Poster: Modelling Bitcoin Contracts by Timed Automata

Marcin Andrychowicz[†], Stefan Dziembowski[*], Daniel Malinowski[†] and Łukasz Mazurek[†]
Cryptology and Data Security Group
www.crypto.edu.pl
University of Warsaw, Poland
Email: {marcin.andrychowicz, stefan.dziembowski, daniel.malinowski, lukasz.mazurek}@crypto.edu.pl
[*]professor, on leave from *Sapienza* University of Rome [†]student

I. Introduction to Bitcoin

Bitcoin is a digital currency system introduced in 2008 by an anonymous developer using a pseudonym "Satoshi Nakamoto" [15]. Despite its mysterious origins, Bitcoin became the first cryptographic currency that got widely adopted — as of January 2014 the Bitcoin capitalization is over $ 10 bln. The enormous success of Bitcoin was also widely covered by the media (see e.g. [12], [3], [13], [14]) and even attracted the attention of several governing bodies and legislatures, including the US Senate [13]. Bitcoin owes its popularity mostly to the fact that it has no central authority, the transaction fees are very low, and the amount of coins in the circulation is restricted, which in particular means that nobody can "print" money to generate inflation. The financial transactions between the participants are published on a public ledger maintained jointly by the users of the system, which is called the *block chain*.

II. Bitcoin contracts

One of the very interesting, but slightly less known, features of Bitcoin is the fact that it allows for more complicated "transa7ctions" than the simple money transfers between the participants: very informally, in Bitcoin it is possible to "deposit" some amount of money in such a way that it can be claimed only under certain conditions. These conditions are written in the form of *Bitcoin scripts* and in particular may involve some timing constrains. This property allows to create the so-called *contracts* [16], where a number of mutually-distrusting parties engage in a Bitcoin-based protocol to jointly perform some task with financial consequences. The security of the protocol is guaranteed purely by the properties of Bitcoin, and no additional trust assumptions are needed. This Bitcoin feature can have several applications in the digital economy, like creating the assurance contracts, the escrow and dispute mediation, the rapid micropayments [16], the multiparty lotteries [6]. It can also be used to add some extra properties to Bitcoin, like the certification of the users [7], or creating the secure "mixers" whose goal is to enhance the anonymity of the transactions [8]. Their potential has even been noticed by the media (see e.g. a recent enthusiastic article on the *CNN Money* [14]).

This work was supported by the WELCOME/2010-4/2 grant founded within the framework of the EU Innovative Economy (National Cohesion Strategy) Operational Programme.

Despite the plenty of potential applications, the contracts have not been widely used in real life so far. In our opinion, the main reason for that is the fact that the contracts are tricky to write and analyze. As experienced by ourselves [4], [5], [6], developing such contracts is hard due to the distributed nature of the block chain and a huge number of possible interleavings. Moreover, the protocols that involve several parties and the timing constraints are naturally hard to analyze by hand. Therefore, since mistakes in the contracts can be exploited by the malicious parties for their own financial gain, it is natural that users are currently reluctant to use this feature of Bitcoin.

III. Our contribution

We propose an approach that can help designing secure Bitcoin contracts. Our idea is to use the methods originally developed for the computer-aided analysis for hardware and software systems, in particular the timed automata [1], [2]. They seem to be the right tool for this purpose due to the fact that the protocols used in Bitcoin contracts typically have a finite number of states and depend on the notion of time. This time-dependence is actually two-fold, as (1) it takes some time for Bitcoin transactions to appear on the block chain, and (2) Bitcoin transactions can come with a "time lock" which specifies the time when a transaction becomes valid.

We propose a framework for modeling Bitcoin contracts using timed automata. Our method is general and can be used to model almost any contracts. As a proof-of-concept we used this framework to verify the security of two Bitcoin contracts from our previous work [6], [5] in the Uppaal system [9], [10].

IV. Modelling Bitcoin contracts

In our framework the parties (both honest and malicious) are modelled as timed automata, which communicate with each other using shared variables.

The main challenges in modelling Bitcoin contracts as timed automata are (1) modelling the state of the block chain and the peer-to-peer network, (2) modelling the knowledge of the parties, and (3) modelling the behavior of the adversary. The way we face these problems is described below.

A. The block chain

In Bitcoin, whenever a party wants to post a transaction on the block chain she broadcasts it over a peer-to-peer network

and the transactions becomes confirmed (i.e. included in the block chain) after some time, usually about 10 min. We assume that there exists an upper bound on this waiting time (1-2 hours, say). In our models it is captured as follows. We model the block chain as a shared structure denoted `BlockChain`, which keeps information about all transactions broadcast or already confirmed. Moreover, we use a timed automaton denoted `BlockChainAgent` (presented on Fig. 1), which is responsible for maintaining the state of the `BlockChain`. In particular, it is responsible for ensuring that the transactions become confirmed with appropriate time frames. In order to post a transaction an automaton denoting a party simply communicates this fact via the shared `BlockChain` structure.

B. Knowledge of the parties

We need to model the knowledge of the parties (both the honest users and the adversary) in order to be able to decide whether they can perform a specific action in a particular situation (e.g. sign a given transaction). Our representation of knowledge is symbolic and based on Dolev-Yao model [11]. We assume that there is a fixed set of private/public keys and secret strings and for each party we keep information, which of these values are known to her. Moreover, for each party we keep a set of signatures received by her during the execution of the protocol.

C. Adversary

The real-life Bitcoin adversary can create an arbitrary number of transactions with arbitrary output scripts, so it is clear that we need to somehow limit his possibilities, so that the space of possible states is finite and of a reasonable size. We show that without loss of generality we can consider only scenarios in which an adversary broadcasts transactions only from a finite set depending only on the protocol being verified.

Hence, an adversary is modelled as a (nondeterministic) timed automaton, which can broadcast arbitrary transactions from the mentioned set at arbitrary time assuming that his knowledge allows him to do it. He is also allowed to send messages to the other parties and intercept the transactions broadcast in the network.

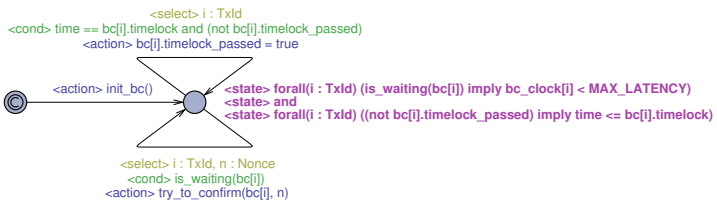

Fig. 1. The `BlockChainAgent` automaton

V. Implementation

As a proof-of-concept we implemented our framework in the Uppaal system and verified the security of the *Bitcoin-based timed commitment scheme* from [6] and a version of the *simultaneous Bitcoin-based timed commitment scheme* from [5]. Both protocols turned out to be secure, but the verification process showed that there was a bug in our first implementation of the latter protocol. Uppaal provides diagnostic traces, which allows to easily find bugs like the one mentioned.

VI. Conclusions

We propose a quite general method of modelling Bitcoin contracts by timed automata, which we used to verify security of the two contracts from the literature in Uppaal. Our experiments confirmed that the computer aided verification and in particular timed automata provides a very good tool for verifying Bitcoin contracts.

References

[1] Rajeev Alur and David L. Dill. Automata for modeling real-time systems. In Mike Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming (ICALP'90)*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer-Verlag, July 1990.

[2] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[3] Marc Andreessen. Why Bitcoin Matters, Jan 2013. The New York Times, dealbook.nytimes.com/2014/01/21/why-bitcoin-matters, accessed on 26.01.2014.

[4] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek. Fair two-party computations via the bitcoin deposits. Cryptology ePrint Archive, Report 2013/837, 2013. http://eprint.iacr.org/2013/837, accepted to the 1st Workshop on Bitcoin Research.

[5] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek. How to deal with malleability of BitCoin transactions. *ArXiv e-prints*, December 2013.

[6] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek. Secure Multiparty Computations on BitCoin. Cryptology ePrint Archive, 2013. http://eprint.iacr.org/2013/784, accepted to the 35th IEEE Symposium on Security and Privacy (Oakland) 2014.

[7] Giuseppe Ateniese, Antonio Faonio, Bernardo Magri, and Breno de Medeiros. Certified bitcoins. Cryptology ePrint Archive, Report 2014/076, 2014. http://eprint.iacr.org/.

[8] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. Bitter to better - how to make bitcoin a better currency. In AngelosD. Keromytis, editor, *Financial Cryptography and Data Security*, volume 7397 of *Lecture Notes in Computer Science*, pages 399–414. Springer Berlin Heidelberg, 2012.

[9] Gerd Behrmann, Re David, and Kim G. Larsen. A tutorial on uppaal 4.0, 2006.

[10] Johan Bengtsson, Kim Guldstrand Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi 0001. Uppaal - a tool suite for automatic verification of real-time systems. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems*, volume 1066 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 1995.

[11] D. Dolev and A. C. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 1983.

[12] The Economist. The Economist explains: How does Bitcoin work?, Apr 2013. www.economist.com/blogs/economist-explains/2013/04/economist-explains-how-does-bitcoin-work, accessed on 26.01.2014.

[13] Timothy B. Lee. Here's how bitcoin charmed washington, Nov 2013. The Washington Post, www.washingtonpost.com/blogs/the-switch/wp/2013/11/21/heres-how-bitcoin-charmed-washington, accessed on 26.01.2014.

[14] David Z. Morris. Bitcoin is not just digital currency. It's Napster for finance, Jan 2014. CNN Money, finance.fortune.cnn.com/2014/01/21/bitcoin-platform, accessed on 26.01.2014.

[15] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[16] Bitcoin wiki. Contracts. en.bitcoin.it/wiki/Contracts, Accessed on 26.01.2014.