

Use of Elliptic Curve Cryptography for Integrity and Authentication

Swati Sucharita Barik^{#1}, Smruti Sephalika Barik^{*2}

[#]Department Of Computer Science & Engg,
Centurion University Of Technology and Management,
Bhubaneswar.

Abstract— Now a days, Internet becomes the media of communication. Intruders always keep on trying to get the access of the data, which gives rise to the protection of data. The Data needs to be protected from the attackers. To achieve this, a no. of techniques have been followed. One of the Popular algorithm used for it is RSA. The main drawback, RSA is having with it is increasing key length. To overcome it, Elliptic curve cryptography (ECC) comes into existence.

Keywords— Cryptography, ECC, Integrity, Authentication

I. INTRODUCTION

Cryptography deals with the principles of securities like Confidentiality, Data integrity, Authentication. In earlier days cryptography has its applications in defence and military purposes to secure the data. But now a days, it becomes useful in protecting data in online transactions, ATM, credit cards, etc.

Depending on the key, cryptography can be categorized into two parts: Symmetric and Asymmetric. In Symmetric Key Cryptography, same key is used for both encryption as well decryption. i.e. if K is the key and M is the message, then, we have $DK(EK(M)) = M$.

Asymmetric or Public key cryptography use a pair of keys. One is used for encryption and the other key is used for decryption. One of the keys is made public or sharable while the other key is kept a secret. i.e. let k1 and k2 be public and private keys respectively. Let M be the message, then $Dk2(Ek1(M)) = Dk1(Ek2(M)) = M$.

II. INTRODUCTION TO ECC

ECC (Elliptic Curve Cryptography) is a kind of public key cryptography. The drawback of Using RSA is the use of increasing key length, which is overcome by ECC.

The main difference between RSA and ECC is that ECC provides the level of security as that of RSA using smaller key length.

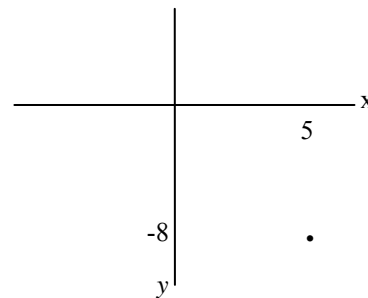
A. Elliptic Curves

ECC is based on elliptic curves.

Elliptic curve is similar to the normal curve drawn as a graph. Graph has x and y axes having points.

Each point can be designated by (x,y) coordinates.

For example, a point can be designated as (5,-8), that means it is 5 units on right hand side of x-axis from center and 8 below y-axis from center.



(fig1. point in reference with x-axis and y-axis)

Assume an elliptic curve, E with a point P. A random number is generated i.e. d.

$$Q = d \times P$$

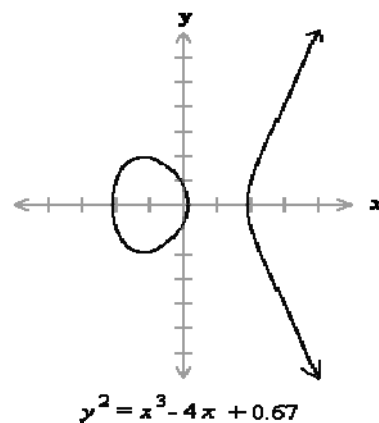
E, P and Q are public values.

The objective is to find d.

This problem is known as elliptic curve discrete logarithm problem.

B. Example:

An Elliptic Curve is a set of point on a curve given certain real numbers a and b. fig.2 shows the elliptic curve.



(Fig.2)

III. ELLIPTIC CURVE EQUATION

An elliptic curve is a cloud of points, which satisfies curve equation.

The equation is:

$$y^2 = x^3 + ax + b \pmod{p}$$

Here a, b, x, y are within elliptic curve.

The coefficients determine points on curve.

The curve coefficients have to fulfil the condition, i.e.

$$4a^3 + 27b^2 \neq 0$$

A point is represented on the curve done in the affine projection. The Points represented in affine coordinates are vectors with an x and y component, like in euclidian coordinate system. The only difference is that the x and y values are also integers modulo p.

There is one exception: One point at infinity i.e O, exist on any curve.

To denote points, uppercase letters will be used and to denote integers, lowercase letters will be used.

$$A = \begin{pmatrix} a_x \\ a_y \end{pmatrix}$$

IV. ECC OPERATION

A. Curve cryptosystem parameters

1. p: The prime number which defines the field in which the curve operates, F_p . All point operations are taken modulo p.
2. a, b: The two coefficients which define the curve. These are integers.
3. G: The generator or base point. A distinct point of the curve which resembles the "start" of the curve. This is either given in point form G or as two separate integers g_x and g_y .
4. n: The order of the curve generator point G. This is, in layman's terms, the number of different points on the curve which can be gained by multiplying a scalar with G. For most operations this value is not needed, but for digital signing using ECDSA the operations are congruent modulo n, not p.
5. h: The cofactor of the curve. It is the quotient of the number of curve-points, or $E(F_p)$, divided by n.

B. Generating a keypair

To get the private key, a random integer is chosen d_A ,

$$0 < d_A < n$$

To get the public key Q_A is equally trivial, we have to use scalar point multiplication of the private key with the generator point G:

$$Q_A = d_A \cdot G$$

The public and private key are not equally exchangeable. the private key d_A is a integer, but the public key Q_A is a point on the curve.

V. ELLIPTIC CURVE DIGITAL SIGNATURE

The Elliptic Curve Digital Signature Algorithm (ECDSA) was standardized in FIPS 186-4.

The signer generates a key pair (d, Q), which consist of a private key d and a public key $Q = dG$.

To sign a message m, the signer first chooses a random integer k such that $1 \leq k \leq n - 1$, we have to compute the point $(x_1, y_1) = kG$, it transforms x_1 to an integer and computes $r = x_1 \bmod n$.

The message m is hashed to a bit string of length not more than the bit length of n, which is then transformed to an integer e.

The signature of m is the pair (r, s) of integers modulo n, where $s = k^{-1}(e + dr) \bmod n$.

r and s need to be different from 0, and k must not be revealed and must be a per-message secret, which means that it must not be used for more than one message.

The secret signing key d can be computed by $d \equiv r^{-1}(ks - e) \bmod n$ as r and s are given in the signature and e can be computed from the signed message.

if the same value for k is used to sign two different messages m_1 and m_2 using the same signing key d and producing signatures (r, s_1) and (r, s_2),

then k can be easily computed as

$$k \equiv (s_2 - s_1)^{-1}(e_1 - e_2) \pmod{n}$$

VI. APPLICATIONS.

A. BITCOIN:

The cryptocurrency Bitcoin is a distributed peer-to-peer digital currency which allows "online payments to be sent directly from one party to another without going through a financial institution". The public Bitcoin block chain is a journal of all the transactions ever executed. Each block in this journal contains the SHA-256 hash of the previous block, hereby chaining the blocks together starting from the so-called genesis block. In Bitcoin, an Elliptic Curve Digital Signature Algorithm (ECDSA) private key serves as a user's account. Transferring ownership of bitcoins from user A to user B is realized by attaching a digital signature (using user A's private key) of the hash of the previous transaction and information about the public key of user B at the end of a new transaction. The signature can be verified with the help of user A's public key from the previous transaction.

B. SECURE SHELL (SSH):

Elliptic curve cryptography can be used in three positions in the SSH protocol. In SSH-2, session keys are negotiated using a Diffie-Hellman key exchange. Each server has a host key. It allows the server to authenticate itself to the client. The server sends its host key to the client during the key exchange.

The user verifies that the key fingerprint matches their saved value. The server then authenticates itself by signing a transcript of the key exchange. This host key may be an ECDSA public key. the clients can use ECDSA public keys for client authentication.

C. TRANSPORT LAYER SECURITY (TLS):

In TLS, elliptic curves can arise in several locations in the protocol. elliptic curve cipher suites for TLS. All the cipher suites specified in this RFC use the elliptic curve Diffie-Hellman (ECDH) key exchange. The ECDH keys may either be long-term. TLS certificates also contain a public key that the server uses to authenticate itself; with ECDH key exchanges, this public key may be either ECDSA or RSA.

VII. CONCLUSIONS

ECC ensures authentication as well as integrity by generating key ECDSA and ECDH are helpful in key exchange.

REFERENCES

- [1] V. Miller, "Use of elliptic curves in cryptography", *Crypto* 85, 1985.
- [2] Vipul Gupta, Sumit Gupta, Sheueling Chang and Douglas Stebila, "Performance Analysis of Elliptic Curve Cryptography for SSL", *WiSe'02*, September 28, 2009.
- [3] Vipul Gupta, Douglas Stebila, and S.C. Shantz, "Integrating Elliptic Curve Cryptography into the Web's Security Infrastructure" *WWW2004*, May 17–22, 2004 .
- [4] White paper "Elliptic Curve Cryptography: The Next Generation of Internet Security", Industry Announcement Next Generation Internet Security.
- [5] Vivek Kapoor, Vivek Sonny, Abraham and Ramesh Singh "Elliptic Curve Cryptography", *ACM Ubiquity*, Vol. 9, No. 20 May 20–26, 2008.
- [6] Robert Milson, "Introduction to Public Key Cryptography and Modular Arithmetic"
- [7] Aleksandar Jurisic and Alfred J. Menezes, *Elliptic Curves and Cryptography*
- [8] William Stallings, *Cryptography and Network Security-Principles and Practice* second edition, Prentice Hall publications.
- [9] V Gupta, S Gupta, D Stebila, "Performance Analysis of Elliptic Curve Cryptography for SSL", *WiSe'02-ACM Workshop on Wireless Security*, 2002, portal.acm.org
- [10] A Lenstra, E Verheul, "Selecting cryptographic key sizes" "Selecting Cryptographic Key Sizes", *Journal of Cryptology* 14 (2001) 255-293,