

DESIGN OF A NEW SECURITY PROTOCOL USING HYBRID CRYPTOGRAPHY ALGORITHMS

S. Subasree and N. K. Sakthivel

School of Computing, Sastra University,
Thanjavur – 613401, Tamil Nadu, 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.*

1. INTRODUCTION

Curiosity is one of the most common human traits, matched by the wish to conceal private information. Spies and the military all resort to information hiding to pass messages securely, sometimes deliberately including misleading information [12]. Steganography, a mechanism for hiding information in apparently innocent pictures, may be used on its own or with other methods.

Encryption fundamentally consists of scrambling a message so that its contents are not readily accessible while decryption is the reversing of that process[14]. These processes depend on particular algorithms, known as ciphers. Suitably scrambled text is known as cipher text while the original is, not surprisingly, plain text. Readability is neither a necessary nor sufficient condition for something to be plain text.

The original might well not make any obvious sense when read, as would be the case, for example, if something already encrypted were being further encrypted. It's also quite possible to construct a mechanism whose output is readable text but which actually bears no relationship to the unencrypted original.

A key is used in conjunction with a cipher to encrypt or decrypt text. The key might appear meaningful, as would be the case with a character string used as a password, but this transformation is irrelevant, the functionality of a key lies in its being a string of bits determining the mapping of the plain text to the cipher text.

1.1 Why we need cryptography?

Protecting access to information for reasons of security is still a major reason for using cryptography. However, it's also increasingly used for identification of individuals, for authentication and for non-repudiation. This is particularly important with the growth of the Internet, global trading and other activities[12]. The identity of e-mail and Web users is trivially easy to conceal or to forge, and secure authentication can give those interacting remotely confidence that they're dealing with the right person and that a message hasn't been forged or changed.

In commercial situations, non-repudiation [12] is an important concept ensuring that if, say, a contract has been agreed upon one party can't then renege by claiming that they didn't actually agree or did so at some different time when, perhaps, a price was higher or lower. Digital signatures and digital timestamps are used in such situations, often in conjunction with other mechanisms such as message digests and digital certificates.

The range of uses for cryptography and related techniques is considerable and growing steadily. Passwords are common but the protection they offer is often illusory, perhaps because security policies within many organizations aren't well thought out and their use causes more problems and inconvenience than seems worth it[14,15].

In many cases where passwords are used, for example in protecting word processed documents, the ciphers used are extremely lightweight and can be attacked without difficulty using one of a range of freely available cracking programs.

2. TYPES OF CRYPTOGRAPHIC ALGORITHMS

2.1. Elliptic Curve Encryption

When using elliptic curves in cryptography[11], 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 $a = 0, b = 1, c = 2, \dots$ or there are other methods, such as ASCII, which accomplish the same task. Now, if we have $E: y^2 = x^3 + Ax + B \pmod{p}$, a curve in Weierstrass form, we want to let $m = x$. But, this will only work if $m^3 + Am + B$ 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 $1/2^K$ is an acceptable failure rate for embedding the information into a point on the curve. Also, make sure that $(m + 1)K < p$. Let $x_j = mK + j$ for $j = 0, 1, 2, \dots, K - 1$. Compute $x_j^3 + Ax_j + B$. Calculate its square root $y_j \pmod{p}$, if possible. If there is a square root, we let our point on E representing m be $P_m = (x_j, y_j)$. If there is no square root, try the next value of j [4,5].

So, for each value of j we have a probability of about $1/2$ that x_j is a square modulo p . Thus, the probability that no x_j is a square is about $1/2^K$, 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 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[10].

2.2. Dual 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 $dp = d \pmod{p - 1}$ for the computation in Z_p and with $dq = d \pmod{q - 1}$ for the computation in Z_q , 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 [7,8].

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, q) and a valid ciphertext $C \in Z_N$, the CRT-decryption algorithm is as follows:

- 1) Compute $C_p = C^{dp} \pmod{p}$.
- 2) Compute $C_q = C^{dq} \pmod{q}$.
- 3) Compute $M_0 = (C_q - C_p) \cdot p^{-1} \pmod{q}$.
- 4) Compute the plaintext $M = C_p + M_0 \cdot p$.

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 $(dp, dq, p, q, p^{-1} \pmod{q})$, 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[7].

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.

2.3 MD5 Algorithm

MD5[2] 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. M_i denotes a 32-bit block of the message input, and K_i denotes a 32-bit constant, different for each operation. s is a shift value, which also varies for each operation[1].

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[9].

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 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.

3. HYBRID SECURITY PROTOCOL ARCHITECTURE

It is desired to communicate data with high security. 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. This new security protocol has been designed for better security using a combination of both symmetric and asymmetric cryptographic techniques.

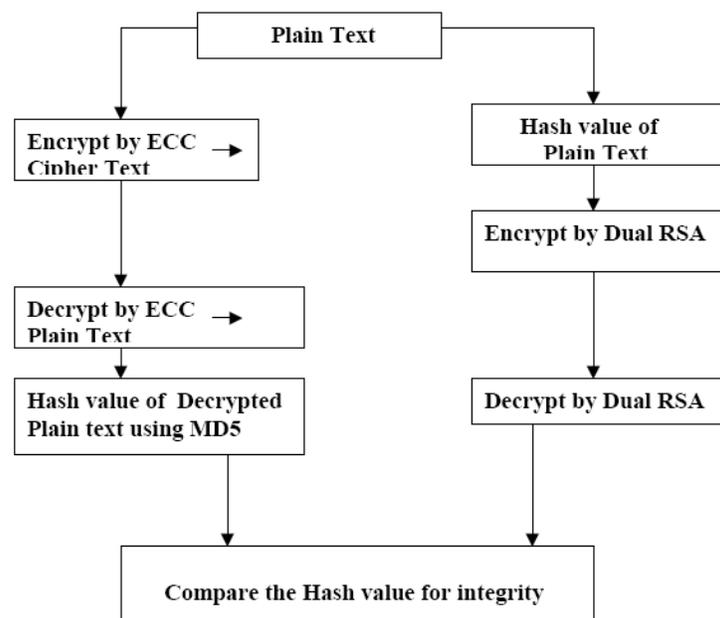


Figure 1 : Hybrid Protocol Architecture

As shown in the figure, the Symmetric Key Cryptographic Techniques such as Elliptic Curve Cryptography, and MD5 are used to achieve both the Confidentiality and Integrity. The Asymmetric Key Cryptography technique, Dual RSA used for Authentication.

The above discussed three primitives can be achieved with the help of this Security Protocol Architecture. The Architecture is as shown in the **Figure 1**. As shown in the figure, the Symmetric Key Cryptographic Techniques such as Elliptic Curve Cryptography, and MD5 are used to achieve both the Confidentiality and Integrity. The Asymmetric Key Cryptography technique, Dual RSA used for Authentication.

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 get 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.

4. RESULTS AND CONCLUSION

4.1 Comparison of RSA and Dual RSA

1) The Public Key Algorithms, RSA and Dual-RSA have been implemented in VC++ and we got the following results. As shown in the Figure 2, the original message for communication is stored in MyFile.txt and its size is 547 Bytes, which is shown in the report file.

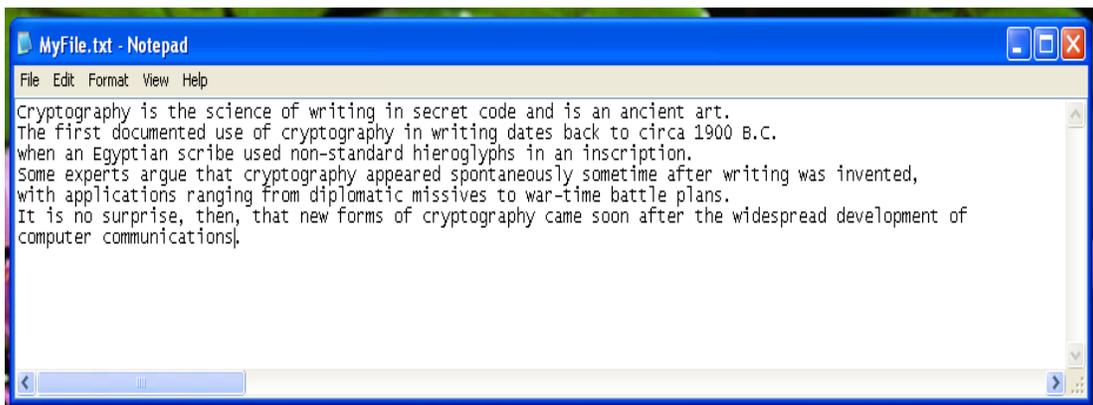


Figure 2 : Input File MyFile.txt

Figure 3 shows that the project main menu, which consists of various features. They are i. RSA Encryption, ii. RSA Decryption, iii. Dual RSA Encryption, iv. Dual RSA Decryption, and v. Graph, which is used to compare the computational costs of both the RSA and Dual-RSA

Figure 4 shows that RSA Encryption and Figure 5 shows that Dual RSA encryption. From the figure 6 it is clear that the RSA take one block at a time for encryption and decryption at a time. But the dual RSA take more time for encryption of two block at a time, but it take less time for decryption of two blocks. So, the RSA encryption and decryption time is greater than Dual RSA because Dual RSA perform the encryption and decryption operation for two blocks.

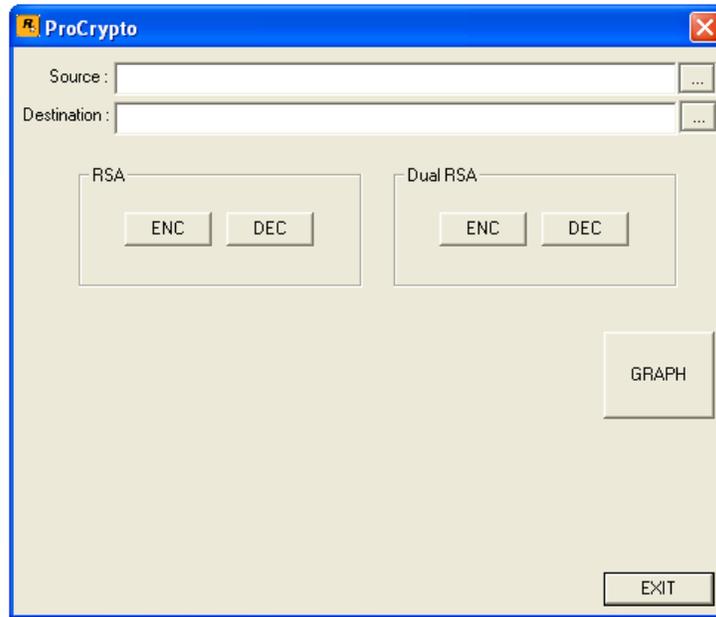


Figure 3 : Process of RSA and Dual RSA Encryption/Decryption

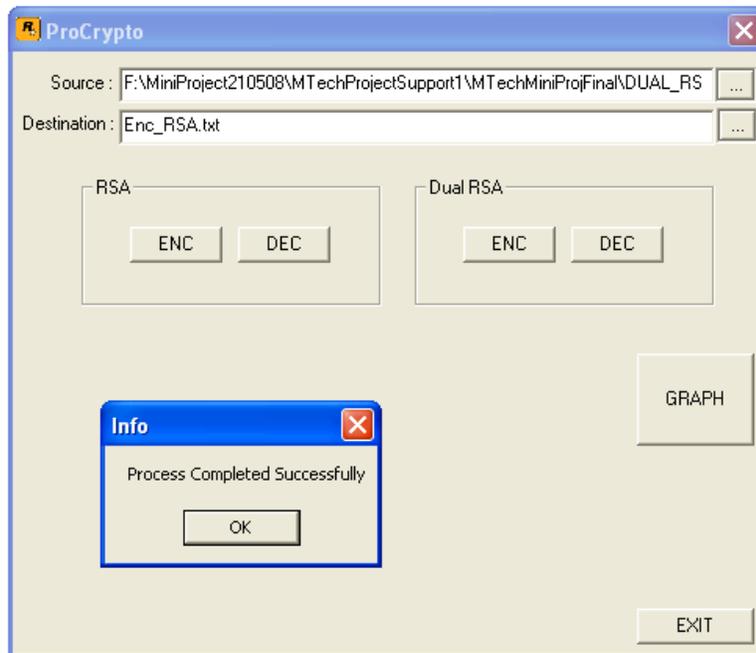


Figure 4 : RSA Encryption

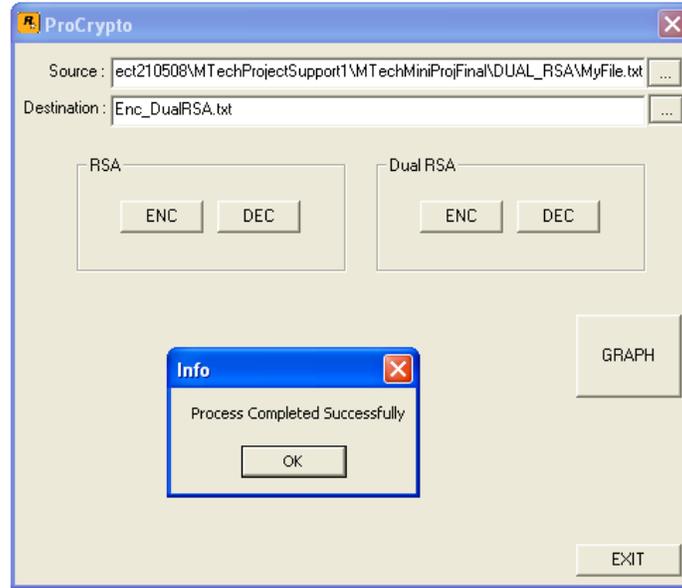


Figure 5 : Dual - RSA Encryption

5.2 Performance analysis of RSA and Dual RSA

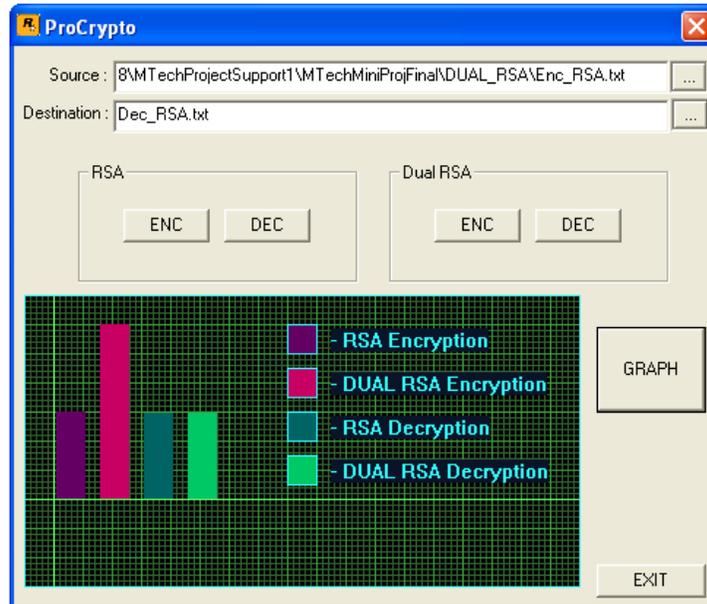


Figure 6 : RSA vs Dual RSA

S.No	Block Size	RSA		Block Size	Dual RSA	
		Encryption	Decryption		Encryption	Decryption
1	547	1078ms	1079 ms	1094ms	2140ms	1063ms
2	1094	2156ms	2159ms	2188ms	4120ms	2138ms

Figure 7 : Computational costs of RSA vs Dual RSA

Figure 6 shows that the Performance Analysis of RSA vs Dual RSA. From this figure, it is clear that the total computation time for Encryption and Decryption of Dual-RSA is less than that of ordinary RSA. From the Figure 7, it is observed that the total computation time for Encryption and Decryption of RSA is 4314ms as compared with the total computation time for Encryption and Decryption of Dual - RSA is 3203ms for the file size 547 Bytes. From the analysis it is clear that Dual RSA is better than RSA algorithm. So, for authentication we are going to use Dual RSA. Dual RSA take two block for encryption and decryption simultaneously.

5.3 Results of Hybrid protocol Architecture

Here, we are using three different mode of operation. The sender, Receiver and Intruder. We have to select the mode and process the information. The following figure represent the three different mode.



Figure 8 : Mode selection

If the mode is the sender, then we have to provide the key value and messages in the specified location.

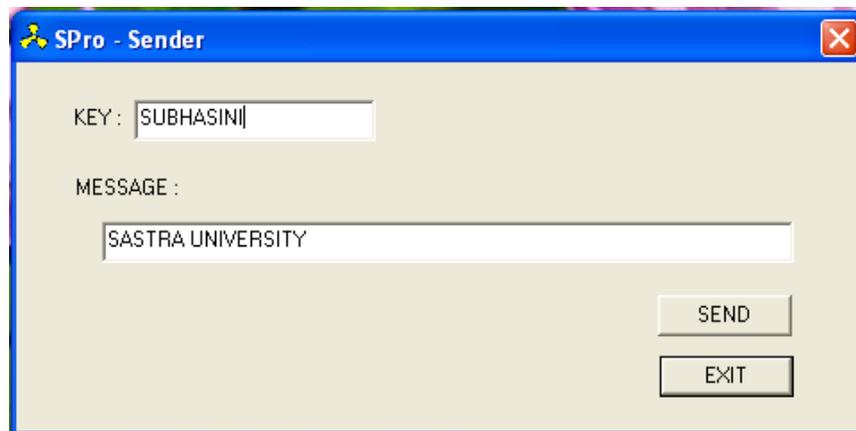


Figure 9 : Sender Mode

Figure 10 shows that the Receiver received the sender message with the key. From the figure, it is noted that, the intruder also received the key and not the message. Because, the message is encrypted with ECC and key is encrypted by using Dual RSA. And also noted that, the intruder derived different key for decryption, which is equivalent to the original key. Even though the intruder got the key he cannot able to get the original message because of Dual RSA. Because of Dual RSA we got two advantages one is the message cannot be decrypted and time required to perform the encryption and decryption operation less compare to RSA because Dual RSA perform encryption and decryption by two block at a time.

The new Public Key Cryptographic algorithm, Dual – RSA has been developed for better performance in terms of computation costs and memory storage requirements. It is also called RSA-CRT, because it is used Chinese Remainder Theorem, CRT for its Decryption. From the output, it is noted that Dual-RSA improved the performance of RSA in terms of computation cost and memory storage requirements.

It achieves parallelism. The CRT Decryption is achieved roughly $\frac{1}{4}$ times faster than original RSA.

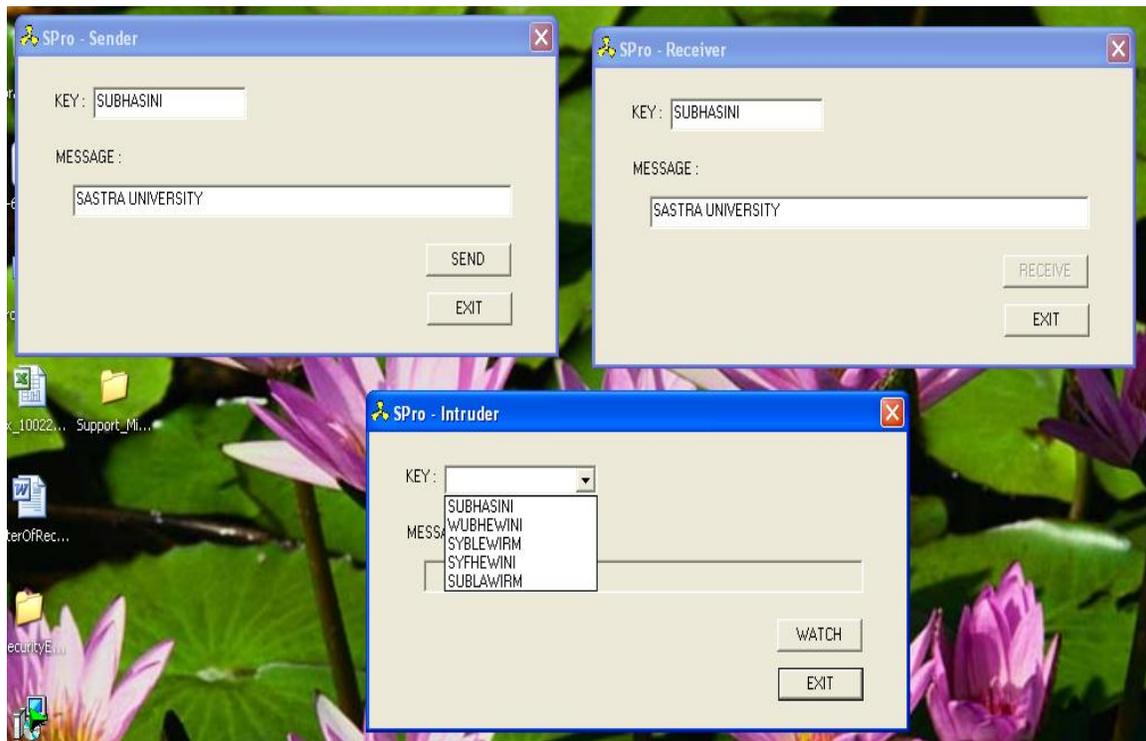


Figure 10 : Secured communication of Hybrid Protocol

6. REFERENCES

- [1] B. den Boer and A. Bosselaers, “An attack on the last two rounds of MD4”, Advances in Cryptology, Crypto '05, pages 194-203, Springer-Verlag, 2005.
- [2] B. den Boer and A. Bosselaers, “Collisions for the compression function of MD5”, Advances in Cryptology, Eurocrypt '07, pages 293-304, Springer-Verlag, 2007.
- [3] D. Bleichenbacher and A. May, “New attacks on RSA with small CRTexponent in Public Key Cryptography”, PKC 2006, volume 3968 of Lecture Notes in Computer Science, pages 1–13. Springer-Verlag, 2006.
- [4] D. Bleichenbacher and A. May, “New attacks on RSA with small secret CRT-exponents,” in *Public Key Cryptology—PKC 2006*, ser. Lecture Notes in Computer Science. New York: Springer, 2006, vol. 3958, pp. 1–13.
- [5] D. Boneh and G. Durfee, “Cryptanalysis of RSA with private key d less than N ,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1339–1349, Jul. 2000.
- [6] E. Jochemsz and A. May, “A polynomial time attack on standard RSA with private CRT-exponents”, 2007.
- [7] Hung-Min Sun, and et al., “Dual RSA and its Security Analysis”, *IEEE Transaction on Information Theory*, Aug 2007, pp 2922 – 2933, 2007
- [8] H.-M. Sun, M. J. Hinek, and M.-E. Wu, On the design of Rebalanced-RSA, revised version of [37] Centre for Applied Cryptographic Research, Technical Report CACR 2005-35, 2005 [Online]. Available: <http://www.cacr.math.uwaterloo.ca/techreports/2005/cacr2005-35.pdf>
- [9] H. Dobbertin, “The Status of MD5 after a Recent Attack”, *CryptoBytes*, 2(2): 1-6, 2007.
- [10] M. J. Hinek, “Another look at small RSA exponents,” in *Topics in Cryptology-CT-RSA 2006*, ser. Lecture Notes in Computer Science, D. Pointcheval, Ed. New York: Springer, 2006, vol. 3860, pp. 82–98.
- [11] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz, “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs”. *Proceedings of Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004)*, 6th International Workshop, pages 119–132, 2004.
- [12] Ravindra Kumar Chahar and et.al., “ Design of a new Security Protocol”, *IEEE International Conference on Computational Intelligence and Multimedia Applications*, pp 132 – 134, 2007
- [13] Ramaraj, E and Karthikeyan, S, “ A Design of Enhanced Security Protocol for Wireless Communication using Hybrid Encryption Technique, *Indian Journal of Computing Technology*, pp 22-29, May, 2006.
- [14] S. D. Galbraith, C. Heneghan, and J. F. McKee, “Tunable balancing of RSA”, 2005. Updated version of ACISP 2005.
- [15] S. D. Galbraith, C. Heneghan, and J. F. McKee, “Tunable balancing of RSA,” in *Proc. Inf. Security and Privacy, 10th Australasian Conf., ACISP 2005*, C. Boyd and J. M. G. Nieto, Eds., 2005, vol. 3574, pp. 280–292, Springer, Lecture Notes in Computer Science.

BIOGRAPHY

Dr. S Subasree got Bachelor Degree from Madras university in 1991 and she done her post graduate degree from Bharathidasan Univeristy in 1995 and M.phil from Manonmaniam Sundaranar Univeristy in 2001. She done her M.Tech and Ph.D in SASTRA University in 2006 and 2009 respectively. She got 13 years teaching experience. Now she will be serving as a Senior Assistant Professor in SASTRA Univeristy, Tamil Nadu, India. She has published more than 15 papers in International and National Journals and Conferences. Her research area includes Network Security, High Performance Soft Computing Techniques, Communication Network, and Biometric Cryptography.

Dr. N K Sakthivel got Bachelor Degree from Madras university in 1991 and she done her post graduate degree from Bharathidasan Univeristy in 1994 and M.phil from Bharathidasan Univeristy in 2000. She done her M.Tech and Ph.D in SASTRA University in 2004 and 2009 respectively. She got 15 years teaching experience. Now She will be serving as a Professor in SASTRA Univeristy, Tamil Nadu, India. She has published more than 18 papers in International and National Journals and Conferences. Her research area includes High Speed Communication Networks, Network Security, High Performance Computing, and Biometric Cryptography.