Using Generated Digital Images to modify the PGP Cryptography Protocol

Dr. Hilal M. Yousif Al-Bayatti
Amman Arab University
For Graduate Studies,
Amman, Jordan.
hilalyousif@yahoo.com

Dr. Abdul Monem S. Rahma
Computer Science Department,
University of Technology,
Baghdad, Iraq.
monemrahma@yahoo.com

Dr. Hala Bahjat AbdulWahab
Computer Science Department,
University of Technology,
Baghdad, Iraq.
hala_bahjat@yahoo.com

Abstract-*The strong cryptography employed by PGP is one of 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 paper, we propose to use a generated digital images capability in the PGP protocol stages to increase protocol robustness and to make the protocol more difficult in front of the counterfeiter.*

Keywords: Applied Cryptography, Graphics, Image Generation, Random Number Generator, and PGP Protocol.

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 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 2D image using parametric Lagrange curve and rolling circle movement

In this section, a new method proposed to generate 2D image (digital image) using moving rolling circle around the parametric Lagrange curve that as a tool for generating the digital image.

The image that we want to generate must have the following properties:-

a. The image must not be regular, i.e. does not contain identifiable objects or pattern and cannot described to any one by any body.
b. It is difficult or infeasible reproduction of the image by counterfeiter unless one knows all the algorithms used to generate the image are known and all the parameter values.
c. The image must have randomness color property (pixels values) that makes the image useful in security field.

The process to generate 2-dim image consist of the following stages:

Stage One

Initialize 2D-mesh of control points that used to generate a curve according to algorithm (1).

Initializing 2D-mesh which achieved by selecting a set of control points according to a determined increment value between control points. Increment value for x-coordinate or y- coordinate or both, and the increment value may be a fixed value or a variable value.

All these choices studied and we concluded that the increment value plays an important role in the generated image, since any change in increment value generates new mesh of control points and will lead to a new image with the new features. This property gives security condition to the image, because the counterfeiter will face a difficult process to guess the start control points and the increment value of the x-coordinate or of the y-coordinate or both.

Figure (1) shows an example of a 2D- mesh of the control points with equal increment to x-, y-coordinates.

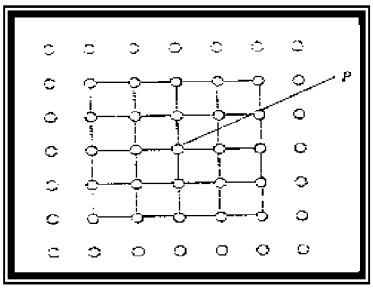


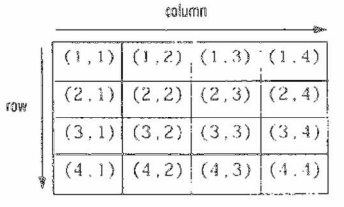
Figure (1): Mesh of control points (p).

Stage Two

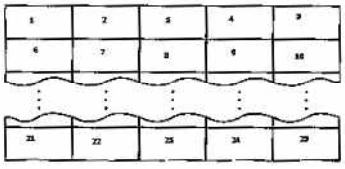
In this stage, we are moving the generated curve according to algorithm (2) through the 2D-mesh that initialized in stage one.

This process is achieved by marking the control points on the mesh by a simple way like counting the control points for example 1, 2, 3…and so on, and entering number of control points of 2D-mesh to a simple pseudo-random generator.

The pseudo-random generator gives each time a set of numbers that is represented as addresses of the control points in the 2D-mesh in a random way using to generate (interpolate) the curve. Figure (2) shows two examples for marking 2D-mesh;where figure (2-a) shows an example of marking a mesh of size (4×4) of control points and Figure (2-b) shows an example of marking a mesh of size (5×5).



(a)Mesh size (4×4)



(b) size (5×5)

Figure (2): Examples for marking meshes of control points.

Stage Three

In this stage, after executing stage two, determining the generated image boundaries performed. A large number of recursive pixels will be obtain and spread on the screen of the computer by an isolation way according to the isolation movement of the generated curve.

Determining the image boundaries is achieved by delete all the pixels out of the fixed boundaries of the image. The size of the image (boundaries) is suitable to keep secret between the sender and the receiver only. In the following, we describe the proposed complete algorithms for generating 2D image using parametric Lagrange curve with a moving rolling circle around it.

Algorithm (1): Generate curve using parametric Lagrange method.

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

Output: Generate 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
Step7: loop,
k=k+1, loop,
Step8: Goto from step 1 to 4.
res =re s+ (P*y (i))
next i
Step9: New_y(h)=res, res=0,h=h+1,
Temp(h)=Temp(h-1)+step
Loop, k=k+1, loop
Step10: For i= 1 to h
Plot (New-x, New-y)
Next i
Step11: End.

Algorithm (2): Moving the rolling circle around Lagrange curve.

Input: Take pairs of data (New-x_i, New-y_i), i=0,..,h-1, that computes in algorithm (1) to represent the x –coordinate (x-c) and y-coordinate (y-c) to center of circle, and input the radius circle.

Output: Moving rolling circle around parametric Lagrange curve that generate in algorithm (1).

Process:

Step1: Set radius=15
Step2: For i= 0 to h-1
Set x-c= New-x, y-c=New-y
For index=0 to 360
X=radius*cos (index) +x-c

Y=radius*sin (index) +y-c
Step3: Plot (X, Y)
Step4: Next index, Next i
Step5: End.

Algorithm (3): Generate 2D Image

Input: Input a first control point, increment value (Inc), size of mesh control points (N×N) and size of the image that want to generate.
Output: Generate 2D digital image.
Process:
Step1: Initialize the mesh of control points according to the start control points; increment the value and the size of the mesh.
Step2: Mark the control points of the mesh.
Step3: Enter the number of marks to simple pseudo-random generator.
Step4: Take the sequence of output from the generator to represent the addresses of the set of control points in the mesh.
Step5: Perform the Algorithms (1) and (2) to draw the parametric Lagrange curve with rolling circle.
Step6: Repeat step4 with new sequence of numbers and step 5 until obtaining the recursive pixels that covers the image size that need to generate.
Step7: Clip the image according to the size the user entered.
Step8: Obtain the 2D generated image.
Step9: 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-,y-coordinates equal to (10), and using a radius for the rolling circle equal to (15), and the image size that need to be generated is equal to (256×256) pixels.

Figure (3) shows the oscillation curve movement by random way through the 2D-mesh is due to the movement of the curve path out of the mesh boundary.

Figure (4) shows the final stage of generating the 2D-image by clipping the image size to (256×256) pixels.

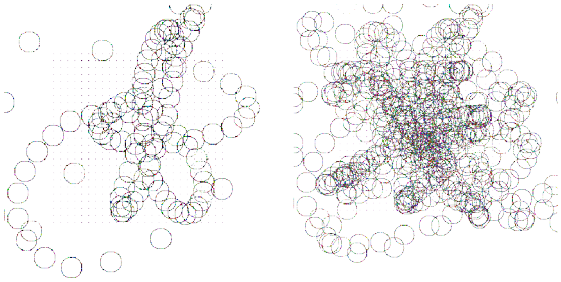


Figure (3): Example shown the oscillation moving curve through 2D-mesh.

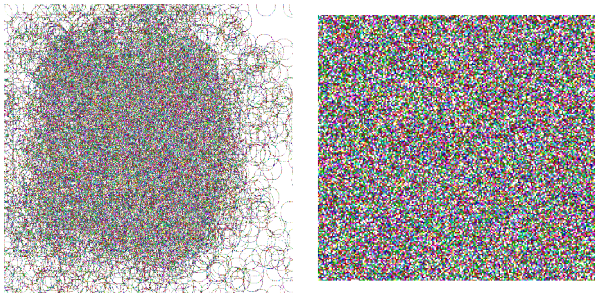


Figure (4): Example to clip image size 256×256 pixels.

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 [4].

PGP combines some of the best features of both conventional and public-key cryptography. PGP is a hybrid cryptosystem [5].

When a user encrypts plaintext with PGP, PGP first compresses the plaintext. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security. Most 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 that 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 your mouse and the keystrokes you type. The session key works with a very secure, fast conventional encryption algorithm to encrypt the plaintext; the result is a ciphertext. Once the data encrypted, the session key is then encrypted to the recipient's public key [6]. This public key-encrypted session key transmitted along with the ciphertext to the recipient. Figure (5) shows the send process.

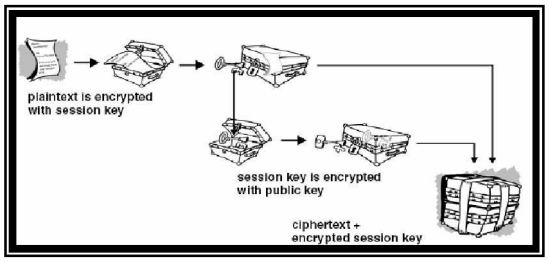


Figure (5): Send process.

Decryption works in the reverse order. 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 (6) shows the received process.

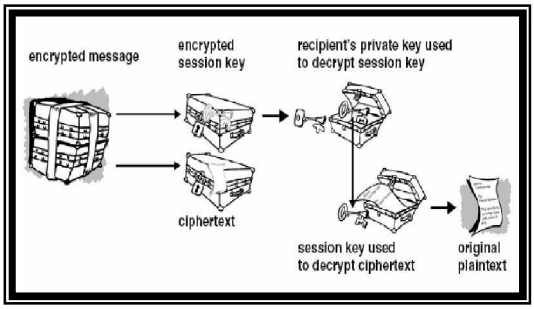


Figure (6): Received process

4. Cryptography Protocol (PGP) Modifications.

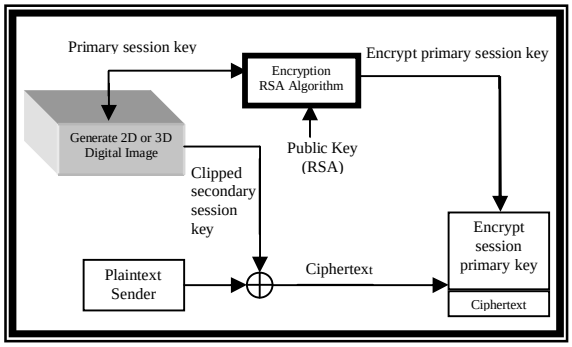
The strong cryptography employed by PGP is the best available today. The PGP protocol is a hybrid cryptosystem that combines some of the best features of both conventional and public-key cryptography.

In this section, we propose to insert the generated digital images capability to the PGP protocol stages to increase protocol robustness and make the protocol more difficult in front of the counterfeiter.

The generated (2D or 3D) images have the randomness property according to the results of randomness tests (five popular randomness testes) after clipping process to many keys with different sizes. These properties make the generated images very useful to be used as a source of clipping randomness session keys and the capability to generate many keys from images help to use these session keys a one-time-only. We propose to use the generated (3D or 3D) images facility to generate session key instead of generating the session key by using movement of the mouse or the keystrokes type. The clipped session key from the generated image, in this case consist of two session keys, *primary session key* and *secondary session key*.

Primary session key represent the Bezier curve coordinates and secondary session key represent the stream of randomness bits sequence that is clipped according to curve equations. The works with New-PGP begin when a user encrypts plaintext. *First,* compress the plaintext (we mentioned the reason for compression). *Second,* creates a session key by generating 2D or 3D images according to the proposed algorithms in this paper. *Third,* the user enters primary session key to the clip secondary key from digital mage. *Forth,* the user XOR the stream of the secondary session key bits sequence with the plaintext after compression process. *Fifth,* the sender uses the public key from RSA algorithm to encrypt the primary session key. *Sixth,* the sender transmits the encrypted 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 RSA algorithm to recover the primary session key that is used to generate the secondary key from generated image to decrypt the conventionally encrypted ciphertext.

Figures (7) and (8) show the send and received process according to new-PGP protocol.



Figure(7):Send Process

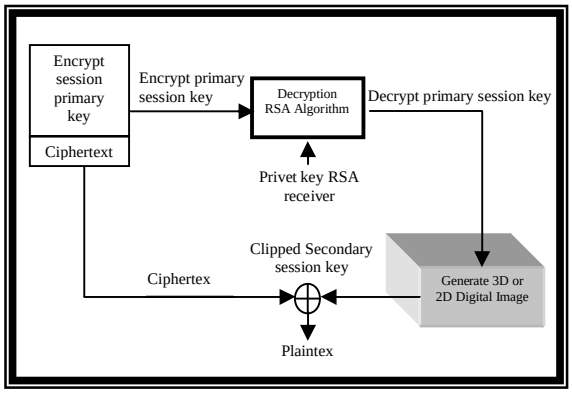


Figure (8): Received Process

Example

To explain how the New-Protocol works, and indicate the protocol behavior. We proposed to generate a 3D-image size (100×100 ×100) pixels, by using mesh size of (50×50×50) control points, and use a primary session key (PSK) that consists of (4) control points with the coordinates (10,10), (20,20), (30,30), (40,40), and with increment step u equal to 0.01. According to the primary session key we clipped a secondary session key (SSK) of size equal to 260 random bits. The public key (PK) of the RSA algorithm consist of (n=997517,e=193) where (secret p =977) and (secret q=1021), and the private key of RSA algorithm equal (d=727297).Figures (9) and (10) illustrates the proposed protocol with the example values.

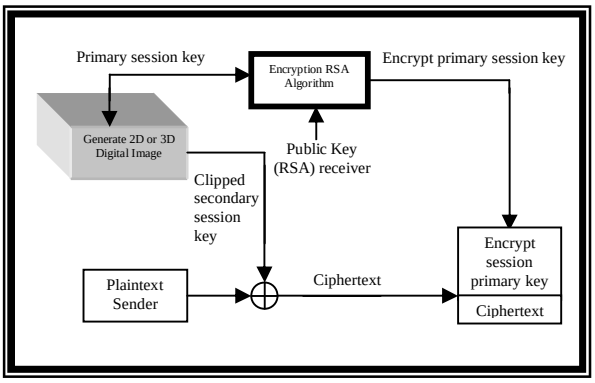


Figure (9): Send Process

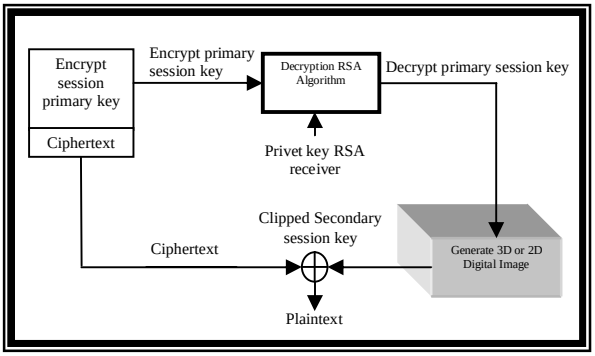


Figure (10): Receive Process.

5. Application

To explain how the New-Protocol works, and indicate the protocol behavior.

We proposed to generate a 3D-image size (100×100 ×100) pixels, by using mesh size of (50×50×50) control points, and use 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 secondary session key (SSK) of size equal to 260 random bits, the public key (PK) of RSA algorithm consist of (n=997517,e=193) where (secret p =977) and (secret q=1021), and the private key of RSA algorithm equal (d=727297). Figures (11) and (12) illustrate the proposed protocol with the example values.

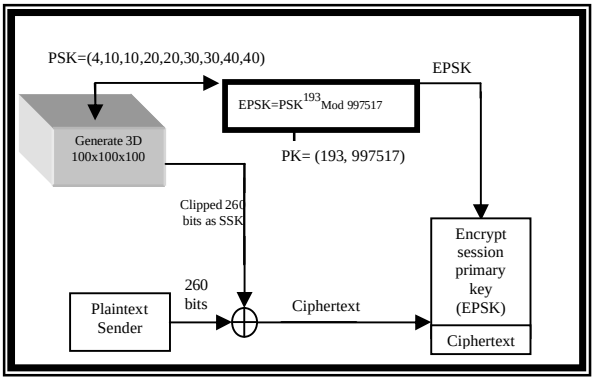


Figure (11): Send process

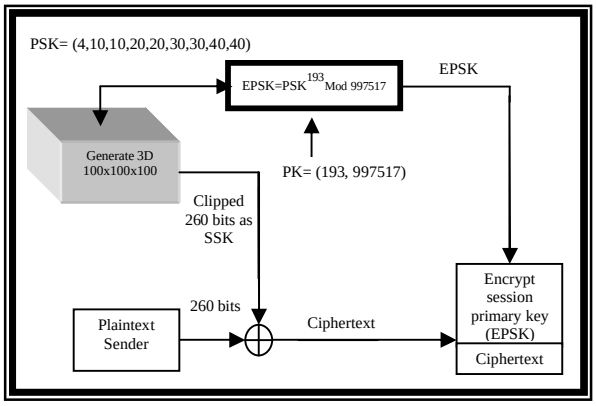


Figure (12): Receive process

6. Conclusions

From the New-PGP method, we reached to the following conclusions:-

a. The process to guesses the primary session key from the secondary key is infeasible, because there is no correlation between the two session keys.
b. If the counterfeiter succeed to solve the factorization problem from RSA , and find the private key from public key , the key that is obtained can not help him to recover the plaintext from primary session key unless knowing the secondary session key.
c. All the Secondary session keys have randomness property according to the randomness tests.
d. The New-PGP increased secure condition to the PGP protocol that made the protocol more robust and efficient.

7. References

[1] Denning D. E., "Cryptography and Data Security"; Addison-Wesley Publishing Company, Inc, 1983.

[2] Newman W.M., "Sproull R.F.,"Principles of Interactive Computer Graphics"; Mc Graw-Hill Book Company London, 1981.

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

[4] Schaefer E. D. "An introduction to Gryptography"; Santa Clara University, 1999.

[5] GPG Corporation,"An Introduction of Cryptography"; www.pgp.com, 2004.

[6] Lewis, P., Goodman, A., and Miller, J."A Pseudo-Random Number Generator for the System/360."; IBM systems Journal, No.2 ,1969.