An Introduction to Bitcoin, Elliptic Curves and the Mathematics of ECDSA

N. Mistry B121555
Supervisor: Dr B. Winn
Module Code: MAC200

21.4.2015

Abstract

Bitcoin is a completely revolutionary peer to peer electronic cash system that is decentralised and removes the need for trusted third parties like banks. Instead Bitcoin creates trust by using an area of mathematics known as cryptography. Cryptography is usually based on the mathematics of prime numbers, however Bitcoin uses the mathematics of elliptic curves as the foundation for its cryptography. By using cryptographic methods the Bitcoin protocol can replace a central institution such as a bank or reserve and carry out the functions of cash generation and transaction authorisation using its decentralised peer to peer network. In this project I will introduce the fundamental concepts of cryptography and explain how transactions are created and processed in the Bitcoin system. The Bitcoin system relies on both cryptographic hashes and digital signatures to keep the system secure and create trust. I will provide a quick primer on elliptic curves and point operations and will explain how Bitcoin uses digital signatures based on elliptic curves to authenticate transactions and keep the system secure.

Contents

1 Introduction

Bitcoin is a cryptographically secure decentralised peer to peer (P2P) electronic payment system which enables transactions involving virtual currency. To compare this to everyday examples; PayPal, Visa and MasterCard are all payment systems that most readers are familiar with. PayPal has the most parallels with Bitcoin since they both involve transactions occurring online with digital cash. All the mentioned systems involve payments being made in everyday currencies (known as fiat currencies) such as the pound, dollar and euro. However the currency used by the Bitcoin system to store and transfer value is a virtual currency called bitcoin (BTC). Fiat currencies rely on banks and reserves to generate new currency and validate transactions. Since Bitcoin is decentralised, it can't rely on a single entity to control the currency, instead Bitcoin relies on cryptography to generate currency and validate transactions. Unlike traditional fiat currencies, bitcoins are entirely virtual meaning there are no tangible coins or digital files. The coins are implied in transactions that transfer value between sender and recipient [1].

Cryptography is a relatively new area of mathematics that deals with keeping message sent between two parties secure. Cryptography has its foundations in number theory, specifically using the mathematics of prime numbers to keep messages secure. Bitcoin does not make use of prime numbers in its cryptographic functions instead it uses elliptic curves. Bitcoin uses the elliptic curve digital signature algorithm (ECDSA) as the basis of its security and trust. The reason Bitcoin uses elliptic curves instead of prime numbers is because computations using elliptic curves use less CPU and memory, making it a more efficient method.

This report will outline how the Bitcoin system operates, i.e. how currency is generated, how transactions are kept secure and processed. There are two major areas of mathematics used in the Bitcoin system. The first is related to digital signatures and how transactions are authenticated and the second is related to the SHA-256 hashing algorithm used in the processing of transactions. This report will focus on the authentication of Bitcoin transactions. The Bitcoin system uses the Elliptic Curve Digital Signature (ECDSA) cryptographic algorithm to authenticate Bitcoin transactions.

The structure of this report is as follows. We will first introduce in Chapter 2 some introductory concepts and explain key definitions such as decentralised and peer to peer. A brief history of Bitcoin will be given along with a quick background in cryptography. This chapter should give the reader sufficient background knowledge to understand the rest of the report. In Chapter 3 the reader will be shown how the Bitcoin system works. First we will explain the concepts of a Bitcoin wallet and the block chain. We will then use an example to explain how transactions are created and processed. Chapter 4 will go on to introduce the mathematics required to understand elliptic curves. We will discuss operations on elliptic curves and explain how and why these are used by Bitcoin to secure transactions.

2 Background

2.1 Styling

Firstly we need to introduce some styling that will be used throughout this report to differentiate between Bitcoin the payment network and bitcoin the currency. According to the official Bitcoin Foundation "Bitcoin - with capitalization, is used when describing the concept of Bitcoin, or the entire network itself. e.g. 'I was learning about the Bitcoin protocol today.' bitcoin - without capitalization, is used to describe bitcoins as a unit of account. e.g. 'I sent ten bitcoins today.'; it is also often abbreviated BTC or XBT." [2]

2.2 Key Vocabulary

This section will provide key definitions that will be used throughout this report.

2.2.1 Decentralised

Bitcoin is described as a decentralised currency. To explain this I will first outline what a centralised currency is. Fiat money is currency that derives its value from government legislation. Examples of this include the dollar, pound and euro. Lets take the instance of the dollar. The dollar is controlled by the US Federal Reserve. It controls the printing, distribution and value of the currency. The dollar is controlled by a single entity and this is what we call a centralised currency. Hence a decentralised currency is the opposite of this; it is not controlled by one single entity. In the case of Bitcoin the currency is controlled by all the members that participate in Bitcoin transactions.

2.2.2 Peer to Peer

The Bitcoin system is a Peer to Peer (P2P) network. This is a network with no central location, where a group of computers are connected to enable the sharing of resources and information by users [14]. On the Bitcoin network the peers are the individual parties taking part in transactions and the P2P nature of the network means that payments are sent directly from one peer to another without the need to go through a third party such as a bank. The P2P nature of the network is what allows Bitcoin to be decentralised. Examples of P2P systems include Skype and BitTorrent.

2.2.3 Digital and Crypto Currencies

As per Wagner's article in bitcoin magazine [11] digital money are currencies that are stored and transferred purely electronically. A subset of digital currencies are cryptocurrencies. The concept of cryptocurrency was first described by Wei Dai in 1998 on a mailing list. Dai [6] suggested a new form of currency that used cryptographic techniques to regulate the generation of currency and the verification of transactions. This removed the need for a central authority to carry out these functions, Bitcoin is built upon the ideas of Dai along with many other ideas.

2.3 History

Bitcoin was first proposed by Satoshi Nakamoto in 2008 when he published the paper, 'Bitcoin: A Peer-to-Peer Electronic Cash System' through a cryptography mailing list [16]. Satoshi Nakamoto is a pseudonym, and to this day it is not known who he/she is. Bitcoin was first introduced in 2009 as open source software and became the world's first decentralised cryptocurrency [16]. As per the Nakamoto paper Bitcoin is essentially a "peer to peer version of electronic cash that would allow online payments to be sent directly from one party to another without going through a financial institution"[9].

2.4 Cryptography Background

"Cryptography is the study of methods of sending messages in disguised form so that only the intended recipients can remove the disguise and read the message" [8]. "The message we want to send is called the plaintext [P] and the disguised message is called the ciphertext [C]" [8]. The process of converting a plain text to a cipher text is called enciphering or encryption, and the reverse operation is called deciphering or decryption [8]. Encryption and decryption is usually done by something called cryptographic keys.

2.4.1 Private vs Public Key

Private key cryptosystems, are algorithms that use the same cryptographic keys for enciphering plaintext and deciphering ciphertext. The keys may be identical or there may be a simple transformation to go between the keys. The keys are shared between two or more parties and are used to maintain a private channel to share messages. The requirement of this system is that both parties already know the secret keys. Herein lies the disadvantage of a private key system. If for example two parties are geographically separated and wish to set up a new private channel to share messages, how do they share the enciphering and deciphering keys ?(assuming that all other means of communication is insecure).

This is where the newer Public key system comes in. Public key algorithms require two separate keys; a private key which is kept secret, and a public key which is shared with other parties. The public key is mathematically derived from the private key. The public key is used to encrypt plaintext (and as discussed below verify digital signatures), whereas the private key is used to decrypt ciphertext (or create digital signatures).

The key property of a public key cryptosystem is that someone who knows the public encrypting key cannot use this to derive the private decrypting key without carrying out a long winded computation which makes the process essentially impossible. So in other words the encrypting function:

$$f : P \to C$$

is easy to compute with knowledge of the encrypting key K_E. However it is very difficult to compute the inverse function:

$$f^{-1} : C \to P$$

with only the knowledge of the encrypting key K_E.

So we can conclude that f is not an invertible function with regards to realistic computability when only the encrypting key K_E is known and the decrypting key K_D unknown, as is the case in real life [8]. Of course the inverse f^{-1} is easy to compute for someone who has the deciphering key K_D. Such a function f is called a *trap door function.*

So a *trap door function* f is "a function which is easy to compute but whose inverse f^{-1} is hard to compute without having some additional auxiliary information beyond what is necessary to compute f" [8]. Trap door functions are based on mathematical problems that are inherently easy to compute in one direction, but difficult in the opposite. Examples of such problems include prime number factorisation, the discrete logarithm problem and elliptic curve multiplication. Bitcoin is built upon the public key cryptosystem and as mentioned previously relies on elliptic curve mathematics. The trap door function employed by Bitcoin is based upon the multiplication of points on an elliptic curve over a finite field. An in depth insight into this will be given in Section 4

2.4.2 Authentication [8]

A signature is an important part of a message since it is difficult to reproduce by an imposter. It gives the recipient assurance that the message is actually from the person who sent it. In an electronic message, there are no physical signatures and hence other methods are required to authenticate a message.

Let A (Alice) and B (Bob) be two users of a public key system. Let f_A be the encrypting transformation with which any users of the public key system sends a message to Alice (i.e. Alice's public key), and let f_B be the same for Bob. Let P be Alice's signature. Alice could send Bob the encoded message

$$f_B(P)$$

as a signature. However Bob has no way of knowing that the signature P is not forged. Anyone could forge Alice's signature and send it to Bob. To overcome this Alice transmits

$$f_B f_A^{-1}(P)$$

at the end of the message. When Bob decrypts the whole message by applying his decryption function based on his private key f_B^{-1}, he will find that everything becomes plain text except for

$$f_A^{-1}(P).$$

Since Bob knows that the message is from Alice, he applies f_A (which is known, since Alice's encrypting key is public) and obtains P. Since no one other then Alice could have applied Alice's decryption function f_A^{-1}, since Alice keeps this private, he knows that this message is from Alice. This is the basis of the digital signatures used by Bitcoin to authenticate transactions and ensure that imposters cannot spend other people's bitcoins.

2.4.3 Hash functions

"A hash function is an easily computable map

$$f : x \mapsto h$$

from a very long input x to a much shorter output h called the hash (e.g. from about 10^6 bits to string of 150 or 200 bits)" [8]. This output h has the property that it is "not computationally feasible to find two different inputs x and x' such that $f(x') = f(x)$" [8]. This means that two different inputs will never give the same output. Hence the hash function gives unique outputs.

The hash function can be used to help sign a document. Alice should take a hash of her message x giving the hash $h = f(x)$. She should then include this in her signature when sending the message to Bob. Bob can now verify that the message x hasn't been interfered with by taking a hash of the message he receives and comparing it to the hash Alice included in her signature. Since the hash function gives unique outputs, if A's and B's hashes match, Bob knows that the message has not been tampered with.

Hash functions are used throughout the Bitcoin system, however the ones used here are cryptographic hash functions. These are hash functions that make use of trap door functions in the same way as public key cryptography does. As a result these hash functions are considered impossible to invert. That is, if someone knows h they are not able to recreate the original message x using the hash function f. Cryptographic hash functions are most notably used in the creation of Bitcoin addresses, Digital signatures and the processing of Bitcoin transactions by members of the network called miners. The two cryptographic hash functions used by Bitcoin are the Secure Hash Algorithm (SHA-256) and the RACE Integrity Primitives Evaluation Message Digest (RIPEMD160).

3 How Bitcoin Works

The majority of the background and theory for this Chapter has been taken from the book 'Mastering Bitcoin' by Andreas Antonopoulos [1]. This book provides an excellent insight into Bitcoin, and I recommend that any readers wanting to ascertain a deeper understanding read this text. Nevertheless in this chapter I have built upon the explanations of Antonopoulos and added my own understanding, analogies and explanations, whilst linking the content back to how elliptic curves are used to keep Bitcoin secure.

3.1 Further Background

3.1.1 Mining, Proof of Work and the Money Supply

We have outlined that Bitcoin is an online payment system that allows two parties to exchange payments directly without using an external party. Bitcoin is not a physical currency it is completely virtual, and therefore no physical coins or digital files are transferred in transactions. Instead "[t]he coins [bitcoins] are implied in transactions which transfer value from sender to recipient"[1]. Bitcoin is decentralised and hence there is no equivalent of a central bank that prints money and regulates the money supply. Instead bitcoins are created and added to the money supply through a process known as mining.

Mining involves participating members of the Bitcoin network known as miners solving a mathematical problem that helps process transactions. This mathematical problem is known as a proof of work and uses cryptographic hash functions. Miners are able to verify a block of transactions every ten minutes. The miner that verifies these first is rewarded with new bitcoins. Hence mining creates a global competition with an financial reward to process transactions whilst also decentralising the currency issue and clearing operations of a central bank [1].

The hashing algorithms used to process transactions also regulate the speed of mining. As per Antonopoulos [1] the difficulty of the proof of work task is adjusted dynamically so that on average someone succeeds every ten minutes regardless of how many miners are working on the task. The number of bitcoins rewarded is halved every four years and the issuance of bitcoins stops when a total of 21 million bitcoins are in existence. This means that the circulation of bitcoins follows an easy to predict curve (Figure 1) that reaches 21 million in 2140 [1]. After this no more bitcoins are produced. Miners also earn bitcoins from fees that are built into transactions. After 2140 this will be the main way to earn bitcoins for miners.

Mining of bitcoins has many analogues to the mining of gold. There is only a finite supply of gold and there is diminishing returns; the more you mine it, the more scarce it gets and the more difficult it is to mine.

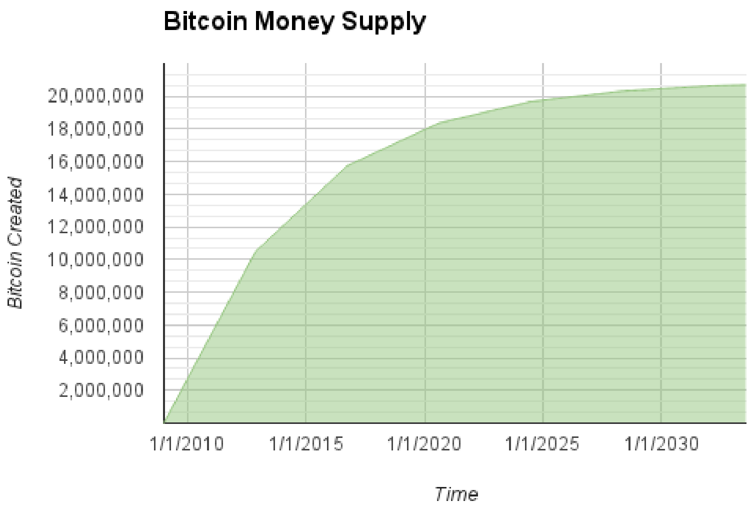


Figure 1: Supply of bitcoin currency over time [1]

3.1.2 The Problems of Digital Money

The two basic issues for anyone accepting digital money are counterfeiting and the double spending problem.

The double spending problem is the result of successfully spending some money twice. This was a major issue for early digital currencies. The problem of double spending is largely irrelevant in paper money, since a physical coin cannot be in two places at once and hence cannot be spent twice at the same time. When money is transmitted digitally (e.g. Paypal), the problem of double spending is overcome by the use of central clearing houses that have a global view of the money supply [1]. Paper money attempts to overcome counterfeiting by using increasingly advanced printing techniques such as holograms. The aim of a decentralised digital currency is for there to be no central authority to clear transactions. Instead Bitcoin uses cryptography as "the basis for trusting the legitimacy of a user's claim to value" [1]. Cryptography is based on mathematics, whose operations and proofs are irrefutable and hence provides a solid foundation for trust in Bitcoin. Bitcoin specifically employs cryptographic digital signatures (2.4.2) that enable users to sign a digital transaction proving ownership of that transaction.

In his work paper Satoshi Nakamoto built on the ideas of Dai's 'b-money' and Back's 'HashCash - a denial of service counter-measure' to develop a decentralised electronic cash system. As Antonopoulos describes " [t]he key innovation was to use a distributed computation system called a 'Proof-Of-Work algorithm' [as described above] to conduct a 'global election' every ten minutes, allowing the de-centralised network to arrive at a consensus about the state of transactions" [1]. This key innovation overcomes double spending and also is the mechanism that allows transactions to be verified. This invention by Nakamoto is what has made Bitcoin the most successful digital currency and differentiated it from earlier attempts.

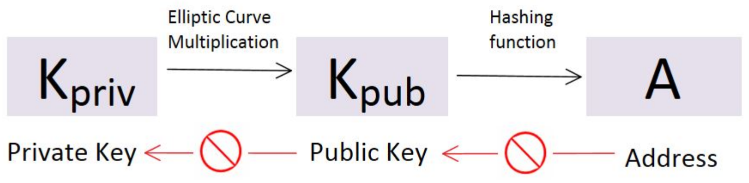


Figure 2: Private key, public key and bitcoin address

3.2 Bitcoin Wallet and Addresses

To start using Bitcoin the user must first download an application that serves as a Bitcoin wallet. This is essentially the equivalent of a "physical wallet on the Bitcoin network" [2]. Bitcoin uses a type of public key cryptography (see 2.4.1) that involves the use of elliptic curves. The wallet software creates a public and private key. The private key can be thought of as a PIN that enables you to access your bitcoins and authorise payments, and must be kept secret. The public key can be thought of as a bank account number. The public key is mathematically derived from the private key using elliptic curve multiplication. A cryptographic hash function (see 2.4.3) is applied to the public key to produce a Bitcoin address. This sequence is illustrated in Figure 2.

The mathematical properties of both elliptic curve multiplication and hash functions mean that it is computationally impossible to reverse. Hence a user can publicly publish their Bitcoin address in the knowledge that other people cannot compute their private key from it. In Section 4.5 we explain with reference to elliptic curves how the public key is generated. Each address has its own balance of bitcoins, and transactions are essentially the transfer of bitcoins between these addresses. In a transaction between Alice and Bob worth 5BTC a transaction can be thought of an instruction to reduce the balance of Alice's address by 5BTC and increase Bob's address by 5BTC.

3.3 Block Chain

The next important concept to explain is the block chain. The block chain is a ledger that records every transaction that occurs between peers on the Bitcoin P2P network in chronological order. The twist here is that this ledger is public. It can be accessed by anyone, and so every confirmed transaction that has ever taken place in the history of Bitcoin is publicly available. One can use block chain explorers such as `blockchain.info` to explore such transactions. The block chain is constantly updated with new transactions. Users of the network compete to process a block of transactions and add them to the block chain in a activity known as mining. As we know from 3.1.1 mining also results in the generation of new bitcoins. The new block chain in then broadcast out to the network and all users are up to date.

Since a transaction is a transfer of bitcoins between addresses, the block chain which contains all details of these transfers can be used to reconstruct the balances of each addresses at any given time. This means everyone is able to verify that a party spending bitcoins actually owns them. This is key in the process of verifying transactions as will be described in 3.4.

3.4 Transactions

3.4.1 Motivating Example

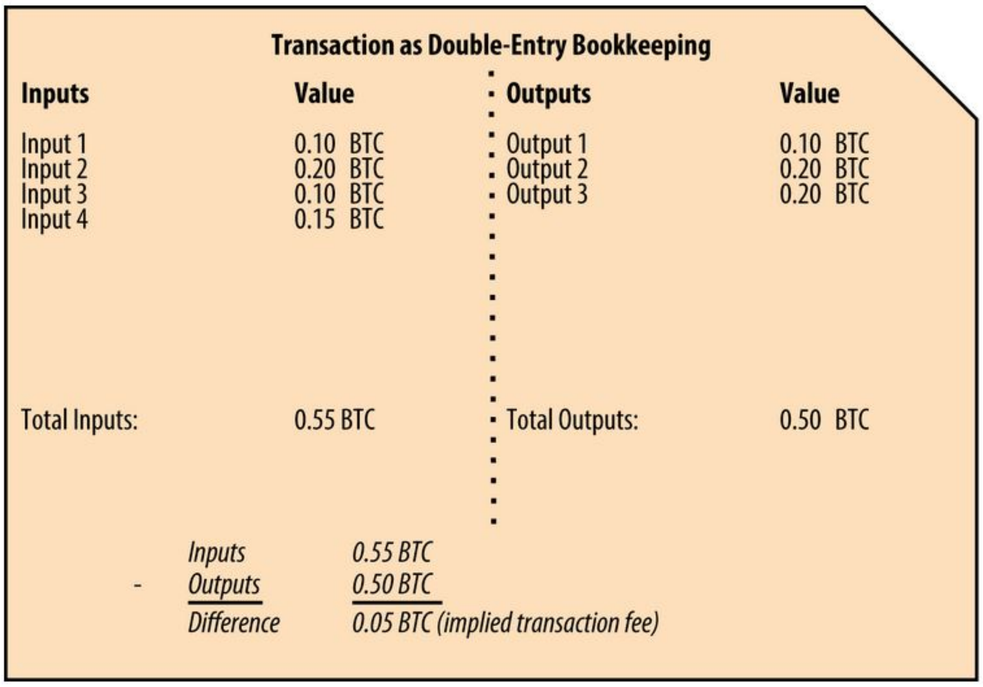
We will explain how transactions work by using an example taken from Mastering Bitcoin, Antonopoulos [1]. The parties involved in this example are Alice (A) her friend Joe (J) and the coffee shop owner Bob (B). Let's say A exchanges cash with J for some bitcoins. The transaction created by J funded A's wallet with 0.1BTC. Using her bitcoin balance A is now able to buy coffee for 0.015BTC from B who now accepts bitcoins at his store. As Antonopoulos notes, simplistically "a transaction tells the network that the owner of a number of bitcoins has authorised the transfer of some of those bitcoins to another owner. The new owner can now spend those bitcoins by creating another transaction that authorises transfer of them to another owner, and so on, in a chain of ownership" [1].

3.4.2 Components of a Transaction

Antonopoulos [1] likens Bitcoin transactions to that of a double entry book keeping ledger and this provides a novel way to think of transactions. Such a ledger has two sides with debits on the left side (inputs) and credits (outputs) on the other side. Money coming in is written on the left hand side, and any outgoings are recorded on the right. In a similar way every transaction has one or more inputs and outputs. The transaction inputs/outputs are debits/credits to the Bitcoin account.
Usually the value of the inputs are greater than the output, the difference being a transaction fee taken by the miner that adds the transaction to the block chain as mentioned in 3.1.1. A Bitcoin transaction represented as a book keeping ledger is shown in Figure 3.

As well as containing an input and an output, a transaction also contains digital signatures which serve as proof that the spender owns the bitcoins (i.e. the inputs of the transaction). The value of a coin comes from the input which is usually an output of a previous transaction. The owner of this value needs to provide a digital signature using their private key to prove that they own these bitcoins and can spend them in a new transaction. Once this is verified the transaction can occur. An output of a transaction assigns the value of the coin to a new owner by associating it with the new owner's Bitcoin address (derived from their public key). By doing this, these bitcoins are 'locked' to the new owner, and the new owner must present their digital signature in order to use these bitcoins as inputs in a new transaction.

Using this we can ascertain that spending a bitcoin is essentially the transfer of value from transaction inputs to outputs and this is authorised by the use of a digital signature. This process creates a chain of digital signatures as more outputs are used as inputs in new transactions and value is moved between different Bitcoin addresses. So this creates a chain of ownership for each bitcoin. It is useful to now think of a bitcoin as a chain of digital signatures rather than an actual coin.



Transaction as Double-Entry Bookkeeping

Inputs	Value	Outputs	Value
Input 1	0.10 BTC	Output 1	0.10 BTC
Input 2	0.20 BTC	Output 2	0.20 BTC
Input 3	0.10 BTC	Output 3	0.20 BTC
Input 4	0.15 BTC		
Total Inputs:	0.55 BTC	Total Outputs:	0.50 BTC

	Inputs	*0.55 BTC*
-	*Outputs*	*0.50 BTC*
	Difference	*0.05 BTC (implied transaction fee)*

Figure 3: A transaction represented as double-entry book keeping [1]

3.4.3 Application to Our Example

Going back to our example Alice has 0.1BTC locked to her Bitcoin address as a result of the transaction between her and Joe. The 0.1BTC to Alice was an output of this transaction. Alice will now use these bitcoins as inputs for any new transactions she makes. However before she can use these bitcoins as inputs in the transaction to buy Bob's coffee for 0.015BTC she must provide a digital signature to prove that she owns these 0.1BTC. Once this is done she can reference these bitcoins as her transaction input. Now since the cost of coffee is not 0.1BTC, Alice will receive change. This is done by adding her Bitcoin address as an output as well as Bob. So the transaction is constructed as follows:

1. The output from the transaction between Alice and Joe is used as the input for the new transaction between Alice and Bob. Alice proved she owned these inputs by using her digital signature.

2. Two outputs are created. One output is to Alice's address for 0.0845BTC and this represents the change she will receive. The second output is to Bob's address for 0.0150BTC and this is the payment for the coffee.

3. Note the inputs and outputs do not total to the same value and this is due to the transaction fee taken by miners. In the case of this transaction the fee is 0.0005 BTC.

Recall that when Alice 'locks' the 0.015 BTC to Bob's Bitcoin address, Bob must present his digital signature to spend these bitcoins in future transactions as inputs. The chain of transactions can be seen in Figure 4. The wallet software of the user builds the transaction by selecting appropriate inputs and outputs. All Alice has

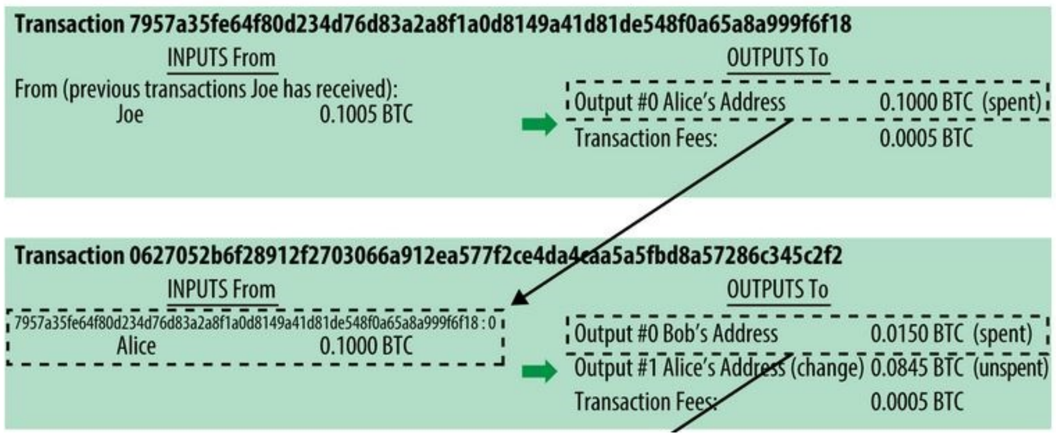

Transaction 7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18
INPUTS From
OUTPUTS To
From (previous transactions Joe has received):
Joe 0.1005 BTC
Output #0 Alice's Address 0.1000 BTC (spent)
Transaction Fees: 0.0005 BTC
Transaction 0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2
INPUTS From
OUTPUTS To
7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18 : 0
Alice 0.1000 BTC
Output #0 Bob's Address 0.0150 BTC (spent)
Output #1 Alice's Address (change) 0.0845 BTC (unspent)
Transaction Fees: 0.0005 BTC

Figure 4: The chain of transaction between Joe Alice and Bob [1]

to do to initiate a transaction is provide a Bitcoin address along with the amount she wishes to pay. Once the transaction is created by the wallet, it is transmitted to the Bitcoin P2P network. The transaction propagates through the system as users/peers share it.

3.5 Mining

So the transaction has been broadcast to the whole network, however it doesn't become part of the block chain till it is verified by a miner. The process of mining as discussed in 3.1.1 firstly generates new bitcoins and secondly verifies transactions. To confirm that a transaction is legitimate we need to place trust in it. Bitcoin use computation to create trust. During the process of mining transactions are sorted into blocks and the miner must solve the proof of work to verify these transactions and add them to the block chain. The proof of work requires a lot of computation to solve but it is easy to verify as solved. Transactions are only confirmed (i.e. marked as trusted) if enough computational power was dedicated to the block that contains it [1]. The principle is that "[m]ore blocks mean more computation which means more trust" [1].

3.5.1 A Giant Sudoku Competition [1]

Antonopoulos [1] explores an interesting idea that helps explain how the mining process works by likening it to a Sudoku puzzle. Let's suppose there is a "giant competitive game of Sudoku that resets every time someone finds a solution and whose difficulty automatically adjusts so that it takes approximately 10 minutes to find a solution" [1]. Let this giant Sudoku puzzle have a large number of rows and columns. If you are shown a "completed puzzle you can verify it quite quickly. However, if the puzzle has a few squares filled and the rest is empty, it takes a lot of work to solve! The difficulty of the Sudoku can be adjusted by changing its size (more or fewer rows and columns), but it can still be verified quite easily even if it is very large" [1]. The 'puzzle' to be solved is the proof of work algorithm and has similar characteristics to the Sudoku puzzle: "it is asymmetrically hard to solve but easy to verify, and its difficulty can be adjusted" [1].

3.5.2 Proof of Work

A proof of work is a piece of data which satisfies certain requirements and was difficult (costly, time consuming) to produce [13]. However it must be easy to check that the data satisfies the requirements. Producing a proof of work can be a random process with low probability involving a lot of trial and error [13].

Bitcoin uses the HashCash proof of work. The difficulty of this work is dynamically changed so that a new block is generated once every ten minutes. The probability that a successful block is generated is low, meaning that it is unlikely that one worker computer can generate consecutive blocks. So what must a miner do to solve a proof of work? Well firstly the miner collects a group of transaction in a block. This block will have something called a header which contains a hash of the previous block, a hash of the transactions in this block and a time stamp. To solve the proof of work the miner must repeatedly hash the header of the block and a random number called a nonce with the SHA256 cryptographic algorithm until a solution matching a predetermined pattern emerges. This pattern usually requires the hash to start with a defined sequence of zero bits.

So the solution to the mathematical problem/the proof of work that we have mentioned throughout the report is this random number - the nonce. Miners are searching for the nonce that when hashed with the transactions produces a hash with a defined number of zero bits at the start. This is a process of trial and error and can only be achieved by brute force computation (trying random numbers till you get the solution). “The average work required [to do this] is exponential in the number of zero bits required but can be verified by executing a single hash” [9]. This gives us a puzzle that is asymmetrically hard to solve but easy to verify, and its difficulty can be adjusted (by changing the number of zero bits required).

So recall that mining can be thought of as a big competition, and we now know that the competition is to find the nonce for the block in question. The first miner to find the nonce wins the competition and publishes his block along with the nonce that when hashed with the block header produces the required zero bits. All the other miners are able to easily verify that this nonce satisfies the zero bit requirement by hashing the block with the nonce. So it is computationally difficult to find the nonce for a block, but it is easy to verify, much like a Sudoku puzzle once you are given the solution you can easily verify that it works.

Once the other miners agree that the block has been ‘solved’ and hence all the transaction within it verified, the block is added to the block chain and the successful miner receives his reward of bitcoins (currently 25BTC). An updated block chain is distributed throughout the network, and once the miners receive this new block chain they know they have lost the competition to mine the previous block. Transactions are constantly entering into the network, and so miners group another lot of transactions into a block and begin solving the proof of work for these transactions. Since the proof of work takes approximately ten minutes to solve, a new block is mined every ten minutes. Also note that each block contains a hash of the previous block, and this links blocks together in a chain - hence the term block chain.

3.5.3 Alice's Transaction

Referring back to our example from Mastering Bitcoin [1]. Once the transaction between A and B is broadcast to the network, the miners include it in a new block along with other transactions and begin to solve the proof of work. After about 10 minutes a miner would find the nonce that solves the proof of work and publish this for the other miners to verify. As we have mentioned this is a quick and easy process.

Once verified this block will be added to block chain as block 277316 for example and the successful miner gets the reward of mined bitcoins and the transaction fees. The new block chain is sent out to the network, and once other miners receive this new updated block chain they begin to mine the next block. Lets say another miner mines the next block - 277317. This block must have a hash of the previous block 277316 within it, thus linking 277317 and 277316. Since the new block is based on the previous block (277316) that contained Alice's transaction, we have added extra computation on top of that block (277316). Recall that trust is created in Bitcoin by computation; the more computation the more trust. Block 277316 now has another blocks worth of computation on top of it which increases the trust in the transactions in this block. One of the transactions in the block is Alice's, and we now have one 'confirmation' of this transaction. Each block mined on top of the one containing Alice's transaction is an additional confirmation [1].

As the blocks are added on top of each other, it becomes exponentially harder to reverse the transaction. The later blocks are chained after Alice's block, and the work needed to change this block would include redoing all the blocks after it, which would take a lot of computation. Hence the more blocks added on top of a transaction the more computation is needed to change the transaction and hence the more trusted it is by the network. "By convention, any block with more than six confirmations is considered irrevocable, as it would require an immense amount of computation to invalidate and recalculate six blocks" [1].

This chapter has given a brief insight into the workings of Bitcoin. As per the aim of this report, we go on to discuss the mathematics of key generation and digital signatures.

4 Bitcoin and Elliptic Curves

As mentioned in 3.2 the Bitcoin wallet generates a public and private key. The public key is mathematically derived from the private key using elliptic curve multiplication on a finite field. Hence to give a more rigorous explanation of how this is done, we will first introduce some theory on elliptic curves and fields.

4.1 Finite field

4.1.1 Fields [8]

A **field** is a set F with a multiplication and addition operation which satisfy the familiar rules:

- Associativity of both addition and multiplication
- Commutativity of both addition and multiplication
- The distributive law
- The existence of an additive identity 0
- The existence of an multiplicative identity 1
- Additive inverses for all non zero elements
- Multiplicative inverses for all non zero elements

Example 1 *The following are examples of fields*

1. *The field $\mathbb{Q}$ consisting of all rational numbers,*
2. *The field $\mathbb{R}$ consisting of real numbers,*
3. *The field $\mathbb{C}$ of complex numbers,*
4. *The field $\mathbb{Z}/p\mathbb{Z}$ of integers modulo a prime number p.*

4.1.2 Finite Fields

Finite fields are fields with finitely many elements. For example F_7 is a field with seven elements. In a finite field there is a integer n such that the addition of n terms equals zero i.e. $1 + 1 + \ldots + 1 = 0$. The smallest such n must be a prime number and is called the characteristic of the field [15]. A basic class of finite fields are the fields F_p with p elements, p being a prime number :

$$F_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$$

where the operations are defined by performing the operation in the set of integers $\mathbb{Z}$, dividing by p and taking the remainder [15]. This is called modular arithmetic. The key point here is that a finite field contains only integer numbers and no decimals.

4.2 Elliptic Curves

Let K be a field. K will be either the field $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{C}$ or the finite field F_q of $q = p^r$ elements. Let K be a field of characteristic $\neq 2, 3$, and let $x^3 + ax + b$ (where $a, b \in K$) be a cubic polynomial with no multiple roots. An elliptic curve over K is the set of points (x, y) with $x, y \in K$ which satisfy the equation,

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

together with a single element O called the point at infinity [8]. An example of an elliptic curve can be seen in Figure 5

4.3 Elliptic Curves over $\mathbb{R}$

For this section we will assume that $K = \mathbb{R}$, i.e. the elliptic curve is an ordinary curve in the plane. The theory for this section has been taken from Koblitz[8] and Certicom[3]. Let E be an elliptic curve over the real numbers, and let P and Q be two points on E. If P is the point at infinity (O as per Definition 4.2) then we define:

- $-P$ to be O
- $P + Q$ to be Q

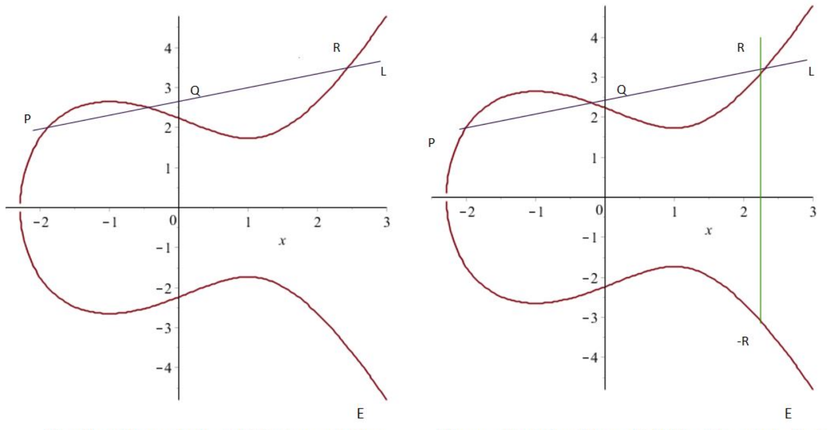
Hence O serves as the the additive identity (zero element) of the group of points. In what follows we will assume that $P \neq$ O. Now we define $-P$. This is the point with the same x coordinate but negative y coordinate. So if the coordinates of P are (x, y) then $-P$ is $(x, -y)$, which is also on the curve E. We will define two operations for elliptic curves; point addition and point multiplication.

4.3.1 Point Addition

Firstly let E be an elliptic curve over the real numbers, and let P and Q be two points on E where $P \neq Q$. To add these two points, first draw a line l through both of these points. The line l will intersect the curve E at a third point R. This is shown in the first image of Figure 5. We then draw a vertical line through R until it hits another point on E. The point is the mirror image of R in the x axis and as per the notation before we denote this $-R$, as shown in the second image of Figure 5. We define the sum $P + Q$ on E to be this reflected point $-R$.

4.3.2 Point Doubling

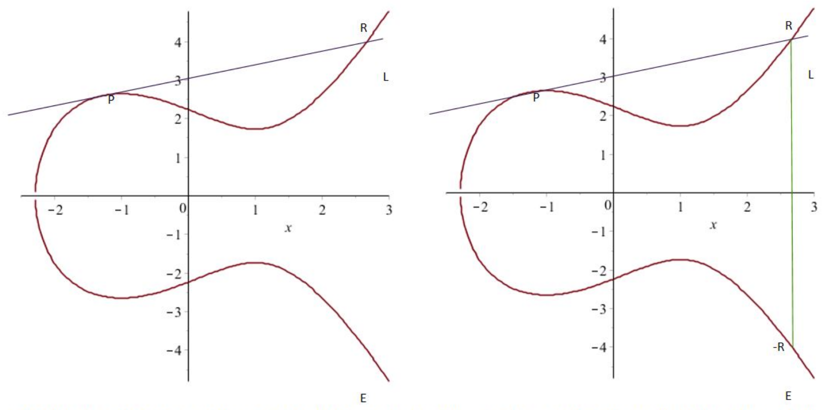
Now we describe how we add a point to itself on the elliptic curve, i.e. the case where $P = Q$. This is called point doubling. In this case we take the line l to be the tangent of P. This line then intersects a third point R on E, and we again reflect this across the x axis to obtain the point $-R$ which we call $2P$. See Figure 6 for a diagrammatic representation of this.



The line through P and Q intersects the curve E at a third point R.

The vertical line through R hits the curve E at point -R which we define to be P+Q

Figure 5: Point addition of P and Q



The line L is the tangent line at P and intersects E at R

Again we take the vertical line through R which hits the curve E at -R, which we define to be 2P.

Figure 6: Point doubling of P

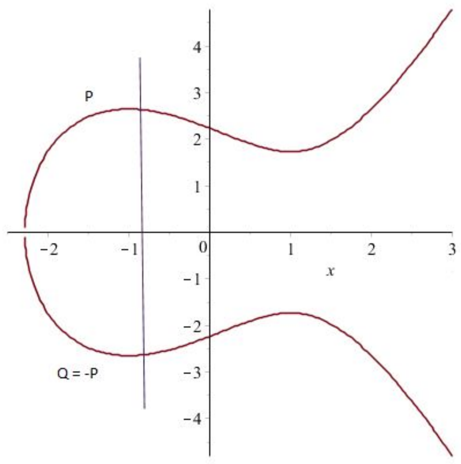


Figure 7: Vertical line through P

4.3.3 Vertical lines

As illustrated in Figure 7, the vertical line through P and $-P$ does not intersect E at a third point, and we require a third point to find $P + (-P)$. This point is O, the point at infinity, and this is a point on every vertical line.

4.3.4 Point Multiplication

In point multiplication a point P on the elliptic curve is multiplied by a scalar k to obtain another point Q on the same elliptic curve i.e. $kP = Q$. The point multiplication operation is carried out by using both point addition ($M + N = L$) and point doubling ($2M = S$). Let P be a point on an elliptic curve. Let k be a scalar that is multiplied with the point to obtain another point Q on the curve, i.e. to find $Q = kP$. Taking an example from [3], where $k = 23$, we have

$$kP = 23.P = 2(2(2(2P) + P) + P) + P.$$

Thus point multiplication uses point addition and doubling repeatedly to find the result. This method is called the *double and add method for point multiplication.*

A key point to note here is that if you have the point Q and you know P, there is no way to find the value of k. Since there is no point subtraction or point division, you cannot simply solve $k = Q/P$. This makes point multiplication essentially an irreversible operation, i.e. a one way function. As per 2.4.1 this is the trap door function employed by Bitcoin to keep the system cryptographically secure. The irreversibility of point multiplication is the basis of the security of the ECDSA algorithm.

4.3.5 An algebraic explanation: Point Addition [8]

Before we move onto elliptic curves on finite fields, we will first give an algebraic explanation of point addition and doubling. As you will see these operations are difficult to visualise when the curve is on a finite field, but the underlying mathematics presented here will be the same. The workings below have been taken from Koblitz [8].

Let (x_1, y_1), (x_2, y_2) and (x_3, y_3) denote the coordinates of P, Q and $P + Q$ respectively. Suppose $P \neq Q$. We want to express x_3 and y_3 in terms of x_1, y_1, x_2, y_2. Let $y = \alpha x + \beta$ be the equation of the line through P and Q. Then,

$$\alpha = \frac{(y_2 - y_1)}{(x_2 - x_1)} \text{ and } \beta = y_1 - \alpha x_1.$$

A point on the line l i.e. a point $(x, \alpha x + \beta)$, lies on the elliptic curve if and only if

$$(\alpha x + \beta)^2 = x^3 + ax + b.$$

Hence, there is one intersection point for each root of the cubic equation

$$x^3 - (\alpha x + \beta)^2 + ax + b.$$

We already know that there are two roots x_1,x_2, because $(x_1, \alpha x_1 + \beta), (x_2, \alpha x_2 + \beta)$ are the points P and Q on the curve. Since the sum of a monic polynomial (A polynomial where the leading coefficient is equal to 1) is equal to minus the coefficient of the second-to-highest power, we conclude that the third root in this case is

$$x_3 = \alpha^2 - x_1 - x_2.$$

This leads to an expression for x_3, and hence $P + Q = (x_3, -(\alpha x_3 + \beta))$, in terms of x_1, x_2, y_1, y_2:

$$x_3 = (\frac{y_2 - y_1}{x_2 - x_1})^2 - x_1 - x_2; \qquad y_3 = -y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x_1 - x_3) \tag{2}$$

4.3.6 An algebraic explanation: Point doubling [8]

Now assume $P = Q$. This case is similar, however α is now the derivative dy/dx at P. Implicit differentiation of Equation 1 leads to the formula $\alpha = (3{x_1}^2 + a)/2y_1$, and so we obtain the following formulas for the coordinates of twice P:

$$x_3 = (\frac{3x_1^2 + a}{2y_1})^2 - 2x_1; \qquad y_3 = -y_1 + \frac{3x_1^2 + a}{2y_1}(x_1 - x_3) \tag{3}$$

4.4 Elliptic Curve on a Finite Field F_p

Decimal numbers are irrational and computers do not have space to store all of the digits. As a result decimal numbers are difficult for a computer to process. Since Bitcoin uses elliptic curves throughout its system, we must find a way to remove all decimal numbers so that a computer can quickly carry out computations. In 4.3 the field used by the elliptic curve were the rational numbers, which produce decimal results. This is why the curve is smooth and continuous. However in order to remove all decimal results we now consider elliptic curve over finite fields. Recall from 4.1.2 that finite fields use modular arithmetic to produce only integer results. This removes all decimals and hence makes computations easier for a computer. An elliptic curve over a finite field becomes very different to the curves we saw in 4.3. Instead of a smooth curve, the graph looks like a pattern of dots. Although the operations of point addition and doubling are difficult to see visually, the underlying mathematics of how to find these points stay the same.

Bitcoin uses a specific elliptic curve with a set of constants which is defined in the `secp256k1` standard established by the National Institute of Standards and Technology (NIST). As per [4] the elliptic curve domain parameters over F_P associated with a Koblitz curve `secp265k1` are specified by the sextuple:

$$T = (p, a, b, G, n, h)$$

where the finite field F_P is defined by (note parameters are given in hexadecimal):

$p =$ FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F
$= 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$

The curve $E : y^2 = x^3 + ax + b$ over F_p is defined by:

$a =$ 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

$b =$ 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007

The base point G in compressed form is:

$G =$ 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

The order n of G and the cofactor are:

$n =$ FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141

$h =$ 01

In other words the `secp256k1` curve is defined by the following:

$$y^2 = x^3 + 7 \text{ over } F_p \text{ or}$$
$$y^2 \text{ (mod p)} = x^3 + 7 \text{ (mod p)} \text{ [1]}$$

With $mod\, p$ (modulo prime number p) meaning the curve is over a finite field of prime order p, which is also written as F_p, where in the case of Bitcoin

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$

a very large prime number. Also note G is the base point/generator point and is very important in the proceeding sections.

The graph of this is very big, so in order to provide an example of what such a graph looks like, we will use a smaller finite field of prime order 17. As you can see in Figure 8 this graph is also symmetrical about x.

4.5 Key Generation

The public key is produced by the following elliptic curve multiplication:

$$K_{pub} = K_{priv} \times G$$

So we are trying to find the public key K_{pub} which is a point on the elliptic curve. We find this by multiplying the private key in the form of a randomly generated number K_{priv} by a point on the elliptic curve called the generator point G. The generator point is specified as part of the `secp256k1` standard and is always the same for all Bitcoin keys (see T in 4.4). Hence a private key multiplied with G will always result in the same public key.

It is quite straight forward to find K_{pub} from K_{priv} using the double and add method described in 4.3.4. However the reverse operation is very difficult to do, and this is why the public key can be shared without anyone deriving the initial private key.

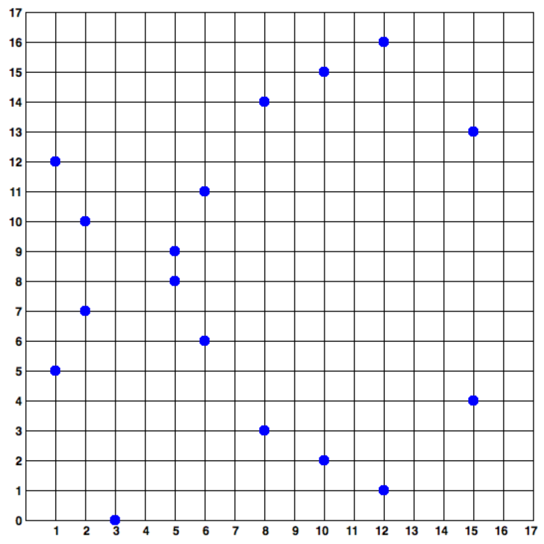


Figure 8: Elliptic curve over F_{17} [1]

In order to visualise this process we will use an elliptic curve over the real numbers, the mathematics of elliptic curves over finite fields and real numbers is the same. Our goal is to find the multiple $K_{priv}G$. As we saw in 4.3.4 point multiplication on an elliptic curve can be broken down into point addition and doubling. For example lets take $K_{priv} = 8$ and hence we are calculating $8G$. The resulting process is showed in Figure 9. Taking into consideration that K_{priv} is usually a very large number, we can see that it is very difficult to start with the public key (other users are not aware that this is $8G$) and generator point and deduce what K_{priv} is.

4.6 ECDSA

A brief outline of how digital signatures work was given in 2.4.2. Bitcoin uses the mathematics of elliptic curves as the underlying basis for its digital signature. Recall elliptic curves are defined by $T = (p, a, b, G, n, h)$, with Bitcoin using parameters prescribed by `sep256k1`. We also have the private and public key pair (K_{priv}, K_{pub}) where $K_{pub} = K_{priv} \times G$, as explained in 4.5. If Alice (A) and Bob (B) wanted to send a message (or transaction) to each other, this is how they would create and verify a digital signature.

4.6.1 Signature Generation [7]

To sign a message m Alice would do the following.

1. Select a random integer $k, 1 \leq k \leq n-1$.
2. Compute $kG = (x_1, y_1)$ and convert x_1 to an integer $\overline{x_1}$.
3. Compute $r = \overline{x_1} \pmod{n}$. If $r = 0$ then go to step 1.
4. Compute $k^{-1} \pmod{n}$. Where k^{-1} is the multiplicative inverse and satisfies $k^{-1} \cdot k \pmod{n} = 1$.

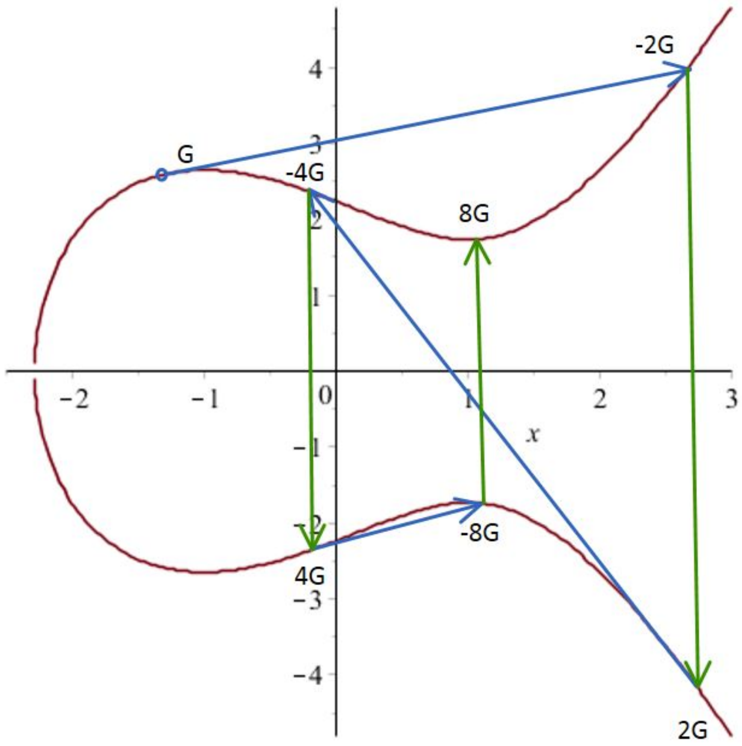


Figure 9: Visual representation of the multiplication of the generator point G by the private key $K_{priv} = 8$ on a elliptic [1]

5. Compute a SHA-1 hash of m, SHA-1(m) and convert this into an integer e.
6. Compute $s = k^{-1}(e + K_{priv} \cdot r) \pmod{n}$. If $s = 0$ then go back to step 1.
7. A's signature for the message m is (r, s).

4.6.2 Signature Verification [7]

Bob knows the parameters T since everyone on the Bitcoin network uses the same ones. He also knows Alice's public key K_{pub}. To verify A's signature (r, s) on m B does the following:

1. Verify that r and s are integers in the interval $[1, n-1]$.
2. Compute SHA-1(m) and convert this into an integer e.
3. Compute $w = s^{-1} \pmod{n}$.
4. Compute $u_1 = ew \pmod{n}$ and $u_2 = rw \pmod{n}$.
5. Compute $X = u_1 G + u_2 K_{pub}$.
6. If $X =$O then reject the signature. Otherwise, convert the x coordinate x_1 of X to an integer $\overline{x_1}$, and compute $v = \overline{x_1} \pmod{n}$.
7. Accept the signature iff $v = r$.

4.6.3 Proof [7]

If a signature (r, s) on a message m was indeed generated by Alice, then

$$s = k^{-1}(e + K_{priv} \cdot r) \pmod{n}.$$

Multiplying by $s^{-1}k$ gives

$$\begin{gathered} k = s^{-1}(e + K_{priv} \cdot r) \pmod{n}. \\ k = s^{-1}e + s^{-1}K_{priv} \cdot r \pmod{n}. \\ k = we + wK_{priv} \cdot r \pmod{n}. \\ k = u_1 + u_2K_{priv} \pmod{n}. \end{gathered}$$

In the verification step we calculate:

$$X = u_1G + u_2K_{pub}.$$

Using the relation between the private and public key, gives:

$$X = (u_1 + u_2K_{priv})G,$$

and using $k = u_1 + u_2K_{priv} \pmod{n}$ gives,

$$X = kG.$$

Which is the same as step 2 in 4.6.1 and so $v = r$ as required.

A References

References

[1] Antonopoulos. A. *Mastering Bitcoin - Unlocking Digital Cryptocurrencies.* O'Reilly Media, 2014.

[2] Bitcoin Foundation. *Some Bitcoin words you might hear.* [online]. N.d. [Last accessed February 25th 2015]. Available from: https://bitcoin.org/en/vocabulary.

[3] Certicom. *1.0 Introduction.* [online]. N.d. [Last accessed March 14th 2015]. Available from: https://www.certicom.com/10-introduction.

[4] Certicom Research. *STANDARDS FOR EFFICIENT CRYPTOGRAPHY, SEC 2: Recommended Elliptic Curve Domain Parameters v1.0* [online]. September 20 2000 [Last accessed April 9th 2015]. Available from: http://www.secg.org/SEC2-Ver-1.0.pdf

[5] CoinDesk. *How Bitcoin Mining Works.* [online]. December 22nd 2014. [Last accessed March 8th 2015]. Available from: http://www.coindesk.com/information/how-bitcoin-mining-works/.

[6] Dai. W. *b-money.* [online]. 1998. [Last accessed February 25th 2015]. Available from: http://www.weidai.com/bmoney.txt.

[7] Johnson.D, Menezes.A, Vanstone.S. Certicom Research Canada *The Elliptic Curve Digital Signature Algorithm (ECDSA).* N.d [Last accessed April 9th 2015].

[8] Koblitz. N. *A course in number theory and cryptography.* Second edition. New York: Springer-Verlag, 1948.

[9] Nakamoto. S. *Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System.* [online]. 2008. [Last accessed February 25th 2015]. Available from: https://bitcoin.org/bitcoin.pdf

[10] Simonite. T. *What Bitcoin Is, and Why It Matters.* [online]. May 25nd 2011. [Last accessed February 25th 2015]. Available from: http://www.technologyreview.com/news/424091/what-bitcoin-is-and-why-it-matters/.

[11] Wagner. A. *Digital vs. Virtual Currencies.* [online]. August 22nd 2014. [Last accessed February 25th 2015]. Available from: https://bitcoinmagazine.com/15862/digital-vs-virtual-currencies/.

[12] *The Royal Fork* [online]. N.d. [Last accessed April 8th 2015]. Available from: http://www.royalforkblog.com/

[13] *Bitcoin - Proof of Work* [online]. N.d. [Last accessed April 25th 2015]. Available from: https://en.bitcoin.it/wiki/Proof_of_work

[14] *Bitcoin definition investopedia* [online]. N.d. [Last accessed May 4th 2015]. Available from: http://www.investopedia.com/terms/b/bitcoin.asp

[15] *Field (mathematics)* [online]. N.d. [Last accessed May 4th 2015]. Available from: http://en.wikipedia.org/wiki/Field_(mathematics)

[16] *Bitcoin History: The Complete History of Bitcoin (Timeline).* [online]. N.d. [Last accessed February 25th 2015]. Available from: http://historyofbitcoin.org/.