Dual-Layer Video Encryption Using RSA and ECC Algorithm

Prity Kumari[1] , Upendra Kumar[2], Shyam Krishna Singh[3]

[1]Magadh University, Bodh Gaya, India
[2]Birla Institute of Technology, Patna Campus, Patna, India,
[3] A. N. College, Patna, India

Abstract- Cryptography is the study of techniques for ensuring the secrecy and authentication of the information. In this paper use two algorithm RSA and ECC cryptography to encrypt and decrypt the video file. Elliptic Curve Cryptography recently gained a lot of attention in industry. The principal attraction of ECC compared to RSA is that it offers equal security for a smaller bit size, thereby reducing processing overhead .In RSA we firstly generate key pair and secondly encryption decryption algorithm to encrypt and decrypt the video then apply ECC algorithm to encrypt/decrypt of encrypted/decrypted RSA video. It gives more and more security to our video file. Elliptic curve arithmetic can be used to develop a variety of elliptic curve cryptographic (ECC) schemes including key exchange, encryption and digital signature. The mathematic background of ECC is more complex than other cryptographic systems just as Geometry, abstract algebra, number theory. ECC provides greater security and more efficient performance than the first generation public key techniques (RSA and Diffie-Hellman), Mobile systems, Systems required high security level (such as 256 bit AES).

Index Terms- Video encryption/decryption; RSA ; ECC

I. INTRODUCTION

Elliptic Curve Cryptography (ECC)[1] and RSA is an approach to public key cryptography. Elliptic curves (EC) were suggested for cryptography by Victor Miller and Neal Koblitz in 1985 in the form of Elliptic Curve Cryptography (ECC) and RSA system was invented by three scholars Ron Rivest, Adi Shamir, Len Adleman and hence it is called RSA cryptography. The most popular public-key algorithm over the past twenty years has been RSA. The security of RSA comes from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large prime numbers. These numbers may be 100 or 200 digits or even larger. Recovering the message from the public key and the encrypted message is thought to be equivalent to factoring the product of the two prime numbers.

Recently, however, with the advent of faster computers, RSA has become susceptible to brute force attacks. A brute force attack on an encryption system such as RSA means that every possible combination is tried until the correct solution to the problem is achieved. Therefore, new systems are needed to maintain the security of private information. One new method for encryption to solve this problem is called elliptic curve encryption.

ECC based on the algebraic structure of elliptic curves over finite fields. With smaller key sizes and lower processing requirements than other public key cryptosystems, elliptic curve cryptography lends itself well to sending information securely over the internet where bandwidth and processing capabilities are limited. Ensuring the timely and reliable access to make use of information.ECC offers security with smaller key sizes, faster computation, lower power consumption as well as memory and band width saving .This is especially useful for mobile devices, wireless pagers which are limited in bandwidth, memory and low Power and network connectivity. The mobile apps such as multimedia messages, whatsapp are new standards in mobiles era used for communication.

Elliptic Curve Cryptography (ECC) [2,3,6,7] makes use of the elliptic curve in which the variables and coefficients are all restricted to elements of the finites fields [10]. Two families of elliptic curves are used in cryptographic applications: Prime curves [9,15] over F_p and binary curves $F_2{}^m$. For a prime curve over F_p , we use a cubic equation in which the variables and the coefficients all take on values in the set of integers from 0 through p-1 and the calculations are performed with respect to modulo p. Elliptic Curve Cryptography provides an excellent solution not only for the data encryption but also for the secure key transport between two communicating parties [16], and authentic session key establishment protocols [6,11,13]. Moreover, because of the apparent hardness of the underlying elliptic curve discrete logarithm problem (ECDLP), ECC systems are also well suited for applications that need long-term security requirements.

Elliptic Curve Cryptography (ECC) is a public key technology that offers performance advantages at higher security levels. Every user taking part in public key cryptography will take a pair of keys, a public key and a private key. Only the particular user knows the private key whereas the public keys are distributed to all users taking part in the communication. Some public key algorithm may require a set of predefined constants to be known by all the devices taking part in the communication. In ECC we call these predefined constants as 'Domain Parameters'. Understanding ECC needs full mathematical background on elliptic curves. Elliptic curves are not ellipses. The general cubic equation of elliptic curves is $y^2+axy+by=x^3+cx^2+dx+e$. But for our purpose it is sufficient to limit the equation to the form $y^2 = x^3 + ax + b$. Say $E_P(a,b)$ consisting of all the points (x,y) that satisfy the above equation together with element at infinity O. A group can be defined based on the set $E_P(a,b)$ for specific values

of a and b[8]. If P,Q, R are points on $E_P(a,b)$ the relations commutatively, associatively, existence of an identity element and existence of inverse hold good[4]. The heart of ECC is discrete logarithm problem that can be stated as "it should be very hard to find a value k such that **Q=kP** (scalar multiplication) where P and Q are known'. But 'it should be relatively easy to find Q where k and P are known' P, Q are points on the elliptic curve [5].

II. RELATED WORK

In today's modern day where we depend on video transmission for news and media, protecting digital video from attacks during transmission is of prime importance. Due to the large size of digital videos, they are usually compressed and then transmitted using formats like MPEG.

The Naïve Algorithm is the simplest way to encrypt every byte in the whole MPEG stream using standard encryption schemes like DES or AES. The idea of the Naïve algorithm is to treat the MPEG bit-stream as text data and does not use any special structure [17,18]. It ensures the security as no known can effectively break the AES or the triple DES encryption if a sufficiently large key space is used [14].

Quiao and Nahresdt introduced a new algorithm called Video Encryption Algorithm (VEA) [17]. It is based on the statistical properties of the MPEG video standard and symmetric key algorithm which reduces the amount of data encrypted [14]. VEA divides the input video stream into two chunks (a1, a2, a3, a4, a5, a2n-1, a2n). These chunks are further divided in to data segments into even list (a2, a4, a6…a2n) and odd list (a1, a3, a5..a2n-1). After this, an encryption key is applied to the even list, E (a2, a4, a6…a2n), where E denotes the Encryption function used. The resultant encoded list is XORed with the odd list and the concatenated result is the final cipher text. As a result, the VEA is protected from known-only plain text attack as each frame will have a different key [14].

Four new algorithms were introduced by Bhargava, Shi and Wang [19,20]. The algorithms are Algorithm 1, Algorithm 2 (VEA), Algorithm 3 (MVEA), and Algorithm 4 (RVEA). Algorithm 1 uses the permutation of the Huffman code words in I-frames, combining encryption and compression in a single step. A secret permutation p is used which is used to permute standard MPEG Huffman code word list.

The AEGIS technique was developed by Maples and Spanos [21,22]. It only encrypts the I-frame of the MPEG stream. The sequence header is also encrypted which makes the MPEG video stream unrecognizable. The MPEG bit-stream is further hidden by encrypting the IOS end code. The DES encryption is used for the entire process.

III. PROPOSED SYSTEM

RSA encryption, named for the surnames of the inventors, relies on multiplication and exponentiation being much faster than prime factorization. The entire protocol is built from two large prime numbers. These prime numbers are manipulated to give a public key and private key. Once these keys are generated they can be used many times. Typically one keeps the private key and publishes the public key. Anyone can then encrypt a message using the public key and sent it to the creator of the keys. This person then uses the private key to decrypt the message.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below −

- **Generate the RSA modulus (n)**
  - Select two large primes, p and q.
  - Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.
  - Totient of n=□(n)=□□(p)*□□(q)=(p-1)(q-1)
- **Find Derived Number (e)**
  - Number **e** must be greater than 1 and less than (p − 1)(q − 1).
  - There must be no common factor for e and (p − 1)(q − 1) except for 1. In other words two numbers e and (p – 1)(q – 1) are co-prime.
- **Form the public key**
  - The pair of numbers (n, e) forms the RSA public key and is made public.
  - Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.
- **Generate the private key**
  - Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
  - Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1).
  - This relationship is written mathematically as follows −

Ed= 1 mod (p-1)(q-1)

The pair of number (n,d) forms the RSA private key.

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.

- **RSA Encryption**

$C = P^e \mod n$, where C is ciphertext and P is plaintext.

- **RSA Decryption**

$P = C^d \mod n$

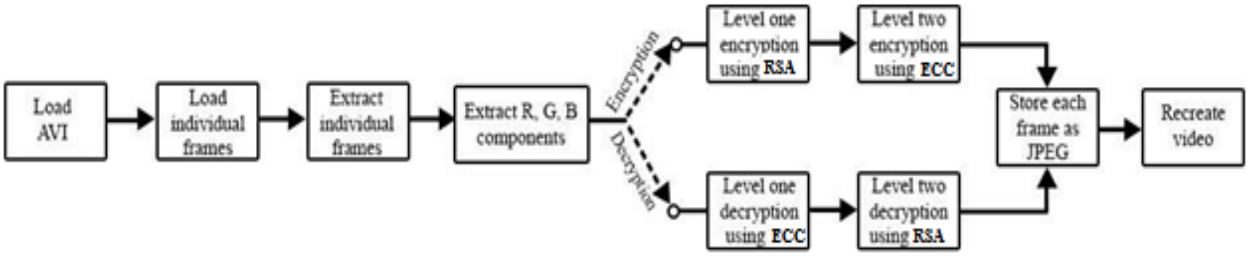


Fig. 1 Process of Encryption and Decryption

In the public key elliptic curve cryptosystems, we assume that entity A wants to send a message m to entity B securely. Order of a point on the curve can be defined as a value n such that nP = P+P+..+P.. n times = O (infinity)[9].

Key generation:

Both the entities in the cryptosystem agree upon a,b,p,G,n which are called 'Domain Parameters' of ECC[3]. G is called generator point and n is the order of G. Now A generates a random number $n_A < n$ as his private Key and calculates his public key $P_A = n_A G$. B generates a random number $n_B < n$ as his private Key and calculates his public key $P_B = n_B G$.

Key Exchange:

Entity A computes his Shared Key by Computing $K = n_B P_A$
Entity B computes his Shared Key by Computing $K = n_A P_B$
The two above keys have same value because:
$n_A * P_B = n_A * (n_B * G) = n_B * (n_A * G) = n_B * P_A$

Encryption:

A sends $C_m = 2$ ciphertext points those are $\{ kG, P_m + k P_B)\}$.
Where G - generator Point
P_m - plaintext point on the curve
k - a random number chosen by A
P_B - public key of B

Decryption:

$P_m + kP_B - n_B(kG) = P_m + k(n_B)G - n_B(kG) = P_m$

IV. VIDEO ENCRYPTION/DECRYPTION

Steps of encryption

(1) The AVI file is loaded into the system.

(2) The frames of the file loaded, is extracted one by one.

(3) After the extraction process, the frames are loaded.

(4) The loaded frames are then segregated into their RGB components and the encryption takes place on the individual color components of the frame.

(5) The RGB frames are encrypted individually using the ECC algorithm.

(6) To initiate the RSA process, accept an input string from the user. The sum of the ASCII values of each character of the string input by the user is stored as *x*. Two large consecutive prime numbers are selected which are immediately next to *x* and pass them on as inputs to the RSA algorithm.

(7) Key distribution takes place.

(a) The public key is sent from the sender to the receiver.

(b) The receiver then sends a message *M* to the sender.

(c) The message *M* is first converted into an integer m, such that $0 < m < n$ by using an agreed-upon reversible protocol known as a padding scheme.

(d) The cipher text c is calculated corresponding to $C_{RSA} = M^e (mod\ n)$.

(8)To initiate the ECC process, accept the value of C_{RSA}.

(9) Encode this value to a point on the elliptic curve.

Koblitz's Method for Encoding Plaintext [12]:

Step1: Pick an elliptic curve Ep(a,b).

Step 2: Let us say that E has N points on it.

Step 3: Let us say that our alphabet consists of the digits 0,1,2,3,4,5,6,7,8,9 and the letters A,B,C,. . . , X,Y,Z coded as 10,11,. . . , 35.

Step 4: This converts our message into a series of numbers between 0 and 35.

Step 5: Now choose an auxiliary base parameter, for example k = 20. (both parties should agree upon this)

Step 6: For each number $C_{RSA}k$ (say), take $x = C_{RSA}k + 1$ and try to solve for *y*.

Step 7: If you can't do it, then try $x = C_{RSA}k + 2$ and then $x = C_{RSA}k + 3$ until you can solve for *y*.

Step 8: In practice, you will find such a *y* before you hit $x = C_{RSA}k + k - 1$. Then take the point
(x,y). This now converts the number C_{RSA} into a point on the elliptic curve. In this way, the entire message becomes a sequence of points.

(10) The cipher text point (x,y) is then sent from the receiver to the sender.

(11) The encrypted RGB components are then combined as a JPG file (with a frame number in the filename).

(12) Steps 3 to 9 for all frames are repeated for all the frames.

(13) The result is obtained after utilizing all the stored frames to create a video file with each stored encrypted image as an individual frame of the video.

Steps of decryption

(1) The encrypted AVI video file is loaded into the system.

(2) The frames are extracted one by one.

(3) Each frame is loaded in the system.

(4) The RGB components are then extracted from the loaded frames.

(5) The RGB components are decrypted using ECC.
Consider each point (x,y) and set C_{RSA} to be the greatest integer less than $(x\text{-}1)/k$. Then the point (x,y) decodes as the symbol C_{ECC} .

(6)After decryption by ECC, the ECC components are then decrypted using RSA.

(7) The sender recovers m from C_{ECC} by using the private key exponent via computing $m = C_{ECC}^{\,d}$ *(mod n).*

(8) Each decrypted RGB component is then combined as a JPG file (with a frame number in the filename).

(9) Steps 3 to 7 are repeated for all frames.

(10) The result is obtained after utilizing all the stored frames to create a video file with each stored decrypted image as an individual frame of the video.

Example:

Let us take p and q values are very small. Practically, these values are very high.

- Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.
- Select e = 5, which is a valid choice since there is no number that is common factor of 5 and $(p - 1)(q - 1) = 6 \times 12 = 72$, except for 1.
- The pair of numbers (n, e) = (91, 5) forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.
- Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d = 29. (5.d=1 mod 72, starting with 2 until will reach 5^k mod 72=1. Doing so we obtained5^6 mod 72=1, which implies that 5^5 mod 72=29 is 5^{-1} in U(72)[i.e e^{-1}=d in U(72)])
- Check that the d calculated is correct by computing –

$de = 29 \times 5 = 145 = 1 \bmod 72$

- Hence, public key is (91, 5) and private keys is (91, 29).
- Say we have to send character ‘a’. ‘A’ is first encoded as number 10.
- Plaintext P = 10, we get ciphertext C_{RSA} –

$C_{RSA} = 10^5 \bmod 91 = 82$

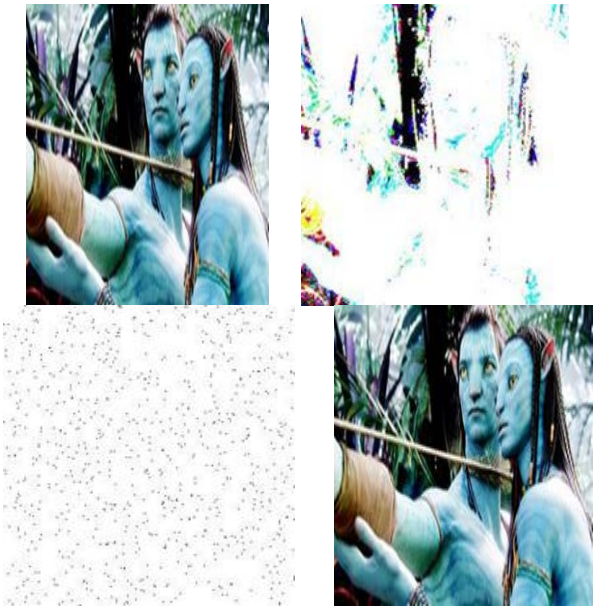
- Say the parameters of curve are: p = 751, a = -1, b = 188, n = 727.
- x=mk+1 i.e. 82*20+1=1641cannot solve it for a y such that $y^2 \bmod p = x^3 + ax + b \bmod p$, where $m = C_{RSA}$.
- So go for x=mk+2 , x=1642 , no y exists. x=mk+3, x=1643, no y exists. x=mk+4, x=1644, no y exists. x=mk+5, x=1645, no y exists.
- x=mk+6 so x=1646 can solve it for y and y=1692.
- Now the point (1646,1692) is point is encrypted and decrypted as a message.
- To decode just compute P_{ECC} = (x-1)/k i.e. (1646-1)/20=1645/20 i.e. 82.25.
- Return 82 as ECC plaintext P_{ECC} (greatest integer less than (x-1)/k ,that is 82.
- After decrypting m to P_{ECC} it again decrypt by RSA decoder i.e.

$P_{RSA} = 82^{29} \bmod 91 = 10$ = original plaintext.

- The number 10 is now decoded to character ‘A’.

V. RESULT AND ANALYSIS

We select one frame of MPEG video and using RSA algorithm first to encrypt video then after we use ECC algorithm to again encrypt the encrypted video. After encryption we get figure c. Then we decrypt the encrypted image through ECC algorithm. We get original image after decryption of decrypted video using RSA decryption algorithm. The results of encryption and decryption are shown below:



(a) (b) (c)

(d)

Fig. 2 Video Encryption and Decryption

VI. HOW IS IT BETTER IN EXISTING WORK

Upon implementing RSA encryption for the video, the encrypted video turned out to visually resemble the original video, we can see it clearly in figure 3 below that the encrypted image bears some similarities to the original image. For highly sensitive data it is imperative that the encrypted video should not bear any visual resemblance to the original video. Hence, we try to remove any resemblance whatsoever from the encrypted video to the original video, to do this we propose the use of two levels of encryption.

Fig. 3 R – Component after RSA encryption

our proposed system the first level encryption is obtained by RSA algorithm sequence whereas, level 2 encryption is obtained by using ECC algorithm. The result of combining these two levels is shown in below figure.

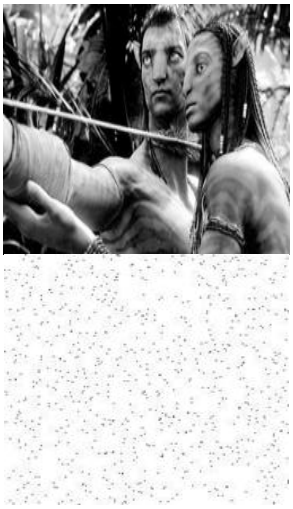


Fig 4 : Comparison of image before and after 2 levels of encryption

From above figure 4 we can see that there is no resemblance between the original and encrypted image, which is essential to provide maximum security.

VII. CONCLUSION

We have proposed a new method of dual layer encryption methodology which enables to achieve zero visual resemblance and high security. This is achieved by applying the techniques individually rather than encrypting and decrypting all the frames in one go. The dual layer approach presents a promising approach to achieving a highly secure way of video encryption while not being very computationally intensive and time consuming.

REFERENCES

[1] Certicom, Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography, Version 1.0, September 2000, Available at http://www.secg.org/download/aid-385/sec1_final.pdf

[2] Anna M. Johnston, Peter S. Gemmell, “Authenticated key exchange Provably Secure Against the Man-in-Middle Attack”, Journal of Cryptology (2002) Vol. number 2 pages 139-148.

[3] Antoines Joux, “A one round protocol for Tripartite Diffie-Hellman”, Journal of Cryptology, 2004, Volume 17, Number 4,pages 263-276.

[4] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[5] William Stallings, Cryptography and Network Security, Principles and Practice. ed.,Prentice Hall, New Jersey,2003.

[6] Asrjen K. Lenstra and Eric R. Verheul, “Selecting Cryptographic key size”, Journal of Cryptology, 2001, Volume-14, Number 4, pages 255-293.

[7] A. Chandrasekhar et.al. “Some Algebraic Curves in public Key crypto systems”, International Journal of Ultra Scientists and Physical Sciences, 2007.

[8] R. Schoof. Elliptic Curves Over Finite Fields and the Computation of Square Roots mod p. Mathematics of Computation, Vol. 44, No. 170, pp.483-494, April 85.

[9] Darrel Hankerson, Alfered Menezes, Scott Vanstone, “A Gide to elliptic curve Cryptography”, Springer, 2004.

[10] Gura N., Shantz S., Eberle H., et al “An End-to End Systems Approach to Elliptic Curve Cryptography”, Sun Microsystems Laboratories; 2002; Retrieved May, 10, http://research.sun.com/projects/crypto

[11] Kin Choong Yow and Amol Dabholkar, "A Light-Weight mutual authentication and key-exchange protocol based of elliptical curve cryptography for energy-constrained devices, International Journal

[12] N. Koblitz, \Elliptic curve cryptosystems", Mathematics of Computation, 48 (1987), 203-209.ournal of Network Security & its Applications Vol. 2 No. 2 April 2010.

[13] Mohsen Machhout eat.al., coupled FPGA/ASIC implementation of elliptic curve crypto-processor, International Journal of Network Security & its Applications Vol. 2 No. 2 April 2010.

[14] M. Abomhara, Omar Zakaria, Othman O. Khalifa, "An Overview of Video Encryption Techniques", International Journal of Computer Theory and Engineering. Vol.2, 103-110, 2010.

[15] Rosing M. "Implementing elliptic curve cryptography", Greenwich, CT: Manning publications, 1999.

[16] Ch. Suneetha, D. Sravana Kumar and A. Chandrasekhar, "Secure key transport in symmetric cryptographic protocols using elliptic curves over finite fields", International Journal of Computer Applications, , Vol. 36, No. 1 November 2011

[17] S. Aly, "A light-weight encrypting for real time video transmission", CTI Symposium Conference, DePaul University, Chicago, USA, 2003.

[18] C.P. Wu, C.C.J. Kuo, "Design of integrated multimedia compression and encryption systems", IEEE Trans. Multimedia. Vol. 7, 828-839, 2005.

[19] C. Shi and B. Bhargava, "A Fast MPEG Video Encryption Algorithm", Proceedings of the 6th International Multimedia Conference, Bristol, UK, 81-88, 1998.

[20] C. Shi, S.Y. Wang and B. Bhargava, "MPEG Video Encryption in Real-Time Using Secret key Cryptography", International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, NV, 1999.

[21] G. Spanos and T. Maples, "Performance study of selective encryption scheme for the security of networked real-time video", Proceedings of the 4th International Conference on Computer and Communications, Las Vegas, NV, 2-10, 1995.

[22] T. Maples and G. Spanos, "Security for Real-Time MPEG Compressed Video in Distributed Multimedia Applications", Conference on Computers and Communications, 72-78, 1996.

AUTHORS

First Author – Prity Kumari, Magadh University, Bodh Gaya, India

Second Author – Upendra Kumar, Birla Institute of Technology, Patna Campus, Patna, India

Third Author – Shyam Krishna Singh, A. N. College, Patna, India