Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-5, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Integrity and Authentication using Elliptic curve cryptography

Sunil Kumar Das[1], Gaurav Sharma[2] & Dr. Pramod Kumar Kevat[3]
[1,2]Department of Applied Mathematics, Indian School of Mines, Dhanbad
[3]Assistant professor, Department of Applied Mathematics Indian School of Mines, Dhanbad

***Abstract*: —** *Internet is a media of communication. Access of data is often sought after by intruders. Therefore, protection of data is one of the foremost demands of this decade. Various techniques have been developed to achieve this means, RSA being one of them. Howsoever strong RSA is, its misses out because the key length required for use in RSA is ever increasing. To overcome it, Elliptic curve cryptography is tool which makes the lives of security analysts easier.*

Introduction

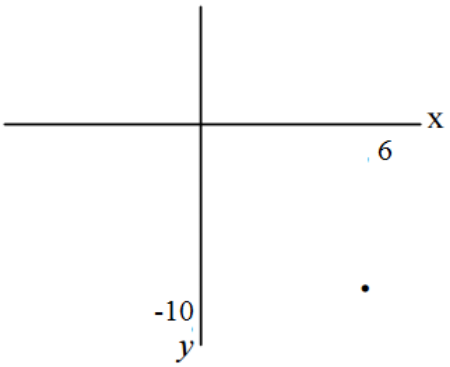
Principles of security, like Confidentiality, Data Integrity and Authentications etc are primarily dealt using Cryptography. In earlier days cryptography has its applications in defense and military purposes to secure the data. Past two decades, however, have seen its rise in protecting data in online transactions, ATM, credit cards, etc. Cryptography can be characterized into two primary parts, depending on the key: Symmetric and Asymmetric. Symmetric Key Cryptography, uses same key 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..

1. Elliptic Curve Cryptography: An introduction

ECC (Elliptic Curve Cryptography) is a kind of public key cryptography. RSA suffers from various drawbacks, primary being the increase in key length. This drawback is overcome by using Elliptic Curve Cryptography (ECC). This also turns out to be the main difference between RSA and ECC. ECC provides the same level of security as that of RSA while using a key of smaller length at the same time.

i. 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 (6, -10).that means it is 6 units on right hand side of x-axis from center and 10 below y-axis from center.



(figure1. 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 r.

$$Q=r \times P \; E,P$$

and Q are public values. The objective is to find r. This problem is known as elliptic curve discrete logarithm problem.

ii. An 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.

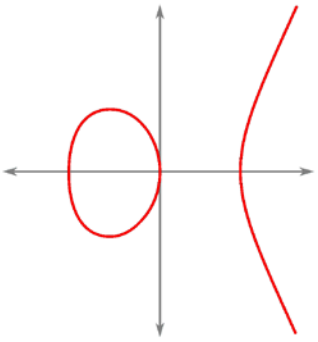


Figure 2. Elliptic curve with eqn. $y^2 = x^3 - x$

2. Equation for an elliptic curve

An elliptic curve is a cloud of points, which satisfies curve equation. The equation is: $y^2=x^3+ax+b$ mod p Here a, b, x, y are within elliptic curve. The coefficients determine points on curve. The curve coefficient shave 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.

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

3. Operations used in Elliptic Curve Cryptography

A. Curve cryptosystem has the following parameters:

1. p: The prime number which defines the field in which the curve operates, Fp. 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 gx and gy

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(Fp), 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 QA 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 dA is a integer, but the public key QA is a point on the curve.

4. Digital signatures of an Elliptic Curve

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 (x1, y1) = kG, it transforms x1 to an integer and computes r = x1 mod 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 m1 and m2 using the same signing key d and producing signatures (r, s1) and (r, s2), then k can be easily computed as $k \equiv (s2{-}s1){-}1(e1{-}e2) \pmod n$

5. Applications of Elliptic Curve Cryptography

A. 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

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. 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 Signal 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.

6. Conclusions

Thereby in this paper we have discussed how ECC ensures authentication as well as integrity by generating key. Also we discussed how ECDSA and ECDH are helpful in key exchange.

7. Acknowledgements

I would like to thank my mentor Dr. Pramod Kumar Kevat for his esteemed guidance.

8. References

[1] D. Hankerson, A. Menezes, and S.A. Vanstone, Guide to Elliptic Curve Cryptography, Springer-Verlag, 2004.

[2] White paper "Elliptic Curve Cryptography: The Next Generation of Internet Security", Industry Announcement Next Generation Internet Security.

[3] I. Blake, G. Seroussi, and N. Smart, editors, Advances in Elliptic Curve Cryptography, London Mathematical Society 317, Cambridge University Press, 2005. [4] V. Miller, "Use of elliptic curves in cryptography", Crypto 85, 1985.

[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] K. Malhotra, S. Gardner, and R. Patz, Implementation of Elliptic-Curve Cryptography on Mobile Healthcare Devices, Networking, Sensing and Control, 2007 IEEE International Conference on, London, 15–17 April 2007 Page(s):239–244

[10] Luca De Feo, David Jao, Jerome Plut, Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies, Springer 2011.