International Journal of Research in Engineering, Science and Management
Volume 4, Issue 3, March 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Security in Ad-hoc Network Using Encrypted Data Transmission and Steganography

Ankita Patil[1*], Dhiraj Kuslekar[2], Ajay Jha[3]
[1,2,3]*Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, New Mumbai, India*

***Abstract*: Currently, there has been an increasing trend in outsourcing data to remote cloud, where the people outsource their data at Cloud Service Provider(CSP) who offers huge storage space with low cost. Thus users can reduce the maintenance and burden of local data storage. Meanwhile, once data goes into cloud they lose control of their data, which inevitably brings new security risks toward integrity and confidentiality. Hence, efficient and effective methods are needed to ensure the data integrity and confidentiality of outsource data on untrusted cloud servers. The previously proposed protocols fail to provide strong security assurance to the users. In this paper, we propose an efficient and secure protocol to address these issues. Our method allows third party auditor to periodically verify the data integrity stored at CSP without retrieving original data. To compare with existing schemes, our scheme is more secure and efficient.**

***Keywords*: Least significant bit, Steganography, Cryptography, Cloud service provider, Elliptical curve cryptography.**

1. Introduction

With the invention of Storage, the days of keeping all your documents, photos, music files etc. on your computer's hardware are gradually coming to a close. Today, the storage is fulfilling the need for more storage space to hold all of your digital data. Storage space providers operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them. The data center operators, in the background, virtualizes the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers. Storage can be used from smaller computing devices to desktop computers and servers. Storage services may be accessed through a web service API or through a Web-based user interface. The storage architectures build a single virtual storage system. The data when stored on Storage space has the following threats:

1. When data is distributed, it is stored at multiple locations increasing the risk of unauthorized physical access to the data.
2. The number of people with access to the data who could be compromised (i.e. bribed or coerced) increases dramatically.
3. It increases the number of networks over which the data travels. Instead of just a local area network (LAN) or storage area network (SAN), data stored on a Storage space requires a WAN (wide area network) to connect them both.
4. Sharing of storage and networks with many other users/customers it is possible for other customers to access your data.

To secure data, most systems use a combination of techniques, including:

1. Encryption, which means they use a complex algorithm to encode information. To decode the encrypted files, a user needs an encryption key. While it's possible to crack encrypted information, most hackers don't have access to the amount of computer processing power they would need to decrypt information.
2. Authentication processes, which require creating a user name and password.
3. Authorization practices -- the client lists the people who are authorized to access information stored on the storage system. Many corporations have multiple levels of authorization. For example, a front-line employee might have very limited access to data stored on a storage system, while the head of human resources might have extensive access to files. Storage approach poses a potential security threat to your data and moreover, only the password access to storage is not sufficient as the password can be hacked by an intruder. Also the data can be captured en-route to the storage services. The need to access storage on thin clients and mobile devices is becoming an emerging application. But due to smaller processor speed and run time memory; these devices need an algorithm which can be used in such small computing devices. Security of stored data and data in transit may be a concern when storing sensitive data at a storage space provider.

2. Related Work

RSA algorithm is the most widely used public key cryptography algorithm for encryption and decryption by many vendors today. This is the first generation algorithm that was used for providing data security. It can be used to encrypt a message without the need to exchange a secret key separately. The RSA algorithm can be used for both public key encryption

*Corresponding author: anki42111@gmail.com

and digital signatures. Its security is based on the difficulty of factoring large integers. Party A can send an encrypted message to party B without any prior exchange of and B decrypts it using the private key, which only he knows. RSA can also be used to sign a message, so A can sign a message using their private key and B can verify it using A's public key.

Encryption of a message, m, involves exponentiation, c = me mod n, which requires a lot of mathematical computations. In RSA cryptosystem, user A (say Alice) picks up two large primes p and q and computes their product, n = p*q. Now Alice's public key is a pair of integers {n, e} and the private key is d.

Key Generation:

INPUT: Security parameter l.

OUTPUT: RSA public key (n, e) and private key d.

1. Select two primes p and q of the same bit length l/2.
2. Compute n = p*q and φ = (p−1)*(q −1).
3. Select arbitrary integer e with 1 < e <φ and gcd(e, φ) = 1.
4. Compute integer d satisfying 1 < d <φ and e*d ≡ 1 (mod φ).
5. Return (n, e, d).

Encryption:

INPUT: RSA public key (n, e), plaintext m ∈ [0, n−1].

OUTPUT: Cipher text c.

1. Compute c = me mod n.
2. Return(c).

Decryption:

INPUT: Public key (n, e), private key d, cipher text c.

OUTPUT: Plaintext m.

1. Compute m = cd mod n.
2. Return (m).

3. Literature Survey

Cloud storage is a model of networked enterprise storage where data is stored not only in the user's computer, but in virtualized pools of storage which are generally hosted by third parties. Ravi Shankar Dhakar et al talk about the "Modified RSA Encryption Algorithm (MREA)" where the talk about factorization in RSA cryptosystem, and their implementation compares the existing system and their system with key sizes up to 1024 bit. The authors claim their system to be better than existing system for the brute-force attack. Suli Wang et al talk about the "File encryption and decryption system based on RSA algorithm" where theyused RSA for encryption and decryption of files with smaller sizes. Maryam Savari et al in "Comparison of ECC and RSA Algorithm in Multipurpose Smart Card Application" compare the security of RSA 1024-bit key versus ECC 160- bit key sizes. P.R. Vijayalakshmi et al in "Performance Analysis of RSA and ECC in Identity-Based Authenticated New Multiparty Key Agreement Protocol" compare ECC algorithm with 128 bits with that of RSA algorithm with 1024 bits' key size. Kamlesh Gupta et al in "ECC over RSA for Asymmetric Encryption: A Review" demonstrated the use ECC for portable devices and applications. Arjun Kumar et al propose a method that allows user to store and access the data securely from the cloud storage in "Secure Storage and Access of Data in Cloud Computing". Xiao Zhang et al talk about the physical security of data in data centers "Ensure Data Security in Cloud Storage". Somani, U et al proposed implementation digital signature with RSA algorithm to enhance data security in cloud storage. Chakraborty, T.K et al proposed a model for data security in cloud. Over last 10 years, a great deal of work has taken place to ensure that ECC meets these goals and is specified in an ever-increasing number of standards. It started with the IEEE P1363 in 1994 (becoming a standard in 2000), and now includes many accredited standards organizations:

i. ISO (in ISO 14888-3: ECDSA and other ECC-based signature schemes)
ii. IEEE (in IEEE 1363-2000 for public-key cryptography)
iii. The American National Standards Institute (in ANSI X9: cryptography for financial-services industry).

NIST also specifies ECC in FIPS 186-2: Federal Information Processing Standards ECDSA and SP 800-56: Special Publication on Key management. While in Europe, BSI in Germany also specifies ECC. Though there are several papers published on the comparison of ECC and RSA in terms of key sizes and security, this paper talks about the reduced key generation time, comparison of encrypted file (cipher text) cloud storage. The simulation system provides the key sizes of up to 15360 bits. In this work the used file sizes are up to 40 MB for the simulation i.e. encryption and decryption. This work compares the security of ECC in the key range of 160 - 512 bits and RSA key sizes ranging from 512 - 3072 bits. The simulation experiments compare the ECC and RSA at different levels of key sizes and block sizes.

4. Problem Statement

The problem statements are as follows:

User-Id and password are not covered in some applications. As a consequence, anyone who is interested in using the programme can do so. Sending a plain text of data to the receiver is not secure. Since the data is readable, everyone can access it. Even if the message is encoded before being sent, the hacker will decode it with the help of a certain algorithm. It's possible that the systems aren't properly linked at times. As a consequence, the data being transmitted could not arrive in the correct format at its destination. Maintaining software stability is extremely difficult. The reliability comes at a price. When the message falls into the hands of a hacker, the hacker has the ability to insert, erase, or change the original message's content. If the message is not adequately secured, the message's confidentiality is lost. A hacker can impersonate the sender and lead the recipient astray. As a result, a hacker can get around the authentication process. In certain cases, hackers are unable to obtain the original message's content. Thus they perform the exponential attack. The hacker loses the content of the original message in an exponential assault.

5. Methodology

User interface, Embed Module, Retrieve Module, Sender Module, Receiver Module, and Help Module are the six major

modules in this project. These modules will be built separately first, and then all of them will be combined.

1) User Interface Module (Steganograph)

This module will essentially provide the key form for accessing the application's features. This module will function as a parent form for other child forms since the application will be implemented as an MDI parent child property for user interface.

2) Embed Module

This module will essentially allow you to insert a message as well as a text or data file into an image, audio, or video file. It will also have a sub module called "Encrypt" that will encrypt both the message and the data file. This encrypt module will be used by the embed module to encrypt data. This module will take care of the Embed File form's backend tasks.

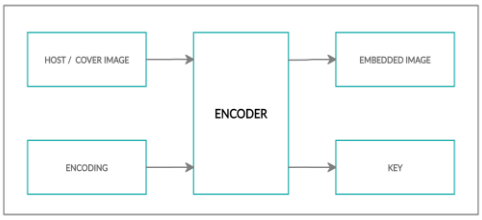

Fig. 1. Watermark embedding

3) Retrieve module

This module will essentially have the ability to retrieve the message as well as the text or data file from image, audio, and video files. It will also have a sub module called "Decrypt" that will be in charge of decrypting the message and data file. This decrypt module will be used by the retrieve module to decrypt data. The backend role for the Retrieve File type will be handled by this module.

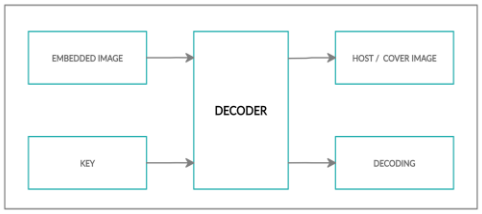

Fig. 2. Watermark detection

4) Sender module

This module would essentially have the ability to transfer files from one computer to another. For sending the file to other computers, this module will use socket programming. This module will also have a child type user interface (Send File).

5) Receiver module

This module will primarily act as a receiver for files received from other machines. Socket programming can also be used in this module to receive files from other computers. There will be no user interface for this module. It will run quietly in the background.

6) Help module

This module will be in charge of providing the application with support. Both JAVA and HTML will be used to implement it. This module's type will be created in JAVA, while the material for the support will be created in HTML.

6. Guidelines

A. Overview of System Architecture

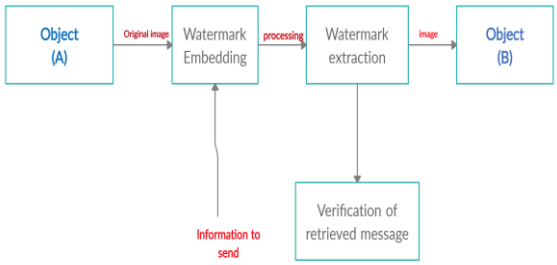

Fig. 3. Architecture diagram

B. Hardware requirement

1. Display drive that should support 32-bit color scheme.
2. Display resolution that should be 1024 x 768 pixels.
3. Graphics Drive that can support 800 x 600 display resolution.
4. Minimum of 128 MB RAM is required.
5. The processor preferably should be Pentium III or above /its equivalent.

C. Software requirement

1. JDK 1.5
2. Any version of Windows, Macintosh, UNIX and Solaris.

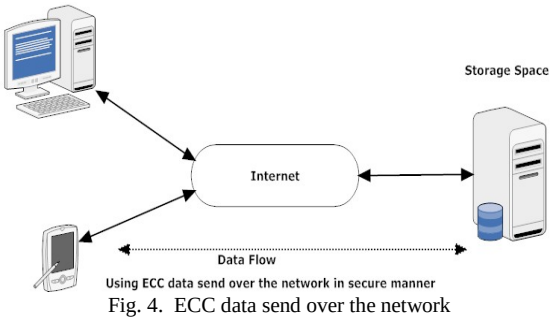

Fig. 4. ECC data send over the network

D. Cryptography

Cryptography can be used to provide message confidentiality and integrity and sender verification. The basic functions of cryptography are encryption, decryption and cryptographic hashing. In order to encrypt and decrypt messages, the sender and recipient need to share a secret. Typically, this is a key, like a password, that is used by the cryptographic algorithm. The key is used by the sender to encrypt the message (transform it into cipher text) and by the recipient to decrypt the message (reverse the cipher text back to clear text). This process can be done on a fixed message, such as an e-mail, or a communications stream, such as a TCP/IP connection. Cryptographic hashing is the process of generating a fixed-length string from a message of arbitrary length. If the sender provides a cryptographic hash with the message, the recipient can verify its integrity. Modern cryptographic systems are based on complex mathematical relationships and processes.

Let's focus on the common cryptography standards used to secure computer communications and how they are used.

The three basic types of cryptography in common use are symmetric key, asymmetric (public) key systems and cryptographic hash functions. Typically, the strength of a crypto system is directly related to the length of the key. This assumes that there is no inherent weakness in the algorithm and that the keys are chosen in a way that fully utilizes the key space (the number of possible keys). There are many kinds of attacks that can be used against crypto systems, but these are beyond our scope here. That said, if you use public algorithms with no known vulnerabilities, use reasonable key lengths and choose good keys (which are normally chosen for you), your communications will be very secure.

The elements that are essential in cryptosystems are as follows:

1. Plain text (input)
2. Encryption algorithm
3. Secret key
4. Cipher text
5. Decryption algorithm

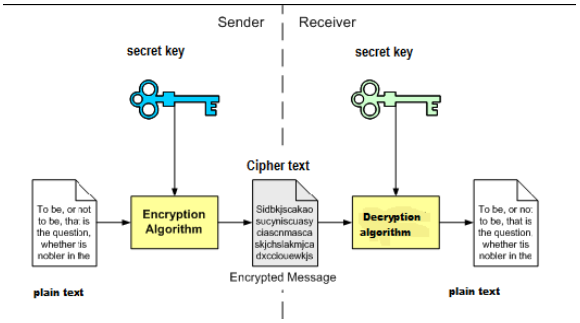

Fig. 5. General model of cryptographic system

Plain text: The original piece of data required to submit data to the intended recipient. Plain Text is the name of the encryption algorithm. Any cryptographic system's main key is known as it. The plain text is subjected to different substitutions and transformations in this encryption algorithm. The secret key is used as an input to the encryption algorithm that the user specifies.

Various substitutions and transformations on the plain text can differ based on this key. The performance of the encryption algorithm is called cypher text. The jumbled text is the cypher text. Each secret key that has been given to the encryption algorithm results in a different cypher text. The "decryption algorithm" is the inverse of the "encryption algorithm." It will take in cypher text and the secret key as input and output plain text.

7. Analysis

A. Basics if Steganography

Steganography aims to hiding information in a cover data in such a way that non-participating persons are not able to detect the presence of this information by analysing the information detection. Unlike watermarking, steganography does not intended to prevent the hidden information by opponents of removing or changing the hidden message, which is embedded in the cover data but it emphasizes on remains it undetectable. Steganography is particularly interesting for applications in which the encryption cannot used to protect the communication of confidential information.

B. Image Steganography

Hiding the data by taking the cover object as image is referred as image steganography. In image steganography pixel intensities are used to hide the data. In digital steganography, images are widely used cover source because there are number of bits presents in digital representation of an image.

C. Audio Steganography

In audio steganography, secret message is embedded into digitized audio signal which result slight altering of binary sequence of the corresponding audio file. There are several methods are available for audio steganography. We are going to have a brief introduction on some of them. It involves hiding data in audio files. This method hides the data in WAV, AU and MP3 sound files. There are different methods of audio steganography. These methods are

i) Low Bit Encoding
ii) Phase Coding
iii) Spread Spectrum.

D. Video Steganography

It is a technique of hiding any kind of files or data into digital video format. In this case video (combination of pictures) is used as carrier for hiding the data. Generally discrete cosine transform (DCT) alter the values (e.g., 8.667 to 9) which is used to hide the data in each of the images in the video, which is unnoticeable by the human eye. H.264, Mp4, MPEG, AVI are the formats used by video steganography.

In all of these methods, the basic principle of steganography is that a secret message is to be embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So it cannot be detected easily to be containing hidden information unless proper decryption is used.

E. Functional Analysis

1. Login: Login function will authenticate the sender if username and password are correct otherwise it will exit the system.
2. Secret Text Message File: In this file you will have to write secret message to hide or you can select any text file of secret message.
3. Cover Object: Cover Object is the object which is to be selected in which secret text message can be hidden.
4. Stego Encryption LSB implementation is performed on cover object to hide secret text message by replacing bits of cover object by the bits of message.
5. Sender: In this Sender send this stego object file to intended recipient to which he does want to communicate.
6. Receiver: In this receiver receives the stego object and opens in decryption option for getting hidden text

message inside that image.

F. Non- Functional Analysis

1. *Safety requirements:* Sender and Receiver should make sure that only they are having the same software to encrypt and decrypt data inside image. Both should take care of eavesdropping.
2. *Security requirements:* We are going to develop a software in which embedding secret text data in object (Image, Audio, Video). Only sender and receiver should be aware of encrypted file. User should not unfold the message regarding sent image as well as receiver information.
3. *Software quality attributes:* The Quality of the software is maintained in such a way that only sender and receiver can communicate through image, video, audio. There is no probability of knowing secret object.

G. Steganography with LSB Algorithm

Bytes of pixels are sufficient to hold one message byte. Rest of the bits in the pixels remains the same. Steganography is the art and science of communicating in a way which hides the existence of the communication. Steganography plays an important role in information security It is the art of invisible communication by concealing information inside other information. The term steganography is derived from Greek and literally means covered writing. A Steganography system consists of three elements: cover image (which hides the secret message), the secret message and the stegano-image (which is the cover object with message embedded inside it). A digital image is described using a 2-D matrix of the colour intestines at each grid point (i.e. pixel). Typically, gray images use 8 bits, whereas coloured utilizes 24 bits to describe the colour model, such as RGB model. The Steganography system which uses an image as the cover, there are several techniques to conceal information inside cover-image. The spatial domain techniques manipulate the cover-image pixel bit values to embed the secret information. The secret bits are written directly to the cover image pixel bytes. Consequently, the spatial domain techniques are simple and easy to implement. The Least Significant Bit (LSB) is one of the main techniques in spatial domain image Steganography.

The concept of LSB Embedding is simple. It exploits the fact that the level of precision in many image formats is far greater than that perceivable by average human vision. Therefore, an altered image with slight variations in its colours will be indistinguishable from the original by a human being, just by looking at it. In conventional LSB technique, which requires eight bytes of pixels to store 1byte of secret data but in proposed LSB technique.

8. Conclusion

Elliptic Curve Cryptography is more safe and reliable than first-generation public key techniques like RSA, which are currently in use. When it comes to upgrading their systems, vendors should seriously consider the elliptic curve option because of the computational and bandwidth benefits it provides while maintaining comparable security. Although the security of ECC has not been fully assessed, it is expected to be widely used in the future in a variety of fields. When comparing the RSA and ECC cyphers, it was discovered that ECC has significantly lower overheads than RSA. Since it can have the same degree of protection as RSA by using shorter keys, the ECC has a lot of advantages. However, one flaw that could obscure its appeal is its lack of maturity, as mathematicians conclude that not enough research has been done in ECC. Since today's applications (smart cards, pagers, and cellular telephones, for example) cannot bear the overheads introduced by RSA, ECC appears to have a better future than RSA. ECC can be used for encryption and decryption in today's small computing devices because it needs smaller key sizes and has less computing complexity than RSA. As a result, ECC is an excellent option for compact, mobile, and low-power applications, as well as cloud integration. The time taken by the two algorithms for key generation and encryption is compared in this paper. The significance of this research is that it demonstrates the use of the ECC algorithm in cloud storage, which provides improved protection. This research can be expanded to equate ECC with other algorithms for digital signatures, key exchanges, and data integrity.

References

[1] S. Swapnil and D. B. Megherbi, "A Robust Double-Blind Secure High Capacity Watermarking and Information Hiding Scheme for Authentication and Tampering Recovery Via the Wavelet and Arnold Transforms," 2018 IEEE International Symposium on Technologies for Homeland Security (HST), Woburn, MA, USA, 2018, pp. 1-5.

[2] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," Proceedings of International Conference on Image Processing, Santa Barbara, CA, USA, 1997, pp. 680-683, vol. 2.

[3] Elliptic curve cryptography, https://en.wikipedia.org/wiki/Elliptic_curve_cryptography

[4] RSA (algorithm), http://en.wikipedia.org/wiki/RSA_(algorithm)

[5] JavaTM Cryptography Extension (JCE), Reference Guide. http://docs.oracle.com/javase/1.5.0/docs/guide/security/jce/JCERefGuide.html