SECURITY SERVICES PROVISION AND ENHANCEMENT IN CLIENT/SERVER NETWORKS USING AES

Dr. SIDDEEQ Y. AMEEN*

Received on :28/6/2004
Accepted on :30/12/2004

Abstract

The paper attempts to provide most of the well-known security services through a design and implementation of a client/server-based network security system. The system is based on recommendation of the Advanced Encryption Standard (AES) together with some secure techniques. These techniques include message digest (MD5), data compression, data scrambling, traffic padding, novel PN sequence generator that is based on AES and the RSA for secure key distribution.

The generator is based on the standard ANSI X9.17 with triple AES instead of triple DES. Recommended statistical tests are carried out to check the randomness of the new generator. The results of these tests show that the new generator passes all the required tests successfully. The latter results being even better than those achieved by the standard generator ANSI X9.17 using AES or triple DES.

Finally, the security system was implemented and tested between two PCs in a local area network. The results clearly demonstrate the successful operation of the security system through the secure transmission of data. The security of the proposed system is shown to be efficient since all the combined techniques are well-established and recommended. Furthermore, with the multi-keys required to perform the operation, the tasks of cryptanalyst with available computing power will be difficult.

الخلاصة

يحاول البحث توفير معظم الخدمات الأمنية من خلال تصميم وتنفيذ منظومة لحماية شبكة حاسوب نوع العميل والخادم معتمداً على المشفر القياسي المتقدم (AES) ومجموعة من عدة تقنيات حديثة وكفوءة . تضمنت هذه التقنيات بصمة الرسائل نوع (MD5)، وكبس البيانات وبعثرتها، ومولد ارقام شبه عشوائي مبتكر، وحشو البيانات العشوائي، ومنظومة تشفير كتلية حديثة ومقرة عالمياً من نوع AES، ونظام تشفير المفتاح المعلن من نوع (RSA) للتوزيع الآمن للمفاتيح.

يعتمد مولد الأرقام شبه العشوائي المبتكر على طريقة (ANSI X9.17) ومنظومة التشفيرالمتقدمه الثلاثية (triple AES)، ويهدف الى قياس عشوائية البيانات من المولد. تم تطبيق عدد من الاختبارات الإحصائية المعروفة والتي أظهرت نتائجها نجاح الاختبار، إذ امتاز المولد بعشوائية أفضل من نظام التوليد القياسي المقر عالمياً ANSI X9.7 والذي يستخدم (triple DES).

أخيراً، تم تنفيذ وفحص المنظومة بالكامل من خلال تحقيق اتصال بين حاسوبين في شبكه محلية والذي أثبت نجاح منظومة الحماية الأمنية المطبقة، فضلاً عن السرية العالية بسبب كون المنظومات المستخدمة كفوءة وحجم المفاتيح المستخدمة، والتي تجعل من عملية كسرها بأستخدام طرق التحليل المعروفة والقدرات الحاسوبية الحالية أمراً صعباً.

KEYWORDS: Network Security, PN Sequence generators, Data Security.

* Department of Computer Engineering and Information Technology, University of Technology, Baghdad, Iraq,

1. Introduction

Recently, messages are increasingly being exchanged over computer networks, which has resulted in tremendous increase in the speed of communication. At the same time, this has given rise to host new problems; security threats. Therefore, a form of security is required such as cryptography. Cryptography is not the only way of exchanging messages securely in a network. Network security means a protection of the network assets from different kinds of threats in the network by implementation of different security services using various security mechanisms [1,2]. Security services that attempt to enhance the network security fall into six different categories: cofidentiallity, integrity, authentication, non-repudation, access control, and avilability [1,3]. Several application security mechanisms in a number of application areas, including electronic mail, web access, client/server interation and others were developed. These involve the Secure/Multipurpose Internet Mail Extension (S/MIME), Pretty Good Privacy (PGP), Secure Socket Layer (SSL) [4]. However, because of advance in attacking methodology, these mechanisms need to be revised to consider the recent updates in security algorithms and computation facilities.

The Data Encryption Standard (DES) is the most well-known cryptographic mechanism in history. It remains the standard means for securing electronic commerce for many financial institutions around the world. The most striking development in the history of cryptography came in 1976 when Diffie and Hellman published Diffie-Hellman key exchange and later the RSA public key cryptosystem [2].

In October 2000, the American National Institute for Science and Technology (NIST) announced that the AES, the Rijndael algorithm, will replace the DES. This is based on the Federal Information Processing Standard (FIPS) proposal to specify a symmetric that must satisfy a number of criteria: strong security, simple design, and good performance [5].

Thus the paper attempts to design and implement a security system for computer networks that satisfy most of the security services requirements using

* Department of Computer Engineering and Information Technology, University of Technology, Baghdad, Iraq,

the most recent available standard AES with the other standards MD5, LZW and ANSI-X9.17.

2. Network Security Techniques

2.1 Message Digest System and MD5

The system takes input message of arbitrary length and produces a 128-bit message digest as an output. The input is processed in 512-bit blocks [6]. The following five steps are performed to compute the message digest of the message [6]:

a- The message is "padded" (extended) so that its length (in bits) is congruent to 448 modulo 512. Padding is always performed, even if the length of the message is already congruent to 448 modulo 512. In padding, a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448 modulo 512. In all, at least one bit and at most 512 bits are appended.

b- A 64-bit representation of message's length (before padding bits were added) is appended to the result in previous step.

c- A four-word buffer (**A, B, C, D**) is used to compute the message digest. Each of the word **A, B, C, D** is a 32-bit. These are initialized to the following values in hexadecimal, low-order bytes first:

Word A: 0x01 23 45 67

Word B: 0x89 ab cd ef

Word C: 0xfe dc ba 98

Word D: 0x76 54 32 10

d- Four auxiliary functions that each takes as input three 32-bit words and produces as output one 32-bit word are applied as follows;

F (B, C, D) = BC v NOT (B) D.

G (B, C, D) = BD v (C NOT (D)).

H (B, C, D) = B XOR C XOR D.

I (B, C, D) = C XOR (C v NOT (D)).

Now, the main loop of the algorithm begins. This loop will be computed for the whole blocks in the message. The four variables are copied into different variables: a gets A, b gets B, c gets C and d gets D. The main loop has four rounds, all very similar. Each round uses a different operation 16 times. Each operation performs a nonlinear function on three of the variables a, b, c, and d. Then, it adds that result to the fourth variable, a sub-block of message and constant. The result rotated to the left a variable number of bits and added to the result of a, b, c, or d. Finally, the result replaces one of a, b, c, or d.

e- The final output is the concatenation of A, B, C, and D.

2.2. Data Compression System and LZW Fundamentals

LZW compression replaces strings of characters with single codes. It does not do any analysis of the incoming text. Instead, it just adds every new string of characters it sees to a table of strings. Compression occurs when a single code is output instead of a string of characters.

The code that the LZW algorithm outputs can be of any arbitrary length, but it must have more bits in it than a single character. The first 256 codes (when using eight bit characters) are by default assigned to the standard character set. The remaining codes are assigned to strings as the algorithm proceeds. A quick examination of the algorithm shows that LZW is always trying to output codes for strings that are already known. Each time, a new code is output; a new string is added to the string table [7]. The algorithm adds a new string to the string table each time it reads in a new code. All it needs is translation of each incoming code into a string and send it to the output [7].

There is a single exception case in the LZW compression algorithm that causes some trouble to the decompression side. If there is a string consisting of a (STRING, CHARACTER) pair already defined in the table, the input stream then sees a sequence of STRING, CHARACTER, STRING, CHARACTER, STRING, the compression algorithm will output a code before the decompression gets a chance to define it. Fortunately, this is the only case where the decompression algorithm will encounter an undefined code. Since it is in fact the only case, we can add an exception handler to the algorithm. The modified algorithm just looks for the special case of an undefined code and handles it [7].

2.3. Block Cipher System

The encryption algorithm used in this paper is the AES or Rijndael algorithm. Rijndael is a symmetric block encryption algorithm that encrypts blocks of 128, 192, or 256 bits and uses symmetric keys of 128, 192, or 256 bits, where all combinations of block and key lengths are possible [1]. It has the following features [5]:

a- The implementation of Rijndael can at the least cost be protected against attacks that are based on measurements of the time behavior of the hardware (so-called timing attacks) or change in electrical current use (so-called power or differential power analysis attacks).

b- Rijndael algorithm can most rapidly encrypt and decrypt data.

c- Rijndael makes use of very limited resources of RAM and ROM memory.

d- Rijndael has the best performance in hardware implementation.

Each block of plaintext is encrypted several times with a repeating sequence of various functions, in so-called rounds [1]. The number of rounds Nr depends on the block length Nb and key length Nk. If at least the block or key length is 256 bits, there are 14 rounds, if both the block and key length are 128 bits; there are 10 rounds [1]

The algorithm is based on powerful substitution–linear transformation networks. It consists of an initial round (AddRoundKey), and Nr standard rounds. The first Nr-1 rounds are similar and they consist of four transformations, called: ByteSub (Substitution Bytes), ShiftRow (Shift Rows), MixColumn (Multiply Columns), and AddRoundKey (XORed by key). The last round has only the transformations ByteSub, ShiftRow, and AddRoundKey]. Further details about the Rijndael algorithm and its operation can be found elsewhere [5].

Triple encryption is a better idea that operates on a block three times with three different keys. The sender first encrypts the plaintext using the third key; then decrypts it using the second key, and finally encrypts it with the first key. The receiver decrypts the ciphertext using the first key, then encrypts it using the second key, and finally decrypts it with the third key. Three keys can be defined as follows:

$$C = Ek_3\ [Dk_2\ [Ek_1\ [P]]] \qquad (1)$$

$$P = Dk_1\ [Ek_2\ [Dk_3\ [C]]] \qquad (2)$$

where C, P, E, D and k_i are ciphertext, plaintext, encryption, decryption and its key respectively.

2.4. Data Scrambling and Pseudo-Random Number Generator

Data scrambling is done by making exclusive-OR (XOR) process between the data and pseudorandom number generator (PRNG) output. The

Pseudo-random numbers play an important role in the use of encryption for various network security applications. A sequence generator is pseudo-random, if it has the following properties [6]:

a- It looks random. This means that it passes all the statistical tests of randomness.

b- It is unpredictable. It must be computationally infeasible to predict what the next random bit will be, given complete knowledge of the algorithm or hardware generating the sequence and all of the previous bits in the stream.

The output of a generator satisfying these properties will be good enough for one-time pad, key generation, and any other cryptographic applications that require a truly random number generator [6].

There are many types of pseudo-random number generators (PRNGs) that depend on the cryptographic applications such as: cyclic encryption, DES output feedback mode, Blum Blum Shub generator, and ANSI X9.17 pseudo-random number generators. One of the strongest pseudorandom number generators is specified in ANSI X9.17. Fig. 1 illustrates the generator, which makes use of triple DES [1].

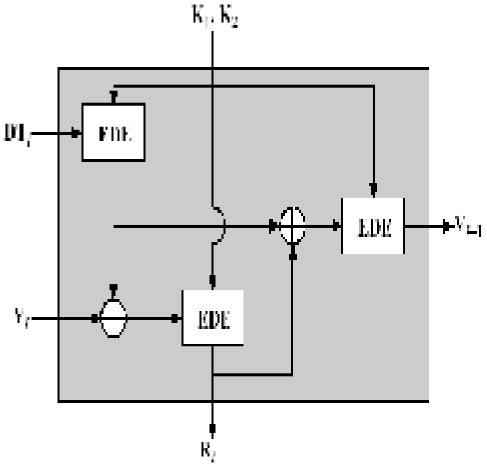

Fig. 1 ANSI X9.17 Pseudorandom Number Generator

From Fig. 1, the following quantities are defined;

DT_i: Date/Time value at the beginning of the ith generation stage.

V_i: Seed value at the beginning of the ith generation stage.R_i: Pseudorandom number produced by the ith generation stage.

K_1 and K_2: DES keys used for each stage.

These quantities are related to each other by the following expressions [1]:

R_i=EDE K_1, K_2 [$V_i \oplus$ EDE K_1, k_2 [DT_i]] (3)

V_{i+1} = EDE K_1, K_2 [$R_i \oplus$ EDE K_1, K_2DT_i]] (4)

where, EDE refers to the sequence encrypt-decrypt-encrypt.

The proposed pseudorandom number generator uses triple Rijndael instead of triple DES in the standard ANSI X9.17. Other parameters are the same as that specified by the standard.

It is impossible to give a mathematical proof that the generator is indeed a random bit generator. Tests help detect certain kinds of weaknesses the generator may have. This is accomplished by taking a sample output sequence of the generator and subjecting it to various statistical tests [8].

Let S=S_0, S_1, S_2,….,S_{n-1} be a binary sequence of length n. The five statistical tests that are commonly used for determining whether the binary sequence **S** possesses some specific characteristics that a truly random sequence would be likely to exhibit are:

Frequency, serial, poker, run and auto-correlation tests. In these tests, the value of χ^2 is calculated for different degrees of freedom (DOF) and compared with the theoretical value that is required to pass the randomness test as follows [8];

a- For frequency test (monobit test)

χ^2= $(n_0-n_1)^2/n$ (5)

where n_0 and n_1 denote the number of 0's and 1's in (S), respectively.

b- For serial test

$$c^2 = \frac{4}{n-1}\sum_{i=0}^{1}\sum_{j=0}^{1} n_{ij}^{2} - \frac{2}{n}\sum_{i=0}^{1} n_i^{2} + 1 \quad (6)$$

where n_{00}, n_{01}, n_{10}, and n_{11} denote the number of occurrences of 00, 01, 10, 11 in S, respectively.

c- For poker test

$$c^2 = \frac{2^m}{F}\sum_{i=0}^{m}\frac{x_i}{(i^m)}^{2} - F \quad (7)$$

where in this test n is partitioned into blocks of size m bits and F=(n/m). The frequency of each type of section of length m in a sequence is counted and the numbers x_0, x_1,, x $_{m-1}$ where x_i is the number of m-bit blocks having i ones and m-i zeros are evaluated.

d- For autocorrelation test

The value of χ^2 is as follows:

$\chi^2 = (A(d)-\mu)^2 / \mu$ (8)

where

$$A(d)= \sum_{i=1}^{n-d} a_i * a_{i+d} \quad 0 \le d \le n-1 \quad (9)$$

and

$\mu = (n_1^2\ (n-d)) / n^2$ (10)

A(d) is the autocorrelation function of the n-component sequence S and its shifted versions $a_1, a_2,\ldots, a_n$,.

Computer simulation has been carried out to investigate the performance of the generators. The tested generators include the standard ANSI X9.17 that uses the triple DES, the modified ANSI X9.17 but with single Rijndael and the proposed generator of the triple Rijndael instead of triple DES. All computer simulation programs have been implemented in C^{++} and executed using PIV personal computer running under Windows XP operating system. In the tests, the values of χ^2 for the frequency, serial, poker and autocorrelation tests were calculated and tabulated for different degree of freedom (DOF) as shown in Tables 1 and 2 for n set to 100. It is worth to mention that the theoretical value of χ^2 to pass these tests are less than 3.84, 7.81, 11.1, and 3.84, for frequency, poker, serial and auto-correlation tests, respectively.

The results of the tests for the three different generators show the remarkable performance achieved by that of the triple AES when it replaces the triple DES in the original standard. It is worth to mention that extra tests were also carried out, whose results are not considered, such as run test, tests for different degrees of freedom and different block sizes. All these tests results, which are not included, were agreed with that shown in Tables 1 and 2.

Table 1: Different statistical tests for PN-sequence generators

Type	DOF	3DES	AES	3AES
Freq.	1	3.800	0.160	0.001
Poker	5	5.410	4.000	1.700
Serial	3	6.160	7.080	0.240

Table 2: Auto-correlation test with degree of freedom =1

Shift	3DES	AES	3AES
1	0.050	1.685	0.162
2	1.574	0.072	0.514
3	0.887	2.755	0.243
4	3.452	0.032	1.363
5	1.263	3.073	0.051
6	1.166	2.923	0.032
7	0.051	0.456	0.099
8	0.073	0.130	1.829
9	3.766	3.424	3.297
10	2.645	1.176	0.033

2.5. The RSA Public Cryptosystem

The system developed by Rivest, Shamir and Adleman makes use of an expression with exponentials. The mathematical description of RSA is as follows: take two large primes, p and q, and compute their product n = pq; n is called the modulus. Choose a number, e, less than n and relatively prime to ϕ (n), where ϕ (n)= (p-1)(q-1). This means that e and ϕ (n) have no common factors except 1. Find another number d such that ed=1 mod ϕ (n). Plaintext is encrypted in blocks, with each block having a binary value less than some number n. Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C [6]:

$C = M^e \bmod n$ for encryption (11)

$M = C^d \bmod n$ for decryption (12)

Both sender and receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus the public key is KU={e, n} and the private key is KR={d, n}.

2.6. Link Encryption

To deter a listener who is monitoring, a network administrator inserts “noise” into the system by randomly sending messages to each node in the network [1]. This can be applied by using a traffic-padding device. Traffic padding is essentially a link encryption function. It produces ciphertext output continuously, even in the absence of plaintext. When plaintext is available, it is encrypted and transmitted. When input plaintext is not present, random data is encrypted and transmitted. This makes it impossible for an attacker to distinguish between true data flow and padding and therefore impossible to deduce the amount of traffic [1].

3. Network Security System Design

The proposed security system involves many techniques mentioned in section 2 to achieve most of the security services. These techniques have been combined together as shown in Figs. 2 and 3.

The system has been proposed to provide a security for the local area operating in a client server mode or even peer-to-peer network. The design is based on the well known system pretty good privacy PGP that still has the security power in its usage in Swiss banks. Thus, the security level of the proposed system should have a security level as that of PGP or even better since the inclusion of AES, link encryption and other modifications will improve the performanceofPGP.

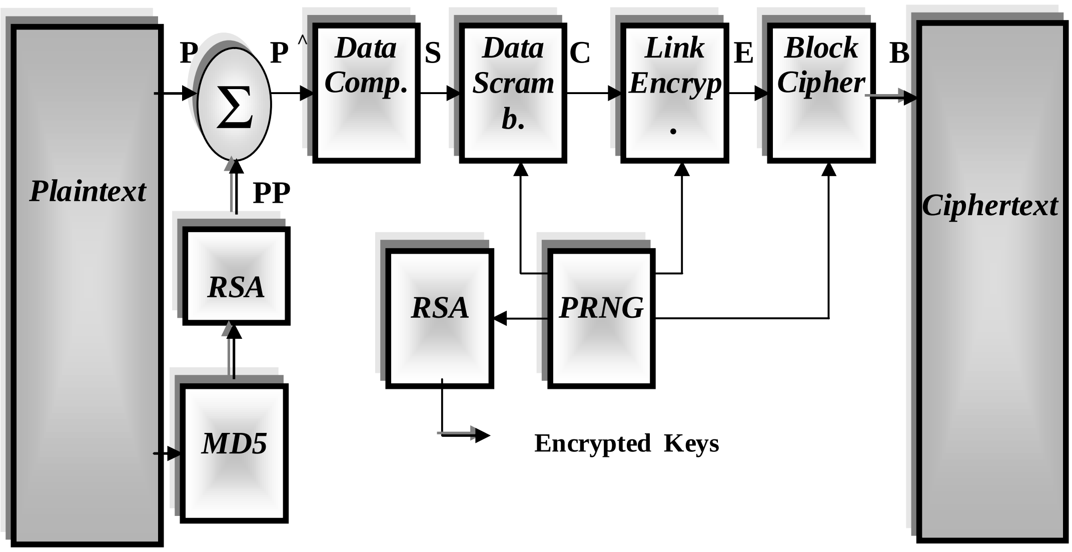

Fig. 2 The Encryption Process in the Client Side

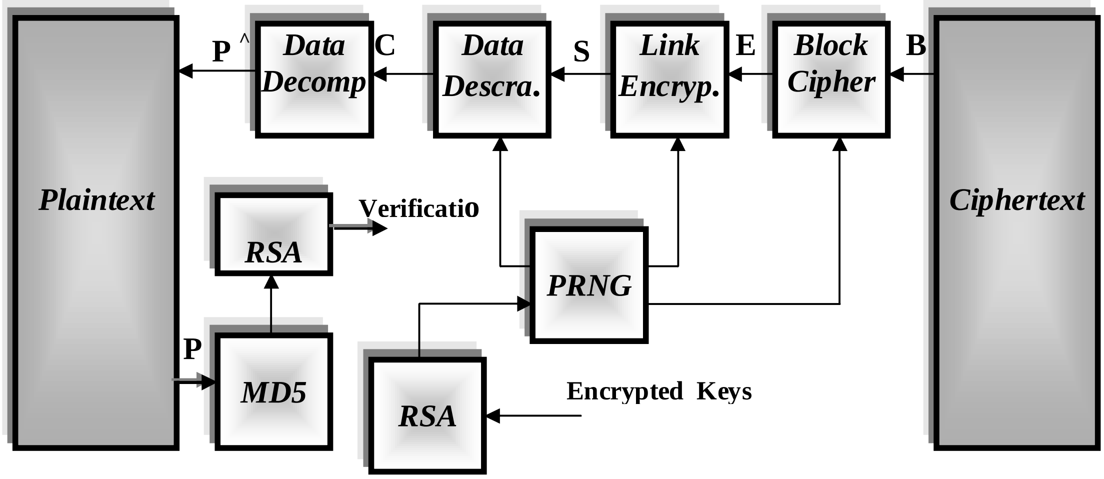

Fig. 3 The Decryption Process in the Server Side

The proposed system shown in Fig. 2 has two paths from the client to the server and vise versa. The first path provides the secure transmission of the plaintext.

On the other hand, the second path provides the security for the session keys using the RSA algorithm. It is worth to mention that the channel between the client and the server assumed to be unsecured.

The operation of such system can be described by the following procedure used to send a message from the client to the server. A detailed description of such procedures can be summarized in the implementation phase.

4. System Implementation And Evaluation

A software package has been developed to perform the security issues required in the network suggested. The software package was written in C^{++} and Visual Basic under the window XP.

In the design and implementation of the system, a special user-friendly page has been designed using visual basic to enter the required keys to implement the scheme. It has been designed such that the initial phase of the path setup will begin with interrogation between the user and the security system asking for keys required which can be entered in the form of characters. These keys include; three keys used for triple AES, data, time, seed and the two keys required by the PN-sequence generator, and the initial vector required by the message digest.

Finally, software implementation of the above scheme has been tested on a LAN and two Internet nodes as follows:

4.1. Client Side (The sender)

Let the message that client wants to send to the server is P as shown:

The proposed design has been software simulated and implemented in LAN and between two Internet nodes. The implemented security system has been tested and the results of the test have shown a remarkable performance that has been achieved without any problem in real time transfer of data in the network

The system calculates the message digest for the message P using (MD5) technique. This digest M in hexadecimal format is:

3c4a359ade745c27537e3b08672425cb

The latter then signed (encrypted using the client private key)

$$pp=M^{dc} mod n_c \qquad (13)$$

where n_c is the product of large prime numbers p_c and q_c.

The RSA algorithm used by the system uses a key length of 1024 and 2048 bits that corresponds to an RSA modulus of at least 309 and 617 decimal digits, respectively. Bearing in mind that time required to factor 200 decimal digits number is estimated to be 52,000,000 years in 1998. Therefore, this can be considered to be safe in the future [9]. However, work can be done to increase the size of the prime number used in the RSA encryption.

The encrypted message pp appended to the plaintext P to give the message $P^{\wedge,}$ which is compressed using LZW compression to give the compressed message C.

```
xœm  Mnf0 …÷H-aNPa□Y†  J•  ®z'□
¤ِسه□=‛¶é¢^Œ5z&”7¾´  ذِ□ىفى#c_ذ  F à„h
پ¸!V˜¤ث6,□هGي“A›yBƒ6ع̨B  ¼?6´„لFق-X
        ¬cى  €ئٔ؟ ؤç!k؛A¸ Bك“ٲX"ˆٱvy
ثDµ°-□Qه  ص¼¯èV   Œ□u  0M`F?8o□مَ=“u-
5^™¹jف□Bظ³ٳ¾جbâ           ¤□ص
+×ْ□•ن□RQث|¹êü%Mث¢»/|Kرش̤B6b›¨„ج±ٱ³ه/N
‹ح®ضه¾  m‹4‘•  E□ح²µ”Yv*²تْ/ء;□بv‡¬K  Qe
wëViق□#mهN™□x@~¾4ù  weٌ
```

The compressed message C is scrambled with random data generated by the PN sequence generator to give the scrambled data S shown:

```
 sÆd.  v#Ÿ;1Ûx+L·UÛ€²Ú   Ù  2nÊú}¹®.ªù
™ù¥  W×   ÁWùÈRI*—
§  âIœ¬•  ŽÅÜáXX'æ(z2sùliÞ´Ý¯$MÅ| ~õ©AÁ·.
AÕY{ÎWBæE×=¹íhÆ¾¾2ˆ]ç¡ÁÃ¸    ÃÐ„¸Þ  ›
2Õx×i˜´M·‰Ç-GS  ÂÇ-ì  ¨›Ý…ÔÿPUMJŸG4ê_
ŽBd8  "è-«#d<.ŸA™-õ  Ýäëi?  'Í³»bK  ,1É8ƒ
 2¡2ÊË„Ú¾x³®ƒàx)¬Ÿ©úÚÝ€ÌP³½î:  Å-MTÊ\ö
ášÔö¢/p©4E5ú{Df„Ó°'uÕ  Ý{àk(
```

The scrambled message S is splitted into several groups, each of 256 bits length, to be encrypted by the triple AES algorithm using three time session keys of 256 bits length. Link encryption using traffic padding has been employed on the scrambled message S before the triple AES encryption to reduce the opportunity for traffic analysis. This has been achieved by padding continuous false random data generated by PN sequence generator. The padding occurs randomly according to certain random sequence that needs to be informed to the other side.

The padding procedure includes the transmission of twelve of 256 bits size only one of these blocks will be the true data block. The remaining eleven false blocks are taken from the PN sequence generator. The selection of the position of the true block is achieved through the use of random number generator that generates random numbers between 1 and 12. It is worth to mention that the expansion obtained depends on the security level required and can be reduced from 12 to any size depending on the security level required.

The triple AES encryption output together with the client ID and the session keys used are then transmitted to the server. The seeds used by the random number generator and the keys used by the triple AES together with any initial vectors used were all encrypted by the RSA server public key and sent to the server using path 2 as shown in Fig. 2.

4.2. The Server (The Receiver)

On the receipt of the packet from the sender, the receiver recovers all the keys by the RSA algorithm using the server private key as shown in Fig. 3. The server then uses the recovered key and the symmetric Rijndael to decrypt the message **B** using the three required keys to give the message E. The scrambled **S** message is obtained by dropping all the padded blocks. XORing **S** with the PRNG will return the compressed version of the message **C**. The server uses the same compression algorithm to recover **P**^ which includes the original plaintext and the encrypted message digest. It is up to the receiver to check the client's message digest by using MD5 technique on the message **P** and performing RSA decryption, but with the client public key. The latter procedure will authenticate and provide the signature required by the security system.

Finally, for the presentation purposes, the visual basic program is used to transmit and receive data between the client and server network with a user who friendly interface Graphical User Interface GUI. This program uses a winsocket technique to interface with the other encryption/decryption programs. Furthermore, the GUI for transmitting/ receiving operation has been used to simplify the execution of the program to the user and to hide all the actual processing work. This will provide certain transparency to the security system.

5. Assessment Of Results

The designed system in this research uses the hash function (message digest MD5) to ensure that the original message did not change through the transmission. This will provide the authentication and digital signature services. This algorithm is simple to implement and do not require large programs or substitution tables. The cryptanalyst attempts to use differential cryptanalysis against a single round of MD5 and was ineffective against all four rounds [9].

The compression algorithm used in this work is the LZW algorithm. This method is popular in practice. Its advantage over other algorithms is in the speed because there are not too many string comparisons to perform [7].

The RSA algorithm has been used to encrypt all the keys used in this system. This will provide the access control and authentication services. The researchers in the field of RSA found that the time required to break 200 decimal digits number was estimated to be 52,000,000 years in 1998. Therefore, this algorithm can be considered to be safe in the future, since the computation power is still limited [9].

The keys used by the system for data scrambling and link encryption were generated by the new generator based on ANSI X9.17 pseudorandom number generator and triple AES, while the initial values of the generator are generated from the computer random engine. The statistical test results show that the generator output passes most of the randomness tests and even is better than that of the ANSI X9.17 generator.

The traffic-padding device used in this system makes it impossible for an attacker to distinguish between true and false data and therefore impossible to deduce the amount of traffic. Finally, the integrity has been achieved by MD5 and the confidentiality through the use of compression, scrambling, traffic padding and AES.

By the length of the AES keys, it is guaranteed that there is a huge amount of different keys exactly 2^{128}, 2^{192} or 2^{256} possible keys to be guessed using the well-known method brute force attack. Furthermore, the AES has other advantages such as the speed, memory requirement and others [10]

6. Conclusions

Many techniques have been developed to secure network. Not all of these techniques provide the same services and security issues, so a combination of several technologies has been used to ensure better level of security. Of course, the technologies adopted in the scheme are well-recommended with high security that should leads to a high security system.

The test of the system with such complex combination in data transfer between two computers in LAN and between two internet nodes without any delay and problem suggest the ability of using the system in a network that require high level of security. Furthermore, the use of AES with the length and the amount of the keys used will guarantee that there is really a huge amount of work from the cryptanalyst to guess the keys using brute force attack or other types of attacks.

Finally, two suggestions can be proposed to enhance this system. The first is the inclusion of expert system in the scheme that can select the complexity of padding according to the security level, compression technique, key size of triple AES or the signature requirement that will support more secure system implementation. The second suggestion will be the replacement of the RSA system by the elliptic curve cryptosystem and the implementation of the system in hardware using the FPGA.

References

1. W. Stallings, “Cryptography and Network Security Principles and Practice”, Prentice Hall, 2003.
2. A. S. Tanenbaum, “Computer Networks”, Mc-Graw Hill, 2003.
3. A. Lee, “Guideline for Implementing Cryptography in the Federal Government", National Institute of Standards and Technology (NIST) Special Publication, November 1999.
4. P. Barreto, H. Kim, B. Lynn, and M. Scott, “Client Algorithms for Pairing-Based Cryptosystems”, Advances in Cryptology-Proceedings of CRYPTO2002, Springer-Verlag, 2002.
5. M. Welschenbach, “Cryptography in C and C++”, New York, USA, 2001.
6. B. Schneider, “Applied Cryptography”, K. Schowalter, 1996.
7. S. Roman, ”Introduction to Coding and Information Theory”, Springer-Veraig, 1997.
8. H. Beker, and F. Piper, ”Cipher Systems”, London, Northward Books, 1982.

9. D. Wagner, “Cryptanalysis of a Provably Secure CRTRSA Algorithm”, ACM CCS2003, ACM Press, Oct. 2004.

10. J. Baek and Y. Zheng, “Simple and Client Threshold Cryptosystem from the Gap Diffe-Hellman Group”, Proceedings of IEEE GLOBECOM Conference, Communication Security, SC04-7, IEEE, 2003.