# Generate Cryptographic key using generated 3D- Digital Image

**Dr.Hala B. Abdul Wahab*, Dr.Suhad M. Kadhum* & Ekhlas K. Gbashi***

Received on: 20/ 1 /2008
Accepted on: 4/9 /2008

**Abstract**

Every few years, computer security has to re-invent itself. New technologies and new application bring new threats, and force us to invent new protection mechanisms. Cryptography became important when businesses started to build networked computer systems .The main goal of this work is combining the curve security methods with cryptography algorithms in order to increase the capability of cryptography. The weakness of the cryptographic key generated from normal color image is clear due to the nearest pixel values of image. This led to propose in this paper a new method in order to generate a cryptographic key depending on generated (2D & 3D) mathematical models (digital image) and clipping the key according to an algorithm and the data of curve generation.

**Keywords:** Applied Cryptography, Graphics, Image Generation, Random Number Generator,

**توليد مفتاح تشفير باستخدام توليد صور رقمية ثلاثية الابعاد**

**الخلاصة**

في السنوات الاخيرة الماضية أمنية الحاسوب أخذ بأعادة تطوير نفسها حيث أن ظهور التكنولوجيا الحديثة و التطبيقات الجديدة جلبت تهديدات جديدة خاصة مع نمو المتزايد في مجال الاتصالات ونقل البيانات . هذا الذي جعل الحاجة الى التشفير وأمنية البيانات من المتطلبات الاساسية في كافة مجالات الاتصالات ونقل وحماية البيانات . أن الهدف الاساسي في هذا البحث هو دمج بين أمنية المنحني و خوارزميات التشفير من أجل زيادة أمكانيات التشفير. حيث ان الضعف في مفتاح التشفير المولد بواسطة الصور الملونه الاعتيادية واضح بسبب التقارب القيم العنصريه للصورة هذا الذي قادنا في هذا البحث الى تقديم طريقة جديدة لتوليد مفتاح التشفير بالاعتماد على توليد نماذج رياضية ثنائية وثلاثية الابعاد(صور رقمية) ومن ثم أستقطاع مفتاح التشفيربالأعتماد على خوارزمية و بيانات توليد المنحني .

## 1. Introduction

Counterfeiting is a growing threat in recent years, especially with the ever-increasing growth of data communication, the need for security and privacy has become a necessity. Cryptography and data security are an essential requirement for communication privacy. Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot read by anyone except the intended recipient [1].

Computer graphics is a topic of rapidly growing importance in the

computer field. It always has been one of the most visually spectacular branches of computer technology, producing images whose appearance and motion make them quite unlike any other from of computer output [2]. The shape of the curve is basically based upon a set of control points that fundamentally describe its properties

**Computer Science Department, University of Technology / Baghdad**

https://doi.org/10.30684/etj.27.2.12

2412-0758/University of Technology-Iraq, Baghdad, Iraq

and its curvature. The algorithms that used to generate the curves are primarily based on these control points. Thus if the intruder knows the set of control points it may lead to discover the shape of the curves with a trial and error on the method or algorithms that are originally used to produced the curve[3].

**2. Generating 3D-image**

This section, introduces a new idea in generating 3D-image by inserting a third coordinate (z) to all of the mathematical equations that used to generate the 2D- image. Generating a 3D- image is very useful to increase the key space, and make the process of estimating the key by the counterfeiter infeasible [4,8].

Because searching in 3D increase the choices of estimating the control points for the counterfeit. To make the idea easy to be imagined when working with 3D, we suggest that the image look like an empty cube space and we want to fill it completely with random color pixels, and then clip slide from the cube by different ways. The clipped slide from the three-dimensional image ( 3D) can be used directly as a secret key, or clipping a curve from this slide according to a secret set of control points by using parametric Bezier curve ( control curve) to represent the secret key between the sender and receiver.

**2.1 The Proposed Algorithms to Generate 3D Image**

Generating a 3D-image process requires to modifying all the equations of the work with 3D-coordinates(x, y, z) instead of 2D- coordinates(x,y). For this reason, we develop the algorithms that generate the control curves and the algorithms that generate the oscillation curve. This leads to proposing algorithms to generate (interpolate) 3D parametric Lagrange curve, and convert the algorithm for moving rolling circle around parametric curve to work with the three dimensional coordinates. In the following all algorithms that are proposed work with 3D coordinates [5,8].

***Algorithm1: (*Generate 3D parametric Lagrange curve).**

**Input**: Given set of data($x_i$,$y_i$,$z_i$), i=0…N, and given values of parameter $t_i$=0,…,N, where $\Delta t=(t_{i+1}-t_i)=10$.

**Output**: Generate 3D parametric Lagrange curve.

**Process:**

Step1: Set res=0, k=1, step=1, N=10, h=1
Step2: Temp (h) =t (k)
Step3: While (k <> N) do
While(Temp(h)>= t(k) and (Temp(h)<=t(N)) do
Step4: res=0
For i=1 to N
P=1
For j= 0 to N
If i <> j then p =p *(Temp (h) - t (j))/ (t (i)-t (j))
Next j
Step5: res=res+ (p*x (i))
Next i
Step6: New_x (h) =res,
res=0, h=h+1,
Temp (h) =Temp (h-1) +step
Step 7: loop,
k=k+1, loop.
Goto from step 1 to 4.
Step8: res=res+ (P*y (i))
Step 9: next i
Step 10: New-y (h)=res,
res=0, h=h+1,
Temp (h) =Temp (h-1) +step
Loop, k=k+1, loop
Step 11: Goto from 1 to 4
res=res+ (P*z (i))
Step 12: next i
Step 13: New-z (h) =res,
res=0, h=h+1,

Temp (h) =Temp (h-1) +step
Loop, k=k+1, loop
Step 14: for i= 1 to h
Plot (New-x, New-y, New-z)
Next i
Step 15: End.

*Such that:*
*N: number of control points*
*K: number of time*

**2.2 Proposed Algorithm to Moving 3D Rolling Circle around 3D-Parametric Lagrange Curve.**

To create a parametric form from a 3D curve, we have to invent three functions x(.), y(.), z(.), and say that the curve is "at" P(t)=(x(t),y(t),z(t)) at time t [6]. The circular helix given parametrically by: x (t) = cos (t), y (t) =sin (t) and z (t)=bt. For some constant b. The curve illustrated in figure (1).

Many variations on the circular helix are possible. Such as the conical helix, with equation P (t) = (t cos (t), t sin (t), bt) . Any 2D-curve (x(t), y(t)) can, of course, be converted to a helix by appending z(t)= bt, or some other form for z(t).

***Algorithm2:*** **(move 3D-rolling circle around 3D-parametric Lagrange curve).**

**Input:** Take the triple data (New-$x_i$, New-$y_i$, New-$z_i$), i= 0,…,h-1, that was computed in algorithm (1) to represent the circle center coordinates (x-c, y-c, z-c).,and the radius circle value.

**Output**: Moving 3D-rolling circle around 3D-parametric Lagrange curve that generated in algorithm (1).

**Process:**
Step1: Set radius=15, b=10
Step2: For i= 0 to h-1
Set x-c= New-x, y-c=New-y, z-c=New-z
For index=0 to 360
X=radius*cos (index) +x-c
Y=radius*sin (index) +y-c
Z=index*b+z-c
Step 3: Plot (X, Y, Z)
Step4: Next index
Step5: Next i
Step6: End.

**2.3 Generating 3D Image Stages**

In section (2.1) we mentioned clearly the basic idea to generate the 3D-image by using 3D mathematical models and explained the algorithms that are needed to perform this generation. Additionally, we suggest that 3D image looks like an empty cube and the aim is to fill completely the cube by random color pixels, this process consists of a set of stages. These stages initialize a 3D-mesh of control points, marking the control points to produce a sequence of control points by using pseudo-random generator[7]. Implementation of the algorithm (2) is to move the 3D- Lagrange curve with 3D rolling circle through the 3D mesh of control points, delete all the pixel that are out of cube and the final stage is obtaining cube (image) which is completely filled by random color pixels. The following illustrates the stages for generating 3D-image [8]:-

***Stage One***

3D shape is to be divided into slides (planes) as each slide takes a 2D- matrix, its axis's dimensions are x and y, and the number of their slides will be equal to the depth z-axis of shape.

Figure (2) shows an example of initialized 3D mesh.

Initialize a 3D mesh is achieved by the same algorithm that was used to generate the 2D mesh of control points, but inserting the third coordinate (z)to the control point

coordinates, and determining the increment values to x-,y-,z-coordinates. In addition, when changing the increment values x-, y-, z-coordinates we obtain different types of 3D mesh of control points that make the process for estimating the mesh by counterfeiter to be more difficult. Figure (3) shows an example of the cubic mesh (3D mesh) with equal increment values(x, y and z) between the control points.

***Stage Two***

Marking the 3D- mesh is achived by counting the 3D-mesh of control points ,and one can suggest that the cube mesh to look like a set of 2D-mesh slides marked with page number(z-coordinate) or an array of three dimensional coordinates. Figure (4-a ) is an example that shows the marking of the 3D mesh size(4×4×3), and in figure(4-b) another example of marking the 3D mesh.

***Stage Three***

Pesudo-random generator is used to produce random addresses of control points, from 3D-mesh which are used as a set of control points to implement algorithm(2) for moving the 3D-curve ( parametric lagranege curve) with 3D-rolling circle through the 3D-mesh. Figure (5) shows moving curve through the 3D as an example to approximate the idea of moving the curve through the cube.

***Stage Four***

This stage eliminates all the pixels that draw out of the cube due to the isolation of moving Lagrange curve. The deleting process is to remove all the pixel that are placed out of the 3D, is implicitly implemented in the algorithm when it computes any pixel the algorithm tests the coordinates of the pixel, if the generated coordinates of pixels out of boundaries of the 3D coordinates.

In the following, the complete algorithm shows how to generate 3D-image using 3D mathematical models.

**Algorithm 3: Generate 3D-Image using 3D-mathematical models**

**Input:** Given first control point, increment value, size of mesh control points (N×N×N) and size of the 3D image that need to be generated.

**Output:** Generate 3D image and save the color values in 3D array size (N×N×N).

**Process:**

Step1: Initialize a 3D-mesh of control points according to the start control point; increment the value and the size of the 3D-mesh.

Step2: Mark the control points of 3D-mesh.

Step3: Enter the number of marks from 3D-mesh to simple pseudo-random generator.

Step4: Take a sequence of numbers from pseudo-random generator to represent the addresses of the control points in the mesh.

Step5: Execute Algorithms (1) and (2) to draw the parametric Lagrange curve with rolling circle and save the results for color pixels value for points in 3D array size (N×N×N).

Step6: Repeat step4 with new sequence of numbers of control points and step 5 until fill the cube by recursive pixels completely.

Step7: Delete all the pixels that are placed out of the cube due to the isolation-moving curve.

Step8: Obtained 3-dim generated image.

Step9: End.

To explain the work of the algorithm, figure (6) shows the implementation of algorithm (3). This example used the 3D size (100×100×10), (i.e. the 3D consisted of 100000 of random color pixels) that was used to generate 3D-mesh of control points, and the 3D-mesh size of (50×50×50), (i.e. the mesh consist of 125000 of control points), and equal increment value between the control points.

***3. Clipped cryptography Key From 3D-generated Image***.

Work with 3D-image requires to determined the level of (z)-coordinate from the cube, which is represented in this case the slide number form the cube to clip 2-dim slide from 3D-image.

Step1: Input the number of level of z-coordinate (z-coordinate), and the two control points ($x_1$, $y_1$) and ($x_2$, $y_2$) from z-slide, and clipped slide of color pixels from the three dimensions array (3D).

Step2: Set z= z-coordinate
For i= $x_1$ to $x_2$
For j= $y_1$ to $y_2$
slide (i, j) =3D(i , j, z)
Next j, Next i.

Step3: end.

***4. Tests of Randomness to the Clipped Keys.***

Different sizes of keys are used in these tests and the results of the tests proved that the keys have the randomness property and can be used as a symmetric key in cryptography field. Table (1) show the results of randomness test to the keys
that are clipped from 3D images.

**5 .Conclusions**

This work presents a method that combines curve security and the cryptographic algorithms by clipping cryptographic key from 2D or 3D-mathematical models (digital images) that are generated using curve fitting generation methods.

The main conclusions are:

The generated 3D-mathematical models increase the key space and the computation time to estimate the key from the attacker.

1) The clipped curve from generated 2D or 3D mathematical models gives good results to be use as symmetric cryptographic key according the randomness tests.
2) The clipped curve generated from 2D or 3D generated mathematical models (digital images) gives reasonable results in order to be used as a key in image cryptography according to the measure tests that used in the field.

**6. References**

[1] Alfred J.M., Paul V. C. and Scott A. V., (2001), "***HandBook of Applied Cryptography***", Fifth Addition.

[2] Alan Watt, (2000)."***3D Computer Graphics***", Published by Addison Wesley Company, Inc.

[3] Jean Gallier, "***Curves and Surfaces in Geometric Modeling***"; Morgan Kaufman Publishers, 2000.

[4] Hill F. S. , (2001)."***Computer Graphics Using OPENGL***", second edition, Prentice Hall.

[5] Dallwitz M. J. , (2004). "***An Introduction to Computer Images***", http:// delta-intkey.com.

[6] Goldman Ron, (2002). "***Lagrange Interpolation and Neville's Algorithm***", Department of computer Science ,Rice University.

[7] Jean Gallier, (2000). "***Curves and Surfaces in Geometric Modeling",*** Morgan Kaufman Publishers.

[8] Hala Bahjat, (2006)." Proposed 3D Generated Digital Image in

Ceryptography", PHD, Universitiy of Technology, Department Science.

**Appendix**

**Table (1) Randomness Test to the First Four Keys.**

| *Tests* | | *Key1= 130 bit* | *Key2= 260 bit* | *Key3= 400 bit* | *Key4= 520 bit* | *Pass value* |
|---|---|---|---|---|---|---|
| **Frequency test** | | 0.277 | 1.246 | 2.250 | 1.025 | Must be ≤ 3.84 |
| **Run test** | $T_0$ | 3.638 | 5.184 | 14.102 | 0.338 | Must be ≤ 22.362 |
| | $T_1$ | 4.638 | 9.954 | 2.698 | 15.038 | |
| **Poker test** | | 4.585 | 5.600 | 9.840 | 6.862 | Must be ≤ 11.1 |
| **Serial test** | | 2.969 | 2.923 | 3.560 | 1.723 | Must be ≤5.99 |
| **Auto Correlation test for ten bits** | Shift 1 | 1.310 | 0.988 | 0.203 | 1.10 | Must be ≤3.48 |
| | Shift 2 | 0.281 | 0.003 | 0.251 | 0.124 | |
| | Shift 3 | 0.386 | 0.035 | 3.086 | 0.095 | |
| | Shift 4 | 2.571 | 0.141 | 0.495 | 1.116 | |
| | Shift 5 | 0.392 | 0.192 | 0.124 | 0.049 | |
| | Shift 6 | 0.516 | 0.252 | 0.091 | 0.070 | |
| | Shift 7 | 0.008 | 0.478 | 0.206 | 1.639 | |
| | Shift 8 | 0.295 | 3.111 | 0.092 | 2.133 | |
| | Shift 9 | 1.860 | 0.896 | 0.003 | 0.440 | |
| | Shift 10 | 2.700 | 0.256 | 0.656 | 0.635 | |

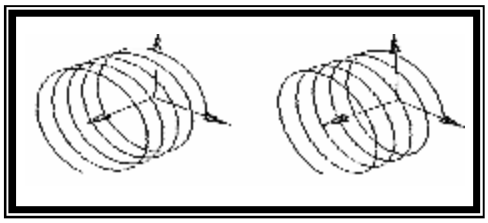

**Figure (1) Circular helix**

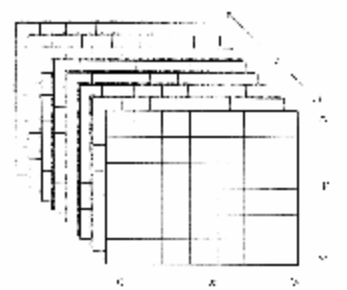

**Figure (2) Initialized 3D mesh.**

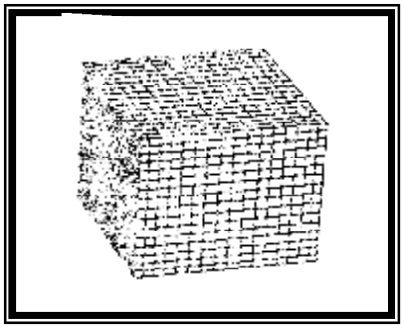

**Figure (3) Example of cube mesh.**

(a) (b)

**Figure (4) Examples of marking meshes cube.**

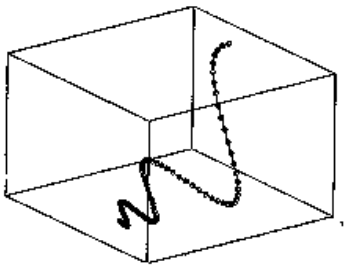

**Figure (5) Example of moving curve through cube.**

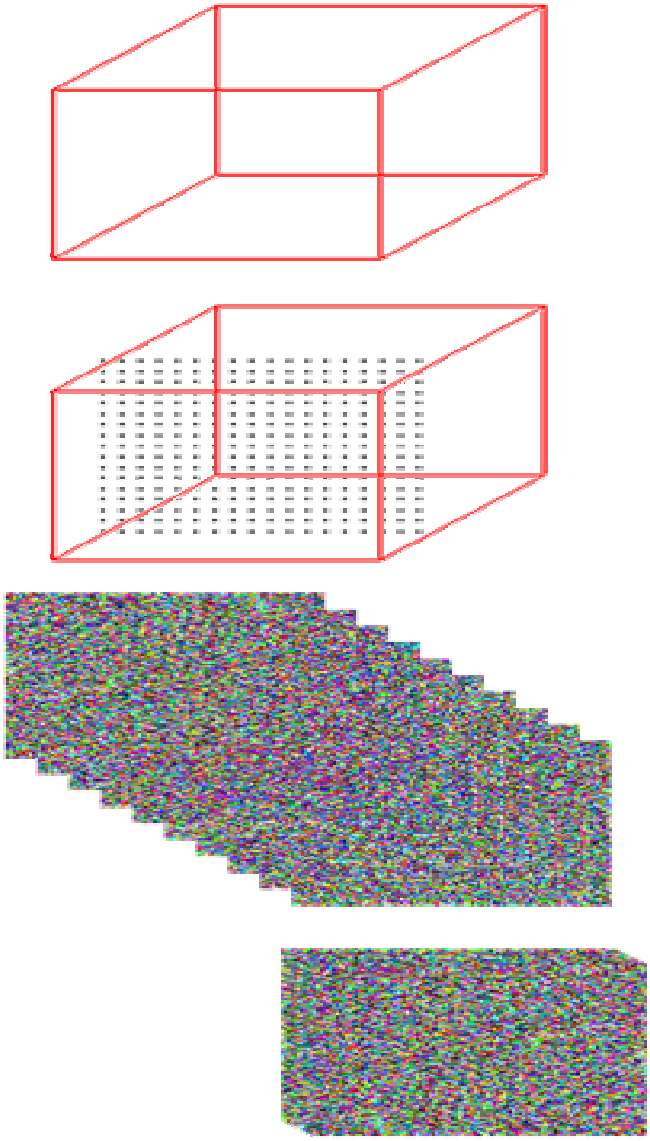

**Figure (6) Example to generate 3D-image.**