On Detection of Bitcoin Mining Redirection Attacks

Nicolas T. Courtois[1], Pinar Emirdag[2] and Zhouyixing Wang[1]
[1]Computer Science, University College London, London, U.K.
[2]Independent Market Structure Professional, London, U.K.

Keywords: e-Payment, Crypto Currencies, Bitcoin, Double-spending attacks, Hash Functions, Man-In-the-Middle Attacks, Stratum Protocol.

Abstract: In this paper we study the question of centralisation in bitcoin digital currency. In theory bitcoin has been designed to be a totally decentralized distributed system. Satoshi Nakamoto has very clearly postulated that *each node* should be collecting recent transactions and trying to create new blocks (Satoshi08). In bitcoin transactions are aggregated in block in order to authenticate them and form an official ledger and history of bitcoin transactions. In practice as soon as expensive ASIC bitcoin miners have replaced general-purpose hardware, production of bitcoins and the validation of transactions has concentrated in the hands of a smaller group of people. Then at some moment in early 2012 an important decision was taken: the Stratum protocol was designed (Palatinus12) which took a deliberate decision to move the power of selecting which transactions are included in blocks from miners to pool managers. The growing difficulty of mining and large standard deviation in this process (Rosenfeld13; CourtoisBahack14) made that majority of miners naturally shifted to pooled mining. At this moment bitcoin ceased being a decentralized democratic system. In this paper we survey the question of a 51% attacks and show that there is a large variety of plausible attack scenarios. In particular we study one particularly subversive attack scenario which depends on non-trivial internal details of the bitcoin hashing process. How does it compare with the current mining practices? We have study the Stratum protocol in four popular real-life mining configurations. Our analysis shows that pools could very easily cheat the majority of people. However the most subversive versions of the attack are NOT facilitated and could potentially be detected.

1 BITCOIN AND HASH POWER

Bitcoin is a collaborative virtual currency and payment system. It was launched in 2009 (Satoshi08). Bitcoin implements a certain type of peer-to-peer financial cooperative without trusted entities such as traditional financial institutions. Even though it has been in operation for nearly 6 years since early 2009 (Satoshi08) it remains an experimental rather than a mature electronic currency system. The security of bitcoin is based on the idea that bitcoin participants verify the correctness of all bitcoin transactions and include them in 'blocks' of data, which cryptographically authenticate them using a hash function. Network participants which specialize in this task are called miners because they are paid in freshly created bitcoins for this work of cryptographic hashing. However they are only paid if their work is later accepted by the majority of other miners and other participants, which creates strong incentives to be honest.

Many hundreds of millions of dollars have been invested in bitcoin mining in the recent 18 months. In contrast bitcoin adoption has not been such a great success recently. The number of bitcoin peer-to-peer network nodes has been in steady decline and has reached critically low levels, less than 6,000 which is actually also much less than the number of bitcoin miners(!), cf. (Cawrey14; Courtois14). The usage of bitcoin as a currency for ordinary commercial transactions has NOT increased either,

cf. Section 2.5 of (Courtois14).

1.1 Incredible Hash Rate Increase

The combined power of bitcoin mining machines has nearly doubled each month and overall it has been multiplied by an incredible 1000 factor in the last 12 months prior to April 2014, cf. (Courtois14) and in the recent months it was still increasing by at least 50 % month after month, incredibly fast.

A 1000-fold increase in hash power, and further steady increase each month is a very disturbing fact. This is unheard of in new technology business, and clearly much faster than the Moore's law. However this growth is NOT a pure technology improvement curve. This tremendous increase was due to a

T. Courtois N., Emirdag P. and Wang Z..
On Detection of Bitcoin Mining Redirection Attacks.
DOI: 10.5220/0005245600980105
In *Proceedings of the 1st International Conference on Information Systems Security and Privacy* (ICISSP-2015), pages 98-105
ISBN: 978-989-758-081-9

combination of both technology factors and investment/market factors. On the technical side there was a 10,000 times improvement in energy efficiency of bitcoin mining technology cf. (CourGrajNaik14a). On the market size there was an increased interest in bitcoin and a dramatic 10 times or more increase in bitcoin market price at the end of 2013, cf. (Courtois14).

1.2 Hash Power and Security

The following citation seems to reflect a dominant opinion in the bitcoin community. In (Sams14) we read:

> ”The amount of capital collectively burned hashing fixes the capital outlay required of an attacker to obtain enough hashing power to have a meaningful chance of orchestrating a successful double-spend attack on the system

This is basically correct, however. It is NOT true that one needs to match the total effort spend on hashing by the whole planet in order to execute a double-spend attack. One only needs to acquire substantial hash power for a short time (say less than 1 hour) which is easier, and rather through hash power **displacement** than by permanently acquiring some very substantial computing capability.

In general we point out that the security of the bitcoin network does not increase as the combined hash power in bitcoin grows. However high is the hash rate, the cost to compute the next block for the next 10 minutes is approximately 25 bitcoins. We basically expect that miners operate near a financial equilibrium point and the cost is nearly the same as the reward which is 25 bitcoins per block. Now the money at risk in this single block is nowadays typically at least 1,000 bitcoins in each block. Roughly 40 times more. This ratio is likely to grow with time because of the bitcoin monetary policy cf. (CourGrajNaik14a; Courtois14) which will in the future pay miners 12.5 bitcoins, and later even less, while the money at risk can only increase with time if bitcoin is adopted by more people and is more widely used. We see that in bitcoin the risk is likely to increase with time, and the ratio between money at risk and cost of mining does **NOT** depend on the hash rate. Increased hash rate does NOT increase the security.

1.2.1 Bitcoin vs. Bitcoin Clones

However a smaller hash rate means less security for an alt-coin. Smaller crypto currencies are more fragile and some have known a rapid erosion of their security due to a decline in their hash power cf. (Courtois14).

1.2.2 Concentration of Hash Power

There is yet another problem. The hash power can be acquired in a subversive way, by Man-In-The-Middle (MITM) attacks. At this moment we have some 10 large mining pools which control some 75 % of all hash power in bitcoin. These pools also concentrate the discretionary “decision powers” regarding individual bitcoin transactions, cf. Section 5.1 below.

Overall we discover that in current bitcoin the large hash power provides a yet weak and highly unsatisfactory protection against double spending attacks. In this paper we analyse some specific attack scenarios which allow double spending and later analyse if they could really occur with the currently used Stratum protocol.

2 THE LONGEST CHAIN RULE

According to the initial design by Satoshi Nakamoto (Satoshi08) the initial bitcoin system is truly decentralized. It could work in a totally asynchronous way in very poor network propagation conditions. The key underlying principle which allows to achieve this objective is **the Longest Chain Rule** due to Satoshi Nakamoto (Satoshi08):

1. At any moment of the history of bitcoin, miners are trying to extend one existing block, and sometimes two solutions will be found.

   We call this (rare) situation a fork.
2. Different nodes in the network have received one of the versions first and different miners are trying to extend one or the other branch. Both branches are legitimate and the winning branch will be decided later by consensus.
3. The Longest Chain Ruleof (Satoshi08) says that if at any later moment in history one chain becomes longer, all participants should switch to it automatically.

Bitcoin is quite stable in practice. Forks are relatively rare, and wasted branches of depth greater than one are even much less frequent, see Table 1 in (CourtoisBahack14). However forks could become more frequent in poor network conditions or due to attacks, cf. (CourtoisBahack14). It is remarkable that in bitcoin literature this rule is taken for granted without any criticism. The first observation is that this consensus mechanism in bitcoin has two distinct purposes:

1. It is needed in order to decide **which blocks** obtain a monetary reward and resolve potentially arbitrarily complex fork situations in a simple elegant and convincing way.

2. It is also used to decide **which transactions** are accepted and are part of official history, while some other transactions are rejected (and will not even be recorded, some attacks could go on without being noticed, cf. (DeckerWatten14)).

In principle there is NO REASON why the same mechanism should be used to solve both problems. On the contrary. This violates one of the most fundamental principles of security engineering: the principle of *Least Common Mechanism* [Saltzer and Schroeder 1975]. One single solution rarely serves well two distinct problems equally well without any problems.

We need to observe that the transactions are generated at every second. Blocks are generated every 10 minutes. In bitcoin the receiver of money is kept in the state of incertitude for far too long and this for no apparent reason. It is a source of instability which makes people wait for their transactions to be approved for far too long time, especially for larger transactions.

3 IS THE LONGEST CHAIN RULE HELPING THE CRIMINALS AND ARE THERE ALTERNATIVES?

We are now going to show a simple attack which allows double spending. The attack is not very complicated and we do not claim it is entirely new.

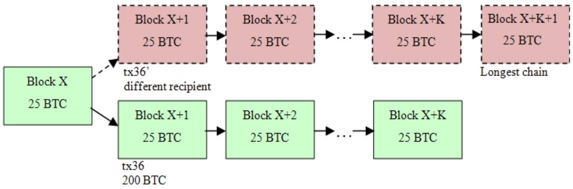

Figure 1: A simple method to commit double spending. The attacker tries to produce the second chain of blocks in order to modify the recipient of some large transaction(s) he has generated himself. Arguably under the right conditions, this is easy to achieve and clearly profitable. The only problem is the timing: to produce these blocks on time requires one to temporarily acquire very substantial computing power such as more than 51 % at the expense of other miners or other crypto currencies.

The attacker produces a fork in order to cancel some transaction[s] by producing a longer chain in a fixed interval of time, see Fig. 1. The attack clearly can be profitable. In contrast the question of actual feasibility of this attack is a complex one, it depends on many factors and these questions are further studied in (Courtois14) and this paper.

Important Remark: The attack does NOT limit to defraud people who would accept a single large payment in exchange of goods or another quantity of a virtual currency (mixing services, exchanges, some sorts of shares). The attacker can in the same way issue a large number of small transactions and cancel all of them simultaneously in the same way.

3.1 Discussion

Our attack could be called a 51 % attack. This name however is very highly misleading. There are many very different things which can be done with 51 % of computing power, for example running a mining cartel attack (CourtoisBahack14) or undoing any subset of past transactions. Very frequently we hear that a 51 % attack require some very powerful entity to ”own” 51 % of bitcoin hash power and therefore it is rather infeasible. In reality there is a much broader range of attacks which are actually feasible in practice. The attacker does not need to own a lot of computing power, he can rent it for a very short time. He could also present himself as a mining pool and cheat a large number of miners to participate in his attack. Calling something a 51 % attack is also misleading because a ratio between the hash rates at two different moments does NOT have to be between 0 and 100 %. Crypto currencies need nowadays to compete with other crypto currencies and hash power can instantly be moved from one crypto currency to another and a 500 % attack is perfectly possible, cf. (Courtois14).

In the following sections we are going to analyse the risks which result from this and similar attacks.

3.2 Feasibility

The most shocking discovery is that anyone can commit such fraud and steal money. The attacker does NOT even need to implement a Man-In-The-Middle (MITM) attack. The redirection of the hash power can be achieved also in another way which does would at all look suspicious. They just need to rent some hashing power from a cloud hashing provider. Bitcoin software does not know a notion of a double spending attack and if it occurs possibly nobody would notice: only transactions in the official dominating branch of the blockchain are recorded in the current bitcoin network, cf. (DeckerWatten14). In a competitive market they do not need to pay a lot for this. Not much more than 25 BTC per block (this is because miners do not mine at a loss, the inherent cost of mining per block should be less than 25 BTC). The attacker just needs to temporarily displace the hashing power from other users or other crypto currencies for a very short period of time. This can be easy to achieve by paying a small premium over the market price. The spare hash power could also be obtained from older miner devices which have been switched off because they are no longer profitable. except for criminals able to generate an additional income from attacks.

3.3 Cross-currency Attacks

There is yet another way to execute such attacks: to offer a large number of miners a small incentive (as a premium over the market price) to go mine for another crypto currency, **before** the attack begins. This can lead to massive displacement of hash power before the attack starts. Then at the moment when block X+1 is mined following the notations of Fig. 1, the double spending attack costs less. Thus we could have something like a 500 % attack, much stronger than the usual 51 % attack, cf. (Courtois14). Further advanced attacks scenarios with malicious pool managers and which can easily be combined with this preliminary displacement of hash power are proposed and studied in Section 4.1.

4 SUBVERSIVE ATTACK SCENARIO H0

We examine the process of double hashing which is used in bitcoin mining according to (CourGrajNaik14a; CourGrajNaik14b).

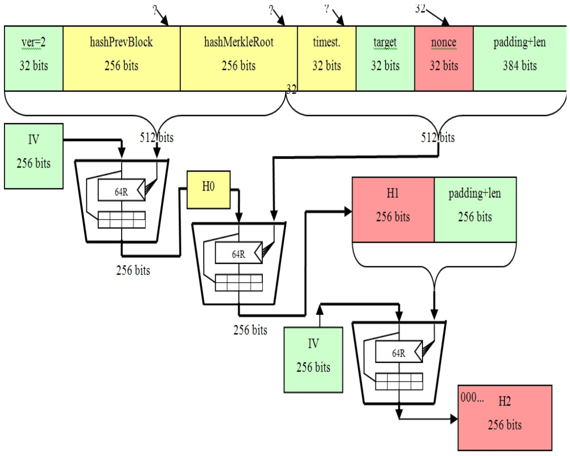

Figure 2: Bitcoin mining internals following (CourGrajNaik14a; CourGrajNaik14b).

One thing jumps to our attention. For every H0, the miner needs to check many possible nonces. The miners do NOT need to know the value of hashPrevBlock. They only need to know the value H0 which changes very slowly and which could be computed for them by the pool manager. Miners could be easily made to mine without any precise knowledge about which block they are mining on. Only an excessively small number of miners, will actually manage to find a winning block. Only these miners might be able to know on which block they have mined as they will see their block appear in the blockchain. However in practice they can see it ONLY if they have also recorded all hundreds of thousands of shares produced by their miner and sent to the pool manager over the weeks and months. We see that pool managers CAN implement arbitrary subversive strategies, for example accept certain transactions only to overthrow them within less than one hour and accept another transaction with another recipient. Nobody will notice: miners will never know that they have been involved in some major attacks against bitcoin such as producing two different versions of the blockchain in order to double spend some large amount of money. Moreover even those miners who have produced winning blocks and therefore will be made aware of the previous block on which they have been mining, still cannot claim they have participated in some sort of attack. Fork events do happen in the bitcoin network.

4.1 Further Variants

The same attack works **across digital currencies**. Miners may think that they mine bitcoin, while in fact they are made to mine Unobtanium, and vice versa. All this is the discretionary power of the pool manager, this is due to the fact that one can mine only knowing H0 and most of the time no other information is disclosed to miners. In rare cases miners could discover that they found a block for another crypto currency which they have never mined. In practice miners do NOT store vast quantities of H0 values with which they have mined. Miner devices do NOT have enough memory to store them.

4.2 Further Manipulation with Deflected Responsibility

Our attack can also be made to work in the scenario in which it is not possible for the attacker to corrupt pool managers. It can be run in a different way in which pool managers are going to corrupt themselves and there will be no reason to accuse them of acting with any sort of malicious or criminal intention. Basically it is possible for an attacker to manipulate the price of a small crypto currency such as Unobtanium to be 10 % MORE profitable[1] than bitcoin mining. Then we can hope that the pool managers themselves are going to implement code to switch to this crypto currency for a short time (real-time switching mechanism mining for the most profitable currency at the moment). Pool manager can now re-direct 100 % of the hashing power they command to another entity. They are NOT going to tell this to miners and simply pocket the difference, and they will still pay miners in bitcoins. Again, in principle miners will probably not notice.

4.3 The Unthinkable Double Spending as a Service

In the bitcoin community there is already a service `bitundo.com` which is trying to convince miners to help to cancel other people's bitcoin transactions on demand. This is done by including a transaction which is a genuine double spend transaction (sending the same money to a different address). It incentivizes miners to help to undo bitcoin transactions for a certain fee which can improve their mining income.

4.4 Can We Fix It?

The general question of potential abuse/redirection of hash power and for-profit blockchain manipulation, which we have explained above, is the central question in this paper. The question whether this could and also whether actually this should be fixed has been discussed in bitcoin forums after our earlier paper on this topic (Courtois14) was first made public:

> " [...] making sure miners know what block they're building on would make certain classes of attack (diverting pool miners to another coin, using pool miners to build an unpublished blockchain, etc) which are currently easy to make undetectably, leave incontrovertible evidence.
> That is a good idea and we should do it."

Source: https://bitcointalk.org/index.php?topic=600436.msg6626004♯msg6626004.

4.5 Solutions

The next question is **how** this problem can be fixed. There are two sorts of solutions to this problem known to us: one method is to make sure that the attacks cannot be executed: strict enforcement of standard protocols and detection of the attack (this paper) or the hash process in bitcoin has to be modified in order to proceed in a different way than on Fig. 2. This leads to the notion of *plaintext aware hashing*, cf. Section 8.8. in (Courtois14). Plaintext aware hashing is an attempt to find a purely cryptographic solution to this problem: a method in which it would be impossible to abuse miners and for miners it will be impossible to claim that they could not know that they have participated in a large scale attack. The key idea is to make final stages of hashing (as opposed to initial stages of hashing only, cf. Fig. 2) on the hash of the previous block in such a complex inextricable way (this is what cryptography is good at) that we can be confident that in order to compute a valid H2 the miner must know hashPrevBlock.

5 CURRENT MINING PRACTICE - STRATUM PROTOCOL

The Stratum algorithm was developed after December 2011 in order to specify a layer above the current bitcoin network, (Palatinus12). The intention behind this protocol was to distribute metadata which are NOT recorded in the blockchain: for example: signed messages about transactions (a potential protection against 51% attacks, wallets which rely on queries from trusted nodes by "light nodes", etc. In practice it primarily used for pooled mining as pools and miners are now distinct entities on the network. Messages are formatted in plain text JSON-RPC (Remote Procedure Call) format. It is a line-based protocol using plain TCP sockets. This protocol became necessary because previous solutions simply did not scale up to allow to mine at higher speeds as implied by ASIC mining cf. (Palatinus12). In absence of alternatives it seems that all the miners worldwide has adopted it.

5.1 The Control Shift

With Stratum miners cannot choose Bitcoin transactions on their own. This according to the designer himself, (Palatinus12). The author explained that in his view "99% of real miners dont care about transaction selection anyway". and the miners only care about the reward. This was a key point in history where bitcoin became more centralized AND miners lost control of what they mine.

We should note that the author has also thought about alternatives, and he wrote:

[1] Typically such currencies are in a sort of equilibrium situation in which the profitability is similar as for bitcoin cf. (Courtois14).

> ”I already have some ideas for Stratum mining protocol extension, where miners will be able to suggest their own merkle branch (I call it internally ”democratic mining”), which will solve such issues as centralized selection of transactions. For now I decided to focus on such a solution, which will fit to majority of miners and do some extensions later.”

This ”democratic” version was never developed by the author, however Stratum is not the only bitcoin mining protocol.

5.2 The GBT Protocol

At the same time as Stratum protocol was developed, a more decentralized solution GBT (GetBlockTemplate) or BIP022/23 was developed and standardized. However Stratum was backed by a major mining pool and GBT adoption suffered.

5.3 The Question of Entropy

In current bitcoin the probability that a random block header is valid is approximately 2^{-66}. However the nonce in Satoshi code has only 32 bits, cf. Fig. 2. Therefore miners must vary other fields in the mined block in order to obtain sufficient entropy. In theory miners could tamper with the ’ntime’ timestamp which is another 32 bits, however this is not sufficient (32+32 bits is less than 66) and it is NOT recommended at all, as this might very seriously confuse the bitcoin self-adjustment mechanism, which makes that the bitcoin clock functions at the right speed and that bitcoins are mined every 10 minutes.

In practice miners vary the so called coinbase transaction, which does not have a well defined role in bitcoin. It can be seen as a way to publish a certain long string of bytes which pertains to each 25 bitcoins mined with the current block. It could be for example used in digital notary services.

5.4 Initial Subscription

In this process a work will be authorized to mine and will receive a unique value of ExtraNonce1 which characterizes this worker and will appear in every coinbase transaction processed by that worker. Assuming that the pool manager is not able to somewhat cheat and attribute the same ExtraNonce1 to several workers, this mechanism allows to know which miners have really mined a particular block. At this stage the size of ExtraNonce2 is also fixed by the pool manager server. It is typically 4 bytes. Together with nonce on 32 bits cf. Fig. 2, this is sufficient in order to span 2^{64} different hashes. Knowing that the timestamp changes every second, unless a miner can do 2^{64} hashes in one second, which would be faster than the whole bitcoin network, this is sufficient for virtually unlimited time (not shorter than until the Unix timestamp overflows).

5.5 Distribution of Work

Work is not directly sent to ASIC but to controllers which typically are tiny PCs of type Rapsberry Pi. Several ASICs can be connected to one controller. This is done on the principle of pushing work from time to time, and getting shares at a constant adjusted rate of one approximate every 3 or more seconds (as observed in real-life situations). We proceed as follows:

1. Work is sent to miner controllers. A prototype block contains all the block data. This prototype contains also a number of Merkle branches which will be hashed together to from a Merkle root.
2. The controller just need to append his ExtraNonce1 (always the same) and a consecutive ExtraNonce2 on 4 bytes.
3. Then the controller can carry on mining for a very long time, without contacting the server.
4. The controller generates the block header and sends it to miners. Each miner tries 2^{32} hashes with all possible values for nonce as in Fig. 2.
5. Typically shares has some 32 zeros only. Depending on the difficulty level for this mining job, for example 512 (a realistic value observed) one share in 512 will have about 41 zeros, as $41 = 32+9$ and $512 = 2^9$.
6. Such shares with about 41 zeros are transmitted to the pool.
7. The difficulty is actually variable and can change for example after 300 seconds. It is adjusted in such a way that one controller sends less than 1 share every few seconds. This minimizes the server load and yet allows to know the amount of work contributed with sufficient precision, so that miners are paid for their work. Miners may lose at most a few seconds of work.
8. At any moment the current work suddenly becomes obsolete. This happens when a new block was mined by anybody in the whole bitcoin network worldwide. The server notifies immediately and sends new work.

When a share is found the miner sends to the pool manager the following data:

1. its identity,
2. the job id
3. the timestamp
4. ExtraNonce2
5. nonce

From this the pool manager needs to recompute the whole block header (all the other data were previously provided by him) and check that it has for example at least 41 leading zeros.

5.6 Attack Feasibility Analysis

In this protocol the exact block used by the miner is entirely determined by the pool manager and the miners can change only a bit more than 8 bytes: ExtraNonce2, nonce, and potentially he could also generate an inaccurate timestamp (not recommended as the pool manager could simply have a policy to reject such shares).

Importantly, this prototype contains also a number of Merkle branches and it appears that all of them must be used and the miner has no choice (at least this was claimed by BTCGuild pool). We have checked 4 prominent pools: Eligius, Bitminter, Ghash and DiscusFish, we have recorded the data exchaned with the remote server when mining with these 4 pools. In all the 4 cases we observed that the miner is given typically up to 11 hashes to form a Merkle tree. Individual hundreds of hashes which form the whole Merkle tree are NOT provided.

However another important element is provided. In all the cases the previous hash is sent in cleartext. If only instead H0 value was sent to the controller, as suggested in Section 4, the possibilities for manipulation would be very substantial. Given the previous hash the miner **can** detect if there is an attack. He needs to record incoming packets with method being ”mining.notify” and check if the second parameter after ”params” is the hash of the last block in the blockchain. However the detection is not easy: Unhappily most miners will not do these checks. This typically requires specialized hardware (a Network Tap) and software (e.g. Wireshark) to sniff network packets. Therefore in practice miners could can still be abused at any moment and maybe the attack, which would be rather of short duration (a small multiple of 10 minutes) would not be detected.

We conclude that there isn’t currently a possibility for fraud such as 51 % attacks which would be impossible to detect without deviating substantially from the widely used Stratum protocol.

6 CONCLUSION

In this paper we have looked at the question of miner attacks with temporary displacement of hash power in bitcoin digital currency. In early 2012 the Stratum protocol has taken a deliberate decision to shift the power of selecting which transactions are included in blocks from miners to pool managers. This decision has stripped miners from the ability to decide on which block they mine. In addition in Section 4 we show that due to the technicalities in the exact hashing mechanism in bitcoin, it is possible to design a malicious mining protocol such that the attack cannot be detected, not even in theory. However our attack remains theoretical. Could the attack work with the exact Stratum protocol as it is used in practice? In this paper we have studied this protocol in details in order to see if this type of attack could or not be implemented in practice. We have analysed data sniffed in the operation of 4 prominent bitcoin mining pools. We have found that Stratum operates in such a way that the attack COULD be detected if only miners do the effort. They would need to permanently monitor the packets exchanged over the network and check if the hash of the previous block is correct.

This does NOT mean that the attack will be detected, as most miners just mine and do not inspect the data exchanged in the network. However the most subversive man-in-the middle attacks such as our attack of Section 4 are NOT facilitated with the current protocol. We recommend that miners should be very careful when mining with proprietary protocols other than Stratum or in non-standard mining configurations as potentially they could be a part of a large scale attack.

In this research we have discovered two quite surprising things. First of all, the security of bitcoin against double-spending attacks is now beyond the scope of the strict open-source system and code created by Satoshi, it depends on additional protocols specified later. When Stratum was specified, it took a critical decision to shift the decision of which transactions are mined to the pools. This is precisely what has led to the current excessive centralization of bitcoin. Consequently bitcoin is no longer exactly a decentralized open-source system. Crucial decisions about the content of the bitcoin blockchain now depend on a number of powerful stake holders with vested interests Bitcoin is no longer a transparent open source system either. The software used by pool managers is not open source, and for example highly subversive forms of a 51 % attack (cf. Section 4) could be executed if some closed-source ASIC controllers would have hidden commands which would allow them to hash with the value of H0 and without knowing the previous block hash. This would facilitate the attacks and it would be impossible for a majority of miners to detect if they are part of an attack. Miners would not even be able to determine which pools are operating honestly, on which crypto currency they are mining, and how their hash power is used.

REFERENCES

Daniel Cawrey: *What Are Bitcoin Nodes and Why Do We Need Them?*, 9 May 2014, http://www.coindesk.com/bitcoin-nodes-need/

Nicolas Courtois, Marek Grajek, Rahul Naik: *The Unreasonable Fundamental Incertitudes Behind Bitcoin Mining,* at http://arxiv.org/abs/1310.7935, 31 Oct 2013.

Nicolas Courtois, Marek Grajek, Rahul Naik: *Optimizing SHA256 in Bitcoin Mining,* in proceedings of CSS 2014, Springer.

Nicolas T. Courtois, Lear Bahack: *On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency,* at http://arxiv.org/abs/1402.1718, 28 January 2014.

Nicolas T. Courtois: *On The Longest Chain Rule and Programmed Self-Destruction of Crypto Currencies*, 20 May 2014, http://arxiv.org/abs/1405.0534.

Christian Decker, Roger Wattenhofer: *Information propagation in the bitcoin network,* 13-th IEEE Conf. on Peer-to-Peer Computing, 2013.

Christian Decker, Roger Wattenhofer: *Bitcoin Transaction Malleability and MtGox,* http://arxiv.org/pdf/1403.6676.pdf

Luke-Jr: getblocktemplate protocol, BIP 022 and BIP023, available from https://en.bitcoin.it/wiki/Getblocktemplate.

Satoshi Nakamoto: *Bitcoin: A Peer-to-Peer Electronic Cash System,* At http://bitcoin.org/bitcoin.pdf

Robert Sams: *The Marginal Cost of Cryptocurrency,* Blog entry at cryptonomics.org, http://cryptonomics.org/2014/01/15/the-marginal-cost-of-cryptocurrency/

Marek (slush) Palatinus, *Stratum mining protocol,* the official documentation of lightweight bitcoin mining protocol, https://mining.bitcoin.cz/stratum-mining, developped in 2011-12. A compact thrid-party description can also be found at https://www.btcguild.com/new_protocol.php.

Meni Rosenfeld: *Mining Pools Reward Methods,* Presentation at Bitcoin 2013 Conference. http://www.youtube.com/watch?v=5sgdD4mGPfg

Technical specification of the bitcoin protocol, 2009-2014, https://en.bitcoin.it/wiki/Protocol_specification