Private Balance-Checking on Blockchain Accounts Using Private Integer Addition *

Birenjith Sasidharan[1] and Emanuele Viterbo[2]

[1] Department of Electronics & Communication Engineering
Government Engineering College, Barton Hill
Trivandrum, India
Email: birenjith@gecbh.ac.in
[2] Dept. of Electrical and Computer Systems Engineering
Monash University
Clayton, Australia
Email: emanuele.viterbo@monash.edu

Abstract. A transaction record in a sharded blockchain can be represented as a two-dimensional array of integers with row-index associated to an account, column-index to a shard and the entry to the transaction amount. In a blockchain-based cryptocurrency system with coded sharding, a transaction record of a given epoch of time is encoded using a block code considering the entries as finite-field symbols. Each column of the resultant coded array is then stored in a server. In the particular case of PolyShard scheme, the block code turns out to be a maximum-distance-separable code. In this paper, we propose a privacy-preserving multi-round protocol that allows a remote client to retrieve from a coded blockchain system the sum of transaction amounts belonging to two different epochs of time, but to the same account. At the core of the protocol lies an algorithm for a remote client to privately compute a non-linear function referred to as *integer addition* of two finite-field symbols representing integer numbers, in the presence of curious-but-honest adversaries. Applying it to balance-checking in a cryptocurrency system, the protocol guarantees information-theoretic privacy on account number and shard number thereby ensuring perfect user anonymity, and also maintains confidentiality of half of the input bits on average. The protocol turns out to be a useful primitive for balance-checking in lightweight clients of a PolyShard-ed blockchain.

1 Introduction

Identity management of parties involved in a financial transaction is an important problem. In blockchain-based cryptocurrencies, payment verification systems do not require to know the identities of the involved parties. However, every miner stores the entire history of transactions, and as a consequence the entire data is visible to all participants. Thus in spite of little use, the identity of parties involved in a transaction is publicly revealed resulting in loss of privacy. In Bitcoin [18], an approach of using pseudonyms in place of account addresses is adopted to tackle this problem. As long as the pseudonym cannot be linked to the real network or account address of the involved party, privacy is preserved. However, this approach is vulnerable to two kinds of deanonymization attacks. In the first kind of blockchain-based deanonymization, patterns in transactions and other observable side information are made use of by attackers. In [20], global properties of Bitcoin transaction graph,

* This work was supported by the Australian Research Council through the Discovery Project under Grant DP200100731.

i.e., a graph with account addresses as nodes and transactions as edges, are studied with the aid of empirical data of Bitcoin up to block number 215, 399. The structure and dynamics of the graph are shown to play a key role in determining how far a user can remain anonymous. In [16], authors identify heuristic methods for linking multiple addresses controlled by the same user in Bitcoin, thus partly deanonymizing certain users. Similar evaluation of user privacy in Bitcoin has been carried out in many papers [1]. The lack of privacy becomes more severe in smart contract systems like Ethereum [25], wherein transactions not only contain payment details, but also include function calls to specific applications. A cryptographic primitive known as zero knowledge proof provides a promising solution to fix some of the blockchain-based deanonymization risks. Essentially, a zero knowledge proof allows a prover to convince a verifier of some claim without revealing confidential information associated to that claim. Various strategies based on the notion of zero-knowledge proofs [2, 4–6, 12] have been proposed in literature. In the second kind of network-based deanonymization [3], adversaries observe network traffic for long enough duration to decipher network addresses associated to pseudonyms. In Bitcoin, when a node generates a transaction, it broadcasts the transaction over the P2P network by flooding. An adversarial node can observe the spreading dynamics of a given transaction to infer source IP of the transaction. A framework proposed in [8], named as Dandelion++ [8], is claimed to defend large-scale deanonymization attacks by malicious adversaries with near-optimal information-theoretic guarantees on privacy. The ubiquitous use of lightweight clients opens doors to a second instance of network-based deanonymization attacks. Users who are constrained in resources such as bandwidth or storage space use lightweight clients, primarily tailored for devices such as smart phones. It is estimated that $4.2 - 9.8$ million Bitcoin wallets are lightweight clients [15]. Lightweight clients download necessary parts of the blockchain data or verify transactions pertaining to specific accounts, making them more vulnerable to deanonymization attacks. In [15], authors propose an architecture named as BITE that attempts to solve privacy leakage of lightweight clients without sacrificing on performance.

1.1 Private Information Retrieval and Function Computation

Private information retrieval (PIR) protocols permit to ensure information-theoretic privacy on user identity under network-based deanonymization attacks. Private information retrieval was introduced in [7]. The potential use of PIR in the context of blockchain is observed in [10, 15, 21, 22]. In [7], Chor et al. considers the problem of accessing x_i of a k-length binary string $(x_1, x_2, \ldots, x_k)$ replicated across n servers without revealing to them which bit is accessed. In [23], PIR is extended to query from coded databases, and it is shown that one extra bit is enough to achieve privacy with coded data, if the number of servers grow exponentially in number of messages. Explicit PIR protocols for data stored in an encoded manner by a maximum-distance-separable (MDS) code are first presented in [24] both for the case of colluding servers and for no collusion. Later, PIR protocols for MDS-coded servers under various adversarial scenarios are studied [9]. As a natural extension to PIR, there are recent works that investigate private computation of functions on distributed data. Many of these works [17,19] consider retrieving linear combination of message symbols from coded systems without leaking privacy.

1.2 PIR Protocols for PolyShard-ed Blockchain

The feasibility to design a truly scalable blockchain system is answered in the affirmative in [13]. In [13], authors propose an architecture named as PolyShard

that stores sharded blocks encoded by a generalized Reed-Solomon code and that permits validation of coded blocks. A PIR protocol to access data in blockchains employing coded sharding, for instance PolyShard, is proposed in our previous work [22]. The protocol in [22] fetches a single transaction record ensuring information-theoretic privacy on anonymity of account addresses in the presence curious-but-honest adversaries.

1.3 Our Contributions

In the present paper, we propose a privacy-preserving multi-round protocol that allows addition of two integers stored in a distributed storage system. A collection of integer numbers is organized as a two-dimensional array of finite-field symbols, then encoded using an MDS code, and finally stored in a dynamic distributed storage system. Our protocol permits to retrieve at a remote client the output of an integer-addition function referred to as *integer-sum,* ensuring privacy of row-index and column-index of the entries that are added. Computing integer-sum of corresponding entries of two distinct arrays easily translates to balance-checking on blockchain-based cryptocurrencies. In that way, the protocol provides a useful primitive for balance-checking in lightweight clients of a PolyShard-ed blockchain, with information-theoretic guarantees on user anonymity (in the presence curious-but- honest adversaries) and partial confidentiality on transaction amounts.

In Sections 2, 3 and 4, we introduce an abstract system model for a Polyshard-ed blockchain, describe the integer-addition function and the protocol, and subsequently establish the privacy guarantees. In Section 5, we describe how the protocol can applied to a Polyshard-ed cryptocurrency system to carry out private balance-checking on an account.

2 Private Integer Addition in a Distributed Storage System

2.1 The Dynamic Distributed Storage System

We consider a dynamic distributed storage network with n nodes. Time is slotted into T epochs and in every epoch $t \in [T]$, k independent data blocks enter into the network. A data block is an element of $\mathbb{F}_q^\rho$ with $\rho > 1$. Thus k data blocks jointly constitute a $(\rho \times k)$ array of message symbols from $\mathbb{F}_q$. Let us denote such array as the *message matrix*

$$U = (\underline{u}_1, \underline{u}_2, \ldots, \underline{u}_k) \in \mathbb{F}_q^{\rho \times k}$$
$$= \begin{bmatrix} u_{11} & u_{21} & \cdots & u_{k1} \\ u_{12} & u_{22} & \cdots & u_{k2} \\ \vdots & & & \vdots \\ u_{1\rho} & u_{2\rho} & \cdots & u_{k\rho} \end{bmatrix}$$

where its j-th column $\underline{u}_j = [u_{j1}\ u_{j2}\ \cdots u_{j\rho}]^T, j = 1, 2, \ldots, k$ is the j-th data block. The message is encoded by an $[n, k]$ block code $\mathcal{C}$ to produce n data blocks $\{\underline{x}_i\}_{i=1}^n$ The matrix

$$X = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n) \in \mathbb{F}_q^{\rho \times n}$$
$$= \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & & & \vdots \\ x_{1\rho} & x_{2\rho} & \cdots & x_{n\rho} \end{bmatrix}$$

is referred to as a *codeword matrix*[3] where i-th column $\underline{x}_i = [x_{i1}\ x_{i2}\ \cdots x_{i\rho}]^T, i = 1, 2, \ldots, n$ is stored in the i-th node. To make things precise, both U and X must be associated to a particular time t by using a notation $U(t)$ and $X(t)$ respectively. However, we drop this association for brevity, and the time index, if at all it becomes important to consider, will be clear from the context.

Let $G = [g_{ij}] \in \mathbb{F}_q^{k\times n}$ and $H = [h_{ij}] \in \mathbb{F}_q^{(n-k)\times n}$ respectively denote the generator matrix and a parity-check matrix of the code $\mathcal{C}$. Then we have $X = UG$ and $XH^T = 0_{\rho\times(n-k)}$. When the code is a generalized Reed-Solomon code with generator matrix G_{GRS} and parity-check matrix H_{GRS}, then the dynamic distributed storage system becomes an abstraction of a PolyShard-ed cryptocurrency system.

2.2 The Problem of Integer Addition

Consider two consecutive time instants $t, t+1$ in a dynamic distributed storage system. Let U, V be the associated message matrices in order. The (p, j)-th entry of U, V are denoted by u_{jp} and v_{jp}, respectively. We assume that $\mathbb{F}_q$ is of characteristic 2. Then $q = 2^m$ for some positive integer m, and $\mathbb{F}_q = \mathbb{F}_2(\alpha)$ where α is a primitive element of $\mathbb{F}_q$. Furthermore, the ordered set $\mathcal{B} = \{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$ forms a basis of $\mathbb{F}_q$ over $\mathbb{F}_2$. We remark here that every result in this paper will hold true with suitable modifications for any other finite characteristic value.

Expanding over the basis $\mathcal{B}$, we can write

$$u_{jp} = \sum_{b=0}^{m-1} u_{jp}[b]\alpha^b,$$
$$v_{jp} = \sum_{b=0}^{m-1} v_{jp}[b]\alpha^b,$$

where $u_{jp}[b]$ and $v_{jp}[b], b = 0, 1, \ldots, (m-1)$ are the unique coefficients, all belonging to $\mathbb{F}_2$. Thus u_{jp} (or v_{jp}) can equivalently be represented by a binary vector of length m, and we write

$$u_{jp} \equiv (u_{jp}[0], u_{jp}[1], \ldots u_{jp}[m-1])$$
$$v_{jp} \equiv (v_{jp}[0], v_{jp}[1], \ldots v_{jp}[m-1]).$$

In turn, these binary vectors can be viewed as integers and the canonical transformation is defined by the function *int-val*$(\cdot)$:

$$\textit{int-val}(u_{jp}) = \sum_{b=0}^{m-1} u_{jp}[b]2^b$$

where the summation and multiplication on the right hand side is over $\mathbb{Z}$, the set of integers. We define *ff-val*$(\cdot)$ as the inverse transformation to convert an integer back to the finite-field element as:

$$u_{jp} = \textit{ff-val}(\textit{int-val}(u_{jp})).$$

Next, we define a binary operation $\boxplus$ in $\mathbb{F}_q$ as

$$u_{jp} \boxplus u_{jp} := \textit{ff-val}(\textit{int-val}(u_{jp}) + \textit{int-val}(v_{jp})).$$

[3] Observe that we use an unconventional notation of denoting the entry in p-th row and j-th column by u_{jp}, x_{jp}. We encode U as $X = UG$ going by standard coding-theoretic notation. At the same time, i-th column of X belongs to i-th node, making it natural to denote it by $\underline{x}_i$. The unconventional indexing helps to satisfy both.

In fact, $u_{jp} \boxplus u_{jp}$ is the finite-field element corresponding to the sum of integers *int-val*(u_{jp}) and *int-val*(u_{jp}). It may so turn out that *int-val*(u_{jp})+*int-val*$(v_{jp}) > 2^m$ and thus can no longer be represented by a binary string of m bits. However, we assume that u_{jp}, v_{jp} are such that a condition of overflow does not arise. It is straightforward to see that

$$u_{jp} \boxplus v_{jp} = u_{jp} + v_{jp} + r \tag{1}$$

for a unique $r \in \mathbb{F}_q$, referred to as the *carry element*. The carry element r is a function of u_{jp}, v_{jp}, and can be computed recursively (see [11]) as given below:

$$\begin{aligned} r[0] &= 0 \\ r[1] &= u_{jp}[0]v_{jp}[0] \\ r[2] &= u_{jp}[1]v_{jp}[1] + r[1](u_{jp}[1] + v_{jp}[1]) \\ &\vdots \\ r[m-1] &= u_{jp}[m-2]v_{jp}[m-2] + \\ &\qquad r[m-2](u_{jp}[m-2] + v_{jp}[m-2]). \end{aligned} \tag{2}$$

We refer to $u_{jp} \boxplus v_{jp}$ as the *integer-sum* of u_{jp} and v_{jp}. It is also clear by (1) and (2) that integer addition is a non-linear function of $2m$ input bits.

2.3 Protocol and Privacy Requirements

Consider a client that accesses the distributed storage system over the network. As mentioned in Sec. 2.2, $u_{jp}, v_{jp}, j \in [k],\ p \in [\rho]$ denote two symbols corresponding to (p, j)-th entry of message matrices associated with two consecutive time instants. The client would like to retrieve the integer-sum $u_{jp} \boxplus v_{jp}$ querying the servers. The client shall send suitably constructed set of queries $\{q_i(j, p, \tau) \mid \tau \in \Gamma\}$ to i-th server, $i = 1, 2, \ldots, n$. If Γ is a singleton set, then there is only one query sent by the client to every server. If $|\Gamma| > 1$, we obtain a multi-round protocol. Each server will respond to queries with answers $\{a_i(\tau) \mid \tau \in \Gamma\}$. The client computes $u_{jp} \boxplus v_{jp}$ making use of answers transmitted by every server.

We consider a scenario where $d \in [n], d \leq (n-k)$ servers can possibly be adversarial. The adversarial servers can collude among themselves, and are assumed to be curious-but-honest, i.e., they fully respect the protocol, while making all efforts to gather information about the queried data. In such a setting, we consider three notions of privacy.

(a) Privacy on p: The queries transmitted to colluding servers must not reveal any information about p.
(b) Privacy on j: The queries transmitted to colluding servers must not reveal any information about j.
(c) Privacy on input data of the function: The answers transmitted by servers should not reveal any more information about u_{jp}, v_{jp} than what is revealed by $u_{jp} \boxplus v_{jp}$.

The first two notions (a) and (b) imply user anonymity, and the third (c) relates to confidentiality of input data.

3 A Protocol for Private Retrieval of Integer-Sum

In this section, we present a protocol that permits private retrieval of integer-sum $u_{jp} \boxplus v_{jp}$. Before we describe the protocol, we discuss retrieval of integer-sum in an uncoded single-server system and make certain observations.

3.1 Retrieval of Integer-Sum in An Uncoded Single-Server System

Suppose $u, v \in \mathbb{F}_q$ are stored in a server. A remote client would like to access $u \boxplus v$. A trivial strategy is to download u, v into the client and compute $u \boxplus v$. A second strategy is for the server to compute $u \boxplus v$ and transmit it to the client. In the first, we require $2m$ bits of network download, with no privacy on input data at the client. In the second strategy, client downloads m bits, with the best achievable privacy on input data. In what follows, we consider a strategy that "lies between" these two strategies.

Let $u \odot v$ denote finite-field symbol resulted by bit-wise AND of binary vectors associated to u and v. Then for the client to compute $u \boxplus v$ using (1), (2), it is sufficient that client gets access to bit-wise XOR $u+v$ and bit-wise AND $u \odot v$ as in:

$$u + v \equiv (u[0] + v[0], u[1] + v[1], \ldots, u[m-1] + v[m-1])$$
$$u \odot v \equiv (u[0]v[0], u[1]v[1], \ldots, u[m-1]v[m-1]).$$

Let $b \in \{0, 1, \cdots, m-1\}$. For every b such that $u[b] + v[b] = 1$, bits $u[b]$ and $v[b]$ are different to each other and therefore $u[b]v[b]$ must be 0. Thus it is redundant to transmit $u[b]v[b]$ given the knowledge that $u[b] + v[b] = 1$. On the other hand, for every b such that $u[b]+v[b] = 0$, clearly $u[b] = v[b] = u[b]v[b]$. Therefore transmitting $u[b]v[b]$ to the client is equivalent to transmitting $u[b]$. Thus the server can transmit $u + v$ and $\{u[b] \mid u[b] + v[b] = 0\}$ so that the client gets access to $u + v$ and $u \odot v$. Then the client can do remaining computations to determine $u \boxplus v$. This strategy consumes $1.5m$ bits of download on average, partly preserves privacy on input data at client, and requires only linear operations at the server. We shall use this key observation while designing a protocol for the dynamic distributed storage system.

3.2 Description of The Protocol

The i-th server, $i = 1, 2, \ldots, n$ contains $\{\underline{x}_i, \underline{y}_i\}$, the coded blocks associated to message matrices U and V. The protocol to retrieve $u_{jp} \boxplus v_{jp}$ can involve multiple queries sent to the same server, i.e., $|\Gamma| \geq 1$. In each phase of the protocol, indexed by $\tau \in \Gamma$, queries are sent to every server, and answers are collected. The number of phases can vary between 1 and $m + 1$. The protocol can be described in three steps that are executed in order:

(1) The client sends a query $\underline{q}_i(j, p, -1) \in \mathbb{F}_q^\rho$ to i-th server at phase $\tau = -1$. The i-th server responds with answer $a_i(-1) = \underline{q}_i(j, p, -1)^T(\underline{x}_i + \underline{y}_i)$. The client computes

$$\sum_{i=1}^{n} a_i(-1) = u_{jp} + v_{jp}.$$

(2) For every $b \in \{0, 1, \ldots, m-1\}$ such that $u_{jp}[b]+v_{jp}[b] = 0$, the client sends query $\underline{q}_i(j, p, b) \in \mathbb{F}_q^\rho$ to i-th server at phase $\tau = b$. The server responds with answer $a_i(b) = Tr(\underline{q}_i(j, p, b)^T \underline{x}_i)$, where $Tr(\cdot)$ is the trace function. The computes

$$\sum_{i=1}^{n} a_i(b) = u_{jp}[b] \ = \ u_{jp}[b]v_{jp}[b].$$

(3) After first two steps, $u_{jp} + v_{jp}$ and $u_{jp} \odot v_{jp}$ are known at the client. The client computes $u_{jp} \boxplus v_{jp}$ using (1), (2).

Our protocol builds on top of a protocol proposed in [22] that permits private retrieval of a single message symbol u_{jp} from a dynamic distributed storage system. It turns out that the query constructed exactly as in [22] works well at phase $\tau = -1$, i.e., to retrieve $u_{jp} + v_{jp}$. However, it does not directly apply for later phases $\tau \in \{0, 1, \ldots, m-1\}$, i.e., to retrieve $u_{jp}[b], b \geq 0$. We present the query construction in a general framework that applies to every value of τ. However, we remark that the construction of $\underline{q}_i(j, p, \tau = -1)$ was known from [22], whereas that of $\underline{q}_i(j, p, b), b \geq 0$ is developed in the present paper. Let us write $H_{\text{GRS}} \in \mathbb{F}_q^{(n-k)\times n}$ and $G_{\text{GRS}} \in \mathbb{F}_q^{k\times n}$ as

$$H_{\text{GRS}} \;=\; \begin{bmatrix} \underline{h}_1^T \\ \underline{h}_2^T \\ \vdots \\ \underline{h}_{n-k}^T \end{bmatrix}, \;\; G_{\text{GRS}} \;=\; \begin{bmatrix} G_1 \; G_2 \end{bmatrix},$$

where G_1 is a $(k \times k)$ invertible sub-matrix of G_{GRS} and $\underline{h}_\ell^T = [h_{\ell 1} \; h_{\ell 2} \; \cdots \; h_{\ell n}], 1 \leq \ell \leq (n-k)$ is the ℓ-th row of H_{GRS}. With $G_1^{-1} := \begin{bmatrix} \tilde{\underline{\phi}}_1 \; \tilde{\underline{\phi}}_2 \cdots \tilde{\underline{\phi}}_k \end{bmatrix}$ we obtain an n-dimensional row-vector $\underline{\phi}_j^T := \begin{bmatrix} \tilde{\underline{\phi}}_j^T & \underline{0} \end{bmatrix}$ by appending $(n-k)$ zeros to $\tilde{\underline{\phi}}_j^T$.

Construction of $\underline{q}_i(j, p, \tau)$ at phase τ The phase τ takes on values from the set $\Gamma = \{-1\} \cup \mathcal{I}$, where $\mathcal{I} \subseteq \{0, 1, \ldots, m-1\}$. For every phase $\tau \in \Gamma$, pick $\underline{r}_{\tau 1}, \underline{r}_{\tau 2}, \ldots \underline{r}_{\tau d} \in \mathbb{F}_2^\rho$ uniformly at random independent of all random vectors generated in previous phases, where d is the number of curious-but-honeset servers. Compute $\underline{r}_{\tau,d+1} = (\sum_{i=1}^d \underline{r}_{\tau i}) + e_p$. Here $\underline{e}_p$ denotes the ρ-dimensional standard basis vector with 1 at p-th position. Let the ordered set $\mathcal{B}' = \{\beta_0, \beta_1, \ldots, \beta_{m-1}\}$ be the dual basis to $\mathcal{B}$. We set $\beta_{-1} = 1$. Then define

$$\begin{aligned} R_{\tau p} &= \begin{bmatrix} \underline{r}_{\tau 1} \; \underline{r}_{\tau 2} \cdots \; \underline{r}_{\tau,d+1} \end{bmatrix} \;\in\; \mathbb{F}_q^{\rho \times d+1} \\ \Psi_{j\tau} &= \begin{bmatrix} \underline{h}_1^T + \beta_\tau \underline{\phi}_j^T \\ \underline{h}_2^T + \beta_\tau \underline{\phi}_j^T \\ \vdots \\ \underline{h}_{n-k}^T + \beta_\tau \underline{\phi}_j^T \\ \beta_\tau \underline{\phi}_j^T \end{bmatrix} \;\in\; \mathbb{F}_q^{d+1 \times n} \\ Q_\tau &= R_{\tau p} \Psi_{j\tau} \;\in\; \mathbb{F}_q^{\rho \times n}. \end{aligned}$$

The query vector $q_i(j, p, \tau)$ for the i-th node at phase τ is the i-th column of Q_τ, $1 \leq i \leq n$. This completes the description of the protocol.

4 Correctness and Privacy Guarantees

4.1 Correctness

It follows from Theorem 3.3 of [22] and linearity of inner product that

$$\sum_{i=1}^n \underline{q}_i(j, p, -1)^T (\underline{x}_i + \underline{y}_i) = u_{jp} + v_{jp}, \tag{3}$$

which ensures correctness of Step (1) of the protocol. We first state a useful known fact in Prop. 1, and subsequently establish correctness of Step (2) of the protocol in Prop. 2.

Proposition 1. *[14] Let $u \in \mathbb{F}_q = \mathbb{F}_2(\alpha)$ where α is a primitive element. Let $\mathcal{B}' = \{\beta_0, \beta_1, \ldots, \beta_{m-1}\}$ be the dual basis to $\mathcal{B} = \{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$. If $u = \sum_{i=0}^{m-1} u[i]\alpha^i$, then $u[b] = Tr(\beta_b u),\ 0 \leq b \leq (m-1)$.*

Proposition 2. *For $j \in [k], p \in [\rho], b \in \{0, 1, \ldots, m-1\}$*

$$\sum_{i=1}^{n} Tr(\underline{q}_i(j,p,b)^T \underline{x}_i) = u_{jp}[b]. \tag{4}$$

Proof. Since all semi-random vectors $\underline{r}_i, i = 1, 2, \ldots d+1$ belongs to $\mathbb{F}_2^\rho$, we can invoke linearity of trace and expand the answer $a_i(b), 0 \leq b \leq (m-1)$ as:

$$\begin{aligned} a_i(b) &= Tr\left(\sum_{\ell=1}^{d} \underline{r}_\ell^T (h_{\ell i} + \beta_b \phi_{ji}) x_i\right) + Tr\left(\underline{r}_{d+1}^T \beta_b \phi_{ji} x_i\right) \\ &= \sum_{\ell=1}^{d} \underline{r}_\ell^T Tr\left((h_{\ell i} + \beta_b \phi_{ji}) x_i\right) + \underline{r}_{d+1}^T Tr\left(\beta_b \phi_{ji} x_i\right). \end{aligned}$$

The sum of all answers yields:

$$\begin{aligned} \sum_{i=1}^{n} a_i(b) &= \sum_{\ell=1}^{d} \underline{r}_\ell^T Tr\left(\sum_{i=1}^{n} h_{\ell i} x_i + \beta_b \sum_{i=1}^{n} \phi_{ji} x_i\right) + \\ &\qquad \underline{r}_{d+1}^T Tr\left(\beta_b \sum_{i=1}^{n} \phi_{ji} x_i\right) \\ &= \sum_{\ell=1}^{d} \underline{r}_\ell^T Tr\left(\beta_b u_j\right) + \underline{r}_{d+1}^T Tr\left(\beta_b u_j\right) \\ &= Tr\left(\sum_{\ell=1}^{d} \underline{r}_\ell^T \beta_b u_j + \underline{r}_{d+1}^T \beta_b u_j\right) \\ &= Tr\left(\beta_b u_{jp}\right) \ = \ u_{jp}[b] \end{aligned}$$

The last line follows by Lem. 1. □

4.2 Privacy Guarantees and Confidentiality

Among the three notions of privacy introduced in Sec. 2.3, first we establish privacy on p when $d = (n-k)$. Let $\mathcal{D} = \{i_1, i_2, \cdots, i_d\} \subset [n]$ be the set of curious-but-honest servers. If $\underline{a} = [a_1\ a_2\ \cdots\ a_n] \in \mathbb{F}_q^n$, we denote by $\hat{\underline{a}}$ an $(1 \times d)$ vector obtained by restricting $\underline{a}$ to $\mathcal{D}$, i.e., $\hat{\underline{a}} = [a_{i_1}\ a_{i_2}\ \cdots\ a_{i_d}]$. The same notation extends to matrices as well, i.e., if A is matrix with n columns, then $A_\mathcal{D}$ restricts A to columns indexed by $\mathcal{D}$. With this notation in place, $Q_{\tau,\mathcal{D}}$ is a $(\rho \times d)$-matrix consisting of query vectors sent to nodes in $\mathcal{D}$ at phase τ. Then we can write

$$Q_{\tau,\mathcal{D}} = R_{\tau p} \cdot \Psi_{\tau,\mathcal{D}} \ := \ R_\tau \Phi + E_p \Delta_{\tau j}$$

where $R = \left[\underline{r}_{\tau,1}\ \underline{r}_{\tau,2}\ \cdots\ \underline{r}_{\tau,d}\right]$, E_p is an $(\rho \times d)$ rank-one matrix $\left[\underline{e}_p\ \underline{e}_p\ \cdots\ \underline{e}_p\right]$, Φ is a $(d \times d)$ matrix $\left[\hat{\underline{h}}_1\ \hat{\underline{h}}_2\ \cdots\ \hat{\underline{h}}_d\right]^T$ and $\Delta_{\tau j}$ is a $(d \times d)$ diagonal matrix

$$\Delta_{\tau j} = \begin{bmatrix} \beta_\tau \hat{\phi}_{j1} & & & \\ & \beta_\tau \hat{\phi}_{j2} & & \\ & & \ddots & \\ & & & \beta_\tau \hat{\phi}_{jd} \end{bmatrix}$$

Since Φ is a square submatrix of H_{GRS}, it follows that Φ is invertible. Considering P as a uniform random variable sampled from the set $\{1, 2, \ldots, \rho\}$, we will show that

$$I(Q_{\tau,\mathcal{D}}, \tau \in \Gamma; P) = 0, \tag{5}$$

thus establishing perfect privacy on p. Since we have multiple rounds in the protocol, there is a collection of query random variables, making it distinct from the case considered in [22]. Thus (6) in the following series of equations,

$$\begin{aligned} I(Q_{\tau,\mathcal{D}}, \tau \in \Gamma; P) &= H(Q_{\tau,\mathcal{D}}, \tau \in \Gamma) - \sum_{p=1}^{\rho} \frac{1}{\rho} H(R_\tau \Phi + E_p \Delta_{\tau j}, \tau \in \Gamma | P = p) \\ &= H(R_\tau \Phi + E_P \Delta_{\tau j}, \tau \in \Gamma) - \sum_{p=1}^{\rho} \frac{1}{\rho} H(R_\tau \Phi + E_p \Delta_{\tau j}, \tau \in \Gamma) \\ &= 0 \end{aligned} \tag{6}$$

requires to be established. We show below that $(R_\tau \Phi + E_P \Delta_{\tau j},\ \tau \in \Gamma)$ and $(R_\tau \Phi + E_p \Delta_{\tau j},\ \tau \in \Gamma)$ have the same joint distribution for any p, and that will imply that (6) holds. Let $\underline{a} = (a_\tau, \tau \in \Gamma)$ be value taken by the random vector.

$$\begin{aligned} &\Pr((R_\tau \Phi + E_P \Delta_{\tau j},\ \tau \in \Gamma) = \underline{a}) \\ &= \sum_{p=1}^{\rho} \Pr((R_\tau \Phi + E_P \Delta_{\tau j},\ \tau \in \Gamma) = \underline{a}, P = p) \\ &= \sum_{p=1}^{\rho} \Pr(P = p) \Pr((R_\tau \Phi + E_P \Delta_{\tau j},\ \tau \in \Gamma) = \underline{a} | P = p) \\ &= \sum_{p=1}^{\rho} \frac{1}{\rho} \Pr((R_\tau \Phi + E_p \Delta_{\tau j},\ \tau \in \Gamma) = \underline{a}) \\ &= \sum_{p=1}^{\rho} \frac{1}{\rho} \Pr(R_\tau = (a_\tau - E_p \Delta_{\tau j}) \Phi^{-1}, \tau \in \Gamma) \\ &= \Pr(R_\tau = (a_\tau - E_p \Delta_{\tau j}) \Phi^{-1}, \tau \in \Gamma), \quad \forall p = 1, 2, \ldots, \rho \\ &= \Pr((R_\tau \Phi + E_p \Delta_{\tau j},\ \tau \in \Gamma) = \underline{a}), \quad \forall p = 1, 2, \ldots, \rho. \end{aligned}$$

If set of curious-but-honest servers $\mathcal{D}_1$ is of size $|\mathcal{D}_1| < d$, then we can append $\mathcal{D}_2$ so that $\mathcal{D}_1 \cup \mathcal{D}_2 = \mathcal{D}$ with $|\mathcal{D}| = d$. Then $0 \leq I(Q_{\tau,\mathcal{D}}, \tau \in \Gamma; P) = I(Q_{\tau,\mathcal{D}_1},\ \tau \in \Gamma; P) + I(Q_{\tau,\mathcal{D}_2},\ \tau \in \Gamma; P \mid Q_{\tau,\mathcal{D}_1}, \tau \in \Gamma)$ implying that

$$I(Q_{\tau,\mathcal{D}_1}, \tau \in \Gamma; P) = 0.$$

A similar set of arguments hold true if we pick J as a uniform random variable sampled from $\{1, 2, \ldots, k\}$ leading to

$$I(Q_{\tau,\mathcal{D}_1}, \tau \in \Gamma; J) = 0. \tag{7}$$

That proves perfect privacy of the protocol on j.

The confidentiality of the input data depends on the the size $|\Gamma \setminus \{-1\}|$, in turn depending on the nature of inputs. Out of $2m$ input bits, it is not possible to decode $u_j p[b]$, $v_j p[b]$ where $b \in \{b' \mid u_{jp}[b'] + v_{jp}[b'] = 1\}$ is preserved. Thus on an average, confidentiality of m input bits are preserved.

5 An Application to Blockchain

Consider a synchronous blockchain-based cryptocurrency system with k shards. Suppose there are $(\rho/2)$ accounts associated with each shard. It is straightforward to see that the entire set of transactions at an epoch t can be represented by message matrix $U(t) \in \mathbb{F}_q^{\rho \times k}$, as defined in Sec. 2.1. The top submatrix of $U(t)$ of size $(\rho/2 \times k)$ represents the spent amount. The bottom submatrix of same size represents the received amount. In a PolyShard-ed system, these transaction matrices from all shards are encoded and stored. Then (p, j)-th entry of $U(t)$ denoted by $u_{jp}(t), j \in [k], p \in [\rho/2]$ is the amount spent by account p belonging to shard j. The total amount $D(T)$ spent over epochs 1 to T is clearly

$$D(T) = u_{jp}(1) \boxplus u_{jp}(2) \boxplus \cdots \boxplus u_{jp}(T).$$

Thus iterative private computation of $\boxplus$ without leaking the value of (j, p) to adversaries easily translates to private balance-checking in a PolyShard-ed blockchain system. The protocol also helps in hiding the transaction amounts from the client, while only revealing the balance.

References

1. Androulaki, E., Karame, G., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: Sadeghi, A. (ed.) Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers
2. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014. pp. 459–474 (2014)
3. Biryukov, A., Khovratovich, D., Pustogarov, I.: Deanonymisation of clients in bitcoin P2P network. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014. pp. 15–29 (2014)
4. Bowe, S., Chiesa, A., Green, M., Miers, I., Mishra, P., Wu, H.: Zexe: Enabling decentralized private computation. IACR Cryptol. ePrint Arch. p. 962 (2018), `https://eprint.iacr.org/2018/962`
5. Bünz, B., Agrawal, S., Zamani, M., Boneh, D.: Zether: Towards privacy in a smart contract world. In: Financial Cryptography and Data Security - 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers
6. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA. pp. 315–334 (2018)
7. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: Proceedings of IEEE 36th Annual Foundations of Computer Science. pp. 41–50. IEEE (1995)
8. Fanti, G.C., Venkatakrishnan, S.B., Bakshi, S., Denby, B., Bhargava, S., Miller, A., Viswanath, P.: Dandelion++: Lightweight cryptocurrency networking with formal anonymity guarantees. In: Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2018, Irvine, CA, USA, June 18-22, 2018. pp. 5–7 (2018)
9. Freij-Hollanti, R., Gnilke, O.W., Hollanti, C., Karpuk, D.A.: Private information retrieval from coded databases with colluding servers. SIAM Journal on Applied Algebra and Geometry **1**(1), 647–664 (2017)

10. Henry, R., Herzberg, A., Kate, A.: Blockchain access privacy: Challenges and directions. IEEE Security Privacy **16**(4), 38–45 (2018). https://doi.org/10.1109/MSP.2018.3111245
11. Jäschke, A., Armknecht, F.: (Finite) field work: Choosing the best encoding of numbers for FHE computation. Cryptology ePrint Archive, Report 2017/582 (2017), `https://ia.cr/2017/582`
12. Kosba, A.E., Miller, A., Shi, E., Wen, Z., Papamanthou, C.: Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016. pp. 839–858 (2016)
13. Li, S., Yu, M., Yang, C., Avestimehr, A.S., Kannan, S., Viswanath, P.: Polyshard: Coded sharding achieves linearly scaling efficiency and security simultaneously. IEEE Trans. Inf. Forensics Secur. **16**, 249–261 (2021)
14. Lidl, R., Niederreiter, H.: Finite Fields. Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2 edn. (1996). https://doi.org/10.1017/CBO9780511525926
15. Matetic, S., Wüst, K., Schneider, M., Kostiainen, K., Karame, G., Capkun, S.: BITE: bitcoin lightweight client privacy using trusted execution. In: Heninger, N., Traynor, P. (eds.) 28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019. pp. 783–800. USENIX Association (2019)
16. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of bitcoins: characterizing payments among men with no names. ACM Commun. **59**(4), 86–93 (2016)
17. Mousavi, M.H., Maddah-Ali, M.A., Mirmohseni, M.: Private inner product retrieval for distributed machine learning. In: IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019. pp. 355–359 (2019)
18. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2009), `http://www.bitcoin.org/bitcoin.pdf`
19. Obead, S.A., Lin, H.Y., Rosnes, E., Kliewer, J.: Private linear computation for noncolluding coded databases. IEEE Journal on Selected Areas in Communications pp. 1–1 (2022). https://doi.org/10.1109/JSAC.2022.3142362
20. Ober, M., Katzenbeisser, S., Hamacher, K.: Structure and anonymity of the bitcoin transaction graph. Future Internet **5**(2), 237–250 (2013)
21. Qin, K., Hadass, H., Gervais, A., Reardon, J.: Applying private information retrieval to lightweight bitcoin clients. In: 2019 Crypto Valley Conference on Blockchain Technology (CVCBT). pp. 60–72 (2019)
22. Sasidharan, B., Viterbo, E.: Private data access in blockchain systems employing coded sharding. In: 2021 IEEE International Symposium on Information Theory (ISIT). pp. 2684–2689 (2021). https://doi.org/10.1109/ISIT45174.2021.9517900
23. Shah, N.B., Rashmi, K., Ramchandran, K.: One extra bit of download ensures perfectly private information retrieval. In: 2014 IEEE International Symposium on Information Theory. pp. 856–860 (2014)
24. Tajeddine, R., Gnilke, O.W., El Rouayheb, S.: Private information retrieval from mds coded data in distributed storage systems. IEEE Transactions on Information Theory **64**(11), 7081–7093 (2018)
25. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper **151**, 1–32 (2014)