Elliptic Curve Digital Signatures and Their Application in the Bitcoin Crypto-currency Transactions

Benjamin K. Kikwai

16 October 2017

Abstract- . The Elliptic Curve Digital Signature Algorithm (ECDSA), defines a technique for generating and validating digital signatures. We start by reviewing the mathematics behind the *Digital Signature Algorithm* (DSA) and its elliptic curve analogue (ECDSA). Secondly, we describe how the ECDSA is used in *Bitcoin technology.*

Bitcoin is a type of crypto-currency that has been in existence since 2009. It was introduced by Satoshi Nakamoto (possibly a pseudonym) in their much cited paper [8]. Its design and distribution is not controlled by any central organization. Despite this, Bitcoin has defied all odds to become a phenomenal currency widely accepted by thousands of merchants worldwide. At the time of writing this article, 1 unit of Bitcoin (1 BTC) has a value of approximately USD 5,500. The internal workings of Bitcoins is based on elliptic curve digital signatures and is not well understood by many people including a large percentage of Bitcoin users. The the second part of this article we make give an overview to illuminate how the internal nuts and bolts of Bitcoin works.

Keywords: *Elliptic curves, digital signatures, cryptography, blockchains, Bitcoin.*

1. Introduction

This article is a short overview of elliptic curves digital signatures and their application in the blockchain technologies. The theory of elliptic curves and their application to cryptography is a much wider subject. The interested readers are encouraged to consult much deeper expositions contained in [11, 5].

2. Discrete logarithm problem

Let G be a finite cyclic group generated by $a \in G$ and $x \in G$ be an element. This means that $x = a^n$ for some integer n. The integer n is called the *discrete logarithm* of x to the base a. Given the pair a, n the problem of computing a^n in G is generally considered to be an easy problem. On the other hand, given an element $x \in G$ then the problem of obtaining n such that $x = a^n$ is generally a difficult problem. This problem is referred as the *discrete logarithm problem* (DLP).

In 1976, Deffie and Hellman [1] proposed the public key cryptography whose security relied on the difficulty of the discrete logarithm problem. Since then there has been intense research in finding efficient ways to solve the discrete logarithm problem as well as identifying potential finite groups whose corresponding DLP would be much harder to solve.

Elliptic Curve Cryptography (ECC) was introduced independently by Miller [6] and Koblitz [5]. They proposed a cryptographic system based on the group of points of an elliptic curve defined over $\mathbb{F}_q$ - a finite field of order q. One of the reasons for their suggestion was based on the evidence that the elliptic curve discrete logarithm problem was much harder to solve compared to their counterparts over $\mathbb{Z}_p$.

3. Elliptic curves over finite fields

An elliptic curve over $\mathbb{R}$ is a curve in the two dimensional plane whose points satisfy the *Weierstarss equation*

$$y^2 = x^3 + ax + b. \tag{1}$$

Arithmetics over $\mathbb{R}$ often yields irrational numbers which, due to truncation errors, cannot be stored efficiently in computer memory. This is the reason why it is preferable to work with elliptic curves over $\mathbb{F}_q$ - i.e. a finite field of order q where $q = p^k$ for some integers p, k with p a prime. In this case we are interested in integers that satisfy

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

if $p \neq 2$. If $p = 2$, equation (2) above is usually replaced by

$$y^2 + xy = x^3 + ax^2 + b. \tag{3}$$

Definition 1. *An elliptic curve $E(\mathbb{F}_q)$ defined over $\mathbb{F}_q$ consist of the set of points $P_i = (x_i, y_i)$ satisfying equations* (2) *or* (3) *above together with an additional point $\mathcal{O}$ called the point at infinity.*

For practical uses, it is required that the curve be *non-singular*. This amounts to choosing numbers a, b such that $4a + 27b$ is not congruent to 0 modulo p.

Over $\mathbb{R}$, there is a natural geometric construction (cord and tangent process [11, III §2]) that trasnforms

the points of an elliptic curve into an abelian group having $\mathcal{O}$ as the neutral element. The geometric construction relies on a special case of the Bézout Theorem [4, I.7.8] stating that any pair of projective curves of degrees n, m intersects at exactly nm points. For the purposes of this short article, we shall give explicit formulas for addition of points.

The group operation on $E(\mathbb{F}_q)$ is written additively. Given $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ points on $E(\mathbb{F}_q)$, one can obtain a third point $P_3 = (x_3, y_3) = P_1 + P_2$ using the following explicit formulas. Assuming that $P_1 \neq P_2$ and $p \neq 2$ then

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \tag{4}$$

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1. \tag{5}$$

On the other hand, if $P_1 = P_2$ and $p \neq 2$ then we have the *point doubling* formulas as follows

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \tag{6}$$

$$y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1. \tag{7}$$

If $p = 2$ and $P_1 \neq P_2$ then equations (4) and (5) are replaced by

$$x_3 = \left(\frac{y_2 + y_1}{x_2 + x_1}\right)^2 + \left(\frac{y_2 + y_1}{x_2 + x_1}\right) + a + x_1 + x_2 \tag{8}$$

$$y_3 = \left(\frac{y_2 + y_1}{x_2 + x_1}\right)(x_1 + x_3) + y_1 + x_3. \tag{9}$$

Finally if $p = 2$ and $P_1 = P_2$ then the addition formulas are given by

$$x_3 = \left(\frac{x_1^4 + b}{x_1^2}\right) \tag{10}$$

$$y_3 = \left(\frac{x_1^2 + y_1}{x_1}\right)(x_3) + x_1^2 + x_3. \tag{11}$$

The operations in equations (4) - (11) are performed modulo p and thus divisions should be interpreted as multiplications by multiplicative inverses. Though the equations above are obtained from a geometrical construction, these equations are evaluated using modular arithmetics, thus loosing their geometric interpretations.

4. Elliptic Curve Cryptosystems

In practice, a cyclic subgroup of $E(\mathbb{F}_q)$ is used rather than the entire group. This amounts to choosing parameters a, b, p, k and an additional point $G \in E(\mathbb{F}_q)$ generating a sufficiently large cyclic subgroup of $E(\mathbb{F}_q)$. These parameters are shared publicly and are used for generating public keys. For simplicity, we will assume that $E(\mathbb{F}_q)$ is cyclic generated by a point $G \in E(\mathbb{F}_q)$ (This is not true in general - see [7] for details). Our assumption implies that any point P on the elliptic curve can be expressed as

$$P = nG = G + \cdots + G \tag{12}$$

where n is a non-negative integer.

Definition 2 (Elliptic Curve Discrete Logarithm Problem - ECDLP)**.** *Let P be a point in the cyclic subgroup of $E(\mathbb{F}_q)$ generated by G. Find the smallest integer n such that $P = nG$.*

The security of an elliptic curve based cryptosystem relies on the difficulty of the ECDLP. Given an integer n it's easy to compute the point nG, while on the other hand if a point P is of the form nG for some integer n then it is *computationally infeasible* to obtain the integer n. This means that one would require enormous amount of resources to be able to solve the problem. A related and more relevant problem is the Elliptic Curve Diffie-Hellman Problem (ECDHP) stated as follows.

Definition 3 (ECDHP)**.** *Let Q, R be points in the cyclic subgroup of $E(\mathbb{F}_q)$ generated by G such that $Q = n_Q G$ and $R = n_R G$ for some n_Q, n_R. Determine $P = n_Q n_R G$.*

It is easy to see that if one can solve ECDLP efficiently then one would be able to solve the ECDHP.

We now discuss how the ECDHP is used in generation and verification of elliptic curve based digital signatures. Recall that a digital signature is a cryptographic primitive which is fundamental in authentication, authorization and non-repudiation. Creation and verification of digital signature relies on the concept of a *one-way hash function.*

Definition 4. *A hash function H is a computationally efficient function mapping binary strings of arbitrary length to binary strings of a fixed length.*

The hash function H is required to satisfy the following properties:

- Collision resistance - it should be computationally infeasible to find distinct inputs m_1, m_2 such that $H(m_1) = H(m_2)$,
- Pre-image resistance - given an output value x it should be computationally infeasible to find an input m such that $H(m) = x$,
- Second pre-image resistance - given an input message m it should be computationally infeasible to find another input m' such that $H(m) = H(m')$.

4.1. Elliptic Curve Digital Signature Algorithm

In this scenario, Alice needs to send a message to Bob. Alice would digitally sign the message before transmitting it to Bob. Upon receipt, Bob would need to verify that indeed the message is from Alice and not from an impersonator Eve.

It is assumed that an elliptic curve $E(\mathbb{F}_q)$ and a point $G \in E(\mathbb{F}_q)$ of order l have been chosen and made public. For Alice to sign a message m she needs to choose a random integer n_A as her private key, compute her public key as $K_A = n_A G$. The public key is made public by depositing it at a Trusted Authority (TA).

ECDSA - Message Signing Algorithm:

1. Alice selects a random integer k such that $1 < k < l$.
2. She computes $kG = (x_1, y_1) \in E(\mathbb{F}_q)$ and set $r = x_1 \bmod l$. If $r = 0$ go back to 1.
3. She computes $k^{-1} \bmod l$.
4. She computes $s = k^{-1}(H(m) + n_A r) \bmod l$, where H is a publicly available hash function.
5. She sends the signed message (m, s, r) to Bob.

ECDSA - Signature Verification Algorithm: Bob receives the message as (m', s', r'). He will first obtain Alice's public key K_A from the TA then use the following procedure:

1. Bob verifies that r' and s' lies in the interval $[1, l-1]$,
2. He then computes $w = s'^{-1} \bmod l$ and the message hash $H(m')$.
3. He then computes $u_1 = H(m')w \bmod l$ and $u_2 = r'w \bmod l$.
4. Finally he computes $u_1 G + u_2 K_a = (x_0, y_0)$ and set $v = x_0 \bmod l$.

If the message if authentic i.e. $(m', s', r') = (m, s, r)$ then we have

$$\begin{aligned} k &= s^{-1}(H(m) + n_A r) \bmod l \\ &= w(H(m) + n_A r) \\ &= u_1 + u_2 n_A. \end{aligned}$$

Thus $kG = (u_1 + u_2 n_A)G = u_1 G + u_2 K_a = (x_0, y_0)$. Hence Bob accepts the message as authentic if and only if $v = r'$.

Now suppose that Eve would like to impersonate Alice. She creates a fictitious message m, picks a random integer k, computes $kQ = (x_1, y_1)$ and chooses $r = x_1 \bmod l$. She also computes $H(m)$ and k^{-1}. Her task will then be to figure out the correct s such that when used in the verification procedure it yields v such that $v = r$. But s depends on Alice's secret key n_A and so Eve would have to employ any technique that either gives her the correct s without n_A or solve the elliptic curve discrete logarithm problem to obtain Alice's secret key by using Q and Alice's public key $K_A = n_A Q$. If we assume that Eve has succeed in finding the correct s then we know that

$$r = v = H(m)wQ + rwK_A$$

where $w = s^{-1} \bmod l$. Thus $skQ = H(m)Q + rK_A$. Letting z be the inverse of $r \bmod l$ then we have

$$z(sk - H(m))Q = K_A = n_A Q$$

This means $z(sk - H(m)) = n_A = \log_Q(K_A)$ and thus Eve would have solved the elliptic curve discrete logarithm problem.

5. Use of Digital Signatures in Bitcoin

For the rest of this article, we describe how digital signatures are applied in Bitcoin technology. In a nutshell, the basic concepts of Bitcoin are:

- Bitcoin address - a digital analogue of a physical wallet,
- Transactions - records of transfer of Bitcoins from one address to another address,
- Blockchain - a publicly available ledger of all the transactions that has ever taken place.

The elliptic curve chosen by Satoshi Nakamoto defined by the equation

$$y^2 = x^3 + 7 \bmod p \tag{13}$$

where p is the integer

$$\begin{aligned} p &= 2^{256} - 2^{32} - 977 \\ &= 115792089237316195423570985008687907853269984665640564039457584007908834671663. \end{aligned}$$

The data above defines an elliptic curve $E(\mathbb{F}_q)$ of order

$$q = 115792089237316195423570985008687907852837564279074904382605163141518161494337.$$

Other piece of publicly available data used in Bitcoin is the point $G = (x_0, y_0) \in E(\mathbb{F}_q)$ - a generator of a cyclic subgroup of $E(\mathbb{F}_q)$. The coordinates of G are

$$\begin{aligned} x_0 &= 55066263022277343669578718895168534326250603453777594175500187360389116729240, \\ y_0 &= 32670510020758816978083085130507043184471273380659243275938904335757337482424. \end{aligned}$$

The hashing algorithm used in Bitcoin is the SHA-256 introduced by the American National Institute for Standards and Technology [3, 12, 10]. In practice this hashing algorithm is applied twice (denoted by SHA-256^2) so as to improve the security of transactions. For our purposes we shall simply represent SHA-256^2 as H.

5.1. Bitcoin Addresses

While handling physical money, one needs a wallet, an account, a pocket e.t.c to temporarily store the money. Similarly in Bitcoin technology, one needs a digital address to hold the digital money. These addresses will look like the following:

12C4GbvvWGihHjda4y8y48he1EGjDAnMDr

To create a Bitcoin address, Alice chooses a private key n_A which is a random integer, computes and publishes her public key $K_A = n_A G$. To generate an address, the public key K_A is hashed with $SHA-256$ and RIPEMD-160 [9, 2] a number of times before being encoded using to Base-58. Note that it is computationally infeasible to obtain the private key n_A given either the public key or the Bitcoin address. This will amount to solving the ECDLP.

5.2. Bitcoin Transactions

Bitcoin transactions are public messages which are digitally signed and broadcasted to the entire Bitcoin network for verification. Assume that Alice has 1.0005 BTC and would like to send 1 BTC to Bob. In layman's terms Alice would broadcast a message like

Other participants in the Bitcoin network needs to verify that indeed Alice is the owner of the Bitcoin amount and additionally verify that its indeed Alice who has initiated the transaction. All these verifications are done using ECDSA. Specifically, the message will comprise of three main parts:

- **An input**: a Bitcoin address from which Alice received the amount she is about to send to Bob,
- **An amount**: the specified amount in BTC that Alice intends to send to Bob,
- **An output**: a Bitcoin address owned by Bob which is set to receive the amount in BTC.

Alice would digitally sign the broadcasted message which would then be effected if and only if the Bitcoin network is able to verify (ECDSA Verification Algorithm) that the message is indeed authored by Alice.

5.3. The Bitcoin Blockchain

In a nutshell, we can say that the Bitcoin blockchain is a publicly available digital ledger that records the transactions between various Bitcoin addresses. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. And, once entered, information can never be erased. The blockchain contains a certain and verifiable record of every single transaction ever made.

6. Conclusion

The security of a crypto-system based on elliptic curves derives its security from the computational infeasibility of the Elliptic Curve Discrete Logarithm Problem. It is the difficultly of solving this problem that also assures the Bitcoin network of the security and authenticity of transactions on the blockchain.

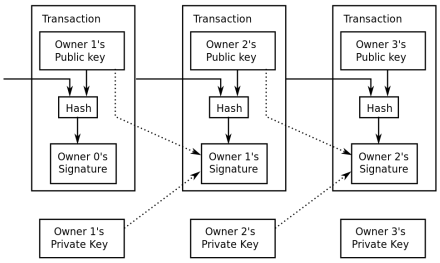

Figure 1: Graphical representation of entries in a Bitcoin blockchain.

References

[1] Diffie, W., and Hellman, M. New directions in cryptography. *IEEE transactions on Information Theory 22*, 6 (1976), 644–654.

[2] Dobbertin, H., Bosselaers, A., and Preneel, B. Ripemd-160: A strengthened version of ripemd. In *Fast Software Encryption* (1996), Springer, pp. 71–82.

[3] FIPS, P. 180-1. secure hash standard. *National Institute of Standards and Technology 17* (1995), 45.

[4] Hartshorne, R. *Algebraic Geometry*. Graduate Texts in Mathematics. Springer New York, 2013.

[5] Koblitz, N. Elliptic curve cryptosystems. *Mathematics of computation 48*, 177 (1987), 203–209.

[6] Miller, V. S. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques* (1985), Springer, pp. 417–426.

[7] Mordell, L. J. On the rational resolutions of the indeterminate equations of the third and fourth degree. In *Proc. Cambridge Phil. Soc.* (1922), vol. 21, pp. 179–192.

[8] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system, 2008.

[9] Preneel, B., Bosselaers, A., and Dobbertin, H. The cryptographic hash function ripemd-160, 1997.

[10] Pub, N. F. 197: Advanced encryption standard (aes), federal information processing standards publication 197, us department of commerce/nist, november 26, 2001. available from the nist website.

[11] Silverman, J. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer New York, 2013.

[12] Standard, S. H. Fips pub 180-2. *National Institute of Standards and Technology* (2002).