Threshold Signature in Off-Chain Components to Manage Inter-chain Transactions

Alessandro Bigiotti
Division of Computer Science
University of Camerino
Camerino, 62032, Italy
Email: alessandro.bigiotti@unicam.it

Leonardo Mostarda
and Alfredo Navarra
Department of Mathematics
and Computer Science
University of Perugia, 06123, Italy
Email: leonardo.mostarda@unipg.it
alfredo.navarra@unipg.it

Purav Shah
and Ramona Trestian
Faculty of Science and
Technology, Middlesex University,
London NW4 4BT, UK
Email: p.shah@mdx.ac.uk
r.trestian@mdx.ac.uk

***Abstract*—Blockchain is now a widespread technology. However, achieving mass adoption for specific applications requires overcoming several challenges inherent to the use of distributed algorithms and cryptographic primitives, which are central to blockchain technology. One of the main obstacles is interoperability between different blockchains, as each blockchain acts as an isolated environment. To address the interoperability problem, it is essential to use off-chain processes capable of interconnecting the blockchains in question. If the interconnected blockchains are private or consortium-based, attention must be paid to both the confidentiality of the data exchanged and the authority of those authorised to operate on them. We propose a novel approach in which each blockchain implements its own off-chain component. Each off-chain component does not communicate with the others and uses a threshold digital signature scheme as cryptographic proof aimed at improving the control and management of inter-chain transactions, alongside computational complexity reductions.**

1. Introduction

Blockchain is a disruptive technology, originally conceived as a distributed ledger to support the secure exchange of cryptocurrencies between untrusted parties. The transactions are kept within a ledger structured into a series of blocks. The blocks are linked and secured using cryptographic principles. Each block contains transactions exchanged by users within a certain period of time. Over the last decade, blockchain technology has evolved significantly, expanding functionality beyond what its pioneer, Bitcoin, offered. One of the most significant advancements is the development of smart contracts, digital contracts executed within the blockchain. Ethereum introduced a Turing complete programming language called Solidity [1], turning the blockchain into a programmable system. Within Ethereum, the smart contracts can interact with participants or other smart contracts. The availability of a variety of blockchains and their adoption is gradually permeating into the private sector as well. As more and more blockchains were developed, the need for them to interact with each other became apparent. Inter-chain transactions, or cross-chain communication, are critical to achieving a fully integrated blockchain environment, but they introduce a number of challenges, particularly in the areas of privacy and trust, especially for permission-based (permissioned) blockchains. It is known that interoperability is not possible without the presence of off-chain third parties in charge of conducting communications [2]. The off-chain components are in charge of verifying the presence of a transaction on a source blockchain and executing its counterpart on a destination blockchain. Projects such as Polkadot [3], Cosmos [4], and the Cross-chain Interoperability Protocol (CCIP) [5] have proposed solutions that aim at interconnecting public blockchains within a unified framework. Cosmos and Polkadot require specific configurations in the architecture of the interconnected blockchains, limiting their applicability. They make use of intermediate blockchains and relays responsible for moving transactions from a source blockchain to a destination blockchain. The presence of an intermediate blockchain adds a level of complexity that could affect the scalability of the entire infrastructure and furthermore, an economic incentive is needed to make an inter-chain transaction. CCIP is more versatile, being based on oracles and smart contracts, that involves the presence of different oracles that implement a consensus protocol. These oracles require read and write access to the interconnected blockchains. This problem arises in contexts where blockchains are permissioned, with CCIP also requiring economic incentives to carry out inter-chain transactions. Furthermore, none of the proposed protocols provide a mechanism that governs the initiation of an inter-chain transaction. This can be problematic in contexts where interconnected blockchains have significantly different block production times, generating bottlenecks. To mitigate these existing challenges, this paper explores a novel off-chain component for interoperability protocols that does not implement consensus algorithms but uses an threshold signature scheme to handle inter-chain transactions. The introduction of a threshold signature aims at making the protocol more efficient, as it requires less computational effort than consensus protocols. The protocol aims at promoting interoperability between permissioned

blockchains, improve control over cross-chain transactions, and preserve confidentiality. The main idea is to implement specific off-chain components for each blockchain, with these off-chain components do not directly communicate but work indirectly. One one hand, the off-chain components have the task of regulating user activities in sending inter-chain transactions, avoiding the saturation of the slower blockchain, favouring synchronisation between the inter-connected blockchains; but on the other hand, they aid in finalising an inter-chain transaction. In both cases, it is necessary to use a cryptographic proof that must be verified on specific smart contracts that resides on interconnected blockchains.

2. Interoperability Protocol

In this section, we present a protocol that aims at connecting two permissioned blockchains, or a permissioned blockchain to a public one. The idea of the protocol is based on the following points: 1. Users can initiate inter-chain transactions independently, like normal transactions; 2. Inter-chain transactions occur exclusively through smart contracts; and 3. The inter-chain process is in charge of verifying the activities in the inter-chain actions and finalising the transactions in the target blockchain. The protocol aims at combining the properties of relays and oracles. On one hand, the protocol is similar to relays, which are responsible for monitoring events and generating cryptographic proof; but on the other hand, it is similar to oracles, where the off-chain components themselves read the data needed to generate and send transactions. Figure 1 shows the steps needed by the protocol in case the interconnected blockchains are permissioned. Each blockchain (B_A and B_B) has its own off-chain component (OC_A and OC_B), capable of reading and writing to the blockchain to which it is connected, and can read only the portions of data subject to inter-chain transactions from the destination blockchain. An inter-chain transaction occurs according to the following steps: (1) a user sends an inter-chain transaction to the *Source* smart contract. After the transaction is validated, it generates an event or system log; (2) the off-chain component OC_B of the target blockchain B_B reads the contents of the inter-chain transaction; (3) using the data contained therein, it prepares an inter-chain transaction and sends it to the destination blockchain. If the transaction is valid, it is validated by the validator nodes of the destination blockchain; (4) once validated, the *Target* smart contract generates an event or system log that is captured by the off-chain component OC_A of the source blockchain B_A; (5) with the data contained therein, the off-chain component OC_A sends a transaction to the *Source* smart contract as an *ack* of completion. This completes an inter-chain transaction. In case the transaction on the target blockchain fails (step (3) of Figure 1), the off-chain component OC_A of the source blockchain B_A must execute a roll-back transaction that restores the state of the source blockchain, reverting any changes made by the transaction that generated an inter-chain transaction (step (1) of Figure 1), i.e., in case of transfer of fungible or non-fungible tokens, it is necessary to re-credit the user who initiated the transaction.

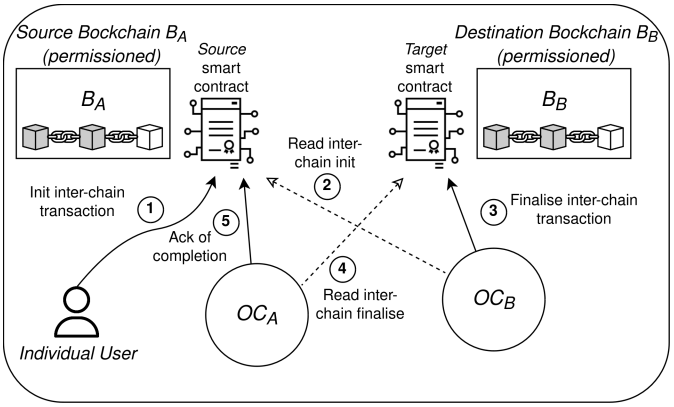

Figure 1. The figure shows the main steps needed by an inter-chain transaction from the source blockchain B_A to the destination blockchain B_B in the case both B_A and B_B are permissioned.

The protocol is simpler in the case of communications from a permissioned blockchain to a public one as a single off-chain component managed by the permissioned blockchain is required.

2.1. Observations

The computational complexity for the protocol implementation is quite low. Each permissioned blockchain has its own trusted off-chain component responsible for managing communications with other blockchains. Each blockchain has interface smart contracts that allow the execution of inter-chain transactions. An inter-chain transaction can be initiated by users via simple calls to the *Source* smart contract.

In the proposed protocol, the simplest way to define off-chain components is to provide individual processes responsible for verifying the status of inter-chain transactions and making decisions, such as finalising transactions and notifying their completion or rejection. However, structuring off-chain components as single processes opens up a significant number of problems. First, it introduces single points of failure, threatening the distributed or decentralised nature of blockchains. Second, delegating the power to change the state of one blockchain, based on the state of another blockchain, to a single process is very risky. Individual processes could be compromised or act maliciously. Moreover, in the case of consortium blockchains, this can open up disputes regarding the authority in charge of managing the node that implements the off-chain component.

A possible solution is to define the process as distributed or, at most, decentralised. At the same time, however, we want to avoid implementing a consensus algorithm, as it would introduce further scalability issues, and we want the off-chain process to be able to send transactions directly to the blockchain. With these properties in mind, in the next section, we present a decentralised off-chain component

that conducts cross-chain transactions and makes use of a threshold signature scheme.

3. Off-chain component based on a threshold signature scheme

The off-chain component must be distributed or decentralised and must not implement consensus algorithms to manage inter-chain transactions. This is essential to avoid scalability issues in inter-chain communications. In recent years, research on the so-called (t,n)-threshold signatures has been very active, proposing innovative solutions that have made this cryptographic scheme versatile and efficient. The main idea of a threshold signature is to enable a group of participants to collaboratively generate a single digital signature. Some of the most recent advances concern the introduction of multi-party computation (MPC) to improve the efficiency and security of threshold signature, especially in the distributed key generation protocol in charge of generating the individual keys of the participants. The major developments in this sense concern threshold signature schemes based on BLS [6] and ECDSA [7]. Therefore, we propose the introduction of a (t,n)-threshold signature scheme in order to increase the security and trust of off-chain nodes. The presence of a threshold signature guarantees that inter-chain transactions must be signed by a minimum number of nodes t, where t is the threshold required to produce valid signatures. So, the off-chain component is composed of n nodes OC = $\{p_1,\ p_2,\ldots,\ p_n\}$, each of which is equipped with a private key s_{k_i} known only to the node p_i itself. The set-up of the threshold scheme is entrusted to the same bodies that manage the validator nodes (i.e., referring to Figure 1, OC_A is set up by the authorities of B_A, and OC_B is set up by the authorities of B_B). The off-chain nodes are divided into two sets called primary and secondary. The primary node is responsible for requesting partial signatures from the secondary nodes, verifying their correctness, producing the threshold signature, and sending the transactions on their own blockchain. Secondary nodes must verify the primary node's requests and, if valid, produce their own partial signatures to return to the primary nodes. Each node is associated with a reputation score that discriminates between nodes that can be primary and those that are secondary. Primary nodes are selected among those with a high reputation score and rotate in a round-robin fashion. Each node has access to the state of the source blockchain and to the state inherent to the inter-chain transactions of the destination blockchain. In this way, as soon as a signature request is received from a primary node, each secondary node is able to verify whether the request is valid and can decide whether to produce the partial signature or reject it. Furthermore, only data relating to inter-chain transactions is requested by the destination blockchain, preserving confidentiality.

In case a selected primary node is malicious, it may try to request the signature of a fraudulent transaction. This behaviour is prevented by the threshold signature structure, as long as the number of honest nodes needed to reach the threshold (t) is greater than 50%; otherwise, there is a risk of producing valid signatures on fraudulent transactions. Alternatively, a malicious node may not send transactions intended to finalise an inter-chain transaction, degrading the performance of the off-chain component. In both cases, the malicious node's reputation score is degraded. If the score gets too low, the node is kicked off the network and must be replaced by a new one.

Figure 2 shows the case where the off-chain component OC_B of the destination blockchain B_B finalises an inter-chain transaction, i.e., steps (2) and (3) of Figure 1. After a user initiates an inter-chain transaction, the *Source* smart contract generates an event containing the data needed by the off-chain component of the target blockchain. The semantics of the events are well defined and follow a structure similar to [8]. In particular, the event must contain the following *data*: the *address* of the user who initiated the inter-chain transaction in the source blockchain; the *virtual nonce*, a progressive number that records the number of inter-chain transactions carried out by the user of the source blockchain; and *parameters*, the data subject to the inter-chain transaction.

The off-chain component of the destination blockchain works as follows:

1) the primary node of the off-chain component OC_B is aware of the event and requests the generation of partial signatures from the secondary nodes. Each secondary node has access to the same event and can verify the consistency of the request. So, each secondary node, using the *data* of the event, individually constructs a message m as follows:

$$m := [parameters] \ || \ address \ || \ virtual \ nonce$$

where *parameters* contain the input parameters of the inter-chain function to be executed, *address* is the public identifier of the user who initiated the inter-chain transaction, and *virtual nonce* is a progressive number that records the inter-chain transactions carried out by that user; the symbol || indicates the concatenation operation.
The *address*, *virtual nonce* pair guarantees that the events generated are unique and always identifiable by the off-chain components. Once the message m has been constructed, the off-chain nodes calculate the hash h = Hash(m), on which they produce their partial signatures σ_i. Then, they send back the partial signature to the primary node. Once the partial signatures σ_i have been collected, the primary node verifies if at least t signatures produced are valid. If so, the latter are aggregated to form the output signature σ. Once the signature is produced, the primary node prepares the transaction to finalise the inter-chain transaction. If the minimum threshold t is not reached, the primary node sends a transaction with an empty signature $\sigma = \emptyset$ that will produce an event notifying the rejection.

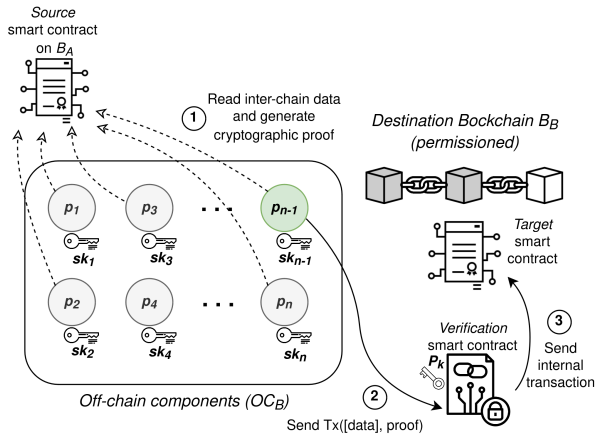

Figure 2. The figure shows how an off-chain component finalises an inter-chain transaction. The primary node (the green one) is in charge of collecting the partial signatures from the secondary nodes (the grey ones), verifying their validity, and producing the threshold signature needed to finalise an inter-chain transaction.

2) the transaction is sent to the *Verification* smart contract, in charge of verifying the threshold signature. If the threshold signature is recognised as valid,
3) an internal transaction is sent to the *Target* smart contract, responsible for verifying the validity of the transaction itself. The same scheme is used for the transactions that notify the completion of the inter-chain transaction, i.e., steps (4) and (5) of Figure 1.

4. Discussion

We have presented that using a threshold signature could be a good alternative to consensus protocols for managing inter-chain transactions, that allows for significant complexity reductions as well as enhancing interoperability of various blockchains. However, allowing individual users to initiate an inter-chain transaction could lead to scalability issues in the target blockchain, especially if the throughput of the interconnected blockchains is very different (i.e., the target blockchain is much slower than the source blockchain). Limiting the transaction rate may not be a sufficient countermeasure, as there is a risk that individual users could saturate the network, excluding other users from sending inter-chain transactions. Furthermore, depending on the use case, there may be a need to keep the state of the interconnected blockchain synchronised. A possible idea to mitigate these issues could be to introduce a lock and unlock mechanism for users to regulate activities in inter-chain transactions. Finding the most efficient solution to this issue requires further research and analysis.

5. Conclusion

The study presents a novel off-chain component that aims at interconnecting permissioned blockchains. Each blockchain has its own specific off-chain component, managed by the same authorities that manage the validator nodes. The off-chain components are not interconnected with each other but work indirectly via events or log systems generated by specific smart contracts resident on the different blockchains. Each off-chain component has access only to the portion of data affected by inter-chain transactions, increasing confidentiality. Off-chain nodes don't implement consensus algorithms, as it would affect scalability in inter-chain communication. To increase trust and security, off-chain nodes make use of a threshold signature scheme that must be validated on specific smart contracts. The presence of the threshold signature guarantees that a digital signature is valid only if a minimum number of nodes t sign it, keeping the off-chain component decentralised and avoiding single points of failure.

Acknowledgments

This research was funded by Ministero dell'Università e della Ricerca (MUR), issue D.M. 351/2022 "Borse di Dottorato" - Dottorato di Ricerca di Interesse Nazionale in "Blockchain & Distributed Ledger Technology", under the National Recovery and Resilience Plan (NRRP), and by the Italian National Group for Scientific Computation GNCS-INdAM. Special thanks go to Fadi Barbara for his useful discussions on threshold signature.

References

[1] G. Zheng, L. Gao *et al.*, "Solidity," 2016. [Online]. Available: https://docs.soliditylang.org/

[2] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, and W. J. Knottenbelt, "Sok: Communication across distributed ledgers," in *Financial Cryptography and Data Security*, N. Borisov and C. Diaz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 3–36.

[3] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," 2019, https://assets.polkadot.network/Polkadot-whitepaper.pdf.

[4] J. Kwon and E. Buchman, "Cosmos, a network of distributed ledgers," 2019, https://whitepaper.io/document/582/cosmos-whitepaper.

[5] C. Labs, "Chainlink cross chain interoperability protocol," 2023. [Online]. Available: https://blog.chain.link/introducing-the-cross-chain-interoperability-protocol-ccip/

[6] S. Garg, A. Jain, P. Mukherjee, R. Sinha, M. Wang, and Y. Zhang, "hints: Threshold signatures with silent setup," Cryptology ePrint Archive, Paper 2023/567, 2023, https://eprint.iacr.org/2023/567. [Online]. Available: https://eprint.iacr.org/2023/567

[7] B. Kachouh, L. Sliman, A. E. Samhat, and K. Barkaoui, "Demystifying threshold elliptic curve digital signature algorithm for multiparty applications," in *Proceedings of the 2023 Australasian Computer Science Week*, ser. ACSW '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 112–121.

[8] A. Bigiotti, L. Mostarda, A. Navarra, A. Pinna, R. Tonelli, and M. Vaccargiu, "Interoperability between evm-based blockchains," in *Advanced Information Networking and Applications*, L. Barolli, Ed. Cham: Springer Nature Switzerland, 2024, pp. 98–109.