Security Mechanisms in Digital Currencies

Andrew J G Walker James C Moss
Email: {andwa932, jammo108}@student.liu.se
Supervisor: Ulf Kargén, {ulf.kargen@liu.se}
Project Report for Information Security Course
Linköpings Universitet, Sweden

Abstract

As the use of digital currencies increases along with the monetary value held in those currencies, it is also increasingly important to understand the underlying security requirements and mechanisms used in those currencies. This paper will not discuss older digital currencies but will instead focus on modern decentralised crypto currencies; in particular, Bitcoin, Litecoin and their variants. Due to this, throughout this paper "Bitcoin" will be used when referring to Bitcoin and its variants. This paper will discuss the security requirements of cryptocurrencies, how these requirements are implemented and possible attack types. There will be a focus on two attacks in particular; the 51% attack and what shall be called the "Android Java RNG attack".

1. Introduction

Over the past few years the general public have begun to hear about digital currency with the large scale media coverage about booming prices of a "made up coin" while being able to transfer this into fiat currency. Many people fail to understand digital currencies, the security behind digital currencies and how they work. A common philosophy is that legal tender can be copied, so surely digital currencies can be too? Many people believe that they could possibly make a breakthrough into everyday use in the near future, whilst others believe that governments will intervene and legislate against their ownership by citizens and of its use as a currency. "Cryptology represents the future of privacy (and) by implication (it) also represents the future of money, and the future of banking and finance."[1] Another quote that represents what people in the technology sector think of cryptocurrencies: "You can't stop things like Bitcoin (...) it's like trying to stop gunpowder."[2] The fiat (money) value of Bitcoin has increased dramatically over the last two year with the value of a Bitcoin standing at $443 USD [3].

This report will cover the security concepts behind various digital currencies and how it currently keeps the digital currency world secure. To understand the security mechanisms we must first analyse and understand the security requirements of a digital currency; this is covered initially in this report. The cryptography used is also explained as well as how these requirements are implemented in software. Finally two attacks are discussed in detail; the theoretical 51% attack and the "Android Java RNG attack".

2. Security Requirements

The basic security requirements of a cryptocurrency are as follows:

- The creation of coins must be done in a secure and verifiable way that cannot be short circuited. An amount of work must be expended that can be proved. Proof of work.
- Each coin, or part coin must have a single owner and cannot be re spent. To do this without a central authority a history of all transactions much be kept by each entity using the network. In Bitcoin and similar currencies this is called the blockchain.
-Transactions must happen securely between two parties without a central authority.

3. Implementation and terminology

This sections aims to explain how Bitcoin and it's variants work at a high level. Terminology that is used throughout the rest of the paper is also explained.

Definitions

A **hash function** takes an arbitrary input known as a message, performs a one way function on it, and returns a fixed size bit string known as the hash value. Non identical messages should generate different and unpredictable hash values. **Collision resistance** is a required feature of a cryptographic hash function. A hash

function is said to be collision resistant when it is hard to find two inputs that produce the same hash output.

An **ASIC unit** is a specifically designed chip for a particular purpose. A Bitcoin ASICs sole purpose is to calculate SHA-256 hashes.

SHA256 - Is part of a group of cryptographic algorithms (SHA-2), and is the one way hash function used in the creation of Bitcoin[4] and Peercoin[5].It is collision resistant and has a an output size of 256 bits. It is difficult for everyday users to mine cryptocurrencies that use SHA256 without large investments due to ASIC units being required for profitable mining.

A **cryptocurrency** is a medium of exchange that is awarded by solving cryptographic problems and can then be used as a form of currency for purchases or money transfer.

Each and every user that makes a transaction into or out of a wallet in cryptocurrencies has a wallet **address**. This is essentially a number that represents your wallet and only your wallet. It is unique. When a transaction takes place a sender will use this address to send their coins to.

The **Proof of work** system in cryptocurrencies involves proving that an amount of computation effort was expended to solve a problem. For example in SHA256 it is easy to verify a hash but computationally expensive given a hash to work out the input. Therefore if the problem is to find the input that creates a hash with certain qualities then the proof is that input.

A **block** is a group of transactions that are "solved". A block references the block before it (it's parent) and becomes part of the blockchain when it forms part of the longest (most complex) chain of blocks.

The **Blockchain** is made up of all valid blocks and therefore contains the history of all transactions and enables each node to verify the claimed owner of a Bitcoin or part Bitcoin. This means that double spending cannot occur as each transaction must be verified by the majority of users. To **double spend** means that the same input coin or part coin is used in two transactions.

3.1 Transactions and Mining

In Bitcoin processing transactions and mining are closely linked.

A blockchain consists of thousands of **blocks.** Put simply a **block** is "mined" by taking a group of transactions that are waiting to be processed along with a nonce value and other parameters and using them as input into a hash function to create a hash with a leading number of 0's. The nonce value is iterated by the miner to brute force the hashing algorithm so that the output contains an "x" amount of leading 0's. Once this criteria is satisfied the block is said to be "solved" and the block reward is given to the person providing the solution. This process is known as mining. The block reward changes depending on the cryptocurrency and protocol, but in Bitcoin it is currently 25 BTC. The amount of leading zeros is known as the block difficulty and the act of obtaining a hash with the required amount of leading zeros is the proof of work.

An **blockchain fork** is created when two block are discovered at the same time and worked on by miners independently as the parent block. As different sections of the network choose different blocks to use as a parent a fork is created. The fork is solved in subsequent blocks as only the longest (most complex) chain is eventually followed.

A block becomes an **orphan block** when it doesn't have a parent in the longest chain this can also be thought of as the losing block(s) in the fork.

A **mining pool** is a service where multiple users that mine a cryptocurrency can work together to solve a block. With more people working together the reward and work is divided and produced more often than when "solo mining" (singular entity) which gives a smoother amount of rewards over a long period.

A **digital wallet** is the same as a real life wallet, you keep money in it. However, its more like a safe, you can encrypt it to keep your money safe which prevents people accessing it.

SCRYPT - Is used in the creation of Litecoin[6] and Dogecoin[7]. Everyone can mine these coins with GPUs as the protocol is designed so that it is not cost effective to implement within an ASIC. It was designed to make it difficult for hardware to brute force and therefore requires a lot of money to invest into mining them, making them unfeasible for a large scale mining operation. No ASIC chips have been developed for these as of yet.

Digital signatures are used in asymmetric public key cryptography. It identifies the creator of the transaction as it can only be created by the sender and it is completely unique.

ECDSA - The Elliptic Curve Digital Signature Algorithm is used to create the digital signatures used in Bitcoin and its variants. An advantage of ECDSA [8] over standard DSA [9] is that even with the same security level (measured in bits) the public key generated using ECDSA is much smaller than in DSA.

4. Attacks and attack types

5. 51% Attack

The 51% attack is probably the biggest threat to all cryptocurrencies. The power that this attack gives to the owner is incredible. In reality it could entirely destroy a cryptocurrency, and at minimum cripple it for a very long period.

If an attacker has control of more than 51% of the network, they can for the time that they are in control, exclude and modify the ordering of all transactions. This is a huge problem. As quoted comically by Gavin Andreson a chief scientist at Bitcoin, "That would be bad."[10]. It would be far more than bad. An additional protective protocol would need be implemented by Bitcoin to avoid this happening as there is currently nothing in the protocol to stop a 51% attack.

Bitcoin is based works on a majority consensus principal. When one block is mined and then solved, the next block is moved onto. If two blocks are discovered at the same time, the blockchain forks. The block is then sent across the network and is accepted and worked upon by other miners. Eventually only the longest (most complex) chain forms the blockchain and the "loosing" non-forked blocks are orphaned.

When a mining pool controls more of the network than another, it essentially allows 1 user, or a group of people, to be the operator of the consensus network. To achieve this, a lot of computing power is required. There are so many different variants to working out an approximate cost of an attack. As of May 2014, Bitcoin hashes peaked at 67 million giga hashes per second (GH/s)[11]. To perform a 51% attack you would need to add computing power equivalent to 68.5 million GH/s on top of what already is being used and sustain this. An approximate estimate of this is $829 million (as of May 2014)[12]. Included in this approximate cost is the cost of hardware, electricity and purchasing hashing power from other sources.

Calculating the cost with multiple variants from a personal point of view, with the original cost of productions of ASIC units, electricity and computing hardware, this number can be brought down to around $200 million. At this cost, or even the predicted $829 million, governments, banks, and random groups of people could plan an attack. If it reaches a point in the future where banks are losing money due to Bitcoin taking over other fiat currencies used by powerful nations, an attack could be imminent.

The thought of this attack led to Gavin Andreson making a suggestion (in 2012) [13] to avoid the 51% attack blocking transactions (transaction denial of service) by ignoring long chains of 0 transaction blocks in the blockchain. Once these are discarded it would create an orphan block chain. This however can be worked around by the attacker by blocking (e.g) 10% - 20% of the transactions. It would then not be noticed as a blocking attack due to some transactions taking place, but still change how Bitcoin works, making it unreliable. This would also cause multiple transactions to be redundant and allow double spending to still happen as an attacker still has control of the consensus network, therefore is no longer a viable option.

With control of the network it is also possible to win all block rewards, or the majority, giving huge profits. This is known as a mining monopoly. It bypasses the long orphan chain defence technique by the attacker allowing all transactions to take place, but mining all of the final block pieces receiving all of the rewards. As per the design of the protocol six blocks are aimed to be produced each hour with the difficulty adjusted every 10 blocks to try and meet this requirement. With a reward of 25 BTC per block, an average day produces 3600 BTC. This is equivalent with current prices[14], to $1.53 million a day. A successful block attack using the 51% technique, could produce this each day if the prices did not drop, which in reality, they would.

The first and only case so far of the 51% pool occurrence was avoided as it was not an intended attack. In January 2014, ghash.io's mining pool managed to reach 42%[15][16] before an uproar from the Bitcoin community combined with the compliance of ghash.oi, caused the problem to be avoided. The mining pool was very popular with the community as there was no pool fee, meaning people were not charged for mining with them. A spokesman from ghash.io said, "GHash.IO does not have any intentions to execute a 51% attack, as it will do serious damage to the Bitcoin community, of which we are part of."[17].

Further measures were then taken by ghash to avoid reaching 51%. They stopped accepting new individual people into the mining pool and implemented a system

that when users purchased GH's that they could use any pool without a fee.

"We will temporarily stop accepting new independent mining facilities to the Ghash.IO pool."[17]

"We will implement a feature, allowing CEX.IO users to mine Bitcoins from

other pools. So when they purchase GH/s they can put it towards any pool

they choose."[17]

This is the closest and the only occurrence that has managed to come close to a 51% takeover of the network. This was also unintentional as far as everyone is aware and an intentional attack has never occurred.

6. Android Java RNG Attack

This section of the report aims to explain the flaw that makes the Java PRNG (Pseudo Random Number Generator) attack possible and how these flaws were used to steal Bitcoins.

The Android Java RNG attack was possible due to poorly generated pseudo random numbers. This was caused by a bug within the Android implementation of the Java SecureRandom class which contained a vulnerability that prevented the generation of secure random numbers [18]. The SecureRandom class is supposed to obtain an entropy seed, which is essentially pseudo random data generated by the operating system, from a file located at dev/urandom, but in the case of this bug the file was not accessed at all. This means that a random seed was not generated to create the random number.

These poorly generated "random" numbers were then used to create the ECDSA signature. The formula to calculate as ECDSA signature is as follows:

Publically shared values:
p,a,b,G,N (elliptic curve parameters)
Q = public key
e = hash of data
R,S = signature

Private values:
m = random integer
k = private key

A created signature has two publically shared values R and S, calculated as follows:
R=(mG)x
S=e+kR/m.

If m is not randomly generated and two signatures are created using the same m then the R value in both signatures is identical which makes it possible to calculate the private key value along with the random m value used. The calculation is as follows:

R = (mG)x
S1 = e1 + kR / m
R = (mG)x
S2 = e2 + kR / m

S1 − S2 = e1 − e2 / m
m = e1 − e2 / S1 − S2

k = mSi − ei/R [=e1S2−e2S1R(S1−S2)]

As can be seen above if the same random number and private key were used to sign two different messages the private key can be calculated. This can be viewed in the blockchain as a reoccurrence of the first value (R value) in the ECDSA signature.

This is similar to the flaw that allowed the private key used for signing software on the Playstation 3 to be discovered [19]. However in the case of the Playstation 3 a static value was used as a seed in the algorithm making it ever easier to discover the private key. Once "active" wallets were discovered with this flaw coins could be transferred using the calculated public key to a new wallet under the control of the hacker. Research carried out in January 2013 [20] scanned the entire Blockchain and discovered all vulnerable address to have a balance of zero Bitcoins.

7. Conclusion

There are varying opinions on the future of digital currencies, but what is true is the value held in those currencies is considerable with the monetary value having increased dramatically over the last two years. This paper began with an overview of key concepts and terminology necessary to understand cryptocurrency security mechanisms and attacks and went onto focus on two key attacks in detail.

Those attacks were chosen as it is important for the community to understand past and possible attacks. Understanding possible attacks such as the 51% attack allow those using and investing in the currency to understand the risks inherent to Bitcoin as a decentralised currency; whilst knowledge of past attacks such as the Java RNG attack allows developers and users to learn from past implementation mistakes. The cryptography is only strong if the mechanisms are implemented correctly as is the case with random number generation.

The security requirements of modern digital currencies are well known and the mechanisms used to implement those requirements are able to be reviewed and are open source. Attack types such as the 51% attack will always be possible in a consensus network such as Bitcoin and can only be mitigated against without a change to the underlying protocol. Conversely attacks such as the Java RNG attack can be prevented as the issue is not with the underlying cryptography or protocol but with poor implementation. As always developers should review and test their code for bugs with past mistakes in mind.

References

[1] Orlin Grabbe, Economist.

[2] John McAfee, Founder of McAfee Inc.

[3] https://blockchain.info/charts/market-price?timespan=all&showDataPoints=false&daysAverageString=1&show_header=true&scale=0&address=

[4] https://en.bitcoin.it/wiki/FAQ

[5] http://99bitcoins.com/peercoin/

[6] https://litecoin.org/

[7] http://www.businessinsider.com/what-is-dogecoin-2013-12

[8] https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm

[9] http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf (DSA Pages 15-22)

[10] Gavin Andresen, Chief Scientist at Bitcoin. See [13].

[11] https://blockchain.info/charts/hash-rate

[12] http://www.coinometrics.com/bitcoin/rix

[13] http://gavintech.blogspot.se/2012/05/neutralizing-51-attack.html

[14] https://coinbase.com/

[15] http://www.coindesk.com/bitcoin-miners-ditch-ghash-io-pool-51-attack/

[16] http://newsbtc.com/2014/01/09/ghash-io-mining-pool-nears-51-network-hashing-power/

[17] ghash.io, Mining Pool. https://ghash.io/ghashio_press_release.pdf

[18] A.Klyubin,: Some SecureRandom Thoughts, http://android-developers.blogspot.ie/2013/08/some-securerandom-thoughts.html , August 2013.

[19] Mittwach,: Console Hacking 2010 (Page 123-128), December 2010 http://events.ccc.de/congress/2010/Fahrplan/attachments/1780_27c3_console_hacking_2010.pdf

[20] J.Moore, E.Wustrow,: Bitcoins and entropy, January 2013 https://plus.google.com/106313804833283549032/posts/X1TvcxNhMWz