Advanced Cross-Chain Token Transfers

Philipp Frauenthaler*, Marten Sigwart*, Christof Spanring†, Stefan Schulte*

* Distributed Systems Group
TU Wien, Vienna, Austria
{p.frauenthaler, m.sigwart,
s.schulte}@dsg.tuwien.ac.at

† Pantos GmbH
Vienna, Austria
contact@pantos.io

***Abstract*—Known implementations for cross-blockchain token transfer do not provide mechanisms that prevent tokens from getting lost in transit. Ideally, such finalization mechanisms do not rely on a centralized authority in order to avoid undermining the blockchain's general principle of decentralization.**

In this paper, we present extensions for a cross-blockchain token transfer protocol to ensure that tokens in transit are eventually recreated on the destination blockchain and to provide transfer confirmations on both blockchains involved in the transfer in a fully decentralized manner. Further, we evaluate the proposed protocol extensions in terms of transfer cost and duration by transferring ERC20 tokens from Rinkeby to Ropsten.

I. Introduction

The Token Atomic Swap Technology (TAST) research project[1] aims to create a platform for cross-blockchain interoperability. The overarching goal is to investigate possible means of interconnecting various blockchains [1]. As an important step towards achieving this goal, we aim to create a cross-blockchain token, i.e., a token that can be freely exchanged between various blockchains [3].

So far, tokens that can be transferred to other blockchains either rely on centralized entities coordinating the exchange [6] or a user needs to find another party willing to swap tokens, e.g., via atomic swaps [5]. However, within TAST we aim to provide a token that can be transferred between blockchains in a decentralized manner without having to swap tokens with another party.

A cross-blockchain transfer occurs when burning a certain amount of tokens on the source blockchain and then recreating the same amount of tokens on the destination blockchain [3]. Of course, the tokens should only be recreated on the destination blockchain if the burning of the tokens has actually occurred on the source blockchain [3]. Hence, the destination blockchain needs a way to verify the existence of the transaction burning tokens on the source blockchain.

One possibility to verify transaction inclusions across blockchains are so-called blockchain relays [2]. Relays replicate the state of a source blockchain within a destination blockchain and as such enable the destination blockchain to verify the existence of certain pieces of state on the source blockchain. The replication of the source blockchain happens in a completely decentralized way and consequently does not require trust in a centralized entity [2]. Using a blockchain relay, it becomes possible to verify on the destination blockchain that a transaction burning some tokens has occurred on the source blockchain [7].

In the last White Paper [4], we presented a protocol that leverages blockchain relays to enable such decentralized cross-blockchain token transfers as envisioned by TAST. However, the existing protocol suffers from a couple of limitations. First, the protocol does not guarantee transfer finality. That is, the protocol does not ensure that tokens that are burned on the source blockchain are eventually recreated on the destination blockchain—tokens can get lost in transit. Second, even if transfer finality was guaranteed, the source blockchain currently would have no possibility to know when a transfer has actually been finalized on the destination blockchain. This is important in case certain actions need to take place on the source blockchain as a result of a successful transfer.

To tackle these issues, the work at hand presents two extensions to the protocol. The first extension enables transfer finality. It remains fully decentralized by deploying a sophisticated incentive scheme. The second extension provides functionality to report back successful transfer finalizations to the source blockchain. Further, we evaluate the extended protocol in terms of transfer cost and duration by transferring ERC20 tokens from Rinkeby to Ropsten.

II. Recap: White Paper VIII

In White Paper VIII [4], we describe a protocol for realizing cross-blockchain token transfers based on the cost-efficient blockchain relay that has been developed within TAST.

To recall, a blockchain relay is operated by off-chain clients who continuously submit block headers from a source blockchain to a destination blockchain. With the source blockchain essentially being replicated within the destination blockchain, it becomes possible from within the destination blockchain to make queries such as *"Is a certain transaction tx included in and confirmed by the source blockchain?"*. Whenever such a query is requested, off-chain clients need to submit the transaction's Merkle proof of Membership [3] to the blockchain relay. As this proof is verified on-chain, no trust in the client submitting the Merkle proof is required.

In White Paper VIII, we show how the functionality of blockchain relays can be leveraged to realize cross-blockchain token transfers. Consider the case in which Alice wants to transfer tokens from the source blockchain to the destination

White Paper, TU Wien; May 2020, version 1.0.

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

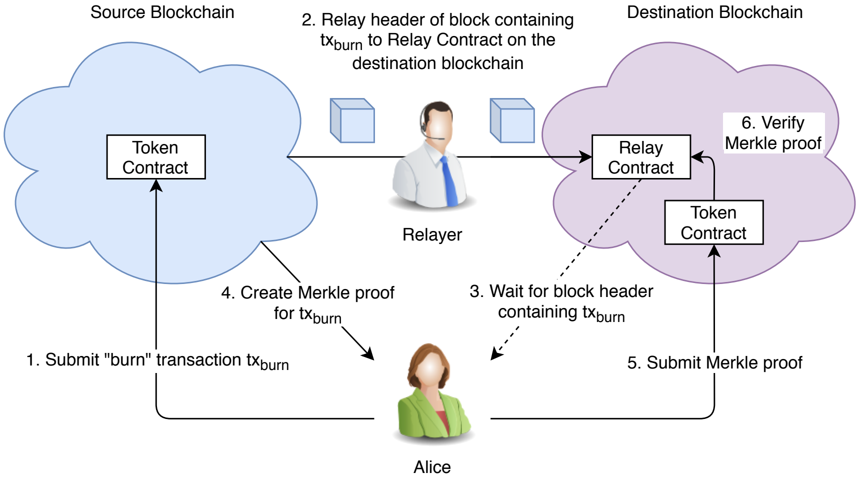
Source Blockchain
2. Relay header of block containing tx_{burn} to Relay Contract on the destination blockchain
Destination Blockchain
Token Contract
Relay Contract
6. Verify Merkle proof
Relayer
Token Contract
4. Create Merkle proof for tx_{burn}
3. Wait for block header containing tx_{burn}
1. Submit "burn" transaction tx_{burn}
5. Submit Merkle proof
Alice

Figure 1: Cross-blockchain Token Transfer with Blockchain Relay

blockchain (Fig. 1). First, Alice burns some amount of her tokens by submitting a BURN transaction tx_{BURN} invoking the smart contract responsible for managing the tokens on the source blockchain (token contract). When being invoked, the contract reduces the balance of Alice. Eventually, tx_{BURN} is included in a new block and appended to the source blockchain. When this happens, off-chain clients (so-called relayers) forward the header of the new block to the blockchain relay contract running on the destination blockchain. Once the relay contract is aware of the new block header, Alice creates the Merkle proof of Membership for tx_{BURN} and submits it via a CLAIM transaction to the smart contract authorized for managing tokens on the destination blockchain (token contract). When being invoked, the contract forwards the provided Merkle proof to the relay contract. If the relay contract confirms the inclusion of tx_{BURN} within the source blockchain, the token contract recreates the tokens on the destination blockchain, i.e., the balance of Alice on the destination blockchain is increased by the burned amount. Otherwise, the CLAIM transaction is rejected.

As such, a cross-chain token transfer essentially involves the burning of tokens on the source blockchain and their recreation on the destination blockchain. Further, to ensure that no tokens can be burned or claimed illegally, protocols for cross-chain token transfers must adhere to the following requirements [8]:

(1) *Ownership:* Only the rightful owner of tokens can initiate their transfer.
(2) *No Claim Without Burn:* Each CLAIM needs to reference a valid BURN.
(3) *Double Spend Prevention:* Each BURN can be claimed at most once.
(4) *Decentralized Finality:* Each burned token is eventually claimed on another blockchain. This finalization process should not depend on a central authority.
(5) *Transfer Confirmation:* For any cross-blockchain transfer, the source blockchain is eventually informed of the success of the transfer.

III. Protocol Extensions

The protocol presented in [4] fulfills requirements (1) to (3). However, it lacks with regards to requirements (4) and (5). Hence, it is possible that tokens are burned on the source blockchain without ever being recreated on the destination blockchain, reducing the total supply of tokens over time. Further, even if the transfer is eventually finalized on the destination blockchain, the source blockchain never retrieves a finalization confirmation. It can never be certain that a transfer was actually finalized. To circumvent these issues, we augmented the protocol by two extensions which we also present in a scientific publication [8]. This section provides a concise summary of the two extensions.

A. Decentralized Finality

As mentioned above, it is important that protocols for cross-blockchain token transfers ensure that tokens are eventually recreated on the destination blockchain. Notably, finalization should not depend on a single centralized actor. For instance, even if Alice is indisposed to finalize a transfer, finalization should take place regardless.

To allow this, we add a finality period t to transfers. This period starts when tx_{BURN} is included in the source blockchain. If the burned tokens are not claimed by Alice within the finality period t, any user can post the claim to finalize the transfer. To encourage other users to post the claim in case Alice is indisposed to do so, the user submitting the claim gets a fraction of the transferred tokens as reward. The remaining tokens are still transferred to the account of Alice.

Thus, before recreating the tokens, the token contract on the destination blockchain needs to know whether t has already elapsed. For that, the token contract leverages the blockchain relay by sending a query whether the block containing tx_{BURN} is confirmed by at least t succeeding blocks. If the relay confirms that this block has at least t successors, the period t is considered elapsed, otherwise not.

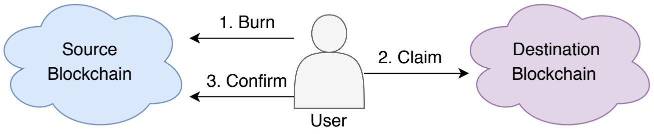

Figure 2: Confirmation of the Transfer Finalization

As users incur cost when finalizing transfers (e.g., transaction fees), the reward must be high enough to compensate these cost. The described concept guarantees transfer finality as long as at least one user follows the protocol [8].

B. Transfer Confirmation

While the above extension ensures the recreation of tokens on the destination blockchain, the source blockchain does not learn about the finalization of the transfer. This may be essential in case the source blockchain needs to perform some actions (e.g., transferring ownership) if a certain cross-blockchain token transfer has been executed successfully (i.e., the tokens have been successfully recreated on the destination blockchain). To allow the source blockchain to react to the finalization of a transfer, we implement a further extension. This extension augments the protocol by a third kind of transaction (CONFIRM) that can be used to report the successful token transfer back to the source blockchain.

The steps introduced by this protocol extension are outlined in Fig. 2. As shown in the figure, after claiming the tokens on the destination blockchain, users can post a CONFIRM transaction to the source blockchain. This third transaction type enables the source blockchain to trigger further actions (e.g., transfer of ownership) on the basis of a successful transfer finalization.

However, analogous to CLAIM transactions, the user posting the CONFIRM transaction on the source blockchain incurs cost. To reward users for their service, we introduce an incentive similar to that used for transfer finalization. If the user that burned the tokens on the source blockchain (in our example Alice) does not confirm the finalization of the transfer within a time period c, any user can post the CONFIRM transaction. To ensure that users confirming the finalization on the source blockchain get a reward, Alice has to provide some stake when she first burns the tokens. This stake is locked on the source blockchain for the duration of the time period c. If Alice confirms the finalization of the transfer within c, she gets back control of the locked stake. If not, any user posting the CONFIRM transaction can get the locked stake. This incentivises users to confirm the transfer finalization on the source blockchain. As long as at least one user is honest (i.e., they follow the protocol rules), it is guaranteed that the successful execution of a transfer is reported back to the source blockchain [8].

With the presented extensions in place, the initial protocol now fulfills all defined requirements for cross-blockchain token transfers. In the following section, we present results obtained from a comprehensive evaluation of the extended protocol.

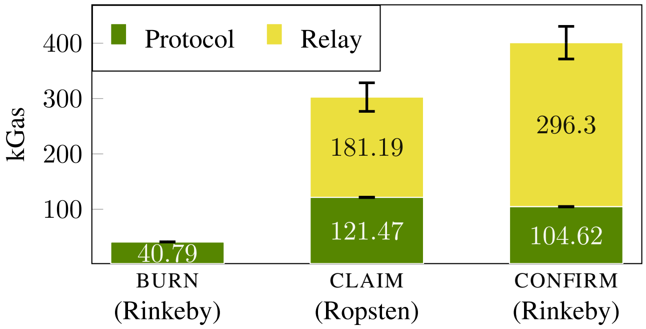

Figure 3: Avg. Transaction Gas Consumption [8]

IV. Evaluation

We analyze the extended protocol regarding the defined requirements, transfer cost and duration. A comprehensive presentation of the obtained results as well as a detailed protocol specification is provided in the corresponding scientific publication [8]. In the following section, we provide a concise summary of the obtained results.

To get an estimation of transfer cost and duration, we conduct cross-blockchain asset transfers between the public Ethereum test networks Rinkeby and Ropsten. In particular, we perform 500 transfers of 1 ERC20 token from Rinkeby to Ropsten. For simplicity reasons, as the second protocol extension (Section III-B) also includes the steps of the first extension (Section III-A), we only use the implementation of the second extension for our experiment. The corresponding smart contract has been deployed on both Rinkeby and Ropsten.

A. Transfer Cost

For every performed transfer, we measure the gas consumption of all three transaction types (BURN, CLAIM, and CONFIRM). The obtained results are outlined in Fig. 3. Note that the figure contains the gas consumption for the protocol as well as the gas consumption of the blockchain relay used for verifying the inclusion of transactions.

The total gas consumption of the first protocol extension (see Section III-A) is about 343.5 kGas (standard deviation 25.81 kGas), calculated as the sum of the BURN and CLAIM transactions. The total gas consumption of the second extension (see Section III-B) is about 744.4 kGas (standard deviation 46.01 kGas) as it additionally includes the gas consumption of the CONFIRM transaction. With an exemplary exchange rate of about 130.10 EUR[2] per ETH and a gas price of 1.5 GWei[3], this results in transfer cost of about 0.07, and 0.15 EUR for the first and second protocol extension, respectively. Notably, the transfer cost strongly depends on the used blockchain relay. In case other mechanisms for verifying transaction inclusions are used, the cost may change.

[2] 02.04.2020
[3] 26.03.2020

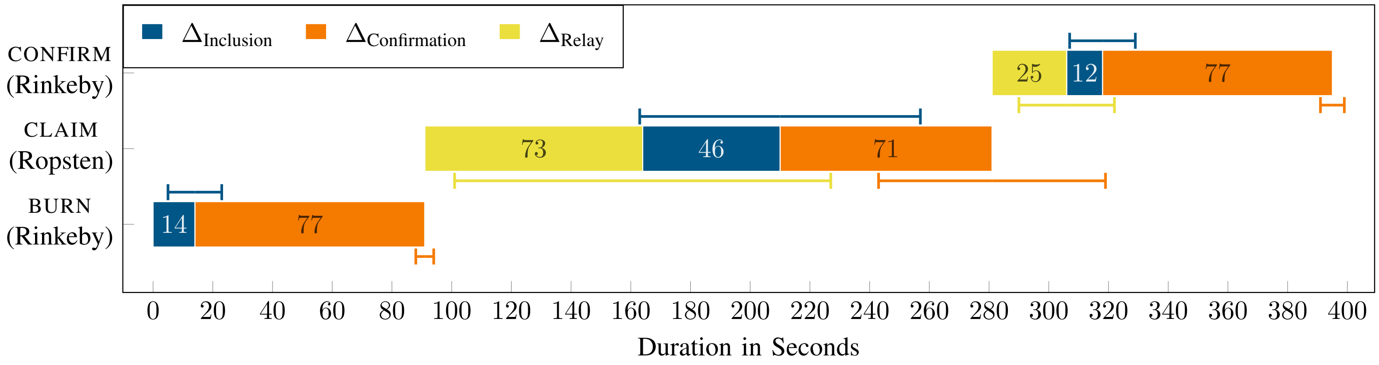

Figure 4: Avg. Transaction Durations [8]*
* thin bars depict the standard deviation

B. Transfer Duration

In this section, we analyze the minimum transfer duration. For that, we submit each transaction at the earliest possible time, i.e., as soon as the preceding transaction is included in the blockchain and confirmed by enough blocks. Further, we require each transaction on Rinkeby as well as on Ropsten to be confirmed by at least 5 succeeding blocks. Both blockchains have an inter-block time of approximately 15 seconds.

As tokens are sent from Rinkeby to Ropsten, BURN and CONFIRM transactions are submitted to Rinkeby while CLAIM transactions are submitted to Ropsten. Hence, durations for BURN and CONFIRM transactions are measured on Rinkeby whereas for CLAIM transactions on Ropsten.

Essentially, CLAIM (CONFIRM) transactions can be submitted to Ropsten (Rinkeby) as soon as the corresponding BURN (CLAIM) transactions are included and confirmed on Rinkeby (Ropsten). However, users need to wait until the relay running on Ropsten (Rinkeby) has been brought up to date before they can submit the corresponding CLAIM (CONFIRM) transactions. Otherwise, the transactions would not be successful as the relay does not have enough information to verify the inclusion of the transactions yet. To this end, $\Delta_{\text{Inclusion}}$ denotes the duration from the moment a transaction is submitted to Rinkeby (Ropsten) until it is included in some block, $\Delta_{\text{Confirmation}}$ specifies the time it takes for an already included transaction to be confirmed by enough succeeding blocks, and Δ_{Relay} denotes the time it takes for the relay to collect enough information to be able to verify the inclusion of the transaction.

Figure 4 shows the average duration for each transaction type as well as for both extensions. With an average duration of 91 seconds (standard deviation of 9 seconds), BURN transactions clearly achieve the smallest duration, followed by CONFIRM transactions (average duration of 114 seconds, standard deviation of 22 seconds) followed by CLAIM transaction (average duration of 191 seconds, standard deviation of 103 seconds). As shown in the figure, the durations of CLAIM and CONFIRM transactions strongly depend on the used blockchain relay. If other mechanisms for verifying transaction inclusions are used, the durations of these transaction types may change.

The total duration of the first protocol extension is calculated by summing up the durations of BURN and CLAIM transactions, while the total duration of the second extension also contains the duration of CONFIRM transactions. This yields an average transfer duration of 282 seconds (standard deviation of 103 seconds) and 395 seconds (standard deviation of 106 seconds), respectively. Transfers with the second protocol extension clearly take longer as an additional transaction is required.

V. Conclusion

In the last White Paper, we presented a protocol that realizes cross-blockchain token transfers by leveraging blockchain relays. Despite enabling decentralized token transfers as envisioned by TAST, the protocol neglects important requirements such as decentralized finality and transfer confirmations. These requirements need to be fulfilled to ensure that tokens are not lost in transit.

In this paper, we described two extensions of the protocol that take these requirements into account. First, we added an incentive structure encouraging any user to finalize token transfers in case the sender is indisposed to do so. Second, functionality for reporting transfer finalization back to the source blockchain was implemented. This feature allows the source blockchain to take further actions on the basis of successful token transfers (e.g., transfer of ownership rights if the tokens are successfully sent to the receiver on the destination blockchain). We evaluated the extended protocol regarding the defined requirements, transfer cost and duration in a scientific publication which also contains detailed specifications of the described extensions. In the work at hand, we presented a concise summary of the obtained results.

The proof of concept implementation used for the evaluation is currently restricted to Ethereum-based blockchains. In future work, the approach will be extended to other blockchain platforms as well. Finally, we will explore blockchain interoperability solutions beyond cross-blockchain token transfers.

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 TAST research project. 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] M. Borkowski et al. *Cross Blockchain Technologies: Review, State of the Art, and Outlook*. 2019. URL: http://dsg.tuwien.ac.at/projects/tast/pub/tast-white-paper-4.pdf. White Paper, Technische Universität Wien. Version 1.0. Accessed 2020-02-06.

[2] V. Buterin. *Chain Interoperability*. https://allquantor.at/blockchainbib/pdf/vitalik2016chain.pdf. Accessed 2020-02-10.

[3] P. Frauenthaler et al. *Towards Efficient Cross-Blockchain Token Transfers*. 2019. URL: http://dsg.tuwien.ac.at/projects/tast/pub/tast-white-paper-5.pdf. White Paper, Technische Universität Wien. Version 1.0. Accessed 2020-02-06.

[4] P. Frauenthaler et al. *Leveraging Blockchain Relays for Cross-Chain Token Transfers*. 2020. URL: https://www.dsg.tuwien.ac.at/projects/tast/pub/tast-white-paper-8.pdf. White Paper, Technische Universität Wien. Version 1.0. Accessed 2020-04-20.

[5] M. Herlihy. “Atomic cross-chain swaps”. In: *Proceedings of the 2018 ACM symposium on Principles of Distributed Computing*. 2018, pp. 245–254.

[6] Loom Network. *Transfer Gateway*. https://loomx.io/developers/docs/en/transfer-gateway.html. Accessed 2020-02-10.

[7] M. Sigwart et al. *Preparing Simplified Payment Verifications for Cross-Blockchain Token Transfers*. 2019. URL: http://dsg.tuwien.ac.at/projects/tast/pub/tast-white-paper-7.pdf. White Paper, Technische Universität Wien. Version 1.0. Accessed 2020-02-06.

[8] M. Sigwart et al. *Decentralized Cross-Blockchain Asset Transfers*. 2020. URL: https://arxiv.org/abs/2004.10488.