Handout 8 (Bitcoins)

In my opinion Bitcoins are an elaborate Ponzi scheme[1]—still the ideas behind them are really beautiful and not too difficult to understand. Since many colourful claims about Bitcoins float around in the mainstream and not-so-mainstream media, it will be instructive to re-examine such claims from a more technically informed vantage point. For example, it is often claimed that Bitcoins are anonymous and free from any potential government meddling. It turns out that the first claim ignores a lot of research in de-anonymising social networks, and the second underestimates the persuasive means a government has at its disposal.

There are a lot of articles, blogposts, research papers etc. available about Bitcoins. Below I will follow closely the very readable explanations from

http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/ and

http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html

The latter also contains a link to a nice youtube video about the technical details behind Bitcoins. I will also use some of their pictures.

Let us start with the question who invented Bitcoins? You could not make up the answer, but we actually do not know who the inventor is. All we know is that the first paper

https://bitcoin.org/bitcoin.pdf

is signed by Satoshi Nakamoto, which however is likely only a pen name. There is a lot of speculation who could be the inventor, or inventors, but we simply do not know. This part of Bitcoins is definitely anonymous so far. The paper above is from the end of 2008; the first Bitcoin transaction was made in January 2009. The rules in Bitcoin are set up so that there will only ever be 21 Million Bitcoins with the maximum reached around the year 2140. Currently there are already 11 Million Bitcoins in 'existence'. Contrast this with traditional fiat currencies where money can be printed almost at will. The smallest unit of a Bitcoin is called a Satoshi, which is the 10^{-8}th part of a Bitcoin. Remember a Penny is the 10^{-2}th part of a Pound.

The two main cryptographic building blocks of Bitcoins are cryptographic hashing functions (SHA-256) and public-private keys using the elliptic-curve encryption scheme for digital signatures. Hashes are used to generate 'fingerprints' of data that ensure integrity (absence of tampering). Public-private keys are used for signatures. For example sending a message, say *msg*, together with the encrypted version

[1] http://en.wikipedia.org/wiki/Ponzi_scheme

$$msg, \{msg\}_{K^{priv}}$$

allows everybody with access to the corresponding public key K^{pub} to verify that the message came from the person who knew the private key. Signatures are used in Bitcoins for verifying the addresses where the Bitcoins are sent from. Addresses in Bitcoins are essentially the public keys. There are 2^{160} possible addresses, which is such a vast amount that there is not even a check for duplicates, or already used addresses. If you start with a random number to generate a public-private key pair it is very unlikely that you step on somebody else's shoes. Compare this with the email-addresses you wanted to register with, say Gmail, but which are always already taken.

One major difference between Bitcoins and traditional banking is that you do not have a place, or few places, that record the balance on your account. Traditional banking involves a central ledger which specifies the current balance in each account, for example

account owner	balance
Alice	£10.01
Bob	£4.99
Charlie	-£1.23
Eve	£0.00

Bitcoins work differently in that there is no such central ledger, but instead a public record of all transactions ever made. This means spending money corresponds to sending messages of the (oversimplified) form

$$\{\text{I, Alice, am giving Bob one Bitcoin.}\}_{K^{priv}_{Alice}} \tag{1}$$

These messages, called transactions, are the only data that is ever stored in the Bitcoin system (we will come to the precise details later on). The transactions are encrypted with Alice's private key so that everybody, including Bob, can use Alice's public key K^{pub}_{Alice} to verify that this message came really from Alice, or more precisely from the person who knows K^{priv}_{Alice}.

The problem with such messages in a distributed system is that what happens if Bob receives 10, say, of these transactions? Did Alice intend to send him 10 Bitcoins, or did the message get duplicated by for example an attacker replaying a sniffed message? What is needed is a kind of serial number for such transactions. This means transaction messages shoul look more like

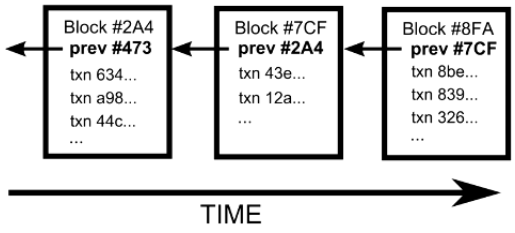
$$\{\text{I, Alice, am giving Bob Bitcoin \#1234567.}\}_{K^{priv}_{Alice}}$$

There are two difficulties, however, that need to be solved with serial numbers. One is who is assigning serial numbers to Bitcoins and also how can Bob verify that Alice actually owns this Bitcoin to pay him? In a system with a bank as trusted third-party, Bob could do the following:

- Bob asks the bank whether the Bitcoin with that serial number belongs to Alice and Alice hasn't already spent this Bitcoin.
- If yes, then Bob tells the bank he accepts this Bitcoin. The bank updates the records to show that the Bitcoin with that serial number is now in Bob's possession and no longer belongs to Alice.

But for this banks would need to be trusted and would also be an easy target for any government interference, for example. Think of the early days of music sharing where the company Napster was the trusted third-party but also the single point of "failure" which was taken offline by law enforcement. Bitcoins is more like a system such as BitTorrent without a single central entity that can be taken offline.[2]

Bitcoins solve the problem of not being able to rely on a bank by making everybody the "bank". Everybody who cares can have the entire transaction history starting with the first transaction made in January 2009. This history of transactions is called the *blockchain*. Bob, for example, can use his copy of the blockchain for determining whether Alice owned the Bitcoin he received, and if she did, he transmits the message that he owns it now to every other participant on the Bitcoin network. An illustration of a three-block segment of the blockchain is (simplified) as follows



(2)

The chain grows with time. Each block contains a list of individual transactions, written txn in the picture above, and also a reference to the previous block, written prev. The data in a block (txn's and prev) is hashed so that the reference and transactions in them cannot be tampered with. This hash is also the unique serial number of each block. Since this previous-block-reference is also part of the hash, the whole chain is robust against tampering. I let you think why this is the case?…But does it actually eliminate all possibilities of fraud?

We can check the consistency of the blockchain by checking whether all the references and hashes are correctly recorded. I have not tried it myself, but it is said that with the current amount of data (appr. 12GB) it takes roughly a day to check the consistency of the blockchain on a normal computer. Fortunately this "extended" consistency check usually only needs to be done once. Afterwards the blockchain only needs to be updated consistently.

[2]There is some Bitcoin infrastructure that is not so immune from being taken offline: for example Bitcoin exchanges, HQs of Bitcoin mining pools, Bitcoin developers and so on.

Recall I wrote earlier that Bitcoins do not maintain a ledger, which lists all the current balances in each account. Instead only transactions are recorded. While a current balance of an account is not immediately available, it is possible to extract from the blockchain a transaction graph that looks like the picture shown in Figure 1. Each rectangle represents a single transaction. Take for example the rightmost lower transaction from Charles to Emily. This transaction has as receiver the address of Emily and as the sender the address of Charles. In this way no Bitcoins can appear out of thin air (we will discuss later how Bitcoins are actually generated). If Charles did not have a transaction of at least the amount he wants to give Emily to his name (i.e. send to an address with his public-private key) then there is no way he can make a payment to Emily. Equally, if now Emily wants to pay for a coffee, say, with the Bitcoin she received from Charles she can essentially only forward the message she received. The only slight complication with this setup in Bitcoins is that "incoming" Bitcoins can be combined in a transaction and "outgoing" Bitcoins can be split. For example in the leftmost upper transactions in Figure 1, Fred makes a payment to Alice. But this payment (or transaction) combines the Bitcoins that were send by Jane to Fred and also by Juan to Fred. This allows you to "consolidate" your funds: if it were only possible to split transactions, then the amounts would get smaller and smaller.

In Bitcoins you have the ability to both combine incoming transactions, but also to split outgoing transactions to potentially more than one receiver. The latter is also needed. Consider again the rightmost transactions in Figure 1 and suppose Alice is a coffeeshop owner selling coffees for 1 Bitcoin. Charles received a transaction from Zack over 5 Bitcoins, say. How does Charles pay for the coffee? There is no explicit notion of *change* in the Bitcoin system. What Charles has to do instead is to make one single transaction with 1 Bitcoin to Alice and with 4 Bitcoins going back to himself, which then Charles can use to give to Emily, for example.

Let us consider another example. Suppose Emily received 4 Bitcoins from Charles and independently received another transaction (not shown in the picture) that sends 6 Bitcoins to her. If she now wants to buy a coffee from Alice for 1 Bitcoin, she has two possibilities: She could just forward the transaction from Charles over 4 Bitcoins to Alice split in such a way that Alice receives 1 Bitcoin and Emily sends the remaining 3 Bitcoins back to herself. In this case she would now be in the possession of two unspend Bitcoin transactions, one over 3 Bitcoins and the independent one over 6 Bitcoins. Or, Emily could combine both transactions (one over 4 Bitcoins from Charles and the independent one over 6 Bitcoins) and then split this amount with 1 Bitcoin going to Alice and 9 Bitcoins going back to herself.

I think this is a good time for you to pause to let this concept of transactions to really sink in…You should come to the conclusion that there is really no need for a central ledger and no need for an account balance as familiar from traditional banking. The closest what Bitcoin has to offer for the notion of a balance in a bank account are the unspend transactions that a person (more precisely a public-private key address) received. That means transactions that can still be

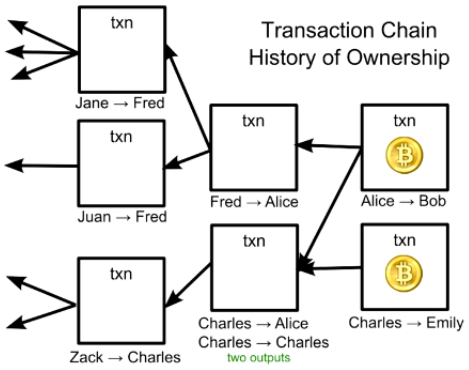


Figure 1: Transaction graph that is implicitly recorded in the public blockchain.

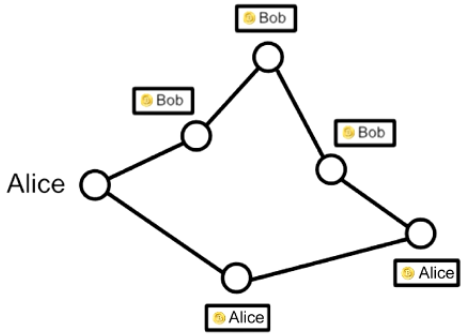
forwarded.

After the pause also consider the fact that whatever transaction is recorded in the blockchain will be in the "historical record" for the Bitcoin system. If a transaction says 1 Bitcoin goes from address A to address B, then this is what will be—B has then the possibility to spend the corresponding Bitcoins, whether the transaction was done fraudulently or not. There is no exception to this rule. Interestingly this is also how Bitcoins can get lost: One possibility is that you send Bitcoins to an address for which nobody has generated a private key, for example because of a typo in the address field—bad luck for fat fingers[3] in the Bitcoin system. The reason is that nobody has a private key for this erroneous address and consequently cannot forward the transaction anymore. Another possibility is that you forget your private key and you had messages forwarded to the corresponding public key. Also in this case bad luck: you will never be able to forward this message again, because you will not be able to form a valid message that sends this to somebody else (we will see the details of this later). But this is also a way how you can get robbed of your Bitcoins. By old-fashioned hacking-into-a-computer crime, for example, an attacker might get hold of your private key and then quickly forwards the Bitcoins that are in your name to an address the attacker controls. You will never again have access to these Bitcoins, because for the Bitcoin system they are assumed to be spent. And remember with Bitcoins you cannot appeal to any higher authority. Once the Bitcoins are gone, they are gone. This is much different in traditional banking where at least you can try to harass the bank to roll back the transaction.

This brings us to back to problem of double spend. Suppose Bob is a merchant. How can he make sure that Alice does not cheat him? She could for example send a transaction to Bob. But also forward the "same" transaction to Charlie, or even herself. If Alice manages to get the second transaction into the

[3] http://en.wikipedia.org/wiki/Typographical_error

blockchain, Bob will be cheated out of his money. The problem in such conflicting situations is how should the network update their blockchain? You might end up with a picture like this



where Alice convinced some part of the "world" that she is still the owner of the Bitcoin and some other part of the "world" thinks it's Bob's. How should such a disagreement be resolved? This is actually the main hurdle where Bitcoin really innovated. The answer is that Bob needs to convince "enough" people on the network that the transaction from Alice to him is legit.

What does, however, "enough" mean in a distributed system? If Alice sets up a network of a billion, say, puppy identities and whenever Bob tries to convince, or validate, that he is the rightful owner of the Bitcoin, then the puppy identities agree. Bob would then have no reason to not give Alice her coffee. But behind his back she has convinced everybody else on the network that she is still the rightful owner of the Bitcoin. After being outvoted, Bob would be a tad peeved.

The reflex reaction to such a situation would be to make the process of validating a transaction as cheap as possible. The intention is that Bob will easily get enough peers to agree with him that he is the rightful owner. But such a solution has always the limitation of Alice setting up an even bigger network of puppy identities. The really cool idea of Bitcoin is to go into the other direction of making the process of transaction validation (artificially) as expensive as possible, but reward people for helping with the validation. This is really a novel and counterintuitive idea that makes the whole system of Bitcoins work so beautifully.

Proof-of-Work Puzzles

In order to make the process of transaction validation difficult, Bitcoin uses a kind of puzzle. Solving the puzzles is called *Bitcoin mining*, where whoever solves a puzzle will be awarded some Bitcoins. At the beginning this was 50 Bitcoins, but the rules of Bitcoin are set up such that this amount halves every 210,000 transactions or so. Currently you will be awarded 25 Bitcoins for solving a puzzle. Because the amount will halve again and then later again

and again, around the year 2140 it will go below the level of 1 Satoshi. In that event no new Bitcoins will ever be created again and the amount of Bitcoins stays fixed. There will be still an incentive to help with validating transactions, because there is the possibility in Bitcoins to offer a transaction fee to whoever solves a puzzle. At the moment this fee is usually set to 0, since the incentive for miners is the 25 Bitcoins that are currently awarded for solving puzzles.

What do the puzzles that miners have to solve look like? The puzzles can be illustrated roughly as follows: Given a string, say `"Hello, world!"`, what is the salt so that the hash starts with a long run of zeros? Let us look at a concrete example. Recall that Bitcoins use the hash-function SHA-256. Suppose we call this hash function `h`, then we could try the salt `0` as follows:

```
h("Hello, world!0") =
  1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64
```

OK this does not have any zeros at all. We could next try the salt `1`:

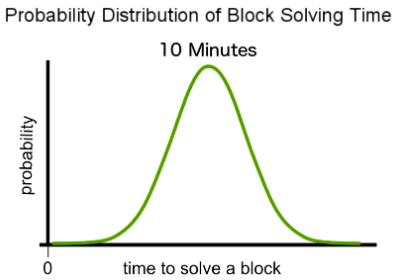
```
h("Hello, world!1") =
  e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8
```

Again this hash value does not contain any leading zeros. We could now try out every salt until we reach

```
h("Hello, world!4250") =
  0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9
```

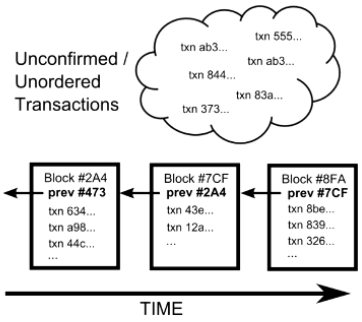
where we have four leading zeros. If four zeros are enough, then the puzzle would be solved with this salt. The point is that we can very quickly check whether a salt solves a puzzle, but it is hard to find one. Latest research suggest it is an NP-problem. If we want the output hash value to begin with 10 zeroes, say, then we will, on average, need to try $16^{10} \approx 10^{12}$ different salts before we find a suitable one.

In Bitcoins the puzzles are not solved according to how many leading zeros a hash-value has, but rather whether it is below a *target*. The hardness of the puzzle can actually be controlled by changing the target according to the available computational power available. I think the adjustment of the hardness of the problems is done every 2060 blocks (appr. every two weeks). I am not sure whether this is an automatic process. The aim of the adjustment is that on average the Bitcoin network will most likely solve a puzzle within 10 Minutes.



It could be solved quicker, but equally it could take longer, but on average after 10 Minutes somebody on the network will have found a solution.

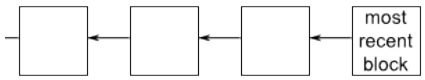
Remember that the puzzles are a kind of proof-of-work that make the validation of transactions artificially expensive. Consider the following picture with a blockchain and some unconfirmed transactions.



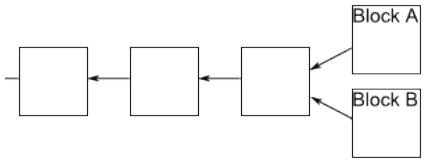
(3)

The puzzle is stated as follows: There are some unconfirmed transactions. Choosing some of them, the miner (i.e. the person/computer that tries to solve a puzzle) will form a putative block to be added to the blockchain. This putative block will contain the transactions and the reference to the previous block. The serial number of such a block is simply the hash of all the data. The puzzle can then be stated as the "string" corresponding to the block and which salt is needed in order to have the hashed value being below the target. Other miners will choose different transactions and therefore work on a slightly different putative block and puzzle.

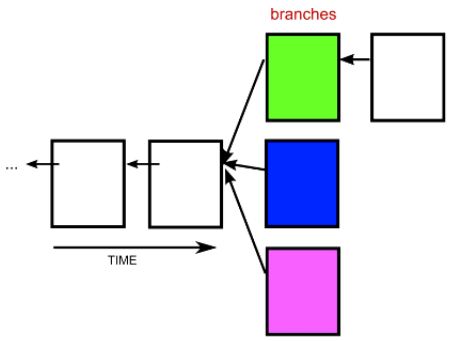
The intention of the proof-of-work puzzle is that the blockchain is at every given moment linearly ordered, see the picture shown in (3). If we don't have such a linear ordering at any given moment then it may not be clear who owns which Bitcoins. Assume a miner David is lucky and finds a suitable salt to confirm some transactions. Should he celebrate? Not yet. Typically the blockchain will look as follows



But every so often there will be a fork



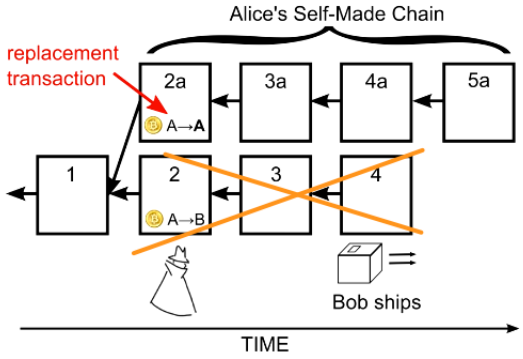
What should be done in this case? Well, the tie is broken if another block is solved, like so:



The rule in Bitcoins is: If a fork occurs, people on the network keep track of all forks (they can see). But at any given time, miners only work to extend whichever fork is longest in their copy of the block chain. Why should miners work on the longest fork? Well their incentive is to mine Bitcoins. If somebody else already solved a puzzle, then it makes more sense to work on a new puzzle and obtain the Bitcoins for solving that puzzle, rather than waste efforts on a fork that is shorter and therefore less likely to be "accepted". Note that whoever solved a puzzle on the "loosing" fork will actually not get any Bitcoins as reward. Tough luck.

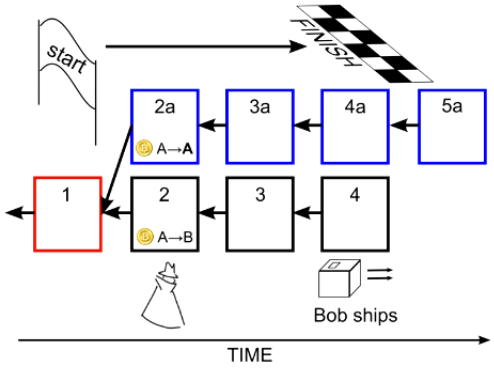
Alice against the Rest of the World

Let us see how the blockchain and the proof-of-work puzzles avoid the problem of double spend. If Alice wants to cheat Bob, she would need to pull off the following ploy:



Alice makes a transaction to Bob for paying, for example, for an online order. This transaction is confirmed, or validated, in block 2. Bob ships the goods around block 4. In this moment, Alice needs to get into action and try to validate the fraudulent transaction to herself instead. At this moment she is in a race against all the computing power of the "rest of the world". Because the

incentive of the rest of the world is to work on the longest chain, that is the one with the transaction from Alice to Bob:



As shown in the picture she has to solve the puzzles 2a to 5a one after the other, because the hash of a block is determined via the reference by all the data in the previous block. She might be very lucky to solve one puzzle for a block before the rest of the world, but to be lucky many times is very unlikely. This principle of having to race against the rest of the world avoids the ploy of double spend.

In order to raise the bar for Alice even further, merchants accepting Bitcoins use the following rule of thumb: A transaction is "confirmed" if

(1) it is part of a block in the longest fork, and

(2) at least 5 blocks follow it in the longest fork. In this case we say that the transaction has 6 confirmations.

A simple calculation shows that this amount of confirmations can take up to 1 hour and more. While this seems excessively long, from the merchant's point of view it is not that long at all. For this recall that ordinary creditcards can have their transactions been rolled-back for 6 months or so. The point however is that the odds for Alice being able to cheat are very low, unless she can muster more than 50% of the world Bitcoin mining capacity. In this case she could out-race the rest of the world. The point is however that amassing such an amount of computing power is practically impossible for a single person or even a moderately large group.

Connected with the 6-confirmation rule is an interesting phenomenon. On average, it would take several years for a typical computer to solve a proof-of-work puzzle, so an individual's chance of ever solving one before the rest of the world, which typically takes only 10 minutes, is negligibly low. Therefore many people join groups called *mining pools* that collectively work to solve blocks, and distribute rewards based on work contributed. These mining pools act somewhat like lottery pools among co-workers, except that some of these pools are quite large, and comprise more than 20% of all the computers in the network. It is said that BTC, a large mining pool, has limited its number of

members in order to not solve more than 6 blocks in a row. Otherwise this would undermine the trust in Bitcoins, which is also not in the interest of BTC, I guess. Some statistics on mining pools can be seen at

https://blockchain.info/pools

Bitcoins for Real

Let us now turn to the nitty gritty details. As a participant in the Bitcoin networ you need to generate and store a public-private key pair. The public key you need to advertise in order to receive payments (transactions). The private key needs to be securely stored. For this there seem to be three possibilities

- an electronic wallet on your computer
- a cloud-based storage (offered by some Bitcoin services)
- paper-based

The first two options of course offer convenience for making and receiving transactions. But given the nature of the private keys and how much security relies on them (recall if somebody gets hold of it, your Bitcoins are quickly lost forever) I would opt for the third option for anything except for trivial amounts of Bitcoins. As we have seen earlier in the course, securing a computer system that it can withstand a breakin is still very much an unsolved problem.

An interesting fact with Bitcoin keys is that there is no check for duplicate addresses. This means when generating a public-private key, you should really start with a carefully chosen random number such that there is really no chance to step on somebody's feet in the 2^{160} space of possibilities. Again if you share an address with somebody else, he or she has access to all your unspend transactions. The absence of such a check is easily explained: How would one do this in a distributed system? The answer you can't. It is possible to do some sanity check of addresses that are already used in the blockchain, but this is not a fail-proof method. One really has to trust on the enormity of the 2^{160} space for addresses.

Let us now look at the concrete data that is stored in an transaction message:

```
{"hash":"7c4025...",
 "ver":1,
 "vin_sz":1,
 "vout_sz":1,
 "lock_time":0,
 "size":224,
 "in":[
    {"prev_out":
      {"hash":"2007ae...",
       "n":0},
       "scriptSig":"304502... 042b2d..."}],
```

```
12  "out":[
13    {"value":"0.31900000",
14     "scriptPubKey":"OP_DUP OP_HASH160 a7db6f...
15                    OP_EQUALVERIFY OP_CHECKSIG"}]}
```

The hash in Line 1 is the hash of all the data that follows. It is a kind of serial number for the transaction. Line 2 contains a version number in case there are some incompatible changes to be made. Lines 3 and 4 specify how many incoming transactions are combined and how many outgoing transactions there are. In our example there are one for each. Line 5 specifies a lock time for when the transaction is supposed to become active—this is usually set to 0 to become active immediately. Line 6 specifies the size of the message; it has nothing to do with the Bitcoins that are transferred. Lines 7 to 11 specify where the Bitcoins in the transaction are coming from. The has in line 9 specifies the incoming transaction and the `n` in Line 10 specifies which output of the transaction is referred to. The signature in line 11 specifies the address (public key K^{pub}) from where the Bitcoins are taken and the digital signature of the address, that is $\{K^{pub}\}_{K^{priv}}$. Lines 12 to 15 specify the value of the first outgoing transaction. In this case 0.319 Bitcoins. The hash in Line 14 specifies the address to where the Bitcoins are transferred.

As can be seen there is no need to issue serial numbers for transactions, the hash of the transaction data can do this job. The hash will contain the sender addresses and hash-references to the incoming transactions, as well as the public key of the incoming transaction. This uniquely identifies a transaction and the hash is the unique fingerprint of it. The in-field also contains the address to which a earlier transaction is made. The digital signature ensures everybody can check that the person who makes this transaction is in the possession of the private key. Otherwise the signature would not match up with the public-key address.

When mining the blockchain it only needs to be ensured that the transactions are consistent (all hashes and signatures match up). Then we need to generate the correct previous-block link and solve the resulting puzzle. Once the block is accepted, everybody can check the integrity of the whole blockchain.

A word of warning: The point of a lottery is that some people win. But equally, that most people lose. Mining Bitcoins has pretty much the same point. According to the article below, a very large machine (very, very large in terms of June 2014) could potentially mine $40 worth of Bitcoins a day, but would require magnitudes more of electricity costs to do so.

http://bitcoinmagazine.com/13774/
government-bans-professor-mining-bitcoin-supercomputer/

Bitcoin mining nowadays is only competitive, or profitable, if you get the energy for free, or use special purpose computing devices.

This about "free" energy can actually hurt you very badly in unexpected ways. You probably have heard about, or even used, Amazon's Elastic Compute Cloud (EC2). Essentially, Amazon is selling computing power that you

can use to run your web site, for example. It is *elastic* in the sense that if you have a lot of visitors, you pay a lot, if you have only a few, then it is cheap. In order to bill you they, you need to set up an account with Amazon and receive some secret keys in order to authenticate you. The clever (but also dangerous) bit is that you upload the code of your web site to GitHub and Amazon will pull it from there. You can probably already guess where this is going: in order to learn about Amazon's API, it gives out some limited computing power for free. Somebody used this offer in order to teach himself Ruby on Rails with a mildly practical website. Unfortunately, he uploaded also his secret keys to GitHub (this is really an easy mistake). Now, nasty people crawl GitHub for the purpose of stealing such secret keys. What can they do with this? Well, they quickly max out the limit of computing power with Amazon and mine Bitcoins (under somebody else's account). Fortunately for this guy, Amazon was aware of this scam and in a goodwill gesture refunded him the money the nasty guys incurred over night with their Bitcoin mining. If you want to read the complete story, google for "My $2375 Amazon EC2 Mistake".

Anonymity with Bitcoins

One question one often hears is how anonymous is it actually to pay with Bitcoins? Paying with paper money used to be a quite anonymous act (unlike paying with credit cards, for example). But this has changed nowadays: You cannot come to a bank anymore with a suitcase full of money and try to open a bank account. Strict money laundering and taxation laws mean that not even Swiss banks are prepared to take such money and open a bank account. That is why Bitcoins are touted as filling this niche again of anonymous payments.

While Bitcoins are intended to be anonymous, the reality is slightly different. I fully agree with the statement by Nielsen from the blog article I referenced at the beginning:

> *"Many people claim that Bitcoin can be used anonymously. This claim has led to the formation of marketplaces such as Silk Road (and various successors), which specialize in illegal goods. However, the claim that Bitcoin is anonymous is a myth. The block chain is public, meaning that it's possible for anyone to see every Bitcoin transaction ever. Although Bitcoin addresses aren't immediately associated to real-world identities, computer scientists have done a great deal of work figuring out how to de-anonymise 'anonymous' social networks. The block chain is a marvellous target for these techniques. I will be extremely surprised if the great majority of Bitcoin users are not identified with relatively high confidence and ease in the near future."*

The only thing I can add to this is that with the Bitcoin blockchain we will in the future have even more pleasure hearing confessions from reputable or not-so-reputable people, like the infamous "I did not inhale" from an US president.[4]

[4] www.youtube.com/watch?v=Bktd_Pi4YJw

The whole point of the blockchain is that it public and will always be.

There are some precautions one can take for boosting anonymity, for example to use a new public-private key pair for every new transaction, and to access Bitcoin only through the Tor network. But the transactions in Bitcoins are designed such that they allow one to combine incoming transactions. In such cases we know they must have been made by the single person who knew the corresponding private keys. So using different public-private keys for each transaction might not actually make the de-anonymisation task much harder. And the point about de-anonymising 'anonymous' social networks is that the information is embedded into the structure of the transition graph. And this cannot be erased with Bitcoins.

One paper that has fun with spotting transactions made to Silk Road (2.0) and also to Wikileaks is

`http://people.csail.mit.edu/spillai/data/papers/bitcoin-transaction-graph-analysis.pdf`

A paper that gathers some statistical data about the blockchain is

`https://eprint.iacr.org/2012/584.pdf`

Government Meddling

Finally, what are the options for a typical Western government to meddle with Bitcoins? This is of course one feature the proponents of Bitcoins also tout: namely that there aren't any options. In my opinion this is far too naive and far from the truth. Let us assume some law enforcement agencies would not have been able to uncover the baddies from Silk Road 1.0 and 2.0 (they have done so by uncovering the Tor network, which is an incredible feat on its own). Would the government in question have stopped? I do not think so. The next target would have been Bitcoin. If I were the government, this is what I would consider:

- The government could compel "mayor players" to blacklist Bitcoins (for example at Bitcoin exchanges, which are usually located somewhere in the vicinity of the government's reach). This would impinge on what is called *fungibility* of Bitcoins and make them much less attractive to baddies. Suddenly their "hard-earned" Bitcoin money cannot be spent anymore. The attraction of this option is that this blacklisting can be easily done "whole-sale" and therefore be really be an attractive target for governments & Co.

- The government could attempt to coerce the developer community of the Bitcoin tools. While this might be a bit harder, we know certain governments are ready to take such actions (we have seen this with Lavabit, just that the developers there refused to play ball and shut down their complete operation).

- The government could also put pressure on mining pools in order to blacklist transactions from baddies. Or be a big miner itself. Given the gigantic facilities that are built for institutions like the NSA (pictures from the Utah dessert)


  this would not be such a high bar to jump over. Remember it "only" takes to be temporarily in control of 50%-plus of the mining capacity in order to undermine the trust in the system. Given sophisticated stories like Stuxnet (where we still do not know the precise details) maybe even such large facilities are not really needed. What happens, for example, if a government starts DoS attacks on existing miners? They have complete control (unfortunately) of all mayor connectivity providers, i.e. ISPs.

  There are estimates that the Bitcoin mining capacity outperforms the top 500 supercomputers in the world, combined(!):

  http://www.forbes.com/sites/reuvencohen/2013/11/28/global-bitcoin-computing-power-now-256-times-faster-than-top-500-supercomputers-combined/

  But my gut feeling is that these are too simplistic calculations. In security (and things like Bitcoins) the world is never just black and white. The point is once the trust is undermined, the Bitcoin system would need to be evolved to Bitcoins 2.0. But who says that Bitcoin 2.0 will honour the Bitcoins from Version 1.0?

A government would potentially not really need to follow up with such threads. Just the rumour that it would, could be enough to get the Bitcoin-house-of-cards to tumble. Some governments have already such an "impressive" trackrecord in this area, such a thread would be entirely credible. Because of all this, I would not have too much hope that Bitcoins are free from interference by governments & Co when it will stand in their way, despite what everybody else is saying. To sum up, the technical details behind Bitcoins are simply cool. But still the entire Bitcoin ecosystem is in my humble opinion rather fragile.

Further Reading

Finally, finally, the article

http://www.extremetech.com/extreme/155636-the-bitcoin-network-outperforms-the-top-500-supercomputers-combined

makes an interesting point: If people are willing to solve meaningless puzzles for hard, cold cash and with this achieve rather impressive results, what could we achieve if the UN, say, would find the money and incentivise people to, for example, solve protein folding puzzles?[5] For this there are projects like Folding@home.[6] This might help with curing diseases such as Alzheimer or diabetes. The same point is made in the article

http://gizmodo.com/
the-worlds-most-powerful-computer-network-is-being-was-504503726

A definitely interesting and worthy use of Bitcoins has been explored in the thesis

http://enetium.com/resources/Thesis.pdf

where the author proposes ways of publishing information that is censor resistant as part of the blockchain. The idea is that if a government wants to use Bitcoins, it would also have to put up with plain-text data that can be included in a transaction.

[5] http://en.wikipedia.org/wiki/Protein_folding
[6] http://folding.stanford.edu