Hybrid Cryptosystem using Elliptic Curve Cryptography and Caesar Cipher

C. Swetha
Department of Computer Science and Technology
Yogi Vemana University
Kadapa, A.P, India

G. Amruthavani
Department of Computer Science and Technology
Yogi Vemana University
Kadapa, A.P, India

B. Reddaiah
Department of Computer Science and Technology
Yogi Vemana University
Kadapa, A.P, India

ABSTRACT

Security is a global issue, and it is one that requires global attention. Security has many facets. The main aspect of the security problem is privacy. Data privacy and security is not a simple job. So, security is possible with suitable security strategies. Security in addition also involves access control, data veracity, system accessibility, and auditing. In fact, 80% of data thrashing is caused by hackers. Encryption is only one approach to securing data. In this work Caesar cipher combined with Elliptic curve cryptography and product cipher system is developed as one of the solutions for security related problems that global environment is facing. This developed product cipher is quite simple with pattern formula. This combination helps in developing strong system. The role of Elliptic curve is very good in developing applications for protection and digital signatures. It also helps in generating pseudo-random numbers and helps in fast encryption and decryption.

Keywords

Security, Elliptic curve cryptography, Caesar cipher, Symmetric Key, Encryption, Decryption

1. INTRODUCTION

Different organizations and individuals in the world prefer to use different security mechanisms to overcome security related threats. There are mechanisms like firewall, antivirus, antispyware and strong passwords to protect systems. But cryptography is science that deals with securing data while travelling in network. Encryption and Decryption are the two parts of Cryptography. This cryptography is used to hide and unhide information that is intended to travel in network and it is the science of using mathematics. Cryptography allows us to hold delicate data or transfer data through uncertain nets like Internet, so that it can be read by only the intended receiver, not by the others. Cryptography is of three types based on nature of key. The first one is secret key that shares same key between sender and receiver for both encrypting and decrypting. The second is private key that is used by the receiver and third is public key system that is publicized to all the users.

Based on the key usage, cryptographic techniques are commonly divided into two types. They are symmetric key system that uses single key and asymmetric key system that uses double key as shown in the figure 1.

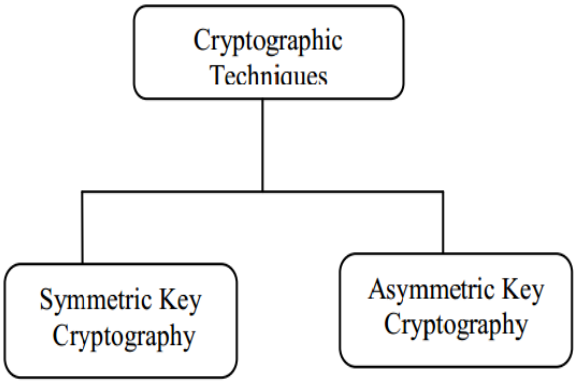

Fig 1: Types of Cryptography

1.1 Classification of Cryptography

Cryptographic technique operates by combining key which may be numerical value or it may be a word or it may be a phrase and plain text to encrypt. This process of encryption carries with the help of different keys to encrypt the same plain text to different secret form of data. As discussed, there are two kinds of Cryptographic methods, one is Symmetrical cryptography. It is the simplest kind of encryption technique that involves only one key for encryption that is used by the sender and the same for decryption that is used by the receiver of information. The other form is Asymmetric cryptography or public key cryptography which applies different keys. One for encryption and other for decryption, namely private key and public key.

1.1.1 Symmetric key Cryptography

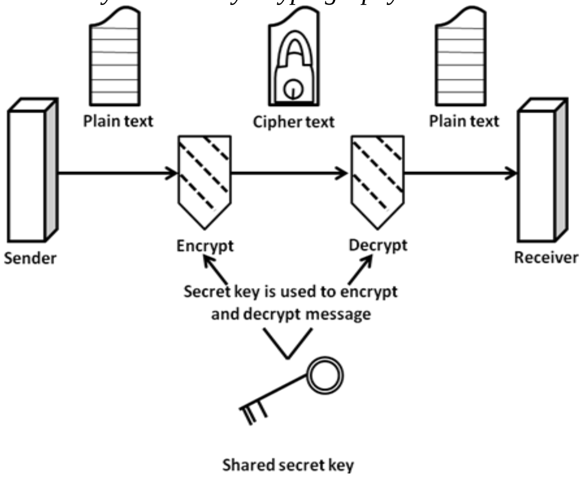

Fig 2: Symmetric key cryptography

Symmetric key cryptography is also known as conventional cryptography where a secret key is attached for both encryption and decryption functions as shown in figure 2.

1.1.2 Asymmetric key Cryptography

Asymmetric cryptography is also referred as public-key cryptography. This procedure practices a pair of interrelated keys. In this pair a key is public key and other is private key used to encrypt and decrypt as shown in figure 3.

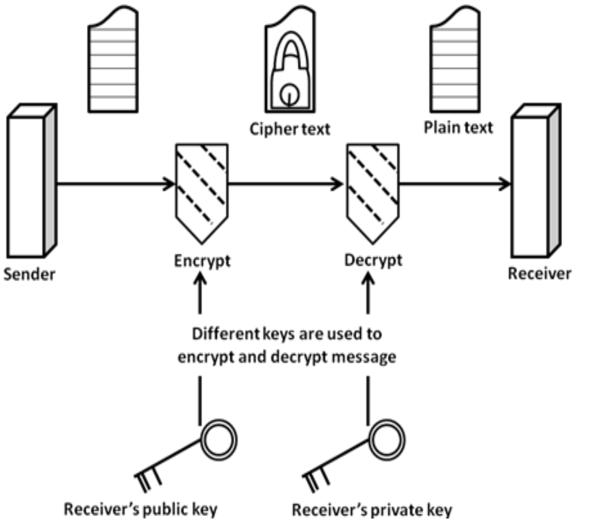

Fig 3: Asymmetric key cryptography

2. BACKGROUND STUDY

Neal Koblitz [2] in 1985 and Victor Miller [3] used elliptic curves in cryptographic systems to develop public key. In Sep'2000 Daniel V. Bailey and Christof Paar [11] in 2000 presented effective mathematics in finite field extensions through the application in elliptic curve cryptography. Elliptic curve cryptography was standardized after 1990 by different organizations. Among the list of organizations ANSI [4, 5], IEEE [4,6], ISO[7, 8], NIST[9, 10] are prominent ones and ECC started getting viable acceptance.

Whitfield Diffie and Martin Hellman [1] discussed about the problems of key distribution and its related issues. Rivest, shamir and Adleman [12] developed new method of publicly available key that is used in cryptography. Miller and Adleman [13] worked on index calculus and work on fields is discussed. Koblitz[14] proposed the simplification elliptical curve to curves of higher genus that are termed as hyper elliptic curves (HEC).

2.1 Caesar Cipher

More than 2000 years ago, Roman empire used to keep the military secrets by using cryptography. Caesar cipher as it is currently called was utilized by Julius Caesar to encrypt messages by shifting letters alphabetically. An integer value is needed to cipher a plain text, called as shift. It indicates numeralpositions each letter of the plain text has been moved down. Caesar cipher is an example for substitution method. Caesar is the first person who ever used encryption for securing messages. Ease of use is one of the strengths of this cipher. This strength is significant for Caesar cipher. This is because uneducated soldiers at that time who could not use complex systems can use this simple system. Further this cipher practices modulo twenty-six for key to encrypt which is more than twenty-six. This cipher is also referred as shift cipher as it moves the text based on key value.

2.2 Elliptic Curve Cryptography

Elliptic curves are mathematical objects that can be used to develop algorithm for cryptography. Elliptic curve cryptography (ECC) helps in achieving better security and it is one of the best solutions for providing better security when the key size is small in size and it is a public key cryptography. Elliptic Curve Cryptography helps in developing security systems with keys of smaller size, helps in quicker computation, helps in using memory, energy and bandwidth in an efficient way. This cryptography gives the impression in providing equivalent security level for a distant smaller key size algorithms as well, thereby reducing processing overhead.

Elliptic curves are represented by cubic equations and they are helpful to assess the boundaries of an ellipse. The form of cubic equations for elliptic curves is y2 + axy + by=x3 +cx2 +dx + e, here a, b, c, d and e are real numbers. Point operations on the Elliptic curve are used for implementation of Elliptic curve cryptographic algorithms. Those point operations are Addition, doubling a point and scalar multiplication. Let E be an elliptical curve over Fp. A general form of Elliptic Curve takes form as $E=y^2 \bmod p=x^3+x+1 \bmod p$. Where x, y are elements of E(Fp) and a, b is modulo p, p is randomly chosen prime number that creates Elliptic Curve finite field. Initially, points are required and generated, to do an operation with the help of elliptic curve points in order to encryption and decryption.

For Elliptic curve decryption the input is an Elliptic curve parameter (P_M, n_B, G, K, P_B), Where P_M plain text, n_B is receiver secret key, P_B is public key. To get plaintext consider the two cipher text points, each letter and calculate by using formula $(P_M + K. P_B) – [(n_B. K. G)] = (P_M + K. n_B. G) – (n_B. k. G)$

3. OUR SCHEME

Cryptography is a scientific way used to secure confidential data from various cyber-attacks. Throughout this work, a notion is intended during which Caesar cipher algorithm is initiated through elliptic curve cryptography to give conventional privacy scope and better privacy. Caesar cipher algorithm is tougher to disrupt because of its linearity. Elliptical curve cryptography provides text-based security by generating points on Elliptic curve over the finite field. By using Caesar cipher, it commences with conversion of plain text then converted into its ASCII value to recommend points on Elliptic curve. Finally, it executes scalar multiplication to hide the given text and to get public and secret key.

4. PROPOSED ALGORITHM

4.1 Caesar Cipher Encryption Algorithm

Step 1: Consider lowercase letters of a String
Step 2: The specified shift must be an integer range between 0-25.
Step 3: Based on key the given traverse text character at a time.
Step 4: Apply the encryption formula where Cipher text = (Plaintext + Key) mod 25.

4.1.1 Encryption Example of Caesar Cipher

Consider the message "BAT" which is plain text.
key should be numeric value which ranges from 0 to 25. The key considered for this example is "4".

We have B=1, A=0 and T=19.

Caesar Cipher encryption formula is Cipher text =

(plaintext + key) mod 25
C(B)= (plaintext(B)+Key) mod 25
C(B)= (1+4) mod 25
C(B) = (5) mod 25
C(B) = 5
5 plaintext is 'F'.
C(A) = (plaintext(A) + Key) mod 25
C(A) = (0+4) mod 25
C(A)=4
4 plaintext is 'E'.
C(T)=(plaintext(T) + Key) mod 25
C(T) = (19+4) mod 25
C(T) = 23
23 plaintext is 'x'.

Finally, Caesar cipher encryption result is: "FEX"

4.2 Elliptic Curve Encryption Algorithm

Step 1: The general form of Elliptic Curve and considered is E: y^2 mod p=x^3+x+1 mod p

where x, y are elements of E(Fp) and a, b is modulo of p, p and randomly chosen prime number creates the Elliptic Curve finite field.

Step 2: Initially, points are required, to generate and to do an

operation with the help of elliptic curve points for encryption and decryption.

Point generation method

Input: a, b, p

Where a, b are integers and p is large prime number.

Output: (x, y)

Compute y^2 =x^3+ax+b mod p

Consider prime number p as 277 and a=1, b=1

Y^2mod 277 = x^3 + x + 1 mod 277.

The points generated on the Elliptic curve as shown in figure 4 in graphical form.

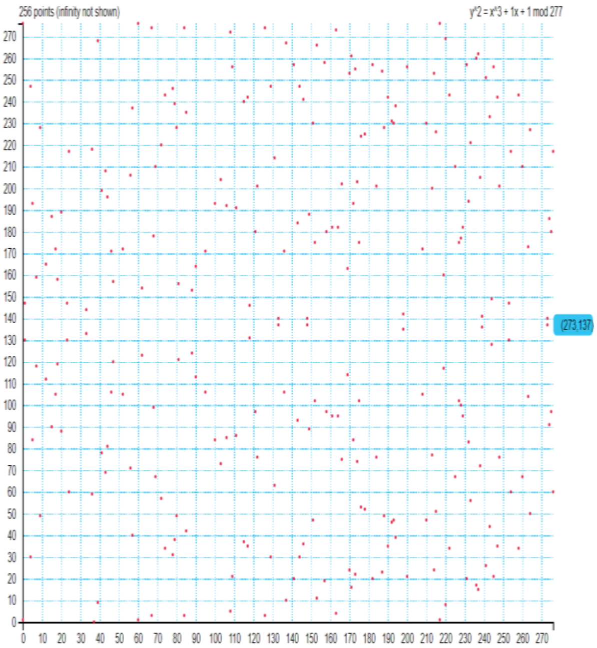

Fig 4: Generating point on Elliptic Curve

Step 3: Encrypt the formula PC= [(K, G), (PM+K.PB)]

4.2.1 *Encryption Example of Elliptic Curve*

Message to be sent from sender to receiver is "FEX"
This message is encoded as the plaintext point PM = (18,158), (241,251), (210,230) € E277 (1,1),
Sender must use receiver`s public key to encrypt it.
Sender`s secret key is nB = 85,
then B`s public key will be:
P_B=nB. G=85. (0,276)
P_B = (237,15)
Sender selects a random number K
K = 113,
and uses receiver`s public key P_B = (237,15) to encrypt the message point into the ciphertext pair of points:
F letter encryption:
plaintext point P_M = (18,158)
P_C = [(K. G)], (P_M + K. P_B)]
P_C = [113 · (0,276), (18,158) + 113 · (237,15)]
P_C = [(260,67), (18,158) + (253,130)]
P_C = [(260,67), (68,178)]
F letter cipher pair points: [(260,67), (68,178)]
E letter encryption:
Plaintext point P_M= (12, 165)
P_C = [(K. G)], (P_M + K. P_B)]
P_C= [113. (0,276), (12, 165) +113. (237,15)]
P_C = [(260,67), (12, 165) + (253,130)]
P_C = [(260,67), (241,251)]
E letter cipher pair points: [(260,67), (241,251)]
X letter encryption:
Plaintext point P_M= (36,59)
Pc= [(K, G), (PM+K.PB)]
Pc= [113. (0,276), (36, 59) +113(237,15)]
Pc = (260,67), (36, 59) + (253,130)
Pc= [(260,67), (210,230)]
X letter cipher pair points: [(260,67), (210,230)]

4.3 Elliptic Curve Decryption Algorithm

Step 1: Input Elliptic curve parameter (P_M, n_B, G, K, P_B)
where P_M plain text, n_B is receiver secret key, P_B is public key.
Step 2: Consider two cipher text points each letter
Step 3: Calculate (P_M+K.P_B)–[(n_B.K.G)]=(P_M+K.n_B.G)–(n_B. k.G)
The result gives plaintext.

4.3.1 *Example for Elliptic Curve Decryption*

First letter is 'F' for decryption, by receiving the ciphertext pair of points is PC [(260,67), (68,178)],
private key n_B=85
Decryption process
(P_M+K.P_B)–[(n_B.K.G)]=(P_M+K.n_B.G)–(n_B.k.G)
(P_M+K.P_B)–[(n_B.K.G)]=[(68,178)-[85. (260,67)]
(P_M+K.P_B)–[(n_B.K.G)]=[(68, 178)+(253, -130)]
Because -P= (X_P, -Y_P)
(P_M+K.P_B)–[(n_B.K.G)]=[(68, 178)+(253, 147)
Because -130=147 mod 277
(P_M+K.P_B)–[(n_B.K.G)]=(18,158)
And then maps the plaintext point P_M= (18,158) back into the original plaintext message "F" point

Second letter is 'E' for decryption, by receiving the ciphertext pair of points is PC [(260,67), (241,251)],
private key n_B=85
Decryption process
(P_M+K.P_B)–[(n_B.K.G)]=(P_M+K.n_B.G)–(n_B.k.G)
(P_M+K.P_B)–[(n_B.K.G)]=[(241, 251)-[85. (260,67)]

$(P_M+K.P_B)–[(n_B.K.G)]=[(241, 251)+(253, -130)]$
Because $-P= (X_P, -Y_P)$
$(P_M+K.P_B)–[(n_B.K.G)]=[(241, 251)+(253, 147)]$
Because -130=147 mod 277
$(P_M+K.P_B)–[(n_B.K.G)]=(12, 165)$
And then maps the plaintext point P_M= (12, 165) back into the original plaintext message "E" point

Last letter is 'X' for decryption, by receiving the ciphertext pair of points is PC [(260,67), (210, 230)],
private key n_B=85
Decryption process
$(P_M+K.P_B)–[(n_B.K.G)]=(P_M+K.n_B.G)–(n_B.k.G)$
$(P_M+K.P_B)–[(n_B.K.G)]=[(210, 230)-[85. (260,67)]$
$(P_M+K.P_B)–[(n_B.K.G)]=[(210, 230)+(253, -130)]$
Because $-P= (X_P, -Y_P)$
$(P_M+K.P_B)–[(n_B.K.G)]=[(210, 230)+(253, 147)]$
Because -130=147 mod 277
$(P_M+K.P_B)–[(n_B.K.G)]=(36, 59)$
And then maps the plaintext point P_M= (36, 59) back into the original plaintext message "X" point
Finally full plain text is =" FEX"

4.4 Caesar Cipher Decryption Algorithm

Step 1: The specified shift must be an integer range between 0-25.
Step 2: Based on key the given text inversely traverse character at a time.
Step 3: To apply decryption, plain text =(Cipher text + Key) mod 25.

4.4.1 Example of Caesar Cipher Decryption

Decryption formula is PT=(CT-K) mod 25
where PT is plaintext and CT is cipher text integer number,
K is key value.
Decrypt Text "FEX"
F decryption process
PT(F)=(CT(F)-Key) mod 25
PT(F) = (5 - 4) mod 25
PT(F) =1 mod 25
PT(F) = 1
PT(F) = 'B'
E decryption process
PT(E)=(CT(E)-Key) mod 25
PT(E) = (4 - 4) mod 25
PT(E) =0 mod 25
PT(E) = 0
PT(E) = 'A'
X decryption process
PT(X)=(CT(X)-Key) mod 25
PT(\X) = (23 - 4) mod 25
PT(X) =19 mod 25
PT(X) = 19
PT(X) = 'T'
Caesar cipher Decryption is "BAT".
Hence original text retrieved by Caesar cipher after description.

5. RESULTS

The following Table 1 and Table 2 illustrate the results of encryption process and decryption process.

5.1 Encryption Results

Table 1. Encryption Results

Plain Text	**Caesar Cipher Encryption**	**Elliptic curve Ended points $Y^2=(X^3+X+1)$ mod 277**	**Elliptic curve Encryption Points P_C= [K. G], $(P_M+K.P_B)$**
BAT	BAT⇒FEX	F ⇒ (18,158) E ⇒ (12,165) X ⇒ (36,59)	F⇒ [(260,67), (68,178)] E⇒ [(260,67),(241,251)] X⇒ [(260,67),(210,230)]

5.2 Decryption Results

Table 2. Decryption Results

Elliptic curve Decryption $(P_M+K.P_B)$- $[(n_B$-K. G)]	Elliptic curve decryption Text	Caesar cipher decryption	Final result
(260,67), (68,178)⇒(18,158) (260,67), (241,251)⇒(12,165) (260,67), (210,230)⇒(36,59)	FEX	FEX=BAT	BAT

6. CONCLUSION

In this work Elliptic Curve Cryptography is used along with Caesar Cipher for developing Text Based Cryptosystem. This product cipher helps in enhancing the crypto algorithms and their operations with the goal of increasing the protection for data and speed of execution thereby diminishing the essential memory. When Caesar cipher is used in text-based cryptography, each character is transformed into unreadable form and then the message is represented by its ASCII value. All these character values are changed into an affine point on the Elliptic curve, by using an initial point. Then main objective attained by this work is renovation of the plain text by using an affine point. This anticipated work is analyzed on source of affine point creation. As a part of future enhancement the work can be carried out for high level security with small key size.

7. REFERENCES

[1] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE* Trans. Inf. Thy., 22:644-654, 1976.

[2] Martin E Hellman and Justin M. Reyneri. Fast computation of discrete logarithms in GF(q). In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of Crypto '82*, New York, 1983. Plenum Press.

[3] Daniel V. Bailey, Christof Paar. "Arithmetic in finite field extensions with application in elliptic curve cryptography", Journal of Cryptology. 2001: Vol.(14).

[4] Koblitz: "Elliptic curve cryptosystems. Mathematics of Computation. 1987". Vol. (48):2003-2009.

[5] V. Miller. "Use of elliptic curves in cryptography.

Advances in Cryptology-CRYPTO ’85. 1986”, LNCS 218(483), pp. 417-426.

[6] ANSI X9.62. “Public Key Cryptography for the financial services Industry: The Elliptic curve Digital Signature Algorithm (ECDSA)”, 1999.

[7] ANSI X9.62, “Public Key Cryptography for the financial Services Industry: The Elliptic curve Key Agreement & Key Transport Protocols. 2000”.

[8] “IEEE 1363-2000, “Standard Specifications for Public Key Cryptography”.

[9] ISO/IEC 14888, “Information Technology Security Techniques- Digital Signatures., with Appendix- part 3: Certificate based mechanism.

[10] ISO/IEC 15946, “Information Security Technology – Cryptographic Techniques based on Ellitic curve.” 1999.

[11] Ronald Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. Assoc. Comput. Mach*, 21:120-126, 1978.

[12] NIST, “Digital Signature standard. FIPS publication”, 2000. 186-2.

[13] NIST, “Advanced Encryption Standard”.

[14] N. Koblitz, “A family of Jacobian suitable for Discrete Log Cryptosystem”, Advances in Cryptology – Crypto’88, LNCS, Springer – Verlag. Berlin, Vol. 403. 94-99, 1988.