ISSN:2231-0711

Software Protection from Piracy using Elliptic Curve

Dr. Abdul Kareem Murhij Radhi.
Nahrain University- Faculty of Information Engineering – Baghdad- Iraq

***Abstract*------- Due to the rapid growth of Internet users in the last decades and illegal use of software from piracies without granting authorities from legal vendors, besides expensive of software products and draw backs in distribution, this paper offers a suitable technique to protect software from piracy. It relies on elliptic curve cryptography technique achieved via processor ID of the buyer or user. The bought software after downloading will be protected from illegal distribution through mechanism of elliptic curve with ElGamal cryptographic technique. In order to protect server from piracies and latencies causes from network communication, this paper offers software downloading during virtual computer or VMware. Time consuming for encryption and decryption will be shown. This paper concludes that using Elliptic curve offers equal security for small bit size, thereby reducing processing overhead. Software implemented using Matlab V7.14 (R2012a) with processor AMD Turion(tm) X2 Ultra Dual-Core Mobile ZM-85 2.30 GHZ with platform Vista.**

Keywords-----Elliptic Curve, ElGamal, Virtual Machine, Matlab, Piracy.

I. INTRODUCTION

During the last two decades, the industry in computer software has been increasing rapidly. Today the Internet has become a major distribution channel for digital information. This presents both opportunities and challenges to software vendors. As digital information can be copied and transmitted at great ease, methods of safeguarding the investment are therefore important. Illegal software copying and sharing cause often software companies revenue loss. For every legitimate copy of software that is sold, it is estimated that three or four illicit copies are made. This trend has driven software vendors to implement copy protection mechanisms to protect their applications and revenue by ensuring their applications are used legally [1].

A software product is expensive to produce but cheap to reproduce due to its digital nature. The cost of producing the first copy may be substantial, but the cost of producing additional is negligible. Digital technology poses two challenges for rights management. First, it reduces the cost of making copies. Second, it allows the copies to be distributed quickly, easily, and cheaply. Thus, we face threats against the software product. Software piracy is a major threat to the software production. Illegal downloading of software from the Internet is similar matter. These may cause severe economic harm and threaten software industries. Threats from viruses and product warranties still remains, but it is beyond the scope of this paper [1].

More than 20 years ago, data security was defined by Denning as the science and study of methods of protecting data in networked systems, and to include cryptographic controls, access controls, information flow controls, inference controls, and procedures for backup and recovery. Of these, cryptographic controls have received the greatest academic attention, with emphasis on typically mathematical data-manipulation algorithms involving secret keys – e.g. encryption algorithms for confidentiality, and message authentication codes (MACs) and digital signature algorithms for real-time authentication [2]**.**

Moreover, Software protection falls between the gaps of security, cryptography and engineering, among other disciplines. Despite its name, software protection involves many assumptions related to hardware and other environmental aspects. A significant gulf currently exists between theory and practice. Inconsistencies have arisen in the relatively sparse (but growing) open literature as a result of differences in objectives, definitions and viewpoints. All of these issues provide research opportunities [3]**.**

This Paper was organized as follows : Section II presents concepts of software piracy, followed by section III talking about related works classified as hardware and software protection, while section IV and V illustrate and analyzed previous works, with describing Elliptic curve and ElGamal Cryptographic algorithms, finally sections VI,VII, and VIII presents design and implementation of proposed system.

II. SOFTWARE PIRACIES

Piracy is the illegal use or distribution of property protected under intellectual property laws. Software piracy can divide into following categories [1]:

- End user piracy
- Client-Server Overuse
- Internet piracy
- Hard-disk loading
- Software counterfeiting
-

The end user is the ultimate user of a computer system or product. End user piracy occurs when an individual or organization reproduces or uses unauthorized copies of software. This includes using one licensed copy to install a program on multiple computers; or copying disks for installation and distribution; or acquiring academic restricted software without a license for commercial use.

There are thousands of pirate websites located on the Internet and virtually every software product now available on the market can be located on one of these sites. Therefore, Internet piracy represents perhaps the greatest threats. These pirate websites provide unauthorized copies of software for free download or upload. Internet auction sites also offer counterfeit, infringing copyright software.

Client-server overuse is that the number of users connected to and accessing one server exceeds the

maximum number allowed in the license agreement. Counterfeiting is the illegal duplication of software with the intent of directly imitating the copyrighted product. Hard-disk loading occurs when a computer hardware reseller loads unauthorized copies of software onto the machines it sells.

III. LEGAL PROTECTION APPROACH

In this section, legal protection approaches are described in terms of copyright, patent and license.

A. Copyright

Copyright is a nearly exclusive right of an author to control the distribution and reproduction of his/her original works. In general, a copyright law protects the form of expression of an idea, but not the idea itself. With respect to software, this means that both source code (human-readable form) and object code (machine-executable form), and the related manuals are eligible for copyright protection. But the methods and algorithms within a program are not protected.

Copyright law is a key legal protection mechanism. It can apply to virtually all computer software. The copying of copyrighted software without the permission of its owner may subject the copier to criminal penalties. This is very important for preventing theft and piracy of software products while encouraging developers to promote investments in new products and services.

The advantages of the copyright are the low cost, ease of obtain and speed of implementation.

B. Patent

A patent is a legal right provided by a government entity (e.g. the European Patent Office) that allows an inventor to prevent others from manufacturing, selling or using the patent owner's invention.

As expected, patent protection may become a valuable competitive tool as compared with traditional copyright protection. A focus of patent protection is to keep from selling similarity in an open market without permission. Inversely to copyright, a patent protects ideas and algorithms in a software product rather than the code itself. A typical example is the protection of functions, methods, system, and algorithms, as well as applied mathematical formulas.

C.License

Much of today's software is not purchased by user but licensed to users. In this case the license agreement will state what rights are granted to the licensee while all other rights remain with the copyright owner.

License is a binding agreement in which one party grants certain rights and privileges to another. In the computer field, a software owner will typically grant a nonexclusive right (license) to a user to use one copy of its software and prohibits further copying and distribution of that software to another user. A consistent and clearly formulated licensing scheme will always be beneficial in taking action against the illegal copier. It establishes the boundary between legal and illegal acts by the licensor with respect of his work.

IV. RELATED WORKS

The following is to focus on discussion of technical protection. It involves two major approaches, hardware-based protection and software-based protection.

A. Hardware-Based Protection

Hardware-Based Protection provides a variety of features. It is powerful, fast and autonomous. Table [1] lists some of the hardware options available:

Table 1: Hardware options

Company	Product
Marx Software Security	Crypto-Box, Smart X Card
Az-tech Software	Everkey
DESkey	Hardware options
Aladdin Knowledge System	HASP
Rainbow Technologies	Sentinel Hardware key

These solutions provide a variety of features. The basic features include authentication procedure, data encryption, access control, unique serial number, key generator, reliable communication, and device identification. These solutions mainly focus on the copy protection. Some also support licensing schemes.

The following are some examples of production.

1) A Dongle

A dongle is a hardware-based security device that attaches either to the serial or parallel printer port of a PC. It is a hardware key that uses codes and passwords embedded inside the key, which can control access to software applications. There is currently a wide range of commercially available protection devices. CRYPTO-BOX Hardware Keys developed by MARX Software Security is an example.

Protection is achieved by enabling software developers to include within a protected program a series of validation tests, queries or locks.. The dongle uses a unique algorithm, which is different for each model, to transform the character string into the numerical response, the result of which is passed back to the calling program for evaluation and validation. If the correct dongle is not detected, the program will not function at all.

2) Hidden Serial Numbers

By this protection mechanism, a pseudo-random serial number is synthesized and hidden on the PC when the software application is installed. The serial number is hidden in either an encrypted file or in a special system file. The user must perform a registration process to get the program functioning. During the registration process, software vendor verifies the serial number and supplies customer corresponding password. The serial number can be used to detect the location of software; prevent abusive copies of software.

B. Software-Based Protection

There are a number of software solutions available, and they vary significantly in features and approach. Table [2] lists some options.

Software-Based Protection is easy to implement. Sometimes it is relatively cheaper than hardware-based mechanisms. This software protection mechanism includes features as ease of use, easy maintenance, minimal size increment, redundancy of methods, robust exception handling, strong

encryption etc. The following are some examples of products.

Table 2: Software Options

Company	Product
GLOBEtrotter	FLEXlm
Marx Software Security	Protection Plus
Crpkey	Software Developers Kit(SDK)
AZ-tech	EverLock
Microcosm	Unlock-it
Rainbow Tech	SentineILM

1) FLEXlm

GLOBEtrotter Software and Highland Software developed FLEXlm. It is a widely used license management technology. FLEXlm allows software from several vendors to be supported with a single license management system on a network. It also supports variety of licensing police such as:

- Node locked - software can only run on a particular machine.
- User based - software can only be run by particular user-ID.
- Site licensing - all users at a particular site may run the software Floating license - users anywhere on the network may run the software up to the licensed number of copies The FLEXlm includes four components, which are license manager daemon, vendor daemon, license file, and application program. Disadvantages it is distributed as a shared library; It is relatively easy to decipher the communication between the application program and the license module and then replace the license module with a set of functions that always return a "grant"; The arguments to the license module functions need to be encrypted; Weakness in node locking, kernel support needed.

2) The Protection PLUS

The Protection PLUS system is produced by the MARX Software Security. It is a software licensing toolkit that insures proprietary security and control. The system consists of two parts. One is the License File Editing facility, which creates encrypted, binary License Files and generates Trigger Codes. The other is the language-specific library, which contains functions for implementing the software licensing features.

License Files allow you to store information to control the execution flow of your application. This can be either completed before sending the application or manipulated by the application remotely using Trigger Codes. License Files may be stored in a file or a CRYPTO-BOX hardware key..

PLUS supports following features:

Remotely unlock and extend demo versions Convert illegal copies into demos using software-based protection, Protect network-based applications using fixed or floating licensing schemes (Remotely control the number of allowed users).

Protection is achieved by allowing authorizing a particular computer using a unique ID combination with an encryption algorithm provided in the PLUS library. The system detects illegal copies of the software application. Possible disadvantages Supports only windows system; May not work with specific environment; Robustness may depends on encryption algorithm.

3) IP-Safe

IP-Safe is a set of software libraries and tools developed by Power Technology. With it, software providers can protect their products by mean of license. It can work directly with licensees to customize IP-Safe for particular product and marketing needs.

The protection is achieved by using IP-Safe application to internally generate a unique Machine ID number for each PC. This number is based on physical properties of the CPU and motherboard, and is unaffected by operating system upgrades and installs, disc replacements and system utilities.

Possible disadvantage robustness of the protection mechanisms may depend.

V. ANALYZING PREVIOUS METHODS

The previous methods of software protection are perfect. But methods are only makes breaking the mechanism harder, but not impossible. When it is very important to have a desirable software protection system. This system should effectively prevent the unauthorized access of software programs while allow the authorized use of these programs. Our desired system should have the following features: Inexpensive, Ease of use, Compatible well with existing unprotected programs and with other protection systems Not affected by system utilities, OS upgrades Easy for software vendors to incorporate the system into their software distribution Robust and should not be denied by the license Consideration of management system breaking With prevention, detection and response mechanism

As prevention mechanisms are never perfect. Most software products have security bugs, and users make mistakes. Without detection and response, the prevention mechanisms only have limited value. Detection and response are more cost effective.[software product protection]

VI. Software Protection with Elliptic Curve

Before going any further, we distinguish between two "folklore" notions: the problem of protection against illegitimate duplication and the problem of protection against redistribution (or fingerprinting software). Loosely speaking, the first problem consists of ensuring that there is no efficient method for creating executable copies of the software; while the second problem consists of ensuring that only the software producer can prove in court that he has designed the program. In this paper we concentrate on the first problem [4].

There are three participants in the scheme: buyer, merchant, and key information center *(KIC).* We assume that there is a trusted *KIC* on the Internet who is responsible for managing and issuing smart cards to users. The users must apply for the smart cards with their real identifications. There are three phases in the proposed scheme: registration phase, purchasing phase, and installation phase **[5].**

Many software protection techniques and approaches exist. Some can be classified into major groups: software obfuscation, software tamper resistance, diversity, marking schemes (e.g. watermarking), node-locking schemes, time-limiting schemes, etc.

A. Elliptic Curve

Elliptic curves were first suggested in 1985 by N. Koblitz and V. Miller for implementing public key cryptosystems. They have recently been utilized in designing algorithms for primarily testing and also integer factorization. The main feature of ECC is that it relies on the difficulty of solving ECDLP (Elliptic Curve Discrete Log Problem) in the same way as RSA depends on the difficulty of factoring the product of two large primes. The best known method for solving ECDLP is fully exponential, whereas the number field sieve (for factoring) is sub-exponential. This allows ECC to use drastically smaller keys to provide equivalent security. Moreover, due of their rich structure one has more flexibility in choosing an elliptic curve than choosing a finite field [6]**.**

B. ECC Description

First of all this paper describe the operation geometrically. So consider the two distinct points A and B on an elliptic curve as shown in Figure [1]. Then, to add A and B we draw a line through these points. If this line is not parallel to Y-axis, as the equation of elliptic curve is cubic, this line intersects the curve at exactly one more point. Then C = A + B is defined as the reflection of this point in X-axis. If the line is parallel to Y-axis we define C = O (point at infinity). Hence, if A is any point on elliptic curve then to double the point to get C = 2A, we draw a tangent line at A, so that if this line is not parallel to Y-axis, it will intersect the curve at exactly one more point, then C = 2A is defined as the reflection of this point on X-axis and if the line is parallel to Y-axis then we define C = O (point at infinity). We may sum up the mechanics of addition of two points on elliptic curve with following set of rules:

Point addition: In order to find the sum of two P and Q on elliptic curve E, we draw a line connecting P and Q. This line will intersect E at exactly other point, which we will denote P * Q. P + Q will be defined as the reflection of P * Q across the x-axis.

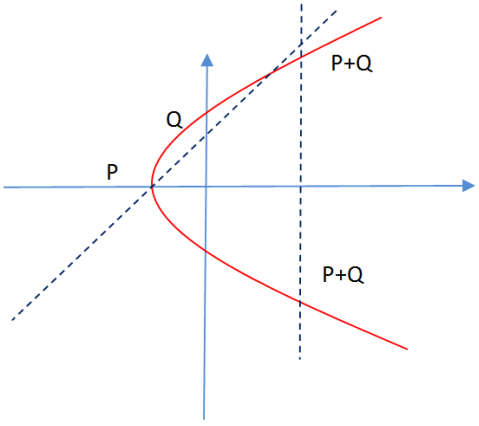


Figure [1] Point addition

Point doubling: When P and Q are the same point, we draw the tangent line to E at P and find the second point where this line intersects E. We call this point P * P. Again, we reflect this point over the x-axis to obtain P + P [8].

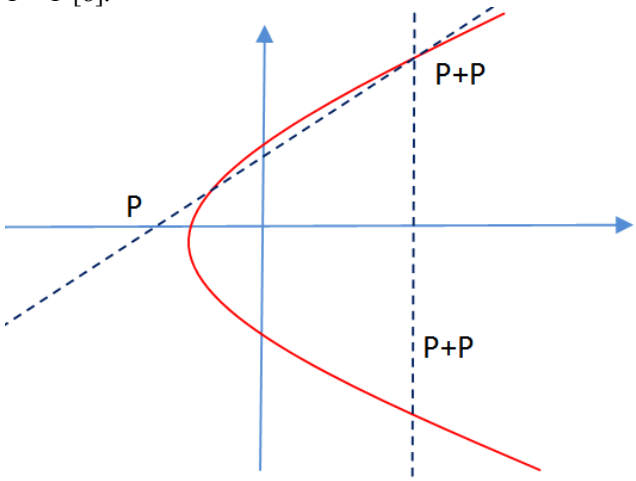


Figure [2] Point doubling

Most of the products and standards that use public key cryptography for encryption and digital signatures use RSA. The key length for secure RSA use has increased over recent years, and this has put a heavier processing load and applications using RSA **[9].**

In general equations for elliptic curves take the form of weierstrass equation:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \text{ --- (1)}$$

So to plot such curve we need to compute:

$$y = \sqrt{x^3 + ax + b} \text{ -------- (2)}$$

For a given values of a and b, the plot consists of positive and negative values of y for each values of x. Thus each curve is asymmetric y= 0.

C. Key generation

Taking in consideration that an elliptic curve can be defined by **[10]**:

$$y^2 + Axy + By \text{ ---- (3)}$$

$p_1+p_2 = p_3 = (x_3, y3)$ as follows:

1- If $x_1 \neq x_2$ then $x_3 = m_2-x_1-x_2$, $y_3=m(x_1-x_3)-y_1$, where $m = \frac{y2-y1}{x2-x1}$

2- If $x_1=x_2$ but $y_1 \neq y_2$ then $p_1+p_2= \infty$

3- If $p_1=p_2$ and $y_1 \neq 0$ then $x_3=m_2-2x_1, y_3=m(x_1-x_3)-y_1$ where $m = \frac{3x2+A}{2y1}$

4- If $p_1=p_2$ and $y_1=0$, then $p_1+p_2= \infty$.

Example:

For $E_{11}(1,6)$ = {(2,4),(2,7),(3,5),(3,6),(5,2),(5,9),(7,2),(7,9),(8,3),(8,8),(10,2),(10,9)}.

- Different examples in this proposed research applied and demonstrated with various prime numbers. The results constructed and checked [12].

Base Points

In EC all Points that are responsible for generating all coordinates in EC called Base points. Source code for this task will be found in the last section of this paper.

Moreover all base points must forming Abelian group. Therefore not all curve equations can be elliptic curve which can be used in cryptography. **[11]**

For equation (3):

$$\{y^2 \bmod 11 = x^3 + ax + b \ \text{Mod } 11\}$$

Where a=b=1

G={(0,1),(0,10),(1,5),(1,6),(2,0),(3,3),(3,8),(4,5),(6,5),(6,6),(8,2),(8,0)}.

Base ={(1,5),(1,6),(4,5),(4,6),(8,2),(8,9)}

Case Base {1, 5}:

G = {(1,5),(3,3),(8,2),(6,5),(4,6),(0,10),(2,0),(0,1),(4,5),(6,6),(8,9),(3,8),(1,6)}

So G = {P,2P,3P,4P,5P,6P,7P,8P,9P,10P,11P,12P,13P}

D. ECC KEY EXCHANGE

- Global Public Elements

$E_q(a,b)$ elliptic curve with parameters a,b & q in the equation $y^2 \bmod q = (x^2+ax+b) \bmod q$**[7]**

- Q Base point on elliptic curve

User A Key Generation

- Select private key k_A $k_A < n$
- Calculate public P $P = k_A \times Q$

User B Key Generation

- Select private key k_B $k_B < n$
- Calculate public M $M = k_B \times Q$
  - Generation of Secret Key by user A $P_1 = K = k_A \times M$
  - Generation of Secret Key by user B $P_2 = K = k_B \times P$

The two calculations produce the same result because $k_A \times M = k_A \times (k_B \times Q) = k_B \times (k_A \times Q) = k_B \times P$

VI. ELGAMAL CRYPTOGRAPHY

This algorithm allows two people to communicate messages secretly over an insecure communications channel.

1. (Setup) A finite cyclic group G of order n and generator $\alpha \in G$ are chosen.

Each user picks a random integer $\mathbf{l} \in \{0, 1, \ldots, n-1\}$(the private key), and makes public $\alpha^{\mathbf{l}}$ (the public key).

Messages are elements of G and that user A wishes to send a message, m, to user B.

2. A generates a random integer $k \in \{0, 1, \ldots, n-1\}$ and computes α^k.

3. A looks up B's public key $\alpha^{\mathbf{l}}$ and computes$(\mathbf{l})^k$ then $m\alpha^{\mathbf{l}k}$

4. A sends to B the pair of group elements$(\alpha^k, m^{\mathbf{l}k})$.

5. B computes $(m^{\mathbf{l}k})((\alpha^k)^{\mathbf{l}})^{-1} = m^{\mathbf{l}k}(\alpha^{\mathbf{l}k})^{-1} = m$ and recovers the message [9].

VII. DESIGN AND IMPLEMENTATION

1- Install Virtual Machine Program (VMware).

2- Setting for the newly software which can be considered as protected program, as shown in figure [1], so in this step open O.S in another O.S .

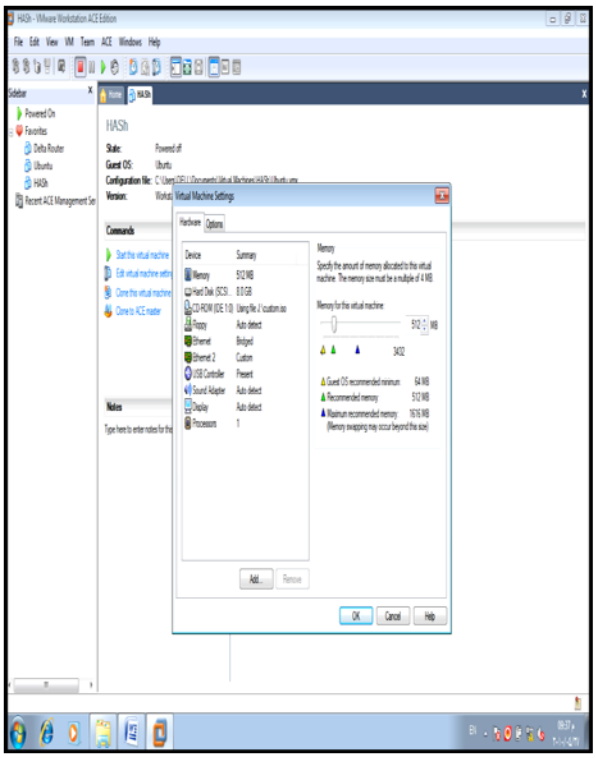


Figure [3] newly software setting

2- For external connection and before installing the target software install another LAN Card. And select appropriate configuration for the LAN Card.

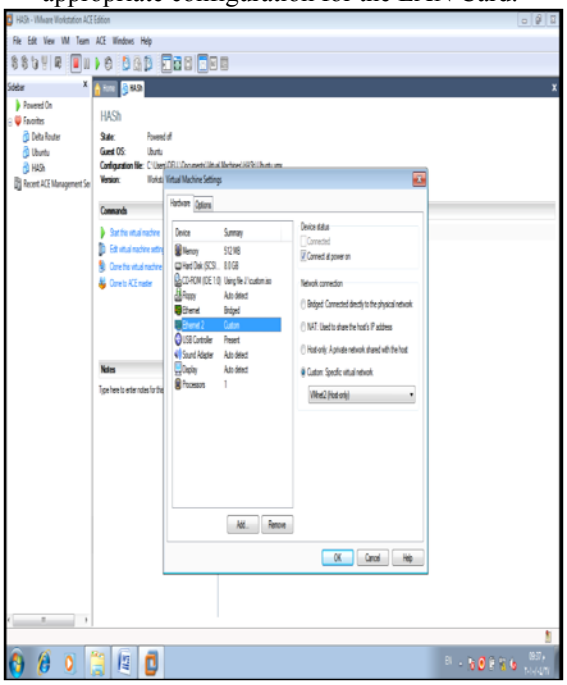


Figure [4] (LAN Card configuration)

3- Select an appropriate configuration for the virtual machine her typical one}.

4- Select a guest operating system as a platform for the target {choose her a Linux-ubuntu}.

5- Select an appropriate location for the target software.

Select type of network connection to the Internet **{choose a bridge connection}**

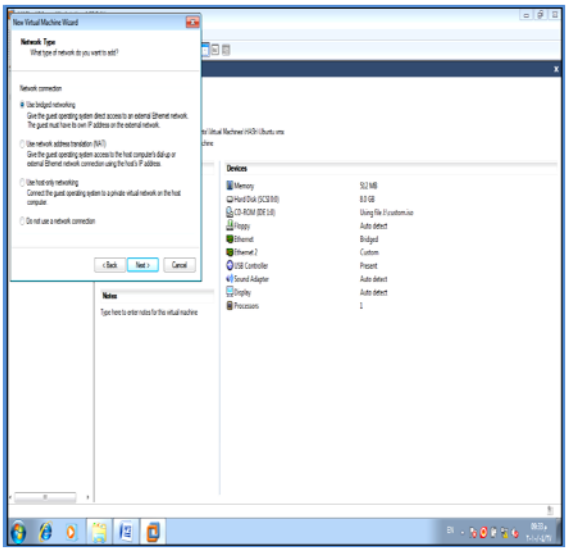


Figure [5] (Bridge connection)

6- Specify the size of the software.
7- Start the software {ECC}.
8- Define IP of the network and Gateway and DNS.

After getting IP and processor ID which are in this proposed project represent a data of a message in corresponding ECC cryptography algorithm the next step will be generating user key using elliptic curve with ElGamal exchanging key . The overall proposed technique can be presented in the figure [6] as follows:

VIII. Discussion Of Results

One of the data which is exploited and examined to get asymmetric points of the elective elliptic curve is :

$y^2 = x^3 + 2x + 3 \mod p$ ---- (4) , where p = 5, and a=b=1, then the extracted points are :
(1,1),(1,4),(2,0),(3,1),(3,4) and (4,0).

Figure (6) below represent the intersected points of the continuous elliptic curve.

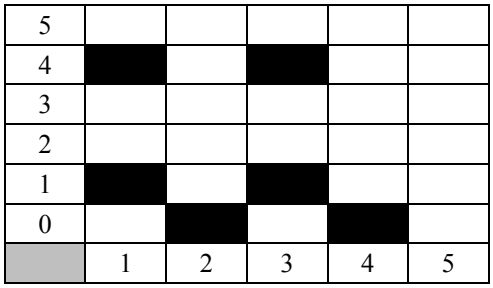


Figure (6) Elliptic Curve $E_5(1,1)$

In general cubic equations for elliptic curves take the form of weierstrass equation: **[9]**.
$y^2 + axy + by = x^3 + cx^2 + dx + e$ - (5)
In this project, Elliptic curve for polynomial

$$y^2 mod\ 17 = x^3 + x + 1\ mod\ 17$$

Then creation of points in this elliptic curve represented in this project as shown in Figure [7]:

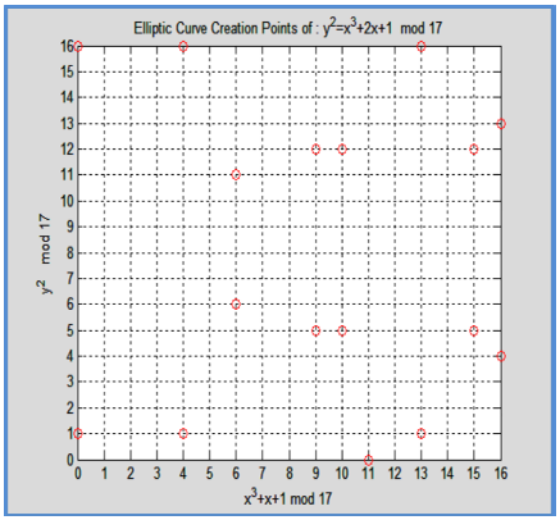


Figure [7]: Tested Elliptic Curve for Polynomial $y^2 mod\ 17 = x^3 + x + 1\ mod\ 17$

So, elliptic curve points are:

x	0	0	4	4	6	6	9	9	10
y	1	16	1	16	6	11	5	12	5
x	10	11	13	13	15	15	15	16	16
y	12	0	1	16	5	12	4	13	6

Source code for generating Base points will be presented in section 10. These base points will be used to generate public keys for both provider and user.
Then the principle attraction of ECC, compared to RSA, is that appears to offer equal security for a far smaller key size, thereby reducing processing overhead. On the other hand, although the theory of ECC has been around for some time, it is only recently that products have begun to appear and that there has been sustained cryptanalytic interest in probing for weakness.

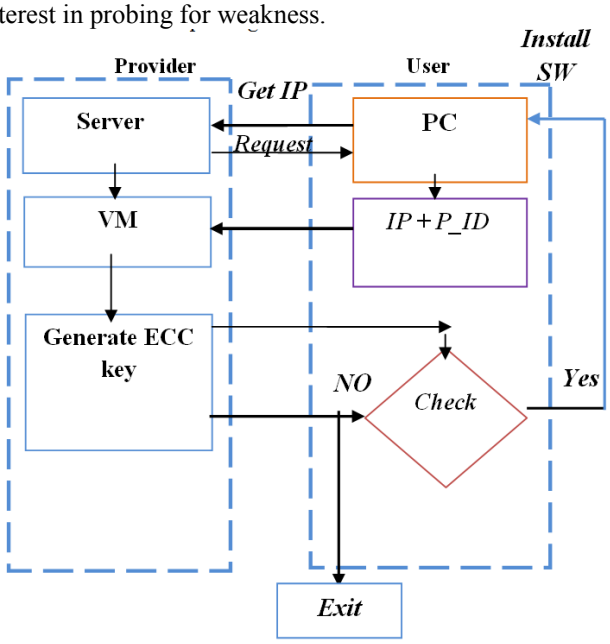


Figure [8] (system Block Diagram)

9- Register the product key after setting the IP through the Installation.
10- Analysis of the software.
11- Checking it.

12- Registering product of the software.

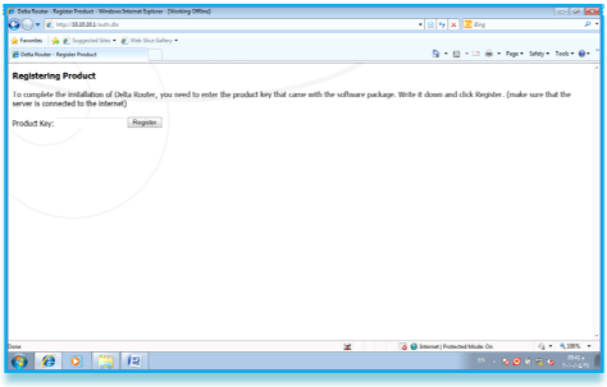


Figure [9] (Registering Software product key)

13- If we are already registering, then we must enter the password and username.

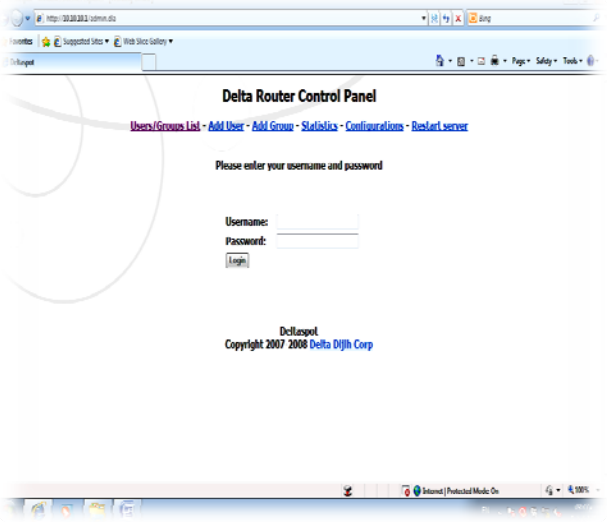


Figure [10] (User Name and Product key checking)

Next generating elliptic curve points, choosing base points, and exchanging key between the provider and user will be presented in the next figures.

IP address and Processor ID are message data.

Time Response for Encryption and Decryption

Time consuming for encryption and decryption using ElGamal algorithm with Elliptic curve can be summarized by the following table:

Table [3]

ElGamal with Elliptic curve (Time consuming)

ElGamal With ECC(Size in bits)	Time of Encryption and Decryption
128	$4.03X10^{-2}$
264	$9.01X10^{-3}$
528	$52X10^{-3}$
729	$61X10^{-3}$

VIII. Conclusions and Future Work

As analyzed above, no methods of software protection are perfect. Any method is only makes breaking the mechanism harder, but not impossible. When managing intellectual property, we should choose the terms and conditions that maximize the value of the intellectual property, not the terms and conditions that maximize the protection. It is very important to have a desirable software protection system, with the following features:

1. This system should effectively prevent the unauthorized access of software programs while allow the authorized use of these programs.
2. Our desired system should have the following features: Inexpensive, Ease of use, Compatible well with existing unprotected programs and with other protection systems, Not As analyzed above, no methods of software protection are perfect. Any method is only makes breaking the mechanism harder, but not impossible.
3. Not affected by system utilities, OS upgrades, Easy for software vendors to incorporate the system into their software distribution.

   To gate protected system from piracy with some or all above useful characteristics, the proposed system tested via virtual machine, choosing asymmetric cipher system using elliptic curve cryptographic algorithm.
4. One of the most powerful technique in this project, that current software poorly interact with internet because remote server are asynchronous, so performance is independent of large network latencies.

The tested results approved that the proposed system consume a little time, ease of use and high degree of security.

In the future work, the proposed project can introduce a function responsible for generating prime numbers which can be added to m.function (points) to calculate all the Elliptic curve points.

Also in the future work can introduce effective function responsible on checking the abelian feature of the function.

IX. SOURCE CODE

This section presents some Matlab source code starting from function for conversion ID of a processor and IP of a user as a data message, a function performing generating Elliptic curve points, addition and doubling previous points, and checking and inversion of points values.

```
function z = convt(x,b);
% This function performs the conversion of ID to text encryption
%  z = x + b mod 26
% assume that x is a text string and b is a number
% The result is kept in text representation
xnum=text2int(x);
ynum = mod(xnum + b, 26);
z=int2text(ynum);
```

(Program to find all ECC Points)

```
a=0:22; %all points of finite field
left_side = mod(a.^2,23);
right_side = mod(a.^3+a+1,23);
points = [];
 for i = 1:length(right_side)
```

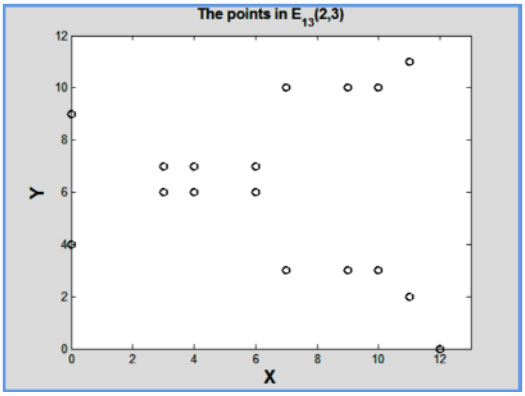
```
    I = find(left_side == right_side(i));
    for j=1:length(I)
       points = [points;a(i),a(I(j))];
    end
  end
  plot(points(:,1),points(:,2),'bo')
  set(gca,'XTick',0:1:22);
  set(gca,'YTick',0:1:22);
  grid on;
  function Q = EllipticCurveFastScalMult(P, x, a, b, p)
  % Q = EllipticCurveFastScalMult(P, x, a, b, p)
  % Inputs:  Two integers a, b, and a prime p > 3.  It is
  assumed that 4a^3 +% 27b^2 is nonzero, so that the
  associated elliptic curve E:
  %        y^2 = x^3 + ax + b
   mult = x;
  Q = [inf inf];
  D = P;
  while mult > 0
     if  mod(mult,2) == 1
  Q = EllipticCurvePointAdditionModp(Q, D, a, b, p);
     end
     mult = floor(mult/2);
     D = EllipticCurvePointAdditionModp(D, D, a, b, p);
  end
        Q = EllipticCurvePointAdditionModp(Q, D, a, b, p);
     end
     mult = floor(mult/2);
     D = EllipticCurvePointAdditionModp(D, D, a, b, p);
  end
  ----------------------------------------------------------
  Function Points = EllipticCurvePointsModp3Mod4(a, b,
  p)
  % Points = EllipticCurvePointsModp3Mod4(a, b, p)
  % Input:  Two integers a, b, and a prime p > 3.  It is
  assumed that 4a^3 + % 27b^2 is nonzero, so that the
  associated elliptic curve
  %        y^2 = x^3 + ax + b
  %is nonsingular.
  % Output:  Points, a 2 column vector listing all points
  belonging to this
  % elliptic curve mod p, including the point at infinity.
  Each row of
  % points will be a point on this elliptic curve.
  % compute the right side r = x^3 + ax + b, and check to
  see whether
  % r^(p+1)/4 (mod p) is a square root of r.
  PointIndex = 1;
  for x = 0:p-1
     rightside = mod(x^3 + a*x + b, p);
     rpower = FastExp(rightside,(p+1)/4,p);
     if mod(rpower^2,p)==rightside %+/- rpower are square
  roots:
        if rightside == 0  %only one square root
   Points(PointIndex, :) = [x 0];  PointIndex = PointIndex +
  1;
        else
           Points(PointIndex, :) = [x rpower];   PointIndex =
  PointIndex + 1;
           Points(PointIndex, :) = [x p-rpower];   PointIndex
  = PointIndex + 1;
        end
     end
  end
  Points(PointIndex, :) = [Inf Inf];
```

Addition

```
function[x3,y3,m]= addition10(x1,y1,x2,y2,A)
% This function m-file performs the Elliptic Curve
addition on read numbers
% the elliptic curve y^2 = x^3 + Ax + B
% Define P1 = (x1,y1)
% P2 = (x2,y2)
% Then P1 + P2 = P3 = (x3,y3)
if x1=='infinity'
x3=x2; y3=y2;
return
end
if x2=='infinity'
x3=x1; y3=y1;
return
end
if x1==x2
if y1==y2
if y1==0
display('P3 is infinity')
x3='infinty'; y3='infinity';
return
end
m = sym( (3*(x1)^2 + A)/(2*(y1)) );
x3 = sym( m^2 - x1 - x2);
y3 = sym( m*(x1 - x3) - y1 );
return
end
display('P3 is infinity')
x3='infinty'; y3='infinity';
return
end
m = sym( (y2-y1)/(x2-x1) );
x3 = sym( m^2 - x1 - x2 );
y3 = sym( m*(x1 - x3) - y1 );
```

```
function [X,Y,n] = FPLOT(A,B,p)
% This function m-file finds and plots all the points that
lie in on the curve y^2 = x^3 + AX + b (mod p)
RHS = zeros(3,1);
LHS = zeros(3,1);
X = zeros(2,1);
Y = zeros(2,1);
for i=0:1:(p-1)
RHS(i+1) = (i)^3 + A*(i) + B;
RHS(i+1) = mod(RHS(i+1),p);
LHS(i+1) = (i)^2;
LHS(i+1) = mod(LHS(i+1),p);
end
ii=1;
for z=0:1:(p-1)
I=find(RHS==z);
J=find(LHS==z);
q1 = isempty(I);
q2 = isempty(J);
if (q1) == 0
```

```
if q2 == 0
n=length(I);
m=length(J);
for h=1:1:n
for g=1:m
X(ii)=I(h)-1;
Y(ii)=J(g)-1;
ii=ii+1;
end end end end end
n=length(X) + 1;
h=plot(X,Y,'ko');
set(h(1),'LineWidth',1.5)
axis([0, (max(X)+1), 0,(max(Y)+1) ])
xlabel('X','FontSize',15,'FontWeight','bold')
ylabel('Y','FontSize',15,'FontWeight','bold')
title(['The points in E_{',int2str(p),'}(',int2str(A),',',int2str(B),')'],'FontSize',12,
'FontWeight','bold');
```



---

function[x3,y3,m]= ADDP(x1,y1,x2,y2,~,p)

```
% Elliptic Curve addition over prime curves.
%  the elliptic curve y^2 = x^3 + Ax + B
% Define P1 = (x1,y1)
% P2 = (x2,y2)
% Then P1 + P2 = P3 = (x3,y3) is defined by as below
% If one if the variables in infinity then we define P + infinity = P
if x1=='infinty'
x3=x2; y3=y2;
return
end
if x2=='infinty'
x3=x1; y3=y1;
return
end
if x1==x2
if y1==y2
if y1==0
display('P3 is infinity')
x3='infinty'; y3='infinity';
return
end
%m = sym( (3*(x1)^2 + A)/(2*(y1))
mnum = 3*(x1)^2 + A;
mden = 2*(y1);
xx = mden\p ;
m = mod((mnum * xx),p);  % x3 = sym( m^2 - x1 - x2)
x3 = mod( (m^2 - x1 - x2) , p);
% y3 = sym( m*(x1 - x3) - y1 );
y3 = mod( (m*(x1 - x3) - y1) , p);
return
end
display('P3 is infinity')
x3='infinty'; y3='infinity';
return
end
% m = sym( (y2-y1)/(x2-x1) );
mnum = y2 - y1;
mden = x2 - x1;
xx =mden\p ;
m = mod((mnum * xx) , p);
% x3 = sym( m^2 - x1 - x2 );
x3 = mod( (m^2 - x1 - x2) , p);
% y3 = sym( m*(x1 - x3) - y1 );
y3 = mod( (m*(x1 - x3) - y1) , p);
```

---

function [I] = inverse(N,p)

```
% This m-file finds the inverse of an element, N, in the group Z_p
N = mod(N,p);
H = zeros(3,1);
for i = 1:(p-1)
H(i) = mod(N*i,p);
end
I = find(H==1);
```

---

function[X2,Y2]= Doubling(X1,Y1,k,A,p)

```
%  successive doubling algorithm
% on prime curves. If P = (X1,Y1) and k is an integer, then this find kP = (X2,Y2)  the elliptic % y^2 = x^3 + Ax + B (mod p),p prime
a = k;
BX = 'infinity';
BY = 'infinity';
CX = X1;
CY = Y1;
while a~=0
gg = mod(a,2);
if gg == 0
a = a/2;
[CX,CY] = ADDP(CX,CY,CX,CY,A,p);
end
if gg == 1
a = a-1;
[BX,BY] = ADDP(BX,BY,CX,CY,A,p);
CX = CX; CY = CY;
end
end
X2 = BX;
Y2 = BY;
```

---

function [flag] = checkl(x,y,A,B,p)

```
% An m-file to check if the point (x,y) lies on the prime curve
% y^2 = x^3 + Ax + B (mod p)
R = x^3 + A*x + B;
R = mod(R,p);
L = y^2;
L = mod(L,p);
```

```
if L == R
flag = 'YES';
display('This point lies on the curve')
else
flag = 'NO';
display('This point does not lie on the curve') end
```

Applied ElGamal Algorithm

Let prime = 65001701

Create Base curve and point Public curve Form: :

H8T9QHMGZZCVY7MDA6FZ5RNZDZ

Base point

X: 13f2 9774d24 f3814df5 5d0be8164

Y: 33d6 fbf7a211 37c3941b 71891a62

Create side 2's private key

Side 2 secret:

QQ43QEUFQPZCHFZFG6HK6WU5DGUQ

Generate side 2's public key

Side 2 public key

X: HB3RWK-ZUXYRS-T6MH8E-XSGMLR

Y: QEDHL5-DT643H-7EKP2E-MJDZ9Q-LW3A

Create message data

send Data from side 1 to side 2

Hidden data

X: XUQULP-F3C6VW-DUKYLG-RQQD7

Y: DBYMEV-Z6WEBP-SANRUX-UHRR

Random point

X: ZJ9DQW-EWNA6M-LSN95B-G4CEUC

Y: QUB6DL-LSBFHS-XK38L3-X8QCKTG

Recover transmitted message

Sent data

MJV2MK-U6FEKV-5DU3NC-GFC2ED

Received data

MJV2MK-U6FEKV-5DU3NC-GFC2ED

REFERENCES

[1]. Chen, Min, "Software Product Protection", Helsinky University of technology, Finland Publication Telecommunication Software and Multimedia, 2001.

[2]. Main, A, Oorschol,Van," Software Protection and Application Security : Understanding the Battleground", Carleton University, Ottawa, Canada,2003.

[3]. Oorschol,Van," Software Protection", Carlton University, Canada, Springer, P 1-13, 2003.

[4]. Ostrovs, Rafail," Software Protection and Simulation on Oblivious RAMs", MIT Ph.D Thesis, 1992.

[5]. Lin, Chu, Lee, Yu." One time installation with traitors Tracing for copyright programs, Tunghai, Univesity, Taichung city, Taiwan Rio, IEEE 2001.

[6]. Bhardwaj, Kuldeep,Chaudhary Sanjay." Implementation of Elliptic Curve in C ",Ambedkar University, Septemper 2012, International Journal on Emerging Technologies.

[7]. Kolnekar, Megha, Jadhav, Anita," Implementation of Elliptic Curve cryptography on Text and Image", International Journal of Enterprise computing and Business Systems, Vol. 1 Issue 2 July 2011.

[8]. Aglawe, Dipti, Gajbhiye Engineering Samta," An Implementation approach of Ecdlp Diffe Helman using Vb. Net", International of computational Research, Vol. 2, Issue October 2012.

[9]. Stalling, William, "Cryptography and Network Security", Fifth Edition, Pearrson Education,Inc., Publishing as perntic,Hall,2011.

[10]. England, Matthew," Elliptic Curve Cryptography", M.Sc Applied Mathematical Science,Heriot-Watt University, Summer 2006.

[11]. Ghamgosar, Mohammad, Mahdavi, Mehregan," Application of Elliptic Curves in Wireless Communications Security", University of Guilan", Journal of Applied Mathematics, Islamic Azad university of Lahijan Vol.3, NO.11, Winter2006.

[12]. England, Matthew," Elliptic Curve Cryptography", MSc Applied Mathematical Science Heriot-Watt University summer 2006.