An Analysis of the Bitcoin Electronic Cash System

Danielle Drainville
University of Waterloo

December 21, 2012

Abstract

In a world that relies heavily on technology, privacy is sought by many. Privacy, among other things, is especially desired when making an online payment. This motivates the use of electronic cash, a form of electronic payment system based on the paper cash system used daily. The most successful and widely used of these services is Bitcoin – a decentralized peer-to-peer electronic cash system. This paper provides a broad introduction to Bitcoin, while analyzing its construction and investigating some of its perks and flaws. It can be seen that, when compared to paper cash and electronic cash, Bitcoin is in a class of its own.

Contents

1 Introduction

Cryptography is an old science that is widely used for everyday tasks in this technological era. The basis of cryptography is to allow for hidden and secure communications. Nowadays, this is mostly used in relation with technological aspects. Many of its applications are based on simple concepts that provide a basis for the level of security required. Paper cash, used for modern day trade, serves as a model for the cryprographic application of electronic cash (for example, the debit and credit card systems). Electronic cash is primarily used when performing online payments, which are primarily done via credit card. Cryptographers have been attempting to design a secure form of electronic cash, based on the security properties found in the paper cash system. Some of these designs include David Chaum's DigiCash, which went bankrupt, and the Chaum-Fiat-Naor scheme, which makes use of the RSA blind signature scheme. Unfortunately, these electronic cash schemes, along with many others, do not have all the desirable properties that one can find in paper cash. Adding these desirable properties to an existing scheme generally come at the cost of other features.

In 2008, the mysterious Satoshi Nakamoto released a paper describing a decentralized peer-to-peer electronic cash system named Bitcoin. People were excited to have finally found a scheme that seemed to provide all the desirable properties of electronic cash. When Bitcoin was finally launched in 2009, it was enthusiastically received by early adopters. The cost incurred by the scheme to have the most desirable security features is that of no central authority or government figure. This paper will present Bitcoin in a clear manner and explain the behind-the-scene workings of the system. It also describes the cryptography supporting Bitcoin and the damages that would occur should the cryptography be broken. This paper will also describe two attacks on Bitcoin, as well as present some applications.

This paper is arranged as follows. Section 2 briefly presents paper cash and its essential security features, while Section 3 presents the concept of electronic cash, how it works, and the security features most commonly present. Section 4 introduces Bitcoin, goes over how the scheme works and which security features are obtained. The section also covers two attacks on the Bitcoin system. Section 5 compares the security features found in paper cash, electronic cash and Bitcoin. My own personal experience with using

Bitcoin is presented in Section 6. Section 7 covers Bitcoin applications. Section 8 covers related work, while Section 9 briefly goes over Bitcoin's future. Concluding remarks are made in Section 10.

2 Paper Cash

Paper cash is the most common form of currency. It is represented by bills and coins, which are backed by the government to assure their value and validity. For example, in Canada, bills come is denominations of $5, $10, $20, $50 and $100, while coins represent denominations under $1, as well as the $1 and $2 coins. These bills and coins are backed by the Bank of Canada, the country's federally appointed central bank. Also, new coins and bills are produced by the Bank of Canada, thus controlling the supply of money. The endorsement that paper cash receives from a nation's government allow users to trust in the validity of the currency.

The general population has a tendency to gravitate towards paper cash for various reasons. These reasons range from being able to better monitor a household's cash flow, to sheer convenience. Also appealing are the security properties found in paper cash, which, for the most part, are as follows.

Recognizability Paper cash is recognized as a valid and legal currency with government endorsement.

Portability Paper cash can be easily carried.

Transferability A user, after having received paper cash during a payment, can subsequently use that same money without having to go through a financial network.

Divisibility There is the ability to "make change."

Unforgeability Paper cash is difficult to duplicate. Mints are continuously thinking of ways to increase the level of difficulty required to duplicate paper cash.

Untraceability It is difficult to keep a record of where money is spent.

Anonymity There is no practical way to associate a bill or coin to a particular user. For example, when Alice withdraws money from her bank, deposits money or makes a payment, her identifying information is not written down alongside the serial number of the bill in question.

Security There is no way a user can spend a bill or coin multiple times. In other words, Alice cannot make a payment with the same $20 bill three times.

3 Electronic Cash

In the past few decades, more and more people have been turning to the Internet to facilitate certain tasks. One of these tasks is online purchases and the main method of payment is via credit card. Unfortunately, this does not offer the same security features as one would get using paper cash. For example, the bank knows where a user has spent their money, while a merchant knows the user's identity. This lack of security features offered through credit card payment led to the creation of electronic cash, or ecash, which is an electronic payment system based on the paper cash system. Ecash payments are similar to payments made using one's debit or credit card, but with additional security features.

3.1 How It Works

Electronic payments involve three different parties – the payer, the payee and a financial network.

Payer This is the individual who wishes to make a purchase, say Alice.

Payee This is the merchant from whom the payer wishes to make a purchase, say Bob.

Financial network This is where the payer and the payee store their funds and is more commonly referred to as the Bank.

This payment system can be performed either online or offline. In an online payment, the payee is in constant communication with the Bank who will verify the validity of a payment by ensuring that money is not being double spent, as well as deposit the money, before the payee issues the goods to the payer. On the other hand, in an offline payment, the payee will issue the goods and, at a later time, will deposit the received money to the Bank who will then verify its validity. Unfortunately, it is difficult to ensure that no users double-spend their coins. The fraud would be detected by the Bank, but there is no way of identifying the culprit. Since the Bank received payments after a transaction is complete, there is no way for the Bank to prevent the malicious user from double-spending coins.

This paper will focus primarily on the online ecash scheme using RSA blind signatures on withdrawal requests to allow for payer anonymity and payment untraceability [18, 19]. Since the Bank will be signing a requested withdrawal amount that has been blinded, it needs to make sure Alice is not committing fraud. One of the solutions is for the Bank to have a public key / private key pair for different denominations (eg. 5$, 10$, 20$, etc.).

Withdrawal Protocol

1. Alice prepares a message $M =$ (This is a \$100 bill, #12345), where #12345 is the requested coin's serial number.

2. Alice obtains the Bank's public key (n, e) for generating \$100 coins.

3. Alice selects $r \in_R \mathbb{Z}_n^*$.

4. Alice computes $m' = H(M)r^e \pmod{n}$, where H is the given cryptographic hash function.

5. Alice asks the Bank for a \$100 withdrawal and sends m'.

6. The Bank debits Alice's account by \$100 and sends Alice $s' = (m')^d \pmod{n}$, where d is the Bank's private key for \$100 coins.

7. Alice computes
$$\begin{aligned} s &= s'r^{-1} \\ &= (m')^d r^{-1} \\ &= (H(M)r^e)^d r^{-1} \\ &= H(M)^d (r^e)^d r^{-1} \\ &= H(M)^d r r^{-1} \\ &= H(M)^d \pmod{n}. \end{aligned}$$
   The coin is (M, s).

Note: A user's money is stored on a card when it is not in the Bank.

Payment and Deposit Protocol

1. Alice hands over the \$100 coin (M, s) to Bob.
2. Bob submits the coin to the Bank.
3. The Bank verifies the signature on the coin using its \$100 coin public key.
4. The Bank verifies that the coin has not been previously spent using the serial number.
5. The Bank enters the coin's serial number in a spent coin database.
6. The Bank credits Bob's account by \$100 and informs him that the payment is valid.
7. Bob finalizes the transaction with Alice.

3.2 Security Features

Recognizability Electronic coins are stored on cards when they are withdrawn from the Bank (eg. a laundry card), which the payee can easily recognize.

Portability Electronic coins are represented by a pair of relatively small integers (M, s). This allows coins to be easily stored on a card.

Transferability This is not offered, since the payee must redeem the coin at the Bank before his account can be credited. The scheme can be modified to allow transferability, however this comes at the expense of other desirable features.

Divisibility The protocol does not allow for divisibility. This could only be offered by creating a new transaction with the payer as the payee and the payee as the payer, or by forfeiting portability.

Unforgeability Since every coin is signed by the Bank, this protects against the forging of coins (assuming the security of the signature scheme).

Untraceability The Bank has no record of which coin Alice withdrew since it was blinded. Suppose the coin (M_2, s_2) is deposited some time after Alice withdrew her coin (M, s). The Bank has no way of determining whether the coin is the one Alice withdrew. To show this, let $r_2 = s's^{-1} \pmod{n}$. Note that $s_2^e = H(M_2) \pmod{n}$. We have

$$\begin{aligned} H(M_2)r^e &= H(M_2)(s's^{-1})^e \\ &= H(M_2)s'^e s^{-e} \\ &= H(M_2)(M'^d)^e s^{-e} \\ &= H(M_2)m's^{-e} \\ &= H(M_2)H(M)r^e H(M_2)^{-1} \\ &= H(M)r^e \\ &= m' \pmod{n}. \end{aligned}$$

Therefore, if Alice had picked r_2 as her blinding factor, the resulting coin would have been (M_2, s_2). Since blinding factors are picked at random in $\mathbb{Z}_n^*$, every incoming coin could have been the one Alice withdrew. Therefore, there is unconditional untraceability.

Anonymity Once again, due to the blinding factor Alice applies on a coin before withdrawing it from the Bank, the latter has no way of knowing who used which spent coin. This protocol therefore provides anonymity.

Security The scheme is secure against double-spending. This is because the Bank verifies the spent coin database before accepting a new coin for deposit. Unfortunately, in an offline scheme, the Bank can merely detect double-spending but not prevent it.

4 Bitcoin

Bitcoin is a decentralized peer-to-peer network. It was introduced on November 1, 2008 in a paper by the mysterious Satoshi Nakamoto [24], which is believed to be a pseudonym [16]. Unlike paper cash or electronic cash, Bitcoin does not rely on a central authority like the government or a bank. Instead, it relies on a proof-of-work system (more on this later) to verify and authenticate transactions, which are also made public for further verification. This new form of currency is also unique in that the number of coins in circulation will increase in a pre-determined way until the goal of 21 million coins in circulation is reached sometime in the year 2140 [14].

As mentioned, Bitcoin is a peer-to-peer based electronic cash system that does not make use of a central authority. In the Bitcoin network, each node represents one of potentially many public keys belonging to a Bitcoin user, and communicates directly with each other node. All the information is made public for every user to see. Also, decisions are made through a majority vote. In Bitcoin, “voting” is primarily done by working with previous transactions and blocks.

4.1 How It Works

4.1.1 Getting Started

An individual wishing to use Bitcoins needs to go through a few simple steps to get started, which are similar to obtaining a bank account. A new user starts by downloading a wallet from the official Bitcoin website. Once that is complete, the user has to wait for the block chain consisting of *all* previously verified transactions to download. Having the block chain and previous transactions allows a user to verify the validity of transactions for themselves and track the path made by coins. The process can take a few hours, but does not require any work by the user. Once the wallet and block chain are downloaded, users can generate as many public keys (also known as Bitcoin addresses) as they wish.

As with any other currency, having a wallet is not enough – funds are necessary. Methods for obtaining coins include:

- Bonus programs – these offer a small amount of coins for completing surveys, making purchases, etc..
- Bitcoin virtual exchanges – for example, Mt Gox [5] and Cavirtex [3]. In these, money can be traded for coins through methods like wire transfer and a form of online bill payment.
- Mining – This is the main method of obtaining Bitcoins, as well as how new coins are introduced in the system. It is done by verifying transactions (more on this later).

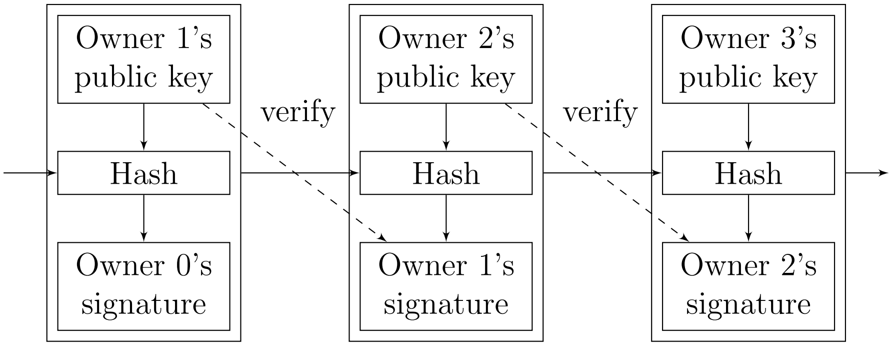

Figure 1: Transaction chain for a Bitcoin

4.1.2 Transactions

A Bitcoin (BTC) can be thought of as a chain of digital signatures. When sending a coin from one user to another, the previous transaction in which this coin was used is hashed together with the recipient's public key, to then be signed by the sender. This hash and signature are then added to the end of the coin chain. Since the sender's public key is included in the previous transaction for the coin in question, any user can use it to verify the validity of the subsequent signature; see Figure 1. As previously mentioned, transactions are publicly broadcasted for authentication and verification. It should be noted that, in Bitcoin, there is no such thing as "my" coin, "your" coin, or "same" coin, since all transactions are simply numbers.

A transaction can contain multiple inputs and multiple outputs. Consider the scenario where Alice received one Bitcoin from each of Bob and Charlie. Suppose she now wishes to send 1 BTC to Carol and 0.5 BTC to Oscar. The transaction in question will have the two coins she received from Bob and Charlie as two separate inputs. It will also have the 1 BTC to be sent to Carol, the 0.5 BTC to be sent to Oscar, and 0.5 BTC in change to be returned to Alice as outputs; see Figure 2. Every output will then add a new link to the transaction chain of the coin in question. It should be noted that a node in the network will not accept multiple transactions using the same inputs. Nodes will only accept the first one they receive and reject the subsequent transactions. See Appendix A for an example of a transaction in the transaction chain.

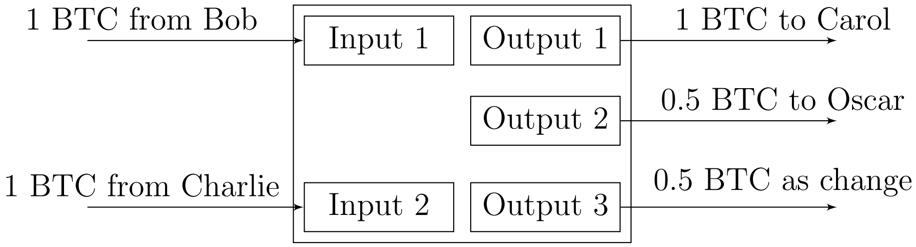

Figure 2: A transaction with multiple inputs and multiple outputs

To prevent a malicious user from double-spending a coin, some form of timestamping needs to be done. This leads to the proof-of-work (PoW) process, which uses a reward system to motivate users, as well as generate new coins.

4.1.3 Proof-of-Work

Proof-of-work is essentially taking the hash of a block of items and publishing this hash to the network. The items in question for the PoW block are transactions that need to be verified, the hash of the previous block, and a nonce. Since each block contains the hash of the previously generated block, the blocks form a chain of hash values as with transactions. The goal is to systematically increase the nonce so that the hash of the block that is currently being generated is less than a predetermined number given as targetted difficulty. This target is updated every 2016 blocks to ensure that the time it takes to generate a block is on average 10 minutes. This implies that the block cannot be altered without redoing all the work required to find a nonce giving a valid hash, as well as the work required to generate all the subsequent blocks in the PoW chain. Users will accept a block if all the transactions contained in it are valid and if the coins have not been previously spent. They will show their acceptance of this block by using the newly found hash in the "previous hash" section of the next block they attempt to generate, thus adding a new block to the chain. This chain is called the block chain; see Figure 3. In the figure, "Previous Hash" is the hash of the previous block and each "Tx" represents a transaction being verified. The transactions can be condensed together to save space using a Merkle hash tree [23]. There is no upper bound to the number of transactions that can be verified in a single block, but there has to be at least one.

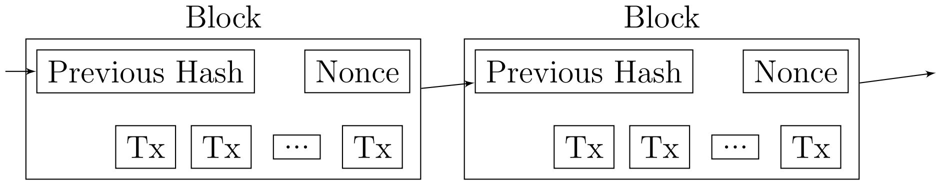

Figure 3: Proof-of-work and creation of the block chain

Bitcoin users can generate new blocks, or “mine”, using their computing power. The more computing power they possess, the greater the chance of being the first to win the race to block generation. As a reward for expending this power, successful users are rewarded with a predetermined amount of Bitcoins. This is also how coins are introduced to the system. The reward is set to decrease by half every 210 000 blocks. It starts at 50 BTCs, then will decrease to 25 BTCs, followed by 12.5 BTCs, and so on until the predetermined cap of 21 million BTCs are in circulation by the year 2140. As a matter of fact, the reward recently reduced to 25 BTCs on November 28, 2012. Certain transactions contain an incentive of a few BTCs that go to the user who generated the block verifying the transactions in question. As an added bonus for spending their computing power for mining, these incentives are added to the reward. Both the reward and the incentives are stored in the block implicating them, in what is called the coinbase. Once a block is generated, this creates a transaction from the coinbase to the successful miner. It should be noted that this is the only type of transaction that does not have a traditional input.

Since multiple users are attempting to generate blocks and obtain the reward, there is a possibility that two blocks are created around the same time thus creating a fork in the chain. It should be noted that users are not necessarily creating blocks verifying the same transactions. To remedy the fork, users will have a tendency to trust the prong with the highest level of difficulty, which usually happens to be the longest chain. The blocks that are not part of the longest chain are then dropped and the transactions they verified are put back in miners’ memory pool. This can be seen in Figure 4 where blue blocks form the main chain, while green blocks represent blocks that are dropped.

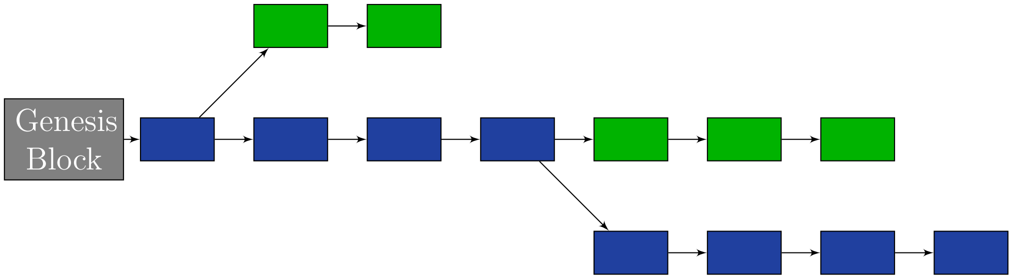

Figure 4: Forks in the block chain

The chain in its entirety stems from the "genesis block". This is the first block in the chain and was generated on January 3, 2009 by Satoshi Nakamoto. It contained the text "The Times 03/Jan/2009 Chancellor on Brink of Second Bailout for Banks" as timestamp [15]. The creator then received the 50 BTC reward, which cannot be spent by design. The hash of the genesis block is hardcored into the Bitcoin software.

Appendix B shows the genesis block (Block 0) comprising the first transaction (generation of the first 50 BTC), and the second block (Block 1) which also contains a single 50 BTC-generating transaction. Also shown in Appendix B is Block 100 000, generated on 29/Dec/2010, containing four transactions. Note that blocks can contain hundreds of transactions. For example, Block 190 000 contains 145 transactions, one of which is shown in Appendix A.

4.2 The Cryptography

To ensure the security and validity of transactions, certain cryptographic primitives are used. For instance, the hash function used for both transactions and block generation is SHA-256 [9]. Also, the signature algorithm used is the elliptic curve digital signature algorithm (ECDSA) [2, 10]. These are used to prevent a malicious user from breaking the system and gaining control of it. In this paper, the scheme is said to be broken if an attacker can impersonate other users by forging signatures or breaking the hash function and double-spending coins. The use of the hash function prevents malicious users from stealing and creating their own coins. This is due to the fact that they are protected by being hashed in a transaction, which is contained in a block, as well as with the added digital signature.

The way Bitcoin is designed implies that the last of the 21 million Bitcoins will be mined by the year 2140. This means that the cryptographic primitives used must remain secure until that time. Taking into consideration the growth of computer power over the past 40 years and the infamous Moore's Law, it is safe to assume that SHA-256 and ECDSA will be deemed insecure in that time frame. One of the more pressing concerns with cryptography nowadays is the pending arrival of large-scale quantum computers. Fortunately, the Bitcoin developers have mentioned a potential solution for this possibility.

4.2.1 SHA-256

As mentioned, the hash function H used for the Bitcoin system is SHA-256 [9]. To prevent a malicious user from breaking the scheme, the function must satisfy all three cryptographic security requirements for hash functions – preimage resistance, second preimage resistance, and collision resistance.

Preimage resistance Given a hash value $y \in \{0,1\}^n$, it is computationally infeasible to find with non-negligible probability of success any input x such that $H(x) = y$.

The hash function must be preimage resistant for one main reason. If this security property were not offered, the proof-of-work would not take an average of 10 minutes to compute. An attacker would possibly be able to modify a block containing one of her transactions and re-do the computations necessary to find a valid hash. She would also possibly be able to recompute the rest of the block chain in a feasible amount of time. This would allow the attacker to double-spend each of her coins as often as she wishes. With the lack of preimage resistance, multiple forks would be created in the block chain and there would be no money in circulation. Not only would malicious users be able to double-spend their coins, but Bitcoin would be rendered useless.

To fulfill the attack and double-spend his coins, a malicious user would act as follows if preimage resistance were absent.

1. After making a transaction, a malicious user, say Oscar, tracks down the transaction in the block chain.

2. Oscar changes the block containing his transaction by removing this transaction and updating the nonce to get the required hash value. Since Oscar knows the target hash values, this may be possible if the hash function does not achieve preimage resistance.

3. Oscar can now re-spend his coin and steal it once more.

For SHA-256, the fastest algorithm known at present for finding preimages takes 2^{256} steps.

Second preimage resistance Given a value x, it is computationally infeasible to find with non-negligible probability of success any input $x' \neq x$ such that $H(x) = H(x')$.

Second preimage resistance is necessary for several reasons. Recall that the hash of a transaction (resp. block) is included in the next transaction (resp. block) in the chain. Without second preimage resistance, a malicious user may be able to change the recipient of any transaction to an address in her control, while still satisfying the hash, and obtain the majority of the coins in circulation. If a malicious user were to make this change with her completed transactions, she would once again be able to double-spend her coins. In the case of a block, an attacker would be able to remove transactions and change the nonce to obtain a result in her benefit. In other words, she would be able to delete any of her transactions from a block and get her coin back, without having to re-do the work necessary to correct the block chain.

A malicious user would be able to mount the aforementioned attack for double-spending coins, in the case the hash function used were not second preimage resistant, in the following way.

1. Oscar, a malicious user, finds a transaction transfering a large amount of coins, that were never spent, to a single user. Being malicious, he wishes to be the owner of those coins.

2. If the hash function is not second preimage resistant, Oscar could modify the address of the true payee to one of his own that satisfies the target hash value.

3. Oscar can also change the block verifying the transaction in question by modifying the nonce to satisfy the target hash value. This is done to take into account the change in the transaction.

4. Oscar sees an increase of coins in his wallet.

As with preimage resistance, the fastest algorithm known for finding second preimages in SHA-256 at this time takes 2^{256} steps.

Collision resistance It is computationally infeasible to find with non-negligible probability of success any two distinct inputs x and x' such that $H(x) = H(x')$.

Finally, there is collision resistance. The goal is once again that of double-spending.

1. The malicious user, say Oscar, prepares two transactions. One is the valid transaction with the vendor, say Alice, as recipient, while the other is manipulated to have the same inputs as the valid transaction with a public key in his control as output. These two transactions should have the same hash.

2. Oscar sends Alice the valid transaction and waits for it to be accepted in the block chain.

3. Oscar switches the valid transaction in the block chain with the fraudulent transaction and gets his coin back. Since the hash value of the two transactions are the same, the transaction chain and block chain will still be correct and accepted by the network.

Currently, the fastest algorithm known for collision finding in SHA-256 is due to van Oorschot and Wiener and takes 2^{128} steps [30].

4.2.2 ECDSA

The signature algorithm used for validating transactions and confirming the identity of the payer is ECDSA [10]. The elliptic curve used is secp256k1 from the SEC2 standard [11]. This curve is a variation of the Koblitz curve

$$E : Y^2 = X^3 + 7$$

taken over a field $\mathbb{F}_p$ of prime order, where the prime

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

is 256 bits long. Note that $\#E(\mathbb{F}_p)$ is a 256-bit prime. Some advantages of ECDSA are that keys and signatures are smaller than with RSA and it tends to yield faster implementations [21]. As previously mentioned, an attacker's goal concerning the signature algorithm would be to attempt to forge another user's signature. If by any chance an attacker were successful, she would be able to "steal" coins. For example, consider the case where Eve is able to forge Alice's signatures. Eve could determine the transactions with Alice's known addresses as output, and then send all the coins in question to an address she controls, before her victim does, using the forged signature. This is made possible due to the fact that transactions are widely published to the network for all to see.

The fastest attack known on ECDSA is to solve the elliptic curve discrete logarithm problem in $E(\mathbb{F}_p)$: Given $P \in E(\mathbb{F}_p)$ and $Q \in E(\mathbb{F}_p)$, determine an integer c such that $Q = cP$. At present, the fastest known algorithm for this problem is Pollard's rho algorithm [25] and its parallelization due to van Oorschot and Wiener [30], which takes 2^{128} steps.

4.2.3 Quantum Computers

One of the looming threats to cryptography is quantum computers. It is known that the discrete logarithm problem can be efficiently solved on a quantum computer [26, 29]. However, at present it is not known whether large-scale quantum computers can be built. A countermeasure to the threat of quantum computers is to use post-quantum cryptography, but a problem arises when it comes to the existing block and transaction chains – these would still have the "weak" cryptography. Satoshi Nakamoto mentioned in one of his elusive message board posts that this could be easily resolved. The proposed solution is to freeze the block and transaction chains at a point where the Bitcoin community deems everything valid, to then restart with safe cryptography. This solution could also be used even if quantum computers do not see the light of day, but when the cryptography currently in use is broken.

4.3 Security Features

Recognizability Bitcoin is its own form of currency and payments are made directly from a user's wallet. Coins are represented by a long line of transactions, which can be found in the block chain. This implies the scheme is recognizable. The value associated to a coin is the value given to it by the Bitcoin community and virtual exchanges.

Portability To reduce the space requirement of the block chain, all transactions contained in a block can be compacted using a Merkle hash tree [23]. It is also possible for users to store a portion of their wallet on their smartphone by simply downloading an application. This makes Bitcoin portable.

Transferability The main feature of Bitcoin is that there is no central authority through which all transactions have to pass to be validated. Since there is no financial network, Bitcoin is transferable.

Divisibility As was previously mentioned in Section 4.1.2, Bitcoins are divisible.

Unforgeability Bitcoins are unforgeable by design. The first thing a user in the network does when including a transaction in a block is verify the coin's history, as well as verify that the input value is greater than or equal to the output value. If either of these tests fail, it is determined that the payer is trying to send money that does not exist and the transaction is canceled.

Untraceability When it comes to untraceability, it is not an intended feature of Bitcoin. By design, transactions are made widely public, thus implying that the path taken by coins can be traced from one address to another.

Anonymity Anonymity is not an intended security feature of Bitcoin. However, it is possible for a user to use their address as a pseudonym, use a different address per transaction and use mixers, among other things, to maintain some degree of anonymity (see Section 4.4.2).

Security Once a coin is spent, it is added to a block in the block chain. Therefore, if a malicious user were to double spend a coin, a miner would detect it. Also, as mentioned in Section 4.2, Bitcoin prevents double-spending as long as the hash function and the signature algorithm are not broken. It must be noted that the Bitcoin system is only secure against double-spending in slow payment situations (situations where payments are left for an average of 10 minutes to be verified and added to the block chain before a good is delivered to the payer), as opposed to fast payment situations (situations where a good is delivered immediately); this is discussed in more detail in Section 4.4.1.

Not only does Bitcoin offer most of these features, but it also prevents a malicious user from spending coins that do not belong to her. This can be determined by comparing the receiving address in the previous transaction in which the coin in question was involved with the given public key. Also, if the transaction and signature do not coincide, it can be determined that the transaction has been tampered with.

4.4 Attacks

As with any cryptographic protocol and application, research has been done to attempt to find potential weaknesses. Several papers were published discussing certain attacks on and vulnerabilities of Bitcoin. This paper will focus on two of these. The first is an attack on security. The paper mentions how malicious users can circumvent the system and double-spend their coins in fast payment situations. The other is an analysis on anonymity, a feature that was never a main Bitcoin goal, but grew to be so. The analysis discusses how Bitcoin users can potentially have their identities revealed through linking of public keys and relating them to outside information.

Aside from the aforementioned attacks, there is also the threat that an attacker has more computing power than the rest of the network. The attacker would thereby be able to modify blocks at his will and alter the subsequent block chain to make the modification valid. He would also be able to generate every block and obtain each and every reward. This means that the attacker would have access to all the Bitcoins currently in the network, leading to the downfall of the system. Some argue that it would be more profitable for the malicious user to play by the rules and legitimately obtain more coins than the rest of the network, making him the wealthiest Bitcoin user. The outcome of this attack is similar to what would happen if the hash function used in Bitcoin were broken.

4.4.1 Attack on Security

When it comes to making a payment with Bitcoins, the transaction time takes at least 10 minutes before a payment is confirmed and the merchandise is delivered. These can be thought of as “slow” payments. On the other hand, there are “fast” payments. This is done when a customer wishes to view a website or purchase fast food. The customer has no interest in waiting 10 minutes, or more, for his payment to get confirmed to simply access a website or obtain food. Bitcoin developers have suggested that vendors should accept these payments and deliver the merchandise once the funds are received, even though not yet confirmed, as long as it is for a small amount. Therefore, if the payer ends up being a malicious user, the loss is minimal.

In their paper *Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin* [22], Karame et al. discuss how, even with the developers’ solutions to prevent double-spending, it is still possible to successfully double-spend coins. The transaction features they exploit is the fact that transactions are broadcasted first to neighbour nodes, who then forward these to their neighbours upon receipt (and so on), as well as the fact that users will not accept two transactions with the same input, but different outputs.

To start, consider the attacker model. The attacker, say Oscar, is a malicious Bitcoin user who wishes to obtain some merchandise from a vendor, say Alice, without having to pay. Oscar has the ability to control some nodes in the network, for example nodes representing his own devices, but they do not have more computing power than the rest of the network combined. He also has no access to Alice’s devices or private keys. Furthermore, Oscar does not mine for blocks. Therefore, when a transaction is confirmed, Oscar cannot modify the block in question. As with every other user, Oscar’s addresses cannot identify him. This implies that the double-spending and the address used will only be detected, but Oscar remains anonymous and can easily generate a new address.

The attacker’s goal is simple – Oscar wishes to trick Alice into accepting a transaction and send the merchandise without waiting for confirmation. To do so, Oscar will create two transactions. One of these, Tx_A will be the valid transaction with Alice as recipient, while Tx_O will be the malicious transaction with Oscar, or one of his peers, as recipient. The idea is for Alice to receive Tx_A first, while the majority of the network receives Tx_O first. This means that Tx_O is more likely to be confirmed in the block chain. To simplify the rest of this section, let t_A^A and t_A^O denote the time at which Alice receives Tx_A and Tx_O respectively.

There are two conditions for the proposed attack to be successful. The first is that Alice must receive Tx_A before she receives Tx_O, i.e. $t_A^A < t_A^O$, or else Alice will add Tx_O to her memory pool and reject Tx_A. She would then ask Oscar to send the funds again. The second condition is that Tx_O must be accepted in the block chain, or else the attack fails and Oscar loses his coin. These two conditions are the main focus in the developing of the attack, which works as follows.

- Oscar connects to Alice as a neighbour in the network. This is possible since Alice's IP address is public and nodes always accept requests. Moreover, the number of neighbours a node can have is 125.

- Oscar has access to one or more helpers, which could originate from the same device. None of these helpers connect directly to Alice and are not an immediate neighbour.

- Oscar sends Tx_A to Alice at time t_A and Tx_O to the helpers at time t_O, where $t_O = t_A + \Delta t$. Alice and the helpers then proceed to send their received transaction to the rest of the network. Therefore, by construction, Alice will receive Tx_A before she receives Tx_O and $t_A^A < t_A^O$.

This satifies the first condition.

The two transactions will continue to be sent along the network until nodes have received Tx_A or Tx_O, or one of the two transactions gets accepted in a block. Oscar has a better chance of having Tx_O accepted in a block before Tx_A by increasing the number of its helpers. With more helpers, there is a greater chance that the majority of the nodes in the network receive the double-spent transaction before the valid transaction. Another method

Oscar could use to have Tx_O accepted before Tx_A is to send Tx_O first. The problem with this solution is that Tx_A cannot be delayed too long, at the risk of Alice receiving Tx_O first and asking for a re-payment. These two methods satisfy the second condition.

Experimental results show that the probability of Oscar being successful in his double-spending attack is significant. Also, his probability of success decreases as Δt increases, since more network nodes will receive Tx_A first, though this is remedied with an increase of helper nodes.

To better evaluate the probability of success when performing the double-spending method presented in their paper, Karame et al. ran tests using wallets under their control. The tests use the setup previously described where the attacker has one or more helper nodes using 10 nodes in the network situated around the world. Also, the attacker connects only to the vendor and creates two transactions Tx_V and Tx_A using the same coins. Tx_V is then sent to the attacker's neighbour via the bitcoin network and Tx_A is sent to the helper nodes via direct TCP connection with a delay $\Delta t \in \{-1, 0, 1, 2\}$ seconds. Upon reception of Tx_A, the helper nodes transmit this transaction to the network. The vender will accept the transaction if she receives Tx_V. Tests were run with the vendor being at 4 different locations and the attacker being in Europe, though the latter's location does not matter since he simply connects to the vendor. The vendor also has a varying number of connections, these being from 8, 40 and 125. The attacker on the other hand has access to either one or two helper nodes, each connected to at least 125 other nodes in the network. It can be seen that for an attacker situated in Asia Pacific with 8 or 125 connections, an attacker with 2 helper nodes and a time delay of 1 second, the probability of success approaches 100%. Even with a vendor from North America with 40 connections, an attacker with one helper node and a time delay of -1 seconds, the probability of success once again approaches 100%.

Karame et al. introduced several solutions to the double-spending problem in fast payments. The first of these is to implement a listening period of a few seconds where Alice would delay giving Oscar the purchased merchandise. This solution entails Alice to monitor every transaction she receives and check to see if any of them have the same input as Tx_A. The time it would take for Alice to receive both Tx_A an Tx_O is on average 3.354 seconds.

Unfortunately, there is a way Oscar can circumvent this. He could delay the transmission of Tx_O such that $\Delta t = t_A^O - t_A^A$ is greater than the listening period. As Δt increases, the probability that Alice's neighbours get Tx_A first increases. This implies that when these neighbour nodes later receive Tx_O, they will not accept it and therefore never forward it to Alice. Depending on the number of helper nodes at hand, Tx_O would still have a good chance of being received by the majority of the network and be accepted in a block before Tx_A. This would render the listening period ineffective. Through experimentation, it can be seen that when Alice has greater than 100 neighbours, the probability that she does not receive Tx_O reduces. This would therefore make the listening period effective. Unfortunately, Alice cannot always guarantee her number of connections.

Another potential solution is for Alice to insert observer nodes in her control in the network. In contrast to Oscar's helper nodes, these observers would relay every transaction they receive back to Alice. This would then allow Alice to hold a listening period and interfere with Oscar's plans. Once more, experiments were run and they show that Alice would need approximately 3 well-connected observers, which would come at a high cost.

One last solution would be to introduce alerts to the system. Whenever a node in the network detects a double-spending attempt, it would send an alert containing both Tx_O and Tx_A as proof. Alerts would not come at a cost to Alice, all the while preventing Oscar from successfully mounting his attack. Not only would this prevent a malicious user from double-spending in fast payment situations, it would be simple to implement in the existing Bitcoin system. In fact, there already exists an alert system that is set to send out a ping if Satoshi Nakamoto's address is used, though this is not implemented.

4.4.2 Attack on Anonymity

Bitcoin has become popular for allowing its users to remain anonymous while performing transactions, even though this is not a design feature. Users take anonymity for granted since they are represented by a public key in the network, and since they can create multiple public keys to allow for different transactions. This is exactly the feature that can allow them to be de-anonymized. The other Bitcoin features allowing users to be de-anonymized

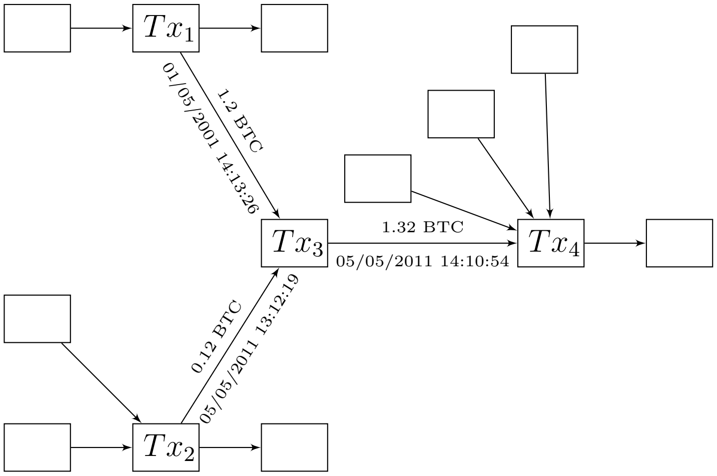

Figure 5: Transaction Network [27]

are the fact that the history is made public for all to see, the input/output relationship between transactions, and the re-use and co-use of public keys. This attack is presented in the paper *An Analysis of Anonymity in the Bitcoin System* by Reid and Harrigan [27].

The authors first introduce the idea of a transaction network and a user network. Both of these can be used to link the path taken by coins and associate different addresses to the same user. Both of these networks were also used to create flow diagrams. The creation of these diagrams was possible due to the highly public information found in the published transactions, which include source and target addresses as input and output values.

Transaction Network This network shows the flow of Bitcoins between transactions over time. In the flow diagrams created, nodes represent transactions and directed edges represent the output of the source transaction serving as input to the target transaction. A source transaction is a transaction's "previous output", while a target transaction is the transaction in progress. Also, each directed edge includes a value and timestamp; see Figure 5.

User Network This network shows the flow of Bitcoins between users over time. Each node represents a user's public key, while each directed

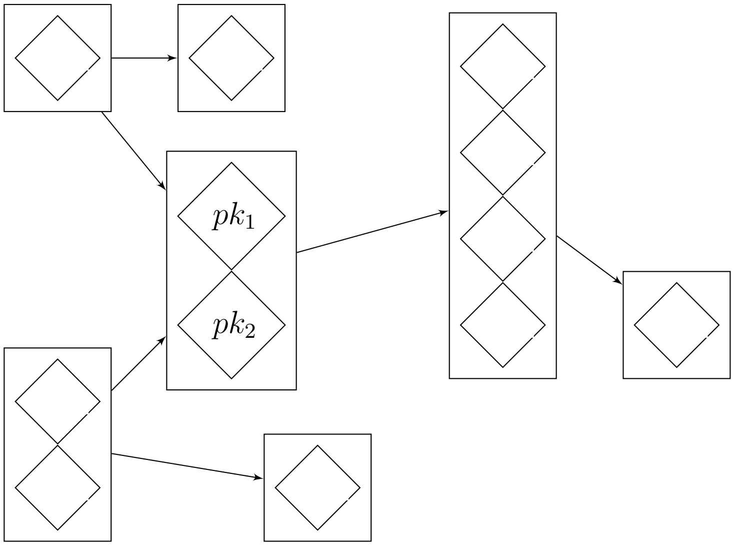

Figure 6: User network associated with previous transaction network [27]

edge represents the input-output pair of a transaction. The input, or payer's public key, is the source of the directed edge, while the output, or payee's public key, is the target of the directed edge; see Figure 6. Here nodes contained in the same box represent public keys owned by the same Bitcoin user.

When it comes to the user network, processing needs to be done. Some of the public keys can be linked when they are used in multi-input transactions, since the input keys necessarily belong to the same user. These public key nodes can be compressed into one node, designating a single user. The user network now becomes a compilation of public keys, with some of these condensed into a single node.

Once the linking of public keys is done, the anonymity factor comes into play. Some users can be associated with off-network information. Suppose that Alice purchased a physical item with her Bitcoins, which needs to be delivered to her physical address. The address could be obtained, thus linking one or more public keys to a previously unknown user. Alice could circumvent this by using a dropbox or an anonyous remailer, but this is beside the point. This scenario is possible due to the fact that certain vendors will ask and store some identifying information, like an email address, physical ad-

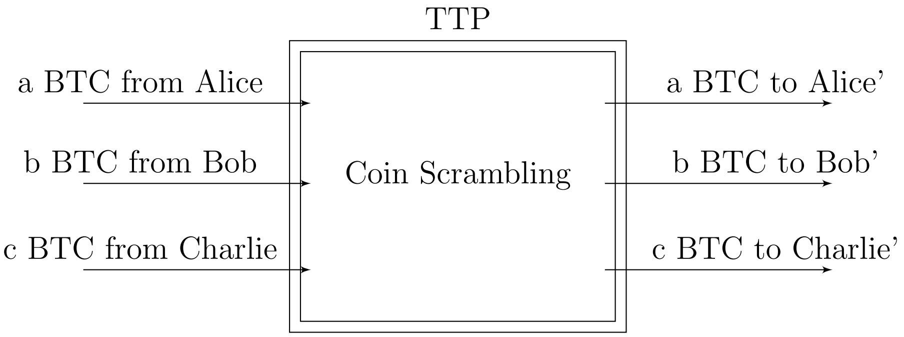

Figure 7: Mixer

dress or name. To avoid this linking, there are services offered, like mixers, to scramble the path taken by a coin.

For example, mixers are run by trusted third parties. They involve users, say Alice, Bob and Charlie, sending coins to a mixer. This mixer will then scramble these coins and send Alice, Bob and Charlie their original amount back, without sending them the original coins. Services, like Bitcoin Fog, receive a user's funds where they are pooled and mixed with other users' funds. The coins are then returned to the users when requested in the form of multiple random transactions sent over a span of time of the user's choosing [1]. On the other hand, Cleanbit uses clusters of wallets to mix coins. These clusters continuously send and receive coins from other wallets in the cluster. When funds are sent back to a user, the coins are taken from a random selection of clusters [4]. The mixing prevents the linking of one's identity to a public key in the case that a vendor's store of users' identifying information is leaked or taken by authorities; see Figure 7.

The all important question now is, why should this linking and identification matter? The network flows that were created would allow anyone to compute the balance held by a single user, making finding a target easy for an attacker. Certain people might also be linked to purchases they would rather keep private, in case it might affect their lives outside the privacy of Bitcoin.

Not only did Reid and Harrigan compile data and create flow diagrams linking users' public keys, they also implemented this in case studies. The reasoning behind this implementation is to show how a thief could potentially be deanonymized. The idea is to look at the user network before contractions and observe the vertices reachable by a path of at most 2 from the thief's public key. One of these was the case of a theft that occured back in the summer of 2011 where a user reported having 25 000 BTC stolen, which had a value of $50 000 USD at the time.

From observing the flow diagrams Reid and Harrigan created using the public information, they were able to determine one of the the thief's public keys. It was also noticed that the user's payout address on his Slush pool account (a pool mining group) was changed not long before the theft. From looking at the user network and using the above method, it was noticed that there was a path between the victim and the culprit other than the path created by the theft. There was also a 1 BTC theft shortly before the 25 000 BTC theft. By comparing the flow network around the theft with outside information, Reid and Harrington were able to identify some of the vertices. Some of these are from the user's main Slush pool account, while others are from a computer hacker group known as LulzSec. Unfortunately, LulzSec cannot be linked to the theft, but they did receive a 0.31337 BTC donation from the thief following the crime. The transactions around the theft show that 441.83 BTC were sent to the victim over a 70 day period from his Slush pool account, while 0.2 BTC were sent to an unknown user from the same Slush pool account. This unknown user also made a donation to LulzSec. All that can be inferred about this unknown user is that he is the owner of at least 5 public keys, is a member of the Slush pool, is a one-time donator to LulzSec, and the donation was the user's last known activity.

The case study presented shows exactly how much information can be determined using the method presented in the paper by Reid and Harrigan. Even though the culprit behind the theft is not identified, there is still a possibility of following the money quite far and linking it with outside information. With more time analyzing the network, the thief might one day be revealed.

Security Feature	**Paper Cash**	**Electronic Cash**	**Bitcoin**
Recognizability	✓	✓	✓
Portability	✓	✓	✓
Transferability	✓	X	✓
Divisibility	✓	X	✓
Unforgeability	✓	✓	✓
Untraceability	✓	✓	X
Anonymity	✓	✓	∼
Security	✓	✓ online X offline	✓ slow payment X fast payment

Table 1: Comparison between schemes

5 Comparison

Table 1 shows the security features offered by paper cash, the online ecash protocol, and Bitcoin. The features in question are extracted from paper cash and are the inspiration behind the design attempts for a successful version of electronic cash. As was presented in Section 3.2, it can be seen that most versions of ecash do not offer transferability or divisibility. If these are sought, other more practical features would have to be sacrificed. As for security and double-spending, it is prevented in the online protocol, but not in the offline protocol. The latter instead detects double-spending, but does not prevent it. Efforts are instead put into de-anonymizing the culprit through various means outside the scope of this paper.

When it comes to Bitcoin on the other hand, it is rather successful with most of the features as seen in Section 4.3. Unfortunately, due to the public nature of the scheme, untraceability is impossible to achieve. As for anonymity, it is possible to achieve if a user is cautious (see Section 4.4.2), but it is not one of Bitcoin's design goals. The only other issue with Bitcoin is when it comes to security. This is offered in slow payment situations, but Section 4.4.1 demonstrates how a malicious user could successfully double-spend in fast transaction situations. It can therefore be seen that Bitcoin is currently the form of electronic cash payment that best resembles paper cash. This comes at the cost of having no central authority and no government to back it up. Whether this cost is an advantage or a disadvantage is left up to debate.

6 Personal Experience

To further grasp Bitcoin and its ease of use, I downloaded my own wallet and investigated what this system has to offer. The wallet download was rather straightforward (I used the wallet found at We Use Coins [8]), though the block chain took several hours to synchronize. It is possible to encrypt one's wallet, and there are recommendations as to what is deemed a "secure" password.

Once I obtained my wallet, the next task was to get coins. There are a few ways to obain free coins, like Bitcoin Faucet [12], and multiple online exchanges allowing a user to purchase coins. Unfortunately, credit card payments and PayPal are not accepted as a form of payment due to the fact that one can cancel a transaction after the product has been obtained calling it fraud or claiming to never have received the product. The other hurdle was that most exchanges only dealt with Europe or USA, while others did not recognize my small Canadian bank as a valid financial institution. Luckily, there is a Canadian virtual exchange that allows users to make a direct deposit from their bank account as if they are paying a bill.

Cavirtex [3], the Canadian virtual exchange in question, was therefore my way in. The only thing a user needs to do is get their bank account verified, which means uploading pictures of various things like proof of address and a valid government issued identification card, as well as giving them your banking information. This meant trusting the exchange, one of the problems when it comes to Bitcoin due to the lack of government figure or central authority. Afterwards, things flowed well with only a few days wait before getting funds on my account and an instant conversion of these to Bitcoins. Transactions are then straightforward and seemingly wait-free.

The other method of obtaining coins I used was through pool mining. This requires a potential miner to download a small application, register with the pool, and leave their computer turned on. Funds received are then put in a wallet run by the mining pool to allow for the user to transfer them to the wallet on their machine. The reward obtained is rather small due to the size of the pools, though there is a semi-constant flow of incoming coins, as opposed to a month wait before one is lucky enough to be the first to mine a block and obtain the 50 BTC reward.

All in all, Bitcoin is simple to use and rather hassle-free. The only problem I encountered was obtaining funds, which took a week or so once my account was verified with the exchange. Making Bitcoin purchases are as easy as making a credit card payment, if not easier. In my experience, this is a user-friendly scheme.

7 Applications

This section presents two Bitcoin applications. The first of these is an application using the Bitcoin scheme called CommitCoin. It was introduced in the paper *CommitCoin: Carbon Dating Commitments with Bitcoin* by Clark and Essex in 2012 [20]. The second is Silk Road [7], an online market place resembling Ebay that only accepts Bitcoins as payment [17].

7.1 CommitCoin

The idea behind CommitCoin is to add a timestamp to a committed message. Consider the situation where Alice makes a wonderful discovery, but wishes to hold onto it for a while to make corrections and wait for the appropriate time to made the discovery public. During this period, Bob could potentially independently make the same discovery and publish it. To prevent Bob taking her credit, Alice could "commit" her discovery and add a timestamp. This is somewhat like Alice putting her discovery in a sealed envelope and sending it to herself, thus adding a timestamp from the postal service. Subsequently, if Bob were to make the same discovery, Alice could produce the sealed envelope and prove she was indeed the first to have made the discovery.

Clark and Essex suggest that Bitcoin can be used for carbon dating in such situations. Users like Alice would leave their commitment value, a number representing a mixture of the message being committed and some randomness, in the Bitcoin history without harming the system. This can be done by putting the commitment value in a transaction. The simplest solution would be to let the receiver's public key be the commitment value in a

1 BTC transaction. Unfortunately, this coin would be unrecoverable and be taken out of circulation. Therefore, another solution needs to be introduced – one that does not take coins out of circulation, all the while allowing the commitment value to be made public and inserted into the Bitcoin transaction history. The proposed method sets the commitment value to be the private key of a new account, which is possible since a user can generate their own private keys. The new private key is used to sign two messages using the same randomness, thereby allowing the private key/commitment value to be recovered.[1] The method works as follows:

1. Alice commits her discovery with some randomness by computing the commitment value (a function on the message and the randomness).

2. Alice generates a new account with the commitment value as private key.

3. Alice makes a transaction sending 2 BTC from herself to the new account and signs it. The transaction and signature are now in the Bitcoin network.

4. Alice makes a second transaction sending 1 BTC from the new account back to herself and signs it. The transaction and signature are now in the Bitcoin network.

5. When subsequently challenged, Alice makes a third transaction sending 1 BTC from the new account to herself and signs it with the same randomness used in the previous signature. This transaction and signature are now in the Bitcoin network.

6. Bob can now obtain the commitment value from the previous two signature, both published to the Bitcoin network.

Bob can now verify that Alice was indeed the first to make the discovery in question. He can also trust the approximate time at which it was made, knowing that there is no feasible way Alice could have faked the block chain. The fact that 2 BTC were sent from Alice to the new account and 2 BTC sent back from the new account to Alice, means that no coins are taken out

[1] ECDSA has the property that if two messages are signed using the same per-message random numbers, then the user's private key can be easily determined from the two signed messages.

of circulation.

The authors of CommitCoin propose to use their protocol for pre-election commitments. They implemented their protocol with Scantegrity, “an open source election verification technology for optical scan voting systems” [6], in the 2011 Takoma Park, MD, municipal election. The use of CommitCoin in the election was to provide carbon dating for pre-election commitments.

7.2 Silk Road

Silk Road [7] was launched in February of 2011 and is run by a user known to others as “Dread Pirate Roberts”. It is a site accepting payments made solely in Bitcoins, that primarily sells illicit goods. Fortunately, items intended to harm others, such as child pornography and credit card skiming devices are banned from being sold on Silk Road. For some time, firearms were sold on a sister site by the name of The Armory, which is now shut down. Though mostly known for selling illicit goods, it must be noted that Silk Road also sells non-illicit products.

There are many aspects to Silk Road that draws users. Firstly, to access the site a user must use the anonymous network Tor, as well as create an account. Secondly, the site makes use of an escrow system. When a user, say Alice, makes a purchase from a merchant, say Bob, the payment is first sent to an escrow. Once Alice receives her order, she confirms this with the escrow who then releases the payment and sends it to Bob. This prevents a malicious user from receiving payments without sending any goods, which has been a problem in the past. Some users, for speed and convenience, choose to avoid the escrow system, though this is done at their own risk. Thirdly, Silk Road makes use of user wallets that mix every incoming and outgoing payments. This service, combined with the use of Tor, allow for untraceability in the financial trail, as well as untraceability in buyer and seller communication. To this date, there have been no Silk Road related arrests.

The introduction and popularization of Silk Road has lead to a rise in popularity for Bitcoin. There are mixed opinions as to whether this service tarnishes Bitcoin’s reputation by relating it to the selling of illicit goods, even though Silk Road is Bitcoin’s largest e-commerce platform. On the other hand, Silk Road is not taking over the Bitcoin econonmy.

8 Related Work

A recent paper by Dorit Ron and Adi Shamir, entitled *Quantitative Analysis of the Full Bitcoin Transaction Graph* [28] utilises the transaction network introduced by Reid and Harrigan in [27]. In their paper, Ron and Shamir analyse the transaction network and observe different user behaviours regarding their funds. These behaviours include how users usually acquire and spend their coins and how users store their coins – whether it be in a wallet on their personal computers or in an online wallet. Ron and Shamir followed the paths Bitcoins take when they are moved around between accounts to protect a user's privacy. They also picked out the largest transactions and followed the paths taken by these coins. They were able to observe that the majority of the coins in the network are not in circulation, whereas the largest transactions are almost all linked to a single large transaction which the user in question seems to have attempted to hide.

Another recent paper *Evaluating User Privacy in Bitcoin* by Androulaki et al. [13] analyses the effects of having the transactions made public in the case where Bitcoin is the primary source of currency. The authors argue that this would affect a user's privacy. To support their claim, Androulaki et al. both analysed the Bitcoin network and ran simulations through a Bitcoin simulator. The simulations were run where Bitcoin is the only form of currency used within a university campus. From user behaviour and patterns, and linking this with the network information, profiles of almost 40% of users were uncovered. This is due to the fact that certain categories of users will have different spending habits than others (eg. professors vs. students in a university setting). It should be noted that the user profiles that were uncovered included users that applied the Bitcoin privacy recommemdations.

9 Bitcoin's Future

Bitcoin is a seemingly flawless electronic cash scheme. The reward serving as incentive is scheduled to halve from 50 BTC to 25 BTC in December 2012. This could potentially harm Bitcoin since there will be less of an incentive for users to spend their computing power on mining. Another concern is the fluctuation in the price of a Bitcoin. Many things have affected the conversion rate ranging from increase in popularity to theft of coins in virtual exchanges and wallet services. Bitcoin has seen a considerable amount of fluctuation in its short lifetime with the largest jump happening from June to November of 2011 where the exchange rate went from $1 USD to $30 USD and then back down to $2 USD. As of November 29 2012, the value of a Bitcoin was $12.56 USD. Since the reward will be lowering, there might be a lack of incentive for miners. This would then cause a lowering of the price of a coin. The risk is not very large since 25 BTC is still a considerable amount of coins for a user, but this will be extremely low when split among the peers of a peer mining group. The individuals with lower mining capacity might lose interest in Bitcoin. Their only other alternatives to obtaining coins would be through daily rewards, which are relatively low, or by purchasing coins with their own money. Users who use Bitcoin for "fun" might lose that fun aspect.

Another problem in Bitcoin's future is the cryptography. By the year 2140, it is highly probable that 256-bit ECDSA and SHA-256 will be broken. The developers will then have to do a system upgrade to keep up to date with the current cryptography. While doing this, they could also implement the alert system for fast payments mentioned by Karame et al. and a listening period to reduce the risk of malicious users attempting to double-cross their peers.

10 Conclusion

This paper has presented a broad overview of possibly the most successful version of electronic cash to date – Bitcoin. It explains how Bitcoin works in a detailed way, while going over the key concepts of the scheme. The paper also describes the cryptographic primitives used in the scheme. It considers what would happen if these applications were absent or broken, or if large-scale quantum computers became a reality – the consequences would be devastating. Fortunately, the developers have thought of potential solutions if such a thing were to happen.

Two separate attack papers are also presented in this Bitcoin overview. One of these is by Karame et al. [22] on double-spending in fast payment situations. This could be easily resolved if an alert system were implemented. The other paper is by Reid and Harrigan [27] and presents an analysis on anonymity. Although anonimity is not a main Bitcoin features, certain ways to increase user anonymity are presented like that of mixing services.

Since Bitcoin is a form of electronic cash scheme that is based on paper cash, this paper presents a comparison between these three concepts. Through this, it can be seen that Bitcoin satisfies all the security features associated with paper cash, other than untraceability and anonymity, which are not design features, and security in fast payment situations. On the other hand, the general form of electronic cash in an online payment situation is not as successful. It can be seen that transferability and divisibility are not achieved, as well as security in offline payment situations. To implement these features would require dropping other features, while in Bitcoin to add the missing features would be relatively simple, other than untraceability.

To conclude, this paper also presents my personal experience using the scheme, as well as Bitcoin applications like CommitCoin and SilkRoad. Potential risks to the future of Bitcoin are also presented, though only time will tell the ultimate success of Bitcoin.

References

[1] Bitcoin Fog: http://www.bitcoinfog.com/.

[2] Bitcoin Wiki: https://bitcoin.it/.

[3] Canadian Virtual Exchange CaVirtex: https://www.cavirtex.com/.

[4] Cleanbit: http://www.cleanbit.org/.

[5] Mt. Gox: https://mtgox.com/.

[6] Scantegrity: http://www.scantegrity.org/.

[7] Silk Road: http://silkroadvb5piz3r.onion.

[8] We Use Coins: www.weusecoins.com/.

[9] FIPS 180-3. Secure Hash Standard, Federal Information Processing Standards Publication 180-3. National Institute of Standards and Technology, 2008.

[10] FIPS 186-3. Digital Signature Standard. Federal Information Processing Standards Publication 186-3. National Institute of Standards and Technology, 2009.

[11] Standards For Efficient Cryptography Group. SEC 2: Recommended Elliptic Curve Domain Parameters. p.15, September 2000.

[12] Andresen, G. Bitcoin Faucet: https://freebitcoins.appspot.com/.

[13] Androulaki, E., Karame, G. O., Roeschlin, M., Scherer, T., Capkun, S. Evaluating User Privacy in Bitcoin. *IACR Cryptology ePrint Archive*, 2012:596, 2012.

[14] Barber, S., Boyen, X., Shi, E., and Uzun, E. Bitter to Better - How to Make Bitcoin a Better Currency. 16th International Conference on Financial Cryptography and Data Security, Lecture Notes in Computer Science, 7397:399 – 414, 2012.

[15] Buterin, V. Being Satoshi: A Look Inside the Man Behind the Currency. *Bitcoin Magazine*, 1:28–31, May 2012.

[16] Buterin, V. Bitcoin: Prehistory, Predecessors and Genesis. *Bitcoin Magazine*, 1:14–18, May 2012.

[17] Buterin, V. The Silk Road Report: http://bitcoinmagaine.net/the-silk-road-report/, July 2012.

[18] Chaum, D. Blind Signatures for Untraceable Payments. Advances in Cryptology – Crypto '82, 199-203, 1983.

[19] Chaum, D., Fiat, A., and Naor, M. Untraceable Electronic Cash. Advances in Cryptology – Crypto '88, Lecture Notes in Computer Science, 403:319–327, 1990.

[20] Clark, J., and Essex, A. Commitcoin: Carbon Dating Commitments with Bitcoin. *IACR Cryptology ePrint Archive*, 2011:677, 2012.

[21] Hankerson, D., Menezes, A., and Vanstone, S. Guide to Elliptic Curve Cryptography. Springer, 2003.

[22] Karame, G. O., Androulaki, E., and Capkun, S. Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin. *IACR Cryptology ePrint Archive*, 2012:248, 2012.

[23] Merkle, R. Protocols for Public Key Cryptosystems. Proceedings of the 1980 IEEE Symposium on Security and Privacy, 122–134, 1980.

[24] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Unpublished, November 1, 2008. www.bitcoin.org.

[25] Pollard, J. Monte Carlo Methods for Index Computation mod p. Mathematics of Computation, 32:918–924, 1978.

[26] Proos, J., and Zalka, C. Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves. Quantum Information and Computation, 3:317–344, 2003.

[27] Reid, F., and Harrigan, M. An Analysis of Anonymity in the Bitcoin System. 2011 International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing, 2011.

[28] Ron, D., Shamir, A. Quantitative Analysis of the Full Bitcoin Transaction Graph. *IACR Cryptology ePrint Archive*, 2012:584, 2012.

[29] Shor, P. Polynomial-Time Algorithm for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing, 26:1484–1509, 1997.

[30] Van Oorschot, P., and Wiener, M. Parallel Collision Search with Cryptanalytic Applications. Journal of Cryptology, 12:1–28, 1999.

A An Example of a Transaction

Hash: 8400bd1e9936f859e0f10bbc3b1353fee3bd194c754eca95ce906febe5e0d825
Appeared in block 190 000 (2012-07-20 22:53:36)
Number of inputs: 2
Total BTC in: 60.48808432
Number of outputs: 2
Total BTC out: 60.48798432
Size: 439 bytes
Fee: 0.0001

Inputs

Previous output	Amount	From address	Type	ScriptSig
c5c2e5f5b4dc...: 1	6.16536895	1HQs9u1H4R...	Address	3045022100...
0803dc7f52fc...: 1	54.32271537	1qw7ETyRrn...	Address	3046022100...

Outputs

Index	Redeemed at input	Amount	To address	Type	ScriptPubKey
0	8da89c74b9de...	6.9999	15GvJVi9mh...	Address	2ee33c8d21...
1	3ed27d2f4229...	53.48808432	1Mj2sZUnuY...	Address	e3558c6af6...

Table 2: A transaction from Block 190 000
(http://blockexplorer.com/t/618oDrswXu)

- **Hash**: The SHA-256 hash of the full transaction.

- **Appeared in block**: The number of the block that contains this transaction.

- **Fee**: The BTC amount claimed by the entity that generated block 190000. It is the difference between the total BTC in and the total BTC out.

- **Address**: A bitcoin address is a human-readable string of numbers and letters (in a customized base-58 encoding) around 33 characters in length, always beginning with the digit 1 or 3. The address is the RIPEMD-160 hash of an ECDSA public key. An example of a bitcoin address is 1HQs9u1H4RU4go8LAAhtkR1vVjAbUyeGHv. The sender of the transaction owns both the addresses in the column "From address". The two address in the column "To address" are the addresses of the two recipients of the BTC outputs of this transaction.

- **Previous output**: The truncated hash of a previous transaction and the index (after the colon) of the output that this input is redeeming; the first output in a transaction has an index of 0. For example, `c5c2e5f5b4dc...:1` refers to the second output of the transaction with hash value `c5c2e5f5b4dc4bddc3e2f91a3f5d6a7a530c878c2fb3ef0aa0291effef30f991` (see http://blockexplorer.com/t/8VfiZzmkoc).

- **ScriptSig and ScriptPubKey**: Bitcion uses a scripting system for transactions. A script is a list of instructions recorded with each transaction that describes how the recipient of a transaction can access the coins. Scripting provides the flexibility to specify what is needed to spend a received Bitcoin; for example, the script could require the use of two ECDSA private keys. **ScriptSig** contains a signature by the sender. **ScriptPubKey** usually contains the RIPEMD-160 hash of the recipient's ECDSA public key.

B Block Chain

Hash: `4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b`
Appeared in block 0 (2009-01-03 18:15:05)
Number of inputs: 1
Total BTC in: 50
Number of outputs: 1
Total BTC out: 50
Size: 204 bytes
Fee: 0

Inputs

Previous output	**Amount**	**From address**	**Type**	**ScriptSig**
N/A	50 + fees	N/A	Generation	`04ffff001d...`

Outputs

Index	**Redeemed at input**	**Amount**	**To address**	**Type**	**ScriptPubKey**
0	Not yet redeemed	50	`1A1zP1eP5Q...`	Pubkey	`04678afdb0...`

Table 3: Transaction from block 0
(http://blockexplorer.com/t/3pTRm5YNJz)

Hash: `000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f`

Next block: `00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048`

Time: 2009-01-03 18:15:05

Difficulty: 1

Transactions: 1

Total BTC: 50

Size: 285 bytes

Merkle root: `4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b`

Nonce: 2083236893

Transaction	**Fee**	**Size (kB)**	**From (amount)**	**To (amount)**
`4a5e1e4baa...`	0	0.204	Generation: 50 + 0 total fees	`1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa`: 50

Table 4: Block 0 (from http://blockexplorer.com/b/0)

<table>
<tr><td colspan="5"><b>Hash</b>: 00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048
<b>Previous block</b>: 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
<b>Next block</b>: 000000006a625f06636b8bb6ac7b960a8d03705d1ace08b1a19da3fdcc99ddbd
<b>Time</b>: 2009-01-09 -2:54:25
<b>Difficulty</b>: 1
<b>Transactions</b>: 1
<b>Total BTC</b>: 50
<b>Size</b>: 215 bytes
<b>Merkle root</b>: 0e3e2357e806b6cdb1f70b54c3a3a17b6714ee1f0e68bebb44a74b1efd512098
<b>Nonce</b>: 2573394689</td></tr>
<tr><th>Transaction</th><th>Fee</th><th>Size (kB)</th><th>From (amount)</th><th>To (amount)</th></tr>
<tr><td>0e3e2357e8...</td><td>0</td><td>0.134</td><td>Generation: 50 + 0 total fees</td><td>12c6DSiU4Rq3P4ZxziKxzrL5LmMBrzjrJX: 50</td></tr>
</table>

Table 5: Block 1 (from http://blockexplorer.com/b/1)

Hash: `000000000003ba27aa200b1cecaad478d2b00432346c3f1f3986da1afd33e506`
Previous block: `000000000002d01c1fccc21636b607dfd930d31d01c3a62104612a1719011250f`
Next block: `00000000000080b66c911bd5ba14a74260057311eaeb1982802f7010f1a9f090`
Time: 2010-12-29 11:57:43
Difficulty: 14 484.162361
Transactions: 4
Total BTC: 103.01
Size: 957 bytes
Merkle root: `f3e94742aca4b5ef85488dc37c06c3282295ffec960994b2c0d5ac2a25a95766`
Nonce: 274148111

Transaction	**Fee**	**Size (kB)**	**From (amount)**	**To (amount)**
`8c14f0db3d...`	0	0.135	Generation: 50 + 0 total fees	`1HWqMzw1jfpXb3xyuUZ4uWXY4tq...`: 50
`fff2525b89...`	0	0.259	`1BNwxHGaFbeUBitpjy2AsKpJ29Y...`: 50	`1JqDybm2nWTENrHvMyafbSXXtTk...`: 5.56 `1EYTGtG4LnFfiMvjJdsU7GMGCQv...`: 44.44
`6359f08681...`	0	0.257	`15vScfMHNrXN4QvWe54q5hwfVoY...`: 3	`1H8ANdafjpqYntniT3Ddxh4xPBM...`: 0.01 `1Am9UTGfdnxabvcywYG2hvzr6qK...`: 2.99
`e9a66845e0...`	0	0.225	`1JxDJCyWNakZ5kECKdCU9Zka6mh...`: 0.01	`16FuTPaeRSPVxxCnwQmdyx2PQWx...`: 0.01

Table 6: Block 100 000 (from http://blockexplorer.com/b/100000)