Hybrid design for cloud data security using a combination of AES, ECC and LSB steganography

Article *in* International Journal of Computational Science and Engineering · January 2018

DOI: 10.1504/IJCSE.2018.10016054

2 authors, including:

Osama Hosameldeen
Higher College of Technology
47 PUBLICATIONS **404** CITATIONS

SEE PROFILE

Hybrid Design for Cloud Data Security Using Combination of AES, ECC and LSB-Steganography

Osama Hosam

The College of Computer Science and Engineering in Yanbu, Taibah University, Yanbu Al Bahar, Saudi Arabia.

Informatics Research Institute, The City for Scientific Research and Technology Applications, Alexandria, Egypt.

Email: mohandesosama@yahoo.com

Muhammad Hammad Ahmed

Computer Engineering Dept, University of Engineering and Technology, Taxila, Pakistan

Email: hammadhum@hotmail.com

Abstract: The ever-growing popularity of cloud systems is embarking a revolutionary change in information technology field. Parallel and flexible services offered by cloud technology are making it the ultimate solution for individuals as well as for organizations of all size. The grave security concerns present in cloud must be addressed to protect the data and privacy of huge number of cloud users. We present a hybrid solution to tackle the key management problem. The data in the cloud is encrypted with AES encryption with private key. The AES 256-bits key is then encrypted with ECC. The ECC encrypted key will be embedded in the user's image with LSB steganography. If the user decided to share cloud data with a second user, he only need to embed the AES key in the second user's image. Using Steganography, ECC and AES we can achieve strong security posture and efficient key management and distribution for multiple users.

Keywords: Cloud security, Encryption, ECC, AES, Steganography, Public Key, Private Key, HLSB steganography.

Reference to this paper should be made as follows: Osama Hosam, Ahmed, M. H. (2018) 'Hybrid Design for Cloud Security Using Combination of AES, ECC and Steganography', *Int. J. of Computational Science and Engineering*, Vol. X, No. Y4, pp.000–000.

Biographical notes: (Osama Hosam) Is a research associate in SRTA-City, Alexandria, Egypt. In 2007 he received his MSc. In computer systems and engineering from Azhar University, He pursued his PhD study in Hunan University, China and worked in parallel in Nanjing University of Technology; in 2011 he received his PhD in Computer Science and Engineering. In 2013 he worked as an Assistant Professor in at the Collage of Computer Science and Engineering in Yanbu. In 2017 he is promoted to be an Associate Professor in the field of Computer and Information Security, Taibah University. His research interests include, Computer Graphics, 3D Watermarking, Stereo Vision, Pattern Recognition and Computer & Information security.

(Mohammad Hammad Ahmad) is an information security researcher in Comspots, Saudi Arabia and has more than 10 years of information security experience. In 2012 he received his MSc degree in computer engineering from University of Engineering and Technology, Taxila, Pakistan. He holds CISSP and CISM certifications. He has developed various information security products for various government and private sector organisations. His research interests include cryptography, information security management, cloud computing and IoT security.

1 Introduction

New technological era has a remarkable impact on data accessibility, storage, privacy and performance. Day by day, data demand has been increasing which further increased challenges. Cloud computation provides solution to challenges to some extent. It enables users to compute data with minimal data handling and management efforts along with cheapest accessibility resources. There is no need for users to own infrastructure for extracting and uploading data which transformed computing technology to service-based approach. U.S. National institute of Standards and Technology (NIST) defined cloud computing as "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Tech. Report NIST 800-145, 2011)

Cloud Computing can be represented by a service model approach categorized into three layers. Each layer provides specific service to users. Top layer is often integrated with lower layers of a model. (M. K Sarkar et al. 2014, Reza, et al. 2016).

Infrastructure as a Service (IaaS) is the lowest layer which focuses on hardware sharing based services. It allows users to share, store, process and manage data by running arbitrary software on virtualized hardware.

Platform as a Service (PaaS) is the middle layer which provides platform-based services. It provides a framework to deploy, develop, test and manage application on cloud infrastructure.

Software as a Service (SaaS) is the topmost layer of model which referred as "Software on demand". This layer ensures availability of fully functional software through web browser.

Despite cloud computation significant opportunities to computation world, it also has amplified new challenges for security professionals. Technical professionals are researching to enhance data privacy, performance and security altogether. Cloud computation has further complicated data security method due to on demand and easy accessibility of data. Traditional methods cannot be beneficial in this computing spectrum. Data can be edited, deleted and retrieved at any time from any place with the need of assurance of correctness of users' data and proper data segregation on cloud. Another challenging issue in cloud paradigm is to ensure user's information and data integrity. Unauthorized accessibility is easily achievable due to possibility of number of access points in cloud domain. Users are reluctant to store their sensitive data in the cloud due to absence of standard processes and policies.

Various techniques have been introduced to address above mentioned issues of data security on cloud. Simple encryption methods do not prove useful for issues of data confidentiality. However, the following research methods have been proposed so far (Yunchun et al., 2014):

*Homomorphic Encryption***:** It performs calculations on encrypted data without decrypting it. Later on, the method was further improved by (Gentry, 2009). However, the computation calculations result in significant increase in cost.

Hybrid Technique: It uses proper arrangement for secured key sharing along with enhanced authentication techniques.

Distributive Storage: In this method, the data is segmented into smaller chunks. Each resulting chunk is encrypted and stored in separate database on the cloud. However, it utilizes a large amount of resources.

Data Concealment: It merges original data with fake visual data to increase overall volume which later on differentiated by authorized user.

Each technique has limitations to certain extent. By increasing complexity, calculations, algorithms can enhance security but on other hand performance will be reduced. Researchers are focusing on optimal utilization of resources along with robust performance and powerful data security. In this paper, we proposed a hybrid design for ensuring data security in cloud and resolve the key management problem. Using an intelligent mix of Steganography, Elliptic Curve Cryptography (ECC) and Advanced Encryption Standard (AES), we are able to achieve fast and reliable data sharing on the cloud. We used AES to encrypt the large blocks of data of any size and then encrypt the symmetric key with public key of any sharing partner. The encrypted symmetric key is also stored in an image using steganography. This methodology helps in improving key distribution and management with only a public-private key pair. The AES key is only stored on cloud using asymmetric encryption and steganography. In our design data encryption is done by symmetric encryption and key sharing is provided by asymmetric algorithm. Steganography is used to hide encryption key and avoid any suspicion of secure mechanisms.

This paper is divided into several sections. Previous work is discussed in section 2. Section 3 discuss the background of our research and established our platform. In section 4 we present our hybrid design and its working methodology. Section 5 elaborates how it is implemented and the deliverables.

2 Previous Work

In 2003, Certificateless Public Key Cryptography (CL-PKC) was first proposed by (Al-Riyami et al., 2003) which was further modified by (Sun et. al. 2007) Both methods were not fruitful due to key revocation problem, due to which compromised public keys were not revoked. While in 2006, (Chow et al, 2006) supported the idea of immediate revocation technique which caused a Certificate revocation issue. Symmetric key encryption method is based upon sharing of identical keys for encryption and decryption. Due to absence of key secrecy, (Shang et al, 2010) presented a scheme to assure privacy of users and resolved key handling issues.

In 2014 deterministic finite automata-based functional Proxy Re-Encryption scheme is proposed by (Liang et al, 2014) to provide security against access control. In this technique data is encrypted along with arbitrary length string which causes secret key to not reveal encrypted message unless tagged DFA receives string. However, this method remains unpractical due to

sharing of proxy keys to all users (Mohis et al. 2016). In cloud computation, due to security concerns, data embedded methods have been introduced. Among those methods, Jsteg (Kodovsky et al, 2010), F5 (Fridrich et al, 2002), LSB (Johnson et al 1998) are commonly used. Researchers are focusing on providing powerful security and enhanced performance of storage. Therefore, the idea of data embedding in image has been proposed. (Mohis et al, 2016) suggested a scheme in which stenographic method is merged with mediated certificateless encryption. This integrated scheme improves overall performance of storage and ensures security of data inside cloud against attackers. In 2014, (M. K. Sarkar et al, 2014), proposed a security model by using stenographic technique. The mechanism transforms data into image and changes the composition of 8-bit pixel by changing the last bit. Therefore, data security on cloud is enhanced. Computation problem arises in this scheme when the data size is increased.

In 2016, data cloud security was provided with multiple layers using steganography and cryptography (Alok et. al. 2016). In this model, data is first encrypted and then transformed into media file depending upon priority of data confidentiality. When data is of high sensitivity, audio or image steganography has been used, otherwise simple cryptography technique is used.

The proposed research in (Dasgupta et al, 2017) used H-LSB technique for hiding message. Hackers find it quite difficult to obtain data due to significant level of complexity. In H-LSB scheme, raw data is first transformed by using AES encryption technique which further embedded in to RGB carrier frame. LSB steganography is done by inserting 8 bits of encrypted data into R, G, B colors with 3, 3, 2 order. This means, for embedding 8-bits in specific RGB pixel, 3 LSBs of R color is used, and 3 LSBs of G color is used and 2 LSBs of B color is used. R, G channels both carry 3 bits and B channel carries 2 bits. The reason for using only 2 bits in B is that human visual system (HVS) is more sensitive to B color than R and G colors. In the embedding process, for example when embedding in the R channel, 3 bits out of 4 LSB bits are used. Hash function is used for determining which 3 bits out of 4 LSBs of the R channel are used. The same process is applied to both G and B.

In 2015, (Hayfaa et al. 2015) presented a technique of integrating stenographic simple LSB and Color Image Based Data Hiding (CIBDH) methods along with cryptographic method for data security on cloud. It enhances capacity, performance and security at a time by controlling PSNR and MSE parameters.

In 2014, (Pye et al, 2014) used a technique in which cryptography and steganography methods are applied. AES encryption technique produces cipher text along with symmetric key k1. Cipher text is transformed into two additional keys k2 and k3. One key is hidden in the cover image and cryptography is used for protecting the other key. A stenographic method based on Discrete Cosine Transform (DCT) is used to hide a part of encrypted message as key in image (Sarkar, et al. 2014). This combination produces further security against attackers.

In 2015, (Harleen et al. 2015) proposed a method for ensuring data over internet. Multiple Least Significant bit (MLSB) steganography method is applied for hiding data. Digital Signature Scheme Algorithm (DSA) is also applied along with steganography for security against data authentication.

In 2015, (Blessy et al, 2015) presented a method for encrypting RGB images based on ECC encryption to provide security against unauthorized attackers. Two levels of security are available depending upon the level of confidentiality needed. XOR function is applied on each pixel in case of low confidentiality. However, two steps of security are applied for higher confidentiality. In first step, an encrypted image is produced by XORing image with key image. In the second step, the XORed image is encrypted with ECC encryption. ECC focuses on processing power, energy and bandwidth parameters.

In 2015, (Nikita et al. 2015) used a combination of Hash based LSB and DWT for improving embedding quality and security. By using H-LSB with RSA algorithm, bit planes are produced, secret data is inserted in LSB of bit planes. DWT technique is used for transforming data bits into image bits. DWT reduces complexity level of embedding and enhances security level. Higher security with combining cryptography and H-LSB steganography is proposed in (Abood 2017)

In 2016, (Ms Nikita et al. 2016), applied ECC technique for securing public key. Combinational security of data and cloud is provided by using integrated mechanism of ECC and digital signature mechanism. (A. Miele et al. 2015) deployed a method in which personalized and short lived ECC parameters are generated rather than pre-generated elliptic curve parameters. In 2017, (Inusha M. et al, 2017) used a combination of ECC and 3-level DWT techniques for scheme using cryptography and steganography. Key generated by ECC technique is well defined in terms of speed, size and storage. For data security, 3-level DWT is used. The cover image is further be split in terms of frequency and directions by using wavelet transform. Various sub levels are generated which embeds secret data.

The proposed techniques for protecting data on the cloud focused generally on improving the encryption security, robustness to attacks, speed, and storage. The authors focused on the technical measures and neglect testing their approach to managerial measures such as key management and key distribution. Therefore, we proposed a hybrid technique that focused mainly on improving key management and distribution procedures.

3 Background

In this research paper different tasks are completed by utilizing strngths of various cryptographic algorithms. It is essential to understand each for detailed picture of the proposed solution.

3.1 Steganography

Steganography is a technique that hides secret information in normal data files such as image files, audio files and video files. Steganography uses several techniques to conceal secret information in apparently unimportant data files. The steganography is an obscure channel of

communication through which secret data can be transferred in complete protection. Steganography also avoids raising any alarms as is done in case of cryptography.

In this research paper, we used image LSB steganography. The changes due to hidden data are invisible to human eye and thus secret keys can be transmitted using normal data files. Least significant bit of each pixel in an image can be used to hide one bit of secret information. Change of LSB does not affect the overall image appearance and thus is an effective medium for hiding information.

3.2 Elliptic Curve Cryptography (ECC)

Asymmetric algorithms play a pivotal role in filling gaps left by symmetric algorithms. Digital signatures and key exchange mechanisms delivered by asymmetric algorithms have proven effective and efficient. However, these algorithms are based on mathematical problems like N-P hard problems. NP hard problems are difficult to solve and compared with non-deterministic polynomial-time hardness as a measure of their toughness.

Integer Factorization Problem (IFP), Discrete Logarithm Problem (DLP) and Elliptic Curve Discrete Logarithm Problem (ECDLP) are all NP hard problems. RSA is based on integer factorization problem (R. L. Rivest et al. 1978), Diffie-Hellman (DH) and El- Gamal algorithms are based on DLP. Elliptic Curve Diffie-Hellman (ECDH), Elliptic Curve El-Gamal (EC El-Gamal) algorithms are based on ECDLP. The main advantage of using asymmetric authentication techniques is that it normally requires one pass to authenticate. Although it requires more extensive computation as compared to symmetric algorithms, but generally mathematical computations take less time. In asymmetric technique selection of algorithm is difficult because all algorithms are based on strong mathematical problems. In public key cryptography, selection of algorithm is based on key size, as it affects the speed of algorithm. Table 1 compares different available algorithms based on their key sizes in bits. The strength of an algorithm depends upon the amount of resources or time or both required to break its security. (Win, 2015)

Table 1 Equivalent Key Strength of Different Algorithm

AES	Diffie Hellman	RSA	ECC
80	1024	1024	160
112	2048	2048	224
128	3072	3072	256
192	7680	7680	384
256	15360	15360	521

(Miller, 1985) and (Koblitz, 1987) proposed the use of elliptic curves in public key cryptography. The values in Table 1 reveal that ECC provides stronger cryptographic work factor with much smaller key size. In this paper, ECC is used to encrypt and secure the AES key. The overall security of mechanism presented in this paper is based on the cryptographic strength of elliptic curve cryptography.

3.3 Advanced Encryption Standard (AES)

Currently AES is the main contender for ensuring security in digital systems. Most of the data security available in cloud is based on AES. Ease of implementation and high speed makes it an ideal choice for encrypting large blocks of data that is stored, processed and transmitted using cloud. The AES algorithm (Federal Information, 2001) was selected in 2001 by the US National Institute of Standards and Technologies (NIST). The AES algorithm is a symmetric block cipher that uses one key to encrypt and decrypt sensitive information. Encryption transforms meaningful data to an unintelligible form called cipher text which cannot be understood without the possession of AES key; decrypting the cipher text converts the data back into original meaningful, called plaintext.

AES provides the benefits of various design possibilities and architectures along with reliable security. The design security provided by AES requires careful implementation for matching cloud requirements.

AES is a 32-bit, block cipher and works on a plain text of size of 16 bytes (128 bits) and performs several iterations to add confusion and diffusion. This cryptographic algorithm can provide security with variable key sizes of 128, 192 and 256 bits. Steps involve in AES are:

- SubBytes
- ShiftRows
- MixColumns
- XORRoundKey

AES is the best choice for encrypting large blocks of data. Asymmetric and steganography techniques fail to match the performance delivered by AES. Any solution for cloud security must consider the large amount of data stored and processed by cloud technology (Ahmed et al, 2011).

4 System Design

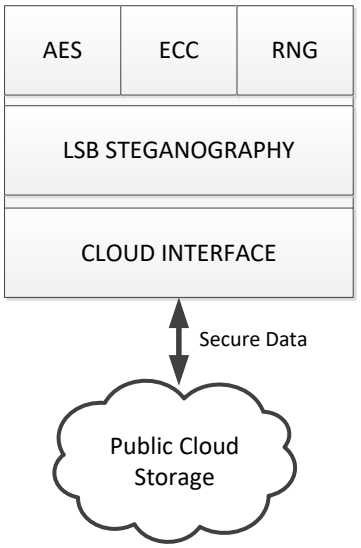
The presented design is based on several key components that provide the holistic security model for storing and sharing sensitive data using public cloud technology.

As shown in Figure 1, three different components are being used to provide strong security mechanism. AES with 256-bit encryption decryption engine is used to encrypt the data that is to be uploaded on the cloud. AES has delivered excellent results for encrypting bulk data at high speeds. (Fed. Info report, 2001).Steganography processes are used for data security. Steganography stores sensitive data inside another data, thus increasing the data size to be stored on cloud. This results in increase bandwidth and cloud reduce storage utilization, which is not cost effective.

The AES key will not be stored in the application. Using ECC encryption, it will be encrypted then embedded in an image using steganography. This process helps to store different files encrypted with varying AES keys. No issue of key management arises as each key is stored along with the data on the cloud. ECC encryption is time consuming process. Therefore, we do not use it to encrypt data. Rather in our design it is used to encrypt a fixed amount of data i.e.

AES 256 bit key. Hence ECC encryption and decryption process will always consume the same time irrespective of the size of data. In this design we are using GF(p) which uses prime field to compute public/private key pair (Khali et al. 2007).

Figure 1: The block diagram of the proposed cloud security system



A random number generator (RNG) is also provided in the application which eases the task of generating new keys for AES operation. It can also be used for generating private key for ECC. However, each participant will have only one ECC private key and one public key. This whole system security is based on the private key of each user for protecting user data and shared data by other users.

LSB steganography is used in our design. Since our objective is to improve key management and distribution, LSB is selected for its simple implementation. However more robust steganography technique can be used. A customized version of H-LSB is used (Alok Ranjan et al., 2016, Dasgupta et al. 2017). We are using fixed size 24-bit coloured images as vessel images for hiding the encrypted AES key. Each cover image used is of size 128 by 128 pixels. Each pixel is represented by 24 bits, 8bits each for red, green and blue. The changes done in image are not detectable by human eye. We are using a variation of H-LSB steganography. Only two pixels in each row are used to store 2 bits. Hence total of 256 bits are stored in 128 rows. The column in each row is selected by adding 100 to each row number (numbering starts with 0) and then taking modulus 128. The second pixel is selected by adding 50 to the row number. For example, in the first row numbered 0, the first pixel used will be 100th and second pixel will be the 50^{th}. To determine which colour byte will be used for storage in the selected column pixel, row number is modulated with 3. The resulting byte positions are 0, 1, and 2 for red, green and blue bytes respectively. Using this technique all the colour pixels will be equally used for storing and much less alteration will be performed on the cover image.

Cloud interface can be altered to collaborate with any specific data storage provider. Currently we are using google drive as a test area for the proposed system implementation.

6 System Procedures

Reaping the benefits of this design is possible after elaboration of working process. All the functions are performed in a sequential order by the application.

As shown in Figure 2, a random number is generated of size 256 bits. This number K is used as key in AES to encrypt any data, which is to be uploaded. Depending upon requirements users can generate this key for each file or folder uploaded to cloud. The user A also generates one more 256-bit random number as P_{rA}. Using this as private key in elliptic curve algorithm and working on prime field to generate a public key of size 512-bits. Each coordinate in the (x, y) plane is 256-bits each. In our application, only the x coordinate is used. Algorithm 1 shows how ECC encryption is performed and Algorithm 2 shows how ECC decryption is performed (Begum et al. 2011).

All the addition, multiplication and subtraction are point addition, point multiplication and point subtraction. These point operations result in point on the same curve that we have chosen i.e. E in algorithm 1 and 2.

Algorithm 1: ECC Encryption

Input: Elliptic curve parameters (E, p, P, n), private key P_r and public key $P_u = P_rP$.
Output: Cipher Text (CT_1, CT_2)
Ensure: *Convert message AES key K to point on elliptic curve M.*

1. Select $k \in R^{[1,n-1]}$
2. **Compute** $CT_1=kP$
3. **Compute** $CT_2=M + kP_u$
4. **Return** (CT_1,CT_2)

Algorithm 2: ECC Decryption

Input: Elliptic curve parameters (E, p, P, n), private key P_r and Cipher Text (CT_1, CT_2)
Output: AES key K as message M
Ensure:

1. **Compute** $M = CT_2 - dCT_1$
2. **Extract key** K from M
3. **Return** (K)

In Figure 3, retrieval of secure information from cloud is elaborated. However, this time private key P_{rA} is used for decryption. It should be mentioned here that scalar multiplication which is the product of a scalar number and a point on elliptic curve, follows commutative property. Extraction of M during decryption is the result of this property. The calculation is elaborated in equations 1 and 2.

$$dCT_1 = kP_u \quad (1)$$

$$dkP = kdP \quad (2)$$

P in this case is the generator point that will be similar for users of this design. The only variable for different users is their public/private key pairs. Sharing of public keys with interested parties is not a security hazard as

it is computationally infeasible to retrieve private key information from public key.

The placement of secure data storage in cloud is shown in Figure 4. If user A wants to share his/her encrypted data with user B or C, he/she will only download image A instead of downloading the whole data (user A can't share it directly because data is encrypted). Then decrypt it with their own private key using process shown in Figure 3. After this step the user A will encrypt the AES key with user B or user C public keys and hide in images B.jpg and C.jpg respectively, using steganography. This process results in secure and simple sharing of data using public clouds.

Figure 2: Upload secure data to Public Cloud

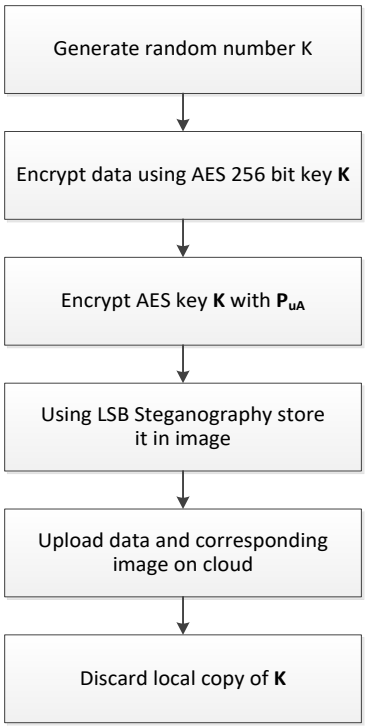


Figure 3: Downloading user data from Public Cloud

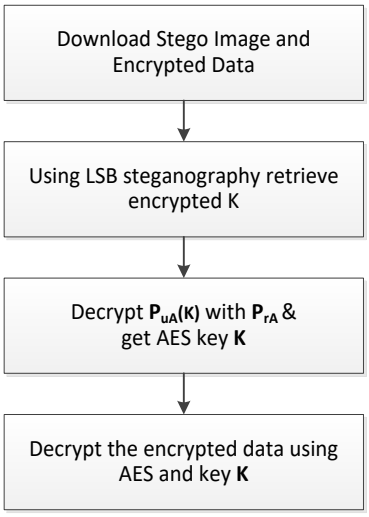
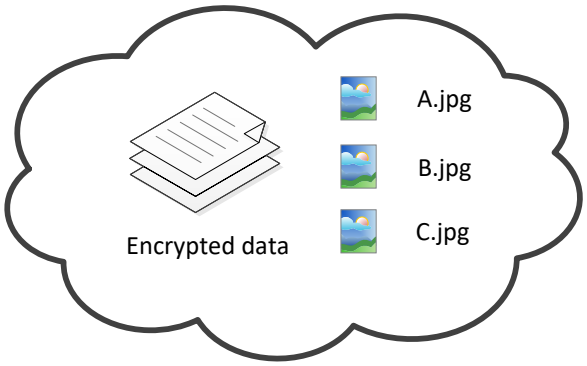


Figure 4: Secure Data on Public Cloud



7 Implementation Results and Discussion

The results for embedding the AES key into the images are presented. Steganography performance test is then followed by analysis and discussion of the adopted cryptography techniques.

7.1 Steganography

Python was used to implement the proposed design and to validate the efficiency and effectiveness of our proposed architecture. As our design is customized for LSB steganographic operations, we could not use the open source libraries for image steganography. Instead we implemented our customized LSB with Python.

We used OpenCV library for Python to read, display and write back images. The image format used by our code was PNG which stands for Portable Network Graphics.

PNG is a lossless image compression format. We are using 24 bit image format in which each pixel is represented by 24 bits, 8 bit each for red, green and blue color. This is little complex as compared to steganography in black and white images in which only one field for each bit is considered. The software forces the user to add vessel image which is of size 128 x 128 pixels and PNG image type. Formats like JPEG are rejected due to the lossy compression algorithm in them, resulting in loss of precious hidden data. The example alteration due to steganography is depicted in Figure 5. Any difference in the two images is not identifiable by human eye.

Figure 5: LSB steganography results (left) The original vessel image or the carrier image (right) Stego-image with 256-bits key embedded into it.

Both images in Figure 5 provide no visual discrimination to human eye. The other reason for less evident difference is the fixed size of data stored randomly in different colour bytes and only 2 pixels in each row are altered in Figure 5 (right).

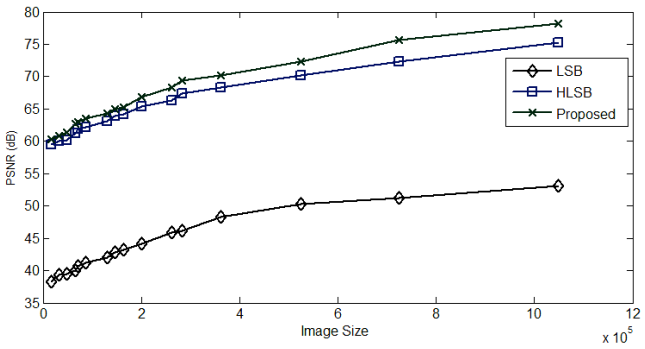
To compare the effect of embedding 256 bits in 128 x 128 pixels image, Peak Signal to Noise Ratio (PSNR) is adopted. PSNR is the standard measure of the image quality and measured in decibels (dB). The image is considered a signal and the embedded key data is considered a noise. PSNR measures how much distortion occurs to the carrier signal due to the embedded noise. Calculating PSNR depends on calculating Mean Square Error (MSE). The mathematical calculations of MSE and PSNR is depicted in equations (3), (4).

$$\text{MSE} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (V(i,j) - S(i,j))^2 \quad (3)$$

$$\text{PSNR} = 10 log_{10} \left(\frac{l^2}{MSE} \right) \quad (4)$$

Where V is the vessel or carrier image and S is the stego-image. *m, n* are number of rows and columns of the cover image, which are typically 128 x 128 in our proposed approach. *l* is the max amplitude of the image pixel value, it is typically set to 255. To be imperceptible to HVS and steganalysis tools, PSNR of the stego-image must be higher than 40 dB. PSNR lower than 40 dB degrades the image and accordingly distorts the embedded data (Hosam et al. 2016).
In the following experiment, 200 coloured images of different sizes are used. The smallest image size is 128 x 128 pixels and the largest image size is 1024 x 1024 pixels. Not all images are of square size, i.e. the width and height are not always equal. However, width and heights are not allowed to be less than 128 pixels. For specific image size, PSNR is calculated for all the images with that specific size. The resulting PSNR values are then averaged.
Figure 6 shows the results of embedding the 256-bits key with using classical LSB, HLSB and the proposed approach. The figure shows that with higher image size, the image quality is increased. The proposed approach has higher quality compared to LSB and HLSB.

Figure 6: A comparison of the proposed LSB approach with HLSB and classic LSB. The increase in images size reduces the distortion caused by embedding the encryption key into the image.



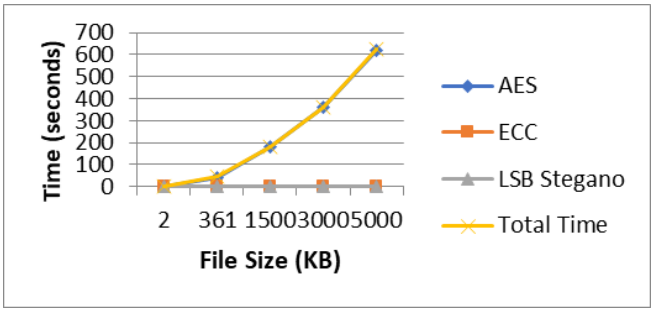
Resilience of our design against steganographic analysis is evaluated. Several techniques are used by researchers to detect or identify the presence of hidden message in images. (Sarker et al. 2014) & (Westfeld et al. 1999) discussed the attacks that can be used to breach the secrecy provided by steganography. However, the emphasis on monochrome images, in which only one byte per pixel is used, showed that using 24 bit colored vessel image is safer. The other reason that facilitated the identification is the large amount of data that is stored in stego image, which results in image distortion. As we are using fixed amount of 256 bits to hide in each image, the alteration is not damaging. As mentioned in (Westfeld et al. 1999), universal as well as specific analysis cannot identify the presence of hidden message. Even in the worst-case scenario; if the hidden image is identified and 256 bits are retrieved, our multilayer design provides defence in depth. As security in our design is not dependent on steganography, ECC encryption provides security in case steganography is compromised. Tampering stego images for user B and C in Figure 4 means tampering the sharing link. Tampering stego-images doesn't mean losing the data integrity. User A can restablish the sharing by repeating the embedding process of the private key into users A, B images.

7.2 Cryptography

AES and ECC modules are also implemented in Python. AES module can encrypt or decrypt any file of variable size. We calculated the time of each individual process and the results are shown in Figure 7.

The results show that time of ECC and steganography is fixed as it's performed on fixed amount of data. The total time is dependent on AES performance and the time increase based on larger file sizes.

Figure 7: A comparison of AES, ECC and LSB time, shows that AES takes an essential amount of time compared to ECC and LSB



The fixed time of ECC and steganography makes it possible to encrypt different files with mutually exclusive AES keys. Therefore, the idea presented provides a robust security mechanism which can withstand several security threats faced by data resident in cloud.

Data is vulnerable to sniffing and prying eyes during transmission from user to cloud storage. Our solution is providing encryption before this step to ensure the confidentiality of data.

Colocation-based attacks, which exploit the shared hardware resources in cloud, can pose significant risks to confidentiality of information. Attackers can compromise cloud segmentation and gain access to victim's data. Our security model protects cloud users from colocation-based attacks. Another problem with storage in the cloud is that of traffic analysis. As one symmetric key is used to encrypt all the data for a user, careful analysis can reveal an idea about key. The mechanism presented in this paper allows the encryption of multiple files with their own respective AES keys. This reduces the chances of traffic or side channel analysis to determine the key.

Wide acceptance of cloud is currently restricted due to privacy concerns. Privacy of data is not ensured from cloud service providers and hence people are hesitant to store their sensitive data on cloud. The idea presented in this paper can strengthen the trust on cloud by providing a secure method from the user location.

Conclusion

The potential of cloud storage is being limited by data privacy and security concerns. In the cloud, key management and distribution problem always arises. We presented an effective and efficient solution to this problem by utilizing a multitier security solution. Symmetric algorithm AES and asymmetric algorithm ECC is use to ensure secure data storage and sharing. LSB image steganography was used to hide encrypted keys to protect them from malicious users.

References

Abood, M. H. (2017, March). An efficient image cryptography using hash-LSB steganography with RC4 and pixel shuffling encryption algorithms. In New Trends in Information & Communications Technology Applications (NTICT), 2017 Annual Conference on (pp. 86-90). IEEE.

Ahmed, M. H., Alam, S. W., Qureshi, N., & Baig, I. (2011, July). Security for WSN based on elliptic curve cryptography. In *Computer Networks and Information Technology (ICCNIT), 2011 International Conference on* (pp. 75-79). IEEE.

Miele, A., & Lenstra, A. K. (2015, September). Efficient ephemeral elliptic curve cryptographic keys. In *International Information Security Conference* (pp. 524-547). Springer, Cham.

Ranjan, A., & Bhonsle, M. (2016). Advanced System to Protect and Shared Cloud Storage Data using Multilayer Steganography and Cryptography. *International Journal of Engineering Research*, *5*(6), 434-438.

Al-Riyami, S. S., & Paterson, K. G. (2003, November). Certificateless public key cryptography. In *International Conference on the Theory and Application of Cryptology and Information Security* (pp. 452-473). Springer, Berlin, Heidelberg.

Begum, S. J., & Purusothaman, T. (2011). A new scalable and reliable cost effective key agreement protocol for secure group communication. In *Journal of Computer Science*.

Blessy Joy A,R. Girish, RGB image encryption based on bitplanes using Elliptic Curve Cryptography, vol. 5, Issue 2, February 2015

Chow, S.S., Boyd, C. and Nieto, J.M.G., (2006). Security-mediated certificateless cryptography. In *International Workshop on Public Key Cryptography* (pp. 508-524). Springer, Berlin, Heidelberg.

Dasgupta, K., Mandal, J. K., & Dutta, P. (2012). Hash based least significant bit technique for video steganography (HLSB). *International Journal of Security, Privacy and Trust Management (IJSPTM)*, *1*(2), 1-11.

Daemen, J., & Rijmen, V. (2001). Announcing the advanced encryption standard (aes). *Federal Information Processing Standards Publication*, *197*.

Fridrich, J., Goljan, M., & Hogea, D. (2002, October). Steganalysis of JPEG images: Breaking the F5 algorithm. In *International Workshop on Information Hiding* (pp. 310-323). Springer, Berlin, Heidelberg. Gentry, C. (2009). A fully homomorphic encryption scheme. *Stanford University*.

Kour, H., & Kaur, S. (2015). Data Hiding Using MLSB Steganography. *International Journal of Advanced Research in Computer Science and Software Engineering. Page*, *994*.

Atee, H. A., Ahmad, R., & Noor, N. M. (2015). Cryptography and Image Steganography Using Dynamic Encryption on LSB and Color Image Based Data Hiding. *Middle-East Journal of Scientific Research*, *23*(7), 1450-1460.

Hosam, O., & Ben Halima, N. (2016). Adaptive block-based pixel value differencing steganography. *Security and Communication Networks*, *9*(18), 5036-5050.

Inusha M M, bY Manjula, cM Z Kurian (2017,May),*Random Number Generation for High Security using 3 Level DWT Steganography and ECC Encryption – A Survey* . In First International Conference On Recent Innovations in Engineering and Technology, Volume-07, May 2017

Johnson, N. F., & Jajodia, S. (1998). Exploring steganography: Seeing the unseen. *Computer*, *31*(2).

Khali, H., & Farah, A. (2007). Cost effective implementations of GF (p) elliptic curve cryptography computations. *Int. J. Comput. Sci. Network Security*, *7*(8), 29-37.

Kodovsky, J., & Fridrich, J. (2010). Quantitative structural steganalysis of Jsteg. *IEEE Transactions on Information Forensics and Security*, *5*(4), 681-693.

Liang, K., Au, M. H., Liu, J. K., Susilo, W., Wong, D. S., Yang, G., ... & Xie, Q. (2014). A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing. *IEEE Transactions on Information Forensics and Security*, *9*(10), 1667-1680.

Sarkar, M. K., & Chatterjee, T. (2014). Enhancing Data Storage Security in Cloud Computing Through Steganography. *International Journal on Network Security*, *5*(1), 13.

Mohis, M., & Devipriya, V. S. (2016, August). An improved approach for enhancing public cloud data security through steganographic technique. In *Inventive Computation Technologies (ICICT), International Conference on* (Vol. 3, pp. 1-5). IEEE.

Chintawar, N. N., Gajare, S. J., Fatak, S. V., Shinde, S. S., & Virkar, G. (2016). Enhancing cloud data security using elliptical curve cryptography. *Int. J. Adv. Res. Comput. Commun. Eng*, *5*(3), 1-4.

Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. *IEEE transactions on Computers*, *100*(1), 90-93.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of computation*, *48*(177), 203-209.

Sharma, N., & Khera, M. (2015). A Novel Approach to Image Steganography Using Hash-LSB and DWT Technique. *IJARCSSE International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(6).

Aung, P. P., & Naing, T. M. (2014). A novel secure combination technique of steganography and cryptography. *International Journal of Information Technology, Modeling and Computing (IJITMC)*, *2*(1), 55-62.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126.

Reza, H., & Sonawane, M. (2016). Enhancing mobile cloud computing security using steganography. *Journal of Information Security*, *7*(04), 245.

Sarkar, T., & Sanyal, S. (2014). Steganalysis: Detecting LSB Steganographic Techniques. *arXiv preprint arXiv:1405.5119*.

Shang, N., Nabeel, M., Paci, F., & Bertino, E. (2010, March). A privacy-preserving approach to policy-based content dissemination. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on* (pp. 944-955). IEEE.

Sun, Y., Zhang, F., & Baek, J. (2007, December). Strongly secure certificateless public key encryption without pairing. In *International Conference on Cryptology and Network Security* (pp. 194-208). Springer, Berlin, Heidelberg.

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. Miller, V. S. (1985, August). Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques* (pp. 417-426). Springer, Berlin, Heidelberg.

Westfeld, A., & Pfitzmann, A. (1999, September). Attacks on steganographic systems. In *International workshop on information hiding* (pp. 61-76). Springer, Berlin, Heidelberg.

Win, K. Z. (2015). *Elliptic curves and cryptography*. Texas Woman's University.

Sun, Y., Zhang, J., Xiong, Y., & Zhu, G. (2014). Data security and privacy in cloud computing. *International Journal of Distributed Sensor Networks*, *10*(7), 190903.