ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (9-16), Month: October - December 2014, Available at: www.researchpublish.com

A Hybrid Security Protocol Using Python

Devi.V.Kumar

University of Calicut, NSS College of Engineering, Kerala, India

Abstract: **A Computer Network is an interconnected group of autonomous computing nodes, which use a well defined, mutually agreed set of rules and conventions known as protocols, interact with one-another meaningfully and allow resource sharing preferably in a predictable and controllable manner. Communication has a major impact on today's business. It is desired to communicate data with high security. Security Attacks compromises the security and hence various Symmetric and Asymmetric cryptographic algorithms have been proposed to achieve the security services such as Authentication, Confidentiality, Integrity, Non-Repudiation and Availability. At present, various types of cryptographic algorithms provide high security to information on controlled networks. These algorithms are required to provide data security and users authenticity. To improve the strength of these security algorithms, a new security protocol for on line transaction can be designed using combination of both symmetric and asymmetric cryptographic techniques. This protocol provides three cryptographic primitives such as integrity, confidentiality and authentication. These three primitives can be achieved with the help of Elliptic Curve Cryptography, Dual-RSA algorithm and Message Digest MD5. That is it uses Elliptic Curve Cryptography for encryption, Dual-RSA algorithm for authentication and MD-5 for integrity. This new security protocol has been designed for better security with integrity using a combination of both symmetric and asymmetric cryptographic techniques.**

Keywords: **Network Security, Elliptic Curve Cryptography, Dual-RSA, Message Digest-5.**

I. INTRODUCTION

Hybrid security protocol blends the conveniences of both symmetric and asymmetric cryptographic algorithms. In this protocol the Symmetric Key Cryptographic Technique MD5 is used to achieve both the Confidentiality and Integrity. The Asymmetric Key Cryptography technique, such as ECC and RSA are used for Authentication. These algorithms were carefully analyzed and designed. The language adopted here is python, because it provides so many built in packages related to cryptography, so that the coding become easier. Also the installation of packages like python-dev, python-pip, libgcrypt and secure is required, where Python-dev consists of header files, a static library and development tools for building Python modules, extending the Python interpreter or embedding Python in applications, Pip is a package management system used to install and manage software packages written in the programming language Python, and secure includes tools for using algorithms based on elliptic curve cryptography .For cross-platform applications, we would have to create GUIs.. WxWidgets is a widget toolkit and tools library for creating graphical user interfaces (GUIs). WxWidgets enables a program's GUI code to compile and run on several computer platforms with minimal or no code changes. It covers systems such as Microsoft Windows, Linux/Unix etc

II. ALGORITHMS OVERVIEW AND SYSTEM PREREQUISITE

A. Elliptic Curve Algorithm

When using elliptic curves in cryptography, we use various properties of the points on the curve, and functions on them as well. Thus, one common task to complete when using elliptic curves as an encryption tool is to find a way to turn information *m* into a point P on a curve E. We assume the information *m* is already written as a number. There are many ways to do this, as simple as setting the letters a = 0, b = 1, c = 2, or there are other methods, such as ASCII, which

accomplish the same task. Now, if we have E: y2 = x3 + Ax + B (mod p), a curve in Weierstrass form, we want to let m = x. But, this will only work if m3 + Am + B is a square modulo p. Since only half of the numbers modulo p are squares, we only have about a 50% chance of this occurring. Thus, we will try to embed the information m into a value that is a square.

Pick some K such that 1/2K is an acceptable failure rate for embedding the information into a point on the curve. Also, make sure that (m + 1) K < p. Let xj = mK + j for j = 0, 1, 2. . . K − 1 Compute x3j + Axj + B. Calculate its square root yj (mod p), if possible. If there is a square root, we let our point on E representing m be Pm = (xj, yj) If there is no square root, try the next value of j so, for each value of j we have a probability of about 1/2 that xj is a square modulo p. Thus, the probability that no xj is a square is about 1/2K, which was the acceptable failure rate [6]. In most common applications, there are many real-life problems that may occur to damage an attempt at sending a message, like computer or electricity failure. Since people accept a certain 16 amount of failure due to uncontrollable phenomenon, it makes sense that they could agree on an acceptable rate of failure for a controllable feature of the process. Though we will not use this specific process in our algorithms

B. RSA

In practice, the RSA decryption computations are performed in p and q and then combined via the Chinese Remainder Theorem (CRT) to obtain the desired solution in Z N, instead of directly computing the exponentiation in Z N. This decreases the computational costs of decryption in two ways. First, computations in Z p and Z q are more efficient than the same computations in Z N since the elements are much smaller. Second, from Lagrange's Theorem, we can replace the private exponent d with dp = d mod (p - 1) for the computation in Z p and with dq = dmod (q - 1) for the computation in Z p, which reduce the cost for each exponentiation when d is larger than the primes. It is common to refer to dp and dq as the CRT-exponents. The first method to use the CRT for decryption was proposed by Quisquater and Couvreur .Since the method requires knowledge of p and q, the key generation algorithm needs to be modified to output the private key (d, p, q) instead of (d,N). Given the private key (d, p, and q) and a valid cipher text C element of Z N, the CRTdecryption algorithm is as follows:

1) Compute Cp = Cdp mod p.

2) Compute Cq = Cdq mod q.

3) Compute M0 = (Cq - Cp). p-1 mod q.

4) Compute the plaintext M = Cp + M0. p.

This version of CRT-decryption is simply Garner's Algorithm for the Chinese Remainder Theorem applied to RSA. If the key generation algorithm is further modified to output the private key (dp, dq, p, q, p-1 mod q), the computational cost of CRT-decryption is dominated by the modular exponentiations in steps 1) and 2) of the algorithm. When the primes p and q are roughly the same size (i.e., half the size of the modulus), the computational cost for decryption using CRT-decryption (without parallelism) is theoretically 1/4 the cost for decryption using the original method. Using RSA-Small-e along with CRT-decryption allows for extremely fast encryption and decryption that is at most four times faster than standard RSA.

C. MD5

MD5 consists of 64 of these operations, grouped in four rounds of 16 operations. F is a nonlinear function; one function is used in each round. Mi denotes a 32-bit block of the message input, and Ki denotes a 32-bit constant, different for each operation. s is a shift value, which also varies for each operation. MD5 processes a variable length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks; the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits less than a multiple of 512. The remaining bits are filled up with a 64-bit integer representing the length of the original message The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted A, B, C and D. These are initialized to certain fixed constants. The main algorithm then operates on each 512-bit message block in turn, each block modifying the state. The processing of a message block consists of four similar stages, termed rounds; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation. Many message digest functions

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (9-16), Month: October - December 2014, Available at: www.researchpublish.com

have been proposed and are in use today. Here are just a few like HMAC, MD2, MD4, MD5, SHA, SHA-1. Here, we concentrate on MD5, one of the widely used digest functions.

D. Hybrid Security Protocol

A hybrid security protocol is one that blends the convenience of asymmetric cryptographic algorithms with the effectiveness of symmetric cryptographic algorithms. Hybrid security protocol that merges two or more cryptographic alorithms.It incorporates a combination of asymmetric and symmetric cryptographic algorithms to benefit from the strengths of each form of security protocols. These strengths are respectively defined as speed and security. The combination of security protocols methods has various advantages. One is that a connection channel is established between two users' sets of equipment. Users then have the ability to communicate through security protocols. Asymmetric protocols can slow down the encryption and decryption process, but with the simultaneous use of symmetric protocols, both forms of encryption and decryption are enhanced. The result is the added security of the transmittal process along with overall improved system performance.

III. HYBRID SECURITY PROTOCOL ALGORITHIM ARCHITECTURE

As Encryption became a vital tool for preventing the threats to data sharing and tool to preserve the data integrity this project focusing on security enhancing by enhancing the level of encryption in network. To improve the strength of these security algorithms, a new security protocol for on line transaction can be designed using combination of both symmetric and asymmetric cryptographic techniques. This protocol provides three cryptographic primitives such as integrity, confidentiality and authentication. These three primitives can be achieved with the help of Elliptic Curve Cryptography, Dual-RSA algorithm and Message Digest MD5. That is it uses Elliptic Curve Cryptography for encryption, Dual-RSA algorithm for authentication and MD-5 for integrity. This new security protocol has been designed for better security with integrity using a combination of both symmetric and asymmetric cryptographic techniques.

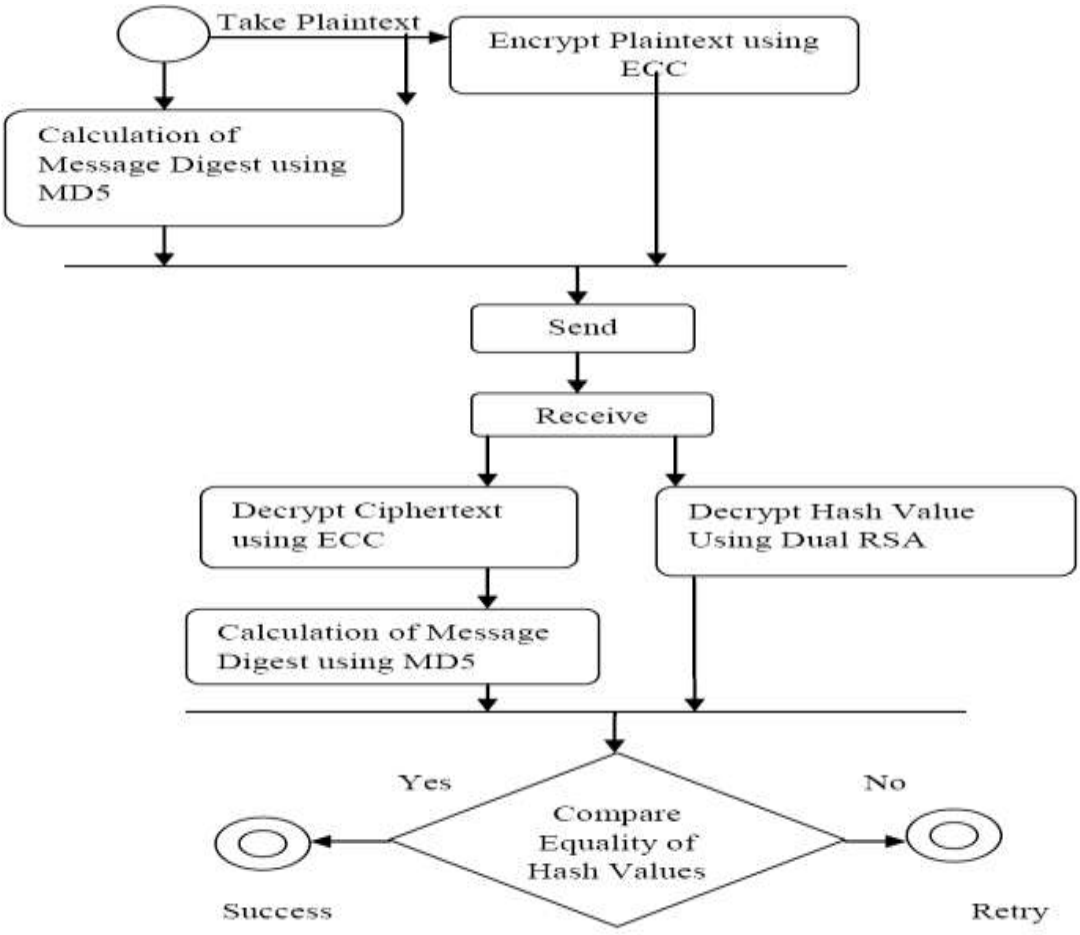


Figure 1: Architecture of Hybrid security protocol

The new Security Protocol has been designed for better security. It is a combination of both the Symmetric and Asymmetric Cryptographic Techniques. It provides the Cryptographic Primitives such as Integrity, Confidentiality and Authentication. The given plain text can be encrypted with the help of Elliptic Curve Cryptography, ECC and the derived cipher text can be communicated to the destination through any secured channel. Simultaneously, the Hash value is calculated through MD5 for the same plain text, which already has been converted into the cipher text by ECC. This Hash value has been encrypted with Dual RSA and the encrypted message of this Hash value also sent to destination. The intruders may try to hack the original information from the encrypted messages. He may be trapped both the encrypted messages of plain text and the hash value and he will try to decrypt these messages to get original one. He might be got the hash value and it is impossible to extract the plain text from the cipher text, because, the hash value is encrypted with Dual RSA and the plain text is encrypted with ECC. Hence, the message can be communicated to the destination with highly secured manner.

The new hash value is calculated with MD5 for the received originals messages and then it is compared with decrypted hash message for its integrity. By which, we can ensure that either the original text being altered or not in the communication medium. This is the primitive feature of this hybrid protocol.The main two modes in this protocol are senders and receivers the sender mode operate basically by entering the option. After the user select the sender mode, the plain text to be send can be entered and associated encryptions and hash function calculations will be performed. The receiver mode when activated receives encrypted message and then performs decryptions and calculate hash function from the plain text obtained after performing decryption. The main part of the protocol is the last phase where the both the hash values, one that obtained from sender and the one that the receiver calculated from plain text, is compared.

Sender will perform the following actions

- Receive user Input of plain Text
- Encrypt the given plain text using Elliptic Curve Cryptography, ECC and output the derived cipher text
- Send the Cipher text to destination.
- Calculate Hash value of plain text using MD5
- Encrypt the calculated message digest using Dual RSA
- Send encrypted message digest to destination

Receiver will perform the following actions

- Receives the cipher text and encrypted hash value...
- Decrypt hash value using dual RSA- hsv 1
- Decrypt Cipher Text using ECC
- Calculate new hash value of the decrypted Text using MD5 – hsv 2
- Compare hsv 1 and hsv 2 for its integrity.

Since the Encrypted text and encrypted hash value are sending separately, any intrusion/ security compromise on this can easily be identified using the above authentication technique.

IV. IMPLEMENTATION

The front end of the application in solely prepared in wxGlade. wxGlade is a GUI designer written in Python with the popular GUI toolkit wxPython that helps you create wxWidgets/wxPython user interfaces. At the moment it can generate Python, C++ and XRC (wxWidgets' XML resources) code. We can use a visual editor for creating forms, menus and toolbars with the mouse. Our design is saved in a .wxg file, which is the wxGlade file format. Then we generate source code or XRC by using visual tools or invoking wxGlade at the command line. We can also use wxGlade in your make file by generating source code only when the .wxg changes. A .wxg file can contain multiple forms, panels, menus and

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (9-16), Month: October - December 2014, Available at: www.researchpublish.com

toolbars and generate either a single file containing all classes or multiple files containing one class each. WxGlade does not manage events; file inclusion, function names, stubs or anything else but graphic interface code

Main Palette

The main window is a palette that hosts the menu and the widget choice buttons. If we pass the mouse pointer over a button a tooltip shows the button's description. The "Add a Frame" button and the "Add a Dialog/Panel" button bring up a dialog to add a frame, a dialog or a panel to our project. The "Add a Menu Bar" button asks you for the name of the class then adds a menu bar to our project. The "Add a Tool Bar" button asks you for the name of the class then adds a toolbar to our project. The other buttons in the main window add widgets to a form. When we click on one, the mouse pointer changes to an arrow. Then we can click on a sizer's empty cell to add the widget to it.

Figure 2: Main Palatte

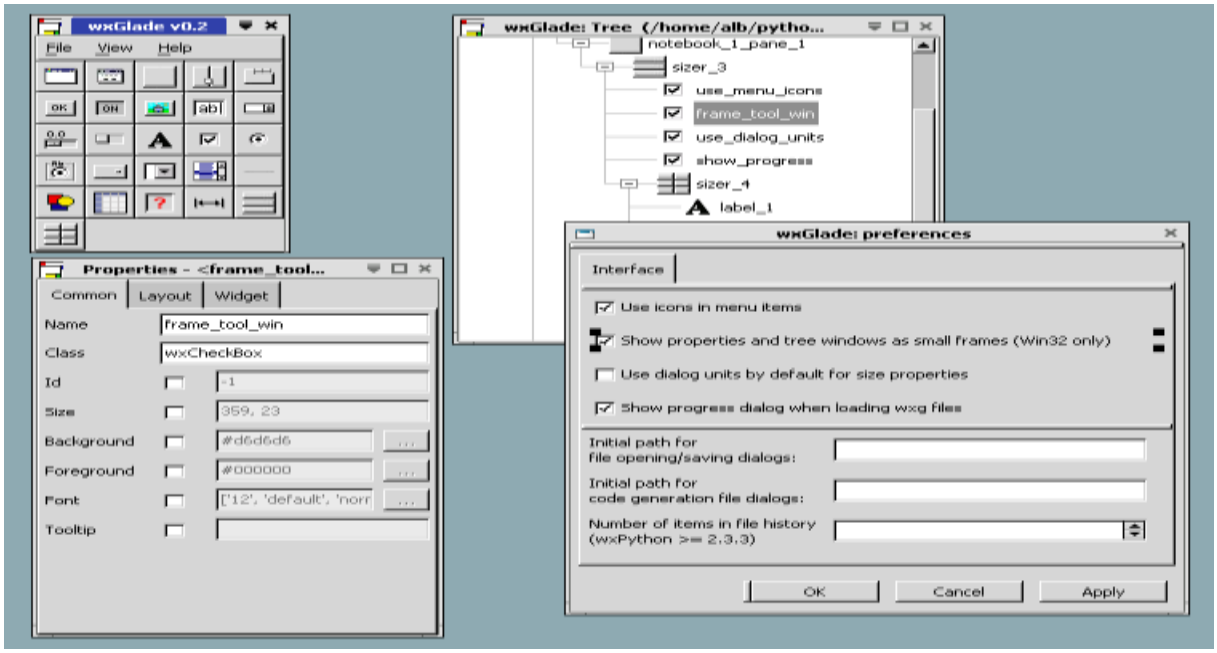


Figure 3: WxGlade properties

Design Window

The design window shows the frame or panel we are creating in WYSIWYG mode and allows us to select a widget from the main palette and to put it on an empty slot of a sizer. We can show the design window by double-clicking on the icon of a frame or dialog in the tree window.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (9-16), Month: October - December 2014, Available at: www.researchpublish.com

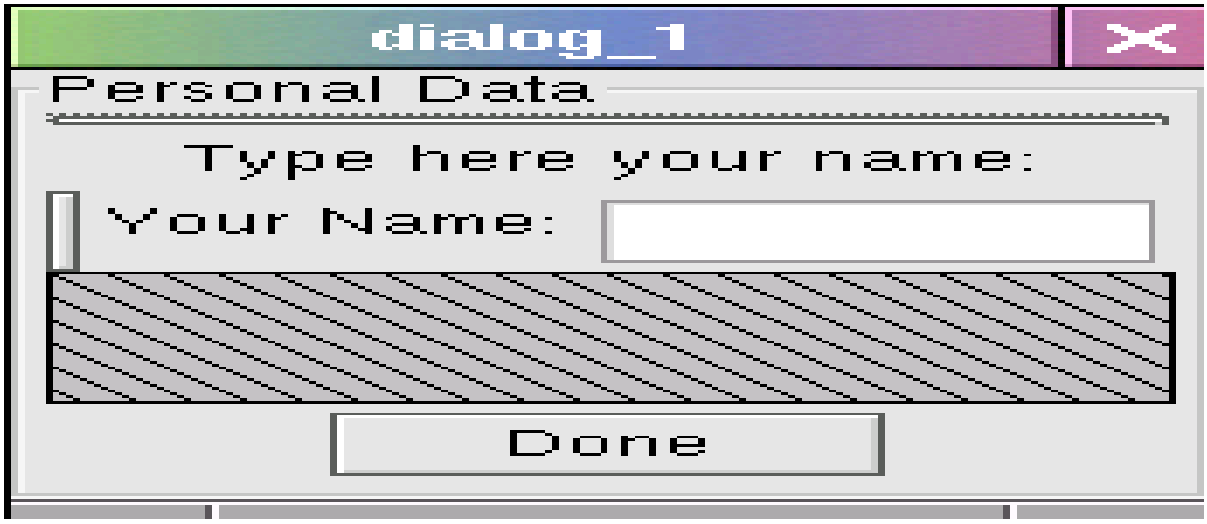


Figure 4: The Design Window

We have created separate interfaces for sender and receiver for each simple plain text transfer, ECC encrypted data transfer and finally for the completely secure data transfer. Screenshots for the above are included in the APPENDIX.

V. CODDING AND DEBUGGING

After designing the front end, we generated code for the designed wx frames, by selecting the Application item on the tree of widgets (it is the root) to make the Application tab appear on the properties window. This panel contains the options for the code generation: After the selection of the various options as described above, click on the Generate code button which created the code for the frames in the case of our particular protocol the code is in python language. Then we created the childs of created files and further codlings are done in that file. The coding is done in PyCharm 2.6.2, which is an Integrated Development Environment (IDE) used for programming in Python. The code is basically developed in python. .We used specialized packages and inbuilt classes to inherit additional capabilities needed for coding. We have installed pyECC and python-RSA so that encryption and decryption of the data can be done using build in functions, which made code generation much easier. The front end which is developed in wxglade should be mapped onto the program. This process can be done with the help of the option 'Generate code' as mentioned above which will create the python code for the frames. Then created the child applications for each of the generated files. Finally corresponding code for the protocol is integrated to this. Debugging is mainly concerned with the run time error that may occur during the execution of the application in phone or other compatible devices. There is also a specific mode available in PyCharm called the Debugging mode which is explicitly used for debugging and testing

VI. EXPERIMENTAL EVALUVATION

We compiled the python codes for the cryptographic algorithms such as RSA, ECC, and MD5 separately using PyCharm and resolved the errors such as import errors and syntax errors and checked whether it has been working properly. Our python interpreter was configured with python 3.2 in PyCharm so import errors occurred .The resolving was done by configuring with python 2.6 or 2.7. We tested the security assures given by the hybrid protocol by setting up wireshark, a free and open-source packet analyser.In wireshark filter we can give the details of the system for which the data has to be captured. (eg:port==12345 && data.data).In normal mode original data can be viewed as such, but by using data transfer using hybrid protocol then only the encrypted form is viewed.

VII. CONCLUSION

The aim was to implement a security protocol using hybrid cryptographic algorithms .Both symmetric and asymmetric algorithms such as ECC, MD5 and dual RSA are used. According to the requirement we have implemented a normal communication between two applications and then implemented Elliptic curve algorithm to encrypt the plain text to

obtain cipher text that was the second phase. In the third phase MD5 is used to calculate the hash value and obtained message digest is then encrypted using RSA. In the fourth phase, ECC encrypted plain text and RSA encrypted hash value are send to receiver. At the receiver side, the encrypted plain text is decrypted back to original plain text using ECC, and then calculated hash value again. Also RSA encrypted hash value is encrypted back. Both the received hash value and the calculated are compared to ensure the security. Finally, in the last phase is to incorporate our protocol into hadoop, which provides an environment for distributed computing. Two systems are to be setup with hadoop so that a bulk amount of data can be encrypted at source and decrypted at destination.

ACKNOWLEDGEMENT

The project is supported by Computer Science Department of University of Calicut. The faculty members especially Mrs. Usha Dileep who was the guide of my research work and provided her full support. The assistant technicians helped to configure the hardware equipments and helped me in case of software failure I would like to thank all my batch mates for their help to complete this work.

REFERENCES

[1] S. Subasree and N. K. Sakthivel." Design of a new security protocol using hybrid Cryptography algorithms." School of Computing, Sastra University, Thanjavur – 613401, Tamil Nadu, India.

[2] William Stallings, *Network Security Essentials Applications & Standards*, *Pearson Education* Asia.

[3] Priyanka Gandi&shabnam parveen,"Enhanced hybrid encryption algorithm for the security of network", kurukshethra university

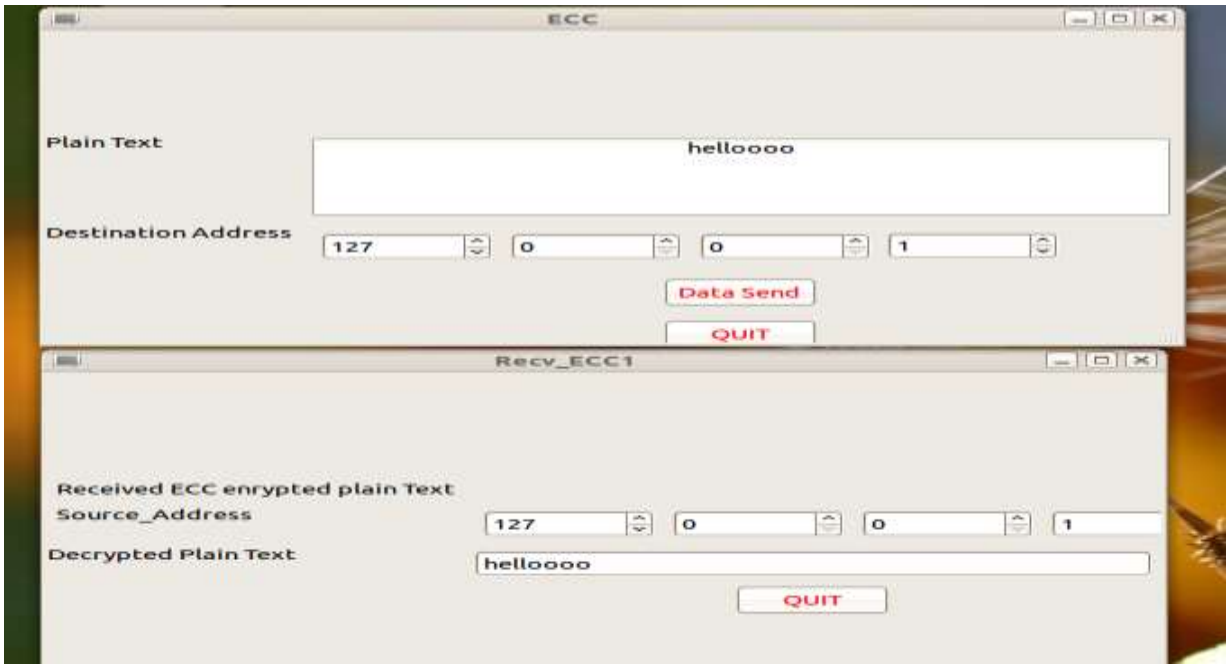
APPENDIX-A

The screen shot of all the output I got during my testing.

1) Initial welcome frame for sender and receiver

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (9-16), Month: October - December 2014, Available at: www.researchpublish.com

2) Sending encrypted data



3) Sending data using hybrid security protocol