Indonesian Journal of Electrical Engineering and Computer Science
Vol. 27, No. 2, August 2022, pp. 959~969
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v27.i2.pp959-969

An asymmetric encryption method for 3D mesh model using elgamal with elliptic curve cryptography

Pongpisit Wuttidittachotti[1], Pornsak Preelakha[2]
[1]Department of Digital Network and Information Security Management, Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
[2]Department of Information Technology, Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

Article Info

Article history:

Received Feb 24, 2022
Revised May 18, 2022
Accepted Jun 3, 2022

Keywords:

3D mesh model
Asymmetric encryption
Elliptic curve
Fischer-Yates shuffling
Entropy

ABSTRACT

The 3D mesh (Polygon mesh) model has been widely used in multiple computer technology fields such as computer graphic design and modern 3D animation. 3D mesh repositories were created to support the contribution of many 3D artist-designers and have become an important data source. This research is aimed at introducing asymmetric encryption for a 3D mesh model to improve encryption using elgamal elliptic curve cryptography with Fischer-Yates shuffling. The researchers evaluated the performance of the proposed model using Entropy, mean squared error (MSE), and peak signal noise ratio (PSNR) as evaluation matrices. The results of a decrypted model using our approach with a double-precision floating point showed zero means squared error and infinite value of PSNR.

Corresponding Author:

Pornsak Preelakha
Department of Information Technology, King Mongkut's University of Technology North Bangkok
1518 Pracharat Sai 1 Rd. Bangsue, Bangkok 10800, Thailand
Email: s5907011956134@email.kmutnb.ac.th

1. INTRODUCTION

The advancement of multimedia technology is becoming increasingly more crucial. Large multimedia information is created in the era of digital technology. While the copying and modification of information [1], [2] can be easily done, most of the information is in computer graphics, digital images, and animation. The 3D model has been used in education, graphical design, industry, medicine, the military, engineering, computer video game industry, video animation, and virtual reality [3], [4]. The 3D model contains several types of models such as the 3D point cloud model [5]-[7], the 3D computer-aided design (CAD) model [8], [9] and the 3D mesh model [10] for different purposes. In this research, the researchers worked with 3D mesh models.

The 3D mesh model is a format that uses a triangle polygon called polygon mesh, consisting of vertices (point cloud) and facets (polygon). Typically, a 3D mesh model is a huge model in which each vertex is connected by the line of a triangle polygon on each side to form a 3D mesh model. In spite of the popularity of the 3D mesh model, the security of the model is a challenging problem in the medical field, industry, and national security. The information on the 3D model must be confidential for any third party to prevent potential and severe consequences. For example, the weapon prototype models in the military must be encrypted before being transferred over the intranet for security purposes so that national security will not be affected [1], [4], [11]-[14].

Several research studies have focused on 3D mesh encryption to enhance encryption security. In 2013, geometry preserving encryption (GPE) was proposed by Eluard *et al.* [15] in both point shuffling (PS) protection and coordinate shuffling (CS) protection. Each of the proposed methods uses PS and CS to make the 3D mesh model visualize the intended information. After performing these methods, the 3D mesh model becomes chaotic as a result of using the Entropy metric to analyze the encryption of the original model. Security cannot be ensured by these methods because the research does not specify the exact random permutation algorithm. Thus, the bit security level could not be estimated directly. Information might be leaked if vulnerable shuffling algorithms are chosen. In 2017, the 3D mesh encryption using advanced encryption standard (AES) was proposed by Sayahi *et al.* [16]. This method converts mesh using ASCII and constructs vectors to find the wavelet coefficient. The encryption result is visualized in a spherical coordinate system and the mesh connection is performed using the least significant bit (LSB) method. In 2018, Pham *et al.* [17] developed the Marc method by proposing an encryption method using a triangular matrix construction from vertices and facet data. Then a discrete cosine was transformed to convert the matrix into the frequency domain. The research used the Entropy matrix to analyze the encryption of the models and found that the proposed method offers more chaos than Marc's method [15]. In 2019, Benson *et al.* [18] proposed an encryption method by converting the coordinates x, y, and z. The model is split into vertices and facets which separate the 2D and 3D images from each other using amold cat map to change the permutation or substitution. This method used symmetric encryption which might be unsafe for the encrypted information. In 2019, Liang *et al.* [14] developed an encryption method from previous research [16]. They proposed that the encryption should use a discrete cosine transform. The encryption was applied using asymmetric encryption and Rivest-Shamir-Adleman (RSA) algorithm to increase the entropy value and the chaos of the encrypted model in spite of the requirement of for a greater bit size to make the 3D mesh model encryption safer.

In this research, the researchers developed the method proposed by Liang *et al.* [14]. The asymmetric encryption mechanism was maintained. The encryption algorithm was changed to elliptic curve cryptography (ECC) which had been proved to offer a higher security level than the RSA algorithm with an equal bit size [19]-[21]. Entropy was used to analyze and calculate the encryption efficiency. The higher entropy resulted in higher security. Thus, the researchers proposed asymmetric encryption using the elgamal elliptic curve cryptography with Fischer-Yates shuffling to enhance the security of the 3D mesh model by considering the entropy and the reconstruction quality from mean squared error (MSE) and peak signal to noise ratio (PSNR) calculations. The rest of the paper is organized as: section 2 is background knowledge, section 3 is methodology, section 4 is results and discussion, and section 5 is conclusion.

2. BACKGROUND KNOWLEDGE

In this article, a 3D mesh model encryption for security is focused on. To ensure the validity of this approach, the researchers reviewed the related literature before the implementation. This research discussed 3 theories of experiment design, including ECC, elgamal with elliptic curve (EEC), and Fischer-Yates shuffling. In section 2.1, we calculate the coordinates between two points of an elliptic curve. Section 2.2 discusses the process of encryption using the Diffie-Hellman key. The Fischer-Yates shuffle algorithm is shown in section 2.3.

2.1. Elliptic curve cryptography

ECC is the asymmetric encryption proposed by Neal Koblitz and Victor S. Miller in 1985. The algorithm of ECC is derived from the elliptic curve [19.] $y^2 = x^3 + ax + b$ where a and b are a positive integer of coefficients from 0 to p. Thus, the value of a and b can be considered by the following constraint $4a^3 + 27b^2 \, mod \, p \neq 0$ where$p > 3$ [22]-[26].

An elliptic curve over a finite field is used to validate a coordinate on a curve $y^2 = x^3 + ax + b$ where the addition (addition and doubling of elliptic points) of any 2 points can be carried out by $P_1 = (x_1, y_1)$ and$P_2 = (x_2, y_2)$ coordinates on an elliptic curve and $P_1 \neq P_2$. Thus, the addition of those points is $P_1 + P_2 = P_3 = (x_3, y_3)$ and $x_3 = (\lambda^2 - x_1 - x_2) \, mod \, p$, $y_3 = (\lambda(x_1 - x_3) - y_1) \, mod \, p$ where $\lambda = \frac{y2-y1}{x2-x1}$, if $P_1 \neq P_2$, and $\lambda = \frac{3x_1^2+a}{2y_1}$, if $P_1 = P_2$ the subtraction of the given 2 coordinates can be calculated by letting$P_1 = (x_1, y_1)$ and$P_2 = (x_2, y_2)$be the coordinates on an elliptic curve $P_1 - P_2 = P_1 + (-P_2)$ where $-P_2 = (x_2, y_2) \, mod \, p$ point multiplication of integer can be calculated by letting P be any given point on a curve. The multiplication of the point and any given integer is $kP = \underbrace{P + P + P + \ldots + P}_{ktimes}$.

2.2. Elgamal with elliptic curve

The elgamal encryption system is based on asymmetric cryptography. The process of encryption using the Diffie-Hellman key exchange protocol was proposed by Taher Elgamal in 1985 [27], [28], as shown in (1) and (2). The encryption algorithm is described as:

a. Define a curve $E: Y^2 = x^2 + ax + b$.
b. Choose a finite field by prime p.
c. Choose a point G (x, y) on E.
d. Choose a secret n.
e. Compute $B = nG$ *(public key)*.
f. Choose random shared secret $k \in \mathbb{Z}$.
g. Compute ciphertext $C1$ and $C2$ from plaintext M.

$$C1 = kG \tag{1}$$
$$C2 = M + kB$$

The decryption equation is described as (2).

$$M = C2 - nC1 \tag{2}$$

2.3. Fischer-Yates shuffling

The toughness of random permutation is the biased distribution of each permutation. In the naïve shuffle algorithm, the researchers considered every single index of an array and performed swapping for two indices that randomly sampled from zero to array length range in each iteration. The problem was that the histogram of permutation possibilities was biased and no uniform distribution was found which ideally suited the best distribution for randomness.

Fischer-Yates Shuffling solved this problem by limiting the sampling range for each iteration. By reducing the range of sampling by one in each inverse iteration, this approach produced a uniform distribution-like for each permutation as shown in Algorithm 1 [29].

```
Algorithm 1. Fischer Yates shuffle
FUNCTION FISCHERYATESSHUFFLE (DECK: LIST<CHAR>):
 FOR I IN RANGE (DECK.LENGTH TO 0)
 INDEX = RANDOMINTEGER (0, I+1)
 SWAP DECK [INDEX] AND DECK [I]

 RETURN DECK
```

3. RESEARCH METHOD

In this research, the 3D mesh encryption process is presented using the ElGamal elliptic curve cryptography with Fischer-Yates shuffling. This section presents the experimental results of the two methods to find the encryption process and the decryption process, respectively. In section 3.1, the researchers show the 7 steps of the encryption process, including: i) construct a triangle matrix from vertices and facets; ii) convert the coordinate values in the triangle matrix to bytes; iii) generate key pairs from the E222 curve; iv) encode byte array values in the triangle matrix to the point on the curve format; v) create a new vertices matrix from the encrypted triangle matrix; vi) shuffle the vertices matrix with the Fischer Yates shuffling algorithm; and vii) ciphertext point embeddings. In section 3.2, the researchers show the inversion of the encryption process, including 6 steps. The process in the 3D mesh encryption pipeline is shown in Figure 1.

3.1. The encryption process

From Figure 1, the process of the Elgamal ECC and Fischer-Yates shuffling Encryption can be briefly described as:

Step 1. Construct triangle matrix from vertices and facets.

The construction of the triangle matrix is performed by looking up each of the 3 coordinates x, y, z of a particular point for every 3 points of a triangle polygon and constructing a new 3×3 matrix as (3).

$$A_i = \begin{bmatrix} Pi_{x1} & Pi_{y1} & Pi_{z1} \\ Pi_{x2} & Pi_{y2} & Pi_{z2} \\ Pi_{x3} & Pi_{y3} & Pi_{z3} \end{bmatrix} \tag{3}$$

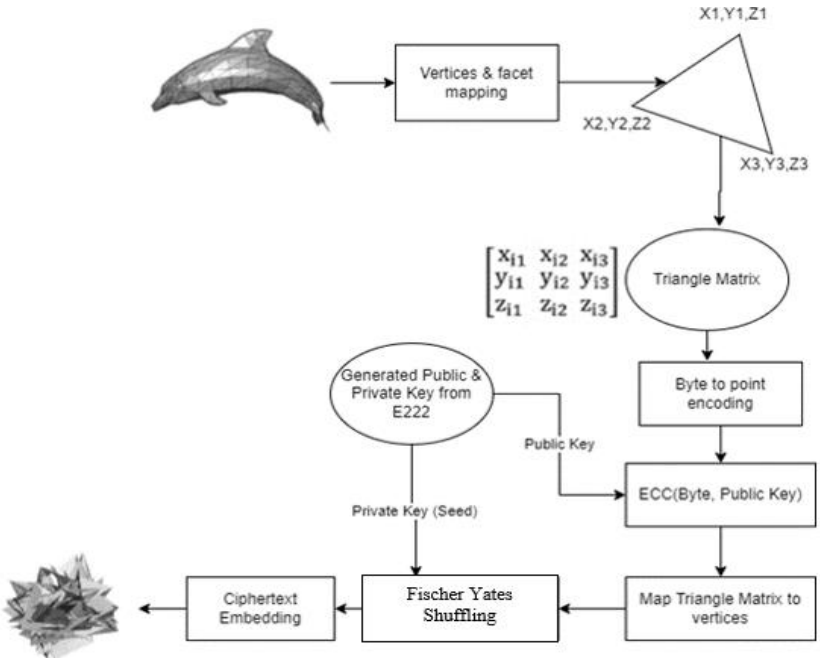


Figure 1. Elgamal ECC+Fischer-Yates shuffling encryption

Step 2. Convert the coordinate values in the triangle matrix to byte array by converting floating-point data to binary according to IEEE 754 floating-point standard.

For example, the researchers can convert 0.60 floating-point values to binary format as shown in (5) and (6).

$$A[0] = \begin{bmatrix} 0.30 & -0.50 & 0.54 \\ 0.60 & 0.40 & -0.67 \\ -0.34 & -0.54 & 0.60 \end{bmatrix} \quad (4)$$

$$0.60_{DEC} = 3F19999A_{HEX} = 00111111\_00011001\_10011001\_10011010_{BIN} \quad (5)$$

$$B(A[0]) = \begin{bmatrix} 3E99999A & BF000000 & 3F0A3D71 \\ 3F19999A & 3ECCCCCD & BF2B851F \\ BEAE147B & BF0A3D71 & 3F19999A \end{bmatrix} \quad (6)$$

Step 3. Generate key pairs from the E222 curve.

Key pairs (a public and private key) are generated by sampling an integer *n* less than the *N* value of the E222 curve *n=n~U(1, N-1)* where *U* is a random function that samples an integer from *1* to *N* from a uniform distribution which gives equal probabilities for each number. Then the researchers use it to find a public key by multiplying the initial point of the E222 curve with a sampled number. The result is *B*, the new point on the curve which is a public key that multiplies to any plaintext point encoded from the bytes values to make a ciphertext from an equation where *G* is equal to the initial coordinates of E222.

Step 4. Encode byte array values in the triangle matrix to the point on the curve format.

Each byte value will be one to one mapped to a set of positive integers. The result of the encoding will be a positive integer for each specific byte of data. The mapping approach includes embedding the size of the byte array in position 0 of the array and appending the values of the random bytes, in case the encoded values are not on the curve and for the convenience of the decoding process. Let *X* be the value of the byte for any plain text. The researchers can perform a byte to point encoding as follows:

In the first step, a plaintext *X* is converted to the byte array format where *x* is an integer of byte array *X*. Then the researchers insert the size of the original byte array at position 0 and convert it to an integer. Then make delta (*δ)* equal zero which can denote whether the current point is on the curve or not. The researchers set up a loop that keeps running *I* delta, not equal to one. After that, let x be the positive integer of byte array *X*. Calculate alpha (α) of the elliptic curve $\alpha = x^3 + ax + b \bmod p$ and a new delta from

$\delta = \alpha^{(p-1)/2 \bmod p}$. If delta is not equal to one, the researchers randomly select a byte value from a uniform distribution and append it to *X*. Then iterate over the loop again until the delta becomes one. The result is a byte array $[s, b_1, b_2, ..., b_2, r_1, r_2, ..., r_n]$ where s is the original size of the array of plaintext, b is any original byte value, and r is any random byte. Then calculate the beta value from $\beta = \alpha \bmod p$. Finally, let y be equal to beta and return to the encoded point $P_m = (x, y)$ which represents the plaintext x in point format on the elliptic curve. We encode every value inside the matrix into the point format as shown in (7).

$$Encode\big(B(A[0])\big) = \begin{bmatrix} (X_1, Y_1) & (X_2, Y_2) & (X_3, Y_3) \\ (X_4, Y_4) & (X_5, Y_5) & (X_6, Y_6) \\ (X_7, Y_7) & (X_8, Y_8) & (X_9, Y_9) \end{bmatrix} \quad (7)$$

We use these points to calculate ciphertexts using the Elgamal ECC algorithm and the E222 curve. For example, the encryption of point X_9, Y_9 is $(X_9, Y_9) \Rightarrow C1, C2$ as shown in (8).

$$\text{ECC (encode (byte } (A[0])\text{), public key)} = \begin{array}{cccc} \text{Matrix No.} & \text{P1} & \text{P2} & \text{P3} \\ i & [C1i_{x1}, C2i_{x1}] & [C1i_{x2}, C2i_{x2}] & [C1i_{x3}, C2i_{x3}] \\ & [C1i_{y1}, C2i_{y1}] & [C1i_{y2}, C2i_{y2}] & [C1i_{y3}, C2i_{y3}] \\ & [C1i_{z1}, C2i_{z1}] & [C1i_{z2}, C2i_{z2}] & [C1i_{z3}, C2i_{z3}] \end{array} \quad (8)$$

Step 5. Construct a new matrix of vertices from the encrpyted triangle matrix by mapping the matrix with the original facets. The result is a vertices matrix with encrypted values as shown in Algorithm 2.

```
Algorithm 2. Triangle matrix to vertices
DEF MAP_TRI_MATRIX_TO_VERT_AR(TRI_MATRIX, FACET):
 VERT = ARRAY WITH SHAPE OF OLD VERTICES
 FOR MAT, INDEX : TRI_MATRIX , MATRIX, FACET:
 FOR POINT, IDX : MAT , INDEX:
 VERT[IDX] = POINT
 RETURN VERT
```

Step 6. Shuffle the vertices matrix with the Fischer Yates shuffling algorithm.

The researchers shuffle the encrypted vertices in the first axis with a random permutation given the seed from the private key generated from ECC. The shuffling algorithm which is used to shuffle the vertices is the Fischer Yates algorithm (Algorithm 1).

Step 7. Ciphertext point embeddings.

After the encryption process, the result for each value in the vertices is a pair of ciphertext (X, Y) of the E222 curve, which is in a positive integer domain ranging from [1, P] where P is a finite field number. To convert the pair of ciphertext points into floating-point (the original datatype of 3D mesh object), the researchers proposed ciphertext point embeddings, which is an approach to derive floating-point values from the points to visualize the encrypted model. We started the process by converting *X* and *Y* coordinates of two ciphertext points of the vertices V into byte array format. Then we performed concatenation in the following order, starting from *X1, Y1, X2,* and *Y2.*

After that, the researcchers simply embedded the byte array lengths for each sub-array in the main byte array at the MSB position with a total of 4 values. Then the array was padded into a specific length that is divisible by 8 to divide the array into groups. The aforementioned processes were performed with every element in vertice V. The result is denoted as Vb. After that, the researchers divided Vb into sub-vertices with the total amount of the length of Vb divided by 8. The result was denoted as Vbs. The values in each sub-vertice are from the equal division of each element in the Vb into bit-group. For each group, the researchers stored those bits in each sub-array from MSB to LSB in order. The byte values in Vbs were converted into a double-precision floating-point format, according to the IEEE 754 standard. Then all the sub-vertices were concatenate along the zero axis to form a new vertices matrix V' which had the length of the Vbs size multiplied by the original vertices size for any 3D model. Finally, the new vertices were assigned into the model object as shown in Algorithm 3.

```
Algorithm 3. Ciphertext point embeddings
FUNCTION CIPHERTEXTEMBED (VERTICES):
 V ← VERTICES
 VB <- []
 FOREACH V IN V
 TEMP <- []
```

```
FOREACH C1,C2 IN V
X1 <- BYTE_AR(C1.X)
Y1 <- BYTE_AR(C1.Y)
X2 <- BYTE_AR(C2.X)
Y2 <- BYTE_AR(C2.Y)
SIZE <- [SIZE(X1), SIZE(Y1), SIZE(X2), SIZE(Y2)]
TEMP.APPEND(SIZE + X1 + Y1 + X2 + Y2 )
VB.APPEND(TEMP)
MAXBYTESIZE = MAXBYTESIZEOF(VB)
IF MAXBYTESIZE >= 64 THEN TARGETSIZE = 128
IF MAXBYTESIZE >= 128 THEN TARGETSIZE = 192
IF MAXBYTESIZE >= 192 THEN TARGETSIZE = 256
VB <- PADEACHELEMENT(VB, PADSIZE=TARGETSIZE)
VBS[(TARGETSIZE/8] <-SPLITTOBYTEGROUP(VB,SPLITSIZE=8)
VBS <- FLOAT64_DECODE(VBS)
V' = CONCATENATE (FOREACH V IN VBS)
RETURN V'
```

3.2. The decryption process

The decryption process of the 3D mesh is an inversion of the encryption process. The processes can be carried out as:

Step 1. Merge the sub-vertices and convert the datatype to construct the vertices of the elliptic curve coordinates.

After performing the ciphertext point embeddings process, a new vertice that has the size of (bytes_array_length/8)*n (P) which is more than the original vertice size n (P) is obtained. Thus, the researchers performed the inversion of the process by merging all the sub-vertices into a single vertice. First, all the floating-point values were converted to byte arrays and then all the bit groups for each sub-vertice were concatenated piecewise. The first sub-vertice has the most significant bit (MSB) and the next sub-vertices have lower significant bits, respectively. After merging the sub-vertices, for each position of the vertice, the researchers divided the byte array of that position into 4 sub-arrays based on the embedded array size at MSB. Then the 4 sub-arrays were converted to an unsigned integer and 2 ciphertexts were constructed from the values $C1(X,Y)$ and $C2(X,Y)$, respectively.

Step 2. Deshuffle encrypted vertice.

In the vertice deshuffling process of the model, the researchers firstly defined a new sequence of an integer index: $P=(0,1,2,...,v-1)$ where v is the size of the vertices of a particular model. Then the sequence P was shuffled using the Fischer-Yates shuffling algorithm with an initial seed from the private key. The vertices according to P, $V'_{P'i}=V_i$ were reordered where V' is the new vertice which was the same size as the old vertice. P'_i is a positive integer at i of the sequence P' which denotes the original index (order) of the vertices.

Step 3. Convert encrypted vertice to triangle matrix.

The encrypted vertice was converted to a triangle matrix in a similar manner to the encryption process by mapping the indices of the facets of the model to the particular index of the vertices. The result was an array with an equal length of the vertices.

Step 4. Decrypt ciphertext of coordinate pairs in encrypted vertices.

Each position of the encrypted triangle matrix consists of 4 values which are the X and Y coordinates of the first and second ciphertexts. The encryption can be performed as (9).

$$M^i_{jk}=C_1+nC_1 \tag{9}$$

where M^i_{jk} is the plaintext (point) of the triangle matrix i at (j, k) position, C_1 is the first ciphertext, C_2 is the second ciphertext, and n is the private key.

Step 5. Point to byte array decoding and floating-point encoding.

After obtaining the plaintext which is a point on the elliptic curve of every decrypted vertex, the researchers decoded points on the E222 curve and then encoded them to the floating-point format accurately. If x is the value to be decoded from any X coordinate $P(X,Y)$, first, convert x to a byte array bX. Then calculate the length of the array nX and the real byte array length nP from the first element of bX which indicates the real byte amount of the array. Take the first byte and bytes within [nP, nX] range out of the bX array. The result is the original byte array of the plaintext.

Step 6. Convert triangle matrix to vertices.

After the researchers obtained the actual byte value of the plain text in the triangle matrix, the values in the triangle matrix with facet indices were mapped to construct the decrypted vertices.

4. RESULTS AND DISCUSSION

The 3D mesh test models including Cow, Bunny, Dragon, Sculpture, Buddha, and Welsh Dragon are from the Computer Graphics Laboratory of Stanford 3D scanning repository. Since this research does not compare the 3D model features encryption method with the previously proposed methods including Éluard *et al.* [15], Pham *et al.* [17] and Liang *et al.* [14]. The researchers used models which have a similar number facets and vertices to those proposed papers shown in Table 1.

Table 1. Model descriptions

Model	Facets	Vertices
Cow	5804	2903
Bunny	69662	34833
Dragon	209227	104855
Sculpture	412669	207285
Buddha	1087474	543524
Welsh Dragon	2210673	1105352

4.1. The security of 3D mesh encryption

The Entropy, PSNR and MSE were used to analyze the security of the proposed algorithm. Entropy can be used to describe the uncertainty and the confusion of information. It is an evaluation matrix of the security of the encrypted model. The entropy value is directly proportional to the security. The researchers made comparisons of the proposed method with that of Éluard *et al.* [15], Pham *et al.* [17], and Liang *et al.* [14] in terms of entropy measurement. The entropy formula consists of the private key length term (k) and the other terms which can be different depending on the proposed encryption algorithm.

In this research, the researchers used the Elgamal ECC of the E222 curve and Fischer Yates Shuffling, which can be used for the vertex encryption. The entropy formula of the proposed method in this study consists of the key length (k), Facet (M), and Vertice (P) as shown in (10).

$$H_{ECC} = k \cdot log_2(k) + (9 \cdot M) \cdot log_2(9 \cdot M) + P \cdot log_2(P) \quad (10)$$

The dataset includes Cow, Bunny, Dragon, Sculpture, Buddha, and Welsh Dragon, which have the number of facets and vertices shown in Table 1. We encrypted the 3D mesh models and calculated the entropy from the (10). The result is displayed in the experiment results in Table 2. The calculated entropies in this study were higher than the candidate methods for every test data. The results of the entropy for each method is explained as follows:

a. Marc's method [15] performs point cloud or vertices shuffling of a model. The entropy formula of the method is $H_{Marc} = k \cdot log(k) + P \cdot log(P)$ where *P* is the vertex size.
b. Pham's method [17] proposes an efficient encryption algorithm with symmetric encryption on frequency domain with a discrete cosine transform. This method only encrypts the last element of each triangle matrix. With encryption, only a single element on the frequency domain prevents the inverse cosine transform process due to the lack of information. Thus, it is unnecessary to encrypt every element of a triangle matrix. The entropy formula can be written as $H_{Pham} = k \cdot log(k) + M \cdot log(M)$ where M is the number of facets.
c. Liang's method [14] applies Pham's method using asymmetric encryption with the use of the RSA algorithm. This method also encrypts every element in a triangle matrix in the frequency domain instead of only the last element. The entropy of the facets is 9 times higher than Pham's method. $H_{Liang} = k \cdot log(k) + 9M \cdot log(9M)$ where M is the number of facets.

Although the method proposed in this research adopts Liang's method by using asymmetric encryption, the researchers used Elgamal ECC over a finite field instead. The security of this algorithm comes from the elliptic curve discrete logarithm problem (ECDLP), which is the best-known algorithm used to solve the problem. This algorithm has exponential time complexity whereas the RSA problem-solving algorithm has sub-exponential time complexity. Thus, the security bits level of ECC are higher with the same number of bits. The unit of measurement is in Bits (N-Bit security requires an attacker to perform operations to break the encryption) [21]. Furthermore, the researchers added the point cloud shuffling using the private key as an initial seed for a random function. Thus, the entropy of this study is higher than Liang's method as shown in the (10).

Table 2. The experimental results

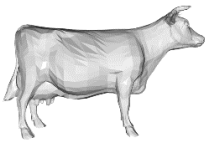
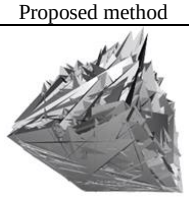
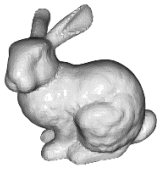
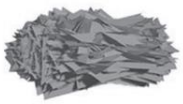
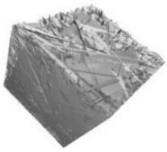
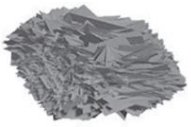
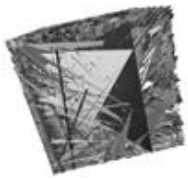
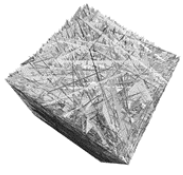
Original model	Encrypted model			
	Proposed method	Liang's method	Pham's method	Marc's method
				
Cow	Entropy=852076	Entropy=818682	Entropy=72566	Entropy=33394
				
Bunny	Entropy=12599529	Entropy=12073963	Entropy=1120728	Entropy=525566
				
Dragon	Entropy=41000119	Entropy=39251343	Entropy=3698026	Entropy=1748775
				
Sculpture	Entropy=84717752	Entropy=81056839	Entropy=7698185	Entropy=3660913
				
Buddha	Entropy=237639746	Entropy=227284534	Entropy=21806626	Entropy=10355210
				
Welsh Dragon	Entropy=504590513	Entropy=482399384	Entropy=46592263	Entropy=22191129

Table 2 shows the cow model with facets 5840 dB and vertices 2903 dB when the model was analyzed using the proposed technique and itshows the value of entropy 852076 dB. When all three methods are compared, it can be seen that the entropy value of the proposed method is higher than that of Liang's, Pham's, and Marc's methods which are 33394 dB, 779510dB, and 818682 dB, respectively. In addition, the Bunny, Dragon, Sculpture, Buddha, and Welsh Dragon models guarantee that the entropy values are higher than that of Liang's, Pham's, and Marc's.

The evaluation of the image reconstruction quality from PSNR is the ratio between the maximum value possible of any signal and the MSE [18], [30]. PSNR is typically used in 2D image reconstruction quality from the compression using any particular algorithm. The large PSNR value indicates higher reconstruction quality. The infinite PSNR value indicates that no reconstruction error occurred due to the divide-by-zero of the denominator which is MSE. The matrix can be applied for the reconstruction quality measurement for 2D images and 3D models.

To apply PSNR with the 3D model encryption, the researchers specified a loss function for the 3D polygon mesh, consisting of vertices and facets. In this research, encryption using the Elgamal ECC algorithm over the vertices of a model was performed. Thus, the researchers calculated the losses based on the vertices.

If V_i is the point cloud at index i for all of the original points, then $\bar{V}_i$ is the point cloud at index i for all of the decrypted points. *N* is the number of vertices of a model in (11).

$$MSE = \frac{1}{N}\sum_{i=1}^{N-1}\left(V_i - \hat{V}_i\right)^2 \tag{11}$$

Due to the varied scale of each model, the range in the vertices values is different in each model. To adjust the vertices values of each model to be on the same scale, the researchers normalize the vertices to [0, 1] range.

$$\bar{V}_i = \frac{V_i - min(V)}{max(V) - min(V)}$$

Then the PSNR can be calculated according to (12).

$$\begin{aligned} PSNR &= 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot log_{10}(MAX_I) - 10 \cdot log_{10}(MSE) \end{aligned} \tag{12}$$

The precise floating-point of the encryption process was chosen after the researchers decoded the point on the elliptic curve to a byte value which affected the error after decryption. Then a test using 16, 32, and 64 bits precision was performed. The MSE and the PSNR are calculated by (11) and (12). The results are shown in Table 3 and Table 4.

Table 3. Mean squared error comparison for each model

Model	Double precision (64 bit)	Single precision (32 bit)	Half precision (16 bit)
Cow	0.0	6.51061752900474e-16	5.395382803997919e-08
Bunny	0.0	1.144405617456389e-15	2.6790357147957838e-08
Dragon	0.0	6.024676162784971e-16	3.276447473207082e-08
Sculpture	0.0	1.048947627449313e-16	2.7124903584596236e-06
Buddha	0.0	3.126163676886491e-16	1.579873271925533e-08
WelshDragon	0.0	8.754025072568139e-16	4.477293439518851e-08

Table 4. PSNR comparison for each model

Model	Double precision (64 bit)	Single precision (32 bit)	Half precision (16 bit)
Cow	∞	151.863778	72.679777
Bunny	∞	149.414200	75.720215
Dragon	∞	152.200663	74.845968
Sculpture	∞	159.792462	55.666318
Buddha	∞	155.049883	78.013777
Welsh_Dragon	∞	150.577922	73.489844

From the results in Table 3 and Table 4, it will be seen that the double-precision floating-point of the vertices causes the MSE to be zero and the PSNR to be infinity for every tested model, indicating that there is no decryption error causing zero MSE, which is the denominator of the PSNR formula, and the limit of PSNR diverges to infinity. The smaller precision bits include the single and half-precision bits which use

32 and 16 bits, respectively. These two precision bits cause a rounding error in the floating-point format after conversion from larger to smaller bits. The PSNR value is directly proportional to the floating-point precision bits, and the error value is inverse. Pham's and Marc's methods research did not suggest evaluating the efficacy of MSE and PSNR. As a result, Pham's and Marc's models couldn't be compared to the model used in this investigation. Pham's technique calculated the model's efficacy based on the amount of time it took to compute the entropy and discovered that the time for processing varied with the number of groups and facets.

Similarly, Marc's technique provided a comparison of PS and CS using random permutation. Liang's technique merely determined the range of possible Alpha values for calculating MSE and LoEC before applying it to MD5 encryption. As mentioned above, this study focuses on the calculation of MSE and PSNR to guarantee that the encryption and decryption are highly secure.

5. CONCLUSION

This study proposed an asymmetric encryption algorithm for 3D mesh using elgamal elliptic curve cryptography and Fischer-Yates shuffling with the data set consisting of Cow, Bunny, Dragon, Sculpture, Buddha, and Welsh Dragon. After the test of the proposed technique and the calculation of entropy, consisting of the key length, the facet, and the vertice, it was found that the proposed method of 3D mesh encryption algorithm achieved higher entropy than the other methods compared in this research. It was found that the proposed algorithm has a higher security level than the other methods. An additional test is a reconstruction quality measurement using PSNR and MSE. Using a double-precision floating-point to store the vertices causes the MSE to be zero and the PSNR to be infinity. This indicates that there no reconstruction losses occur, resulting in highly secure encryption and decryption.

It was suggested that some 3D models might store the vertices at single precision. When it comes to calculating MSE and PSNR for data types with double-precision bit sizes, it might waste storage space. Thus, for practical applications, the floating-point sizes should be selected for the appropriate encryption model.

REFERENCES

[1] M. Ge and R. Ye, "A novel image encryption scheme based on 3D bit matrix and chaotic map with Markov properties," *Egyptian Informatics Journal.*, vol. 20, no. 1, pp. 45-54, 2019, doi: 10.1016/j.eij.2018.10.001.
[2] M. F. Allah and M. M. Eid, "Chaos based 3D color image encryption." *Ain Shams Engineering Journal*, vol. 11, no.1, pp. 67-75, 2020, doi: 10.1016/j.asej.2019.07.009.
[3] M. Xu and Z. Tian, "A novel image cipher based on 3D bit matrix and Latin cubes," *Information Sciences*, vol. 478, pp. 1-14, 2019, doi: 10.1016/j.ins.2018.11.010.
[4] M. A. A. J. A. Mizher, R. Sulaiman, A. M. A. Abdalla and M. A. A. Mizher, "A simple flexible cryptosystem for meshed 3D objects and images," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 6, pp. 629-646, 2021, doi: 10.1016/j.jksuci.2019.03.008.
[5] C. Ji, Y. Li, J. Fan and S. Lan, "A novel simplification method for 3D geometric point cloud based on the importance of point," *IEEE Access*, vol. 7, pp. 129029-129042, 2019, doi: 10.1109/ACCESS.2019.2939684.
[6] A. Khaloo and D. Lattanzi, "Robust normal estimation and region growing segmentation of infrastructure 3D point cloud models," *Advanced Engineering Informatics*, vol. 34, pp. 1-16, 2017, doi: 10.1016/j.aei.2017.07.002.
[7] N. Bold, C. Zhang and T. Akashi, "3D point cloud retrieval with bidirectional feature match," *IEEE Access*, vol. 7, pp. 164194-164202, 2019, doi: 10.1109/ACCESS.2019.2952157.
[8] D. Mouris, C. Gouert, N. Gupta and N. G. Tsoutsos, "Peak your frequency: Advanced search of 3D CAD files in the fourier domain," *IEEE Access*, vol. 8, pp. 141481-141496, 2020, doi: 10.1109/ACCESS.2020.3013284.
[9] F. Lukačević, S. Škec, M. M. Perišić, N. Horvat and M. Štorga, "Spatial perception of 3D CAD model dimensions and affordances in virtual environments," *IEEE Access*, vol. 8, pp. 174587-174604, 2020, doi: 10.1109/ACCESS.2020.3025634.
[10] D. Ma, G. Li and L. Wang, "Rapid reconstruction of a three-dimensional mesh model based on oblique images in the internet of things," *IEEE Access*, vol. 6, pp. 61686-61699, 2018, doi: 10.1109/ACCESS.2018.2876508.
[11] G. Zhou, S. Yuan and S. Luo, "Mesh simplification algorithm based on the quadratic error metric and triangle collapse," *IEEE Access*, vol. 8, pp. 196341-196350, 2020, doi: 10.1109/ACCESS.2020.3034075.
[12] M. A. Mizher, R. Sulaiman, A. M. Abdalla and M. A. Mizher, "An improved simple flexible cryptosystem for 3D objects with texture maps and 2D images," *Journal of Information Security and Applications*, vol. 47, pp. 390-409, 2019, doi: 10.1016/j.jisa.2019.06.005.
[13] Q. Zhang, X. Song, T. Wen and C. Fu, "Reversibility improved data hiding in 3D mesh models using prediction-error expansion and sorting," *Measurement*, vol. 135, pp. 738-746, 2019, doi: 10.1016/j.measurement.2018.12.016.
[14] Y. Liang, F. He and H. Li, "An asymmetric and optimized encryption method to protect the confidentiality of 3D mesh model," *Advanced Engineering Informatics*, vol.42, pp.1-13, 2019, doi: 10.1016/j.aei.2019.100963.
[15] M. Éluard, Y. Maetz, and G. Doërr, "Geometry-preserving encryption for 3D meshes," *Actes de Compression et Représentation des Signaux Audiovisuels (CORESA)*, 2013, pp. 7-12, doi: 10.13140/RG.2.1.3925.6165.
[16] I. Sayahi, A. Elkefi and C. B. Amar, "Join cryptography and digital watermarking for 3D multiresolution meshes security," *Proc. Image Analysis and Processing (ICIAP)*, Oct. 2017, pp 637-647, doi: 10.1007/978-3-319-68548-9_58
[17] N.-G. Pham, K.-S. Moon, S.-H. Lee, and K.-R. Kwon, "An effective encryption algorithm for 3D printing model based on discrete cosine transformation," *Journal of Korea Multimedia society*, vol.21, no.1, pp.61-68, 2018, doi: 10.9717/kmms.2018.21.1.061.
[18] B. Raj, L. J. Anbarasi, M. Narendra and V. J. Subashini, "A new transformation of 3D models using chaotic encryption based on arnold cat map," *Proc. Advances in Internet, Data and Web Technologies (EIDWT)*, Feb. 2019, pp 322-332, doi: 10.1007/978-3-030-12839-5_29.

[19] U. Hayat and N. A. Azam, "A novel image encryption scheme based on an elliptic curve," *Signal Processing*, vol. 155, pp. 391-402, 2019, doi: 10.1016/j.sigpro.2018.10.011.
[20] D. Mahto and D. K. Yadav, "RSA and ECC: A Comparative Analysis," *International Journal of Applied Engineering Research*, vol.12, no. 9, pp. 9053-9061, 2017.
[21] D. Mahto, D. A. Khan and D. K. Yadav, "Security analysis of elliptic curve cryptography and RSA," *Proceedings Of the World Congress on Engineering*, 2016, vol. 1.
[22] Z. K. Obaid and N. F. H. Al Saffar, "Image encryption based on elliptic curve cryptosystem," *International Journal of Electrical and Computer Engineering (IJECE)*, vol.11, no.2, pp. 1293-1302, 2021, doi: 10.11591/ijece.v11i2.pp1293-1302.
[23] N. F. H. Al Saffar and M. R. M. Said, "High performance methods of elliptic curve scalar multiplication," *International Journal of Computer Applications*, vol. 108, no. 20, pp. 39-45, 2014, doi: 10.5120/19028-0047.
[24] S. Deb. and M. Haque, "Elliptic curve and pseudo-inverse matrix based cryptosystem for wireless sensor networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4479-4492, 2019, doi: 10.11591/ijece.v9i5.pp4479-4492.
[25] C. R. Revanna and C. Keshavamurthy, "Hybrid method of document image encryption using ecc and multiple chaotic maps," *International Journal of Recent Technology and Engineering (IJRTE)*, vol.8, no.4, pp. 1615- 1629, 2019.
[26] L. D. Singh and K. M. Singh, "Implementation of text encryption using elliptic curve cryptography," *Procedia Computer Science*, vol.54, pp.73-82, 2015, doi: 10.1016/j.procs.2015.06.009.
[27] Y. Luo, X. Ouyang, J. Liu and L. Cao, "An image encryption method based on elliptic curve elgamal encryption and chaotic systems," *IEEE Access*, vol. 7, pp. 38507-38522, 2019, doi: 10.1109/ACCESS.2019.2906052.
[28] R. A. Haraty, H. Otrok and A. N. El-Kassar, "A comparative study of elgamal based cryptographic algorithms," *ICEIS 2004 - Proceedings of the Sixth International Conference on Enterprise Information Systems*, pp.79-84, 2004.
[29] T. K. Hazra, R. Ghosh, S. Kumar, S. Dutta and A. K. Chakraborty, "File encryption using Fischer-Yates shuffle," *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, 2015, pp. 1-7, doi: 10.1109/IEMCON.2015.7344521.
[30] S. Kumar, N. Agrawal, A. K Jaiswal, N. Nitin and M. Kumar, "Performance analysis of different interpolation technique used for improving PSNR of different images using wavelet transform," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 6, pp. 1367-1372, 2013.

BIOGRAPHIES OF AUTHORS

Dr. Pongpisit Wuttidittachotti is currently an Associate Professor and Head of the Department of Digital Network and Information Security Management at the Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. He received his Ph.D. in Networks, Telecommunications, Systems and Architectures from INPT-ENSEEIHT, in France. He received an outstanding employee award in social service at the university level in 2019, and an outstanding employee award at the faculty level and the university level in 2020. He owns more than 30 recognized certifications, for example, CISSP, CISM, CISA, CRISC, CGEIT, IRCA ISO/IEC 27001:2013 Lead Auditor, COBIT 5 Foundation, COBIT 2019 Foundation, COBIT 2019 Design & Implementation, Data Protection Officer (DPO) etc. So far, Wuttidittachotti has over ten years of working experience covering software development, networks, security, audit, risk management, IT governance, and standards, and compliance. His expertise has been demonstrated as a member of the ISACA Bangkok Chapter Committee since 2015, and as an Accredited Trainer-COBIT® 2019 Foundation for ISACA Bangkok Chapter. He has conducted and published many research articles in information security and related topics. He can be contacted at email: pongpisit.w@itd.kmutnb.ac.th.

Pornsak Preelakha received his B.I.S. in Information Science from Mahasarakham University. He received an M.Sc. in Information Technology from King Mongkut's University of Technology North Bangkok. He is currently studying for his Ph.D. in Information Technology at the Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. He can be contacted at email: s5907011956134@email.kmutnb.ac.th.