Implementation Of ECC For Image And Generation And Verification Of Digital Signature Using ECDSA

[1]G Shruthi Sastry,[2] Dr. Smitha Sasi
[1]Student, MTech, Digital Communication and Networking, [2]Associate Professor
[1] Dept. of Telecommunication Engineering,[2] Dept. of Telecommunication Engineering
[1&2] Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

Abstract : The components of multimedia such as text, image, and audio and video are integrated and transmitted over a network. The protection of information from the unauthorized access is of high concern in the digital era. The advanced cryptographic techniques are employed in order to secure the multimedia information to preserve the data confidentiality and to share the information only with the intended users. The paper proposes encryption and decryption of an image based on the points on the elliptic curve. The pixels of the image are taken in the binary form and linked to a point say, (Xi, Yi) on the elliptic curve. An image is encrypted using the elliptic curve points and successfully decrypted by utilizing the same elliptic curve in Python 2.7 version operating on the Ubuntu operating system.

***Index Terms* - ECC, ECDSA, point addition, point multiplication, curve points, encryption, decryption, signing, verification, digest, SHA-1, image.**

I. INTRODUCTION

Elliptic Curve Cryptography. It is one of the public key cryptographic techniques which have vital application in securing of data in the form of encryption. In public key cryptography every client participating in the data transaction has a couple of keys, namely, a public key ECC is an abbreviation for and a private key. The cryptographic computations are performed by these keys [6]. The private key is known to the specific user while the public key is communicated to all the users in the network. The public key algorithm needs a lot of predefined constants to be known by all the devices participating in the communication such as domain parameters in ECC. The numerical operations of ECC is characterized over the elliptic curve,

$$y^2 = x^3 + ax + b \qquad (1)$$

Where,

$$4a^3 + 27b^2 \neq 0. \qquad (2)$$

Each estimation of the 'a' and 'b' gives an alternate elliptic curve. All points (x, y) which Fulfils the above equation in addition to a point at infinity lies on the elliptic curve. The public key is a point in the curve and the private key is a random number. The public key is acquired by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters 'a' and 'b', together with couple of more constants comprises the domain parameter of ECC. One fundamental preferred standpoint of ECC is its little key size. A 160-bit key in ECC is viewed as verified as 1024-bit key in RSA. Elliptic curve cryptography makes usage of elliptic curves in which the factors and coefficients are through and through restricted to segments of a limited field [1]. In ECC, the encryption system begins with an affine point called Pm(x, y). These centers may be essentially the base point (G) or some other point close toward the base point. Base point suggests that it has the smallest(x, y) co-ordinates, which satisfy the EC.

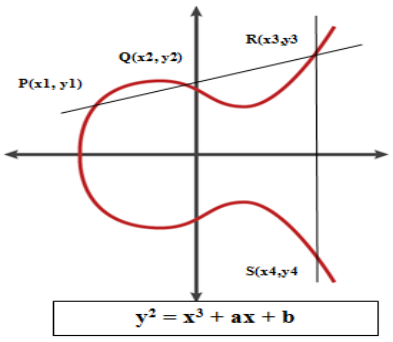

Figure 1: Ideal model view of an Elliptic Curve

II. LITERATURE REVIEW

For the finite field Fp the Elliptic curve has domain parameters p, a, b, G, n and h. p is the prime number characterized for finite field Fp. a and b are the parameters characterizing the curve y2 mod p= x3 +ax + b mod p [2]. G is the generator point (xG, yG), a point on the elliptic curve picked for cryptographic operations. n is the order of the elliptic curve. The scalar for point multiplication is picked as a number among 0 and n – 1. h is cofactor, where h = E(Fp)/n. E(Fp) is the number of points on an elliptic curve. The equation of the elliptic curve on a binary field is y2 + xy = x3 + ax2 + b, where b ≠ 0. Here the components of the finite field are numbers of length at most m bits. These numbers can be considered as a binary polynomial of degree m – 1. In binary polynomial the coefficients must be 0 or 1. The entire task, for example, expansion, subtraction, division, multiplication includes polynomials of degree m – 1 or lesser. The m is picked with the end goal that there is finitely huge number of points on the elliptic curve in order to secure the strength of cryptosystem. The numeric and alphabetic character in the plaintext is expressed by its ASCII value [8]. Considering points on the elliptic curve, the ASCII value is mapped to the point on the curve for encryption. Utilization of affine points for the purpose of plain text transformation to its ASCII value it is benefit of this approach. This transformation's purpose is two folds. First, a single digit of the character's ASCII integer is converted into a set of coordinates to confine within the Elliptic curve. Secondly, the non-linearity is brought out by the transformation in the character covering its identity. The ECC technique is incorporated to encrypt the data in the form of a transformed cipher text. Without the knowledge of private key, the decryption of the original information is infeasible. [3].

RSA is considered as the first real life and practical asymmetric-key cryptosystem. It becomes de facto standard for public-key cryptography. Its security lies with integer factorization problem. RSA's decryption process is not efficient as its encryption process. The comparison of RSA and ECC is made by performing encryption and decryption on three example input information of 8 bits, 64 bits, 256 bits with random keys dependent on NIST suggestion. In light of experimentation, it was discovered that ECC beats RSA in regards to operational proficiency and security with lesser parameters. An ECC is especially most appropriate for devices with limited resources. Compared to RSA, the pith of ECC is that it appears to provide better security for a smaller key size, thus reducing overhead processing [13]. Chip cards, electronic commerce, web servers and cell phones are included in such applications. One of the applications for which the ECC can be used is for large image file encryption [4]. Higher speeds, lower power consumption, bandwidth savings, storage efficiencies, and smaller certificates are the benefits of this higher-strength per-bit. In applications where storage, bandwidths, capacity and power are limited these advantages are particularly beneficial. The ability of RSA is insufficient due the prerequisite of expansive number of bits. RSA calculation can be moderate in situations where extensive data should be encoded by a similar PC. It requires a third party to confirm the dependability of public keys. Data exchanged through RSA calculation could be undermined through go betweens who may temper with the public key framework [5, 6].

III. METHODOLOGY

Elliptic curve digital signature algorithm.

ECDSA Python module is capable of handling and getting implemented into several protocols and has the proficiency of generating the keys and signatures of relatively shorter lengths in contrast to other techniques taking minimal time. There is a rich set of resource of random numbers for formation of keys. The signing key and the verifying key have correspondence with the particular standard elliptic curves defined in the library. The security offered by the curves is high for longer lengths of the keys and signatures. ECDSA comprises of two stages, namely, signature generation and signature verification. The signature and the message are sent to the receiver. As the receiver has public key of the sender and its own private key, it is capable of verifying the signature and thus proves whether the information is received from the trusted user or not [7, 8, 9].

Basics of an image encryption with ECC

Image encryption utilizing ECC is performed by mapping the value of pixel to Elliptic curve axis. Pixel estimation of image in the form of a byte bound between 0 and 255 [3]. For mapping, a different lookup table is required. Point multiplication of base point and pixel is carried out in order to get the cipher image. With the help of the table, the image's pixel values are mapped onto EC. The public key of the receiver is used for encryption. The encrypted image is sent to the receiver. The same lookup table consisting of pixel values running from 0 to 255, associated with the curve points is referred for retrieving the original image at the receiver end [5]. In such cases, the mapping table is required to generate the corresponding pixel value from the cipher image during the decryption process. A group of pixels is considered to reduce the number of computations. The pixel group is transformed into large integer single digits; bearing in mind that the value' p' which is one of the parameters in the elliptic curve equation of the finite field should not be exceeded [10]. These large integer values are paired and given in the ECC operation as input denoted by 'Pm.' This operation helps to ignore the operation of mapping and the need to share mapping table between sender and receiver [11].

Procedure for encryption and decryption of an image

1. Consider an image as input X with dimension M×N .
2. The picture elements of the input image are referred to as 'message' that is denoted by the letter 'm'.

3. 'm' is transformed into a point on the curve (Xi,Yi), which is denoted as Pm = (Xi, Yi).
4. All the pixels are similarly mapped to the (x,y) coordinate pairs that are generated by the procedures of operations on the elliptic curve. By substituting the x value in the elliptic curve equation, the corresponding value of 'y' can be obtained from the general equation, $y^2 = \{(x3 + ax + b) (\text{modulo } P)\}$ where, P is the large prime integer.
5. The sender chooses a private key Ka; a large integer the receiver chooses a private key Kb; a large integer.
6. A generator point G = (Gx, Gy) is noted from the curve.
7. The sender derives a public key Pa by multiplying its private key with the generator point, Pa = Ka * (Gx , Gy).
8. The receiver derives a public key Kb by computing the product of its private key with the generator point, Pb = Kb * (Gx, Gy).
9. The public keys of both the users are shared among them. While Pm forms the mapped data of the original message to be encrypted, the cipher data Pc is formulated as follows, Pc =(Pa , Pm+ Ka Pb) =Pc (X , Y) . This is the encrypted data that the sender transmits to the receiver. At the receiver, the original information can be retrieved by applying receiver's private key Kb. Multiply the first point with receiver's private key Kb and add it to second point.
   Pm = (Pm+ Ka PB - Kb Pa) = Pm (Xi ,Yi)

Flowchart

The process of encryption, decryption, signing and verification are depicted in the figure 2 below.

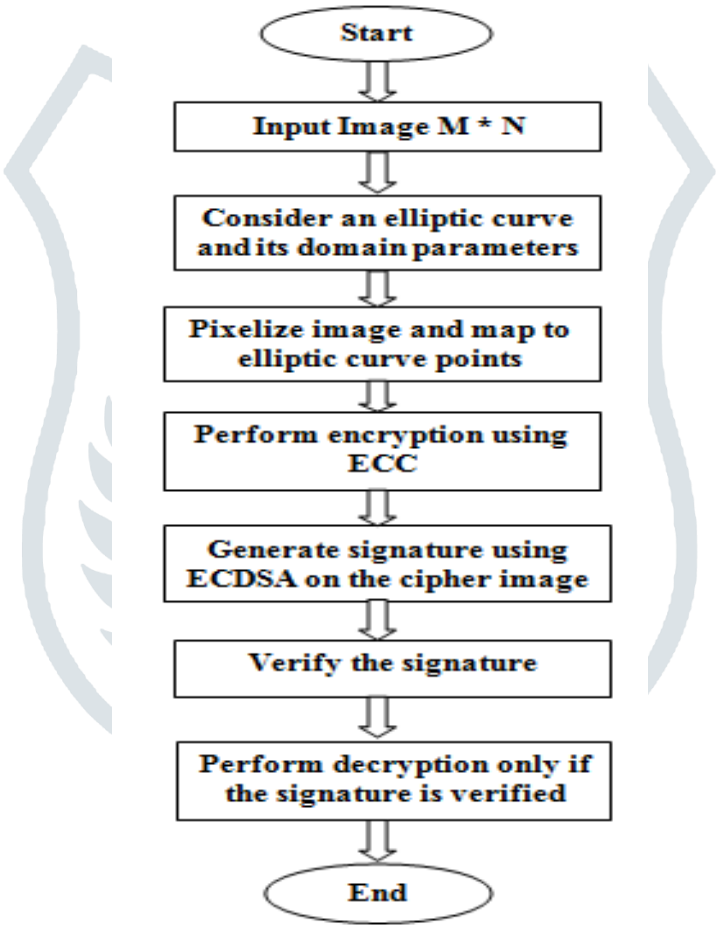

Figure2: Flowchart of proposed cryptosystem

Algorithm for ECDSA Signature Generation

1. pick a random number k in the interval, $1 \le k \le \{p - 1\}$
2. Find the point k* (Gx, Gy) = (X1,Y1)
3. Find r = X1(mod P).
4. if r=0 then
   Go back to step 1.
   End if
5. Find, k−1(mod P) .
6. Apply SHA-1 on cipher text Pc and get the integer equivalent of it say e. .
7. Calculate s = k−1(e +Ka * r).
8. if s = 0 then
   Go back to step 1
   end if

9. For encrypted data Pc, the signature is given by, S = (r,s).

Algorithm for ECDSA Signature Verification

1. Check if r and s belong to the set of integers ranging in the interval (1,P-1).
2. Find SHA-1 of the encrypted data Pc and the resulting value is converted to integer e.
3. Find the value, $w = s^{-1} \pmod P$
4. Find out, $u_1 = e*w \pmod P$ and $u_2 = r*w \pmod P$.
5. Calculate, $X = u_1Pb + u_2Pa$
6. if X = 0
   then
   Disagree S
   else
   Calculate the value, $V = x_1 \pmod P$
   end if
7. Agree only when V == r.

Implementation

The entire cryptographic system is implemented in Python 2.7. The image is read as binary and converted into pixels. A NIST standard prime elliptic curve P-384 is considered for encryption. The domain parameters of the curve specifically, the constants 'a' and 'b' modeling the elliptic curve are of length 384 bits [12]. The prime number 'P' and the order of the curve 'n' are also of length 384 bits long. The total number of points present on the considered P-384 curve is the order ' n' of the curve. The domain parameters 'a', 'b' and 'P' consists of 96 hexadecimal digits summing to 384 bits (96 * 4 = 384). The value of cofactor of P-384 prime curve is one. The various curve points are mapped to the pels of the image. With the help of Python modules 'pyecc', 'ECDSA' and 'hashlib', the encryption and decryption are performed for image data. Also, digital signature is applied for the encrypted data for authentication. The key pairs are created for signing. SHA-1 algorithm is applied to the encrypted content of data. This process converts the string of arbitrary length into a digest that is of fixed length. This resulting content is converted into integer or hex value. The resultant value of the digest generated consists of 40 hexadecimal digits summing upto 160 bits (40 * 4 = 160). The signature is generated for this value of the digest with the help of methods defined in the class of 'ECDSA' python library. For the purpose of verification of this signature generated, first load the verifying key and signature. SHA-1 algorithm is again applied on the received encrypted data Pc and the digest obtained is converted into integer or hex value. By using the 'verify' method defined in the class of 'ECDSA' module, the signature is compared against the digest computed. On attaining match of this value, the authentication is said to be accomplished.

IV. RESULTS

The image is successfully encrypted and decrypted in Python 2.7 operating in Ubuntu platform with the utilization of Python libraries that are featured up with the functionalities supporting the mathematical and logical computations essential for carrying out cryptographic procedures.

Execution procedure and outputs

Open the terminal and set the path where the code for encryption process is saved on the system. Run the command to execute the code in the terminal. A GUI appears on the screen. The followings steps are followed.

1. Click the button 'Generate Keys'. The keys are generated and stored in files.
2. Browse and select the proper files in the following tabs,
   - Key filename : eccPub.pkl
   - Curve filename : eccCurve.pkl
   - Input filename : cheetah.jpg

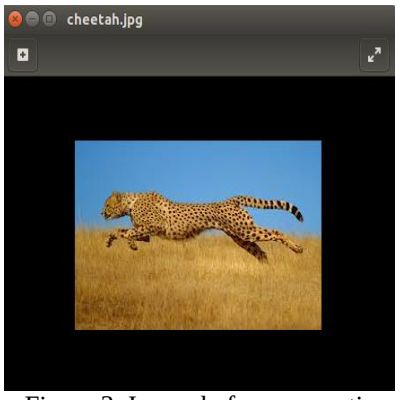

Figure 3: Image before encryption

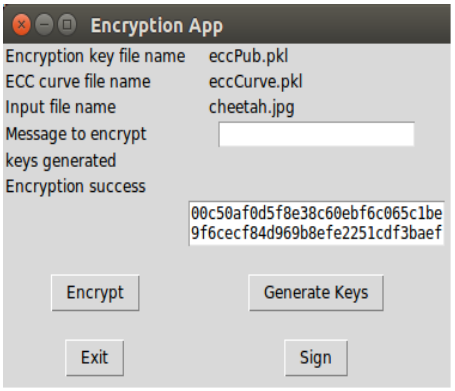

Figure 4: Encrypted image data in HEX format

3. Click on 'encrypt' button. The encryption process is carried out and the encrypted data is displayed in the tab as shown in figure 4 in GUI.Click on 'sign' button.
4. The digital signature is generated for the encrypted content using ECDSA. The figure 5 shows the pop up message box indicating that the digital signature is successfully generated.

Figure 5: Signature generated by the sender

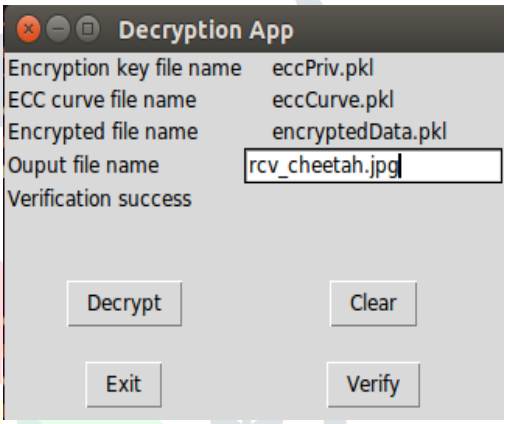

Figure 6: Specifying the filename to store the decrypted image.

5. Click on 'exit' button. This completes the process of encryption. The encrypted data is stored in the file encryptedData.pkl.
6. Run the command to execute the code that performs decryption mechanism. A GUI appears. Browse and select the following files in the tabs.
   - Key filename : eccPriv.pkl
   - Curve filename : eccCurve.pkl
   - Encrypted filename : encryptedData.pkl
7. Enter the output filename as rcv_cheetah.jpg as shown in figure 6.
8. Click on the button 'verify'. A message box pops up and confirms if the signature is verified as shown in figure 7.

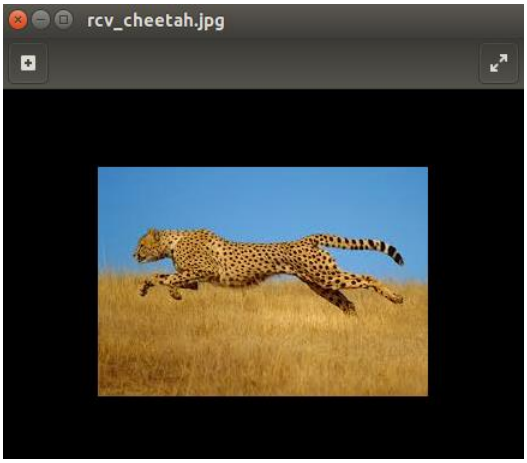

Figure 7: Successful verification of signature by the receiver. Figure 8: Decrypted image at the receiver side

9. Click on the button 'decrypt'.
10. The decrypted image is saved with the given rcv_cheetah.jpg filename. The retrieved image is viewed by opening the file rcv_cheetah.jpg as shown in the figure 8.

V. CONCLUSION

The Elliptic curve cryptographic technique offers elliptic curve digital signature algorithm in addition to encryption mechanism. This feature provides authentication for high degree of security and confidentiality , which ensures that the data is sent from the specific trusted source. An image considered is encrypted with the ECC technique over a prime curve p-384. Hash function is applied for the encrypted data to resize it to fixed length. A digest of length 160 is obtained and the signature is generated for this digest. The verification of the signature is carried out at the decrypting end and the original image is retrieved only if the signature is verified. Thus, the proposed system checks the authenticity of the source along with performing encryption and decryption processes.

REFERENCES

[1] Kamlesh Gupta and Sanjay Silakari , " An ethical way for image encryption using ECC" , First International Conference on Computational Intelligence, Communication Systems and Networks, 2017.
[2] Maria Celestin Vigila, K. Muneeswaran, "Implementation of Text based Cryptosystem using Elliptic Curve Cryptography ", IEEE 2015.
[3] Balamurugan.R and Kamalakannan.V, "Enhancing Security in Text Messages Using Matrix based Mapping and ElGamal Method in Elliptic Curve Cryptography ", IEEE 2018.
[4] Kristin Lauter, "The Advantages of Elliptic Cryptography for Wireless Security",IEEE Wireless Communications, pp. 62-67,Feb.2016.
[5] Laiphrakpam Dolendro Singh and Khumanthem Manglem Singh, " Image Encryption using Elliptic Curve Cryptography" , Eleventh International Multi-Conference on Information Processing", 2015 (IMCIP-2015)
[6] Nissa Mehibel and Mohammed Hamadouche, " A new algorithm for a public key cryptosystem using elliptic curve", 2017 IEEE European Conference on Electrical Engineering and Computer Science.
[7] Ravi Kishore Kodali ," Implementation of ECDSA in WSN", International Conference on Control Communication and Computing (ICCC), 2016
[8] Li Hui-na and Ping Yuan, " A simple limited one time authorization mechanism based on ECDSA" , ICACT 2012.
[9] Yoppy Sazaki Megah Mulya, " The development of android based SMS security using ECDSA with Boolean permutation", IEEE 2016 .
[10] Zhang Chuanrong, Chi Long ," Secure and efficient generalized signcryption scheme based on short ECDSA", IIHMSP 2018.
[11] Karthik Kedarisetti and Roopesh Gamini, " Elliptic curve cryptography for images using fractal based multiple key hill cipher", ICECA 2018.
[12] Certicom, "Standards for Efficient Cryptography", SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2016.
[13] Dindayal Mahto and Dilip Kumar Yadav, " RSA and ECC: A Comparative Analysis ",International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017)