SHAREit.

Scientific Hub of Applied Research in Engineering & Information Technology

Received: 21.03.2022 Revised: 30.03.2022 Accepted: 06.04.2022

check for updates

Research Article

Improved Security of Hidden Data Based on Steganography Using ECC Method

E. K. ArulKarthick[1], R.Sanjai [*1], B. Vignesh[1], A. Rajeshwaran[1], M. Navaneethan [1]

[1]*Department of Electronics and Communications Engineering, Nandha Engineering College, Tamilnadu, India.*

Cryptography and steganography are well-known characteristics in the realm of computer networks for best security purposes. The key concept is to send the data in a secure manner. As a result, providing an adequate level of security for data transfer is critical. It should also minimize the security algorithm's time complexity. To encrypt the data and image, we used the "Elliptic Curve Cryptography" approach. To transfer the data securely, a "Least Significant Bit" steganography algorithm is employed insert encrypted data to be hidden inside the image. The image's encrypted data is subsequently decoded using the decryption method. The decrypted data is then used to extract the hidden data. The image is then compressed before being sent over the internet. To simulate the outcomes, MATLAB is utilized indicating that it has the power of good embedding and protection.

Keywords: ECC, Steganography, LSB, GUI, RSA, MATLAB, DES.

1. Introduction

Elliptic Curve Cryptography (ECC) is a proposal for asymmetric cryptography cryptology. How to set up a short loop numerical position in a specified location. Elliptic curve cryptography requires a small key compared to the non-elliptic cryptographic curve to provide the same level of security compared to other algorithms and techniques. Elliptic curves can be applied to key generation, digital signatures, random key generators, and other functions. Indirectly, it can be used for encryption by combining key sharing with an uneven encryption system. They are also used in other complete factorization algorithms based on elliptic curves that have applications in cryptographic systems such as: Lenstra elliptic curve factorization.

Correspondence should be addressed to Sanjai; sanjair83@gmail.com

The least significant bit (LSB) is the binary number one bit. The value that specifies the unit value, that is, determines whether a number is even to or odd. The less important one is sometimes called the lower order or the rightmost bit. This is related to the norm of writing the least significant digit further to the right in position denotion.

This is the least significant digit of a decimal integer, that is Rightmost digit. It is common to assign a position number in the range 0 to N1 to each bit. The number of bits in the binary encoding utilized in this example is N. Usually this is just an exponent of the corresponding bit weights in the base. The manufacturer of some control processing units assigns bit numbers in the opposite way, but the concept of the least significant bit itself remains unidentified as a threat to the unit bits. As a result, the least significant bit is the number of bits that is closest to the least significant bit and contains the least significant bit.

The block diagram in Fig. 1 shows that the message is sent from the sender to Receiver. The message is then hidden in the LSB of the image. This method is

performed using LSB steganography. The ASCII value of the text is recognized first and then converted to a binary value. The next step is to embed the text in the LSB of the image. The recipient's public key is known to the sender. The original message is encrypted using the recipient's public key using an encryption algorithm. Encryption is done using elliptic curve cryptography. The original message is then converted into a cipher text, which is an encrypted message. The encrypted message is then decrypted using the recipient's private key, which matches the recipient's public key. Decryption is then done using Elliptic Curve Cryptography. The text is then taken from the LSB of the image.

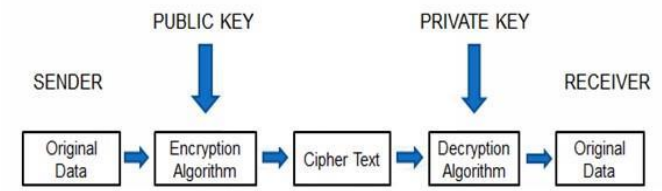

Fig.1. Block diagram of ECC

Steganography is a technology that hides secret data in normal non-arrays, files, or messages to avoid detection. Secret data is extracted in its destination. As an additional step to obscure or safeguard data, steganography can be used with encryption. Word Stage Phone is derived from Greek Word Steganos (that is, hidden or covered) and Greece root graphics (meaning that they write). Hidden data can be hidden in almost other types of digital content. The top-covered content covered by steganography is encrypted before installed in unsightly cover text files or data streams. If not encrypted, Encrypted text is often processed in any way to increase the difficulty of finding confidential content.

Steganography is done by anyone who wants to convey a private message or code. Although steganography has many legitimate uses, it has also been discovered that the authors of a malware program use steganography to hide malicious code transmission. The type of steganography has been used for centuries and includes almost all techniques for concealing confidential messages in harmless containers. For example, use invisible ink to hide harmless private messages. Hide recorded documents in microdots with a diameter of about 1 mm on or inside what appears to be a legitimate response. You can also share information using the multiplayer gaming environment.

2. Steps in Elliptic Curve Cryptography

2.1. Key Generation Module

For data encryption and decryption, the public, secret, and private keys must be generated in order for the data to be transferred in a secure manner. The algorithm is shown below.

User A key generation

Sender A selects a random number kA from 1 to n1.

Sender A then uses the formula to generate the public key

Public key P = kA * G

n / a is the sender's private key

G generation point

User B key generation

Sender B uses the public key expression R = kB * G to generate the public key

kB is the recipient's private key

G generation point

Private key generation

Private key of A, K = kA * R

Private key of B, K = kB * P

B's private key is obtained by multiplying B's private key and A's key.

A's private key is obtained by multiplying A's private key by B's public key.

2.2. Encryption Module

Encryption is done with a short public key. Fig 2 shows a typical elliptic curve of the encryption curve when the encryption and decryption algorithms are executed.

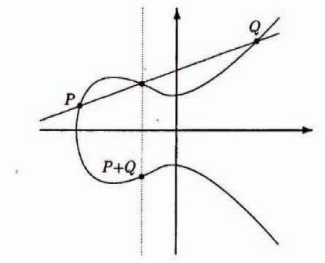

Fig.2. ECC Curve

Encrypted part procedure:

- Select a message to hide secretly in the image.
- Encrypt the message using the public key and the ECC algorithm.

Image encryption algorithm:

- Find the pixel value of the image to encode and randomly add 1 or 2 to each pixel. Note the number of channels in the image.
- Add up the pixels and convert each group into a larger number. The number of pixels in a group using Mathematica is given by grp = Length [Integer Digits [p, 258]] -1.
- Combine the results obtained in step 2 and save as "Pm", a simple message prompt for your ECC system.
- Calculate "kG" and "kPb" using a random "k." The recipient's public key is "Pb."
- Dot addition of "kPb" is performed for each value of "Pm", and it is saved as the cipher text "Pc".
- Convert the cipher text list in step 5 to a value between 0 and 255.
- To make each list the same length, pad each list in step 6 to the left with less than the grp + 1 element with a 0.

2.3. Decryption Module

Steps involved in decryption part:

- Get the text embedded image.
- Resize the image.
- To recover the encrypted message, find the LSBs and separate the message.
- Release the message with the secret key and the ECC algorithm.

2.4. Steganography

Steganography is the process of hiding text in a picture by transforming the text's ASCII value to a binary value and then embedding it in the image's lsb bits.

Steganography module:

- Convert the message to a binary format
- Launch the outgoing image in the same way as the input image
- Open each pixel in the picture and do the following,
- Convert pixel values into binary numbers
- Embed the next part in the message
- Create temporary variables
- If the message bit and pixel LSB are the same, set the temp = 0.
- If the message bit and pixel LSB are different, set the temp = 1.
- This temp setting can be done by installing XORing LSB bits and message pixels.
- Update the output image pixel to input image the pixel value + temp
- Continue updating the output image until all the bits are inserted into the message
- Lastly, write both the input image and the output image in the local program.

2.5. Compression Module

Steps involved in compression part:

- Read the original file.
- Calculate the total number of words, letters, special characters and numbers in a file.
- Find the recurring word in a file.
- Prepare a glossary of the context of the original file.
- Create compressed file. Enter the name number in the compressed file instead of the real name.
- Add the dictionary to the compressed file.
- Save the compressed file to the dictionary.

2.6. Image Decryption

The decryption algorithm says:

- Find the pixel values of the encrypted image and the group with the pixel number of Grp + 1 to create the maximum number for each 256 base group. Note the number of image channels in the encrypted image.
- Pair the values found in step 1.
- Perform multiplication of the dots "kG" and "nB". Where "nB" is the recipient's secret key.
- Perform minus the point between the value in step 2 and the value in step 3.
- Use base 258 to get the values in the range 0-255 from step 4 and subtract 2 at each value.

- Add the flat value obtained in step 5 according to the recorded number of the image cypher of the image channels and divide them into the width of the image of the cypher.
- Convert the values from step 6 into plain image.

3. MATLAB

MATLAB (Matrix Laboratory) is the programming language of MathWorks and the computer environment. With MATLAB, you can modify matric, structural and data functions, use algorithms, create user links, and work with programs written in other languages such as C, C ++, C #, Java, Fortran, and Python. The interface is possible. Although MATLAB is primarily intended to calculate numbers, the toolbox of your choice uses the MuPAD image engine, which provides access to symbolic computer capabilities. An additional package, Simulink, adds multi-domain image simulation and design based on a dynamic and embedded programming model.

Format data types are available in MATLAB. Because all the variables in MATLAB are the arrays, the "layout list" is the appropriate moniker, as each segment of the array has the same field names. MATLAB also enables dynamic field names (field tests, field tricks, etc.). Unfortunately, because MATLAB JIT does not support MATLAB structures, just a few flexible integrations in the structure can be costly.

When you create a MATLAB function, the file name must match the name of the original function in the file. Functional job names start with a letter of the alphabet, sand may contain letters, numbers, or underscores. Functions are very sensitive to litigation. Activity handles, or function indicators, are used in .m files or functions embedded in an anonymous nest in MATLAB to incorporate lambda calculus features.

MATLAB supports object-oriented programs such as classes, inheritance, visual submissions, packages, passing value semantics, and passing reference semantics. On the other hand, syntax and pronouns are very different from those of other languages. MATLAB has a numerical category and a reference category, depending on whether the class has a handle such as a superclass (reference category) or not (value category). MATLAB supports application development of graphics (GUI) applications. GUI Development Environment (GUI Development Environment) is a place to design a visual interface with an image attached to MATLAB. It also has a tightly integrated graph editing feature. For example, you could use a worksheet to create a graph in two vectors x and y.

Tasks and sub-programs built in C or FORTRON programming languages can be accessed by MATLAB. Created wrap function that can be used to transfer MATLAB data types. Upload object file created by merging such activity. Since 2014, Python's dual connection has been added. There are also import and export libraries. Perl, Java, and .NET libraries can be attributed directly to MATLAB, and many MATLAB libraries are used as Java or ActiveX libraries. Driving MATLAB in Java is very complicated, but you can do it using the MATLAB Toolbox, which can be purchased separately through Mathworks, or through an anonymous process known as JMI (Java-to-MATLAB Interface). Added the official Java MATLAB API. Because MATLAB is a product related to MathWorks, users be. MATLAB Builder products can provide MATLAB services such as library files that can be used on .NET or in JAVA application building sites, but future developments will continue to be associated with the MATLAB language. Each toolbox should be purchased separately. When applying for a test license, MathWorks sales department needs more information about the MATLAB project being tested.

4. Encryption and Decryption Using Elliptic Curve Cryptography

Image and text encryption and decryption is done using the Elliptic Curve Cryptography algorithm. This method is mainly used to transfer images and text in a secure way.

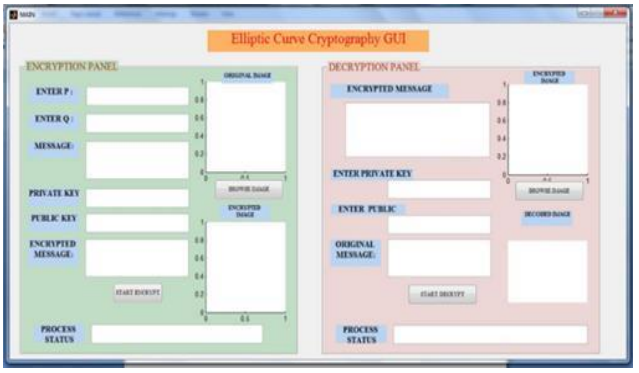

Fig.3. GUI

4.1. Encrypt the Data Using the Encryption Algorithm

Encrypt the secret message with “elliptic Curve Cryptography” with the public key published by the receiver. The data in Fig 4 is given as "nandha engineering college" in the text box. This given data is encrypted in the sender side by using the encryption algorithm. Where we will use the public key for the encryption purpose. By using the encryption algorithm, the text is encrypted by using the encryption algorithm as shown in the Figure 4. Sender A must send a "pm" message to Recipient B. Here, the text pm is embedded by taking the ASCII value of pm and multiplying it by an integer. The text is now embedded in the dots. Here, the cipher text Cm is obtained using the following formula: Cm = (k * G, s + k * R), k random integer, G generation point, pm message R Recipient B's public key The cipher text, which is the cipher text, is in Cm. Cm consists of the following pairs.

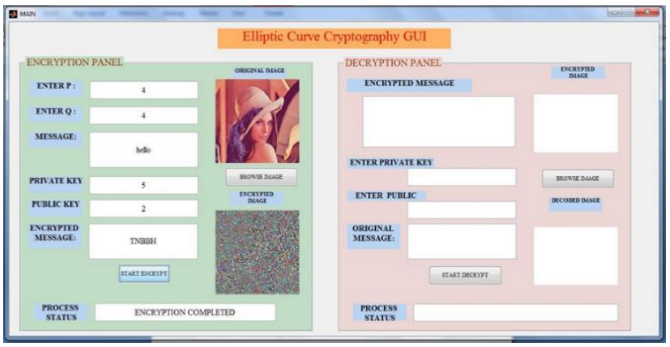

Fig.4. Data before encryption

4.2. Decrypt the Data from the Image

Decrypt the data using the recipient's private key. Decrypts the text using a decryption algorithm to get the original text sent by the sender.

Decryption algorithm:

To decrypt the cipher text, Recipient B first multiplies the first point kG by Recipient B's private key kB.

kB(k*G)

The result is then subtracted from the second point by the receiver.

k*R – kB(k*G) + Pm

The receiver will receive the original text "pm."

Fig 5 shows the output of the decrypted data on the receiving side using the decryption algorithm.

Fig.5. Decrypted data from the image

5. Steganography Method Using Elliptic Curve Cryptography

The steganography method is used to hide the data inside the image at the least significant bit of the pixel in the image. The steganography method requires you to select an image that hides the data. Fig 6. a) is the original image where the data needs to be hidden in order to send it more securely.

Fig.6. Steganography Method

a.) original image b.) image contains encrypted text

Steganography is used in combination with elliptic curve cryptography. It means that the encrypted data is hidden inside the resized image. There is no such difference between the resized image and the data hidden in the image. The left side of Fig 6. b) shows the resized image before the data was hidden. The image on the right shows the encrypted data hidden in the image.

6. Elliptic Curve Cryptography for Image Encryption

Image encoding using ECC is usually done by mapping pixel values on elliptic curve links. Mapping requires a separate viewing table. Alternatively, use the repetition function of the pixel point on the generator "G" to generate the corresponding links in the elliptic curve. In such cases, a map table is required during the decryption process to produce the corresponding pixel values from

the encrypted image. We are working on pixel groups to reduce the number of calculations. The pixel group is converted to a big single-digit integer, with the restriction that it does not exceed the value "p", which is one of the parameters of the finite elliptic curve equation in ECC operations, these huge integer values are delivered as inputs in pairs and displayed as "Pm." This operation helps ignore map functionality and the need to share a map table between the sender and the recipient.

Fig.7. ECC Method

a.) original image b.) image contains encrypted text

The above Fig 7. a) is the original image that undergoes the image encryption and decryption using the elliptic curve cryptography.

Image Encryption Algorithm:

- Fig 7. b) shows an image encrypted from the original image using the encryption algorithm shown below.
- Find the pixel value of the image to be encoded and randomly add 1 or 2 to each pixel. Note the number of channels in the image.
- Group pixels and convert them into a maximum number for each group. The number of pixels in a group using Mathematica is given by grp = Length [Integer Digits [p, 258]] - 1.
- The results of step 2 should be compiled and saved as "Pm," which is a simple message that you enter in the ECC system.
- Select random "k" to calculate "kG" and "kPb". The recipient's public key is "Pb."
- Add "kPb" points to each value of "Pm" and save as "Pc" cypher text.
- Convert the cipher text list in step 5 to a value between 0 and 255.
- Left pad containing less than 0 grp + 1 elements in each list in step 6 to equalize the length of each list.

7. Conclusion

If the public key size is not particularly huge, ECC text encryption and decryption works well. Furthermore, the processing time will be longer than with a straightforward encryption approach. However, it is more secure than the single layer of security maintained by using simply data encryption methods. If the secret data is large, it must be compressed, and an encryption method other than ECC should be employed instead. In this instance, it is necessary to check the method's processing time, as it is a critical metric for the processing cost. We performed the operation of grouping the pixels in picture encryption and decryption using ECC. Instead of transferring the grouped pixel values to elliptic curve coordinates, the values were paired. It's easier to avoid using a reference mapping table for encryption and decryption. Even though the image is made up of the same pixel value, this approach provides a low correlated cypher image.

REFERENCE

[1] Amna Shifa, Muhammad Sher Afgan, and Muhammad Sher Afgan (2018), “Joint Crypto-Stego Scheme for Enhanced Image Protection with Nearest-Centroid Clustering” , IEEE Access PP(99):1-1.

[2] Dr. Rajni Jindal, Swapnil Shalini, Shivani (2021), “High capacity Reversible Data Hiding scheme using prediction tuning model”.

[3] Fan Chen, Yuan Yuan, Hongjie He, Miao Tian (2020), “Multi-MSB Compression Based Reversible Data Hiding Scheme in Encrypted Images”.

[4] Ioan Catalin Dragoi, Member, IEEE, and Dinu Coltuc, Senior Member, IEEE, “On the Security of Reversible Data Hiding in Encrypted Images by MSB Prediction”.

[5] Jayati Bhadra, Banga.M. K, Vinayaka Murthy (2018), “securing data using elliptic curve cryptography and least significant bit steganography”, International conference on smart technology for smart nation, IEEE Explore, volume 86.

[6] Kim.C. R, Lee S.H, Lee.J.H, Park.J. I (2018) “Blind decoding of image Steganography using entropy

model", electronics letters IEEE Explore volume: 54, issues:10.

[7] LaiphrakpamDolendro Singh and Khumanthem Manglem Singh (2015)," Image encryption using elliptic curve cryptography", eleventh international multiconference on information processing, vol 54, pp 472-481.

[8] M. Manimekalai, R. Bakkiyalakshmi (2007), "Hide and Seek: A New Way to Hide Encrypted Data in QR Code Using the Concepts Steganography and Cryptography".

[9] Mariusz Dwzonkowski, Romanrykaczewski (2021) "Reversible Data Hiding in Encrypted DICOM Images Using Cyclic Binary Golay (23, 12) Code".

[10] Muhammad.K (Nov 2016), "A Novel Magic LSB Substitution Method (M-LSBSM) using multi-level encryption and achromatic component of an image", Springer Link, volume 75, issue 22, pp 14867-14893.

[11] Ms. ShrideviShetti and Mrs.Anuja S (2013), "A Secure Image Steganography based on RSA Algorithm and Hash-LSB Technique", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by, ICESMART-2015 Conference Proceedings, issue 2015.

[12] Pauline Puteaux and William Puec (2018) "High-Capacity Reversible Data Hiding in Encrypted Images using MSB Prediction".

[13] Prabhash Kumar Singh, Biswapati Jana and Kakali Datta, "Superpixel based robust reversible data hiding scheme exploiting Arnold transform with DCT and CA".

[14] Prof. AJ kadam, Piyush Patil (2021) "Data hiding under QR code using visual secret sharing".

[15] Rutuja Kakade, Nikita Kasar, Shruti Kulkarni, Shubham Kumbalpuri (2017), "Image Steganography and Data hiding in QR Code".

[16] Shabina N. Ahmed and Vinod Todwal (2019), "A Comparative Study of Image Steganography and Text Cryptography", International journal of research in engineering, science and management, vol 2, issue 3.

[17] Xi-Yan Li, Xia-Bing Zhou, Qing-Lei Zhou (2020) "High-Capacity Reversible Data Hiding in Encrypted Images by Information Processing".

[18] Xu Wang, Li-Yao Li, Ching-Chun Chang (2021) "High-Capacity Reversible Data Hiding in Encrypted Images Based on Prediction Error Compression and Block Selection".

[19] Yan Ke, Minqing Zhang, Xinping Zhang, Member, IEE, Jia Liu, Tingting Su, Xiaoyuan Yang, "A Reversible Data Hiding Scheme in Encrypted Domain for Secret Image Sharing based on Chinese Remainder Theorem".

[20] Yang Ren-Er , Zheng Zhiwei, Tao Shun, Ding Shilei (2014) , "Image Steganography Combined with DES Encryption Pre-processing", 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, volume 03, issue 54.

[21] Yaomin Wang, Zhanchuan Cai, Wenguang He (2020), "High Capacity Reversible Data Hiding in Encrypted Image Based on Intra-Block Lossless Compression".

[22] Zhaoxia Yin, Member, IEEE, Na Xu and Feng Wang, "Separable Reversible Data Hiding Based on Integer Mapping and Multi-MSB Prediction for Encrypted 3D Mesh Models".

[23] Zichi Wang, Guorui Feng, and Xinpeng Zhang (2021) "Repeatable Data Hiding: Towards the Reusability of Digital Images".

[24] Workshop Sensor Netw. Protocols Appl. pp. 149-155 May 2003.

[25] X. Liu "A survey on clustering routing protocols in wireless sensor networks" Sensors vol. 12 no. 8 pp. 11113-11153 2012.

[26] S. Uppugalla, P. Srinivasan, J. Solid State Electrochem. 2019, 23, 295. [53]

[27] Katakam P, Sriram N. Formulation and evaluation of mucoadhesive microspheres of pioglitazone hydrochloride prepared by solvent evaporation technique. Int J Biol Pharm Res 2012;3:1005-15.

[28] Sriram N. Antidiabetic and antihyperlipidemic activity of bark of Casuarina equisetifolia on streptozocin induced diabetic rats. International Journal of Pharmacy Review and Research 2011; 1(1): 4-8.

[29] Male U, Uppugalla S, Srinivasan P. Effect of reduced graphene oxide-silica composite in polyaniline: electrode material for high-performance supercapacitor. J Solid State Electrochem 2015;19(11):3381e8. [48]