Data Encryption of Audio File by using AES and RSA Algorithms

Ngu War Win; Dr. Soe Soe Aye
University of Computer Studies, Kyainge Tong
nguwar8002@gmail.com, soesoeaye74@gmail.com

Abstract

In this age of universal electronic connectivity is needed to protect data and resources from disclosure, to guarantee the authenticity of data and messages, and to protect systems from network-based attacks. The disciplines of cryptography and network security have matured, leading to the development of practical, readily available applications to enforce network security. The cryptography is one of the ways to be data secure.

This paper intends to design a security protocol using hybrid encryption technique for wave files. The hybrid encryption technique is a combination of both symmetric and asymmetric cryptographic techniques. The encryption algorithms are more secured depends on the key value and its size. But, the key distribution is major problem. The various protocols are currently given the solution. The new protocol solves the key management problem using key servers. This paper presents the encryption of wave file by AES algorithm and RSA algorithm is used to secure the AES key.

1. Introduction

The requirements of information security within an organization have undergone two major changes in the last several decades. Before the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means. With the introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident.

The second major change that affected security is the introduction of distributed systems and the use of networks and communications facilities for carrying data between terminal user and computer and between computer and computer. Network security measures are needed to protect data during their transmission. According to the data communication is an important aspect of our living. So, protection of data from misuse is essential.

At present, the various types of cryptographic algorithms provide high security in information, computer and network-related activities. These algorithms are needs to protect the data, integrity and authenticity from various attacks. This paper provides the design of new protocol for better security using key server with hybrid encryption technique.

The organization of this paper is as follows: Section 2 presents the related work of the system. Section 3 describes Cryptography, Symmetric algorithm, Asymmetric algorithm and Hybrid Approach. In Section 4 proposed system has been illustrated along with AES algorithm and RSA algorithm. Section 5 is about system implementation and Section 6 is the conclusion of the system.

2. Related Work

The cryptographic algorithms are classified into two different types such as symmetric and asymmetric method. In symmetric encryption method both sender and receiver share the common key value for encryption and decryption. It requires that the sender find some secure way to deliver the encryption/decryption key to the receiver. The effective key distribution needs to deliver key to the receiver [8].

In [7][8], the authors described about the key distribution difficulties. Large number of protocols provides various techniques. These protocols are to provide more secure but less performance. The public key cryptography or asymmetric cryptographic method solves the problems of key distribution. In this method, uses a pair of keys for encryption. The public key encrypts the data and corresponding private key for decryption. Each user has one pair of keys. The private key kept secret and public key knows by others. Any one wants to send some information to you they read your public key and encrypts the information. After you receive, the encrypted data using your private key to decrypt it. One issue with public key cryptosystems is that users

must be constantly vigilant to ensure that they are encrypting to the correct person's key. In a public key environment you are assured that the public keys to which you are encrypting data is in fact the public key of the intended receiver. The identification of correct public key of proper person is more difficult without using any third party. Everyone knows the cryptographic algorithms functionality. The sender sends his data using any one cryptographic algorithm with key value. The key value is more confidential. The key management is also more complex.

3. Cryptography

The goal of cryptography is to make it possible for two people to exchange a message in such a way that other people cannot understand the message. There is no end to the number of ways this can be done, but here it will be concerned with methods of altering the text in such a way that the recipient can undo the alteration and discover the original text.

The original text is usually called "cleartext" and the encoded or altered text is called "ciphertext". The conversion from cleartext to ciphertext is called "encoding" or "enciphering", and the opposite operation is called "decoding" or "deciphering". If one is trying to read a secret message that was not intended for him and he initially does not know the encoding method, it is called "cracking" the code.

3.1. Symmetric Algorithm

In order to encrypt a source file, the encryption algorithm uses a special number known as the key. The value of this key modifies the detailed operation of the algorithm; that is, the way the contents of the original file will be "scrambled up." This means that if the same file is encrypted using two different keys, the results will be totally dissimilar. Algorithms and keys are created in such a way as to make "cracking the code" by unauthorized parties as difficult as possible. The point is that, in order to open the encrypted file and access its data, the end user also requires access to an appropriate key.

In the symmetric algorithm, the same key is used to encrypt and decrypt the file as shown in figure 1. The advantage of this technique is speed due to its relatively low computational requirements. Using a modern computer, encrypting even a large file using a symmetric algorithm takes only seconds, and the time taken to decrypt the file when it is accessed by an application such as synthesis is unperceivable to the user. The main problem of the symmetric algorithm is the key distribution problem.

3.2. Asymmetric Algorithm

The "asymmetric" appellation is applied because the key used to decode the data is different to the key used to encode it. Asymmetric schemes are also commonly known as public key encryption, because they rely on the use of two keys: a public key and a private key. The idea here is that the public key, which is the product of two prime numbers, is made available to everyone (or at least, to everyone who needs to know about it). This public key is used for encryption, but it cannot be used to decrypt the ensuing file; decryption requires access to the private key, which is one of the prime numbers used to create the public key. The main problem of the RSA algorithm is the performance since it involves the factorization of large integers.

3.3. Hybrid Approach

The solution to the problem of above algorithms is to employ a hybrid symmetric-asymmetric encryption/decryption flow. In the hybrid approach, the two different approaches are used in a complementary manner, with each performing a different function. A symmetric algorithm creates keys that are used for encrypting bulk data and an asymmetric algorithm creates keys that are used for automated key distribution.

First the data is encrypted using an internally generated symmetric key. In this system, those keys will be referred to as the data key. This form of encryption is extremely fast, even on large blocks of data. The result from this step is known as the data block. In this system AES algorithm will be used.

Next the data key is encrypted using the RSA algorithm (public key algorithm). The result is known as a key block. The process of this approach is shown in Figure 1.

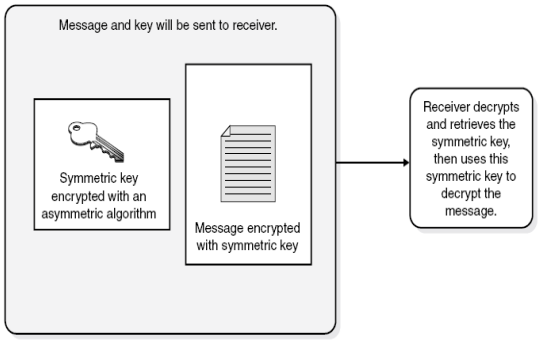

Figure 1: Encryption by Hybrid Approach:

The main thing of Encryption by Hybrid approach is as follows:

1. data is encrypted using symmetric key
2. data key is encrypted using asymmetric key
3. the same encrypted data is sent to end users

4. Proposed System

This system presents the encryption of wave files using AES algorithm. In order to prevent the key distribution problem, RSA algorithm is used to encrypt the AES key. Process flow of the system is as follows:

- It reads Wave file, and AES key is entered into the system.
- Wave Files are processed into byte values in order to encrypt with AES algorithm.
- Then wave data is encrypted using AES algorithm.
- In order to prevent the key distribution problem, AES key is encrypted with RSA algorithm.

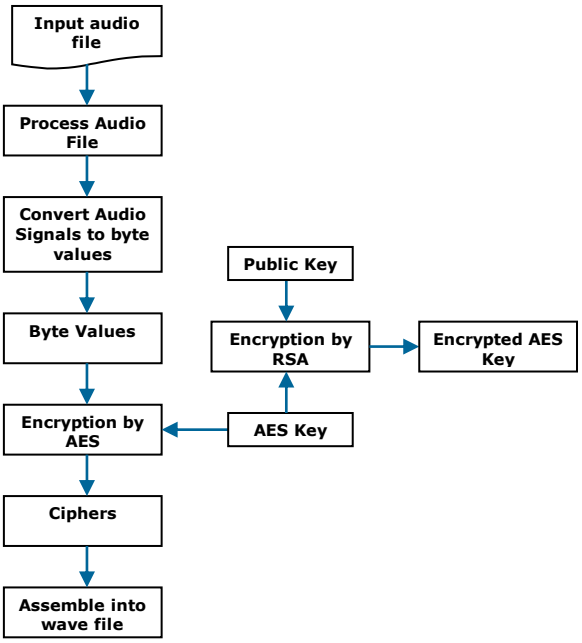

Figure 2: Process flow of Encryption Process

4.1 AES Algorithm

AES algorithm is flexible in supporting any combination of data and key size of 128, 192 and 256 bits. However, AES merely allows a 128 bit data length that can be divided into four basic operation blocks. These blocks operate on array of bytes and organized as a 4×4 matrix that is called the state. For full encryption, the data is passed through Nr rounds (Nr = 10, 12, 14). These rounds are governed by the following transformations:

- Bytesub transformation: Is a non linear byte Substitution, using a substation table (sbox), which is constructed by multiplicative inverse and affine transformation.
- Shiftrows transformation: Is a simple byte transposition, the bytes in the last three rows of the state are cyclically shifted; the offset of the left shift varies from one to three bytes.
- Mixcolumns transformation: Is equivalent to a matrix multiplication of columns of the states. Each column vector is multiplied by a fixed matrix. It should be noted that the bytes are treated as polynomials rather than numbers.
- Addroundkey transformation: Is a simple XOR between the working state and the roundkey. This transformation is its own inverse.

Encryption by AES Algorithm

The encryption procedure consists of several steps as shown by figure. 1. After an initial addroundkey, a round function is applied to the data block (consisting of bytesub, shiftrows, mixcolumns and addroundkey transformation, respectively). It is performed iteratively (times) depending on the key length. The decryption structure has exactly the same sequence of transformations as the one in the encryption structure. The transformations Inv-Bytesub, the Inv-Shiftrows, the Inv-Mixcolumns, and the Addroundkey allow the form of the key schedules to be identical for encryption and decryption. Figure 3 presents the Encryption procedure of AES algorithm.

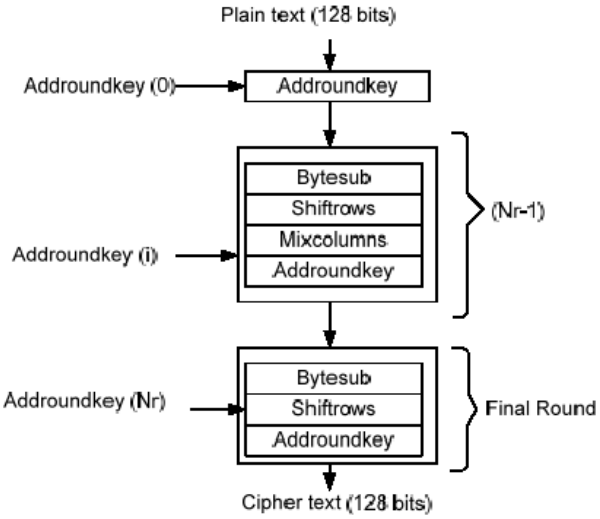

Figure 3: Encryption structure of AES algorithm

4.2. RSA Algorithm

RSA are the initials of the three creators: "Rivest, Shamir and Adleman". It is based on the following idea: It is very simply to multiply numbers together, especially with computers. But it can be very difficult to factor numbers. For example, if it is to multiply together 34537 and 99991, that is a simple matter to punch those numbers into a calculator and 453389167. But the reverse problem is much harder.

Suppose if there is a number 1459160519. This is got by multiplying together two integers. This is a very difficult problem to tell what they are. A

computer can factor that number fairly quickly, but (although there are some tricks) it basically does it by trying most of the possible combinations. For any size number, the computer has to check something that is of the order of the size of the square-root of the number to be factored. In this case, that square-root is roughly 38000.

Key Generation

Public key and private key can be generated in following ways:

1. Choose two large prime numbers. In mathematics, a prime number or prime for short, is a natural number whose only distinct positive divisors are 1 and itself; otherwise it is called a composite number. Hence a prime number has exactly two divisors. The number 1 is neither prime nor com p ≠ q randomly and independently of each other. Compute N = p q.
2. Choose an integer 1 < e < N which is coprime to (p-1)(q-1).
3. Compute d such that d e ≡ 1 (mod (p-1)(q-1)).

(Steps 2 and 3 can be performed with the extended Euclidean algorithm)
(Step 3, rewritten, can also be found by finding integer x which causes d = (x(p-1)(q-1) + 1)/e to be an integer, then using the value of d (mod (p-1)(q-1)).

RSA public-key encryption

SUMMARY: B encrypts a message m for A, which A decrypts.
1. ***Encryption***. B should do the following:
(a) Obtain A's authentic public key (n; e).
(b) Represent the message as an integer m in the interval [0; n − 1].
(c) Compute c = me mod n (e.g., using Algorithm 2.143).
(d) Send the ciphertext c to A.
2. ***Decryption***. To recover plaintext m from c, A should do the following:
(a) Use the private key d to recover m = cd mod n.

5. System Implementation

This system is implemented using Microsoft Visual Studio 2008. C# .Net is used to develop the system. The main data to encrypt in this system is wave file.

5.1. Wave File Processing

WAV files have two basic parts, the header and the data. The data is one giant chunk of bytes that represents the audio. This system has to read the header so that it can understand how to interpret the data. In normal wav files, the header is the first 44 bytes of the file. Everything about the file is contained within those first 44 bytes. Header format of wave file is shown in the Table 1.

Table 1 . Audio format of wave file

Position – byte	Field Name	Field Size	Description
0	Chunk ID	4	This should just contains "RIFF"
4	File Size	4	The size of the rest of the file after this field. Entire File Size – 8
8	File Format	4	It contains "WAVE"
12	Sub Chunk1 ID	4	It contains "fmt"
16	Sub Chunk1 Size	4	
20	Audio Format	2	1 for uncompressed audio. If other values other than 1, audio file is compressed.
22	Number of channels	2	Either 1, 2 or other positive values
24	Sample Rate	4	44100
28	Byte Rate	4	Sample Rate * Number of Channels * BitsperSample / 8
32	Block Alignment	2	Channels * BitsperSample / 8
36	Sub Chunk2 ID	4	It contains "data"
40	Data Size	4	Number of bytes following the header. Size of data

In this system, data following header (in wave file, starts from index 44) is used to encrypt. In creating the encrypted wave file, the same header value is used, and in the place of data, encrypted data is replaced. This file becomes the encrypted wave file. Encrypted wave file can be played with Sound Players (such as Windows Media Player), but producing meaningless sounds.

5.2. Experimental Result

This system is tested with various wave files, for example music, speech, etc. We have collected wave files from different news channel, such as BBC news, CBS news and different types of music, such as classic, pop, blues, etc. We have found that the wave file is totally encrypted and key distribution is also solved. Figure 4 (a) presents the frequency of original wave file (6 minute bbc news) and Figure 4 (b) is the frequency analysis of encrypted wave file of that news.

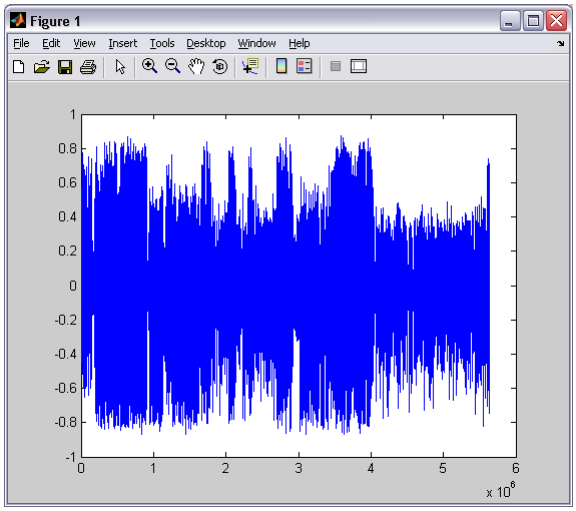

Figure 4 (a): Frequency of 6 minute BBC news

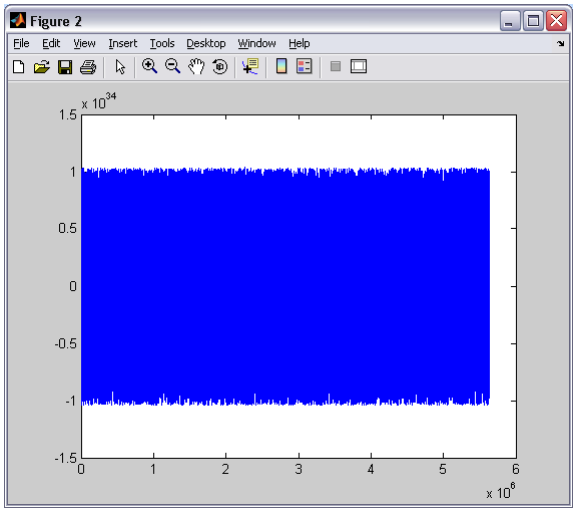

Figure 4 (b): Frequency of encrypted 6 minute BBC news

In Hybrid Encryption, a key is agreed between reciver and the sender. This key can be dynamic or can even be static. Sender has the public key of receiver and receiver has private key to decrypt. Secret key is encrypted using RSA algorithm by receiver's public key. Then it is decrypted with receiver's private key. Secret key is disitributed in such a secured way. Secret key is chosen in different length 16, 24 or 32 byte values. The main advantage of this method is that it takes much lesser time when compared to normal encryption with secure key transformation process.

6. Conclusion

This paper presents secure key management system for encrypting wave files using hybrid approach. This hybrid encryption method increases the performance of cryptographic algorithms. It ensures the confidentiality and authentication. The AES algorithm provides confidentiality and RSA provides integrity and authentication.

7. References

[1] Chein HY, Jan JK, Tseng YM. (2002), “An efficient and practical to remote authentication: Smart Card Security”, ELSEVIER-Computers & Security Journal, 21(4), pp. 372-375.

[2] Eun-Jun Yoon, Eun-Kyung Ryu, Kee-Young Yoo (2005), “An improvement of Hwang-Lee-Tang’s simple remote user authentication scheme”, ELSEVIERComputers & Security Journal, 24(1), pp. 50-56.

[3] Hwang MS, Lf LH. (2000), “A new remote user authentication scheme using Smart Cards”, IEEE Transactions, 46(1). pp. 110-120

[4] James Nechvatal, Elaine Barker and Lawrence Bassham, “Report on the Development of the Advanced Encryption Standard (AES)”, Computer and Security Division, National Institute of Standards and Technology (NIST), US Dept. of Commerce.

[5] Mayer R. Thompson, Abdelilah and Srilekha Mudumbai (2003), “Certificate-Based Authorization Policy in a PKI Environment”, ACM Transactions on Information and System Security, Vol. 6; No. 4, 566-588

[6] Srdjan Capkun, Levente Buttyan and Jean-Pierre Hubaux (2003), “Self organized public key management for mobile ad-hoc networks”, IEEE Transactions, 2(1), pp. 51-63.

[7] Tianjie Cao, Dongdai Lin and Rui Xue (2005), "A randomized RSA-based partially blind signature scheme for electronic cash", ELSEVIER-Computers & Security Journal, 24(1), pp. 44-49.

[8] William Stallings (2003), Cryptography and Network Security-Principles and Practices, 3rd Edition, Pearson Education Asia.

[9] Wu ST, Chieu BC (2003), "A user friendly remote authentication scheme with smart cards:, ELSEVIER-Computers & Security Journal, 22(6), pp. 547-597.

[10] Yang WH, Shieh SP (1999), "Password Authentication Schemes with Smart Card", ELSEVIER-Computers & Security Journal, 18(8), pp. 727-760.