In-store payments using Bitcoin

Mikael Asplund, Jakob Lövhall, Simin Nadjm-Tehrani

Department of Computer and Information Science
Linköping University, Sweden
mikael.asplund@liu.se,jaklo522@student.liu.se,simin-nadjm.tehrani@liu.se

***Abstract*—The possibility of in-store payments would further increase the potential usefulness of cryptocurrencies. However, this would require much faster transaction verification than current solutions provide (one hour for Bitcoin) since customers are likely not prepared to wait a very long time for their purchase to be accepted by a store. We propose a solution for enabling in-store payments with waiting times in the order of a few seconds, which is still compatible with the current Bitcoin protocol. The idea is based on a payment card in combination with a protocol for ensuring that losing a card does not mean losing the money on it. We analyse the required transaction verification delay and also the potentially added risks that the solution brings compared to current systems.**

I. Introduction

As cryptocurrencies are gaining acceptance and being adopted both by large organisations and by ordinary people across the globe it is interesting to consider if there are other areas where they can also be used. In particular the ability to use cryptocurrencies in physical stores would be very valuable as a complement to cash and regular credit card payments.

However, many blockchain-based currency systems today are based on the idea that the longest fork in the chain is the valid one. That means that a powerful attacker with enough hash power can potentially cause the honest nodes to abandon a branch that contains supposedly accepted transactions. To avoid a double-spend attack, the receiver of a payment must wait until the chain has grown enough since the inclusion of the transaction to be sure that it remains there. For Bitcoin, waiting for six blocks correspond to a 0.003% risk of a double-spend attack to be successful if the attacker has 6% of the total hash power [11]. It takes one hour for six blocks to be added to the chain, so this would not be feasible as payment solution in a regular store. Ethereum has a much higher block-rate, but therefore also requires more blocks to achieve the same assurance level. The corresponding waiting time for Ethereum is in the order of 4 minutes [7], which is better, but still not fast enough to compete with regular payment options.

We propose a solution based on a Bitcoin payment card that encapsulates one or multiple key pairs and a client implementation that can sign transactions. Current solutions for Bitcoin payment cards rely on a trusted third party that acts like a bank and interfaces between a bitcoin wallet and the regular money system. In our solution the card together with software at the store is capable of making independent transactions. We assume that the private keys are hidden from the user of the card so that the store knows that payments can only be made with the physical card. This makes it possible to rule out many double-spend attacks since the user has to physically move the card between different stores to make payments, thus removing the need for long waiting times. We analyse the feasibility of this idea by analysing possible attacks and mitigation strategies.

However, the idea with hidden keys in a card creates a major problem for the user if the card is lost or stolen. To avoid this problem we leverage the concept of time-locked transactions to ensure that there is a backup to restore the money if the card is lost. A time-locked transaction includes a time before which the transaction is not valid, and can be invalidated before it is activated by performing a new transaction.

In order to make the backup-scheme resilient both to malicious buyers and malicious store owners, care has to be taken in the design of the backup protocol. We consider two different approaches to enable such backups, one that requires a modification to the existing Bitcoin protocol, and one that is compatible with the current Bitcoin implementation but which makes use of a separate server infrastructure. Moreover, we have implemented a prototype solution based on the separate server approach and measured its timing characteristics to verify that it has sufficiently low latency to be feasible as an in-store payment solution.

Finally, we perform a systematic risk analysis of our proposed approach through our own analysis as well as interviews with an expert panel. The interviews revealed a number of potential problems, some of which must be solved for this approach to be a realistic alternative. We propose potential mitigations for some of these risks, and suggest directions for future work to further analyse and solve remaining issues.

To summarise, the contributions of the paper are:

- Description of a novel Bitcoin-based payment system that uses payment cards with hidden keys together with time-locked transactions.
- Timing analysis of the proposed approach based on experimental results and public Bitcoin data.
- An initial risk analysis of our approach based on interviews with five experts and suggestions for potential mitigations against these risks.

The rest of this paper is organised as follows. Section II presents related studies focusing on Bitcoin-related approaches. The payment protocol is described in Section III, followed by timing analysis in Section IV and risk analysis in Section V. Finally, Section VI conclude the paper.

II. Related work

There have been attempts to create Bitcoin cards before. For example, SpectroCoin[1] cards are tagging on to MasterCard or Visa which makes them usable in most stores and ATMs since the cards work everywhere where MasterCard or Visa works. This provides an agent acting in between the parties to a transaction (similar to escrow) whereas our solution aims to build entirely on a cryptocurrency.

One of the main concerns in the design of our system is the speed of transactions compared to the speed of people moving between payment terminals. The time between sending a message and the time it is received is called propagation delay. To fulfil the requirement that a transaction takes less time between payment terminals than a human, the system needs to have a low propagation delay between the initial transmitter and the other terminals. Bamert et al. [2] propose a concept for fast payments with Bitcoin that relies on transaction propagation speed to protect against double spend attacks. Our approach can be seen as a complementary solution by limiting the possibility of the card owner to issue two simultaneous payments using the same money.

Data gathered by Decker and Wattenhofer [4] shows that in year 2013 blocks propagated to a large portion of the network within a few seconds. In their later work, Decker and Wattenhofer [5] attempt to achieve fast transactions with a system where payments are done outside the blockchain and only using the blockchain when needed. The system presented creates payment channels to do off-blockchain transactions. The problem with known payment channels is that they require a long setup time before they can be used. There is another similar payment channel called Lightning Network [10]. However, being a work in progress, it is only mentioned as a possible alternative. In the longer term perspective Croman et al. [3] capture some of the essential scalability limitations of current blockchain-based currencies and call for radically new designs.

One way of performing a double-spend attack is where the attacker sends transactions to multiple receivers where transactions have overlapping inputs so only one of them will actually be accepted by the Bitcoin network [8]. This way of performing the attack only works if the receivers accept transactions without them being in any block - so called zeroconf transactions. In another way of performing a double-spend attack, the attacker makes a payment, and instantly starts working on an alternative block that does not include the transaction just sent. This is supposed to be very difficult and the chance of succeeding with this rapidly moves towards zero when the number of blocks required by the receiver to accept the transaction increases. However, the success rate for the attacker scales with the fraction of the total hash power that the attacker controls, for example if an attacker controls 40% of the total hash power there is a 49% chance that the double spend attack succeeds even when the receiver waits for six blocks before accepting the payment [11]. Moreover, Eyal and Sirer [6] demonstrate that a group of colluding attackers can gain a disproportionate advantage as long as they have more than 25% of the resources by mining secretly. This is done by revealing their blocks in exactly the right moment to maximise the wasted effort by the honest miners.

All of these works point out the intricacies that govern the security mechanisms and their interaction with timing. We will elaborate on both of these in the next sections. However, our focus is on how to enable card-based Bitcoin transactions, rather than trying to solve challenges in the underlying system.

III. Card-based Bitcoin payment system

In this section we present the proposed payment system. We first provide an overview of the system and present the basic assumptions we make. We then illustrate a basic protocol that does not protect the money associated with the card in case the card is lost or stolen. The rest of the section is devoted to discussing how time locks can be used to make the system resilient to such events through a backup procedure. We analyse two different options for implementing this backup mechanism.

A. Overview

The Bitcoin payment card system presented in this paper builds on the idea that smart card encapsulates one or multiple hidden keys that can be used to sign Bitcoin transactions. Trust in the system relies on the fact that the card owner does not have access to the keys and cannot interfere with the microcontroller logic on the card.

A payment is performed when the user inserts his or her card into a card reader terminal in a store and the terminal proposes to the card a transaction to be performed (including the amount and where the money should go). This requires the store to have a card reader that can handle the Bitcoin card. The payment card creates a new transaction and signs it with one of its private keys. The card has one or more hidden cryptographic keys to use for creating the transactions and claiming transactions sent to it. Multiple keys can be used to strengthen the anonymity of the card owner. The terminal in the store then propagates this transaction to the Bitcoin network, and accepts the payment.

Note that in case an unforeseen fault or attack occurs in the proposed system, there is not much a user can do to reverse its effects. Therefore, this system is intended for smaller purchases where the risks both for the user and the store can be managed. Transactions involving larger amounts require more verification steps and the ability to reverse erroneous transactions.

B. Naive payment card protocol

A naive implementation of the protocol is illustrated in Fig. 1 as a sequence diagram[2]. The card is used to sign transactions to the store which then verifies the payment before broadcasting it to the Bitcoin network.

[1] https://spectrocoin.com/en/

[2] The authors would like to thank Anna Tögel for helping with illustrations.

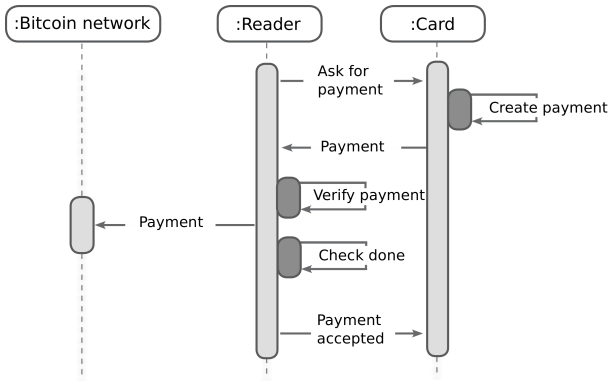

Fig. 1. Sequence diagram for the naive backup protocol.

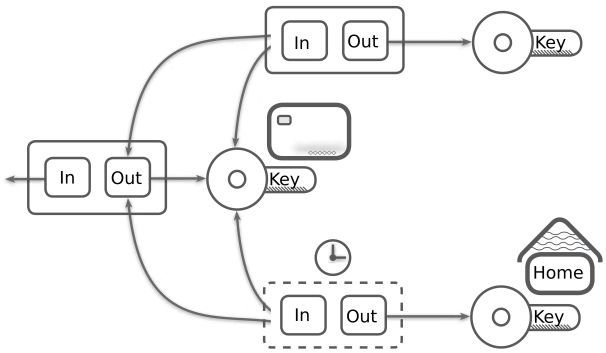

Fig. 2. Backup using a time-locked transaction

The store has several options to define the acceptance condition ("check done" in Fig. 1) at the end of the protocol. One way of doing it is to have a waiting time that is long enough to make sure that the transaction has spread to a sufficiently large fraction of nodes. A complementary mechanism is a warning system in which honest Bitcoin nodes forward double-spending attempts to all nodes. This countermeausure has been proposed and studied by Karame et al. [8]. Technically, this requires changing the behaviour of Bitcoin clients, but this change is compatible with the current policy.

A problem with this solution is that the owner will lose the money on the card if the card is lost, which is why we use time locks to create a backup mechanism.

C. Backups using time locks

Bitcoin transactions the possibility to use a time lock, making the transaction invalid until after the specified time. The time specified in the time lock is an unsigned 4 byte number which is interpreted differently depending on if the number is below or above a threshold of 500 million. If the lock time is less than the threshold, the lock time will be interpreted as block height, meaning that the transaction is valid when the number of blocks in the current longest blockchain in the Bitcoin network is larger than the value specified. Otherwise the lock time will be interpreted as Unix epoch time.

A transaction is considered invalid if any of the outputs that its inputs are pointing to have been used in a prior transaction, which means that the money has already been spent. This can be used to invalidate time-locked transactions by creating a transaction that spends (one or more of) the same output(s) as the time-locked transaction.

We use this mechanism to create a backup of the unspent money on the card. The idea is illustrated in Fig. 2. Every time the owner of the card uses it to make a purchase, the card produces two transactions. The top right transaction in the figure contains the money given to the store, and the remaining amount is put in a time-locked transaction where the recipient is another wallet that the user controls (bottom right in the figure).

If the card is used again before the time lock is released, that transaction will invalidate the time-locked transaction so the money will stay on the card. On the other hand, if the card is lost, the time-locked transaction will become active and make the money available to the owner again.

D. Backup protocol design

We consider two options for implementing the backup procedure, *blockhain backup* and *separate backup*.

In the blockchain backup protocol the card sends a backup transaction in addition to the payment to the store. After making the necessary checks the payment and the backup are broadcast to the network. The payment is sent first because the store does not need to wait for the backup transaction to propagate. The store then waits for a predefined time and listens for possible double spending attacks. If nothing suspicions has been picked up when the waiting time ends, then the purchase is accepted.

In the current Bitcoin protocol time-locked transactions are not stored in the blockchain because they are considered invalid due to the time lock. Thus, the blockchain backup protocol requires a significant change in the Bitcoin specification. It requires time-lock transactions to be globally known so that stores can check if there is any backup transaction which will become valid at the moment of the payment.

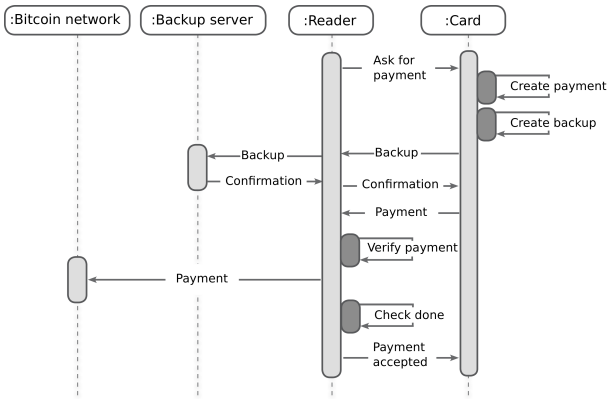

Fig. 3. Sequence diagram for the separate backup protocol.

The separate backup protocol aims to keep the advantage of the backup transactions from the blockchain backup protocol but avoid a major change to the Bitcoin protocol. To enable this, there needs to be a separate subsystem listening for connections and accepting the backups sent to it.

Fig. 3 illustrates the protocol as a sequence diagram. The card sends a backup transaction with an address in addition to the payment to the store. The store sends the backup to the given address, waits for a confirmation to arrive so that it can be forwarded to the card. The card sends the payment to the store after it has received the confirmation from the backup location.

The locking time needs to be globally known because of the risk of double spending when a customer can choose to activate the backup transaction at the same time as a payment is performed. This can for example be accomplished if lock time for the time-locks are set in advance for all cards during manufacturing. If the globally known lock time is T_l and the current time is t_{now}, then a store should not accept payments from a card whose latest payment occurred before $t_{now} - T_l + T_m$, where T_m is the 2h safety margin described in the beginning of this section.

The owner does not need to trust the store to send the backup in this case because it can be sent first and the card could refuse to proceed without getting a confirmation from the receiver of the backup. Trust in this protocol is deferred to the backup server.

Protocol comparison: Table I summarises the protocols discussed in this section. The naive protocol is included as a reference. The table contains five criteria, the transaction overhead, the speed of the payment, the amount of trust required, whether money is lost if the card is lost and whether it is compatible with the current Bitcoin policy.

TABLE I
EVALUATION TABLE FOR THE COMMUNICATION PROTOCOLS.

Protocol	*Naive*	*Blockchain backup*	*Separate backup*
Trans. overhead	Low	High	Medium
Speed of payment	Fast	Fast	Medium
Level of trust	Low	Medium	Medium
Money lost if the card is lost?	Yes	No	No
Compatible w. current Bitcoin?	Yes	No	Yes

The Blockchain backup protocol requires the card to create two transactions for each payment, the store needs to send both, yet more importantly, the Bitcoin network needs to handle and store all these extra backup transactions. The separate backup protocol demands less resources from the Bitcoin network compared to the blockchain backup since the backup is deferred to a separate system. Moreover, the backup transactions which will not be used can simply be removed instead of being a permanent part of the blockchain. Notice that everything in the blockchain is replicated to all Bitcoin nodes, so a transaction that is not needed on the blockchain is worse than a transaction that is not needed on a separate system.

The separate backup protocol is slightly slower due to the added step in communicating with the backup server. Blockchain backup should still be fast as the naive protocol because the store does not need to wait for the backup transaction to have any kind of confirmation, so it can simply broadcast it while waiting for the payment to propagate through the network.

The blockchain backup requires trust in the store (to propagate the backup transaction) and the separate backup protocol requires trust in the backup server. Finally, as already noted, using a protocol for a separate backup system requires less changes to the Bitcoin protocol.

In the rest of this paper we focus on the separate backup protocol as we believe it to be the better of the two alternatives based on the analysis above.

IV. Estimating waiting time

This section describes a timing analysis that we conducted on an implementation of the proposed protocol (using a separate backup mechanism). We are interested in the duration from the time point when a payment is initiated by the store to when it can be accepted (the first and last steps in Fig. 3). For the purpose of this evaluation we consider reaching 50% of the network to be a sufficient proportion for a payment to be accepted (check done).

A. Test setup

We implemented the described protocol using Btcd-cli4j together with Bitcoin Core version 13.1. The tests used a single machine that ran 400 bitcoind instances, referred to as nodes. The number of nodes was chosen as high as possible given the available hardware. This was done to reduce the effects of the trickling behaviour that otherwise dominates the timing characteristics.

Each test starts with giving the Bitcoin nodes time to start up, since they start from scratch each time they need to go through initial setup. Initial blocks are created to amass the funds used in the transactions.

The nodes are instructed to connect to each other to form a connection graph of the form of a 4D torus (with the shape $4 \times 5 \times 5 \times 4$). This layout was chosen as connection graph because it gives a uniform fanout of 8 over all nodes. Yet it still makes it possible to force two nodes to go through a fair number of other nodes to reach each other. According to Decker and Wattenhofer, a node which accepts incoming connections has an average of 32 open connections, and a node which does not accept incoming connections never has more than the default limit of 8 connections [4].

B. Results

Tests were done to collect data about the time it takes for a payment to propagate through the network. Figures 4 and 5 show the proportion of the network reached by different fractions of the transactions. We can see by the S-shaped curves that most payments spread rapidly through the network, but there are a few that take considerably longer time. The

average time to reach 50% of the nodes is just over 4 seconds, 90% of all payments reach 50% in under 6 seconds. However, the slowest 5% of all transactions propagate linearly through the test network taking just over 10 seconds to reach the required 50%.

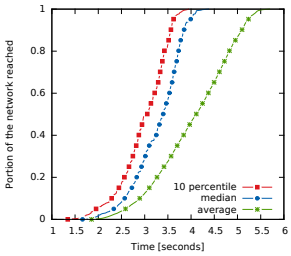

Fig. 4. Payment propagation time, 10th percentile, median and average.

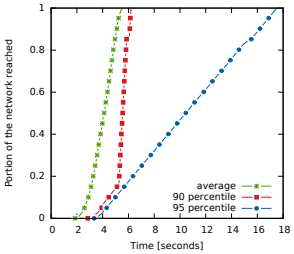

Fig. 5. Payment propagation time, average, 90th and 95th percentile.

We take a closer look at the slow payments by plotting the histograms made on the time it takes for the payment to reach 16% and 100% percent of the network respectively (see Figures 6, and 7 for these histograms). From this, it can be seen that two distinct groups have formed already when reaching 16% of the network (Figure 6). The slow group seems to propagate slower throughout the network, and by the time all nodes have been reached it has taken more than 17 seconds. It is this slow group that forms the linear behaviour seen in the 95th percentile in Figure 5.

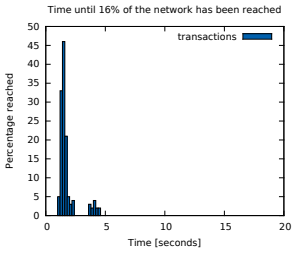

Fig. 6. Distribution of time for payments to reach 16% of the network

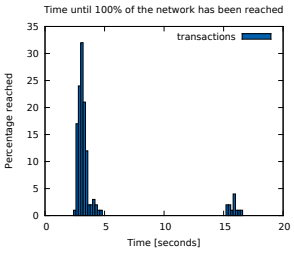

Fig. 7. Distribution of time for payments to reach 100% of the network

These measurements were made in a controlled environment in order to test the performance of our implementation. There are also tests made on the transaction propagation time on the current Bitcoin network. The work by Decker and Wattenhofer [4] has been followed by a website, bitcoinstats.com [1], which tracks statistics for Bitcoin blocks and transactions. Table II shows the transaction propagation times based on data from 2017-04-05.

TABLE II
TRANSACTION PROPAGATION TIME IN THE BITCOIN NETWORK [1].

Portion	50%	75%	90%	95%	99%
Time	3.792s	7.995s	15.048s	22.617s	58.842s

The 50% transaction propagation delay is on par with our testbed measurements of the time to complete a payment, but the slow transactions are even slower. Since the transaction propagation constitutes almost the entire payment delay in our system, these results indicate that as long as reaching 50% provides enough security when using a double-spend detection mechanism [8], the payment delay of the proposed system can be in the order of a few seconds.

V. SECURITY ANALYSIS

The card-based payment system introduces added complexity to the way payments are currently made in Bitcoin, which can potentially bring new vulnerabilities. This section contains a description of a systematic risk analysis that we performed on the system as well as a discussion on how to manage these risks.

A. Attack model

A full security analysis of the entire payment system is out of scope for this paper. The system includes multiple components including the Bitcoin network that in itself has many security issues. Therefore we will assume that the Bitcoin network is controlled by honest nodes and that the network is reachable from the store (not intercepted by malicious entities). Moreover, we assume that the payment card itself is tamper-proof and manufactured by a trustworthy entity.

We consider three potentially malicious actors, the card owner, the store owner, and an external attacker. The relevant system components are the card itself, the card terminal in the store, the backup server and the communication links between these.

B. Risk analysis

To guide the risk analysis of the system we have considered three risk analysis methods from the literature, CORAS, LAVA, and CRAMM. A common factor between these methods is to involve the stakeholders. CORAS was chosen as a base for the analysis since it is the most recent one with a clear intention of incorporating parts from previous methods to create a new modern and general method. We used a subset of the full process to fit with the scope and purpuse of this study (involving steps 4-7). The stakeholders in our case were experts from different organisations to get views on the system from many different stand points. The experts are from the following organizations in alphabetical order, Cinnober (financial technology provider), Linköping University, Nasdaq, Popeller AB (Bitcoin consultancy) and SEB (major Swedish bank). The interviewees' work is related to Bitcoin or blockchain (including an independent researcher at the university). We also made our own security analysis by considering each component separately.

We describe a selected subset of the different risks and threats that were revealed by the interviews (the complete list could not be included due to space restrictions, we refer to Lövhall [9] for a complete listing).

In summary, the risks identified in this process were (excluding those removed by the assumptions such as attacks against the Bitcoin network itself):

- Reduced privacy by connecting purchases made by the same card.
- Intentional or unintentional faults at the Backup server leading to denial-of-service or loss of money if the card is lost or stolen.
- If the terminal is controlled by a malicous entity the user can be tricked to accept a different transaction than he or she believes.
- Failed transactions leading to inconsistencies in the backup.
- Double-spend attacks if a card owner has access to a card terminal.

The double-spend attack can occur if a card owner first makes a payment in a store, and then uses a different terminal to immediately launch a competing transaction. If the new transaction comes with a higher transaction fee it might be included in the next block even if the original transaction indicates that replace-by-fee should not be applied.

C. Mitigation strategies

The identified risks pose real challenges that need to be addressed. We discuss some of these here. Usage of Hierarchical Deterministic Wallets (HD wallets) in the system would mitigate privacy risks. HD wallets are used to generate a deterministic sequence of public and private key pairs and by doing so protect the privacy of the user, reduce exposure of any single key pair and can be used to allow services to generate public keys without knowledge of the private key.

The problems regarding backup inconsistency can be countered by storing historical backups for an extended period of time. The backup server would then send all backups in an ordered fashion to the Bitcoin network. At least one of these backups should be valid.

To reduce the risk of a rogue terminal, the payment procedure could involve the user's phone that would allow the user to inspect the transaction before approving it.

The double spend attack is non-trivial to mitigate. A possible strategy is to ensure that the card would not sign a transaction that competes with its previous transaction. Another way to reduce the risks is to limit the availability of card terminals. Further analysis of this risk is subject to future work.

VI. Conclusions

We have proposed a system that would enable fast card purchases similar to MasterCard and VISA using a cryptocurrency in stores. An important problem is to prevent money loss if a user would lose the debit card. We have shown that time-locked transactions coupled with a backup mechanism is a potential solution to this problem. Having considered two potential backup mechanisms we conclude that a separate backup system is preferable to including backup transactions in the Bitcoin blockchain.

A response time analysis and risk evaluation was also done on the chosen system. The response time analysis indicates that the payment delay would be below 10 seconds, which should be an acceptable delay for in-store payments.

The security analysis was performed using CORAS method of interviewing experts and found a number of problems with mitigations for some of them. Some problems remain, such as the lack of a feedback loop to the store to ensure that the transaction starts to propagate. However, from the perspective of the store, risks need not be completely eliminated as long they can be bounded and the amounts involved are small.

Future work includes a more detailed analysis of unmitigated risks and further evaluation of the impact of mitigations on the timing properties. It would also be interesting to dive deeper in the problems relatd to the energy footprint of the Bitcoin inspired payment systems.

VII. Acknowledgement

This paper has been accepted for publication in Blockchains and Smart Contracts workshop (BSC'2018). This work was partially supported by the Research Centre on Resilient Information and Control Systems (RICS) financed by the Swedish civil contingencies agency (MSB).

References

[1] Bitcoinstats transaction propagation. http://bitcoinstats.com/network/propagation/2017/04/05. Accessed: 2017-04-25.

[2] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer, and S. Welten. Have a snack, pay with bitcoins. In *IEEE P2P 2013 Proceedings*, 2013. doi: 10.1109/P2P.2013.6688717.

[3] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer. *On Scaling Decentralized Blockchains*, pages 106–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016. doi: 10.1007/978-3-662-53357-4˙8.

[4] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P*, 2013. doi: 10.1109/P2P.2013.6688704.

[5] C. Decker and R. Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In *17th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*. Springer, 2015. doi: 10.1007/978-3-319-21741-3˙1.

[6] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *18th International Conference on Financial Cryptography and Data Security (FC)*. Springer, 2014. doi: 10.1007/978-3-662-45472-5˙28.

[7] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016. doi: 10.1145/2976749.2978341.

[8] G. O. Karame, E. Androulaki, and S. Capkun. Double-spending fast payments in bitcoin. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2012. doi: 10.1145/2382196.2382292.

[9] J. Lövhall. Analysis of a Bitcoin debit card - Design of a novel Bitcoin payment system. Master's thesis, Linköping University, 2017.

[10] J. Poon and T. Dryja. The bitcoin lightning network: Scalable off-chain instant payments (white paper). https://lightning.network/lightning-network-paper.pdf, 2016.

[11] M. Rosenfeld. Analysis of hashrate-based double spending. *CoRR*, abs/1402.2009, 2014.