

# PrivHChain: Monitoring the Supply Chain of Controlled Substances with Privacy-Preserving Hierarchical Blockchain

**Abstract**—With rapidly increasing drug abuse across the world, it is imperative to monitor their supply chain with sufficient transparency. Blockchain is a common solution for achieving transparency in supply chain monitoring, but it does not have sufficient throughput for large-scale supply chains. It is challenging to achieve throughput and privacy simultaneously because complex dependencies among the supply chain events and the need for aggregation both make the application of ZKP challenging. We present PrivHChain, a privacy-preserving hierarchical blockchain that preserves transaction privacy even against blockchain peers while allowing them to verify record consistencies. This is enabled by novel modeling of supply chain events which makes it possible to use novel efficient zero-knowledge protocol schemes to verify the complex dependencies. Novel aggregation techniques are proposed to enable the proof aggregation, and the proofs are used to design monitoring protocols. PrivHChain is implemented and validated with extensive experiments and simulations. The results indicate that (i) the extra overhead of encryption and ZKP schemes is acceptable or negligible, and (ii) the throughput is improved by up to 5 times in simulations even with all the encryption/ZKP schemes.

## I. INTRODUCTION

Controlled substances are strictly regulated because of their potential abuse, which significantly impacts public health and safety. The laws of the United States (U.S.) and the Republic of Korea (RoK) require that such substances be handled only by authorized entities with legitimate purposes [1]–[3], and the quantity must be strictly monitored. These laws require licensed manufacturers, medical facilities, researchers, and other relevant entities in the supply chain of controlled substances to report their activities. It is imperative that the supply chain of controlled substances is monitored with *integrity*, *transparency*, and *traceability*. Without such guarantees, it is hard to keep track of how drug-related activities are reported and monitored, and this would have significant impacts on societies considering the rapid and widespread drug overdose and abuse in the U.S. and the RoK [4], [5].

Blockchain has been studied extensively for supply chains because of its tamper-proofness and transparency. However, unlike other supply chains, sensitive information is involved in controlled substance supply chains (e.g., usage of drugs, entity using drugs). Therefore, we should consider privacy-preserving way, e.g. zero-knowledge proof. Furthermore, tens of millions of reports are uploaded annually in controlled substance supply chains (103.38M/yr in the RoK [6]; 16.53M/yr in North Carolina [7]; 67.99M/yr in California [8]). Approximately, permissionless(permissioned) blockchain throughput is

on the order of 1(10)Btx/yr [9], which appears sufficient at first glance. However, the cost of processing simple cryptocurrency transactions is already high (\$250/KB [10]), and controlled substances require more complex verification. We conjecture that it is impractical to rely on a single blockchain to support the entire supply chain in any country/state.

In this paper, we present a novel way to model supply chain events as transactions on blockchains in a way that represents event dependencies with set relations that are amenable to ZKPs. Then, we present several ZKP schemes (existing ones and our novel ones) for hierarchical blockchain and describe how to monitor and verify the supply chain in a privacy-preserving manner. Our work relies on the ZKP schemes for proving authorization/quantities of controlled substances and hierarchical blockchain architecture with two levels for throughput, where the upper level is called *global blockchain* and the lower level is called *local blockchain*.

Our contributions are as follows:

**Privacy-Preserving Hierarchical Blockchain.** We propose the first hierarchical blockchain design PrivHChain that ensures transaction privacy even against blockchain peers. Our blockchain enables blockchain peers to verify the consistency of encrypted records with dependencies within and across different hierarchies. This is achieved through our novel event modeling, making relying on efficient ZKP schemes possible. **Aggregation for Throughput.** We introduce a novel protocol to aggregate proofs in a local blockchain to one proof in the global blockchain. Peers in the global blockchain only need to verify the aggregated proof to ensure the consistency of all relevant records in the local blockchain. **Extensive Experimentation.** We implemented PrivHChain and conducted extensive experiments. The transaction generation process takes approximately 6s, whereas verification transactions require only 17ms. Furthermore, the hierarchical structure improves the throughput of the blockchain by up to 5 times even with ZKPs, and the resulting throughput is much higher than what is needed for the number of records in controlled substance supply chains in the U.S. and the RoK. Anonymized source code is provided for reproducibility [11].

## II. BACKGROUNDS, MODELS, AND GOAL

### A. Supply Chain of Controlled Substances

Both the U.S. and RoK's controlled substance laws obligate the entities involved in the distribution process to report almost all activities, including the handlers' IDs and counterparties'

No.	Handler	Counterparty	Day of Handling	Day of Reporting	work	serial number
...	...	...	...	...	...	...
829	NIMS2019	CTPS2022	2022-08-29	2022-09-04	transfer	A10001
830	NIMS2019	CTPS2022	2022-08-29	2022-09-04	transfer	A10002
831	NIMS2019	CTPS2022	2022-08-29	2022-09-04	transfer	A10003
...	...	...	...	...	...	...




Fig. 1: An example report log

IDs, the amount of controlled substance, and the inventory status after each supply chain event [1], [3]. Additionally, some related laws of both countries require the reporting of a unique serial number assigned at the package level [2], [3], [12].

While the current centralized electronic reporting system maintains confidentiality through encryption, only a few centralized entities, such as the DEA in the U.S., the MFDS, and the RoK, are able to verify the reported information, making it difficult to ensure that all reported information is consistent and accurate. Increasing the number of entities is not a viable option because of the sensitivity of the related records. Therefore, we propose this study to verify the internal consistency of reported logs while maintaining their confidentiality.

We assume that the reported logs accurately reflect real-world behavior, and we do not address the verification of real-world events within the scope of this work. This assumption is reasonable due to the strict surveillance of each country. The main goal of this work is to verify the internal consistency of reported logs in cyberspace.

### B. Hierarchical blockchain

Blockchain-based systems suffer from scalability bottlenecks caused by the consensus protocol. Hierarchical blockchain uses a tree-like multi-layered architecture to enhance scalability and parallelism within the network. It facilitates simultaneous processing in different chains, and transactions in lower-level chains are aggregated and committed to corresponding upper-level chains. We use the idea of a hierarchical blockchain system to cope with the scalability needs of existing supply chain organizations. Our focus is on the challenge of achieving privacy and verifiability in the hierarchical blockchain in which transactions have complex hierarchical dependencies. The common challenges of cross-domain transactions across different chains and levels are orthogonal problems and thus are out of this paper's scope.

### C. System and Adversary Models

In our scenario, we consider four types of entities: global managers, permitted participants, local managers, and blockchain peers. We describe the role of each entities.

- *Global managers* are the entities that authorize participants to handle controlled substances. A given authorization includes the identifier (ID) of the participant, the type of substance, and membership proof showing that this participant belongs to a list of authorized participants. Note that the global manager is somewhat centralized; however, this is unavoidable

in controlled substance regulations, and our ultimate goal is not to achieve decentralization.

- *Permitted Participants* can then generate transactions for the substance using their IDs and membership proofs. Each ID and membership proof are not transferable and must be kept secret from anyone else.

- *Local Managers* are subentities of global managers who are in charge of each local network (e.g., DEA/MFDS agents in charge of some district/area). Their role is to summarize logs and proofs in the local blockchain. After summarization, the local managers upload summarized proofs to higher-level networks. Local managers can access transactions in local networks, meaning that they know the IDs of participants in their local network, which is reasonable because participants must be authorized.

- *Blockchain Peers* are the entities that maintain the blockchain. They verify the submitted logs (e.g., authorization verification, quantity verification) and log them in the blockchain. In the real scenario, blockchain peers may include auditors hired by the central manager, such as DEA/MFDS. As the peers are pre-verified and compensated by the central manager, it is expected that they will not engage in malicious behavior and will be motivated to maintain the network.

Global/local managers are trusted entities that authorize and regulate controlled substances. Participants and blockchain peers are semi-honest adversaries who follow protocol specifications but try to infer others' secrets.

### D. Security Model

In our scenario, transactions should simultaneously support the confidentiality of private data and verifiability transaction validity. To define our security model, we introduce *Private Transaction Schemes*  $\text{PrvTx}$ , which consist of two algorithms:  $\text{TxGen}$  and  $\text{TxVerify}$ .  $\text{TxGen}$  is a transaction generation algorithm that takes the public parameter  $\text{pp}$  and the data  $M = (M_p, M_s)$ , where  $M_s$  and  $M_p$  represent private (secret) and public data, respectively. The former needs to be concealed, but the latter does not. Based on these inputs,  $\text{TxGen}$  outputs a transaction  $(M_p, C, \pi)$ , where  $C$  is a commitment to the private data  $M_s$ , and  $\pi$  is a validity proof for  $\mathcal{R}_{tx}$ , which is a set including all valid pairs  $(M_p, C)$ .  $\text{TxVerify}$  is a transaction verification algorithm that takes the public parameter  $\text{pp}$  and the transaction  $(M, C, \pi)$ , and outputs 1 (accept) or 0 (reject).

The security of transactions relies on  $\text{PrvTx} = (\text{TxGen}, \text{TxVerify})$ . To provide privacy and integrity,  $\text{PrvTx}$  must satisfy the properties of *transaction indistinguishability* and *transaction verifiability* respectively.

**Definition 1** (Transaction indistinguishability). A private transaction scheme  $(\text{TxGen}, \text{TxVerify})$  is said to be *indistinguishability*, if for all PPT adversaries  $\mathcal{A}$ , the following probability is negligible with respect to security parameter  $\lambda$ .

$$\Pr \left[ b = b' \mid \begin{array}{l} \text{pp} \leftarrow \text{Setup}(1^\lambda); M_p, M_{s,1}, M_{s,2} \leftarrow \mathcal{A}(\text{pp}); \\ b \xleftarrow{\$} \{0, 1\}, \text{Tx}_i \leftarrow \text{TxGen}(\text{pp}, (M_p, M_{s,i})); \\ b' \leftarrow \mathcal{A}(\text{pp}, \text{Tx}_i) \end{array} \right] - \frac{1}{2}$$

**Definition 2** (Transaction Verifiability). A privacy transaction scheme  $(\text{TxGen}, \text{TxVerify})$  is said to be verifiable, if for all PPT adversaries  $\mathcal{A}$ , the following probabilities hold, where  $\text{negl}(\lambda)$  is negligible function of security parameter  $\lambda$ .

$$\Pr [1 \leftarrow \text{TxVerify}(\text{pp}, M_p, C, \pi) \mid (M, C) \in \mathcal{R}_{tx}, \text{pp} \leftarrow \text{Setup}(1^\lambda)] = 1$$

$$\Pr \left[ 1 \leftarrow \text{TxVerify}(\text{pp}, M_p, C, \pi') \mid \begin{array}{l} \text{pp} \leftarrow \text{Setup}(1^\lambda); \\ \wedge (M, C) \notin \mathcal{R}_{tx} \end{array} \mid (M, C, \pi') \leftarrow \mathcal{A}(\text{pp}) \right] < \text{negl}(\lambda)$$

### III. ALGORITHMS AND PROTOCOLS

We introduce widely used cryptographic tools in Section III-A-III-B. Next, we propose novel ZKP protocols for set relations in Sections III-C and III-D. All are building blocks of the supply chain of controlled substances with privacy-preserving hierarchical blockchain in Section IV.

**Notations.** Let  $\mathcal{G}$  be a bilinear group generator that takes the security parameter  $\lambda$  and returns a  $\lambda$  bits prime  $p$ , multiplicative groups  $\mathbb{G}_1, \mathbb{G}_2$  and  $\mathbb{G}_T$  of prime order  $p$  with a bilinear map  $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$  and generators  $g_1$  and  $g_2$  of  $\mathbb{G}_1$  and  $\mathbb{G}_2$  respectively. We denote  $\mathbb{Z}_p$  as finite field with size  $p$ . We denote a characteristic polynomial  $A(X)$  for the set  $A$  such that  $A(X) = \prod_{a \in A} (X + a)$ . From this notation,  $A(s)$  is the evaluation of  $A(X)$  at  $s \in \mathbb{Z}_p$ . We also denote  $[n] = \{1, 2, \dots, n\}$  for some natural number  $n$ .

For  $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{Z}_p^n$ ,  $\mathbf{h} = (h_1, \dots, h_n) \in \mathbb{G}_1^n$ , and  $\mathbf{w} = (w_1, \dots, w_n) \in \mathbb{G}_2^n$ , we denote multi-exponentiation as  $\mathbf{g}^{\mathbf{x}} = \prod_{i \in [n]} g_i^{x_i}$ , inner-pairing product as  $\mathbf{E}(\mathbf{h}, \mathbf{w}) := \prod_{i \in [n]} e(h_i, w_i)$ . We use  $\circ$  to express point-wise multiplication, that is  $\mathbf{v} \circ \mathbf{w} = (v_1 w_1, \dots, v_n w_n)$  for  $\mathbf{v}, \mathbf{w} \in \mathbb{G}^n$ .

**Cryptographic Assumptions.** We make two widely-known cryptographic assumptions: q-Strong Diffie-Hellman (q-SDH) assumption [13] and Symmetric eXternal Diffie-Hellman (SXDH) assumption [14]. Due to the space limit, we show full descriptions in the anonymized full version [15].

**Interactive Proof System for Relation  $\mathcal{R}$ .** A set  $\mathcal{R}$  is a polynomial-time verifiable relation consisting of statements and witnesses, denoted by  $x$  and  $w$ , respectively. An interactive proof system for the relation  $\mathcal{R}$  consists of three algorithms:  $\mathcal{K}, \mathcal{P}, \mathcal{V}$ , the key generation, Prover, and Verifier. The  $\mathcal{K}$  algorithm takes the security parameter  $\lambda$  and outputs the public parameters  $\text{pp}$ , which serve as inputs for both  $\mathcal{P}$  and  $\mathcal{V}$ . The  $\mathcal{P}$  and  $\mathcal{V}$  take  $(\text{pp}, x)$  and  $(\text{pp}, x; w)$ , respectively, and interactively generate a transcript. At the end of the transcript,  $\mathcal{V}$  outputs either *reject* or *accept*. The purpose of  $\mathcal{P}$  is to obtain an *accept* response from  $\mathcal{V}$ , while the purpose of  $\mathcal{V}$  is to check that there exists a witness  $w$  such that  $(x, w) \in \mathcal{R}$ .

**Fiat-Shamir Transform.** A public coin interactive proof is one in which the output of the verifier, obtained through interaction, is uniformly random and independent of the prover's messages. Fiat and Shamir [16] proposed a method to transform any public coin interactive proof into a noninteractive proof using the random oracle model. For a given interactive proof system  $\text{Proof}(\mathcal{K}, \mathcal{P}, \mathcal{V})$ , we denote the noninteractive prover algorithm, along with the proof  $\Pi_{\text{Proof}}$ , derived from the protocol as  $\Pi_{\text{Proof}} \leftarrow \text{Proof}(\text{pp}, x, w)$ .

#### A. Bilinear Accumulator Scheme

An accumulator scheme is a set-commitment scheme that supports efficient element addition/deletion and (non-)membership checks in an accumulator representing a set. Our work relies on bilinear trapdoor accumulators [17], [18], which can handle any subset of  $\mathbb{Z}_p$ .

**Definition 3.** A Bilinear Pairing Cryptographic Accumulator can be defined with the following probabilistic polynomial time algorithms:

- $\text{pp} \leftarrow \text{Acc.Setup}(1^\lambda, N, s)$ : Given the security parameter  $\lambda$ , maximum capacity  $N$ , and trapdoor  $s$ , it runs bilinear group generator  $\mathcal{G}$ . Then, returns public parameter  $\text{pp} = \{p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \{g_1^{s^i}, g_2^{s^i}\}_{i=0}^N\}$ :  $\lambda$ -bit prime  $p$ ,  $p$ -order groups  $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$  with bilinear map  $e$ , and power of generators  $g_1^{s^i} \in \mathbb{G}_1$  and  $g_2^{s^i} \in \mathbb{G}_2$  for  $i = 0$  to  $N$
- $C_A \leftarrow \text{Acc.Com}(\text{pp}, A)$ : It takes  $\text{pp}$ , a set  $A$  of size  $k(\leq N)$ , it returns a commitment  $C_A = g_2^{A(s)}$ .
- $C_{A'} \leftarrow \text{Acc.Add}(\text{pp}, C_A, A, I)$ : It takes  $\text{pp}$ , two disjoint sets  $A$  and  $I$ , commitment  $C_A$  of  $A$ , it returns a commitment  $C_{A'} = g_2^{A'(s)}$  where  $A' = A \cup I$  if size of  $A'$  is at most  $N$ ,  $\perp$  otherwise.
- $C_{A'} \leftarrow \text{Acc.Del}(\text{pp}, C_A, A, I)$ : It takes  $\text{pp}$ , a set  $A$  and its commitment  $C_A$  and subset  $I$  of  $A$ , and returns commitment  $C_{A'} = g_2^{A'(s)}$  where  $A' = A \setminus I$ .
- $w_y \leftarrow \text{Acc.MemWitGen}(\text{pp}, A, y)$ : It takes  $\text{pp}$ , a set  $A$ , an element  $y$  of  $A$ . It returns membership witness  $w_y = g_2^{A(s)/y+s}$  for  $y \in A$ .
- $0/1 \leftarrow \text{Acc.MemVF}(\text{pp}, C_A, y, w_y)$ : It takes  $\text{pp}$ , a set commitment  $C_A$ , an element  $y$  and its membership witness  $w_y$ . It returns 1 if and only if  $e(C_A, g_1) = e(w_y, g_2^y g_2^s)$ .
- $\bar{w}_y := (w_1, w_2) \leftarrow \text{Acc.NonMemWitGen}(\text{pp}, A, y)$ : It takes  $\text{pp}$ , a set  $A$ , an element  $y$  which does not belong to  $A$ . It returns non-membership witness  $\bar{w}_y := (w_1, w_2) = (g_1^{\alpha(s)}, g_1^{\beta(s)})$  such that  $\alpha(X) \cdot A(X) + \beta(X) \cdot (y + X) = 1$ .
- $0/1 \leftarrow \text{Acc.NonMemVF}(\text{pp}, C_A, y, \bar{w}_y)$ : It takes  $\text{pp}$ , a set commitment  $C_A$ , an element  $y$  and its non-membership witness  $\bar{w}_y = (w_1, w_2)$ . It returns 1 if and only if  $e(w_1, C_A) e(w_2, g_2^y g_2^s) = e(g_1, g_2)$ .

Under the  $q$ -SDH and SXDH assumptions, the accumulator achieves soundness [17], [18], meaning that it is infeasible to generate both membership and non-membership witnesses simultaneously.

#### B. Zero Knowledge Proof Systems

We first provide a brief introduction to the existing proof systems. Detailed descriptions of the systems and security proofs are available in the anonymized full version [15].

**Schnorr's Protocol.** Schnorr's protocol [19] is a fundamental ZK proof of knowledge (PoK) system used in identification. In our system, we use Schnorr's protocol to claim knowledge of an ID. We apply the well-known generalized Schnorr protocol for  $n$  messages, denoted as  $\text{PoK}_n$ .

**ZK Inner Pairing Product Proof.** An ZK inner pairing product proof (ZKIPP) is a PoK of the inner pairing product relation [20]–[22]. ZKIPP features logarithm size of the number of pairings, which helps to aggregate transactions.

**Polynomial Commitment Scheme.** A polynomial commitment scheme (PCS) allows a prover to produce a commitment  $C$  to polynomial  $p(X) \in \mathbb{Z}_p[X]$ , and later open polynomial evaluation  $z$  at some point  $y$  with proof  $\pi$  from open algorithm  $\text{PC.Open}$ . We use KZG PCS [23] for aggregating transactions.

#### C. ZK Batch Proof for Hidden Accumulator

We describe a novel ZK batch membership proof on an accumulator with a hiding property, which we refer to as a

*hidden accumulator.* A hidden accumulator is essential for transaction privacy because if it is not hidden, attackers can find its member items through brute-force search. Srinivasan et al. proposed a ZK batch (non-)membership proof [18], but they did not consider the hiding property of the accumulator. We construct a ZK batch (non-)membership proof for hidden accumulators by introducing random hiding factors.

**Public Parameter.** To construct a ZK argument system, we add more common reference strings (CRS)  $g, h \in \mathbb{G}_1$  and  $h_2 \in \mathbb{G}_2$ . Then, the public parameter  $pp := \{p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \{g_1^{s_i}, g_2^{s_i}\}_{i=0}^N, g, h, h_2\}$ .

**Hidden Accumulator.** Ghosh et al. [24] proposed a ZK accumulator for privacy set operations. They introduce randomness by multiplying the cover polynomial  $A(X)$  with  $r$ , which ensures indistinguishability. However, this approach can pose a challenge in guaranteeing knowledge of the membership set. Therefore, we take a different approach: multiply the commitment to randomness on a new uniform random  $h_2 \xleftarrow{\$} \mathbb{G}_2$ . Our hidden accumulator is generated as follows:

- $C_A \leftarrow \text{HAcc.Commit}(pp, A, r)$ : It takes  $pp$ , a set  $A$  of size  $k(\leq N)$  and randomness  $r$ , it returns  $C_A = g_2^{A(s)} h_2^r$ . Because  $r$  is private, the hidden accumulator scheme does not follow the membership proof system in Definition 3. We propose a ZK membership proof system for the hidden accumulator scheme.

**ZK Batch Membership Proof.** Let  $C_I = g_2^{I(s)} h_2^{r_I}$  be a commitment to batch set  $I$  with randomness  $r_I$ , and  $C_A = g_2^{A(s)} h_2^{r_A}$  be a hidden accumulator of set  $A$  with randomness  $r_A$ . In this case, a membership witness  $w_I = g_2^{A(s)/I(s)}$  satisfies the following equation:

$$e(w_I, C_I) \cdot e(w_I, h_2)^{-r_I} = e(g_1, C_A) \cdot e(g_1, h_2)^{-r_A}$$

For ZKness, we construct a ZKP system for the following relation.

$$\mathcal{R}_{\text{ZKMP}} = \left\{ \begin{array}{l} (pp, C_I, C_A; w_I, r_I, r_A) : \\ e(w_I, C_I) \cdot e(w_I, h_2)^{-r_I} = e(g_1, C_A) \cdot e(g_1, h_2)^{-r_A} \end{array} \right\} \quad (1)$$

Following existing work [18], we add hiding factors for  $w_I$  and apply generalized Schnorr's protocol.

In the similar manner of ZKMP construction, we can construct ZK non-membership proof ZKNMP system. The description of ZKNMP is deferred to the anonymized full version [15]

**Theorem 1.** *The ZKMP protocol in Fig. 2 provides completeness, soundness and perfect ZKness under SXDH assumption. (Proof available in the anonymized full version [15].)*

#### D. ZK Set-Split Proof

In this section, we propose *ZK set-split proof* (ZKSP). The ZKSP is a ZK argument for the set-split relation in Equation (2); the set  $I$  and  $J$  are split from  $A$ , that is  $I \cup J = A$  and  $I \cap J = \emptyset$ . To prove the relation, we apply ZKMP for the inclusion relations  $I \subset A$  and  $J \subset A$ . Next, we check that the discrete logarithm of  $w_J$  is equal to the evaluation  $I(s)$ . Notice that the membership witness  $w_J$  is a commitment to

ZKMP(pp, C\_I, C\_A; w\_I, r\_I, r\_A)

- $\mathcal{P}$  picks  $\tau_1, \tau_2, r_{r_I}, r_{r_A}, r_{\tau_1}, r_{\tau_2}, r_{\delta_1}, r_{\delta_2} \xleftarrow{\$} \mathbb{Z}_p$  and sets  $\delta_1 = r_I \tau_1$  and  $\delta_2 = r_I \tau_2$
- $\mathcal{P}$  sends:
  - $P_1 = g_1^{\tau_1} g^{\tau_2}, P_2 = w_I g^{\tau_1}$
  - $R_1 = g_1^{r_{\tau_1}} g^{r_{\tau_2}}, R_2 = P_1^{r_{r_I}} g_1^{-r_{\delta_1}} g^{-r_{\delta_2}}$
  - $R_3 = \frac{e(g, C_I)^{r_{\tau_1}} e(P_2, h_2)^{r_{r_I}}}{e(g, h_2)^{r_{\delta_1}} e(g_1, h_2)^{r_{r_A}}}$
- $\mathcal{V}$  sends  $c \xleftarrow{\$} \mathbb{Z}_p$
- $\mathcal{P}$  sends:
  - $s_{r_I} = r_{r_I} + c r_I, s_{r_A} = r_{r_A} + c r_A$
  - $s_{\tau_1} = r_{\tau_1} + c \tau_1, s_{\tau_2} = r_{\tau_2} + c \tau_2$
  - $s_{\delta_1} = r_{\delta_1} + c \delta_1, s_{\delta_2} = r_{\delta_2} + c \delta_2$
- $\mathcal{V}$  checks:
  - $R_1 = P_1^{-c} g_1^{s_{\tau_1}} g^{s_{\tau_2}}, R_2 = P_1^{s_{r_I}} g_1^{-s_{\delta_1}} g^{-s_{\delta_2}}$
  - $R_3 \cdot \left( \frac{e(P_2, C_I)}{e(g_1, C_A)} \right)^c = \frac{e(g, C_I)^{s_{\tau_1}} e(P_2, h_2)^{s_{r_I}}}{e(g, h_2)^{s_{\delta_1}} e(g_1, h_2)^{s_{r_A}}}$

Fig. 2: ZK membership Proof (ZKMP)

ZKSP(pp, C\_I, C\_J, C\_A; w\_I, w\_J, r\_I, r\_J, r\_A)

- $\mathcal{P}$  picks  $r_{r_I}, r_{r_J}, r_{r_A}, (\tau_i, r_{\tau_i}, r_{\delta_i})_{i=1}^4 \xleftarrow{\$} \mathbb{Z}_p$  and sets  $\delta_1 = r_I \tau_1, \delta_2 = r_I \tau_2, \delta_3 = r_J \tau_3, \delta_4 = r_J \tau_4$
- $\mathcal{P}$  sends:
  - $P_1 = g_1^{\tau_1} g^{\tau_2}, P_2 = w_I g^{\tau_1}, P_3 = g_1^{\tau_3} g^{\tau_4}, P_4 = w_J g^{\tau_3}$
  - $R_1 = g_1^{r_{\tau_1}} g^{r_{\tau_2}}, R_2 = P_1^{r_{r_I}} g_1^{-r_{\delta_1}} g^{-r_{\delta_2}}$
  - $R_3 = g_1^{r_{\tau_3}} g^{r_{\tau_4}}, R_4 = P_3^{r_{r_J}} g_1^{-r_{\delta_3}} g^{-r_{\delta_4}}$
  - $R_5 = \frac{e(g, C_I)^{r_{\tau_1}} e(P_2, h_2)^{r_{r_I}}}{e(g, h_2)^{r_{\delta_1}} e(g_1, h_2)^{r_{r_A}}}, R_6 = \frac{e(g, C_J)^{r_{\tau_3}} e(P_4, h_2)^{r_{r_J}}}{e(g, h_2)^{r_{\delta_3}} e(g_1, h_2)^{r_{r_A}}}$
  - $R_7 = e(g_1, h_2)^{r_{r_I}} \cdot e(g, g_2)^{-r_{\tau_3}}$
- $\mathcal{V}$  sends  $c \xleftarrow{\$} \mathbb{Z}_p$
- $\mathcal{P}$  sends:
  - $s_{r_I} = r_{r_I} + c r_I, s_{r_J} = r_{r_J} + c r_J, s_{r_A} = r_{r_A} + c r_A$
  - $s_{\tau_i} = r_{\tau_i} + c \tau_i, s_{\delta_i} = r_{\delta_i} + c \delta_i$  for all  $i \in [4]$
- $\mathcal{V}$  checks:
  - $R_1 = P_1^{-c} g_1^{s_{\tau_1}} g^{s_{\tau_2}}, R_2 = P_1^{s_{r_I}} g_1^{-s_{\delta_1}} g^{-s_{\delta_2}}$
  - $R_3 = P_3^{-c} g_1^{s_{\tau_3}} g^{s_{\tau_4}}, R_4 = P_3^{s_{r_J}} g_1^{-s_{\delta_3}} g^{-s_{\delta_4}}$
  - $R_5 \cdot \left( \frac{e(P_2, C_I)}{e(g_1, C_A)} \right)^c = \frac{e(g, C_I)^{s_{\tau_1}} e(P_2, h_2)^{s_{r_I}}}{e(g, h_2)^{s_{\delta_1}} e(g_1, h_2)^{s_{r_A}}}$
  - $R_6 \cdot \left( \frac{e(P_4, C_J)}{e(g_1, C_A)} \right)^c = \frac{e(g, C_J)^{s_{\tau_3}} e(P_4, h_2)^{s_{r_J}}}{e(g, h_2)^{s_{\delta_3}} e(g_1, h_2)^{s_{r_A}}}$
  - $R_7 \cdot \left( \frac{e(g_1, C_I)}{e(P_4, g_2)} \right)^c = e(g_1, h_2)^{s_{r_I}} \cdot e(g, g_2)^{-s_{\tau_3}}$

Fig. 3: ZK Set-Split Proof (ZKSP)

the polynomial  $A(X)/J(X)$ . If  $I$  is the complement set of  $J$ , then the equation  $I(X) = A(X)/J(X)$  should hold.

$$\mathcal{R}_{\text{ZKSP}} = \left\{ \begin{array}{l} (pp, C_I, C_J, C_A; w_I, w_J, r_I, r_J, r_A) : \\ \frac{e(w_I, C_I)}{e(w_I, h_2)^{r_I}} = \frac{e(w_J, C_J)}{e(w_J, h_2)^{r_J}} = \frac{e(g_1, C_A)}{e(g_1, h_2)^{r_A}} \wedge \\ e(g_1, C_I) \cdot e(g_1, h_2)^{-r_I} = e(w_J, g_2) \end{array} \right\} \quad (2)$$

**Theorem 2.** *The ZKSP protocol in Fig. 3 provides completeness, soundness and perfect ZKness under SXDH assumption. (Proof available in the anonymized full version [15].)*

#### IV. PRIVACY-PRESERVING HIERARCHICAL BLOCKCHAIN FOR SUPPLY CHAIN

With building blocks in Section III, we introduce a privacy-preserving hierarchical blockchain for controlled substance supply chains. The principal idea is to mandate that every participant must commit logs to their local blockchains in the form of blockchain transactions (Tx) soon after the real-world supply chain events. To provide public verifiability without leaking sensitive information, we provide some ZK proofs that are included transactions.

### A. Proofs in Transactions

**Proof of Authorization.** Only the participants authorized by the global manager are allowed to participate in events involving controlled substances. That is, the participants contains proof of authorization per each transaction.

To setup the supply chain, a global manager, who has a list  $\mathcal{I}$  of permitted participants, generates an accumulator  $C_{Aut}$  of the list and then issues ID  $id$  and corresponding *authorization witness*  $w_{id}$  as following:  $C_{Aut} \leftarrow \text{Acc.Commit}(\text{pp}, \mathcal{I}, 0)$  and  $w_{id} \leftarrow \text{Acc.MemWitGen}(\text{pp}, \mathcal{I}, id)$  for  $id \in \mathcal{I}$ .

In order to generate Tx's, each participant related a transaction picks random  $r_{id} \xleftarrow{\$} \mathbb{Z}_p$  and then computes a commitment  $C_{id}$  and ZK proof of authorization (PoA,  $\prod_{\text{PoA}}$ ) as follows:  $C_{id} := g_2^{(id+s)} h_2^{r_{id}} \leftarrow \text{Commit}(\text{pp}, id, r_{id})$  and  $\prod_{\text{PoA}} \leftarrow \text{ZKMP}(\text{pp}, C_{id}, C_{Aut}; w_{id}, r_{id}, 0)$  respectively.

**Proof of ID-knowledge.** To prove the dependency of an input and an output in two Tx's (e.g., some substances being delivered is picked up at a manufacturer, some substance used in prescription came from a lot delivered by a carrier), a participant needs to prove that s/he has the IDs hidden in the output's ID commitment. This can be achieved by providing a ZK proof of ID-knowledge (PoID,  $\prod_{\text{PoID}}$ ), which is derived from the ID commitment in the previous Tx as  $\prod_{\text{PoID}} \leftarrow \text{PoK}_2(\text{pp}, C_I/g_2^s, id, r)$ , where  $C_I$  is a ID commitment to  $id$  with randomness  $r$ .

**Proof of Quantity.** According to the laws in the U.S. and RoK, all SNs of related controlled substances should be recorded for each supply chain event, so we treat SNs as tokens in the blockchain and use them to prove the consistency of quantities.

Contrary to IDs, SNs need to be shared among the participants in the relevant supply chain events. Therefore, a participant may infer the flow of SNs if SNs are not protected securely. To deal with this issue, we use our novel hidden accumulator scheme (Section III-C) and ZKP systems. Let  $S$  be for the sender's SN before transmission, and  $S_1$  and  $S_2$  be assigned the participant's SN by their choice. For example,  $S_1$  is for the SN being transmitted, and  $S_2$  is for the rest of the SN. The sender generates the ZK proof of quantity (PoQ,  $\prod_{\text{PoQ}}$ ) to prove the quantity equalities in the inflow and outflow of local regions, e.g.  $S = S_1 \cup S_2$  and  $S_1 \cap S_2 = \emptyset$ . To generate  $\prod_{\text{PoQ}}$ , the sender picks randomness  $\delta_1, \delta_2 \xleftarrow{\$} \mathbb{Z}_p$  and then compute commitments and membership witnesses  $A_1, A_2$  and  $w_1, w_2$  for  $S_1, S_2$  using  $\delta_1, \delta_2$  respectively. And then run  $\prod_{\text{PoQ}} \leftarrow \text{ZKSP}(A_1, A_2, A; w_1, w_2, \delta_1, \delta_2, \delta)$ .

### B. Modeling Events as Blockchain Transactions

Our system resembles the unspent transaction output (UTXO) model in cryptocurrency, where inputs of Tx's need to refer to outputs in the previous Tx's. We define that Tx's have at most one input and two outputs because PrivHChain aims to prevent side-channel leakages from the Tx sizes. Each input and output consists of four components: address, ID commitment, accumulator of SNs, and various proofs.

Each Tx also contains other metadata: event type, event time, and report time but hiding them, that may not be

considered sensitive, is out of the scope. It is possible to hide the metadata as well with ZKP schemes but we omit them due to the space limit.

**Transaction Input.** Similarly to UTXO model, when a new Tx is created in our system to log a supply chain event, it either (i) has as null input or (ii) the input should contain address, ID commitment, and an accumulator of SNs which refer to the output of the relevant previous Tx. To prove the ownership of the previous Tx's output, the input of the new Tx should contain PoK of the ID hidden inside the ID commitment.

**Transaction Output.** When a Tx has a sender and receiver, it has at most two outputs. One is for the substances sent to the receiver, the other is for the substances left at the sender's end. When there is no leftover, or when the sender/receiver is not in the local region, the Tx has one output only, and these are the only cases where the input quantity is not equal to the output quantity. The difference is the amount of influx/efflux.

**Entry Transaction (Fig. 4a).** When substances enter a local blockchain, participants generate Entry Tx's. These Tx's are responsible for registering new items and do not require Tx inputs. Participants record the relevant SNs and the corresponding hidden accumulator. PoA proof is included in the Entry Tx's. When an Entry Tx occurs in the local chain, the local manager re-generates an Entry Tx on the global chain using the ID of the local manager. This is necessary because it indicates an influx into a local blockchain.

**Exit Transaction (Fig. 4a).** When substances leave a local blockchain, participants generate Exit Tx's. They include the relevant SNs in the input of the Exit Tx's and generate the output for the remaining SNs, accompanied by PoQ. If there are no remaining SNs, the output is null. Both cases involve PoID. Similar to Entry Tx, the local manager checks local Exit Tx and then re-generates a Exit Tx with his/her ID on the global chain. This is necessary because it indicates an efflux from a local blockchain.

**Transfer Transaction (Fig. 4b).** When a sender transfers some substances to a receiver, the sender retrieves the ID commitment and SN accumulator from the previous Tx where the substance came from and uses it to generate the input for the Transfer Tx. Both parties generate new ID commitments with two PoA's, and the sender additionally includes PoQ.

### C. Cross-boundary Transactions

Events in each local region are handled within the corresponding local blockchain. A cross-boundary transfer occurs when an Exit in a local region (called a *source region*) connects to an Entry in a different local region (called a *destination region*). When it occurs, participants report the Tx's to their local managers, and the two local managers jointly generate one Transfer Tx accordingly. Then, the origin Entry Tx in the source region, the last Exit Tx in the destination region, and the Transfer Tx are all committed to the global blockchain by corresponding local managers, and the aforementioned transaction linkages are used to indicate the cross-level dependencies. See Fig. 5 for an example. With this, at the global blockchain, the influx/efflux of local blockchains are logged correctly.



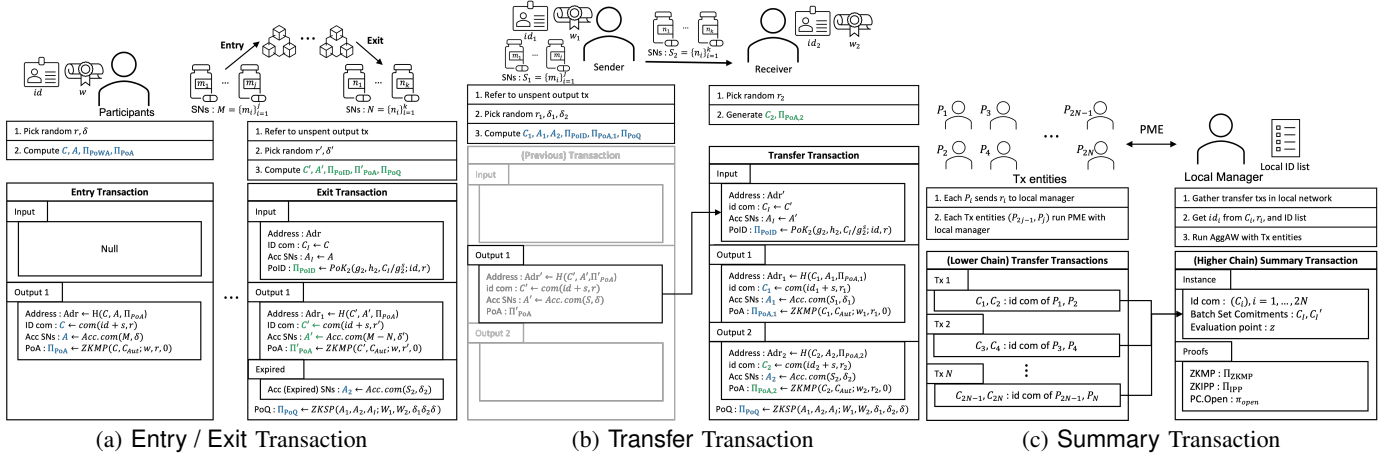


Fig. 4: Description of Transactions

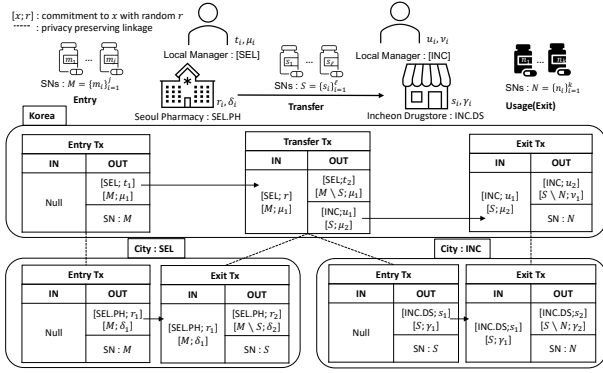


Fig. 5: Cross-boundary Transactions

Synchronization across different levels is needed, and for simplicity, we assume different blockchains are synchronized sufficiently and there are no synchronization issues, e.g., due to existing works [25]–[27]. We focus only on the design of privacy-preserving protocols.

#### D. Aggregating Transactions/Proofs for Global Blockchain

To aggregate Txs and relevant proofs, we need to aggregate authorization witnesses. We explain a few components before presenting the algorithm for the aggregation (AggAW, Fig. 7). **Aggregating Unit Membership Witnesses.** Srinivasan et al. [18] proposed an aggregating technique to update a membership witness  $w_I$  of the set  $I$  with size  $N$ , from unit membership witnesses  $w_i$  for  $x_i \in I$  using the algebraic structure:

$$w_I = \prod_{i \in [N]} w_i^{x_i}, \text{ where } x_i = 1/I'(-x_i) \quad (3)$$

**Private Multi-Exponentiation (Fig. 6).** Unit membership witnesses  $w_i$ 's are needed for aggregation, but local managers do not have access to them. To address this, we introduce a private multi-exponentiation protocol (PME) involving three parties: one is verifier  $\mathcal{V}$ , the others are partial provers  $\mathcal{P}_1$  and  $\mathcal{P}_2$ . Concretely, the purpose of  $\mathcal{V}$  is to get  $w_1^{x_1} \cdot w_2^{x_2}$  without knowledge of  $w_1$  and  $w_2$ , which are private inputs of  $\mathcal{P}_1$  and

- PME Protocol inputs
  - Public :  $g_1, g_2, A, H_1, H_2, G_1, G_2$
  - $\mathcal{V}$  input :  $d_1, d_2, x_1, x_2, \alpha / -\mathcal{P}_i$  input :  $w_i, d_i, H$
- $\mathcal{V}$  sends  $e_i := x_i \cdot \alpha$  to  $\mathcal{P}_i$  for  $i = 1, 2$
- $\mathcal{P}_i$  responds  $P_i = (w_i^{e_i} H^{-1})^{d_i}$
- $\mathcal{V}$  sets  $P = P_1^{d_1} P_2^{d_2}$  and then check  $e(P, g_2^{(s+d_1)}(s+d_2)) = e(g_1^{e_1(s+d_2)+e_2(s+d_1)}, A)$
- If above equation holds,  $\mathcal{V}$  outputs  $P^{\alpha^{-1}}$

Fig. 6: Private Multi-Exponentiation (PME)

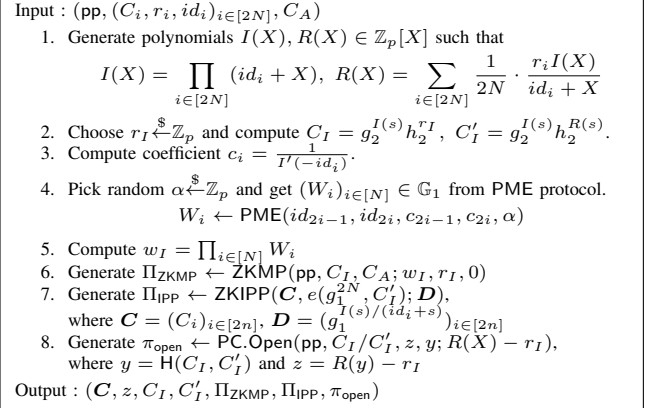


Fig. 7: Aggregation of Authorization Witnesses (AggAW)

$\mathcal{P}_2$  respectively. PME allows for the updating of aggregating membership witnesses through interactions with the owners of the membership witnesses.

**Aggregation of Authorization Witnesses (Fig. 7).** We introduce how a local manager aggregates authorization witnesses via AggAW. Before running AggAW, the local manager collects Transfer Txs in blocks, and then contacts corresponding participants to get their random values  $r_i$ 's used in the construction of  $C_i$ 's. From  $C_i, r_i$  pairs, the local manager gets the IDs of the participants  $id_i$ 's. Then, the local manager and participants run AggAW. AggAW guarantees well-construction of aggregated membership witness  $w_I$ . That is, the set commitment  $C_I$  consists of each  $id_i$  and knowledge proof for membership witness  $w_I$  corresponding to  $I$ . From the PME protocol, one can get  $W_i = w_{2i-1}^{x_{2i-1}} w_{2i}^{x_{2i}}$ . By Equation (3), the

products  $w_I = \prod_{i \in [N]} W_i$  forms a membership witness for set  $I$  and ZKMP guarantees membership relation. Additionally, to guarantee a connection between  $C_I$  and  $\{C_i\}_{i \in [2N]}$ , an additional proof is required. Since  $C_i = g_2^{id_i+s} h_2^{r_i}$  holds for all  $i \in [2N]$ , the following equation holds:

$$e(g_1^{I(s)/id_i+s}, C_i) = e(g_1, g_2^{I(s)} \cdot h_2^{r_i I(s)/id_i+s})$$

From  $id_i$  and  $r_i$ , one can construct  $D_i = g_1^{I(s)/id_i+s}$ . Once  $D$  is constructed,  $D$  should hold  $E(D, C) = e(g_1^{2N}, C'_I)$  and  $\Pi_{PP}$  guarantees the relation. Lastly, to guarantee the equality of the message polynomial between  $C_I$  and  $C'_I$ , we use PC.Open, which guarantees knowledge of the polynomial  $R(X)$ . Thus,  $g_2$  exponents of  $C_I/C'_I$  are zero under the algebraic group model.

In the global blockchain, local managers are responsible for generating Summary (Fig. 4c) TxS periodically to aggregate and summarize the TxS in their local blockchains. By examining the information provided in the Summary Tx, the peers can verify that the participants involved in the Transfer TxS within the local blockchains are permitted participants during the period. Note that Entry TxS and Exit TxS are logged.

To conclude, our TxS satisfy Definition 1 and Definition 2. Formally, we can state it as Theorem 3.

**Theorem 3.** Entry Tx, Exit Tx, Transfer Tx, and Summary Tx satisfy the transaction indistinguishability and verifiability. (Proof available in the anonymized full version [15].)

## V. EVALUATION

Due to the space limit, we present rigorous formal security proofs in the anonymized full version [15] and present performance evaluation only in this paper.

### A. Protocol Overhead

The following evaluation was conducted on an Intel(R) Xeon(R) W-1290P with 128GB of RAM. All protocols were implemented in C++, using the MCL library [28], and the elliptic curve used was BLS12-381. The primary focus was on the computation time of key processes. Our experiments did not consider network latency. Anonymized source code is released for reproducibility [11].

Table I shows the execution time of proof generation and verification, along with the size of the proof. Note that the computation and the proof size do not depend on the accumulator  $A$  or the batch set  $I$ .

**Generation of TX (Fig. 8).** The generation time for all TxS depends on the number of SNs included in the Tx due to accumulating SNs. Most computation time is consumed by the  $Acc_{SN}$  process involving polynomial multiplication, leading to a nearly quadratic increase in TxS' generation time with the number of SNs. When the number of SNs is 1,000, the maximum time to create a Tx is about 6.173 seconds.

TABLE I: Performace of each protocols

Protocol	Prover time	Verifier time	Proof size
ZKMP (Fig. 2)	3.6 ms	5.0 ms	1.9 KB
ZKSP (Fig. 3)	7.2 ms	11.7 ms	3.9 KB

**Verification of Tx.** The verification time of all TxS are independent of the number of SNs. Entry, Exit, and Transfer (Fig. 4) take 5.1 ms, 17.0 ms, and 17.0 ms respectively.

### Aggregation of Authorization Witness and Initial Setup.

The time taken for AggAw process is presented in Fig. 9a. AggAw takes 85.41 seconds when aggregating 1024 TxS, equating to its capacity to aggregate about 378 million lower-layer TxS annually. However, the time consumed by AggAw increases quadratically with the number of TxS. As such, aggregating too many TxS can increase computation time, while aggregating too few TxS can make the purpose of aggregation meaningless. Therefore, determining how many TxS to aggregate at once requires consideration of factors such as the number of peers, computing capabilities, and network conditions. Fig. 9b indicates the time taken in the initial setup process. A public parameter (pp) is created using a trapdoor  $s$  in  $Gen_{pp}$  and  $s$  is securely stored. An ID accumulator is generated using  $s$  in  $Gen_{Acc}$ . Subsequently, for each user corresponding to an ID, an ID commitment and a witness are created in  $Com_{ID}$  and  $Gen_{wit}$  respectively. The setup process for 320000 IDs takes 318.8(sec) in total. The setup, a one-time process, is feasible considering that the total entities in the RoK and the USA are about 46,000 and 300,000 respectively.

### B. Hierarchical Blockchain Simulation

We demonstrate the throughput improvement from the hierarchical blockchain architecture with all the aforementioned ZKP schemes. We use a blockchain sharding simulator *Shard-Eval* [29] to simulate the behaviors. We use the aforementioned Tx verification performance measurements to define the Tx processing time and leave the rest of the settings as default in [29]. Note that the simulation only reflects the trend changing in throughput under different settings. In practice, the actual throughput will be affected by various factors including network latency, number of nodes, etc. Of the three categories of Tx in PrivHChain, the Entry/Exit TxS are considered intra-chain TxS as they always appear in the same lower-level blockchain, and the Transfer TxS are considered cross-chain TxS as they deal with different regions.

Fig. 9c shows the trend of performance changing with different number of local blockchains and cross-boundary Transfer Tx ratios. We first fix the three Tx categories at the same ratio and simulate the throughput under different numbers of local blockchains. The throughput increases as the number of local blockchains goes up ( $\sim 5\times$  of the case without the hierarchical extension), and then plateaus at 11 local blockchains. This is due to the increasing complexity of the consensus protocol. We then choose the setting of 11 local blockchains and increase the Transfer ratio. The simulation shows that the performance starts to drop with more Transfer TxS because they consume blockchain network resources.

## VI. RELATED WORKS

**Blockchain-based Provenance Logging.** Provenance plays critical role in modern systems for accountability [30]. It has been studied extensively and introduced to systems including

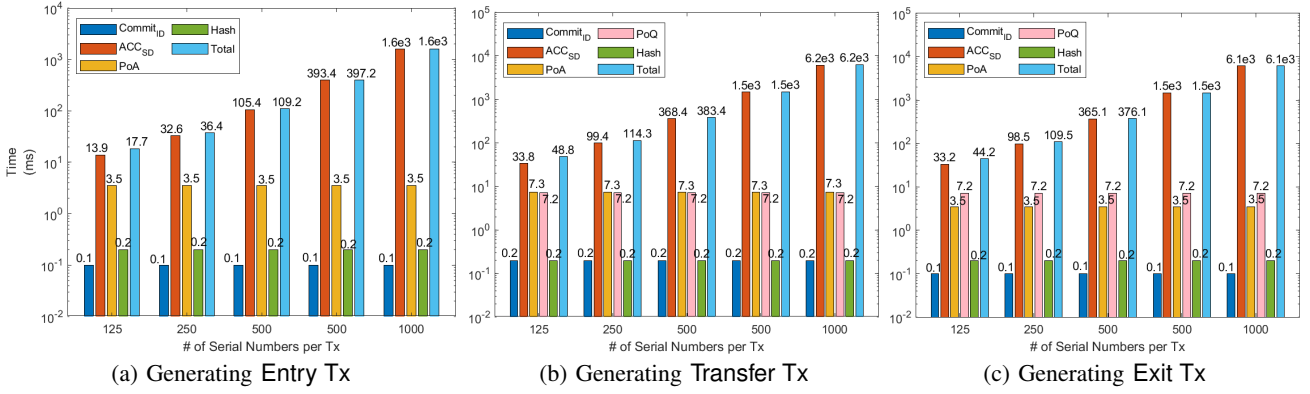


Fig. 8: Performance of Generating Tx in the Local Blockchain

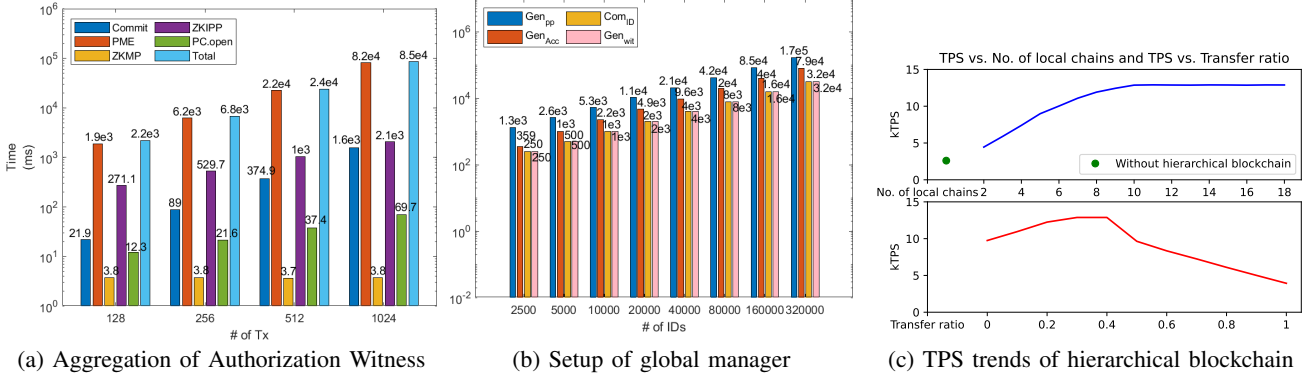


Fig. 9: (a) Generating Tx in the Global Blockchain; (b) Setup of Global Manager; (c) TPS trends of hierarchical blockchain  
 \* Commit<sub>ID</sub>: committing an ID, ACC<sub>SN</sub>: accumulating SNS, PoA: generating PoA, Hash: calculating the  $Tid$  and  $Adr$  of the output.

relational database [31] and collaborative data sharing system [32] and big data platforms [33], [34]. Secure data provenance [35] tackles the security of provenance information, which includes how to protect the provenance information from forgery, modification, and deletion. Several blockchain-based solutions have been proposed for secure provenance [36]. In MedBlock [37], the authors leverage the blockchain to collect the data sharing provenance of health records. In LineageChain [38], provenance records are collected and organized as a Merkle DAG during smart contracts execution. Prv<sup>2</sup>Chain [39] and Trac<sup>2</sup>Chain [40] focus on the linkage privacy of provenance stored on the blockchain. However, there has been no research studying how to apply ZKP schemes to large-scale provenance records with complex dependencies.

**ZK-Rollup.** ZK-rollup is a scaling solution for improving blockchain's scalability, throughput, and cost-efficiency. They leverage ZKP to achieve these goals. In ZK-rollup, transactions are performed off-chain, and a set of transactions is bundled together into a batch or rollup by ZK-rollup operators, which is then submitted to the main blockchain as compressed representation with verifiability. ZK-rollup and our work share the idea of batching the proofs to improve efficiency and throughput. However, the proofs and batching in ZK-rollup cannot handle complex dependencies.

**Blockchain Solutions for Drug Supply Chains.** Blockchain-based drug supply chain systems have been developed to leverage the tamper-proofness provided by blockchain tech-

nology [41]–[43]. Jamil et al. [41] present an efficient supply chain based on Hyperledger. Durgledger by Huang et al. [42] introduces a blockchain system based on UTXO, which supports traceability. Musamih et al. [43] propose a solution based on Ethereum, which incorporates smart contracts to support traceability and transparency. However, none of these approaches protects the privacy of transactions, which must be considered due to the sensitivity of the records. Unfortunately, such a privacy gap cannot be easily filled by existing anonymous blockchain designs due to the nature of the transactions in supply chains.

## VII. CONCLUSION

We present the first privacy-preserving hierarchical blockchain system for monitoring large-scale controlled substance supply chains, where the privacy of individuals is crucial due to the involved sensitive information. To achieve both privacy and verifiability efficiently despite the complex dependencies in the supply chain, we resemble UTXO model of cryptocurrency to model the events as transactions on blockchains, and employ ZKP schemes: our novel schemes (ZK membership and ZK set-split proofs) and existing ones. These ZKP schemes are used to prove authorization and log consistency to blockchain peers without revealing event details. Extensive experiments have shown that the overhead of PrivHChain is negligible/acceptable for real-world deployment.



## REFERENCES

- [1] U. S. Congress, “Controlled substances act,” 2018.
- [2] —, “Drug supply chain security act,” 2013.
- [3] N. A. of the Republic of Korea, “Narcotics control act,” 2020.
- [4] N. C. for Drug Abuse Statistics, Jan 2023. [Online]. Available: <https://drugabusestatistics.org/>
- [5] M. of Food and D. S. Korea, Nov 2021. [Online]. Available: <https://www.data.go.kr/data/15058963/openapi.do>
- [6] K. D. Institute, Jul 2022. [Online]. Available: <https://eiec.kdi.re.kr/policy/materialView.do?num=228577&topic=>
- [7] N. C. D. of Health and H. Services, Jun 2022. [Online]. Available: <https://www.ncdhhs.gov/ncgs-90-11375b-controlled-substances-reporting-system-0>
- [8] S. of California Department of Justice Office of the Attorney General, Dec 2020. [Online]. Available: <https://oag.ca.gov/cures/statistics>
- [9] A. A. Monrat, O. Schelén, and K. Andersson, “Performance evaluation of permissioned blockchain platforms,” in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. IEEE, 2020, pp. 1–8.
- [10] S. Rowden, Feb 2023. [Online]. Available: <https://bitkan.com/learn/how-much-does-ethereum-cost-per-mb-does-eth-have-high-fees-11926>
- [11] Anonymous, “Anonymized github repository,” [Online]. Available: <https://anonymous.4open.science/r/Anonymized-HierarchicalBC-ICBC-2024/README.md>
- [12] N. A. of the Republic of Korea, “Chemical substances control act,” 2021.
- [13] D. Boneh and X. Boyen, “Short signatures without random oracles and the sdh assumption in bilinear groups,” *Journal of cryptology*, vol. 21, no. 2, pp. 149–177, 2008.
- [14] L. Ballard, M. Green, B. de Medeiros, and F. Monrose, “Correlation-resistant storage via keyword-searchable encryption,” *IACR Cryptol. ePrint Arch.*, p. 417, 2005. [Online]. Available: <http://eprint.iacr.org/2005/417>
- [15] Anonymous, “(Full version) PrivHChain: Monitoring the Supply Chain of Controlled Substances with Privacy-Preserving Hierarchical Blockchain,” [Online]. Available: <https://drive.google.com/file/d/1b2UoM6WsHUH0nPCI4AH9xZCgiloY3ms3/view?usp=sharing>
- [16] A. Fiat and A. Shamir, “How to prove yourself: Practical solutions to identification and signature problems,” in *Crypto*, vol. 86. Springer, 1986, pp. 186–194.
- [17] L. Nguyen, “Accumulators from bilinear pairings and applications,” in *Topics in Cryptology—CT-RSA 2005: The Cryptographers’ Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005. Proceedings*. Springer, 2005, pp. 275–292.
- [18] S. Srinivasan, I. Karantaidou, F. Baldimtsi, and C. Papamanthou, “Batching, aggregation, and zero-knowledge proofs in bilinear accumulators,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 2719–2733.
- [19] C.-P. Schnorr, “Efficient identification and signatures for smart cards,” in *Advances in Cryptology—CRYPTO’89 Proceedings 9*. Springer, 1990, pp. 239–252.
- [20] R. W. F. Lai, G. Malavolta, and V. Ronge, “Succinct arguments for bilinear group arithmetic: Practical structure-preserving cryptography,” in *ACM CCS 2019*, 2019, pp. 2057–2074.
- [21] B. Bünz, M. Maller, P. Mishra, N. Tyagi, and P. Vesely, “Proofs for inner pairing products and applications,” in *Advances in Cryptology—ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part III 27*. Springer, 2021, pp. 65–97.
- [22] S. Kim, H. Lee, and J. H. Seo, “Efficient zero-knowledge arguments in discrete logarithm setting: Sublogarithmic proof or sublinear verifier,” in *ASIACRYPT 2022*, ser. LNCS, vol. 13792. Springer, 2022, pp. 403–433.
- [23] A. Kate, G. M. Zaverucha, and I. Goldberg, “Constant-size commitments to polynomials and their applications,” in *Advances in Cryptology—ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings 16*. Springer, 2010, pp. 177–194.
- [24] E. Ghosh, O. Ohrimenko, D. Papadopoulos, R. Tamassia, and N. Triandopoulos, “Zero-knowledge accumulators and set algebra,” in *Advances in Cryptology—ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II 22*. Springer, 2016, pp. 67–100.
- [25] Z. Hong, S. Guo, P. Li, and W. Chen, “Pyramid: A layered sharding blockchain system,” in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.
- [26] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, “OmniLedger: A secure, scale-out, decentralized ledger via sharding,” in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 583–598.
- [27] M. Zamani, M. Movahedi, and M. Raykova, “Rapidchain: Scaling blockchain via full sharding,” in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 931–948.
- [28] S. Mitsunari, “Mcl: a portable and fast pairing-based cryptography library,” 2023.
- [29] V. Priyadarshi, S. Goel, and K. Kapoor, “ShardEval: Sharding-based Blockchain Simulator,” Jul. 2022. [Online]. Available: <https://github.com/vishishtpriyadarshi/ShardEval>
- [30] M. Interlandi, A. Ekmekji, K. Shah, M. A. Gulzar, S. D. Tetali, M. Kim, T. Millstein, and T. Condie, “Adding data provenance support to apache spark,” *The VLDB Journal*, vol. 27, no. 5, pp. 595–615, 2018.
- [31] L. Chiticariu, W.-C. Tan, and G. Vijayvargiya, “Dbnotes: a post-it system for relational databases based on provenance,” in *ACM SIGMOD*, 2005, pp. 942–944.
- [32] Z. G. Ives, T. J. Green, G. Karvounarakis, N. E. Taylor, V. Tannen, P. P. Talukdar, M. Jacob, and F. Pereira, “The orchestra collaborative data sharing system,” *ACM SIGMOD*, vol. 37, no. 3, pp. 26–32, 2008.
- [33] S. Akoush, R. Sohan, and A. Hopper, “HadoopProv: Towards provenance as a first class citizen in MapReduce,” in *TaPP*, 2013.
- [34] J. Wang, D. Crawl, S. Purawat, M. Nguyen, and I. Altintas, “Big data provenance: Challenges, state of the art and opportunities,” in *IEEE BigData*. IEEE, 2015, pp. 2509–2516.
- [35] R. Hasan, R. Sion, and M. Winslett, “Introducing secure provenance: problems and challenges,” in *StorageSS*, 2007, pp. 13–18.
- [36] P. Ruan, G. Chen, T. T. A. Dinh, Q. Lin, B. C. Ooi, and M. Zhang, “Fine-grained, secure and efficient data provenance on blockchain systems,” *PVLDB*, vol. 12, no. 9, pp. 975–988, 2019.
- [37] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, “Medblock: Efficient and secure medical data sharing via blockchain,” *Journal of medical systems*, vol. 42, no. 8, p. 136, 2018.
- [38] P. Ruan, T. T. A. Dinh, Q. Lin, M. Zhang, G. Chen, and B. C. Ooi, “Lineagechain: a fine-grained, secure and efficient data provenance system for blockchains,” *VLDB Journal*, vol. 30, no. 1, pp. 3–24, 2021.
- [39] W. Tang, C. Chenli, and T. Jung, “Prv 2 chain: Storage of tree-structured provenance records in blockchain with linkage privacy,” in *ICBC*. IEEE, 2021, pp. 1–3.
- [40] C. J. Wenyi Tang, Changhao Chenli and T. Jung, “Trac2chain: Trackability and traceability of graph data in blockchain with linkage privacy,” in *ACM/SIGAPP SAC*, 2022, pp. 272–281.
- [41] F. Jamil, L. Hang, K. Kim, and D. Kim, “A novel medical blockchain model for drug supply chain integrity management in a smart hospital,” *Electronics*, vol. 8, no. 5, p. 505, 2019.
- [42] Y. Huang, J. Wu, and C. Long, “Drugledger: A practical blockchain system for drug traceability and regulation,” in *2018 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1137–1144.
- [43] A. Musamih, K. Salah, R. Jayaraman, J. Arshad, M. Debe, Y. Al-Hammadi, and S. Ellahham, “A blockchain-based approach for drug traceability in healthcare supply chain,” *IEEE access*, vol. 9, pp. 9728–9743, 2021.