Privacy-Preserving Initial Public Offering using SCALE-MAMBA and Hyperledger Fabric

Lucas Benmouffok, Kalpana Singh, Nicolas Heulot, Daniel Augot

Privacy-Preserving Initial Public Offering using SCALE-MAMBA and Hyperledger Fabric

Lucas Benmouffok*†, Kalpana Singh*, Nicolas Heulot*, Daniel Augot†
*IRT SystemX, Paris-Saclay, France,
Email: {lucas.benmouffok, Kalpana.Singh, Nicolas.Heulot}@irt-systemx.fr
†INRIA and LIX
Email: Daniel.Augot@inria.fr

***Abstract*—We consider Initial Public Offering (IPO) on blockchains while preserving privacy using Secure Multiparty Computation (MPC), which allows participants to perform a computation on secret data. We provide "MPC as a service", where users requiring a computation distributes shares of their data to MPC *workers* who run an MPC protocol on the shares and return the result. Previous work by Benhamouda et al. considered IPO over Hyperledger Fabric. We improve by providing a tighter and easier integration of MPC protocol in Fabric using the MPC library SCALE-MAMBA. We explain the obtained security benefits and experimental results are provided.**

I. Introduction

Privacy-preserving data management is a major concern in today's societies as the world has turned into a modern information-driven society. Different parties, from individuals to companies and governments, often participate in protocols to collaborate. However, each party's private data can be coveted by other parties, such as a company's unpublished data which could provide a substantial advantage on the stock market i.e. insider's knowledge.

A simple solution is to refer to a third party to handle the private data and deliver the result. But the trust is displaced onto the third party, which could prove to be a security flaw, should this third party be swayed by another. The context of our work revolves around these protocols where every participant wishes to collaborate, but will also try to gain an advantage over the others.

This context of participants willing to collaborate to compute the same function is close to the concept of blockchain and smart contracts. However, to use smart contracts, the input data from a transaction must be published in the blockchain ledger which implies that this data will be visible from all the participants of the blockchain network. To respond to this, we consider the family of collaborative cryptographic protocols known as Multi-Party Computation (MPC) protocols which provide for a set of participants with their inputs, the possibility of collaboratively computing a function f without disclosing their inputs. Thus, intertwining blockchain and MPC can be a potential solution.

Research in this direction is marked out by work such as Enigma [1], and Hawk [2]. In particular, the work of Benhamouda et al. [3] has demonstrated the feasibility of coupling blockchain and MPC. They continued in the same direction with [4] by implementing an Initial Public Offering protocol over Fabric. The IPO protocol is also computed by an MPC protocol to ensure the confidentiality of the data input.

In this paper, we propose an architecture where we combine Fabric with SCALE-MAMBA [5]. This powerful software system allows one to easily write, compile, and run an MPC program. SCALE-MAMBA also provides a communication and execution system, to run an MPC protocol over a peer to peer network. In particular, our architecture relies on coupling the MPC execution with an *external chaincode*, a feature proposed by Fabric v2.0. Thus, we aim at providing a generic way to build any MPC protocol within the Fabric blockchain.

This paper is structured as follow. After an introduction of the necessary preliminaries, we discuss the related work and focus on the work by Benhamouda et al. [3], [4]. Then, we present our architecture that we illustrate with an IPO protocol. We then provide experimental results and conclude.

II. Preliminaries

A Secure Multiparty Computation protocol is an interactive cryptographic protocol, which enables a set of n participants $P_1, \ldots, P_n$ to compute an agreed and public function $f(x_1, \ldots, x_n)$ of their inputs, without P_j disclosing his secret inputs x_j. MPC protocols are now practical after the SPDZ breakthrough [6] that proposed a quite efficient protocol.

A. Shamir's scheme and Multiparty computation

A building block for MPC is Shamir's secret sharing scheme [7], a cryptographic protocol whose aim is to distribute *shares* $\langle x \rangle_i$ of a secret value x amongst participants. No information about x can be obtained when at most t participants collude. Circuit traversal, used in MPC, which requires addition and multiplication, can be done privately. Secret sharing is *linear*: a participant P_j having shares $\langle x \rangle_j$ and $\langle y \rangle_j$ of x and y, and public λ, μ, computes $\lambda \cdot \langle x \rangle_j + \mu \cdot \langle y \rangle_j$, which is his share of $\lambda \cdot x + \mu \cdot y$. The product $x \cdot y$ is done using Beaver triples a, b, c such that $c = ab$ [8] and which are distributed in a precomputation phase. These triples are independent of the function f to be computed and of the inputs. One such triple is consumed for each product to be done on the data. There are many security properties of a MPC protocol. *Passive adversaries* must be distinguished from *active adversaries*. Active adversaries are malicious participants which arbitrarily deviate from the protocol. In SCALE-MAMBA, up to t active

adversaries can be tolerated, with *abort* when a suspicious behaviour is detected. On the other hand, passive adversaries honestly follow the MPC protocol, but may collude to learn other participants' inputs. Here t is a design parameter which is fixed before running the protocol.

B. Hyperledger Fabric

Hyperledger is a project created by the Linux Foundation and IBM in 2015 [9]. This project's goal is to develop blockchain technologies for business interactions between corporations. It is a permissioned blockchain infrastructure framework. Smart Contracts are called *chaincodes*. The ledger is maintained by nodes, called *peers*, who have public identities linked to an authorized organization. Each peer run chaincodes to participate to the update of ledger. The workflow is as follows: a client emits a transaction which is executed on a subset of peers, called *endorsers*. Transactions are simulated locally by endorsers without synchronisation, and state changes are sent back to the client, who forwards them to an *ordering service*, who broadcasts the results to all peers, who then change their local state. The chaincode and its associated state must be the same for all endorsers.

III. Related Work

Blockchain and privacy has become a growing subject over the last few years. The topic of MPC and blockchain has emerged [1], [2], which are oriented towards performing an MPC protocol with smart contract. Also [10], [11] combine MPC and blockchain towards electrical grid management in [10] and data-sharing with [11]. However, since we focus on the Fabric blockchain, we studied Benhamouda et al. 2016 [3] and 2018 [4] papers as our related work in this section.

In 2016, Benhamouda et al. [3] presented a modification to Fabric v1.0 for supporting computation over private data, with the example of a short auction protocol. The authors acknowledge that their objective is to show an initial proof of concept. They enumerate two important components of Fabric which are: a) the local configuration to deal with the data which might be visible to only a limited set of peers b) the communication between peers implemented during endorsement to complete the MPC protocol.

As security was not a concern in this proof of concept, the authors use a *helper server* to implement these components. It stores the local parameters of each peer and help communications between instances of the chaincode at different peers. Thus it acts as a trusted third party, enabling the auction protocol but limiting the secure benefits of the construction.

Then, in 2018 , Benhamouda and al. [4] published an Initial Public Offering (IPO) protocol over a "customized" Fabric version 1.1. Two main issues are to be solved without drastically changing Fabric. The first is that chaincodes in Fabric do not, generally, communicate with each other, whereas MPC participants do. The second is that a chaincode does not know the others. To circumvent the first issue, the authors used an additional System ChainCode (SCC) [12] which permits chaincodes to to send and receive messages using the peer's communication layer. Then, specifying the peers' addresses take care of the second issue. For their implementation, they used the public GMW-MPC [13]. Regarding the MPC side, the authors designed an ad-hoc MPC protocol for finding the selling price, using (shared) bit comparison ad hoc techniques. We chose a more flexible approach by using the external chaincode feature and SCALE-MAMBA to implement our protocol. We chose SCALE-MAMBA over [14] because of the available documentation at the time.

IV. System Architecture

This section details the system architecture. It relies on two main components: a) SCALE-MAMBA which proposes a suitable environment for writing, compiling and executing MPC protocols, b) Fabric which supports a private blockchain network.

A. Description of SCALE-MAMBA

SCALE-MAMBA is an environment that allows to compile and write MPC programs, build peers and run these programs on peers to compute the desired output of a function. It produces a *circuit* for the MPC protocol in the MAMBA code compilation. The standard model of MPC computation is algebraic circuit evaluation, which is far from a more natural, say Python-like, programming language. A huge benefit of SCALE-MAMBA is to fill the gap between these two models. SCALE-MAMBA has two phases: an offline phase and an online phase. These phases are used to compose the MPC protocol execution. During the offline phase, cryptographic data is provided to the participants, such as Beaver triples. This phase is independent of the function computation and the participants' inputs. A new type *sint* is introduced for secret values, which protects the data while allowing computation with this data. It enables to perform comparisons between secret and public integer values, which is of interest to us for implementing a binary search.

B. Roles and Integration of SCALE-MAMBA into Fabric

We consider the following three kind og entities:

1) *End users:* an individual who owns private information.
2) *Intermediaries:* they are trusted third parties that aggregate data coming from end-users, like *Brokers*, and are responsible for sending those data to the workers correctly.
3) *Workers:* they run the MPC on the data coming from Brokers. The Workers are peers running chaincodes and publishing the result on Fabric.

In our protocol, each secret share of data is transferred by a Broker to a particular Worker in a transaction endorsed by Fabric. The Broker sends a Worker this Worker's share encrypted with the Worker's SSL public key. We assume that the public key and IP of each Worker are known at the beginning of the protocol. Notably, SCALE-MAMBA implements (secure) communication between workers in the MPC protocol. Each Worker runs the same chaincode. We use the chaincode to synchronize the MPC programs. It consists

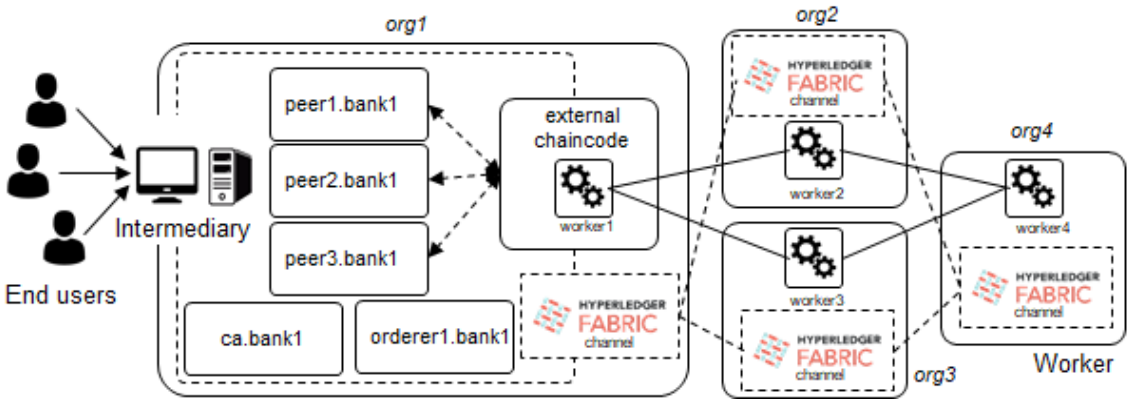

Fig. 1. Architecture overview in the context of an IPO protocol. Each organization (orgN) represents a broker or bank that will aggregate a set of orders coming from investors. Each one manages peers (peerN) connected into the same hyperledger network (channel). Each one also deploys an external chaincode that will run the mpc computation (workerN).

of two steps. At first, we gather the transactions that will contribute to the input of each Worker. For each transaction, each Worker tests if the transaction contains data intended for it (by checking successful decryption). Secondly, when all the data is transferred, a transaction is sent that triggers the MPC protocol. After completion, each Worker stores the result in the ledger, enabling each participant to the Fabric network to have access to the result of the computation.

To write in the ledger, we suppose that all the peers in the endorsing policy are Workers. A write in the Fabric blockchain launches the MPC protocol. It must output the same result for all Workers, which is validated by another write. Then, a new round of data transfer can start a new computation. The chaincode can handle one MPC computation at a time but it can prepare multiple data transfers in parallel by indexing data into computation sessions. In order to integrate SCALE-MAMBA within Fabric, we use an external chaincode [9] a feature proposed by Fabric v2.0. This feature allows to run a chaincode with access to the SCALE-MAMBA library. This facilitates the preparation of MPC program that will be run. Indeed, the MPC compilation using SCALE-MAMBA can take some time and cannot be done on the fly. So, we compile different settings of some MPC programs, such as the size of the input data or the number of workers. We suppose that all workers have access to the same set of pre-compiled programs and a first init transaction sent to the chaincode targets a specified setting between workers. For these reasons, we think that using SCALE-MAMBA within external chaincode makes the integration of MPC easier and more adaptive than using the helper server proposed in [3] and the modified communication layer with the ad hoc solution proposed in [4].

V. Implementation of a secure IPO protocol

A. Initial Public Offering

An IPO is a financial protocol, where a company sells shares to investors. In the protocol from [4], the selling price is determined by the offer and is called the *clearing price*, which is the (MPC) computational goal of the IPO. The IPO protocol of the paper is as follows:

- The Seller is selling a public *fixed amount* S of shares and sets (public) maximum price P
- B Buyers express orders. An (secret) order is a couple (p, v), where the Buyer wants to buy v shares at price p.
- Buyer i places a list $order_i$ of orders (p, v): $order_i = [(1, v_{i,1}), \ldots, (p, v_{i,p}), \ldots, (P, v_{i,P})]$. We also write $order_i(p) = v_{i,p}$. For each Buyer i, the function $p \mapsto order_i(p)$ is decreasing.
- Buyer i secretly sends his list $order_i$ to his Broker.
- Brokers deal shares of orders: $\langle v_{i,p} \rangle_j$ is sent to Worker W_j.
- Workers compute in a MPC way the clearing price p^*:

$$p^* = \max_{1 \leq p \leq P} \{p : \text{Vol}(p) \geq S\} \quad (1)$$

where $\text{Vol}(p) = \sum_{i=1}^{n} order_i(p)$

B. MPC binary search of orders

The difficult step is the computation of p^* (Eq. 1) as specified in Algorithm 1. This is a standard algorithm, using a binary search which, in MPC, requires to execute comparison on hidden data. As opposed to [4], who designed ad hoc comparison and bit-wise operation, we use the comparison in SCALE-MAMBA. The whole protocol is shortly described in Algorithm 2. We let the number of Brokers unspecified.

Algorithm 1 MPCbinarySearch

Input, public: A plaintext integer S
Input, private: an array T sorted in decreasing order
Workers do a binary search in MPC to find p^* as in Eq. 1
Output, public: reveal p^*

C. Integration in Fabric and Security

Using the communication layer of SCALE-MAMBA removes the ad hoc communication chaincode proposed in [4].

Algorithm 2 IPO using MPC

for $i = 1 \dots B$ **do**
  Buyer i sends his list of orders to a Broker
for $p = 1 \dots P$ **do**
  for $j = 1 \dots n$ **do**
    for $i = 1 \dots B$ **do**
      Worker W_j gets $\langle v_{i,p} \rangle_j$ from a Broker
for $j = 1 \dots W$ **do**
  for $p = 1 \dots P$ **do**
    Worker W_j computes $\langle \mathrm{Vol}\, p \rangle_j \leftarrow \sum_i \langle v_{i,p} \rangle_j$
  Each W_j has $\langle \mathrm{Vol} \rangle_j \leftarrow [\langle \mathrm{Vol}(p) \rangle_j : p \in \{1 \dots P\}]$
All workers execute MPCbinarySearch

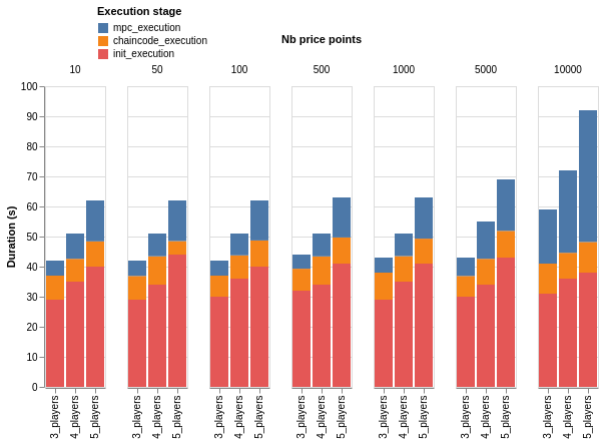

Fig. 2. Measures of the different executions within the protocol: the execution of time of the initialization, the execution time of the external chaincode, and the execution time of the compiled MAMBA program on SCALE.

Using the SCALE-MAMBA comparison removes the burden of programming the method of [4], and will benefit of progress from SCALE-MAMBA. After the MPC protocol is executed, the clearing price is committed on the blockchain. An MPC protocol can not prevent an active adversary from providing a false input to the protocol. In our framework, Workers must honestly input these shares into the MPC protocol: we cannot tolerate active adversarial Workers. They must be passive and complying with the protocol.

VI. Experimental results

We did experiments to analyze the performances of our approach in different setups and to be able to compare to [4]. All our experiments were carried out in a virtual machine running Ubuntu 20.04 and Docker. This machine was using an Intel Core Haswell CPU with 12 cores at 2.3Ghz and 64Go of RAM. We used SCALE-MAMBA 1.12 and Fabric 2.2.1.

The parameters are the number of participants, the threshold t, and the number of price points. Order volumes are random in the range $[0, 99999]$, and we designed the value to find as the one that gives the worst-case behavior for the MPCbinarySearch to provide consistent worst case timings.

The number of price points is impacting the execution and the compilation times. The program size increases linearly with the number of price points. The compilation time is below five seconds for any configuration below a thousand price points and then grows up to 80 seconds for three players to 450 seconds for five players at the 10000 price point mark.

In our 4 players setup, which is the same as [4], we can see that any execution time of the MPC under 5000 price points is better in our experiment. However, it ties at 10000 price points. The execution time of the MPC behaves well concerning the number of price points. Also adding one worker does not drastically impair the performance.

VII. Conclusion

We presented a system architecture which delineates the integration of SCALE-MAMBA within Fabric. We examined and improved the earlier work by Benhamouda et al. by providing a simpler integration of an IPO MPC protocol using SCALE-MAMBA. The experiments showed that our solution had roughly the same efficiency as Benhamouda et al. [4] paper. Concerning IPO, price points are statically defined, and future work would be to have the list of price points dynamically built on the basis of Buyers' actual demand.

References

[1] H. Shrobe, D. L. Shrier, and A. Pentland, *CHAPTER 15 Enigma: Decentralized Computation Platform with Guaranteed Privacy*. MIT Press, 2018, pp. 425–454.

[2] A. Kosba, A. Miller, E. Shi *et al.*, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 839–858.

[3] F. Benhamouda, S. Halevi, and T. Halevi, "Supporting private data on hyperledger fabric with secure multiparty computation," in *2018 IEEE International Conference on Cloud Engineering*, 2018, pp. 357–363.

[4] T. Halevi, F. Benhamouda, A. D. Caro *et al.*, "Initial public offering (IPO) on permissioned blockchain using secure multiparty computation," in *IEEE International Conference on Blockchain*, 2019, pp. 91–98.

[5] A. Aly, K. Cong, D. Cozzo *et al.*, "Scale–mamba v1.12 : Documentation." [Online]. Available: https://homes.esat.kuleuven.be/~nsmart/SCALE/Documentation.pdf

[6] I. Damgård, V. Pastro, N. Smart *et al.*, "Multiparty computation from somewhat homomorphic encryption," in *Advances in Cryptology – CRYPTO 2012*, 2012, pp. 643–662.

[7] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, p. 612–613, Nov. 1979.

[8] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Advances in Cryptology — CRYPTO '91*, 1992, pp. 420–432.

[9] "https://hyperledger-fabric.readthedocs.io."

[10] Z. Guan, X. Zhou, P. Liu *et al.*, "A blockchain based dual side privacy preserving multi party computation scheme for edge enabled smart grid," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[11] Y. Yang, L. Wei, J. Wu *et al.*, "Block-smpc: A blockchain-based secure multi-party computation for privacy-protected data sharing," in *Proceedings of The 2nd International Conference on Blockchain Technology*, 2020, p. 46–51.

[12] E. Androulaki, A. Barger, V. Bortnikov *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference (EurosSys '18)*, 2018.

[13] S. G. Choi, K.-W. Hwang, J. Katz *et al.*, "Secure multi-party computation of boolean circuits with applications to privacy in on-line marketplaces," in *Topics in Cryptology – CT-RSA 2012*, pp. 416–432.

[14] M. Keller, "Mp-spdz: A versatile framework for multi-party computation," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 2020, p. 1575–1590.