystems which are deterministic, homomorphic cryptosystems are probabilistic. The Paillier cryptosystem is an asymmetric homomorphic cryptosystem introduced by Paillier in 1999 [18]. Concerning its security against attacks, the Paillier cryptosystem is IND-CPA secure (*i.e.* indistinguishable under chosen-plaintext attacks). It can be IND-CCA1 secure (*i.e.* indistinguishable under non-adaptive chosen ciphertext attack) depending on the parameters used. However, like all homomorphic cryptosystems – which are known to be malleable – it cannot be IND-CCA2 secure (*i.e.* indistinguishable under adaptive chosen ciphertext attack) [1]. This cryptosystem converts a multiplication in the encrypted domain to an addition in the plain-text domain. To generate the keys, we choose two prime numbers p, q such that:

$$gcd(pq, (p-1)(q-1)) = 1. \tag{2}$$

Set n and λ such that:

$$n = pq \text{ and } \lambda = lcm((p-1), (q-1)). \tag{3}$$

Choose $g \in (\mathbb{Z}/n^2\mathbb{Z})^*$ such that:

$$\exists\, \mu \mid \mu = (L(g^\lambda \bmod (n^2)))^{-1} \bmod (n), \tag{4}$$

where $L(\cdot)$ is defined as:

$$L(x) = \frac{x-1}{n}, \text{ where } x \in \mathbb{N}^*. \tag{5}$$

The public key is given by (n, g) and the private key by (λ, μ). If m is a plaintext to be encrypted, where $0 \leq m < n$, r randomly generated, where $r \in (\mathbb{Z}/n\mathbb{Z})^*$, and $\mathcal{E}(\cdot)$ the Paillier encryption function, then the ciphertext c is:

$$c = \mathcal{E}(m) = g^m \times r^n \bmod n^2. \tag{6}$$

This is the random value of r which guarantees the cryptosystem's probabilistic property. This property indicates that the encrypted value of a plaintext is not unique.

From the ciphertext c, the initial message m is retrieved:

$$m = \mathcal{D}(c) = L(c^\lambda \bmod n^2) \times \mu \bmod n, \tag{7}$$

where $\mathcal{D}(\cdot)$ is the Paillier decryption function.

The Paillier cryptosystem has multiple homomorphic properties which we exploit in our proposed method presented in this paper. The first of which is the Paillier additive homomorphic property which converts an addition in the plain-text domain to a multiplication in the encrypted domain:

$$\mathcal{D}((\mathcal{E}(m_1) \times \mathcal{E}(m_2)) \bmod n^2) = (m_1 + m_2) \bmod n, \tag{8}$$

where m_1 and m_2 are the two plaintexts to be encrypted.

As the homomorphic cryptosystems are probabilistic, by definition there exists multiple values of $\mathcal{E}(m)$ for every m. We can then modify $\mathcal{E}(m)$ such that:

$$\mathcal{D}(\mathcal{E}(m) \times (t^n \bmod n^2) \bmod n^2) = m \bmod n, \tag{9}$$

where t is relatively prime to n.

This property is termed the self-blinding property.

2.3 DH-ED for 3D objects

Several methods have been proposed for 3D object security [2, 29], yet to our knowledge, there exists very few papers for 3D DH-ED objects. In 2018 Jiang *et al.* proposed a DH-ED method which maps the floating point vertex coordinates to integers and encrypts the 3D object by performing an exclusive-or on the mapped coordinates with a pseudo-random bit stream [12]. Some of the encrypted vertices are watermarked for the embedding. The data is then embedded in the watermarked vertices by performing an exclusive-or on the LSB with the data to encrypt. This method has the disadvantage of a low payload, distorted reconstructed 3D objects and a high error rate when extracting the embedded data.

The method of Jiang *et al.* [12] was later improved by Yin *et al.* [34] in 2019. This method improves upon [12] by using an error prediction protocol to mark the vertices to be embedded before the encryption. Note that an auxiliary file is needed to store this information. Data is then embedded by substituting the m most significant bits (MSB), which can be later reconstructed with the vertex ring. We can also note that the payload depends on the 3D object characteristics.

In 2018, Shah *et al.* proposed a DH-ED for 3D objects using the Paillier cryptosystem [26]. This method describes a two tier homomorphic DH-ED scheme. The floating point vertex coordinates are first mapped to positive integers so the Paillier cryptosystem is able to process them. The 3D object is encrypted using the Paillier cryptosystem. The first tier of data hiding is completed by using the Paillier cryptosystem's homomorphic properties to perform a histogram expansion and shifting in the encrypted domain. This results in a significant size expansion. The second tier data embedding is done by using the Paillier self-blinding property. This is the only state-of-the-art method that preserves the embedded message once the 3D object is decrypted.

Very recently, in 2022, Xu *et al.* proposed a DH-ED for 3D objects where the vertex coordinates are mapped to integers, and vertices divided into an embedding set and a reference set [33]. The vertices are then encrypted with an exclusive-or and data is embedded by substituting the MSB of each coordinate in the embedding set. This method has the disadvantage of a low payload and the use of an auxiliary file.

The method of Xu *et al.* [33] was then improved by Lyu *et al.* [15]. This method optimises the distribution between the embedding set and the prediction set using the vertices' parity. The data is embedded by substituting t MSB of the embedding set, where t has a variable length.

In order to overcome these limitations, we propose a new format compliant high-capacity DH-ED (HCDH-ED) method without size expansion and without visual degradation of the 3D object. Moreover, no auxiliary file is needed to extract the embedded messages in the plain-text domain. To the best of our knowledge, we are the only method to retrieve a high-capacity watermarked 3D object after decryption.

3 THE PROPOSED HCDH-ED METHOD FOR 3D OBJECTS

In this section, we present in detail our proposed HCDH-ED method for 3D objects. Our method is based on the Paillier cryptosystem and uses its homomorphic properties (Eq. (8) and Eq. (9)) in order to embed messages in an homomorphically encrypted 3D object, without changing the connectivity of the 3D object. Fig. 1 presents the overview of the encoding phase of our proposed method.

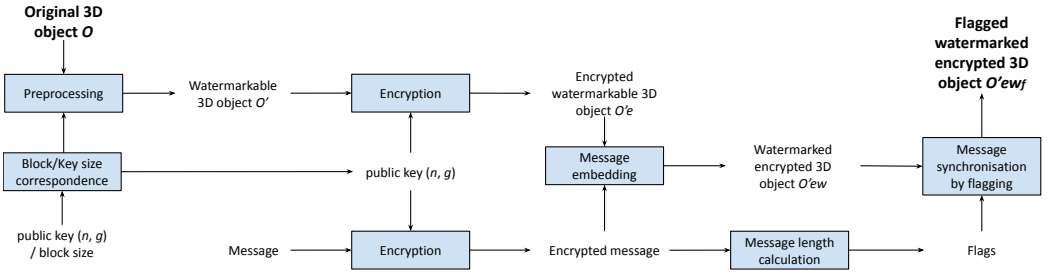


Fig. 1. Overview of the encoding phase of the proposed HCDH-ED method for 3D objects.

In Section 3.1, we first analyse the necessary key size as a function of the desired block size. In Section 3.2, we describe the preprocessing step where the vertices are grouped into blocks, noted B, of size b vertices per block. The block size is directly proportional to the key size. Bits which we wish to use to embed the messages are designated according to the size b as well as the payload per block α, and are set to zero. In Section 3.3, we present our 3D object encryption method based on the Paillier cryptosystem. In Section 3.4, we present the data hiding step in the encrypted domain. A message to embed is encrypted with the same public key used for the 3D object encryption. Both the message and the 3D object are multiplied in the encrypted domain. This is equivalent to an addition in the plain-text domain. Each time a new message is embedded in the encrypted 3D object, the vertex blocks used to embed the message are flagged. Indeed, by exploiting the Paillier probabilistic property in the encrypted domain (Eq. (9)), it is possible to flag the used vertex blocks without impacting the reconstruction of the 3D object in the plain-text domain. Finally, in Section 3.5, we present how the 3D object is reconstructed and then how the embedded messages are extracted in the plain-text domain.

3.1 Key size analysis

We consider each block to have a size of $2k + 1$ bits. We note that the block size is determined by the size of the key. If k is the number of bits per block we want to encrypt, then according to the constraints imposed by the Paillier cryptosystem, the value of n of the public key (n, g), should be represented with at least $k + 1$ bits. Therefore we have:

$$2^{2k} \leq n^2. \tag{10}$$

Due to the modulus n^2 in Eq. (6), the size of the encrypted data is at most n^2. In order to limit the size of the encrypted data to $2k + 1$, and consequently avoid size expansion, we impose the following constraint:

$$2^{2k} \leq n^2 < 2^{2k+1}. \tag{11}$$

Therefore, n is constrained by:

$$2^k \leq n < \sqrt{2} \cdot 2^k. \tag{12}$$

The relationship between n and the size b of a block B is deduced in Section 3.2.

3.2 Preprocessing

We note the original 3D object O. One of the possible ways a 3D object can be represented is by a set of vertices $\mathcal{V} = \{v_0, ..., v_{|\mathcal{V}|-1}\}$ and faces $\mathcal{F} = \{f_0, ..., f_{|\mathcal{F}|-1}\}$, where $\mathcal{F}$ describes the 3D object's connectivity. In our proposed approach, messages are embedded without changing the 3D object's connectivity, and so only the set $\mathcal{V}$ is of interest in our proposed method. Each vertex $v \in \mathcal{V}$ consists of three coordinates x, y and z, where each of which can be represented by a 32-bit floating point.

According to the IEEE 754 standard, a 32-bit floating point $fp \in \{x, y, z\}$ consists of a sign s represented with 1 bit, an exponent e represented with 8 bits and a mantissa $mant$ represented with 23 bits (from MSB to LSB) where:

$$fp = (-1)^s \times mant \times 2^{e-127}. \tag{13}$$

Fig. 2 illustrates how a 32-bit floating point fp is divided into s, e and $mant$.

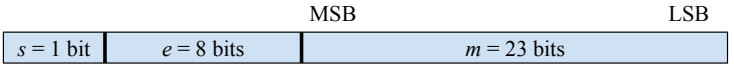


Fig. 2. Representation of a 32-bit floating point according to the IEEE 754 standard.

Homomorphic cryptosystems cannot process floating point values due to the complexity of simple mathematical operations which are used in the encryption and data hiding processes. Therefore the encryption is performed exclusively on the mantissas. Additionally, encrypting only the mantissa allows the encrypted 3D object to remain format compliant. This does not compromise security because the mantissa contains the most relevant information, while s and e contain mainly structural information. The 23 bits of the mantissa of each coordinate are transformed into an integer. This means that the part of each vertex v we want to encrypt is encoded with $23 \times 3 = 69$ bits.

In order to have a key sufficiently large to be secure, vertices are grouped into blocks B of size b vertices per block. Each block therefore consists of $69b$ bits. A block of vertices is then constructed by first grouping the MSB-0 of each vertex coordinate, then the MSB-1, until finally the LSB, as illustrated in Fig. 3. We note that due to the nature of the Paillier cryptosystem, the size of the block cannot exceed the size of the key. The size of the key is in turn limited by the complexity of the Paillier cryptosystem. The size of the block is therefore determined by the size of the key. Dividing the vertices into blocks allows the embedding of multiple messages, as each block can only contain a single message.

We note α the payload in bits per block. Each message to embed is divided into segments of size α bits. To avoid a bit overflow when we embed a segment of a message in a block B, as illustrated in Fig. 4, α bits of the block B are set to zero in the plain-text domain. If k is the number of bits to encrypt in a block B, then the α LSB among the k MSB are set to zero, as illustrated in Fig. 4. We note B' the watermarkable vertex block and O' the corresponding watermarkable 3D object.

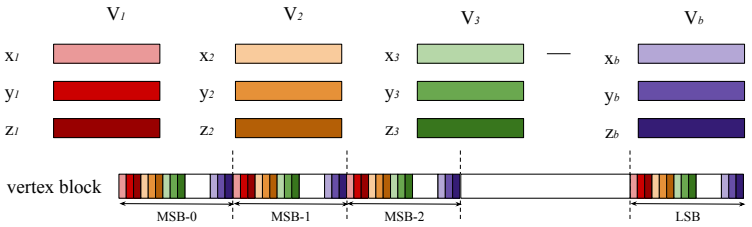


Fig. 3. Construction of a block B composed of b vertices.

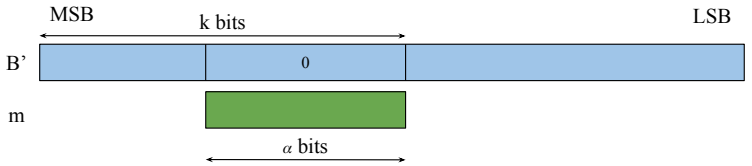


Fig. 4. Preprocessing of a vertex block B in the plain-text domain.

3.3 Encryption

To avoid a size expansion of the encrypted vertex block in relation to the plain-text vertex block, we set the size of the encrypted vertex block $69b = 2k + 1$ bits. This means that, as illustrated in Fig. 4, we should then encrypt:

$$k = \frac{69b - 1}{2} \text{ bits.} \tag{14}$$

We note that in order for $69b = 2k + 1$, then the block size b has to be odd.

To encrypt the k MSB of the block B', which we note $B'_{k_{MSB}}$, we use Eq. (6). We then obtain the $2k + 1$ bits as illustrated in Fig. 5. The resulting $2k + 1$ encrypted bits substitute the bits of B'. The encrypted block B'_e is then divided into individual vertices in order to respect the original format of the 3D object. We note B'_e the encrypted watermarkable vertex block and O'_e the corresponding encrypted watermarkable 3D object.

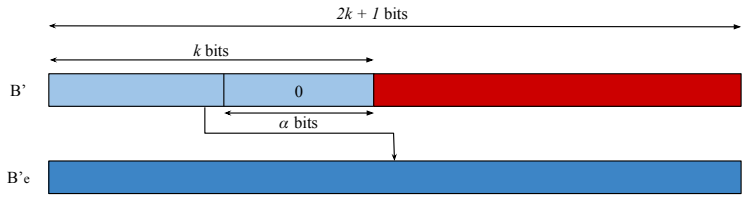


Fig. 5. Encryption process of a watermarkable vertex block B'.

We note that the remaining $k + 1$ LSB of B' are not included in the encryption step. They are then lost. Perceptual-based metrics have been used to measure the visual degradation of 3D objects [8, 11, 14]. According to an analysis proposed by Beugnon *et al.* [3], we assume that we can lose up to 16 LSB in the mantissa of each vertex coordinate, without visual degradation according to the human visual system (HVS). Just as images with a PSNR > 50 dB are considered reversible because there is no visual degradation according to the HVS, these 3D objects are considered to

have no visual degradation because of their very small Hausdorff distances. This signifies that in each vertex there can be a loss of $3 \times 16b = 48b$ bits per block B before there is visual impact on the decrypted 3D object. Therefore, losing $k+1$ LSB is not a problem, since $k+1 < 48b$ according to Eq. (14).

3.4 Data hiding in the encrypted domain

In this section, we describe the data hiding process for the messages embedded in the encrypted domain.

3.4.1 Message embedding. In order to embed a message segment m in each block B'_e of the encrypted watermarkable 3D object O'_e, we use the Paillier additive homomorphic property of Eq. (8), which indicates that a multiplication in the encrypted domain is equivalent to an addition in the plain-text domain. Therefore, to embed the message segment m, we use the following equation:

$$B'_{ew} = \mathcal{E}(B'_{k_{MSB}}) \times \mathcal{E}(m) \bmod n^2, \tag{15}$$

where B'_{ew} is the watermarked encrypted block, $\mathcal{E}(\cdot)$ is the Paillier encryption function and $\mathcal{E}(B'_{k_{MSB}}) = B'_e$.

We note O'_{ew} the corresponding watermarked encrypted 3D object. Note that since this multiplication in the encrypted domain is equivalent to an addition in the plain-text domain, and since we have already cleared space for m by setting the α bits of the payload to 0, this operation is equivalent to an α LSB substitution in the plain-text domain. We can then reduce Eq. (8) to:

$$\mathcal{D}(\mathcal{E}(B'_{k_{MSB}}) \times \mathcal{E}(m) \bmod n^2) = B' + m. \tag{16}$$

As indicated in Section 3.3, Beugnon *et al.* show that we need to conserve at least $23 - 16 = 7$ useful bits per coordinate (u), which results in $3u = 21$ MSB per vertex [3]. By respecting this we do not compromise the visual quality of the decrypted 3D object. Therefore, α, the payload of a block B in bits is:

$$\begin{aligned} \alpha &= k - 3u \times b \\ &= k - u \times \frac{2k+1}{23} \text{ bits.} \end{aligned} \tag{17}$$

This results in a payload p, in bits per vertex (bpv) of:

$$\begin{aligned} p &= \frac{\alpha}{b} \\ &= \frac{k}{b} - 3u \\ &= \frac{69k}{2k+1} - 3u \; bpv. \end{aligned} \tag{18}$$

Fig. 6 shows the payload p in bpv as a function of the value of b. We observe that the curve quickly converges towards 13.5 bpv.

3.4.2 Message synchronization by flagging. With our proposed approach, when a message is embedded in the encrypted domain, the corresponding vertex blocks are flagged in order to synchronize this message with all the previously embedded messages. This flagging is necessary in order to extract the embedded message, in particular in the case of multi-embedding. Concretely, the flags indicate which blocks are still available when another message is to be embedded. A flag

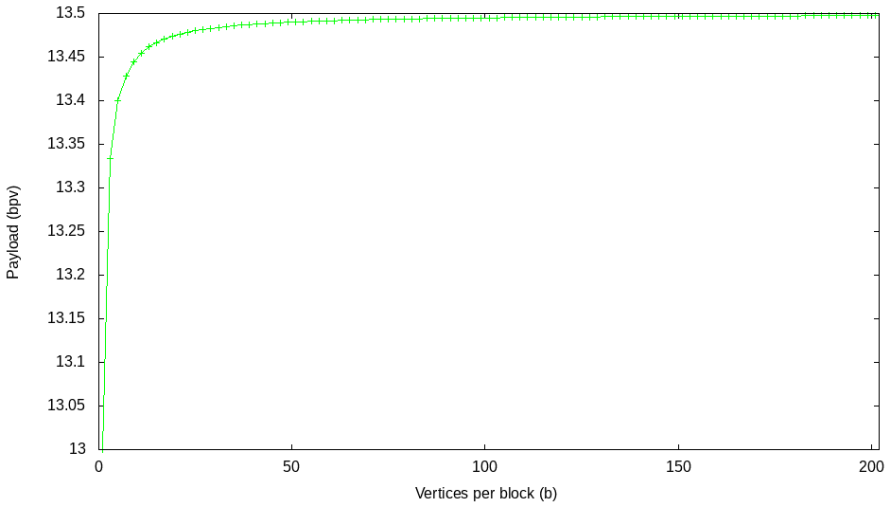


Fig. 6. Payload in bits per vertex as a function of the value of the block size b.

f is embedded using the Paillier probabilistic property which indicates that the encrypted value of the message m is not unique.

During the 3D object encryption, all the encrypted blocks B'_e are then flagged to 0. Based on Eq. (6), we choose r such:

$$B'_e \bmod 2 = 0, \tag{19}$$

where B'_e is the the encrypted watermarkable vertex block where all the flags are initialised at zero.

When a message is embedded in the encrypted 3D object, all the watermarked encrypted blocks B'_{ew} needed to embed this message are flagged as 1, except for the second to last one which is flagged as 0 (so that two consecutive messages can be separated).

To do this we propose using the Paillier cryptosystem's self-blinding property, Eq. (9), where we choose t relatively prime to n such that:

$$(B'_{ew} \times (t^n \bmod n^2) \bmod n^2) \bmod 2 = f, \tag{20}$$

where f is the corresponding flag, with $f \in \{0, 1\}$.

We note B'_{ew_f} the flagged watermarked encrypted block and O_{ew_f} the corresponding flagged watermarked encrypted 3D object. The complexity of our method can be expressed by the probability of choosing the correct r and t respectively so that a modulus 2 results in f:

$$P(X = f) = \frac{1}{2}. \tag{21}$$

3.5 3D object decryption and message extraction in the plain-text domain

In this section, we present the reconstruction of the 3D object and then the extraction of the embedded messages in the plain-text domain. Fig. 7 shows an overview of the decryption and the message extraction steps. The flagged watermarked encrypted 3D object O'_{ew_f} is decrypted using the private key (μ, λ) (Eq. (7)) to give us the reconstructed watermarked 3D object O_w. We note that the data receiver needs only the private key and no other additional information in order to decrypt the 3D object, as the block size is determined by the key size. For each flagged watermarked

encrypted block B'_{ewf}, we obtain a decrypted watermarked block:

$$B_w = \mathcal{D}(B'_{ewf}) = L({B'_{ewf}}^{\lambda} \bmod n^2) \times \mu \bmod n, \tag{22}$$

where $\mathcal{D}(\cdot)$ is the Paillier decryption function.

In parallel to the decryption, the flag extraction from the flagged watermarked encrypted 3D object is performed for each block:

$$f = B'_{ewf} \bmod 2, \tag{23}$$

which allows us to generate a binary location map that indicates which blocks contain messages.

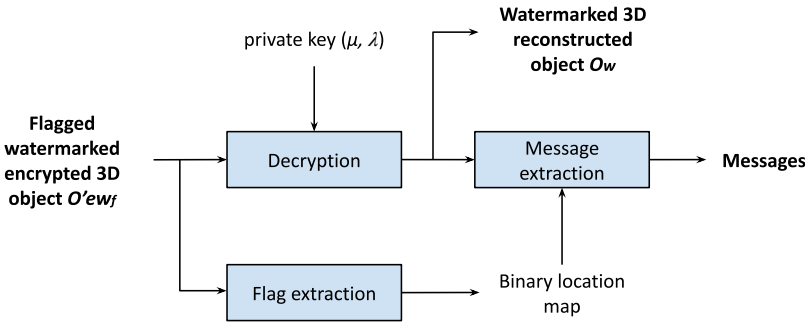


Fig. 7. The decryption and message extraction phases.

All the messages can then be extracted from O_w and a binary location map generated from the extracted flags as illustrated in Fig. 7.

Fig. 8 illustrates the reconstruction of a watermarked vertex block B_w, which is retrieved from Eq. (7). The decryption of the $2k+1$ bits of the block B'_{ewf} results in the original k MSB of the block B'. These bits replace the k MSB in the encrypted vertex block to construct B_w.

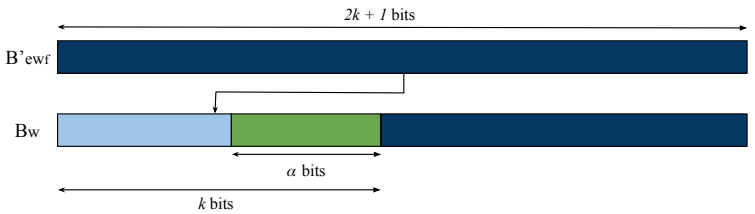


Fig. 8. Decryption of a block B'_{ewf} in order to reconstruct a watermarked block B_w in the plain-text domain.

We extract the α LSB among the k MSB of the vertex block to retrieve the original message segment m.

4 EXPERIMENTAL RESULTS

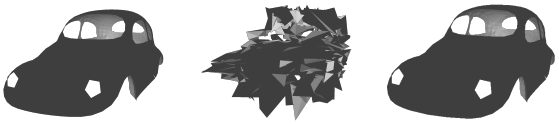
In this section, we present experimental results obtained with our method. First, in Section 4.1, we analyze if the key choice and the block size have an effect on the visual degradation of the decrypted 3D object. In Section 4.2, we present results on a large dataset and in Section 4.3, we compare our method with existing state-of-the-art methods. Finally, in Section 4.4, we present an application of our method to a real-life scenario.

In order to be secure and for real life applications, we need a public key (n, g) where the size of n is at least an estimated 1000 bits[1]. Therefore, we group the vertices into blocks of size $b = 29$ vertices per block and so we have $69 \times 29 = 2k + 1$, which means that $k = 1000$. The value of n is therefore constrained by $2^{1000} \leq n < 2^{1000.5}$. Thus, n is represented by 1001 bits.

We note O the original 3D object, O' the watermarkable 3D object, O'_e the encrypted watermarkable 3D object, O'_{ew} the watermarked encrypted 3D object, O'_{ew_f} the flagged watermarked encrypted 3D object and O_w the watermarked decrypted 3D object.

4.1 Key and block size analysis

Fig. 9a illustrates the original 3D object *Beetle*, Fig. 9b represents *Beetle* when it is encrypted and watermarked with messages with a payload of 13.5 *bpv* and Fig. 9c represents the watermarked reconstruction.



(a) Original 3D object *Beetle* (b) Watermarked encrypted 3D object *Beetle* (c) Reconstructed watermarked 3D object *Beetle*

Fig. 9. Obtained results when the 3D object *Beetle* is watermarked with a payload of 13.5 *bpv* (block size of 29 vertices).

The 3D object *Beetle*, Fig. 9a, has been encrypted, watermarked and then decrypted using 50 different keys of 1001 bits (corresponding to blocks of 29 vertices) drawn at random from a list of eligible keys. Table 1 presents the obtained statistical results between the watermarked decrypted 3D objects O_w and the original 3D object O.

Table 1. Comparison between 50 watermarked decrypted instances of the 3D objects *Beetle* O_w and the original 3D object *Beetle* O.

Beetle (O, O_w)	RMSE (10^{-3})	Hausdorff (10^{-3})
Mean	0.6933	1.741
St. Deviation	0.00169	0.000185
Median	0.6934	1.739
Minimum	0.6920	1.713
Maximum	0.6943	1.774

From the standard deviations of the RMSE and Hausdorff distances which are of the order 10^{-6} and 10^{-7} respectively, we can conclude that the key does not influence the quality of the watermarked decrypted 3D objects. We can also note that there are no outliers, since the minimum and maximum values are very similar to one another. The minimum RMSE value is 0.6920 10^{-3}

[1]Size of 1000 bits is just an example to illustrate our method in this paper. We can apply our method with much larger key sizes.

compared to the maximum value of 0.6943 10^{-3}, and the minimum Hausdorff distance is 1.713 10^{-3} whereas the maximum is 1.774 10^{-3}.

Because of the self-blinding homomorphic property, when we embed a message segment m, the decrypted value of the vertex block watermarked with m does not change. Therefore the embedding does not affect the quality of the watermarked decrypted 3D objects.

Fig. 10 and Fig. 11 illustrate the RMSE and the Hausdorff distances respectively for different values of the block size b ($b = 1, b = 5, b = 9, b = 29$ vertices per block) according to the payload. We can conclude that the block size b does not influence the distortion of the watermarked decrypted 3D object.

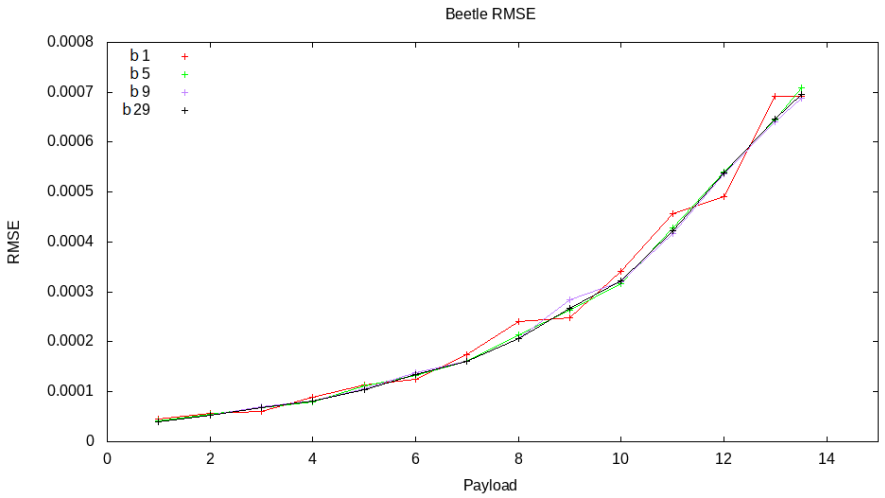


Fig. 10. RMSE between the original 3D object *Beetle* and the reconstructed one as a function of the payload in *bpv* and the block size b vertices per block.

4.2 Performance on a large dataset

We tested our method on the Princeton dataset [27] which consists of 380 different 3D objects. As in Section 4.4, vertices are grouped into blocks of size 29 vertices per block, resulting in a secure key size of 1001 bits.

Table 2 and Table 3 present the Hausdorff distance and RMSE values respectively. We compare the original 3D object O with the encrypted 3D object O'_e, the watermarked encrypted 3D object O'_{ew} and finally the watermarked decrypted 3D object O_w. We also compare O'_e with O'_{ew}.

Table 2. Hausdorff distances obtained when our proposed method is applied to the Princeton dataset [27].

Princeton	O/O'_e	O/O'_{ew}	O'_e/O'_{ew}	O/O_w
Mean	0.4677	0.4686	0.1392	3.769 10^{-3}
St. Deviation	0.1101	0.1100	0.0531	0.443 10^{-3}
Median	0.4833	0.4830	0.1288	3.744 10^{-3}
Minimum	0.1127	0.1124	0.0129	2.580 10^{-3}
Maximum	0.6949	0.6734	0.4181	5.267 10^{-3}

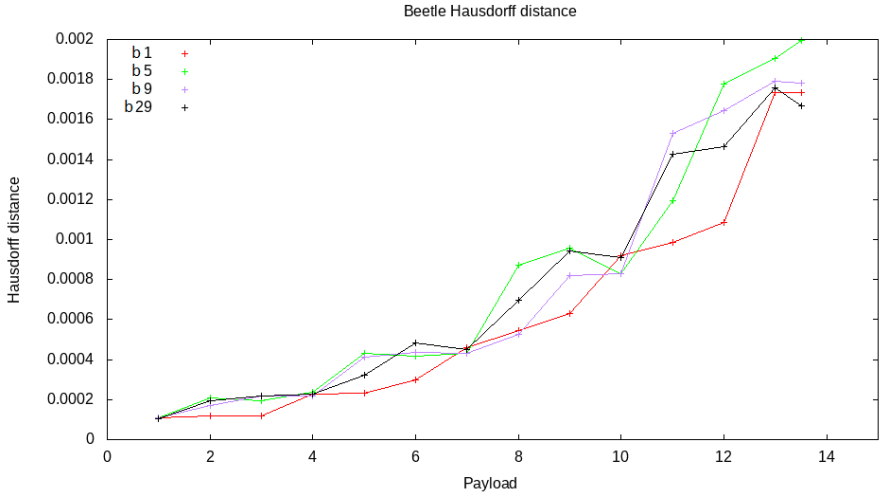


Fig. 11. Hausdorff distance between the original 3D object *Beetle* and the reconstructed one as a function of the payload in *bpv* and the block size b vertices per block.

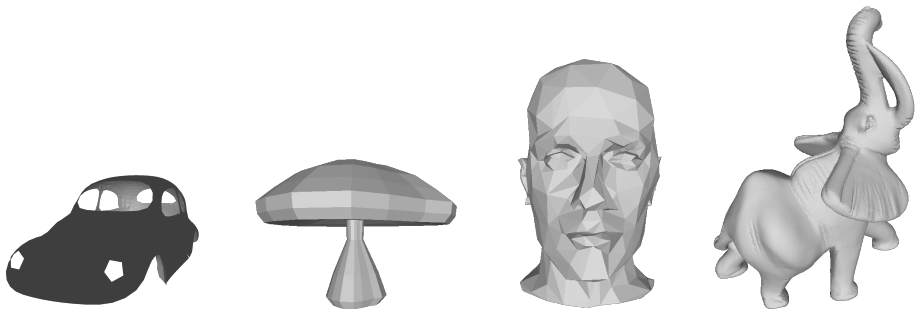
Table 3. RMSE obtained when our proposed method is applied to the Princeton dataset [27].

Princeton	O/O'_e	O/O'_{ew}	O'_e/O'_{ew}	O/O_w
Mean	0.1698	0.1698	0.1668	$1.303\ 10^{-3}$
St. Deviation	0.0290	0.0290	0.0255	$0.199\ 10^{-3}$
Median	0.1636	0.1637	0.1615	$1.263\ 10^{-3}$
Minimum	0.1156	0.1173	0.1166	$0.903\ 10^{-3}$
Maximum	0.2679	0.2671	0.2381	$2.079\ 10^{-3}$

We observe that while O/O'_e and O/O'_{ew} have very similar Hausdorff distances and RMSE, represented in Table 2 and Table 3 respectively, the Hausdorff distance and the RMSE of O'_e/O'_{ew} remain large. Therefore we can conclude that the content of the 3D object remains secure independently of whether there is an embedded message or not. Moreover, the median Hausdorff distance and RMSE of O/O_w are $3.744\ 10^{-3}$ and $1.263\ 10^{-3}$ respectively, which indicates that the resulting watermarked 3D object O_w is similar to the original 3D object O. We note that the mean distances are similar to the median distances. With a maximum Hausdorff distance and RMSE of $5.267\ 10^{-3}$ and $2.079\ 10^{-3}$ respectively, these 3D objects remain visually identical to the original.

4.3 Comparisons with previous work

In this section we compare the results of our method with those of existing work Jiang *et al.* [12], Shah *et al.* [26], Yin *et al.* [34], Lyu *et al.* [15] and Xu *et al.* [33]. In order to compare our obtained results with previous work, we develop our experimentation using four standard test 3D objects: *Beetle* (988 vertices, Fig. 12a), *Mushroom* (226 vertices, Fig. 12b), *Mannequin* (428 vertices, Fig. 12c) and *Elephant* (24,955 vertices, Fig. 12d). For this experiment, in order to make a comparison with other state-of-the-art methods, we encrypt these four 3D objects vertex by vertex. Indeed, while our method can reach a payload of 13.5 *bpv* depending on the block size, we set the block size $b = 1$ vertex per block and the maximum payload for $b = 1$ which is 13 *bpv*.



(a) Beetle (b) Mushroom (c) Mannequin (d) Elephant

Fig. 12. Standard 3D objects used to compare our results with other state-of-the-art methods.

Table 4. Feature comparison between our proposed method and other existing state-of-the-art methods.

	Encrypted Domain					Plain-text Domain	
Methods	Encryption	Size expansion	Auxiliary file	Payload (in *bpv*)	Data error	Marked 3D object	HC Marked 3D object
Jiang *et al.* [12]	Exclusive-or	No	No	0.37	Yes	No	No
Shah *et al.* [26]	Paillier cryptosystem	Yes	No	6 (3+3)	No	Yes	No
Yin *et al.* [34]	Exclusive-or	No	Yes	16.25	No	No	No
Lyu *et al.* [15]	Exclusive-or	No	No	22.83	No	No	No
Xu *et al.* [33]	Exclusive-or	No	Yes	1.07	No	No	No
Proposed $b = 1$	Paillier cryptosystem	No	No	13	No	**Yes**	**Yes**
Proposed $b = 29$	Paillier cryptosystem	No	No	13.5	No	**Yes**	**Yes**

Table 4 presents a feature comparison between our proposed method and five existing state-of-the-art methods Jiang *et al.* [12], Shah *et al.* [26], Yin *et al.* [34], Lyu *et al.* [15] and Xu *et al.* [33]. Our proposed method is the only one to avoid size expansion, an auxiliary file and data error. Note also that our method is able to generate a watermarked 3D object in the plain-text domain.

We note that the payloads of the methods of Jiang *et al.* [12], Yin *et al.* [34], Lyu *et al.* [15] and Xu *et al.* [33] are the average payloads of the four 3D objects, as the payloads of these methods depend on the number of vertices eligible for embedding. The payload of Shah *et al.* is divided into two parts: the payload in the plain-text domain and the possible payload in the encrypted domain. While both the proposed method and the method of Shah *et al.* [26] produce a watermarked 3D object in the plain-text domain, our proposed method has no size expansion and achieves a significantly higher payload. Indeed, the method we propose is the only one which allows us to obtain a high-capacity payload in both plain-text and encrypted domains.

Table 5. Comparison of the payload in both encrypted and plain-text domains, and of the distortion between our method and five significant current state-of-the-art approaches for the four 3D objects *Beetle*, *Mushroom*, *Mannequin* and *Elephant*.

3D object	Methods	Encrypted domain payload (*bpv*)	Plain-text domain payload (*bpv*)	HD (10^{-3})
Beetle	Jiang *et al.* [12]	0.35	0	0.977
	Shah *et al.* [26]	6 (3+3)	3	0.034
	Yin *et al.* [34]	16.51	0	8.60 10^{-3}
	Lyu *et al.* [15]	23.55	0	8.66 10^{-3}
	Xu *et al.* [33]	0.98	0	0.866 10^{-3}
	Proposed	1	1	0.108
	Proposed	7	**7**	0.461
	Proposed	13	**13**	1.73
Mushroom	Jiang *et al.* [12]	0.45	0	0.960
	Shah *et al.* [26]	6 (3+3)	3	0.400
	Yin *et al.* [34]	16.72	0	8.10 10^{-3}
	Lyu *et al.* [15]	21.76	0	8.12 10^{-3}
	Xu *et al.* [33]	1.34	0	75.3 10^{-3}
	Proposed	1	1	0.209
	Proposed	7	**7**	0.881
	Proposed	13	**13**	3.18
Mannequin	Jiang *et al.* [12]	0.34	0	1.01
	Shah *et al.* [26]	6 (3+3)	3	0.370
	Yin *et al.* [34]	13.66	0	4.00 10^{-3}
	Lyu *et al.* [15]	18.05	0	4.00 10^{-3}
	Xu *et al.* [33]	0.95	0	4.00 10^{-3}
	Proposed	1	1	0.655
	Proposed	7	**7**	2.70
	Proposed	13	**13**	8.04
Elephant	Jiang *et al.* [12]	0.34	0	1.08
	Shah *et al.* [26]	6 (3+3)	3	0.0339
	Yin *et al.* [34]	18.12	0	8.60 10^{-3}
	Lyu *et al.* [15]	27.96	0	8.64 10^{-3}
	Xu *et al.* [33]	1.02	0	8.66 10^{-3}
	Proposed	1	1	0.149
	Proposed	7	**7**	0.543
	Proposed	13	**13**	2.82
Average	Jiang *et al.* [12]	0.37 ± 0.05	0	1.01 ± 0.046
	Shah *et al.* [26]	6 (3+3)	3	0.209 ± 0.176
	Yin *et al.* [34]	16.25 ± 1.62	0	(7.325 ± 1.93) 10^{-3}
	Lyu *et al.* [15]	22.83 ± 4.12	0	(7.36 ± 2.25) 10^{-3}
	Xu *et al.* [33]	1.07 ± 0.18	0	(22.21 ± 35.54) 10^{-3}
	Proposed	1	1	0.280 ± 0.219
	Proposed	7	**7**	1.15 ± 0.911
	Proposed	13	**13**	3.94 ± 2.43

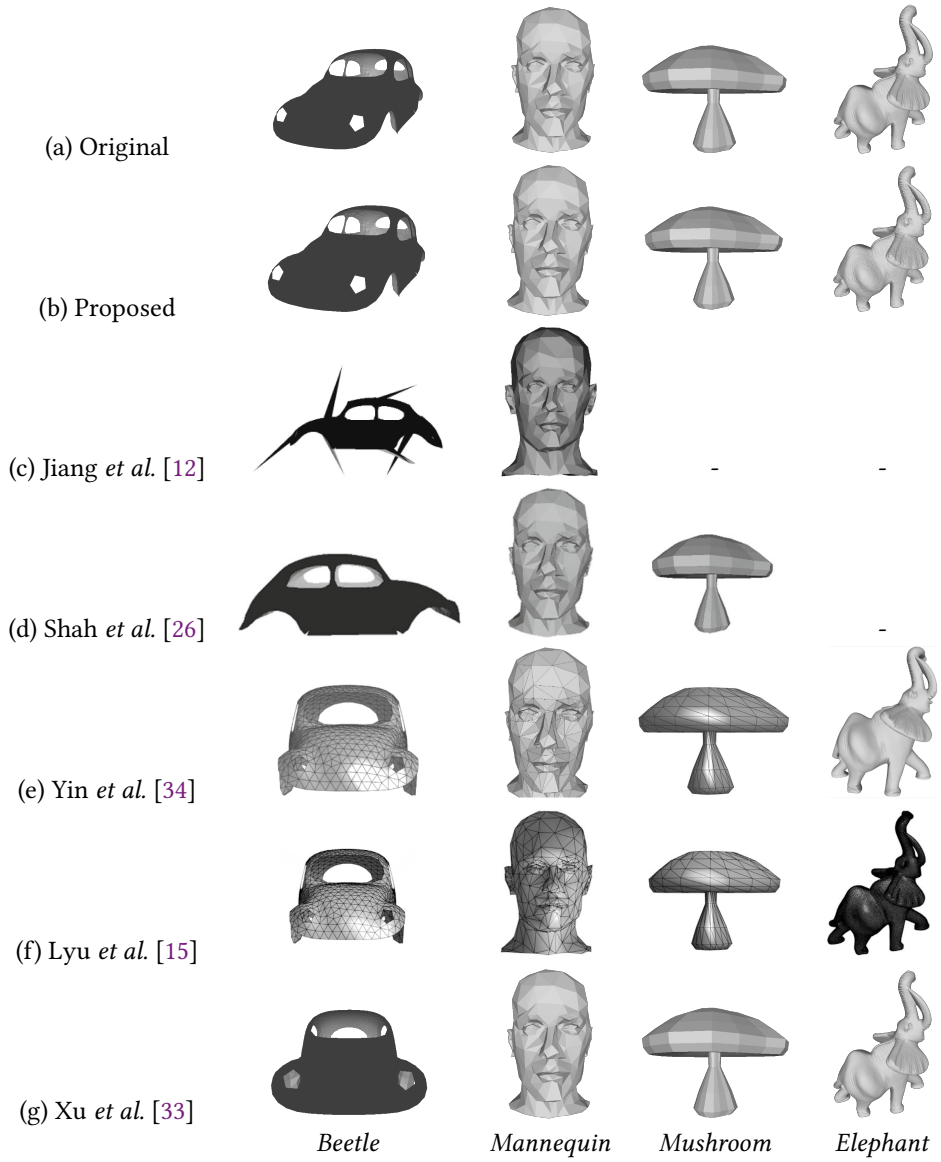


Fig. 13. Visual results of *Beetle*, *Mannequin*, *Mushroom* and *Elephant* with the proposed method compared to current state-of-the-art methods [12, 15, 26, 33, 34].

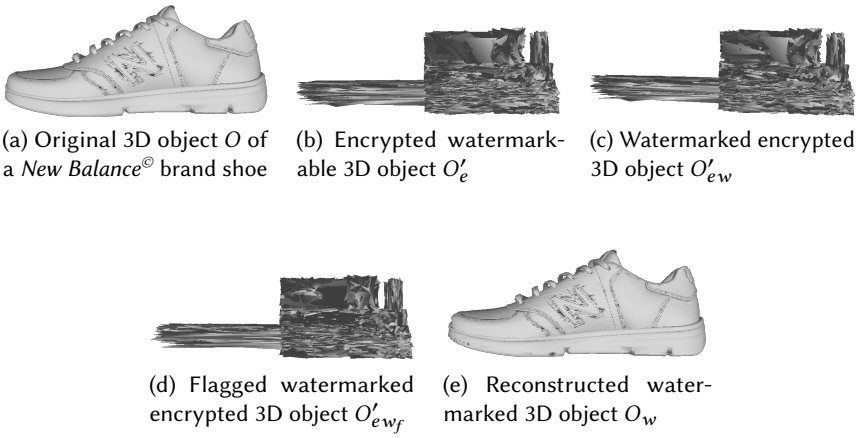
Fig. 13 presents visual results of the proposed method and those of current state-of-the-art methods. Fig. 13.a presents the original 3D objects *Beetle*, *Mannequin*, *Mushroom* and *Elephant*. Fig. 13.b presents the visual results of the proposed method while Fig. 13.c, Fig. 13.d, Fig. 13.e, Fig. 13.f and Fig. 13.g present visual results from previous work (taken from [12, 15, 26, 34] and [33] respectively). We observe that despite a generally higher Hausdorff distance than [15, 26, 34] and [33], like [15, 26, 34] and [33], the results of our proposed method are visually similar to the original 3D objects. However, to the best of our knowledge, we are the only method to achieve a

high-capacity data hiding in the resulting decrypted 3D object, which remains watermarked with hidden messages.

Table 5 represents comparisons between the payloads in both the plain-text and encrypted domains, and the Hausdorff distances of the results of our proposed method and those of the existing state-of-the-art methods. We note that while the state-of-the art methods seek to reconstruct the original 3D object, in the proposed method we retrieve a 3D object which remains watermarked with the hidden messages that were embedded in the encrypted domain. Therefore we do not seek to be statistically identical to the original 3D object. With our method, note that the reconstructed watermarked 3D object remains visually very similar to the original 3D object, as shown in Fig. 15. Our method is the only one that achieves a high payload in both the plain-text and the encrypted domains. With a block size of $b = 1$, once the 3D object is reconstructed, it remains watermarked with a message of up to 13 *bpv*.

4.4 Application to a real-life scenario

In this section, we present results obtained with our method when applied to a real-life scenario. We propose to apply our method on the 3D object of a *New Balance*© brand shoe, which we call *Shoe*, provided by the Stratégies[2] company.



(a) Original 3D object O of a *New Balance*© brand shoe

(b) Encrypted watermarkable 3D object O'_e

(c) Watermarked encrypted 3D object O'_{ew}

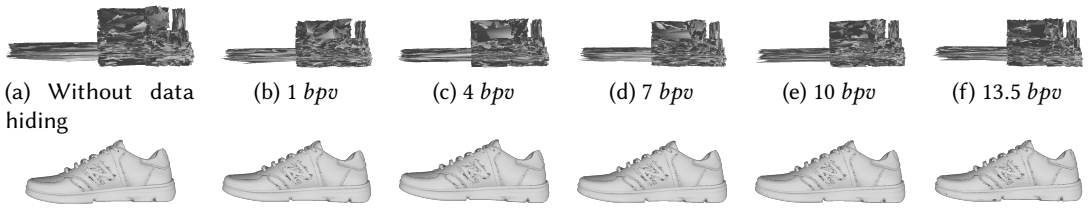
(d) Flagged watermarked encrypted 3D object O'_{ew_f}

(e) Reconstructed watermarked 3D object O_w

Fig. 14. Obtained results on a 3D object O of a *New Balance*© brand shoe, with a payload of 13.5 *bpv* (block size of 29 vertices).

Fig. 14 illustrates the 3D object *Shoe* at different stages of the proposed method, with a payload of 13.5 *bpv*. Fig. 14a presents the original 3D object which a designer creates. Before sending it along the manufacturing chain, the 3D object is encrypted (Fig. 14b) in order to produce an encrypted watermarkable 3D object O'_e. This encrypted 3D object is then sent to multiple different parties in the manufacturing chain over a network. Each time the encrypted 3D object is sent, the server, which does not have the right to access the original 3D object, watermarks the 3D object with information such as the IP address of the sender. Fig. 14c illustrates this watermarked encrypted 3D object. The server also flags the watermarked 3D object in Fig. 14d so that other information can be hidden. We observe that the encrypted 3D object's content remains secure after each message is embedded. Finally, the 3D object is decrypted, resulting in a watermarked 3D object presented in

[2]Stratégies (https://www.romans-cad.com/)



(a) Without data hiding (b) 1 *bpv* (c) 4 *bpv* (d) 7 *bpv* (e) 10 *bpv* (f) 13.5 *bpv*

Fig. 15. Comparison between the watermarked encrypted 3D object O'_{ew_f} and the corresponding reconstructed watermarked 3D object O_w according to the payload from 1 *bpv* to 13.5 *bpv*.

Fig. 14e. We note that the original 3D object O, Fig. 14a, and the resulting watermarked 3D object O_w, Fig. 14e, are visually very similar.

From the 3D object *Shoe*, the top row of Fig. 15 illustrates the watermarked encrypted 3D object O'_{ew} according to the payload from 1 *bpv* to 13.5 *bpv*, while the bottom row illustrates the corresponding watermarked decrypted 3D object O_w. We observe that while the content of the 3D object is secure when encrypted, there are no visual differences between the resulting watermarked decrypted 3D objects.

Table 6. Hausdorff distance measurements when our proposed method is applied to a 3D object of a *New Balance*© brand shoe.

Payload in *bpv*	O/O'_e	O/O'_{ew}	O/O'_{ew_f}	O/O_w
1	0.2332	0.2344	0.2313	$0.1167\ 10^{-3}$
4	0.2317	0.2317	0.2317	$0.2342\ 10^{-3}$
7	0.2306	0.2306	0.2306	$0.4601\ 10^{-3}$
10	0.2317	0.2317	0.2310	$0.9558\ 10^{-3}$
13.5	0.2315	0.2315	0.2305	$1.9337\ 10^{-3}$

Table 6 represents the Hausdorff distances when our method is applied to the 3D object *Shoe*. We observe that for each payload the values of O/O'_e, O/O'_{ew} and O/O'_{ew_f} are similar, which indicates that the content of the 3D object is secure in O'_e, O'_{ew} and O'_{ew_f}, while the content remains clear in O_w.

We note that a cryptosystem is IND-CPA$ secure when an adversary cannot make the distinction between an encrypted 3D object and noise. Therefore the proposed encryption method is not IND-CPA$ secure, just as any format compliant method is not IND-CPA$ secure, since the structure must be preserved.

5 CONCLUSION

In this paper we proposed a new high-capacity DH-ED for 3D objects based on the Paillier cryptosystem. We describe a method which conserves the original format and avoids both size expansion and the use of an auxiliary file, while maintaining the visual quality of the 3D object. Our method uses a large key size, which makes it suitable for real life applications. Most importantly, our approach is a method in which the message can be extracted in the plain-text domain, producing a reconstructed 3D object watermarked with up to 13.5 *bpv*. To the best of our knowledge, our method is the only one that achieves a high payload both in the plain-text and encrypted domains. In the encrypted

domain, the watermarked vertex blocks are flagged, which allows us to have multi-embedding in the encrypted domain.

The proposed method could be further improved by ordering the coordinates within the vertex block B according to the ascending order of the three exponents e of the vertex coordinates in Eq. 13. This would lead to less distortion in the case where the same number of bits are not encrypted in every coordinate.

REFERENCES

[1] F. Armknecht, S. Katzenbeisser, and A. Peter. 2013. Group homomorphic encryption: characterizations, impossibility results, and applications. *Designs, codes and cryptography* 67, 2 (2013), 209–232.

[2] S. Beugnon, W. Puech, and J. Pedeboy. 2019. Format-Compliant Selective Secret 3-D Object Sharing Scheme. *IEEE Transactions on Multimedia* 21, 9 (2019), 2171–2183.

[3] S. Beugnon, W. Puech, and J.-P. Pedeboy. 2018. From visual confidentiality to transparent format-compliant selective encryption of 3D objects. In *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 1–6.

[4] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo. 2016. High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation. *IEEE Transactions on Cybernetics* 46, 5 (2016), 1132–1143.

[5] F. Cayre and B. Macq. 2003. Data hiding on 3-D triangle meshes. *IEEE Transactions on Signal Processing* 51, 4 (2003), 939–949.

[6] Y.-C. Chen, C.-W. Shiu, and G. Horng. 2014. Encrypted signal-based reversible data hiding with public key cryptosystem. *Journal of Visual Communication and Image Representation* 25, 5 (2014), 1164–1170.

[7] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. 2007. *Digital Watermarking and Steganography* (2 ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[8] L. Dong, Y. Fang, W. Lin, and H. S. Seah. 2015. Perceptual Quality Assessment for 3D Triangle Mesh Based on Curvature. *IEEE Transactions on Multimedia* 17, 12 (2015), 2174–2184.

[9] M. Fallahpour, D. Megias, and M. Ghanbari. 2009. High capacity, reversible data hiding in medical images. In *2009 16th IEEE International Conference on Image Processing (ICIP)*. 4241–4244.

[10] W. Hong, T. Chen, and H.-Y. Wu. 2012. An Improved Reversible Data Hiding in Encrypted Images Using Side Match. *IEEE Signal Processing Letters* 19 (2012), 199–202.

[11] S. Jeong and J. Sim. 2017. Saliency Detection for 3D Surface Geometry Using Semi-regular Meshes. *IEEE Transactions on Multimedia* 19, 12 (2017), 2692–2705.

[12] R. Jiang, H. Zhou, W. Zhang, and N. Yu. 2018. Reversible data hiding in encrypted three-dimensional mesh models. *IEEE Transactions on Multimedia* 20, 1 (2018), 55–67.

[13] C. Vinoth Kumar, V. Natarajan, and Deepika Bhogadi. 2013. High capacity reversible data hiding based on histogram shifting for medical images. In *2013 International Conference on Communication and Signal Processing*. 730–733.

[14] G. Lavoué and M. Corsini. 2010. A Comparison of Perceptually-Based Metrics for Objective Evaluation of Geometry Processing. *IEEE Transactions on Multimedia* 12, 7 (2010), 636–649.

[15] W.-L. Lyu, L. Cheng, and Z. Yin. 2022. High-capacity reversible data hiding in encrypted 3D mesh models based on multi-MSB prediction. *Signal Processing* 201 (2022), 108686.

[16] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li. 2013. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transactions on Information Forensics and Security* 8, 3 (2013), 553–562.

[17] A. Malik, H. Wang, T. Chen, T. Yang, A. N. Khan, H. Wu, Y. Chen, and Y. Hu. 2019. Reversible data hiding in homomorphically encrypted image using interpolation technique. *Journal of Information Security and Applications* 48 (2019), 102374.

[18] P. Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*. Springer, 223–238.

[19] W. Puech, M. Chaumont, and O. Strauss. 2008. A Reversible Data Hiding Method for Encrypted Images. *Proceedings of SPIE - The International Society for Optical Engineering* 6819 (2008).

[20] P. Puteaux and W. Puech. 2018. An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images. *IEEE Transactions on Information Forensics and Security* 13, 7 (2018), 1670–1681.

[21] P. Puteaux and W. Puech. 2020. A Recursive Reversible Data Hiding in Encrypted Images Method with a Very High Payload. *IEEE Transactions on Multimedia* (2020), 1–1.

[22] P. Puteaux, M. Vialle, and W. Puech. 2020. Homomorphic Encryption-Based LSB Substitution for High Capacity Data Hiding in the Encrypted Domain. *IEEE Access* 8 (2020), 108655–108663.

[23] Y. Qiu, Q. Ying, X. Lin, Y. Zhang, and Z. Qian. 2020. Reversible Data Hiding in Encrypted Images With Dual Data Embedding. *IEEE Access* 8 (2020), 23209–23220.

[24] O. Regev. 2009. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)* 56, 6 (2009), 1–40.

[25] B. Jansen Van Rensburg, P. Puteaux, W. Puech, and J.-P. Pedeboy. 2021. Homomorphic Two Tier Reversible Data Hiding In Encrypted 3D Objects. In *2021 IEEE International Conference on Image Processing, ICIP 2021, Anchorage, AK, USA, September 19-22, 2021*. IEEE, 3068–3072.

[26] M. Shah, W. Zhang, H. Hu, H. Zhou, and T. Mahmood. 2018. Homomorphic encryption-based reversible data hiding for 3D mesh models. *Arabian Journal for Science and Engineering* 43, 12 (2018), 8145–8157.

[27] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. 2004. The Princeton shape benchmark. In *Proceedings Shape Modeling Applications, 2004.* IEEE, 167–178.

[28] C.-W. Shiu, Y.-C. Chen, and W. Hong. 2015. Encrypted image-based reversible data hiding with public key cryptography from difference expansion. *Signal Processing: Image Communication* 39 (2015), 226–233.

[29] K. Wang, G. Lavoue, F. Denis, and A. Baskurt. 2008. A Comprehensive Survey on Three-Dimensional Mesh Watermarking. *IEEE Transactions on Multimedia* 10, 8 (2008), 1513–1527.

[30] H.-T. Wu, Y.-M. Cheung, and J. Huang. 2016. Reversible data hiding in Paillier cryptosystem. *Journal of Visual Communication and Image Representation* 40 (2016), 765–771.

[31] H.-T. Wu, Y.-M. Cheung, Z. Yang, and S. Tang. 2019. A high-capacity reversible data hiding method for homomorphic encrypted images. *Journal of Visual Communication and Image Representation* 62 (2019), 87–96.

[32] S. Xiang and X. Luo. 2018. Reversible Data Hiding in Homomorphic Encrypted Domain by Mirroring Ciphertext Group. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 11 (2018), 3099–3110.

[33] N. Xu, J. Tang, B. Luo, and Z. Yin. 2022. Separable Reversible Data Hiding Based on Integer Mapping and MSB Prediction for Encrypted 3D Mesh Models. *Cognitive Computation* 14 (2022), 1172–1181.

[34] Z. Yin, N. Xu, and F. Wang. 2019. Separable Reversible Data Hiding Based on Integer Mapping and Multi-MSB Prediction for Encrypted 3D Mesh Models. *arXiv* (2019), arXiv–1908.

[35] X. Zhang. 2012. Reversible data hiding with optimal value transfer. *IEEE Transactions on Multimedia* 15, 2 (2012), 316–325.

[36] X. Zhang, J. Long, Z. Wang, and H. Cheng. 2015. Lossless and reversible data hiding in encrypted images with public-key cryptography. *IEEE Transactions on Circuits and Systems for Video Technology* 26, 9 (2015), 1622–1631.

[37] S. Zheng, Y. Wang, and D. Hu. 2019. Lossless data hiding based on homomorphic cryptosystem. *IEEE Transactions on Dependable and Secure Computing* (2019).

[38] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang. 2015. Secure reversible image data hiding over encrypted domain via key modulation. *IEEE Transactions on Circuits and Systems for Video Technology* 26, 3 (2015), 441–452.

[39] N. Zhou, M. Zhang, H. Wang, Y. Ke, and F. Di. 2020. Separable Reversible Data Hiding Scheme in Homomorphic Encrypted Domain Based on NTRU. *IEEE Access* 8 (2020), 81412–81424.