Deterministic Witnesses for Claim-First Transactions

Michael Borkowski*, Christoph Ritzer†, Stefan Schulte*

* Distributed Systems Group
TU Wien, Vienna, Austria
{m.borkowski, s.schulte}@infosys.tuwien.ac.at

† Pantos GmbH
Vienna, Austria
contact@pantos.io

***Abstract*—Technologies enabling interoperability across blockchains have recently gained much popularity in science and industry. Cross-blockchain asset transfers are one particular use case which has been proposed to foster such interoperability. In previous work, we have presented the concept of claim-first transactions as a method of transferring assets from one blockchain to another in a decentralized way.**

In this white paper, we address existing challenges of this concept. We present a way of creating cryptographic Proof of Intent for cross-blockchain asset transfers, and propose the concept of deterministic witnesses, a solution for assigning witness rewards in claim-first transactions. This work supports the implementation of cross-blockchain asset transfers using claim-first transactions by providing concrete algorithms.

I. Introduction

In recent years, cryptocurrencies, as well as blockchains, the underlying technology, have gained significant interest in finance and economics, research, and public attention in general [20]. The utility and feasibility of decentralized ledgers in the field of cryptocurrencies has been demonstrated by Bitcoin [14], the first implementation of a blockchain protocol in widespread use. Through its rapid rise in interest and value, Bitcoin also sparked significant investment into research and development related to blockchains and cryptocurrencies. These activities cover, among others, the addition of new layers to Bitcoin itself [18], improvements to the Bitcoin codebase [12], and the development of entirely new blockchains [19]. At the same time, increased attention has been given to use cases for blockchains beyond cryptocurrencies, such as runtime verification for business processes [15].

The research field of blockchains is rich and varied, with an ever-increasing number of technologies and implementations [10]. Despite general positive momentum, however, structural problems exist within the blockchain community. The vast amount of blockchains in existence simultaneously causes severe fragmentation of the research and development field. Interoperability is mostly not foreseen, with blockchains instead competing for users and developers [3].

In our previous work [3, 4], we have provided a conceptual foundation for creating a platform for cross-blockchain interoperability. First, we have introduced fundamental background information about blockchains, smart contracts, and digital assets [3]. Second, we have formalized definitions within the blockchain field, and have presented the cross-blockchain proof problem as well as the lemma of rooted blockchains [4]. Furthermore, we have discussed the concept of claim-first transactions, outlining conceptual and practical challenges of implementing this kind of transactions.

In the work at hand, we present a set of solutions to the challenges of implementing claim-first transactions. The concrete contributions of this paper are as follows:

- We provide an example of a cross-blockchain asset transfer using claim-first transactions.
- We describe a method of generating a cryptographic Proof of Intent (PoI).
- We propose the concept of *deterministic witnesses*, a witness reward model for claim-first transactions.

To this end, Section II discusses background information, based on which Section III shows a method of generating a PoI. Section IV discusses witness selection, and Section V provides a brief overview of related work. Finally, Section VI concludes the paper, and outlines future work.

II. Background

As described in our former work [3], we aim to create a platform for blockchain interoperability, where assets can be moved between blockchains at will, in a decentralized way, and without risking loss of funds due to price fluctuations of cryptocurrencies. The development of such a platform will, in turn, foster connections between various cryptocurrency communities and developers, and enable further innovation within the blockchain space in general [16].

One key goal within this venture is the development of protocols for cross-blockchain token transfers. While *atomic swaps* already provide methods for the exchange of two cryptographic assets on two blockchains in an atomic manner [2, 8], these swaps do not *transfer* assets from one blockchain to another. Instead, they ensure atomicity across blockchains of two otherwise independent transactions. In contrast, we aim to transfer assets from one blockchain to another, i.e., to reduce value on one blockchain, and to increase it on another.

A. Claim-First Transactions

Transferring assets across blockchains poses several significant challenges, as we have discussed in our previous

White Paper, TU Wien; December 2018, version 1.0.

white paper [4]. One such challenge is the cross-blockchain proof problem. We have shown that in practice, it is not possible to prove the presence of a given event or transaction on one blockchain to another blockchain in a non-disputable way, which is manifested formally in the *lemma of rooted blockchains* [4]. To overcome the cross-blockchain proof problem, we have proposed the concept of claim-first transactions.

We propose an ecosystem of blockchains participating in claim-first transactions. In their current form, claim-first transactions require the involved blockchains to have sufficient capabilities to verify certain signatures. This can be achieved using smart contracts [19], a feature present on some, but not all blockchains [3]. However, absence of smart contract functionality does not prohibit claim-first transactions, since other ways of signature verification exist. In its simplest form, a regular transaction signature (e.g., the signature of a Bitcoin transaction), can constitute such a verification. Alternatively, functionality can be added to blockchains using layers added on top of their native protocols. Such a technique is used by CounterParty [5, 6] and OmniLayer [18], which add token functionality to the Bitcoin blockchain. Such layers are transparent to nodes not implementing their functionality, allowing both implementing and non-implementing nodes to participate in the network. Therefore, adding required functionality as an additional layer does not constitute a hard fork, because implementing and non-implementing nodes remain compatible to each other [11].

The essence of claim-first transactions [4] relies on the verifiability of a PoI, certifying that the sender is willing to transfer a given amount of cryptographic assets to a wallet on a (potentially) different blockchain. This PoI can then be used to claim the transferred assets on the destination blockchain, and the claim can subsequently be used on the source blockchain to remove coins from the balances, i.e., to propagate the information about the transfer. An appropriate reward system ensures that observing parties (called *witnesses*) are given sufficient incentive for this propagation. The role of these witness rewards is comparable to mining rewards on lower levels of the blockchain, providing an incentive for otherwise uninterested parties to contribute, thus ensuring correct functionality of the overall system. We discuss witnesses and their significance in claim-first transactions in more detail in Section II-D.

The current design of claim-first transactions requires the balances of each wallet to be stored across all blockchains, not only the blockchain it resides on. Therefore, a transfer between two blockchains must also be propagated to all other blockchains within the entire ecosystem. In the following, we clarify and demonstrate this by showing an example of a claim-first transaction.

B. Exemplary Transaction

For simplicity, we assume that three blockchains exist in the ecosystem, called *Chain X* (CHX), *Chain Y* (CHY), and *Chain Z* (CHZ). In reality, the ecosystem can consist of any number of blockchains, however, we use three blockchains as a minimum number to demonstrate the functionality of claim-first transactions.

TABLE I
INITIAL STATE OF BLOCKCHAINS

Blockchain CHX	Blockchain CHY	Blockchain CHZ
TOK on CHX:	TOK on CHX:	TOK on CHX:
- balance of A: 30	- balance of A: 30	- balance of A: 30
- balance of B: 0	- balance of B: 0	- balance of B: 0
- balance of W: 0	- balance of W: 0	- balance of W: 0
TOK on CHY:	TOK on CHY:	TOK on CHY:
- balance of A: 0	- balance of A: 0	- balance of A: 0
- balance of B: 0	- balance of B: 0	- balance of B: 0
- balance of W: 0	- balance of W: 0	- balance of W: 0
TOK on CHZ:	TOK on CHZ:	TOK on CHZ:
- balance of A: 0	- balance of A: 0	- balance of A: 0
- balance of B: 0	- balance of B: 0	- balance of B: 0
- balance of W: 0	- balance of W: 0	- balance of W: 0

We assume that an asset in form of a token (TOK) exists on multiple blockchains. Currently, this is realized by using a separate smart contract on each participating blockchain. Like most contemporary tokens (e.g., ERC20 [17] tokens), this asset is not coupled to any native currency. Instead, it is an independent asset type, which maintains balances for each wallet independently of their balances of native currency (e.g., Ether or Bitcoin).

We consider three parties (e.g., people), named Alice, Bob, and Wioletta. Alice (A) is sending a token, Bob (B) is receiving it, and Wioletta (W) is acting as a witness. We demonstrate a transfer from blockchain CHX to blockchain CHY. Therefore, we call CHX the *source blockchain* and CHY the *destination blockchain*. Since CHZ is neither the source nor the destination blockchain, we call it a *complementary blockchain*.

As described before, the transfer from CHX and CHY must be propagated to all other (complementary) blockchains within the ecosystem. In our scenario, this remainder of the ecosystem is represented by CHZ for simplicity. Naturally, multiple complementary blockchains can exist, since the claim-first token transfer protocol allows for an arbitrary number of involved blockchains.

According to the previously described requirements, claim-first transactions require the balances of all wallets to be stored across all blockchains. In practice, this means that the TOK balances of all parties on all blockchains are recorded on every blockchain. In other words, the balance of a wallet of tokens on CHX is also stored on CHY and CHZ. To demonstrate this, the initial blockchain state is shown in Table I.

Now, Alice decides to transfer 20 TOK from CHX to Bob on CHY. This means that 20 TOK will be removed from her balance on CHX. For the sake of simplicity, we assume that the reward for this witness will be paid in TOK, and that the amount of reward is fixed to 1 TOK for each participating blockchain, i.e., for each blockchain included within the cross-

TABLE II
STATE OF BLOCKCHAINS AFTER POI CLAIM

Blockchain CHX	Blockchain CHY	Blockchain CHZ
TOK on CHX:	TOK on CHX:	TOK on CHX:
- balance of A: 30	- balance of A: 10	- balance of A: 30
- balance of B: 0	- balance of B: 0	- balance of B: 0
- balance of W: 0	- balance of W: 0	- balance of W: 0
TOK on CHY:	TOK on CHY:	TOK on CHY:
- balance of A: 0	- balance of A: 0	- balance of A: 0
- balance of B: 0	- balance of B: 18	- balance of B: 0
- balance of W: 0	- balance of W: 0	- balance of W: 0
TOK on CHZ:	TOK on CHZ:	TOK on CHZ:
- balance of A: 0	- balance of A: 0	- balance of A: 0
- balance of B: 0	- balance of B: 0	- balance of B: 0
- balance of W: 0	- balance of W: 0	- balance of W: 0

TABLE III
STATE OF BLOCKCHAINS AFTER PROPAGATION OF TRANSFER

Blockchain CHX	Blockchain CHY	Blockchain CHZ
TOK on CHX:	TOK on CHX:	TOK on CHX:
- balance of A: 10	- balance of A: 10	- balance of A: 10
- balance of B: 0	- balance of B: 0	- balance of B: 0
- balance of W: 1	- balance of W: 0	- balance of W: 0
TOK on CHY:	TOK on CHY:	TOK on CHY:
- balance of A: 0	- balance of A: 0	- balance of A: 0
- balance of B: 18	- balance of B: 18	- balance of B: 18
- balance of W: 0	- balance of W: 0	- balance of W: 0
TOK on CHZ:	TOK on CHZ:	TOK on CHZ:
- balance of A: 0	- balance of A: 0	- balance of A: 0
- balance of B: 0	- balance of B: 0	- balance of B: 0
- balance of W: 0	- balance of W: 0	- balance of W: 1

blockchain transfer protocol (not only the two blockchains involved in the current transfer). We will discuss a more sophisticated reward system later in Section IV. Alice creates a PoI describing her intent, and signs it using her private key:

> *Alice intends to transfer 20 TOK from CHX to Bob on CHY.*
>
> *— Signed, Alice.*

She transmits this PoI to Bob, either using an off-chain channel, or on-chain. This channel is not required to be secure, since all information included in this PoI will be made public in the following transactions.

Bob then simply counter-signs this PoI:

> *Alice intends to transfer 20 TOK from CHX to Bob on CHY.*
>
> *— Signed, Alice.*
>
> *Bob accepts this transfer.*
>
> *— Signed, Bob.*

Bob now posts this information on CHY, the blockchain on which he is receiving the 18 TOK (20 TOK reduced by 2 TOK of rewards). This is because Bob now updated CHY with the information about the transfer, but two blockchains (CHX and CHZ) remain unchanged. However, since the transfer must be propagated to all blockchains in order to maintain consistency within the ecosystem, two additional transactions (on CHX and CHZ, respectively) are required. In general, if the ecosystem consists of n blockchains, $n-1$ additional transactions (and therefore, $n-1$ witness rewards) are required. Note that this reward can still be chosen to be relatively small (e.g., 0.1% of the transferred value), and we choose 1 TOK per blockchain for simplicity of demonstration.

Since the PoI posted by Bob is signed, and CHY (that is, its miners) can verify that Alice has sufficient funds for the transfer, this information is now reflected in the balances on CHY, as shown in Table II.

We see that the information about the transfer is now present on CHY (the destination blockchain). However, since nothing has yet been posted on CHX or CHZ, i.e., the source and complementary blockchains, these blockchains still have the initial state, including the balances of Alice, Bob, and Wioletta. The system is currently (temporarily) inconsistent, and the total witness reward of 2 TOK has not been assigned yet.

To ensure the propagation of the transfer information across all participating blockchains even without participation of Alice or Bob (e.g., because they are no longer monitoring any blockchain networks), we use witnesses for this task. Wioletta, witnessing the transfer, aims to receive the witness reward. She therefore signs the following statement:

> *Alice intends to transfer 20 TOK from CHX to Bob on CHY.*
>
> *— Signed, Alice.*
>
> *Bob accepts this transfer.*
>
> *— Signed, Bob.*
>
> *Wioletta witnessed this transfer on CHY.*
>
> *— Signed, Wioletta.*

After signing, Wioletta posts this information on all remaining participating blockchains, in our example, CHX (the source blockchain) and CHZ (the remaining, complementary blockchain). This removes the transferred TOK from Alice's balance as recorded on CHX and CHZ, and propagates the information about Bob receiving TOK to these two blockchains. As a reward, the blockchain increases Wioletta's balance by 1 TOK on the blockchains CHX and CHZ. The balances are now as shown in Table III.

We see that apart from Wioletta's reward, all balances have been correctly propagated across all blockchains. None of the actions have required any blockchain to verify any information on any other blockchain, thus, the cross-blockchain proof problem has been avoided.

However, the problem of inconsistent witness reward remains. CHX contains only Wioletta's reward on CHX, and

CHZ contains only Wioletta's reward on CHZ. Therefore, the reward balances on all chains are inconsistent, and CHY contains no reward at all, i.e., Wioletta's wallet still contains 0 TOK. The propagation of the reward balances, in a way that can be trusted by all blockchains, would require another confirming witness (e.g., confirming to CHY and CHZ that Wioletta has received 1 TOK on CHX), requiring another reward. This leads to a recursive problem. In the following section, we present an alternative solution.

C. Witness Rewards

As we have seen, the distribution of witness reward poses a challenge for claim-first transactions. We identify three general solutions to addressing this challenge:

- Use native reward currency, which is not transferrable across blockchains, and therefore does not require cross-blockchain consistency like the TOK in this example.
- Use a dedicated reward currency, e.g., a separate token. Similar to the first solution, this currency must not require cross-blockchain consistency.
- Use TOK, and find a way of propagating the witness information to all blockchains.

As mentioned in our previous work [4], the first solution, using native reward currency, poses the problem of creating this reward ("out of nothing") on the source blockchain. While this can be realized using currency pools from which rewards are paid, this increases complexity and requires an authority responsible for managing these currency pools. The second solution is viable, but requires the introduction of a separate reward currency, thus again increasing complexity. The third solution poses the challenge of propagating the information about which wallet receives the witness reward to other blockchains (e.g., the destination blockchain). Cross-blockchain proofs are not possible in practice [4], therefore, Wioletta herself has no means of informing CHY that she is the recipient of the witness award in a trustworthy way. For instance, another witness, Malice, who did not receive the reward on CHX because Wioletta was faster to post the witness information on CHX, might have been faster on CHZ. In this case, the blockchain CHZ (i.e., a miner of CHZ blocks) has no way of determining whether Wioletta or Malice is the rightful recipient of the witness reward.

In this paper, we present a way of determining the witness beforehand, thus avoiding the problem of propagating the witness information to the source blockchain. This enables the implementation of the third solution, using the transferred currency (in our example TOK) for the witness reward. We present this method in Section IV.

D. Significance of Witnesses

The role of the witness in claim-first transactions is important, but at the same time, witnesses do not have much possibility for manipulating transactions in a malicious way. Since the PoI contains all information about the transaction and is signed, the witness can only post the entire PoI to a blockchain, or decide to refrain from posting it. In the former case, consistency is ensured through verification of the PoI signature. In the latter case, the witness does not receive the witness reward, but since the reward provides incentive, another party observing the PoI on the destination blockchain can be expected to fulfill the role of the witness.

Therefore, we compare the witnesses in claim-first transactions to miners in a proof-of-work blockchain network, e.g., Ethereum. The miner maintains a pool of transactions signed by their respective authors (addresses), and might choose to deliberately exclude a certain transaction from a mined block. However, due to the number of miners, another miner can be assumed to include the transaction, since mining rewards are involved as an incentive. This incentive must, at a bare minimum, be sufficient to cover the transaction fees paid by the witnesses. In addition, they must give sufficient benefit to be considered as an incentive.

Similarly, the witness rewards proposed in claim-first transactions create an incentive for observers to fulfill the role of witnesses. Naturally, this assumes a sufficient number of observers (potential witnesses). In contrast to the blockchain network, however, a $50\% + 1$ attack on the observing nodes is not sufficient to compromise network consistency (assuming that the blockchain itself is not compromised). Even if Malice controls almost all blockchain nodes, and can therefore exclude certain transactions from the mined blocks, one non-malicious observer alone fulfilling the role of a witness is sufficient to maintain consistency across blockchains.

III. Proof of Intent

In this section, we present a method for creating a signed PoI which can be used to implement claim-first transactions. Throughout the following description, we use the example from Section II, where Alice sends TOK to Bob from CHX to CHY.

The destination blockchain must be able to verify that both Alice and Bob consent to the transmission. For this, we propose the following information to be contained in the PoI:

- The sender (Alice), identified by public key a.
- The receiver (Bob), identified by public key b.
- The source blockchain (CHX), denoted as x.
- The destination blockchain (CHY), denoted as y.
- The amount of tokens transferred, denoted as v.
- The reward in TOK, denoted as r.
- The expiration time of the PoI, denoted as t.

Alice initially signs this information using her private key. We denote the signature as $\alpha = \text{sig}_a(a, b, x, y, v, r, t)$, meaning that Alice (using the private key corresponding to the public key a) signs a data vector containing all the described information. Bob then counter-signs this information, yielding the signature $\beta = \text{sig}_b(a, b, x, y, v, r, t, \alpha)$. This resulting tuple, together with β, constitutes our PoI:

$$(a, b, x, y, v, r, t, \alpha, \beta) \tag{1}$$

In practice, the very fact that Bob submits information to the blockchain (e.g., by calling a smart contract) implies that Bob

is using his private key to sign the transaction. Therefore, the signature β does not have to be an explicit parameter. Instead, Bob's signature (and therefore confirmation of intent to receive the transfer) can be inferred from the blockchain transaction itself.

The signatures themselves can use any digital signature algorithm which provides sufficient cryptographic security. For instance, when implementing claim-first transactions on the Ethereum blockchain, using the Ethereum Virtual Machine (EVM), one might use the ECDSA algorithm [9], natively supported by the EVM.

IV. Deterministic Witnesses

As described in Section II-C, we face the challenge of propagating the information about which wallet is supposed to receive the witness reward. We address this challenge by introducing the concept of so-called *deterministic witnesses*. This allows us to reward the witnesses using the same asset type as is being transferred (here: TOK).

A. Witness Contest

While the initial design of a claim-first transaction protocol as presented in [4] gives witness rewards on a *first-come-first-serve* basis, i.e., the first witness to post the PoI to the source blockchain receives the reward, we now introduce a contest among witness candidates, and a deterministic way of selecting a winner in this context. This deterministic contest ensures that one winning witness is selected from the contestants, and that this selection results in the same witness on all blockchains, without any requirement of information exchange.

We therefore divide the transfer into two steps. First, in the *contest*, potential witnesses – i.e., nodes which have observed the posting of the PoI and want to receive the witness reward for propagating the information – register themselves in a list of contestants. This can be done, for instance, by calling a smart contract function, which records the witness candidate's public key in a list. To ensure proper reception of the witness award, it is in the interest of witness to perform this registration on all blockchains, i.e., the source and destination blockchain as well as all complementary blockchains.

After the expiration of a predefined time window (described by t in Section III), the *winning witness* receives the award. The contest must be completely deterministic across all participating blockchains, and depend only on the information submitted in the witness registration during the contest stage. In Section IV-B, we propose a concrete algorithm for witness selection.

Since it is in the interest of each witness candidate to register on all blockchains taking part in the claim-first transfer protocol, each blockchain will reach the same conclusion when selecting a winning witness. This witness is then automatically assigned the witness reward after the expiration of the time window t.

B. Witness Selection Algorithm

For the witness selection, we aim to achieve a deterministic way of finding the winning witness. For this, we propose to use the PoI signature α and compare it to the public key of each potential witness. We define the winning witness to be the witness with the smallest *distance* between the signature and the public key.

In the following, we define a distance function to compare the PoI signature α with a witness candidate's public key. We note that α might consist of multiple values, as is the case for ECDSA, which in the case of the Ethereum implementation yields three values (r, s, and v). Therefore, we first propose to use a hash function to transform a multivariate signature into a univariate hash. In case of Ethereum, the KECCAK algorithm [1] is a suitable candidate for this hashing. We denote the resulting signature hash as $h_\alpha = \text{KECCAK}(\alpha)$.

We now want to compare the signature hash h_α to the public key of a witness candidate. Both values are scalar numbers, however, they might differ in length in bits. For instance, while the Ethereum implementation of KECCAK returns 256 bits (32 bytes), Ethereum public keys, in their most commonly used representation as addresses, are only 160 bits (20 bytes) long. In fact, Ethereum addresses are defined as the 160 least significant (rightmost) bits of the KECCAK hash of the corresponding ECDSA public key [19].

To compare these two numbers with different lengths, we take the n least significant bits of the longer number, compare them to the shorter number, and take their absolute difference. This difference is defined as the distance of the signature and a witness candidate.

In case of Ethereum, where ECDSA and KECCAK are used, this means that the distance d between a signature α and a witness candidate with address c is defined as follows:

$$d = |\,\text{LSB}_{160}(\text{KECCAK}(\alpha)) - c\,| \tag{2}$$

where $\text{LSB}_{160}(\cdot)$ denotes a number's 160 least significant bits.

Both the address c and the signature hash h_α are obtained by using the KECCAK hash function, which has been shown to have good uniformity [1]. Therefore, we can assume that both c and h_α are uniformly distributed. Since h_α is uniformly distributed, its bits are an independent and identically distributed (i.i.d.) [7] sequence of numbers from $\{0, 1\}$. Since the 160 least significant bits of this sequence are also i.i.d., the property of uniform distribution also holds for the 160 least significant bits. Therefore, the two numbers compared by d are uniformly distributed between 0 and $2^{160} - 1$.

Since we define the winner as the candidate with the lowest distance d of the address to the signature hash, it is in the interest of observers to have an address as close to the signature hash as possible. We assume sufficient difficulty in creating a pre-image or collision for KECCAK [1]. If this was not the case, many aspects of the Ethereum blockchain would be faced with severe problems, since Ethereum relies heavily on KECCAK [19].

The only way for an observer to increase the chances of winning the award is to create many accounts (possibly in advance, i.e., before observing a certain PoI), in hope that one of the accounts has an address close to the signature hash of a given PoI. However, since all witnesses can compete this way, the selection process is still considered fair. Furthermore, since creating numerous accounts is essentially equivalent to the process of mining (iterating through numbers looking for a hash value within a given range), and therefore computationally intensive, there is a break-even point between the invested computational power and the witness reward.

In summary, a witness can increase the chances of winning the contest for a witness reward, however, this comes at a given cost (energy cost for computation), reducing the net benefit.

V. Related Work

In our previous work [3], we have provided a thorough analysis of the current state of the art with regards to cross-blockchain technologies and asset transfers. While atomic swaps [8] have been presented in literature, and first prototypical implementations exist, atomic swaps and cross-blockchain asset transfers are different. Atomic swaps can be used to *exchange* two different assets, with each asset remaining on the respective blockchain, and the atomicity of this exchange guaranteed. In contrast, we aim to allow the *transfer* of assets from one blockchain to another, while maintaining asset value and consistency across blockchain.

As presented in our previous work [4], to the best of our knowledge, Metronome [13] is the only project with similar aims, also targeting transfer of assets across blockchains. It is currently under development, and the technical details of these cross-blockchain asset transfers remain unclear. According to the project documentation [13], users can export assets on one blockchain, gaining a receipt, which can be redeemed for assets which are then imported on another blockchain. These receipts are validated by parties called validators. The validation relies on a number of validators confirming that a given receipt is legitimate, but the precise mechanism of these validations (e.g., how validators are authenticated) is currently not specified.

VI. Conclusion

In this white paper, we have discussed the application of claim-first transactions for cross-blockchain asset transfers using an example. We have provided an algorithm for creating a cryptographically verifiable PoI. Furthermore, we have presented the concept of deterministic witnesses for claim-first transactions to maintain consistency of witness information across blockchains, thus allowing the payment of witness rewards using the transferred asset itself.

Disclaimer

Information provided in this paper is the result of research, partly based on publicly available resources of varying quality. Popular use of cryptocurrencies includes investment and speculation on price developments of currencies and assets. The goal of this paper is to describe technical aspects relevant for the Token Atomic Swap Technology (TAST) research project[1]. Economic considerations or future price developments are therefore not discussed. Technologies are described from a purely technical point of view. Therefore, the information in this paper is provided for general information purposes only and is not intended to provide advice, information, predictions, or recommendations for any investment. We do not accept any responsibility and expressly disclaim liability with respect to reliance on information or opinions published in this paper and from actions taken or not taken on the basis of its contents.

Acknowledgment

The work presented in this paper has received funding from Pantos GmbH within the TAST research project.

References

[1] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. *KECCAK sponge function family main document*. Accessed 2018-11-30. 2009. URL: https://keccak.team/obsolete/Keccak-main-1.0.pdf. Submission to NIST.

[2] *Bitcoin Forum: Atomic swaps using cut and choose*. 2016. URL: https://bitcointalk.org/index.php?topic=1364951. Forum Postings between 2016-02-14 and 2016-02-19. Accessed 2018-04-19.

[3] M. Borkowski, D. McDonald, C. Ritzer, and S. Schulte. *Towards Atomic Cross-Chain Token Transfers: State of the Art and Open Questions within TAST*. 2018. URL: http://dsg.tuwien.ac.at/staff/mborkowski/pub/tast/tast-white-paper-1.pdf. White Paper, Technische Universität Wien. Version 1.2. Accessed 2018-11-12.

[4] M. Borkowski, C. Ritzer, D. McDonald, and S. Schulte. *Caught in Chains: Claim-First Transactions for Cross-Blockchain Asset Transfers*. 2018. URL: http://dsg.tuwien.ac.at/staff/mborkowski/pub/tast/tast-white-paper-2.pdf. White Paper, Technische Universität Wien. Version 1.0. Accessed 2018-11-12.

[5] Counterparty. *Counterparty*. URL: https://counterparty.io/docs/. Website. Accessed 2018-04-13.

[6] Counterparty. *Counterparty Protocol Specification*. 2017. URL: https://github.com/CounterpartyXCP/Documentation/blob/master/Developers/protocol_specification.md. White Paper. Accessed 2018-04-13.

[7] B. Gnedenko and A. Kolmogorov. *Independent Random Variables*. Cambridge, Massachusetts: Addison-Wesley, 1954.

[8] M. Herlihy. *Atomic Cross-Chain Swaps*. 2018. URL: http://arxiv.org/abs/1801.09515. White Paper. Accessed 2018-04-13.

[9] D. Johnson, A. Menezes, and S. Vanstone. "The Elliptic Curve Digital Signature Algorithm (ECDSA)". In: *International Journal of Information Security* 1.1 (2001), pp. 36–63.

[1] http://www.infosys.tuwien.ac.at/tast/

[10] I. Konstantinidis, G. Siaminos, C. Timplalexis, P. Zervas, V. Peristeras, and S. Decker. "Blockchain for Business Applications: A Systematic Literature Review". In: *Business Information Systems*. Ed. by W. Abramowicz and A. Paschke. Cham: Springer, 2018, pp. 384–399. ISBN: 978-3-319-93931-5.

[11] I.-C. Lin and T.-C. Liao. "A Survey of Blockchain Security Issues and Challenges". In: *IJ Network Security* 19.5 (2017), pp. 653–659.

[12] *Litecoin*. URL: https://litecoin.org/. Website. Accessed 2018-11-13.

[13] *Metronome: Owner's Manual*. URL: https://www.metronome.io/pdf/owners_manual.pdf. White Paper. Version 0.967, 2018-04-17. Accessed 2018-11-19.

[14] S. Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. 2008. White Paper.

[15] C. Prybila, S. Schulte, C. Hochreiner, and I. Weber. "Runtime verification for business processes utilizing the Bitcoin blockchain". In: *Future Generation Computer Systems* (2018).

[16] S. Underwood. "Blockchain Beyond Bitcoin". In: *Communications of the ACM* 59.11 (Oct. 2016), pp. 15–17. ISSN: 0001-0782.

[17] F. Vogelsteller. *Token standard*. 2015. URL: https://github.com/ethereum/EIPs/issues/20. GitHub Issue. Accessed 2018-04-13.

[18] J. Willett, M. Hidskes, D. Johnston, R. Gross, and M. Schneider. *Omni Protocol Specification*. 2017. URL: https://github.com/OmniLayer/spec. Version 0.5. Accessed 2018-11-13.

[19] G. Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. 2018. URL: https://ethereum.github.io/yellowpaper/paper.pdf. White Paper. Accessed 2018-11-13.

[20] A. Zohar. "Bitcoin: Under the Hood". In: *Communications of the ACM* 58.9 (2015), pp. 104–113.