Time Stamp Based ECC Encryption and Decryption

Addepalli Vnkrishna
Department of Computer Science and Engineering, Pujya Shri Madhavanji College of Engineering and Technology, India

Abstract: *Elliptic Curve Cryptography (ECC) provides a secure means of exchanging keys among communicating hosts using the diffie hellman key exchange algorithm. Encryption and decryption of texts and messages have also been attempted. In the paper on Knapsack over ECC algorithm, the authors presented the implementation of ECC by first transforming the message into an affine point on the EC, and then applying the knapsack algorithm on ECC encrypted message over the finite field GF(p). The knap sack problem is not secure in the present standards and more over in the work the authors in their decryption process used elliptic curve discrete logarithm to get back the plain text. This may form a computationally infeasible problem if the values are large enough in generating the plain text. In the present work a new mathematical model is used, which considers the output of ECC algorithm, a variable nonce value and a dynamic time stamp to generate the cipher text. Thus, by having key lengths of even less than 160 bits, the present algorithm provides sufficient strength against crypto analysis and whose performance can be compared with standard algorithms like RSA.*

Keywords: *ECC, time stamp, nonce value, mathematical model.*

Received August 23, 2011; accepted May 22, 2012; published online April 4, 2013

1. Introduction

Historically, encryption schemes were the first central area of interest in cryptography [14]. They deal with providing means to enable private communication over an insecure channel. A sender wishes to transmit information to a receiver over an insecure channel that is a channel which may be tapped by an adversary. Thus, the information to be communicated, which we call the plaintext, must be transformed (encrypted) to a cipher text, a form not legible by anybody other than the intended receiver. The latter must be given some way to decrypt the cipher text, i. e., retrieve the original message, while this must not be possible for an adversary. This is where keys come into play; the receiver is considered to have a key at his disposal, enabling him to recover the actual message, a fact that distinguishes him from any adversary. An encryption scheme consists of three algorithms: The encryption algorithm transforms plaintexts into cipher texts while the decryption algorithm converts cipher texts back into plaintexts. A third algorithm, called the key generator, creates pairs of keys: An encryption key, input to the encryption algorithm, and a related decryption key needed to decrypt. This work mainly deals with the algorithm which generates sub keys which provides sufficient strength to the encryption mechanism. Partial differential equations to model multi scale phenomena are ubiquitous in industrial applications and their numerical solution is an outstanding challenge within the field of scientific computing [7-9]. The approach is to process the mathematical model at the level of the equations, before discretization, either removing non-essential small scales when possible, or exploiting special features of the small scales such as self-similarity or scale separation to formulate more tractable computational problems.

Any symmetric encryption scheme uses a private key for secure data transfer. In their work on "A new mathematical model on encryption scheme for secure data transfer" [8], the authors considered not only key but also time stamp and nonce values to increase the strength of sub key generated. In addition the nonce value can also be used for acknowledgement support between participating parties. The model can be further improved by considering a non linear model where the key values vary with the data generated [7].

Most of the products and standards that use public-key cryptography for encryption and digital signatures use RSA today. Recently, Elliptic Curve Cryptography (ECC) has begun to challenge RSA. The principal attraction of ECC, compared to RSA, is that it appears to offer better security for a smaller key size, thereby reducing processing overhead. ECC [1, 3, 4, 6, 11, 16, 17] makes use of elliptic curves in which the variables and coefficients are all restricted to elements of a finite field. In ECC we normally start with an affine point called Pm (x, y). These points maybe the base point (G) itself or some other point closer to the base point. Base point implies it has the smallest x, y co-ordinates, which satisfy the EC. A character in a message is first

transformed into an affine point of the elliptic curve by using it as a multiplier of Pm. That is, if the ASCII value of a character is A, then we determine P0 m = A (Pm). This is one step towards introducing sophistication and complexity in the encryption process. The newly evaluated P0 m is a point on the EC, determined by applying the addition and doubling strategy of ECC technique. Then as per ECC algorithm, P0 m is added with kPB, where k is randomly chosen secret integer and PB is the public key of user B, to yield (P0 m + kPB). This now constitutes second part of the encrypted version of the message. The other part, namely, kG, which is the product of the secret integer and the Base point, constitutes the first part. Thus the encrypted message is now made up of two sets of coordinates, namely, (kG, P0 m + kPB). In this paper we have assigned kG = (x1, y1) and (Pm + kPB) = (x2, y2). Not satisfied with the complexity involved in determining the encryption, we wish to introduce further complexity by applying time stamp, a variable nonce value concept to the encrypted version. The whole idea behind these rigorous exercises is to make decryption totally impossible, even if the Base Point G, secret integer k, the affine Point Pm are known to the crypt analyst. Now to recover the information from the encrypted version, first the new model with time stamp has to be reversed. Then we apply the decryption process of ECC, by applying the private key of recipient (nB) on the first element (kG). This is subtracted from the second element to recover P0 m. This promises to afford maximum security from intruders and hackers.

Another public key algorithm, namely RSA, is used to encrypt/decrypt the same message. Unlike the ECC procedure, this yields only one integer for each character of the message. The time and space implications for both the schemes are discussed and analyzed. The paper justifies that despite the harsher requirements of time and space for the ECC methods, it is far superior due to the resistance it offers to any brute force attack.

Some recent works on application of ECC are cited here. Aydos *et al.* [2] discusses the results of implementation of ECC over the field GF (p) on an 80 MHz, 32 bit RAM microprocessor. The works in [2, 10, 18] provides an overview of ECC for wireless security. It focuses on the performance advantages in the wireless environment by using ECC instead of the traditional RSA cryptosystem. Adnan [1] explains the design of coprocessor, which automatically produces a customized ECC hardware that meets user-defined requirements. Shi *et al.* [16] explains the engineering of ECC as a complex interdisciplinary research field encompassing such fields as mathematics, computer science and electrical engineering. Chen *et al.* [3] presents a high performance ECC process for general curves over GF (p). The standard specifications for public key cryptography are defined in [5, 15].

2. The Research Method

The weiestrass equation [12, 13] defining an elliptic curve over GF (p), for $q>3$, is as follows:

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

Where x, y are elements of GF (p), and a, b are integer modulo p, satisfying:

$$4a^3 + 27b^2 = 0 \ \ mod\ p \quad (2)$$

Here p is known as modular prime integer. An elliptic curve E over GF (p) consist of the solutions (x, y) defined by equations 1 and 2, along with an additional element called O, which is the point of EC at infinity. The set of points (x, y) are said to be affine coordinate point representation. The basic Elliptic curve operations are point addition and point doubling. Elliptic curve cryptographic primitives [13] require scalar point multiplication. Say, given a point P (x, y) on an EC, one needs to compute kP, where k is a positive integer. This is achieved by a series of doubling and addition of P. Say, given $k = 386$, entails the following sequence of operations P, 2P, 3P, 6P, 12P, 24P, 48P, 96P, 192P, 193P, 386P. Let us start with P *(xP, yP)*. To determine *2P, P* is doubled. This should be an affine point on EC. Use the following equation, which is a tangent to the curve at point *P:*

$$S = [(3x^2P + a) / 2yP\ ]\ mod\ p \quad (3)$$

Then 2P has affine coordinates xR, yR given by:

$$xR = (S2 - 2xP\ )\ mod\ p \quad (4)$$

$$yR = [S(xP - xR) - yP\ ]\ mod\ p \quad (5)$$

Now to determine *3P*, we use addition of points P and *2P*, treating $2P = Q$. Here P has coordinates (*xP, yP*) and $Q = 2P$ has coordinates (*xQ, yQ*). Then:

$$xR = (S2 - xP - xQ)\ mod\ p \quad (6)$$

$$yR = (S(xP - xR) - yP\ ]\ mod\ p \quad (7)$$

Therefore we apply doubling and addition depending on a sequence of operations determined for k. Every point xR, yR evaluated by doubling or addition is an affine point (points on the elliptic curve).

- *Solution of Linear Algebraic Equations*

The solution of the discretization equations for the one dimensional situation can be obtained by the standard Gaussian elimination method. Because of the particularly simple form of equations, the elimination process leads to a delightfully convenient algorithm. For convenience in presenting the algorithm, it is necessary to use somewhat different nomenclature. Suppose the grid points are numbered 1, 2, 3, …, *ni* where *1* and *ni* denoting boundary points. The discretization equation can be written as:

$$AiTi + BiTi + 1 + CiT\text{-}1 = Di \quad (8)$$

For $I = 1, 2, 3, ..., ni$. Thus the data value T is related to neighbouring data values $Ti + 1$ and $Ti¡1$. For the given problem $C1 = 0$ and $Bn = 0$.

These conditions imply that $T1$ is known in terms of $T2$. The equation for $I = 2$, is a relation between $T1$, $T2$ and $T3$. But since $T1$ can be expressed in terms of $T2$, this relation reduces to a relation between $T2$ and $T3$. This process of substitution can be continued until $Tn-1$ can be formally expressed as Tn. But since Tn is known we can obtain $Tn-1$.This enables us to begin back substitution process in which Tn-2, Tn-3, ..., $T3$, $T2$ can be obtained. For this tri-diagonal system, it is easy to modify the Gaussian elimination procedures to take advantage of zeros in the matrix of coefficients. Referring to the tri-diagonal matrix of coefficientsabove, the system is put into a upper triangular form by computing new Ai:

$$Ai = Ai - (C_i - 1 - Ai) * Bi \text{ where } i = 2, 3, ..., ni \quad (9)$$

$$Di = Di - (C_i\text{-}1\text{-}Ai) * Di \quad (10)$$

Then computing the unknowns from back substitution:

$$Tn = Dn / An \quad (11)$$

Then:

$$Tn = Dk - Ak * Tk+1/ Ak; \; k = ni\text{-}1, ni\text{-}2, ..., 2, 1 \quad (12)$$

3. Implementation Details of the Proposed Algorithm

Once the defining EC is know, we can select a base point called G. G has $[x, y]$ coordinates which satisfy the equation $y^2 = x3 + ax + b$. The Base point has the smallest x, y values which satisfy the EC. The ECC method requires that we select a random integer k $(k<p)$, which needs to be kept secret. Then kG is evaluated, by a series of additions and doublings, as discussed above. For purpose of this discussion we shall call the source as host A, and the destination as host B. We select the private key of the host B, called nB. k and nB can be generated by random number generators to give credibility. That would be digressing away from the main discussion. Hence we make suitable assumptions for these two parameters. The public key of B is evaluated by $PB = nBG$. (3) Suppose A wants to encrypt and transmit a character to B, he does the following. Assume that host A wants to transmit the character 'S'. Then the ASCII value of the character 'S' is used to modify Pm as follows: $Pm=SPm$. Pm we said is an affine point. This is selected different from the Base point G, so as to preserve their individual identities. P0m is a point on the EC. The coordinates of the P0m should fit into the EC. This transformation is done for two purposes. First the single valued ASCII is transformed into a x, y co-ordinate of the EC. Second it is completely camouflaged from the would-be hacker. This is actually intended to introduce some level of complexity even before the message is encrypted according to ECC. As the next step of ECC, we need to evaluate kPB, here PB is a public key of user B. Determining this product involves a series of doubling and additions, depending on the value of k. For a quick convergence of the result, we should plan for optimal number of doubles and additions. The encrypted message is derived by adding P0m with kPB, that is, $P0m + kPB$. This yields a set of $x2, y2$ coordinates. Then kG is included as the first element of the encrypted version. kG is another set of $x1$, $y1$ coordinates. Hence the entire encrypted version for purposes of storing or transmission consists of two sets of coordinates as follows:

$$Cm = (kG, P0m + kPB) \quad (13)$$

Where $kG = (x1, y1)$, $(P0m + kPB) = (x2, y2)$.

- *Mathematical Modelling of the Problem*

The approach to time series analysis was the establishment of a mathematical model describing the observed system. Depending on the appropriation of the problem a linear or nonlinear model will be developed. This model can be useful to generate data at different times to map it with plain text to generate cipher text. The linear data flow problem is presented below.

The Initialization Vector (IV) considered in the problem is when $t = 0$, $T(I) = Y(I) = 300$. Where $I = 1, 2, ..., M$.

Dividing the problem area into M number of points, and for simplicity by assuming data of the ¯rst and Mth grid points are considered to be known and constant. For the grid points 2, $M¡1$, the coefficients can be represented by considering the conservation equation:

$$\alpha/\partial x(T_{I+1}{}^{n+1}\text{-}T_I{}^{n+1})+\alpha/\partial x(T_I{}^{n+1}\text{-}T_{I\text{-}1}{}^{n+1})=(\partial x)/\partial t(T_I{}^{n+1}\text{–}_I{}^{n}) \quad (14)$$

Where T_I represents data value for the considered grid point for the preceding delt, $T_{I+1}{}^{n+1}$ and $T_I{}^{n+1}$ represents data values for the preceding and succeeding grid points for the current delt.

Considering α which is a key for the given model, the coefficients are obtained for each state (grid point) in terms of A (I) refers to data value of the corresponding grid point, C (I) and B (I) refers to data values of preceding and succeeding grid points for the current delt, D (I) refers to data value of the considered grid point in the preceding delt:

$$A(I) = 1 + 2\,\alpha\, delt/(delx)**2 \quad (15)$$

$$B(I) = -\alpha\, delt/(delx)**2 \quad (16)$$

$$C(I) = -\alpha\, delt/(delx)**2 \quad (17)$$

$$D(I) = T_I{}^{n} \quad (18)$$

Where α is the key considered which is a constant value. The model generated is a linear model.

4. Implementation of the Proposed Algorithm

The Elliptic Curve is $y^2 \bmod 487 = (x^3 - 5x + 25) \bmod 487$ [13]. The base point G is selected as (0, 5). Base point implies that it has the smallest x, y co-ordinates which satisfy the *EC*. Pm is another affine point, which is picked out of a series of affine points evaluated for the given *EC*. We could have retained G itself for Pm. However for the purpose of individual identity, we choose Pm to be different from G. Let $Pm = (1,316)$. The choice of Pm is itself an exercise involving meticulous application of the ECC process on the given *EC*, the secret integer k, and the private key nB of the recipient B. We have at our disposal a series of random number generators. But that would be digressing from the main path of thought. Hence we shall assume that $k = 25$, and $nB = 277$. Plaintext is "S", whose ASCII value is 83. Therefore,

$PB = nBG = 277\ (0, 5) = (260, 48)$
$P0m = 83\ (1, 316) = (475, 199)$
$kPB = 225\ (260, 48) = (212, 151)$
$P0m + kPB = (475, 199) + (212, 151) = (51, 58)$
$kG = 225\ (0, 5) = (99, 253).$

Encrypted version of the message is: ((99, 253), (51, 58)), where $x1 = 99$, $y1 = 253$, $x2 = 51$, and $y2 = 58$. By considering an input of 99, 253 to the mathematical model [7], as data input, the secret integer k as key, a time stamp of 6 and a nonce of 25, a set of 25 values are calculated: 27 33 34 17 30 16 26 8 4 20 22 12 24 12 6 21 18 10 23 22 30 15 1 18 0, the middle of 25 values ie value will replace 99, a quarter value data series replace 253. The output for (99, 253) = (24, 26).

A similar procedure will be repeated for (51, 58). The sequence generated is: 15 8 4 20 17 26 13 2 20 28 34 0 18 28 14 25 10 6 21 20 10 23 23 29 0, the output for (51, 58) = (18, 13), the cipher text that is to be transmitted for S is (24, 26), (18, 13).

S equivalent ASCII is 83, Cipher text through ECC is (99 253), (51 58). Final cipher text through the discussed model (24 26), (18 13), a equivalent ASCII is 65, Cipher text through ECC is (99 253), (116 280). Final cipher text through the discussed model (24 26), (15 07), V equivalent ASCII is 86, Cipher text through ECC is (99 253), (427 287). Final cipher text through the discussed model (24 26), (1 1), E equivalent ASCII is 69, Cipher text through ECC is (99 253), (135 341). Final cipher text through the discussed model (24 26), (16 20).

- *Decryption Process:* The given cipher text is considered. Known the private key, time stamp, delx and delt values, the inverse process is used which maps the generated sequence with the cipher text values. The procedure is repeated till the difference is of the order of 10^{-4}. Once the output values are generated, they will be considered as input to ECC algorithm to generate Plain text.

5. Conclusions

ECC itself is a very secure algorithm for encryption. However, not satisfied with it the algorithm is appended with time stamped mathematical model which contains not only private key of ECC algorithm but also a variable nonce value and a dynamic time stamp which makes entire encrypted version turns into an ensemble of confusing integers, thereby discouraging a potential cryptanalyst from attempting a brute force attack.

The advantage with this model is it is free from linear and differential cryptanalysis. Also it is supported with variable nonce value, which acts as acknowledgement between participating parties. It is also supported with dynamic time stamp, which increases the strength of the algorithm. Thus the given model supports the important properties like authentication, security and confidentiality at less computing resources when compared with algorithm like RSA. A comparative study of the proposed model with standard algorithms like ECC and RSA in terms of computational overhead, security strength, data overhead, complexity and security analysis is discussed in Table 1.

Table 1. A comparative study.

Algorithm	Computational Overhead per Block of Data	Security Strength	Data Overhead per Block of Data	Complexity by Its Strength	Security Analysis
RSA	More as the specified key length is 1024 bits	Equal	Equal	Exponential	Relatively free from known & chosen attacks
ECC	Relatively very less as specified key length is 160 bits.	Equal	Equal	Exponential	Relatively free from known & sen attacks
Proposed Model	More 500 instructions for a 8 bit sub key used in addition to ECC.	More	Equal	Exponential	Relatively free from known & chosen attacks

6. Future Work

By using a non linear private key, the strength of the proposed model can be increased still further. The model can also be studied for increased performance against unavoidable factors like noise.

References

[1] Adnan G., "Area Flexible GF (2k) ECC Coprocessor," *the International Arab Journal of Information Technology*, vol. 4, no. 1, pp. 1-10, 2007.

[2] Aydos M., Yanik T., and Kog C., "High-Speed Implementation of an ECC-Based Wireless Authentication Protocol on an ARM Microprocessor," *IEEE Proceedings of Communication*, vol. 148, no. 5, pp. 273-279, 2001.

[3] Chen G., Bai G., and Chen H., "A High-Performance Elliptic Curve Cryptographic Processor for General Curves Over GF(p) Based on a Systolic Arithmetic Unit," *IEEE Transactions on Circuits System-II: Express Briefs*, vol. 54, no. 5, pp. 412-416, 2007.

[4] Cheng R., Baptiste N., Luk W., and Cheung P., "Customizable Elliptic Curve Cryptosystems," *IEEE Transactions on VLSI Systems*, vol. 13, no. 9, pp. 1048-1059, 2005.

[5] Diffie W., "The First Ten Years of Public Key Cryp-Tography," *Proceedings of IEEE*, vol. 76, no. 5, pp. 560-577, 1988.

[6] Finnigin K., Mullins B., Raines R., and Potoczny H., "Cryptanalysis of an Elliptic Curve Cryptosystem for Wireless Sensor Networks," *the International Journal of Security and Networks*, vol. 2, no. 3/4, pp. 260-271, 2006.

[7] Krishna A., "A New Non Linear Model Based Encryption Scheme with Time Stamp & Acknowledgement Support," *the International Journal of Network Security*, vol. 14, no. 1, pp. 27-32, 2012.

[8] Krishna A. and Babu A., "A New Model Based Encryption Scheme with Time Stamp & Acknowledgement Support," *the International Journal of Network Security*, vol. 11, no. 3, pp. 172-176, 2010.

[9] Krishna A. and Babu A., "A New Non Linear, Time Stamped & Feed Back Model Based Encryption Mechanism with Acknowledgement Support" *the International Journal of Advancements in Technology*, vol. 1, no. 2, pp. 197-202, 2010.

[10] Lauter K., "The Advantages of Elliptic Cryptography for Wireless Security," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 62-67, 2006.

[11] Lee J., Kim H., Lee Y., Hong S., and Yoon H., "Parallelized Scalar Multiplication on Elliptic Curves Defined Over Optimal Extension Field," *the International Journal of Network Security*, vol. 4, no. 1, pp. 99-106, 2007.

[12] Moon S., "A Binary Redundant Scalar Point Multipli-Cation in Secure Elliptic Curve Cryptosystems," *the International Journal of Network Security*, vol. 3, no. 2, pp. 132-137, 2006.

[13] Ramasamy R., Prabakar M., Indra Devi M., and Suguna M., "Knapsack Based ECC Encryption and Decryption," *the International Journal of Network Security*, vol. 9, no. 3, pp. 218-226, 2009.

[14] Stallings W., *Cryptography and Network Security*, Prentice Hall, USA, 4th Edition, 2006.

[15] Standard Specifications for Public Key Cryptography, IEEE Standard, pp. 1363-2000, available at: http://grouper.ieee.org/groups/1363, last visited 2008.

[16] Shi Z. and Yan H., "Software Implementation of Elliptic Curve Cryptography," *the International Journal of Network Security*, vol. 7, no. 2, pp. 157-166, 2008.

[17] Wang H., Sheng B., and li Q., "Elliptic Curve Cryptography-Based Access Control in Sensor Net-Works," *the International Journal of Security and Networks*, vol. 1, no. 3/4, pp. 127-137, 2006.

[18] Yongliang L., Gao W., Yao H., and Yu X., "Ellip-Tic Curve Cryptography Based Wireless Authentication Protocol," *the International Journal of Network Security*, vol. 4, no. 1, pp. 99-106, 2007.

Addepalli Vnkrishna received his BE (Mechanical) from Osmania University, M. E (Mechanical) from Sivaji University, M. Tech (Computer Science) from B. I. T. Ranchi and PhD degree from the Department of Computer Science & Engineering, Acharya Nagarjuna University, Andhra Pradesh, India. After his Doctoral Work, he is working as Principal & Professor of Computer Science & Engineering in Pujya Shri Madhavanji College of Engineering & Technology, JNTUH, Hyderabad. His main research interests include computer networks, network security, simulation & modelling etc. He is presently guiding 5 PhD students.