

New approach to Generate, Re-Generate, Encode, and Compress 3D structures using polynomial equations

Jimut Bahan Pal^{1*}, Asoke Nath²

^{1,2}Department of Computer Science, St. Xavier’s College (Autonomous), 30 Park Street, Kolkata, India

*Corresponding Author: asokejoy1@gmail.com, Tel.: +033-22876035

Available online at: www.ijcseonline.org

Accepted: 18/Dec/2018, Published: 31/Dec/2018

Abstract— Almost every 3D objects can be represented with the help of a set of equations. When 3D objects are converted to text in terms of equations, any standard cryptographic techniques can be applied to encrypt it. This method is useful to hide classified military secrets, as we compress it to a large amount in the form of text, yet it can be regenerated in mathmod software, after decrypting it using the key. In this study, we propose a new way to formulate polynomial equation of any objects in context by fitting it using brute force techniques. We find that the complexity of the algorithm is too high to be achieved in practical scenarios, which encourages more efficient work in this new field.

Keywords—: Compression, Cryptographic techniques, Generation, 3D, and Polynomial equations.

I. INTRODUCTION

Almost everything we see in this mundane world can be represented using complex polynomial equations. Aren’t they? Most of the simple objects can be represented directly. The objects in 2D can be easily represented using a set of equations and defining the range and domain of x , and y values as shown in **Fig. 1**, and **Fig. 2**.

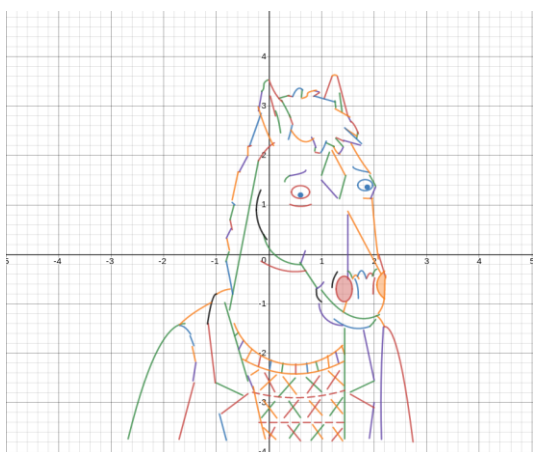


Fig. 1: A 2D representation of DeLauro [1]

We can represent objects in 3D too. Let’s take an example of a simple heart in 3D as shown in **Fig. 3**, and look at the relevant equation carefully. It represents a polynomial equation of 6 degrees.

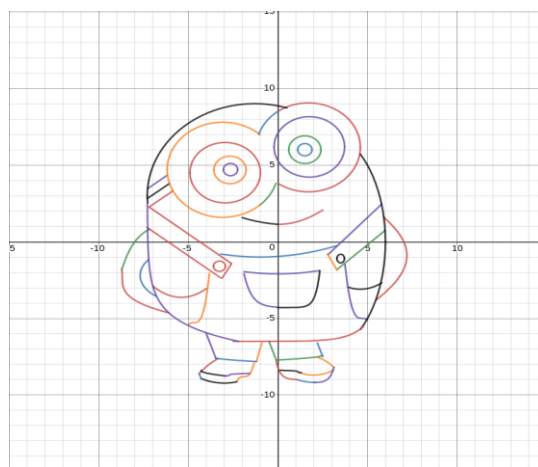


Fig. 2: A 2D representation of Minion [1]

Let’s talk about modelling an object in any standard animation software. There are several modelling softwares, like CAD which has .stl extension. It stores mainly the vertices. Another one being 3DS, now owned by Autodesk Maya. It stores data in the form of binary chunk, similar to XML DOM tree with a .3ds extension. Flimbox or (.fbx) files are also modelled using Autodesk. This can be formatted to represent high quality models with textures.

We are working with vertices, edges and faces of the object to design the model. Now, what is a face? A face is a 2D side of any 3D object. For example, a cube may have 6 faces. It is always necessary to have edges more than 2, i.e., 3 edges to have one face. Now, if we draw a pentagon in a white paper

sheet, it will have one face but should have 5 edges, likewise if we draw a triangle it will have 3 edges and one face, and so on. Similarly, when we work with any modeling software, we will work with face and edges and make necessary changes like a sculptor to change the shape and nature of the object in context. More the number of faces, vertices or edges, more the size of the object. Now, what the .obj (object) files are storing in it? It generally stores the position of edges, vertices in an optimized binary format, with some meta-data about the software, time etc. We can also easily analyze the object file by opening it in any text editor, like vim for opening large files in short time. So, what happens in general case is that those models which are made through the general animation or modeling software are large in size. It takes much memory to store. Here, we presented a new method to compress the existing models, to generate models with the help of equations.

Here we have encrypted some simple objects with the help of equations and represented it with any standard 3D modelling software. Firstly we have shown that even complex models can be represented with the help of equations, then we have encrypted a simple heart shaped structure with the help of software to text equations. We have then encrypted the same with the help of the famous AES algorithm, then decrypted it to show that it is lose less compression. We have also shown other standard encryption techniques and shown why we prefer AES algorithm, we have also developed a brute force algorithm to make complex equations, match it and select the equation of best fit. Further we have analyzed the complexity of the algorithm to find any general structure, and realized that it is too high to be achieved in practical scenarios, so it will encourage more work in this new field to find other methods which will form lose less compression to existing 3D structures to equations. We can use B-spline, T-spline methods, but they may result in loss of 3D data and precision, so some other methods should be devised which will be lose less and be feasible in low hardware resources.

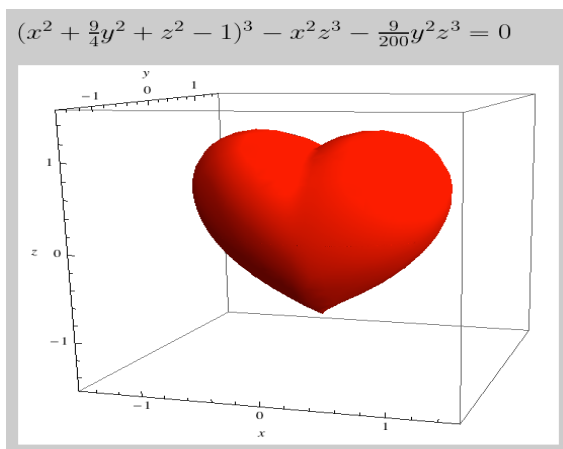


Fig. 3: The 3D shape of a heart with equation [2]

II. RELATED WORK

Innumerable literatures are available in the field of computer graphics, simulation, visualization and real time analysis of pictures to generate, regenerate and fill missing parts using machine learning technologies and mathematical models. Some of the works by different researchers are brought to context which includes fitting of polynomial to take pictures of palm for a very strong biometric security. Furthermore, there has been study of regeneration of missing and broken vertices, creating 3D shapes and structures of a 2D image of an object in context. Researchers found better methods to help solving real time problems like determining the acceleration due to gravity of an irregular shape asteroid.

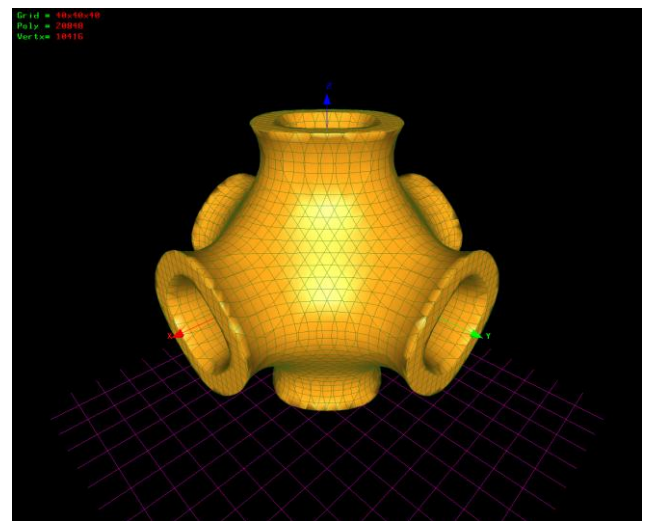


Fig. 4: A 3D model of a complex structure

The property of interpolation, Euclidean, affine invariants and their ability to represent complex objects makes the implicit polynomials represent useful object and data in computer vision [3, 4]. Blane et al. [3] studied and improved the former methods to model the fingers by using implicit polynomial fit strategy, resulting in identifying fingers using biometric methods. They tested it till 5th degree with satisfying results, obviously the higher the degree of polynomial higher is the precision, formulating 3L algorithm, which provides a fast solution to its counterparts, with a better repeatability and numerical stability. It can easily fit polynomial of high degree such as 14th to 18th.

Cost and time could be significantly reduced by procedurally generating the content, leveraging the creativity of the designers, programmers and artists. Quiroz et al. [5] used genetic algorithm to procedurally generate vertex shades of 3D models in web applications. The main problem in robotics, virtual reality, augmented reality, and computer graphics is to determine the proximity relationships between 3D objects cluttered in an environment in real time. The path

planners in robotics [6] computes a dynamically feasible trajectory from starting position to goal, while avoiding obstacles. In game engines, VR, real time simulations, and all objects in a scene shall interact to respond and emulate physics in the real world. Ahmadi et al. [6] studied the problems of spatial analysis of 3D environment using sublevels of polynomials, and performing complex mathematical operations on cloud of data points, to get the knowledge of geometry and surroundings.

Irregular shapes of asteroids may lower the efficiency of calculating the gravitational field that is experienced by it. Hu et al. [7] studied and proposed a method to use Chebyshev polynomial interpolation to adapt and divide the space near the asteroid along spherical co-ordinates. Then it represents gravitational acceleration in each cell. This increases the efficiency in measuring gravitational force accurately. A face recognition method presented by Komorowski et al. [8] is based on sequencing images, which is easy to implement. The use of laser scanners may be expensive, large and may cause damage to human organs. They reconstructed face shape using multi view stereo methods and motion of the object. Since it is based on images only, it is easy to use, and do not require any face model. Human percept objects through semantic reasoning, can forecast about the geometry as well as the 3D shape of the object by limited 2D information. This helps us to interact with the objects easily. Mandikal [9] et al. further studied the reconstruction of image through 3D cloud from a single input image. They showed that when they trained task jointly, it yields better results than training each task individually. Using ShapeNet dataset, they improved results in reconstruction as well as segmentation of objects. A 3D mesh animation of sequences was synthesized with long short term memory (LSTM) blocks and mesh based Convolutional Neural Networks (CNNs) [10]. A Polyharmonic Radial Basis functions (RBF) could be used to reconstruct smooth manifold surfaces from point based cloud data and incomplete meshes could be repaired too as studied by Carr et al. [11].

To enhance navigation, visualization, urban planning of historic and touristic objects, 3D building models becomes important during the past years [12]. It helps to understand scenarios that make complex decisions. Gross et al. studied 3D modeling of buildings from maps by semi-automatic processing of laser scanner elevation of the building using areal imagery. Kholgade et al. presented a method [13], which can perform full range of 3D operations such as scaling, translation, rotation etc. to an object in photograph. It is better than the 2D conventional operation that gives a realistic effect. It takes input from a 2D image and uses publicly available dataset of model to complete the unseen part of the 2D image in context.

III. METHODOLOGY

We can represent any structure in the form of equations. We used mathmod software [14] to represent any objects using equations. The object as shown in Fig. 5 represents an arena. The objects can be represented with a set of equations:

$$\text{iff}(\sqrt{x^2+y^2+z^2}-$$

$$27)<0 \& \text{abs}(x)<(10), \text{ThickIsoExterior}(x,y,z,t), 1)$$

With additional set of functions:

$$\text{Scherk}=\sinh(x)*\sinh(y)-4*\sin(z)$$

$$\text{Scherk2}=\text{Scherk}(x,\sqrt{y^2+z^2})-(16), N*\text{atan2}(z,y), t)$$

$$\text{Thickness2}=(0.6)$$

$$\text{IsoExterior}=\text{Scherk2}(x,y,z,t)$$

$$\text{Dfx2}=(\text{IsoExterior}(x,y,z,t)-\text{IsoExterior}(x+cx,y,z,t))/cx)$$

$$\text{Dfy2}=(\text{IsoExterior}(x,y,z,t)-\text{IsoExterior}(x,y+cy,z,t))/cy)$$

$$\text{Dfz2}=(\text{IsoExterior}(x,y,z,t)-\text{IsoExterior}(x,y,z+cz,t))/cz)$$

$$\text{Rapport2}=(\sqrt{(\text{DFx2}(x,y,z,t)*\text{Dfx2}(x,y,z,t)+\text{DFy2}(x,y,z,t)*\text{Dfy2}(x,y,z,t)+\text{DFz2}(x,y,z,t)*\text{Dfz2}(x,y,z,t))})$$

$$\text{Iso3}=(\text{IsoExterior}(x-$$

$$\text{DFx2}(x,y,z,t)*\text{Thickness2}(x,y,z,t)/\text{Rapport2}(x,y,z,t), y-$$

$$\text{DFy2}(x,y,z,t)*\text{Thickness2}(x,y,z,t)/\text{Rapport2}(x,y,z,t), z-$$

$$\text{DFz2}(x,y,z,t)*\text{Thickness2}(x,y,z,t)/\text{Rapport2}(x,y,z,t,t))$$

$$\text{Iso2}=(\text{IsoExterior}(x+\text{DFx2}(x,y,z,t)*\text{Thickness2}(x,y,z,t)/\text{Rapp}$$

$$\text{ort2}(x,y,z,t), y+\text{DFy2}(x,y,z,t)*\text{Thickness2}(x,y,z,t)/\text{Rapport2}(x,$$

$$y,z,t), z+\text{DFz2}(x,y,z,t)*\text{Thickness2}(x,y,z,t)/\text{Rapport2}(x,y,z,t,t))$$

$$\text{ThickIsoExterior}=(\text{Iso2}(x,y,z,t)*\text{Iso3}(x,y,z,t))$$

And a range of x, y, and z domain values:

$$x = -10.1 \text{ to } 10.1$$

$$y = -28 \text{ to } 28$$

$$z = -28 \text{ to } 28$$

So, if we take this model and sculpt any animation or modelling software, then it would take more than 10 Mb of space depending on the number of vertex and faces to a large extent, but in representing this in terms of equations in text file it takes a less memory to store some bytes of equations. Similarly, we can represent all the 3D objects in Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11, and Fig. 12 using a set of equations.

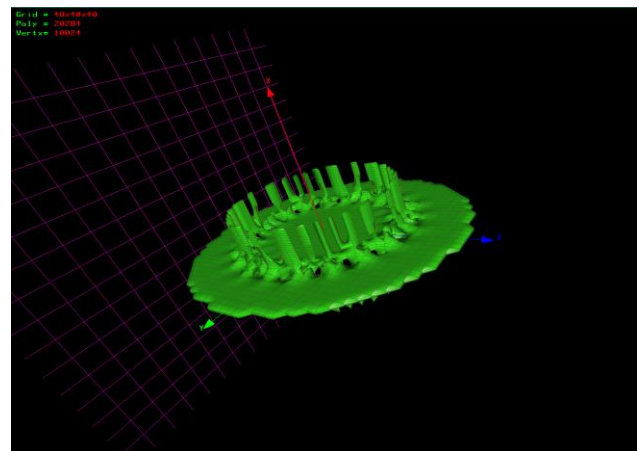


Fig. 5: 3D structure representing arena

Similarly, the broken vase shown in Fig. 9 can be represented using the following set of equations:

$$isoTransform_2(x,y,-z,t)*isoTransform_6(x,y,-z,t)$$

With an additional set of functions:

$$Angle1=atan2(sqrt(x*x+y*y),(-z+cx))$$

$$Angle2=atan2(x,(y+cx))$$

$$CarvinCondition=abs((z-1)-$$

$$0.8*cos(18*Angle2(x,y,z,t)/pi))<10.3/abs((z+3)-$$

$$0.3*cos(18*Angle2(x,y,z,t)/pi+pi/4))<0.5$$

$$Iso=cos(x)*sin(y)+cos(y)*sin(z)+cos(z)*sin(x)$$

$$Torus=(sqrt(x*x+y*y)-3)^2+z*z-1$$

$$Bottom=(x*x+y*y+z*z-1)$$

$$IsoExterior=x*x/3.+y*y/3.-abs(1.5*sin(2*z/pi+0.3)+1.8)$$

$$Dfx2=((IsoExterior(x,y,z,t)-IsoExterior(x+cx,y,z,t))/cx)$$

$$Dfy2=((IsoExterior(x,y,z,t)-IsoExterior(x,y+cy,z,t))/cy)$$

$$Dfz2=((IsoExterior(x,y,z,t)-IsoExterior(x,y,z+cz,t))/cz)$$

$$Rapport2=(sqrt(DFx2(x,y,z,t)*Dfx2(x,y,z,t)+DFy2(x,y,z,t)*Dfz2(x,y,z,t)+DFz2(x,y,z,t)*Dfz2(x,y,z,t)))$$

$$Iso2=(IsoExterior(x+t*Dfx2(x,y,z,t)*Thickness2/Rapport2(x,y,z,t),y+t*Dfy2(x,y,z,t)*Thickness2/Rapport2(x,y,z,t),z+t*Dfz2(x,y,z,t)*Thickness2/Rapport2(x,y,z,t),t))$$

$$ThickIsoExterior=(Iso2(x,y,z,1)*Iso2(x,y,z,-1))$$

$$Iso=cos(x)*sin(y)+cos(y)*sin(z)+cos(z)*sin(x)$$

$$Dfx=((Iso(x,y,z,t)-Iso(x+cx,y,z,t))/cx)$$

$$Dfy=((Iso(x,y,z,t)-Iso(x,y+cy,z,t))/cy)$$

$$Dfz=((Iso(x,y,z,t)-Iso(x,y,z+cz,t))/cz)$$

$$Rapport=(sqrt(DFx(x,y,z,t)*Dfx(x,y,z,t)+DFy(x,y,z,t)*Dfy(x,y,z,t)+DFz(x,y,z,t)*Dfz(x,y,z,t)))$$

$$Iso4=(Iso(x+t*Dfx(x,y,z,t)*Thickness4/Rapport(x,y,z,t),y+t*Dfy(x,y,z,t)*Thickness4/Rapport(x,y,z,t),z+t*Dfz(x,y,z,t)*Thickness4/Rapport(x,y,z,t),t))$$

$$ThickIso2=(Iso4(x,y,z,-1)*Iso4(x,y,z,1))$$

$$isoTransform_2=if((CarvinCondition(x,y,z,t)=0),ThickIsoExterior(x,y,z,t),1)$$

$$Iso6=(Iso(x+t*Dfx(x,y,z,t)*Thickness6/Rapport(x,y,z,t),y+t*Dfy(x,y,z,t)*Thickness6/Rapport(x,y,z,t),z+t*Dfz(x,y,z,t)*Thickness6/Rapport(x,y,z,t),t))$$

$$isoTransform_6=if((CarvinCondition(x,y,z,t)&ThickIsoExterior(x,y,z,t)<0),-ThickIso2(S*x,S*y,S*z,t)*Iso6(x*S,y*S,z*S,-1)*(Iso6(x*S,y*S,z*S,1),1)$$

With x, y, and z domain being:

$$x = -3.5 \text{ to } 3.5$$

$$y = -3.5 \text{ to } 3.5$$

$$z = -5 \text{ to } 4.5$$

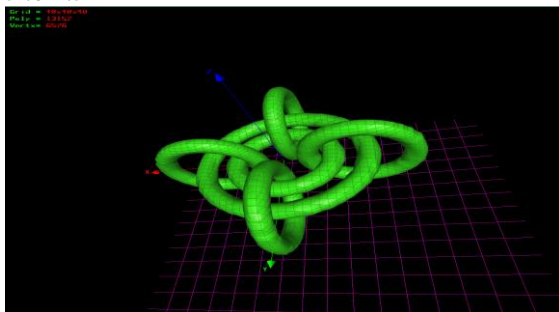


Fig. 6: 3D representation of chain

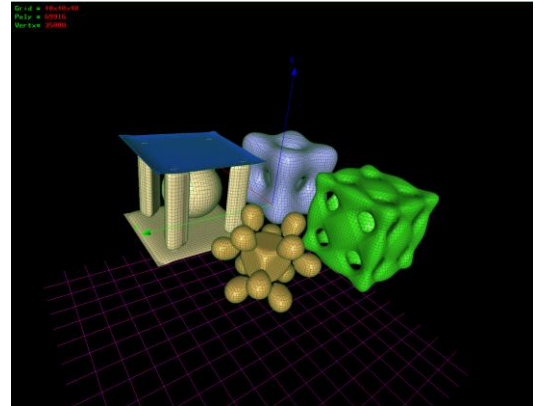


Fig. 7: A composite structure

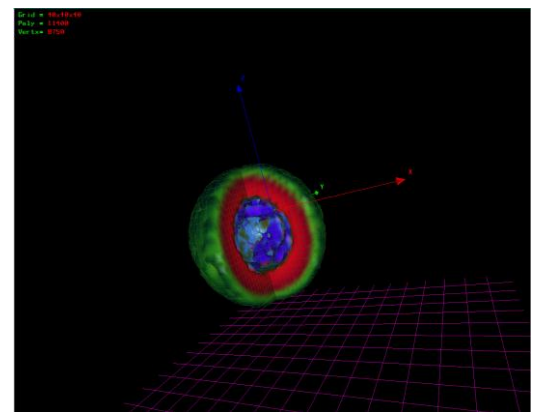


Fig. 8: A geode

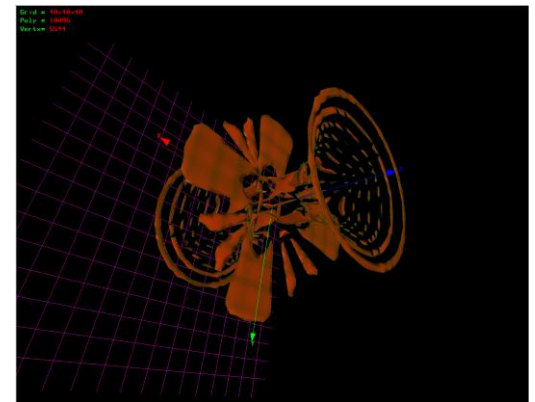


Fig. 9: Implicit Costa, a complex structure

Above two examples simplify the complex structures that may be represented using a set of equations. In this paper we studied the example of a heart, and compared the sizes when it represents terms of equations using plain text, encrypt with AES algorithm and decrypt it using the same to regenerate it using the text. We have also presented here an algorithm to brute force using a model to a complex polynomial equation. We will also discuss the complexity of the model, and its future scope in results and discussion to improving it to a large extent.

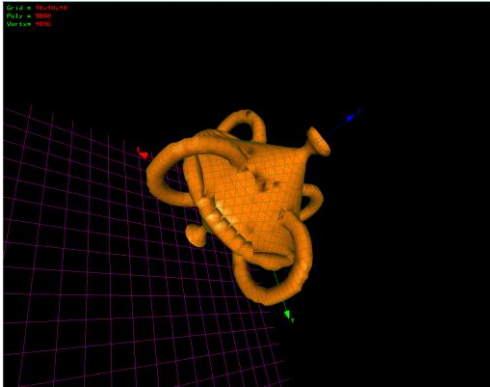


Fig. 10: A lamp

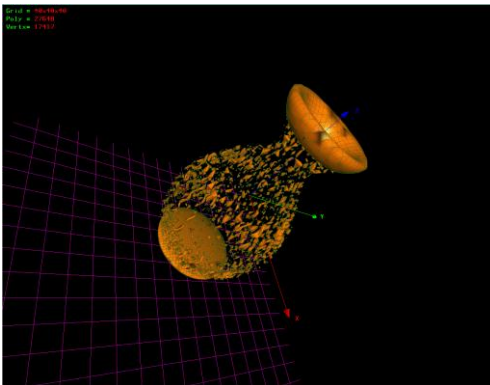


Fig. 11: A broken vase represented in 3D

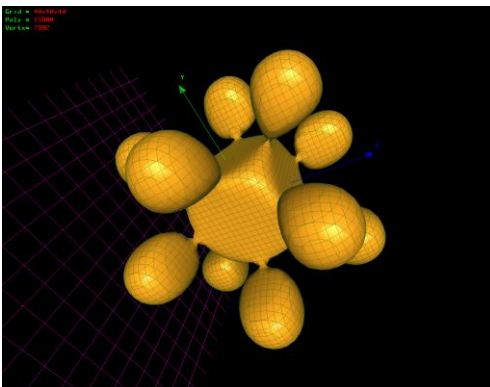


Fig. 12: A virus

A. Generating the model from the equation

The equation of heart is $(x^2 + (9/4)*(y^2) + z^2 - 1)^3 - (x^2)*(z^3) - (9/200)*(y^2)*(z^3)$, we can use this equation to represent it in math-mods as shown in Fig. 13. The equation will take about 75 bytes in plain text. When we sculpt a heart model using Maya, a standard modelling and animation software, its size is increased to 476 KB as shown in Fig. 14. We can thus see that to represent it using Maya, the size has increased to about 6499 times. The only problem with our method is that it doesn't represent a smooth surface but it can be achieved with higher degree of polynomial equations.

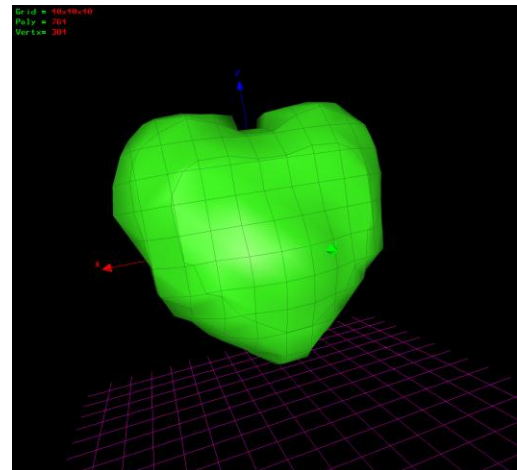


Fig. 13: 3D model of heart using math mod



Fig. 14: 3D rendered image of heart using Maya

Hence, the model can be represented using a polynomial equation, by compressing it to a large extent.

B. Encrypting the equation

Once we converted the model to equation, i.e., in the text form, we can apply it to any standard encryption algorithm for encryption. Some of the standard encryption algorithms [15] are:

1. *Triple Des*: It was designed to replace the Data Encryption Standard (DES) Algorithm. It uses three individual keys of 56 bits each of which adds up to 186 bits. It still serves as encryption algorithm for financial services and other industries.
2. *RSA*: It is an asymmetric algorithm as it has pair of keys. It is the standard for encrypting data sent over the internet, and is a public-key encryption algorithm.

3. *Blowfish*: It replaced DES algorithm. It manages to split messages into blocks of 64 bits and then encrypts them individually.
4. *Twofish*: It is the successor of blowfish. The key can be up to 256 bits in length, and only one key is required as it is symmetric.
5. *Rijndael or Advanced Encryption Standard (AES)*: It is trusted as a standard of encryption [16] by the US government and other organizations. It is extremely efficient in encryption. It has 128-bit form keys (Fig. 15). It also uses keys of length 192 and 256 bits for heavy duty encryption purposes. It is designed to have only single whole bite operations.

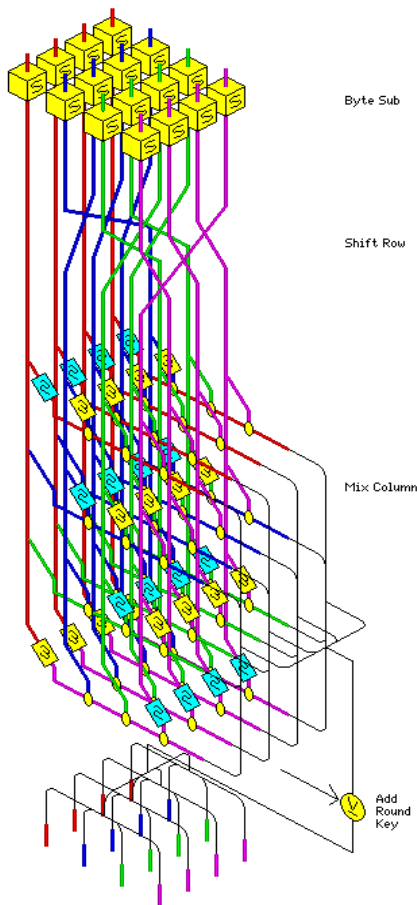


Fig. 15: The AES algorithm, Picture Courtesy: Christopher brown [16]

We have used the AES or Rijndael encryption standard for encrypting our equation, $(x^2 + (9/4)*(y^2) + z^2 - 1)^3 - (x^2)*(z^3) - (9/200)*(y^2)*(z^3)$. After encryption of the equation, we get text as shown in Fig. 16.

Encrypted text:

00000000	b2 9d ba 82 41 cd c2 aa 27 5f 7e c6 6e 39 98 e5	z . 0 . A I A * ' _ - Æ n 9 ä
00000010	d7 8f 7a cd 5c 8f 34 63 51 be b9 70 c1 b5 8e 77	x z İ \ 4 c Q ð ! p Á µ . w
00000020	27 9b 70 04 15 98 32 68 c5 f9 61 6a 19 42 68 50	' . p . . 2 h Å ù a j . B h P
00000030	a6 b5 f6 70 fd 2e e8 6f a3 d3 64 47 5c 5e 88 e0	! µ ö p ý . è o é Ó d G \ ^ ^ à
00000040	c0 9a f1 ba 15 76 82 00 1a 77 ec 82 1f 20 c6 b6	Ä . ñ ° . v . Ð . w i . . Æ ¶
00000050	b9 81 52 f5 40 70 e5 fd 5f 00 a7 81 9e 7f 81 ec	¹ . R ö @ p ä ý . . § i

Fig. 16: Encryption of the equation by AES 256-bit encryption

Till now we successfully encrypted the text using Rijndael 256-bit encryption with a key value of 4. In this study we used the online tool for applying this cryptographic encryption [17]. The size of the binary format of the encrypted downloaded text was 96 bytes, which is still small compared to the rendered model in Maya. Now, we want to decrypt the encryption using the same key. On decrypting we get back the plain text in the form of equation as shown in Fig. 17.

Decrypted text:

00000000	28 78 5e 32 20 2b 20 28 39 2f 34 29 20 2a 28 79	(x ^ 2 + (9 / 4) * (y
00000010	5e 32 29 20 2b 20 7a 5e 32 20 2d 20 31 29 20 5e	^ 2) + z ^ 2 - 1) ^
00000020	33 20 2d 20 28 78 5e 32 29 20 2a 28 7a 5e 33 29	3 - (x ^ 2) * (z ^ 3)
00000030	20 2d 20 28 39 2f 32 30 30 29 20 2a 28 79 5e 32	- (9 / 2 0 0) * (y ^ 2
00000040	29 20 2a 28 7a 5e 33 29 00 00 00 00 00 00 00)* (z ^ 3)
00000050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Fig. 17: Decrypting the equation by AES 256-bit encryption with a key value of 4

From the above investigation we can find that this form of storing information or model can be very useful for classified military objects. Now decrypting we get the text that can easily generate the model by passing it in math-mod as shown in Fig. 13.

C. New brute force Algorithm to encrypt already existing models

We proposed here a new algorithm which can brute force and try all possible combination to match the polynomial.

$$General\ algorithm\ for\ brute\ force = (1 + f(x) + f(y) + f(z)), \sum_{i=0}^n x^i, \sum_{i=0}^n y^i, \sum_{i=0}^n z^i$$

Where,

$$f(x) = \sin(x) + \cos(x) + \tan(x) + e^x + \log(x) + \dots + k\ terms,$$

$$f(y) = \sin(y) + \cos(y) + \tan(y) + e^y + \log(y) + \dots + k\ terms, \text{ and}$$

$$f(z) = \sin(z) + \cos(z) + \tan(z) + e^z + \log(z) + \dots + k\ terms$$

Where f is a function that have all the trigonometric, logarithmic and exponential functions available. We can

increase the no. of components in as much as possible, but the complexity will increase too.

It will generate all possible combination of equations starting from one, and match the model generated, deduct the error and try the next possible iteration, by which it will be more accurate. Though, the complexity will be high, but we can reduce to a great extent after getting minimum error for a certain polynomial equation. A simulation is shown in Fig. 18, as a possible behavior of the equation as a part of the model.

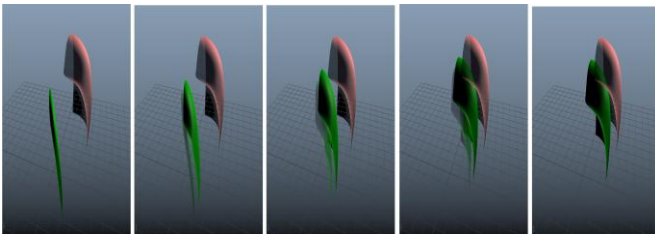


Fig. 18: Possible behavior of brute-force polynomial generating model as shown in green, to fit the part of the model as shown in Red, by tweaking parameters. Picture Courtesy: Bodhiswatta Saha [18].

The above object is easy, simple or small, but when the object is complex, we can break the model into parts and then fit the general brute force equation as shown in Fig. 19.

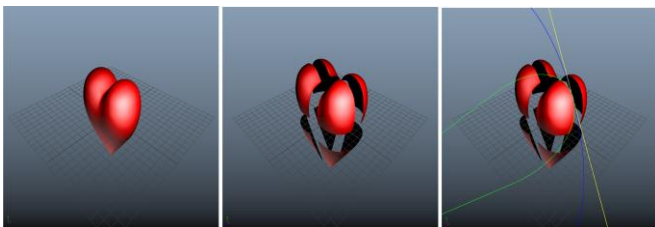


Fig. 19: A possible solution to find the equation for complex models by breaking it into parts and then fitting each equation one by one, and taking the equation with the perfect fit. Picture Courtesy: Bodhiswatta Saha [18]

The model is first broken into parts of equal sizes in order, and then it is tested on equations which give the perfect fit. Then it is arranged and stored in the same fashion as it is broken. The sequence numbers are given. Then it is encrypted and stored, successfully reducing the size. After that it is decrypted with the help of key and it is generated in the same sequence as it was encrypted.

In this way, we can encrypt existing models. We can even encrypt real life pictures, by turning it into models, and then using this brute force polynomial generation algorithm to

match the model. After that we can use any standard encryption technique to successfully encrypt it.

IV. RESULTS AND DISCUSSION

We found that everything in this world can be represented in terms of polynomial equations. We also used the math-mod software to successfully generate a heart and encrypt it using standard encryption techniques, resulting in higher compression as compared to its Maya's counterpart. We also proposed an algorithm to generate all possible combinations of polynomial equations and then match it, and selecting the equation for good fit. In this way we can encrypt existing models. We can even encrypt real life objects, initially making a model of it through standard techniques and fitting the equations.

Let's take an example of the equation for 1 degree polynomial (i.e., $n = 1$) and then testing it on only, $f(x) = \sin(x)$, $f(y) = \sin(y)$, and $\sin(z)$ i.e., $k = 1$. So, the equation becomes:

$$\begin{aligned} &\Rightarrow (1 + \sin(x) + \sin(y) + \sin(z)).(1+x).(1+y).(1+z) \\ &\Rightarrow (1 + \sin(x) + \sin(y) + \sin(z)).(1+x+y+z+y.x+z.x+z.y+x.y.z) \\ &\Rightarrow 1+x+y+z+y.x+z.x+z.y+x.y.z+1.\sin(x)+x.\sin(x)+y.\sin(x)+z.\sin(x)+y.x.\sin(x)+z.x.\sin(x)+z.y.\sin(x)+x.y.z.\sin(x)+1.\sin(y)+x.\sin(y)+y.\sin(y)+z.\sin(y)+y.x.\sin(y)+z.x.\sin(y)+z.y.\sin(y)+x.y.z.\sin(y)+1.\sin(z)+x.\sin(z)+y.\sin(z)+z.\sin(z)+y.x.\sin(z)+z.x.\sin(z)+z.y.\sin(z)+x.y.z.\sin(z) \end{aligned}$$

Now, we will take constants, $C_1, C_2, C_3, \dots, C_{((k*3+1)*(n+1))^3}$, the final equation becomes:

$$\begin{aligned} &\Rightarrow C_1.1 + C_2.x + C_3.y + C_4.z + C_5.y.x + C_6.z.x + C_7.z.y \\ &+ C_8.x.y.z + C_9.\sin(x) + C_{10}.x.\sin(x) + C_{11}.y.\sin(x) + C_{12}.z.\sin(x) + C_{13}.y.x.\sin(x) + C_{14}.z.x.\sin(x) + C_{15}.z.y.\sin(x) + C_{16}.x.y.z.\sin(x) + C_{17}.\sin(y) + C_{18}.x.\sin(y) + C_{19}.y.\sin(y) + C_{20}.z.\sin(y) + C_{21}.y.x.\sin(y) + C_{22}.z.x.\sin(y) + C_{23}.z.y.\sin(y) + C_{24}.x.y.z.\sin(y) + C_{25}.\sin(z) + C_{26}.x.\sin(z) + C_{27}.y.\sin(z) + C_{28}.z.\sin(z) + C_{29}.y.x.\sin(z) + C_{30}.z.x.\sin(z) + C_{31}.z.y.\sin(z) + C_{32}.x.y.z.\sin(z) \end{aligned}$$

After multiplying different variables with each component, we will set a boundary value for tweaking those variables which will be constant for a particular set of equation. For this equation, let it be 10. For each component it will have 10 values, and there are total 32 components, which means that for the total number of equations it will calculate is 10^{32} . The computing complexity will be tremendous, and it could be achieved with the help of a super computer. The computing

complexity for this algorithm is shown in Fig. 20, which means it is almost practically impossible to test this in a normal computer, we can also skip some of the values and then it will be a little bit smaller, for example we can take even values only. After taking the even values, the value of c will be 5 and the number of equations formed will be 5^{32} .

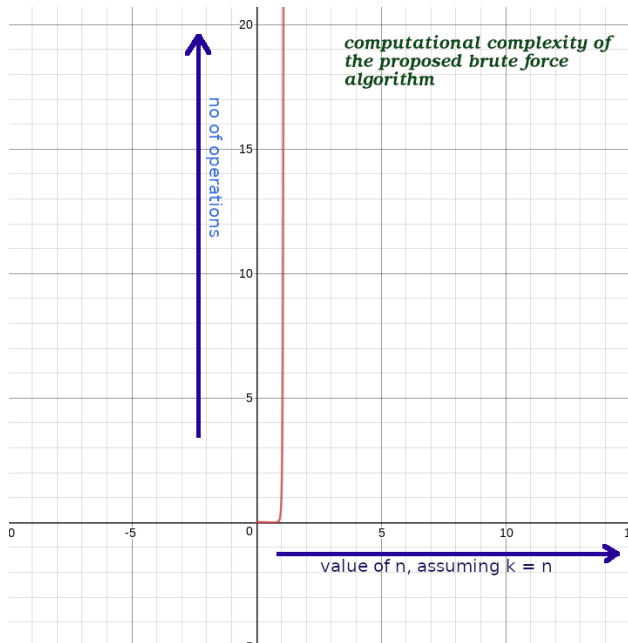


Fig. 20: The complexity of the brute force algorithm is $O(n^{(1+3.n).(n+1)^4})$, assuming $k=n$.

If this is too much, we can ignore more of these equation by ignoring similar values, but this will take more time and resource in a normal computer.

V. CONCLUSION AND FUTURE SCOPE

Above findings show that any model can be represented in terms of equations, i.e., texts form. Which means that they can be encrypted using any standard algorithms, which forms a great help to secure classified military information or models? The above algorithm is using brute force strategy, meaning it have a tremendous amount of time complexity for a given value of $n = 2$, $k = 3$, and a value of $C = 10$, here we get the number of operations = $C^{(1+3.k).(n+1)^3}$, which is operations. This can be further improved by leaving the redundant terms. This algorithm is just based for supercomputers and if it is tested, it will give perfect results, i.e., it will represent all objects in a single equation, further compressing it to a several times depending on how much complex the shape is and hence forming a new way to compress, generate, regenerate and encode objects.

ACKNOWLEDGMENT

I appreciate help received from my father Dr. Jadab Kumar Pal, Deputy Chief Executive, Indian Statistical Institute,

Kolkata, for constantly motivating and support to develop this paper. I am thankful to Mr. Bodhiswata Saha, for constantly inspiring and helping me to design and construct the simulations, which couldn't have been possible without him. I also appreciate the help received from Christopher Brown, University of Rochester, for allowing me to use their pictures in my research paper for demonstration. The authors also acknowledge the help received from Prof. Shalabh Agarwal, HOD of Computer Science department, St. Xavier's College, for his constant support and motivation.

REFERENCES

- [1]. <https://www.desmos.com/calculator/> accessed on 14.10.2018.
- [2]. <https://tpenguinltg.wordpress.com/2014/02/15/representing-the-heart-shape-precisely/> accessed on 09.10.2018.
- [3]. M. M. Blane, Z. Lei, and D. B. Cooper, "The 3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data", <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.8289&rep=rep1&type=pdf> accessed on 10.10.2018, June 1996.
- [4]. C. Oden, A. Ercil, V. T. Yildiz, H. Kirmizita, and B. Buke, "Hand Recognition Using Implicit Polynomials and Geometric Features" <https://pdfs.semanticscholar.org/1432/01f8d05e22882fe01f3f2ac2e17b14e78a93.pdf> accessed on 10.10.2018.
- [5]. J. C. Quiroz, and S. M. Dasalu, "Design and Implementation of a Procedural Content Generation Web Application for Vertex Shaders", arXiv: 1608.05231[cs.GR], <https://arxiv.org/abs/1608.05231> accessed on 10.10.2018, August 2016.
- [6]. A. A. Ahmadi, G. Hall, A. Makaida, and V. Sindhwani, "Geometry of 3D Environments and Sum of Squares Polynomials", arXiv: 1611.07369 [math.OC], <https://arxiv.org/abs/1611.07369> accessed on 10.10.2018, November 2016.
- [7]. S. Hu, and J. Ji, "Using Chebyshev polynomials interpolation to improve the computation efficiency of gravity near an irregular-shaped asteroid", arXiv: 1708.06493 [astro-ph.EP], <https://arxiv.org/abs/1708.06493> accessed on 10.10.2018, August 2017.
- [8]. J. Komorowski, and P. Rokita, "Face Recognition Based on Sequence of Images", arXiv: 1809.11069 [cs.CV], <https://arxiv.org/abs/1809.11069> accessed on 10.10.2018, September 2018.
- [9]. P. Mandikal, N. K. L., and R. V. Babu, "3D-PSRNet: Part Segment 3D Point Cloud Reconstruction from a Single Image", arXiv: 1810.00461 [cs.CV], <https://arxiv.org/abs/1810.00461v1> accessed on 10.10.2018, September 2018.
- [10]. Y. L. Qiao, L. Gao, Y. K. Lai, and S. Xia, "Learning Bidirectional LSTM Networks for Synthesizing 3D Mesh Animation Sequences", arXiv: 1810.02042 [cs.GR], <https://arxiv.org/abs/1810.02042> accessed on 10.10.2018, October 2018.
- [11]. J. C. Carr, R. K. Beaston, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and Representation of 3D Objects with Radial Bias Functions", <http://www.cs.jhu.edu/~misha/Fall05/Papers/cart01.pdf> accessed on 10.10.2018.
- [12]. H. Gross, U. Thoennessen, and W. V. Hansen, "3D-Modelling of Urban Structures", Research Institute for Optronics and Pattern Recognition, Stilla U, Rottensteiner F, Hinz S (Eds) CMRT05. IAPRS, Vol. XXXVI, Part 3/W24,

- http://www.isprs.org/proceedings/XXXVI/3-W24/papers/CMRT05_Gross_et_al.pdf accessed on 10.10.2018, August 2018.
- [13]. N. Kholgade, T. Simon, A. Efros, and Y. Sheikh, "3D Object Manipulation in a Single Photograph using Stock 3D Models", <https://dl.acm.org/citation.cfm?id=2601209> accessed on 10.10.2018.
- [14]. <https://github.com/parisolab/mathmod> accessed on 09.10.2018.
- [15]. StorageCraft, <https://blog.storagecraft.com/5-common-encryption-algorithms/> 5 Common Encryption Algorithms and the Unbreakables of the Future, Contel Bradford, accessed on 9.10.18.
- [16]. Diagram of the AES Algorithm (9.16.04), <https://www.eng.tau.ac.il/~yash/crypto-netsec/rijndael.htm>, colorful crypto rijndael, Christopher Brown, The Advanced Encryption Standard (Rijndael), accessed on 15.10.2018.
- [17]. <http://rijndael.online-domain-tools.com/> -- tool for encryption, accessed on 14.10.2018.
- [18]. <https://bodhis96.wixsite.com/website-3d/> accessed on 09.10.2018.

Authors Profile

Jimut Bahan Pal is a student of Dept. of Computer Science, St. Xavier's College (Autonomous), Kolkata-700016. He is fluent in Python programming language, and has invented various scraping programs. He is also a member of various online learning communities like Coursera, Stanford Online, edX etc. Being a Machine Learning enthusiast, he has done various minor projects in Machine Learning area. He has also built various RPG games with Unity-3D, and Unity-2D game engines in association with MOOCs. He is an aspiring and curious researcher.



Dr. Asoke Nath is working as Associate Professor in the Department of Computer Science, St. Xavier's College (Autonomous), Kolkata. He is engaged in research work in the field of Cryptography and Network Security, Steganography, Green Computing, Big data analytics, Li-Fi Technology, Mathematical modelling of Social Area Networks, MOOCs etc. He has published more than **238** research articles in different Journals and conference proceedings.

