9

PGP Protocol and Its Applications

Hilal M. Yousif Al-Bayatti[1],
Abdul Monem S. Rahma[2] and Hala Bhjat Abdul Wahab[2]
[1]*Applied Science University, Kingdom of Bahrain*
[2]*Computer Science Depart , University of Technology, Baghdad, Iraq*

1. Introduction

Every few years, computer security has to re-invent itself. New technologies and new application bring new threats, 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. One of the newest hot spots in security research is curve security. The forms produced by graphic systems are much harder to counterfeit, especially when the counterfeiter has no information about algorithm and the data that are used in design the shape.

The main goal of this chapter 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 chapter 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 algorithm and the data of curve generation.

2. Key management cryptographic systems

2.1 Conventional encryption

Conventional encryption has benefits; most notably speed. It is especially useful for encrypting data which will not be transmitted. However, conventional encryption alone, as a means for transmitting secure data, can be expensive simply due to the difficulty of secure key distribution (Droste, 1996). For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. If they are in different physical locations, they must trust a courier or some other secure communications medium to prevent the disclosure of the secret key during transmission (Sauer & Chandy, 1981).

2.2 Public-key cryptography

The concept of public-key cryptography was introduced in 1976 by Whitfield Diffie and Martin Hellman and, independently, by Ralph Markle, to solve the key management problem in secret-key cryptography. Each person receives a pair of different but related

keys; the public-key and the private key (Schneir, 1996). The public key is published, while the private key is kept secret. Anyone can send a confidential message using public information, but the message can only decrypted by someone who has the private key. It is not possible to determine the secret key from the public key.

Public-key cryptography is based on the idea of a trapdoor function:

- $f : X \to Y$
- f is one-to-one,
- f is easy to compute,
- f^{-1} is difficult to compute, and
- f^{-1} becomes easy to compute if a trapdoor is known.

Public-key cryptography ensures a secret key exists between communicating parties, without the need to distribute secret keys. The most famous public-key cryptosystem is that devised by Rivest et al. (RSA). The RSA scheme relies on the inability to factor n where n=pq; p and q are large strong prime numbers, and when p and q are of roughly equal length, the modules become harder to factor. Factoring is assumed to be a difficult problem upon which several public-key cryptosystems are based. However, no known polynomial time algorithm can decipher the keys using the public information.

In RSA, algorithm encryption and decryption take the following forms:

$$C = M^e \bmod n$$

$$M = C^d \bmod n$$

Where *M*: plain text, *C*: cipher text, *e*: encryption key, and *d*: decryption key. Finding the factors of an RSA modulus depends on finding the encryption key. If an adversary discovers the factors of *n*, he or she can easily compute the decryption key using the following steps (Guan, 1987):

$$n = p \times q$$

$$\theta(n) = (p-1)(q-1), \text{ then}$$

$$d = e^{-1}(\theta(n)) \bmod n$$

The RSA algorithm has the following characteristics (Menezes et al.,1996):

- It is computationally infeasible to determine the decryption key given only knowledge of the algorithm and the encryption key.
- Either of the two keys can be used for encryption, with the other used for decryption.

2.3 Use of multi keys (huge) in a public-key cryptography system

One of the major roles of public-key encryption has been to address the problem of key distribution. There are two distinct aspects involving the use of public-key cryptography in this regard:

1. The distribution of public keys
2. The use of public-key encryption to distribute secret keys
3. A hybrid schema

2.3.1 Distribution of public keys

Several techniques have been proposed for the distribution of public keys. Almost all these proposals can be grouped within the following general schemes:

1. Public announcement
2. Publicly available directory
3. Public-key authority
4. Public-key certificates

For more details see (Stalings,2005).

2.3.2 Distribution of secret keys using public-key cryptography

Once public keys have been distributed or have become accessible, secure communication that thwarts eavesdropping, tampering, or both is possible. However, few users will wish to make exclusive use of public-key encryption for communication because of the relatively slow data rates that can be achieved. Accordingly, public-key encryption allows the distribution of secret keys to be used for conventional encryption.

1. Simple Secret Key Distribution
2. Secret Key Distribution with Confidentiality and Authentication.

2.3.3 A hybrid schema

A further application of public-key encryption used to distribute secret keys is a hybrid approach based on IBM mainframes .This scheme retains the use of a key distribution centre (KDC) that shares a secret master key with each user and distributes secret session keys encrypted with the master key. A public key scheme is used to distribute the master keys. The following rationale is provided for using this three-level approach:

Performance: There are many applications, especially transaction-oriented applications, in which the session keys change frequently. Distribution of session keys by public-key encryption could degrade overall system performance because of the relatively high computational load of public-key encryption and decryption. With a three-level hierarchy, public-key encryption is used only occasionally to update the master key between a user and the KDC.

Backward compatibility: The hybrid scheme is easily overlaid on an existing KDC scheme, with minimal disruption or software changes. The addition of a public-key layer provides a secure, efficient means of distributing master keys. This is of advantage in a configuration where a single KDC serves a widely distributed set of users.

3. Cryptographic protocol (Pretty Good Privacy)

Pretty Good Privacy (PGP) is a public key system for encrypting electronic mail using the RSA public key cipher (Schaefer, 1999). PGP is a hybrid cryptosystem and combines some of the

best features of both conventional and public-key cryptography. When a user encrypts plaintext, PGP first compresses that plaintext. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security. The majority of cryptanalysis techniques exploit patterns found in the plaintext to crack the cipher. Compression reduces these patterns in the plaintext, thereby greatly enhancing resistance to cryptanalysis (files which are too short to compress or which do not compress well are not compressed). PGP then creates a session key, which is a one-time-only secret key. This key is a random number generated from the random movements of the mouse and the keystrokes. The session key works with a very secure, fast conventional encryption algorithm to encrypt the plaintext; the result is ciphertext. Once the data is encrypted, the session key is then encrypted to the recipient's public key. This public key-encrypted session key is transmitted along with the ciphertext to the recipient. Figure (1) shows the send process.

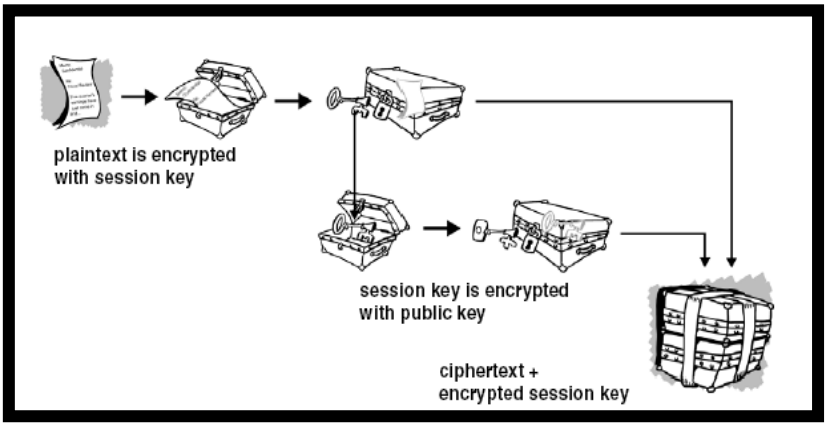


Fig. 1. Send Process.

Decryption works in the reverse way. The recipient's copy of PGP uses his or her private key to recover the session key, which PGP then uses to decrypt the conventionally encrypted ciphertext. Figure (2) shows the receiving process.

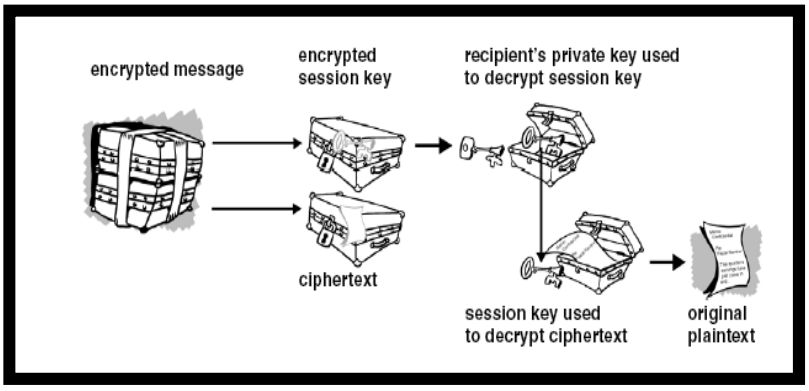


Fig. 2. Received Process

The combination of the two encryption methods combines the convenience of public-key encryption with the speed of conventional encryption. Conventional encryption is approximately 10,000 times faster than public-key encryption. Public-key encryption in turn provides a solution to key distribution and data transmission issues. Used together, performance and key distribution are improved without any compromise in security.

4. Generate randomness digital images

4.1 Introduction

Computer graphics is a topic of rapidly growing importance in the field of information technology. It has always been one of the most visually spectacular branches of computer technology, producing images whose appearance and motion make them quite unlike any other form of computer output . Computer graphics are also an extremely effective medium for communication between man and computer; the human eye can absorb the information content of a diagram or perspective view much faster than it can scan a table of numbers (Demel & Miller, 1984).

It is important to achieve consensus about what *computer graphics* and *visualisation* are. The word *graphics* is associated with charts, graphs, images, pictures and patterns, often in the context of art, design or animation; *visualisation* is the process of converting data into images (Gomes & Velho, 1997) .In order to understand computer graphics, we must study the methods used to create and structure data in the computer, as well as the methods for turning these into images. These two steps correspond to the two main areas of research in computer graphics: modelling and visualisation (Egerton & Hall,1998). Computer graphics are pictures generated by a computer (Hill, 2000)

4.2 Curve fitting generation

There are two main classes of curve generation algorithms:

Firstly, algorithms that interpolate the control points: the algorithm returns points along a curve which pass exactly through the control points at specific instants and form a smooth curve for points in between.

Secondly, algorithms that approximate the control points: this algorithm returns points from a curve that are attracted toward each control point in turn, but do not actually pass through all of them (Hill, 2000).

4.3 Interpolation techniques

Interpolation techniques are of great importance in numerical analysis since they are widely used in a variety of science and engineering domains where numerical methods are the only way to predict the value of tabulated functions for new input data. There are three reasons for using interpolation: firstly, interpolation methods are the basis of many other procedures, such as numerical differentiation, integration and solution methods for ordinary and partial-differential equations. Secondly, these methods demonstrate important theories about polynomials and the accuracy of numerical methods. Thirdly, interpolating with polynomials serves as an excellent introduction to techniques for drawing smooth curves .

Several methods of interpolation and approximation exist, some of which are:-

4.3.1 Lagrangian polynomials

The Lagrange polynomial is the simplest way to exhibit the existence of a polynomial for interpolation with unevenly spaced data. The Lagrange interpolating polynomial $L_{N,K}$ has degree N and is one at $x=x_k$ and $j \neq k$ zero at $x=x_j$ where

$$L_{N,K}(x) = \frac{(x-x_0)(x-x_1)\ldots(x-x_{K-1})(x-x_{K+1})\ldots(x-x_N)}{(x_K-x_0)(x_K-x_1)\ldots(x_K-x_{K+1})\ldots(x_K-x_N)} = \frac{\prod_{\substack{j=0 \\ j\neq K}}^{N}(x-x_j)}{\prod_{\substack{j=0 \\ j\neq K}}^{N}(x_K-x_j)} \quad (1)$$

Note that $\prod_{K=1}^{N} K = 1.2.3\ldots N.$

=The interpolating polynomial may be written:

$$P_N(x) = \sum_{K=0}^{N} y_k L_{N,K}(x) = y_0 L_{N,0}(x) + y_1 L_{N,1}(x) + \ldots + y_N L_{N,N}(x) \quad (2)$$

It is just a linear combination of the Lagrange interpolation polynomials

$L_{N,K}(x)$ with the y_K as the coefficients (Goldman, 2002).

4.3.2 B-spline polynomials

Although Bezier curves and surfaces are well suited to many shape-modelling problems, complex geometric constructions are required to guarantee continuity when piecing curves together (Harrington, 1987). The use of spline functions avoids this by using mathematical constraints to allow only those curves that possess the required continuity at joints. The B-spline function generates a curve section which has continuous slopes so that they fit together smoothly: see figure (3).

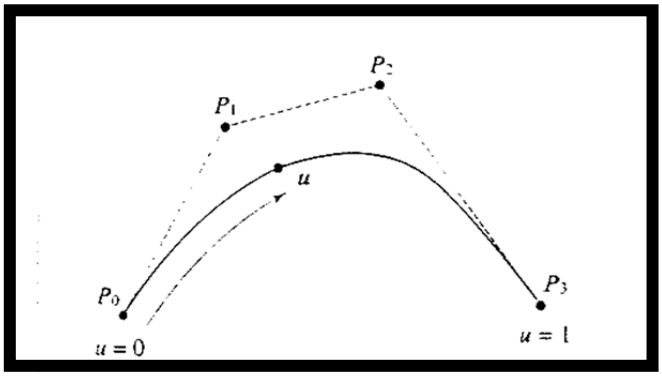


Fig. 3. B-Spline curve segment.

A B-spline of order k=3 consisting of n-k+2 segments is defined by a linear combination of basic functions C_i using n+1

Control points v_i

$$C_i(u) = b_{-1}(u)v_{i-1} + b_0(u)v_i + b_1(u)v_{i+1} + b_2(u)v_{i+2} \quad (3)$$

Where the base functions are defined by (Pham, 1988):

$$b_{-1}(u) = 1/6(-u^3 + 3u^2 - 3u + 2)$$
$$b_0(u) = 1/6(3u^3 - 6u^2 + 4)$$
$$b_1(u) = 1/6(-3u^3 + 3u^2 + 3u + 1)$$
$$b_2(u) = 1/6u^3$$

A B-spline curve exhibits local control- a control point is connected to four segments (in the case of a cubic) and moving a control point can influence only these segments. In figure (4) that shows the effect of changing control points P1. This pulls the segments of curve in the appropriate direction and also affects, to a lesser extent, parts of the curve, thus demonstrating the important locality property of B-spline (Watt, 1999).

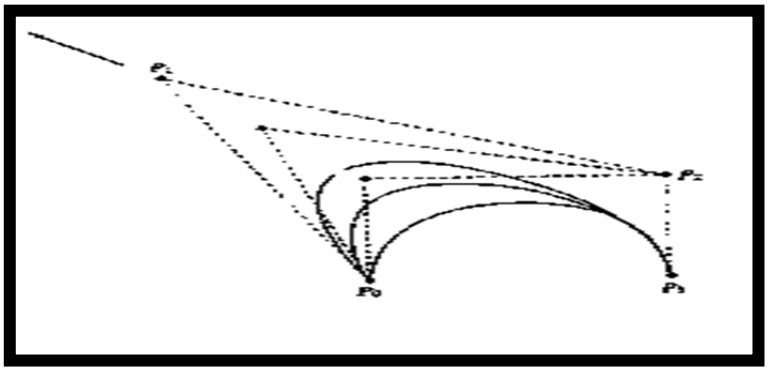


Fig. 4. The effect of changing the position of control point P1.

5. Generating 2D and 3D digital images using curve fitting techniques

A new method is proposed to generate 2D and 3D images using the parametric Lagrange curve and rolling circle movement.

5.1 Generating 2D images using the parametric lagrange curve and rolling circle movement

In this section a new method is proposed for generating a 2D image (digital image); it involves using a rolling circle moved around the parametric Lagrange curve as a tool for generating the digital image.

The image generated must have the following properties:-

a. The image must not be regular; i.e. does not contain identifiable objects or pattern and cannot be described to anyone by anybody.
b. Reproduction of the image by counterfeiters will be difficult or infeasible unless all the algorithms used to generate the image are known, as well as all the parameter values.
c. The image must have the property of random colour (pixel values), allowing the image to be used in the security field.

The process to generate such a 2D image consists of the following stages:

Stage One

- Initialise a 2D mesh of control points to generate a curve.
- Initialising a 2D mesh is achieved by selecting a set of control points according to a determined increment value between control points. This increment value, as well as those of the x-coordinate and y- coordinate, may be fixed or variable. These choices were all studied and the conclusion drawn was that the increment value plays an important role in the generated image, since any change in its value generates a new mesh of control points and will lead to a new image with new features. This property gives security to the image, because counterfeiters would have difficulty guessing the starting control points and the increment values of the x- or y-coordinates. Figure (5) shows an example of a 2D mesh of control points with equal increments to the x- and y-coordinates.

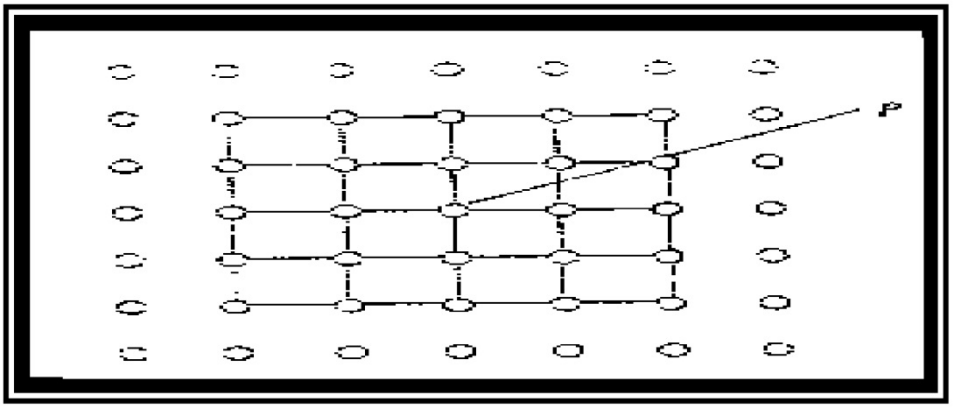
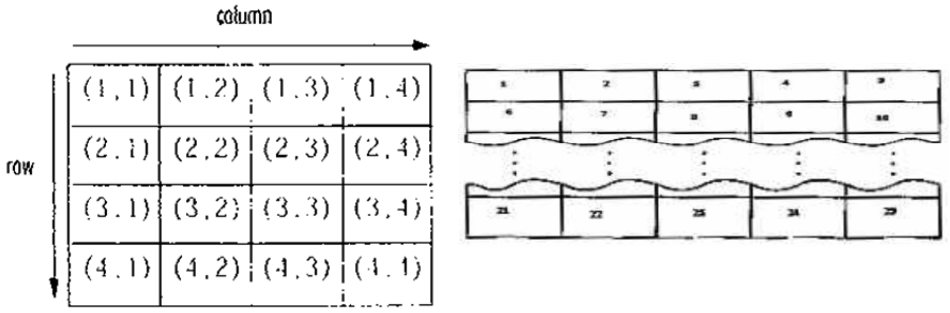


Fig. 5. Mesh of control points (p).

Stage Two

The generated curve is then moved according to algorithm (1) through the 2D mesh initialised in stage one. This process is achieved by marking the control points on the mesh using a simple method , for example 1, 2, 3… etc, and entering the number of control points from the 2D mesh into a simple pseudo-random generator. The pseudo-random generator produces a set of numbers represented as addresses of the control points in the 2D mesh; it does this in a random way and these are used to generate (interpolate) the curve. Figure (6) shows two examples of marking a 2D mesh; figure (2a) shows a mesh of size (4×4) with control points, and figure (2b) shows a mesh of size (5×5).



(a) (b)

Fig. 6. Examples for marking meshes of control points.

Stage Three

After executing stage two, the generated image boundaries are determined. A large number of recursive pixels will be obtained and spread across the computer screen according to the isolation movement of the generated curve. Determining the image boundaries is achieved by deleting all the pixels outside the fixed boundaries of the image. The size of the image (boundaries) is kept secret between the sender and the receiver.

Described below are the proposed complete algorithms for generating a 2D image using a parametric Lagrange curve with a rolling circle moved around it.

5.1.1 Algorithm (1): Generate a 2D image

Input: Input a first control point, increment value (Inc), size of mesh control points (N×N) and the image size desired.

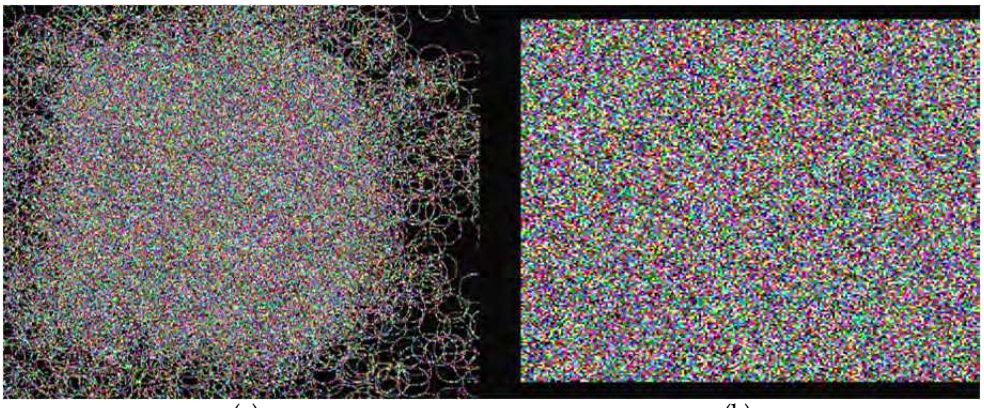
Output: Generate a 2D digital image.

Process:

Step 1. Initialise the mesh of control points according to the start control points; increment the value and the size of the mesh.
Step 2. Mark the control points of the mesh.
Step 3. Enter the number of marks to simple pseudo-random generator.
Step 4. Take the sequence of output from the generator to represent the addresses of the set of control points in the mesh.
Step 5. draw the parametric Lagrange curve with rolling circle.
Step 6. Repeat step 4 with new sequence of numbers and step 5 until obtaining the recursive pixels that covers the image size that need to generate.
Step 7. Clip the image according to the size the user entered.
Step 8. Obtain the 2D generated image.
Step 9. End.

Example:

In the following example a 2D image is generated using a mesh size of (25×25) with the same increment value of x- and y-coordinates equal to (10), using a radius for the rolling circle equal to (15), and the image size required to be generated is equal to (256×256) pixels. Figure (7a) shows the random oscillation curve movement through the 2D mesh is due to the movement of the curve path out of the mesh boundary. Figure (7b) shows the final stage of generating the 2D image by clipping the image size to (256×256) pixels.



(a) (b)

Fig. 7. (a) The random oscillation curve movement through the 2D mesh, (b) image of size (256x256) pixels

5.2 Generating a 3D image

This section introduces a new idea for generating a 3D image by inserting a third coordinate (z) to all the mathematical equations that were used to generate the 2D image. Generating a 3D image is very useful for increasing the key space and thus renders the process of estimating the key infeasible. This is because searching in 3D increases the number of possible control points the counterfeiter needs to estimate. In order to imagine working with 3D, it is suggested that the image be viewed as an empty cube which is to be filled completely with random colour pixels, then slides from the cube will be clipped in different ways. The clipped slide from the three-dimensional image can be used directly as a secret key, or by clipping a curve from this slide according to a secret set of control points by using a parametric Bezier curve (control curve) to represent the secret key between the sender and receiver.

5.2.1 Proposed algorithm to move a 3D rolling circle around a 3D parametric Lagrange curve

To create a parametric form from a 3D curve, it is necessary 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 (Hill, 2000). The circular helix is given parametrically by: x (t) = cos (t), y (t) =sin (t) and z (t)=bt. For some constant b. The curve is illustrated in figure (8):

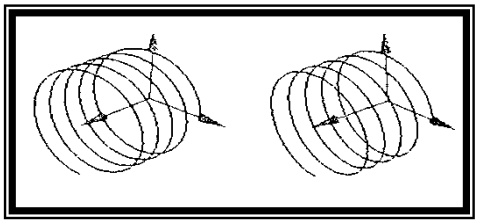


Fig. 8. Circular helix.

Many variations on the circular helix are possible. For example, 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 into a helix by appending z(t)= bt, or some other form for z(t).

5.2.2 Algorithm2: (Move a 3D rolling circle around a 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 centre coordinates (x-c, y-c, z-c),and the radius circle value.

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

Process:

Step 1. Set radius=15, b=10
Step 2. 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)
Step 4. Next index
Step 5. Next i
Step 6. End.

5.2.3 Generating 3D image stages

In section (7.2.1) the method for generating the 3D image by using 3D mathematical models was described and the algorithms needed to perform this generation were explained. In addition, it was suggested that the 3D image resembles an empty cube and the aim is to fill the cube completely with random colour pixels. This process involved a set of different stages, including: initialising a 3D mesh of control points; marking the control points to produce a sequence of control points by using a pseudo-random generator; implementation of algorithm (2) to move the 3D Lagrange curve with a 3D rolling circle through the 3D mesh of control points; deleting all the pixels outside the cube; and the final stage is obtaining the cube (image) which is completely filled with random colour pixels. The following illustrates the stages for generating a 3D image:-

Stage One

3D shape is to be divided into slides (planes) as each slide takes a 2D matrix, its axes are x and y, and the number of their slides will be equal to the depth z-axis of the shape. Figure (9 shows an example of an initialised 3D mesh.

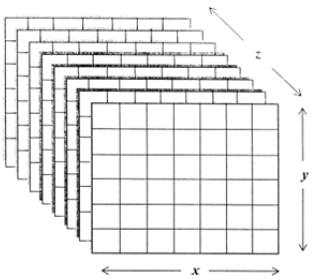


Fig. 9. Initialised 3D mesh.

The initialising of a 3D mesh is achieved using the same algorithm that was used to generate the 2D mesh of control points, but by inserting the third coordinate (z) to the control point coordinates, and determining the increment values of the x-, y-, and z-coordinates. In addition, when changing the increment values of the x-, y-, and z-coordinates, we obtain different types of 3D mesh control points, which makes the process of estimating the mesh more difficult for counterfeiters. Figure (10) shows an example of the cubic mesh (3D mesh) with equal increment values (x, y and z) between the control points.

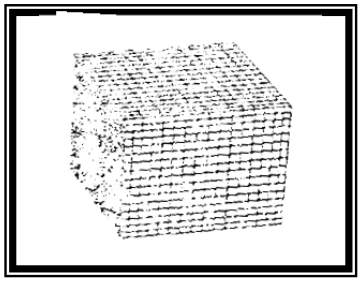


Fig. 10. Example of cube mesh.

Stage Two

Marking the 3D mesh is achived by counting the 3D mesh control points. The cube mesh can be visualised as resembling a set of 2D mesh slides marked with page numbers (z-coordinates), or an array of three dimensional coordinates.

Stage Three

A pesudo-random generator is used to produce the random addresses of control points from the 3D mesh, which are then used as a set of control points to implement algorithm (2) for moving the 3D curve (parametric Lagrange curve) with the 3D rolling circle through the 3D mesh. Figure (11) shows an example of the curve moving through the 3D cube.

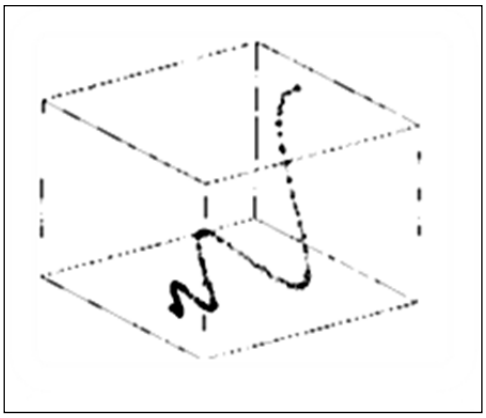


Fig. 11. Example of moving curve through cube.

Stage Four

This stage eliminates all the pixels drawn out of the cube due to the isolation of the moving Lagrange curve. The deleting process aims to remove all the pixels outside the 3D shape, and this is implicitly implemented in the algorithm which computes pixels and tests whether their coordinates are outside the boundaries of the 3D coordinates.

The complete algorithm below shows how to generate a 3D image using 3D mathematical models.

Algorithm 3: Generating a 3D Image Using 3D Mathematical Model

Input: Identify the 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 colour values in 3D array size (N×N×N).

Process:

Step 1. Initialise a 3D mesh of control points according to the starting control point; increment the value and the size of the 3D mesh.
Step 2. Mark the control points of the 3D mesh.
Step 3. Enter the number of marks from 3D mesh into a simple pseudo-random generator.
Step 4. Take a sequence of numbers from the pseudo-random generator to represent the addresses of the control points in the mesh.
Step 5. Execute algorithms (1) and (2) to draw the parametric Lagrange curve with a rolling circle and save the resulting colour pixel values for points in 3D array size (N×N×N).
Step 6. Repeat step4 with a new sequence of control point numbers and step 5 until the cube is completely filled with recursive pixels.
Step 7. Delete all the pixels which were placed outside the cube due to the isolation-moving curve.
Step 8. Obtain 3D generated image.
Step 9. End.

Example

To explain the work of the algorithm, figure (5) shows the implementation of algorithm (8). This example used the 3D size (100×100×10), (i.e. the 3D consisted of 100,000 random colour pixels) that was used to generate a 3D mesh of control points, and the 3D mesh of size (50×50×50), (i.e. the mesh consisted of 125,000 control points), and equal increment values between the control points. Figure (12) shows the results:

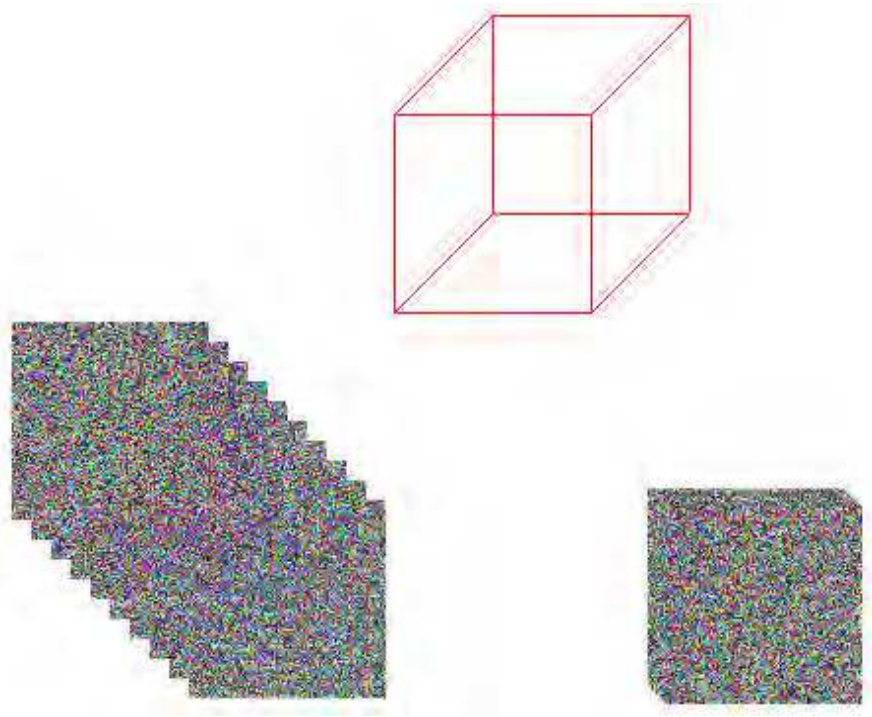


Fig. 12. Example to generate 3D-image

6. Clipping symmetric keys from the 2D and 3D images using the bezier curve method

The weakness of a key that is generated from a normal colour image is clear due to the nearest pixel values of the image. A new method for generating a symmetric cryptographic key is proposed which works by generating 2D and 3D images according to mathematical curve equations which will make the key sufficiently robust. A symmetric key is a string of random bits, and the number of random bits in it determine the key's variability and strength. Cryptographers recommend that to be reasonably secure, keys should be at least 90 bits long. The world standard is 128 bits because this is a convenient size for computers; there is no technical reason to use a shorter key. The second type of encryption is the public key or asymmetric systems, which uses separate keys for encryption and decryption: private key and the public key.

6.1 Proposed algorithm for clipping symmetric key from the (2D and 3D) images using the bezier curve method

Control points play an important role in curve generation as well as in clipping cryptography keys from 2D images and from the slides of 3D images. Control points are used to clip a curve or part of the generated images (2D & 3D). For this reason the control point must remain secret between the sender and receiver, (i.e. control points represent the master key used to clip the secondary key formed by a curve from random pixel colour values).

According to cryptography principles, it is assumed that all algorithms used are public to the attacker and only the key is secret. The secret key consists of a number of control points used to clip the key, and the coordinates of the control points; for example (4,10,10,20,15,30,60,100) which means using 4 control points to clip the curve and use the coordinates respectively (10,10),(20,15),(15,30),(60,100). Additionally, we can change any one of the parameters for the secret key to obtain a new secret key; this makes the process of generating the key more flexible and efficient.

There are different ways to select the control points from a 2D mesh or from a 3D mesh with different increment values between the control points and the control points themselves; this makes the counterfeiter's process of estimating the primary key infeasible. Figure (13) shows an example for selecting a set of control points to clip the curve of points using the Bezier method.

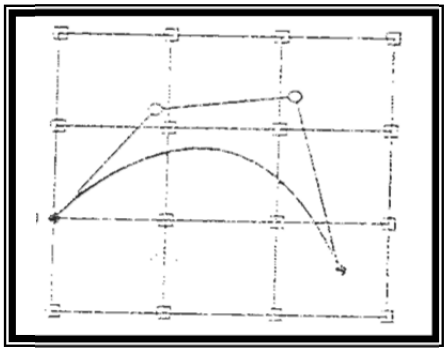


Fig. 13. Example for clip key from 2D mesh using Bezier Curve.

The size of the clipped key from the generated image is flexible depending on the flexibility of the generated image size; for example in the 2D image, the image size used is 256×256 pixels, which is equal to 65,536 pixels with each pixel represented by 24 bits (i.e. the key size is 1,572,864 bits), and the key space used in the 3D image is (100×100×100) pixels, which is equal to 1,000,000 pixels (i.e. the key space size is 24,000,000 bits).

In this example, samples of different key sizes were clipped, and the randomness of the keys was tested according to the five popular tests for randomness mentioned above.

Figure (14) shows an example of how to clip slides from a 3D image; figure (15) shows how to clip a curve as a key from a 3D slide.

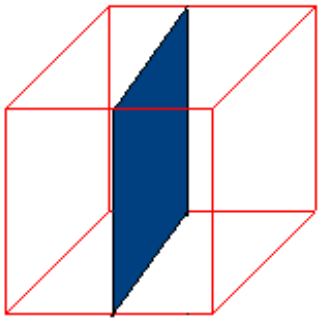


Fig. 14. Example of clipping slides from a 3D image.

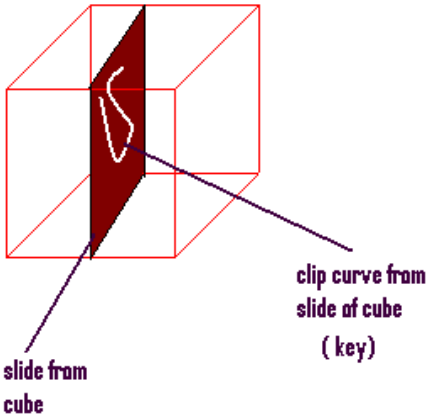


Fig. 15. Example of clipping a curve from slide of cube.

The following illustrates the algorithm for clipping a key from a 2D image and clipping a key from slides of a 3D image.

6.1.1 Algorithm(4): Generating key from generated image using Bezier curve method or by clipping parts of the generated images directly

Input: The sender inputs the number of control points to be used for the generation of the key (N), then inputs the chosen set of control point coordinates (x,y,z) from the mesh of the 2D or 3D image according to the boundaries of the images (i.e. the sender must know the boundaries of the generated image before clipping the keys).

Output: Clip stream of colour pixels from generated image (2D or 3D) using the Bezier Curve method or clip part (slide) from the generated image directly.

Process:

Step 1. Assignment of the set of control points entered by sender to two arrays x (.), y (.).
Step 2. The user inputs his choice for clipping key from 2D or 3D and using curve equations for clipping the key or clipping part from the generated image directly.
If (clipped the key from 2D-image) then go to (**A**)
Otherwise, go to (**B**)
(**A**) (Clipped Key From 2D-Generated Image).
If (Clipped Key Using Bezier Curve Equation) then go to step3.
Otherwise, go to step 12.
Step 3. (Clipped Key Using Bezier Curve Generation).
Set i=0
i<= N-1 do
For u= 0 to 1 step 0.01
Step 4. Let $X = (1-u)^3 x_i + 3(1-u)^2 u x_{i+1} + 3(1-u)u^2 x_{i+2} + u^3 x_{i+3}$
Let $Y = (1-u)^3 y_i + 3(1-u)^2 u y_{i+1} + 3(1-u)u^2 y_{i+2} + u^3 y_{i+3}$
Step 5. Get Pixel (*X, Y*),
Step 6. Open file (Clip.txt) to save the pixel colour for pairs (*X, Y*).
Step 7. Next u
Step 8. i=i-1,
Loop
Step 9. Convert the Contents the files (Clip.txt) to the binary digits (Bin.txt), which represent stream of bits (symmetric key).
Step 10. Execute the five-randomness test on the binary files (Bin.txt) for checking the randomness of the key (i.e. Step10 is an optional step the user can cancel this step).
Step 11. End.
Step 12. (Clipped part from generated image to represent the key). {When the sender decides to clip directly from the 2D-generated image, he chooses only two corner coordinates from the mesh that represent the two corners of a rectangle}.
Step 13. Input the two control points (x1, y1) and (x2, y2).
Step 14. For i= x1 to x2
For j= y1 to y2
Get pixel (i,j)
Step 15. Open file (Clip-p.txt) to save the pixel colour for pairs (i,j).
Step 16. Next j
Next i
Step 17. Convert the Contents of files (Clip-p.txt) to binary digits (Bin.txt), which represent stream of bits (symmetric key).
Step 18. End.

(Clipped Key From 3D-generated Image)

{Work with a 3D image requires user to determine the level of (z)- coordinate from the cube, which is represented in this case by the slide number from the cube to clip 2D slide from 3D image.}

Step 19. Step19: Input the 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 colour pixels from the three dimensions array (3D).
Step 20. Set z= z-coordinate
For i= x_1 to x_2
For j= y1 to y2
slide (i, j) =3D(i , j, z)
Next j, Next i.
Step 21. Go to (A).

6.2 Tests of randomness for the clipped keys

Different sizes of keys are used in these tests and the results prove that the keys have the randomness property and can be used as a symmetric key in the cryptography field.

Tests		Key5= 1024 bit	Key6= 2604 bit	Key7= 3844 bit	Key 8= 5084 bit	Pass value
Frequency test		0.098	0.498	1.000	0.964	Must be ≤ 3.84
Run test	T_0	9.289	13.028	20.680	13.249	Must be ≤ 22.362
	T_1	6.656	3.867	5.475	8.127	
Poker test		7.409	1.568	3.077	3.497	Must be ≤ 11.1
Serial test		4.016	1.705	6.504	5.899	Must be ≤5.99
Auto correlation test for ten bits	Shift 1	0.079	0.031	0.075	0.087	Must be ≤3.48
	Shift 2	3.068	3.086	1.268	2.240	
	Shift 3	0.024	0.019	0.094	0.044	
	Shift 4	0.004	0.886	1.667	1.196	
	Shift 5	0.048	0.010	0.115	0.884	
	Shift 6	0.035	0.006	0.104	0.911	
	Shift 7	0.048	0.003	0.284	0.640	
	Shift 8	0.476	0.006	0.652	0.013	
	Shift 9	0.616	0.420	1.102	0.331	
	Shift 10	0.063	1.124	0.104	0.133	

Table 1. the results of randomness test for the keys clipped from 2D & 3D images.

6.3 Using generated digital images to modify cryptography protocol (PGP)

The strong cryptography employed by PGP is the best available today. The PGP protocol is a hybrid cryptosystem that combines some of the best features of both conventional and public-key cryptography. In this section, we propose to insert the generated digital image capability into the PGP protocol stages in order to increase protocol robustness in the face of counterfeiters. According to the tests in the previous sections, the generated 2D or 3D images have the randomness property and can clip many keys with different sizes. These properties make the generated images very useful as a source of clipping randomness session keys, and there is the capability to generate many keys from the images. We propose to use the generated (3D or 3D) images facility to generate a session key instead of using movement of the mouse or keystrokes. The session key clipped from the generated image in this case consists of two session keys; a *primary session key* and a *secondary session key*. The primary session key represents the Bezier curve coordinates and the secondary session key represents the stream of randomness bits sequence that is clipped according to curve equations. The work with New-PGP begins when a user encrypts plaintext. *Firstly,* the plaintext is compressed. *Secondly,* a session key is created by generating 2D or 3D images according to the proposed algorithms. *Thirdly,* the user enters the primary session key to the clip secondary key from the digital image. *Fourthly,* the user XOR the stream of bits sequence (secondary session key) with the plaintext after the compression process. *Fifthly***,** the sender uses the public key from the RSA algorithm to encrypt the primary session key. *Sixthly,* the sender transmits the encrypt primary session key along with the ciphertext to the recipient. Decryption works in the reverse order. The recipient's copy of new-PGP uses his or her private key from the RSA algorithm to recover the primary session key which is used to generate the secondary key from the generated image to decrypt the conventionally encrypted ciphertext. Figures (16) and (17) show the send and receive process according to new-PGP protocol.

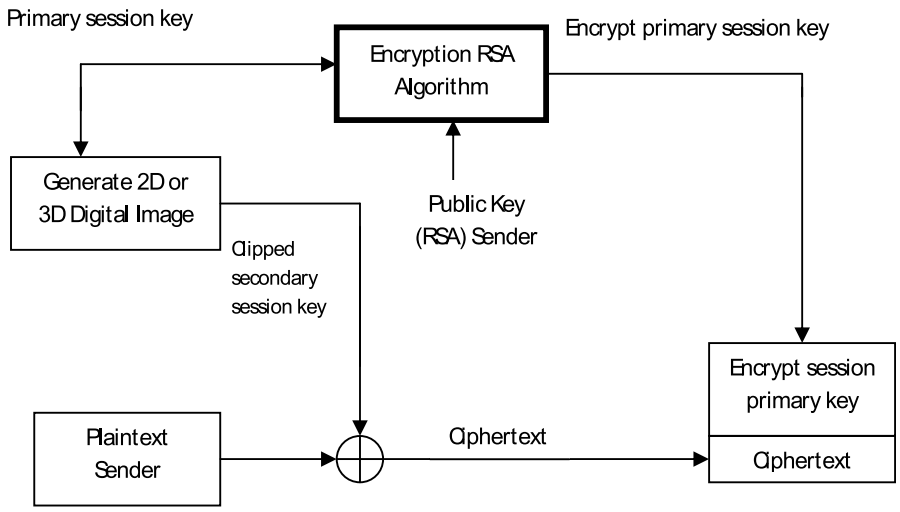


Fig. 16. Send Process

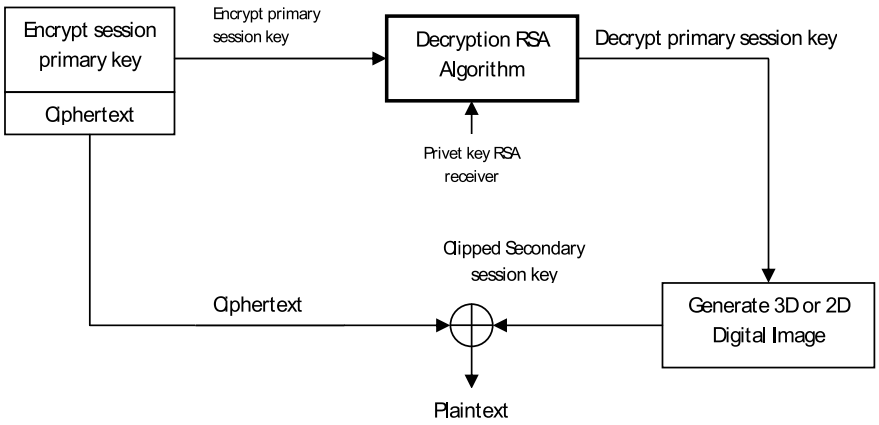


Fig. 17. Receive Process.

Example:-

To explain how the New-Protocol works and indicate the protocol behaviour, we proposed to generate a 3D-image size (100×100 ×100) pixels, by using a mesh size of (50×50×50) control points, and a primary session key (PSK) that consists of (4) control points with the coordinates (10,10), (20,20), (30,30), (40,40), with increment step u equal to 0.01. According to the primary session key, we clipped a secondary session key size (SSK) equal to 260 random bits. According to the randomness tests in table (1), the public key (PK) of RSA algorithm consists of (n=997517,e=193) where (secret p=977) and (secret q=1021), and the private key of RSA algorithm equals (d=727297).

Figures (18) and (19) illustrate the proposed protocol with the example values:

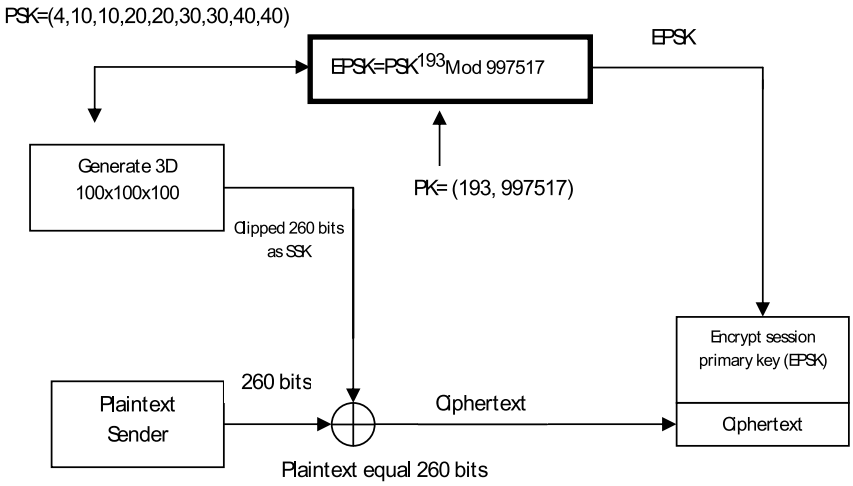


Fig. 18. Send Process.

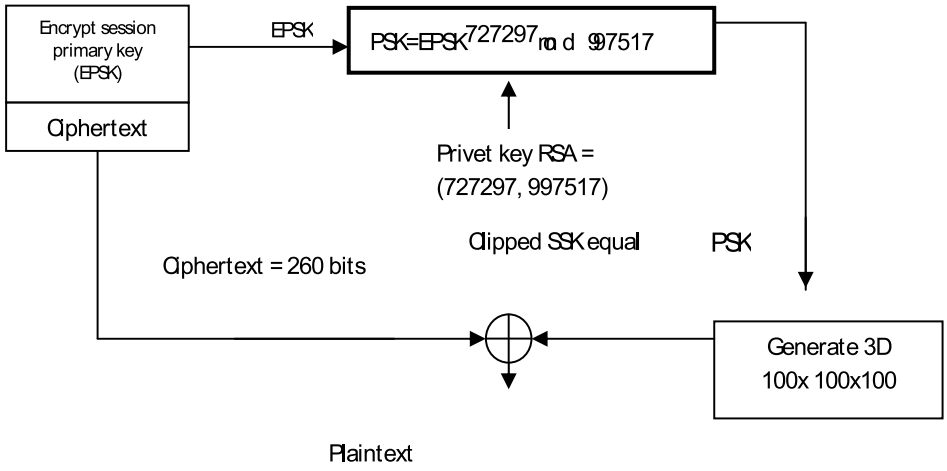


Fig. 19. Receive Process.

7. Conclusions

The proposed methods in this chapter are starts by studying most of the curve fitting methods and selects the curve fitting methods that are suitable to the work. Secondly developing the curve security algorithm and proposed algorithm to generate (2D & 3D) digital images, thirdly producing algorithm to clipping symmetric cryptographic key from 2D or 3D generated image, fourthly proposing algorithm to modify PGP cryptographic protocol by using 2D or 3D generated images.

Finally implementing the proposed generated digital image in image cryptography technique and then testing the results according to the authorized measures that are used in this field.

In this chapter proved a combine curve security with cryptography algorithms increase the cryptography capability. The proposed methods gives reasonable results in generating many randomness keys with different sizes. The 2D & 3D mathematical models succeed to generate randomness digital images that can play important role in cryptography according to the results that obtain from authorized randomness tests and from image cryptography tests, which gave reasonable tests.

From the New-PGP method, we obtained the following information:-

1. The process of guessing the primary session key from the secondary key is infeasible, because there is no correlation between the two session keys.
2. If the counterfeiter succeed in solving the factorisation problem from RSA and found the private key from public key, the key obtained would not help him to recover the plaintext from the primary session key unless the secondary session key was known.
3. All the secondary session keys have the property of randomness according to the randomness tests.
4. The New-PGP increased the security of the PGP protocol, making the protocol more robust and efficient.

8. References

Watt, A. (1999). *3D Computer Graphics (3rd edition)*, Addison Wesley, ISBN 978-0201398557

Demel, J. & Miller, M. (1984). *Introduction to Computer Graphics*, Brook/Cole Pub Co, ISBN 978-0534030537, USA

Droste, S. (1996). *New Result on Visual Cryptography*, CRYPTO 96, Proceedings of the 16th Annual International Cryptology Conference, pp. 401-415, Santa Barbara, CA, USA, August 18-22, 1996

Egerton, P.A. & Hall, W.S. (1998). *Computer Graphics: Mathematical First Steps,* Prentice Hall, ISBN 978-0135995723, London, UK

Goldman, R. (2002). *Lagrange Interpolation and Neville's Algorithm,* Department of Computer Science, Rice University, Available from www.clear.rice.edu/comp360/lectures/lagra.pdf

Gomes, J.; Velho, L. (1997). *Image Processing for Computer Graphics,* Springer-Verlag, ISBN 0-387-94854-6, New York, USA

Guan, P. C. (1987). *Cellular Automaton Public-key Cryptosystem,* Complex Systems, Vol1, Issue 1, (1987), pp.51-57, Available from,www.complex-systems.com/pdf/01-1-4.pdf

Harrington, S. (1987). *Computer Graphics a Programming Approach*, McGraw-Hill, ISBN 978-0070267534

Hill, F. S. (2000). *Computer Graphics Using OPENGL (2nd* edition), Prentice Hall, ISBN 978-0023548567

Menezes, A.; Van Oorschot, P. & Vanstone, S. (1996). *Handbook of Applied Cryptography (1st edition)*, CRC Press, ISBN 978-0849385230

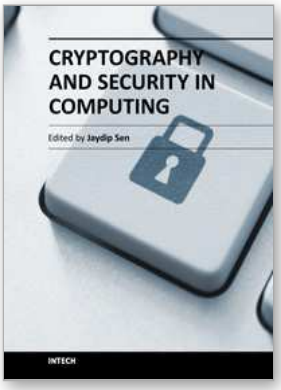
Pham B. (1988). *Offset Approximation of Uniform B-splines*, Computer Aided design, Vol 20, Issue 8, (October 1988), Elsevier Ltd, pp. 471-474

Stallings, W. (2005). *Cryptography and Network Security: Principles and Practice (4th edition)*, Prentice-Hall, ISBN 978-0131873162, USA

Sauer, C.; Chandy, K. (1981). *Computer Systems Performance Modeling*, Prentice-Hall, ISBN 978-0131651753

Schaefer, E. (1999). *An introduction to Cryptography and Cryptanalysis*, Santa Clara University, Available from http://math.scu.edu/-eschaefe/crylec.pdf-united states

Schneir, B. (1996). *Applied Cryptography (2nd edition)*, John Wiley & Sons, ISBN 978-0471117094



Cryptography and Security in Computing

Edited by Dr. Jaydip Sen

ISBN 978-953-51-0179-6
Hard cover, 242 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

The purpose of this book is to present some of the critical security challenges in today's computing world and to discuss mechanisms for defending against those attacks by using classical and modern approaches of cryptography and other defence mechanisms. It contains eleven chapters which are divided into two parts. The chapters in Part 1 of the book mostly deal with theoretical and fundamental aspects of cryptography. The chapters in Part 2, on the other hand, discuss various applications of cryptographic protocols and techniques in designing computing and network security solutions. The book will be useful for researchers, engineers, graduate and doctoral students working in cryptography and security related areas. It will also be useful for faculty members of graduate schools and universities.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821