Elliptic Enigma

Larry Smith • Sara Beshara • Paula Ong

Abstract

Elliptic Curve Cryptography (ECC) is a public key system that has been gaining momentum as a replacement for RSA public key cryptography largely based on its efficiency to create and strength to stay concealed. Also, the US National Security Agency (NSA) has included it in its Suite B recommendations, while excluding the RSA method. Suite B is a set of algorithms that the NSA recommends for use in protecting both classified and unclassified US government information and systems. ECC is now being used nationwide.

Background

Elliptical curve cryptography (ECC) is a type of public key encryption technique. The idea was first proposed in 1985 by Victor S. Miller and Neal Koblitz. The ECC method generates keys using its unique elliptic curve equation instead of the product of very large prime numbers. An elliptic curve is not an ellipse, but rather a looping line intersecting two axes. There are many benefits in using ECC method: For one thing, ECC uses faster, smaller, and more efficient cryptographic keys. For instance, one ECC key can yield a level of security with a 256-bit that other systems require a 3,072-bit key to achieve. Due to this fact, many manufacturers, including 3COM, Cylink, Motorola, Pitney Bowes, Siemens, TRW, and VeriFone use ECC in their products. Also (another vital fact) is that ECC equations have a characteristic that is very valuable for cryptographic purposes: they are relatively easy to perform, but extremely difficult to reverse.

Problem

The elliptic curve (ECC) system is very effective when considering functionality, security, and performance. When considering the security aspect of this system, the fundamental issue is how difficult is the underlying mathematical problem that is necessary for all system protocols for the public key (Certicom, 15). For the elliptic curve system that would be the elliptic curve discrete logarithm problem. Let E be an elliptic curve defined over the finite field $\boldsymbol{F_p}$ and $P \in \boldsymbol{F_p}$. Given Q a multiple of P the elliptic curve discrete logarithm problem is to find $n \in \mathbb{Z}$ such that $\boldsymbol{nP = Q}$. For example, let E be the elliptic curve $y^2 = x^3 + x + 1$ defined over $\mathbf{F_7} = \{\infty, (0, 1), (0, 6), (2, 2), 2, 5)\}$. If $P = (2, 2)$ and $Q = (0, 6)$, then $3P = Q$, so $n = 3$ is the solution to the discrete logarithm problem.

There are well known methods for point addition on elliptic curves defined over $\boldsymbol{F_p}$, where p is a prime number, and satisfy group law, they are the following;

Point Addition: Let $P = (x_1, y_1) \in \boldsymbol{F_p}$ and $Q = (x_2, y_2) \in \boldsymbol{F_p}$ where $P \neq \pm Q$, then $P + Q = (x_3, y_3)$

$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$ and $y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$

Point Doubling: Let $P = (x_1, y_1) \in \boldsymbol{F_p}$ where $P \neq -P$. Then $2P = (x_3, y_3)$

$x_3 = \left(\frac{3x_1 + a}{2y_1}\right)^2 - 2x_1$ and $y_3 = \left(\frac{3x_1 + a}{2y_1}\right)(x_1 - x_3) - y_1$

Figure 1 is the geometric representation of the point addition rule which defines $E(\mathbf{F}_p)$ as an abelian group. With these parameters and the following algorithms the elliptic curve system can be implemented with confidence that communication between two parties will be secure from any third party. The following is an example of how to implement the elliptic curve system algorithms.

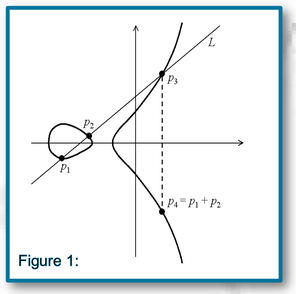


Figure 1:

Elliptical Curve Key Generation:

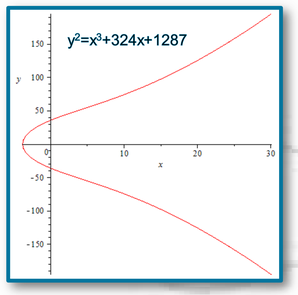
INPUT: Elliptic curve domain parameters (*p, E, P, n).*
OUTPUT: Public key *Q* and private key *d.*
1. Select d ϵR [1, *n*-1].
2. Compute *Q=dP.*
3. Return (*Q,d).*

Basic Elliptic Curve Encryption:

INPUT: Elliptic curve domain parameters (*p, E,P,n*), public key *Q*, plaintext *m.*
OUTPUT: Cipher text (C_1, C_2).
1.Represent the message *m* as a point *M* in *E* (F_p).
2. Select *k* ϵR [1, *n*-1].
3. Compute C_1=*k*P.
4. Compute C_2= *M*+kQ.
5. Return (C_1,C_2).

Basic Elliptic Curve Decryption:

INPUT: Domain parameters (*p, E, p, n),* private key *d*, cipher text (C_1, C_2).
OUTPUT: Plaintext *m.*
1. Compute *M= C*$_2$*-dC*$_1$, and extract *m* from *M.*
2. Return (*m).*



Elliptic Curve Key Generation:

Input: Elliptic Curve domain parameters (p, E, P, n)
Output: Public key Q and private key d.
p = 3851 (prime number)
E:$y^2=x^3+324x+1287$
P(920, 303)
n = 3928 Order of finite field F_{3851}
1. Select d = 21
2. Q=21P=(111,2003) Maple program mult(324, 3851, 21, 920, 303)
3. Return d = 21 and Q=(111,2003)

Basic Elliptic Curve Encryption:

Input: Elliptic Curve domain parameters (p, E, P, n), public key Q, plaintext m, and $R \epsilon E(F_{3851})$
Output: Cipher text (C_1, C_2)

1.Let $m = 89$ represent the word "Hi" and R(9, 358)
$M = mR$
$M = 89R = (621, 2764)$ (mod 3851)

2. Select $k = 10$

3. Compute $C_1 = kP$
$C_1 = 10P = (513, 1372)$ (mod 3851)

4. Compute $C_2 = M + kQ$
$C_2 = (621, 2764) + 10(111, 2003) =$
$(2923, 1500)$ (mod 3851)

5. Return (513, 1372) and (2923, 1500)

Basic Elliptic Curve Decryption:

Input: Domain parameters (p, E, P, n), private key d, cipher text (C_1, C_2)
Output: Plaintext m

1. Compute $M = C_2 - dC_1$
$M = (2923, 1500) - (1072, 3645) = (621, 2764)$
$M = (2923, 1500) + (1072, -3645) = (621, 2764)$
(mod 3851)
2. Extract m from M
$(621, 2764) = m(9, 358)$
$m = 89$ "Hi" (Cipher Substitution)

Summary

After twenty years of research, development, and advancement, ECC has now securely positioned itself as the top public-key mechanism in several businesses worldwide: industry, banking, marketing, and government. The principal reasons ECC has gained widespread exposure and acceptance is due to its efficiency, functionality, performance, and most importantly, its security strength. Its efficiency is based on the small number of elementary steps that needs to be executed by the algorithm and exactly how long that algorithm takes, which, for ECC, is normally within polynomial time- basically, the fastest and most efficient length of time one would want to spend encrypting. Its strength and security also belittles its competitors, RSA and DL. In cryptography there are five standardized security levels and each algorithm method has its own parameters that it can use at those security levels.

Security Level	80 (SKIPJACK)	112 (Triple-DES)	128 (AES-Small)	192 (AES-Medium)	256 (AES-Large)
DL parameter q EC parameter n	160	224	256	384	512
RSA modulus n DL modulus p	1024	2048	3072	8192	15360

When talking parameters, or key sizes, bigger isn't always better. In fact, in cryptography, smaller parameter sizes are equivalent to higher security levels. This is because smaller parameters have advantages such as speed (faster computations), small keys, and smaller certificates. According to the graph, all these benefits are found in ECC. ECC is even many times more efficient than RSA and DL systems in private-key operations such as signature generation and decryption and signature verification and encryption. ECC is always chosen over RSA and DL especially in environments where processing power, storage, bandwidth, or power consumption is constrained.

Future Work

The complex cryptosystem of elliptical curve is the most secured in public key management. Many wireless devices have become dependent on security features to protect its consumers, and ECC allows a better implementation of these features. The further advancement and development of ECC will provide greater security and a more efficient performance than any other public key algorithms.

Bibliography
Image Background: http://en.wikipedia.org/wiki/File:EllipticCurveCatalog.svg
Image1: http://www.techtroubleshooters.com/images/wireless-networking.jpg
Image2:http://www.businesscreditcardprocessing.com/images/verifone_vx610_left_1.jpg
Image3:http://computerrepairservicespro.com/wp-content/uploads/2013/04/Wireless-Internet-Security.png
Image4:https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcRFDqmpDWePvYjAcfyVFcvQf-HHNXhRrCFYIp9-dIMH-qOfvFKBZ0omLzPu
Image5: http://www.appgenius.it/segnala-una-app/
Figure 1 graph: http://www.rsa.com/rsalabs/faq/images/eca.gif
Hankerson, Darrel R., Scott A. Vanstone, and A. J. Menezes. Guide to Elliptic Curve Cryptography. New York: Springer, 2004. Print.
Image6: http://www.itime.cn/Article/mobile/news/14657.shtm
Image7: http://www.technominds.com/services/security-assessment-and-recommendations
Image8: http://www.glogster.com/mpikb3/digital-world/g-6lv23n2uqso7bhanakb53a0
Image9: http://www.genotipo.com/blog/como-optimizar-un-sitio-web/

This material is based upon work supported by the National Science Foundation under Grant No. 1241636. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Dr. Charles Lam • Frank Madrid