A distributed ledger management mechanism for storing and selling private data

Sabyasachi Dutta
University of Calgary
Calgary, Canada
Email: saby.math@gmail.com

Arinjita Paul,
Department of Computer Science
IIT Madras, Chennai, India.
Email: arinjita.paul@cse.iitm.ac.in

Rocki H. Ozaki
Real Technology, Inc.
Japan
Email: rockiozaki@gmail.com

C. Pandu Rangan
Department of Computer Science
IIT Madras, Chennai, India.
Email: prangan@cse.iitm.ac.in

Kouichi Sakurai
Faculty of Information Sc. and Electrical Engg.
Kyushu University,Japan
Email: sakurai@inf.kyushu-u.ac.jp

2021 IEEE Conference on Dependable and Secure Computing (DSC) | 978-1-7281-7534-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/DSC49826.2021.9346258

***Abstract*—Storing and processing huge amount of private data is a challenging problem. The problem becomes, in particular, interesting if the user-side storage and computational power is limited. One way to solve the problem of outsourcing private data and maintaining an access control on the storage is proposed by using blockchain technology. However, blockchain technology requires heavy computational power and machinery. In this paper, we propose an approach to store and sell private data with the help of secret sharing. In comparison to blockchain, our methodology is simpler and preserves the privacy of stored data.**

Keywords: Distributed ledger, secret sharing, blockchain, fair exchange, selling private data.

I. Introduction

Distributed ledger is a mechanism that maintains data across a distributed system while ensuring that every party has the same view of the data, even in the presence of corrupted parties. Ledgers can provide an indisputable history of all "events" logged in a system, thereby offering a mechanism for multiple parties to collaborate with minimal trust (any party can ensure the system's integrity by auditing history). We put forward a new perspective on the concept of distributed ledger, based on the concept of a cryptographic primitive "secret sharing". Following the work of Shamir [1] and Blakley [2], we use the concept of "unequal secret sharing", wherein the secret is broken into shares based on a predefined notion of hierarchy among the peers. The "plaintext" ledger is broken into several pieces so that it is not possible to know what is written in the ledger just by looking at some pieces. However, if certain pieces are combined together, then the content of the ledger can be recovered/ read (and then modified etc.). Moreover, this recovery process requires some "privileged" pieces and some "ordinary" pieces of the distributed files/ledger.

One important business model that has attracted a lot of attention is fair exchange of digital goods. More elaborately, suppose Bob has a repository of digital items in his possession. Alice, a potential buyer, wants to buy some of the items from Bob. In most of the cases, both behave honestly and Bob grants Alice an access to the repository. However, in certain cases, either of Alice or Bob may be under the influence of an adversary and behave maliciously giving rise to a dispute. To resolve the dispute between two mutually distrusting parties, (optimistic) fair exchange protocols were proposed. This may require interference of a third party judge to resolve the dispute. Resolution of dispute or fair-exchange of digital goods has been previously addressed by using the technology of blockchain.

A recent line of research addresses the scenario of secure storage of private data using blockchain technology; some works propose permission management or access control mechanism on stored data. However, blockchain technology suffers from an inherent need for heavy computational resources for creating new blocks. Moreover, when data becomes huge, blockchain technology is quite expensive to run. Another aspect of blockchain technology is that the stored data must occur in a clear format so that everyone sees the same copy of data independently. Protecting privacy of data is not a concern for blockchain technology. Recently, some works [3], [4] use secret sharing technique to distribute data among a set of peers as an alternative way to store data in a blockchain. We discuss more elaborately on these concepts in Section II.

Thus the blockchain architecture works under the assumption that the entire ledger of every transactions is stored in the form of a hash chain at every node. As a result, individual nodes incur an ever-increasing storage cost even though many of the transactions may be meaningless for a particular node. Moreover, the mechanism is not sustainable if the incoming data for future storage has very high arrival rate. Therefore, the architecture of blockchain may not be suitable for storing large amount of ever-increasing private data owing to the heavy storage requirements.

In this paper, we propose an alternative way to create a platform to store private data and later sell them to potential buyers, minimizing the possibility of any dispute and if any dispute arises then settle with the help of centralized authorities e.g. banks. Our methodology can be used, with suitable modification to realize inter-bank payment system without the

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY MADRAS. Downloaded on November 21,2021 at 19:51:47 UTC from IEEE Xplore. Restrictions apply.

need of a blockchain system. Our proposal is effective when the user-side storage and computational power is moderate. In fact, the users are not required to burden themselves with heavy computation in order to settle any dispute issues – any issue can be resolved by the banks/exchanges to which the users are enrolled. Any two authorities can run computationally involved (cryptographic) protocols between themselves if any user complains about the malicious intent or misbehaviour of the other party. We use the idea of hierarchical secret sharing to store private data among a set of peers for storing user's private data with the help of the bank. Neither bank nor user is able to trigger any transaction on their own but they require each others consent to initiate the transaction. Hierarchical secret sharing ensures such a condition is met. The following Table I shows a summary of recent results on fair exchange along with our proposal.

II. Related Concepts

A. Secure Storage of Private data

Privacy of data is an important topic of research nowadays. Organizations, both private and public gather huge amount of personal and sensitive data, and further utilize blockchain to protect personal data. Research along this direction include [7], [8], [9], [10], [11], [12], [13], [14], to name a few.

Zyskind et al. [8] has demonstrated the use of blockchain protocols for permission management. They implement a trusted blind escrow service, storing encrypted data while logging pointers on the blockchain. More precisely, the authors address privacy preservation when using third party mobile services. When signing up to a mobile application for the first time, a set of permissions (such as location, list of contacts, camera, etc.) are required to be granted for the proper functioning of the app. Unlike existing applications, the proposed platform allows only users to change the permissions in accordance with the access control policies stored on the blockchain. The proposed decentralized system consists of three entities: mobile phone users, service providers (applications), and the nodes maintaining the blockchain. In the blockchain network, two types of transactions are accepted - T_{access} for access control management and T_{data} for data storage and retrieval. For every user of a service, identity and the associated permissions are generated and sent to the blockchain in a T_{access} transaction. Data collected from the user's mobile phone is encrypted and stored off-blockchain; only hashes of the data are stored in the public ledger. Both the user and service can query the data in a T_{data} transaction.

In another work, Zyskind et al. [9] uses blockchain for access control management and for audit log security purposes, as a tamper proof log of events. Enigma is a proposed decentralized computation platform based on an optimized version of the secure multiparty computation (sMPC). The different parties altogether store and run computations on data while keeping the data completely private.

Azaria et al. [11] explored a blockchain structure applied to electronic medical records (EMR). The authors proposed "MedRec"- a blockchain implementation addressing the four major issues: fragmented, slow access to medical data; system interoperability; patient agency; improved data quality and quantity for medical research. The authors assemble references to disparate medical data and encode these onto a blockchain ledger. The references are arranged to explicitly create an accessible trail for medical history. The system supplements these pointers with on-chain permissioning and data integrity logic, empowering individuals with record authenticity, auditability and data sharing. They have built robust, modular APIs to integrate with existing provider databases for interoperability. A novel data-mining scheme is proposed to sustain the MedRec network and bring open, big data to medical researchers.

B. Fair Exchange of Digital Goods

Smart contracts have been deployed to control access to digital goods in a decentralized manner, e.g., as a platform for digital music. In this setting, Alice wants to buy a digital good from Bob, where (a) Alice wants the guarantee that she can access her purchased good after her payment and (b) Bob only grants access after receiving Alice's payment. While this fairness approach is well-studied [15], [16], [17], corresponding protocols had to rely on a trusted third party. Introducing cryptocurrency deposits, however, allowed for secure, decentralized payment escrows [18], optionally with only conditional conflict resolution [19]. Recently, [5] settled the question of smart contract-based trading of digital goods by providing a general framework.

In a very recent work, Wagner et al. [6] proposed a way for dispute resolution for smart contract based two party protocols. Although similar in motivation with previous works, the approach taken by the authors is more general in nature. They proposed *SmartJudge*, an extensible framework based on Ethereum smart contract to realize two party protocols. *SmartJudge* consists of a general-purpose *mediator contract* and a protocol specific *verifier contract*. The mediator contract consults with verifier contract and identifies the cheating party, if there is a dispute filed by one of the parties. Furthermore, the parties are forced/ encouraged to behave honestly by including a *punishment* technique in the scheme. Both the parties must submit security deposit to the mediator contract. The entire deposit is transferred to the honest party when it is proved that the other one is cheating. [6] also proposed two applications — *trustless* exchange of digital currencies and *selling* digital goods via SmartJudge. For the latter one, the authors used SmartJudge on top of FairSwap platform [5] to achieve the required property.

C. Bulletin Board

Bulletin board (BB) is a publicly accessible ledger which allows anyone to publish arbitrary strings. Once a party publishes data D on BB, it receives a proof/signature to establish that D is published. A bulletin board is *public* if anyone can see all the contents already published in it. The main security issues with the BB are that its content cannot be *erased* and also that a proof of publication cannot be forged. Both

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY MADRAS. Downloaded on November 21,2021 at 19:51:47 UTC from IEEE Xplore. Restrictions apply.

Scheme	goal	BC	crypto.	network	privacy	# parties
FairSwap [5]	fair-exchange	ETH	symm. enc.	broadcast, P2P	-	2
SmartJudge [6]	resolve dispute	ETH	symm. enc.	broadcast,P2P	-	2
Our Technique	sell priv. data	-	secret sharing	P2P	yes	2

TABLE I
COMPARISON AMONG EXCHANGE PROTOCOLS.

centralized and decentralized implementation of BB exist in the literature. Google's certificate transparency project is an example of centralized BB where certificates are issued to the logs. On the other hand, a decentralized implementation of a BB can be achieved from blockchain-based ledgers such as Bitcoin or Etherium. The blockchain-based implementation is either based on *proof of work* (in case of bitcoin) or *proof of stake* (in case of Ethereum).

Certificate Transparency (CT) is a public audit log operated by a coalition of vendors and certificate authorities. CT allows individual certificate authorities to post newly-issued certificates to a public log. A collection of users known as monitors can access the CT log to view the contents of certificates. While the CT log is itself not fundamentally tamper resistant any tampering is detectable due to the structure of the Merkle hash tree and the location of the entry also acts as a proxy for a monotonically increasing sequence number. Under the assumption that such a trusted public BB exists several important results have been proved - achieving *fairness* in multiparty computation in presence of *dishonest majority* [20], realizing individual and universal verifiability in e-voting [21], [22], dispute resolution for two-party protocols [6], publicly verifiable sealed bid auction preserving privacy of the losing parties [23] to name a few. Recently, Tso et al. [24] replaced the traditional implementation of BB by a smart-contract (and is based on blockchain) to design distributed efficient e-voting and e-bidding systems which guarantees better anonymity, privacy of data transmission, data reliability and verifiability. On another development, Heiberg et al. [25] pointed out some problems and limitations of using public permissionless blockchain based acting as a bulletin board in an e-voting scheme with large number of participants. In fact for eligibility verification of voters, ballot secrecy, consistency verification etc. one has to either rely on a trusted third party or it incurs a huge cost to replace third party by using non-trivial cryptographic primitives.

III. OUR MECHANISM

In this section, we propose a distributed ledger mechanism, as an alternative to the Satoshi Nakamoto model [26]. Our mechanism which is motivated by a white paper of Ozaki-Yokoyama [27], distributes shares of the ledger to each member, without the need to compile every transaction of the individual members into a single file. This eliminates the need for "mining" in order to verify the correctness and authenticity of the ledger, and also the need for blockchaining to maintain integrity of the past history. We next give an overview of our distributed ledger mechanism.

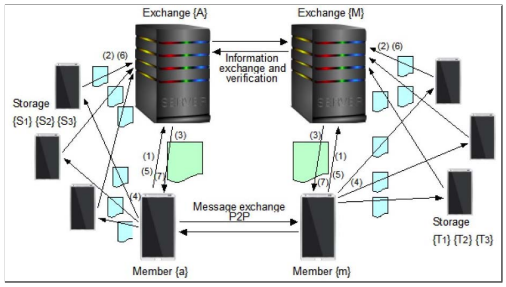

Fig. 1. A schematic of the protocol.

A. An overview of the mechanism

In traditional (k, n)-threshold secret sharing all the participants hold the same power. More precisely, any k shares out of n shares can reconstruct the secret. However, there are other secret sharing schemes where some shares are more important than the rest [28], [29], [30]. Hierarchical secret sharing (HSS) is such an example where without some "privileged" shares, it is impossible to reconstruct the secret [31]. In a hierarchical secret sharing scheme the shareholders are assigned different levels of power such that certain number of shareholders from every level must cooperate in order to retrieve the secret. Several hierarchical secret sharing schemes have been proposed in information-theoretic setting as well as in computational setting to gain efficiency. We will not discuss in detail about hierarchical secret sharing, rather we will assume the existence of such schemes and denote it by $(\mathsf{Share}^{\mathsf{HSS}}, \mathsf{Rec}^{\mathsf{HSS}})$. For some recent development in this field we refer to the following works [32], [33], [30].

Note 1: We note that there always exists a hierarchical secret sharing scheme $(\mathsf{Share}^{\mathsf{HSS}}, \mathsf{Rec}^{\mathsf{HSS}})$ for any predefined choice of number of "privileged" shares and number of "ordinary" shares. The scheme of Shima-Doi [30] gives an efficient construction of hierarchical secret sharing scheme. The scheme is explained in Algorithm 2.

Now we present the design of distributed ledger such that its users cannot manipulate or update. The design inherently calls a $(ShareGen^{\mathsf{HSS}}$ algorithm from the "user/member" side and $Recover^{\mathsf{HSS}})$ algorithm from the "Exchange" side.
The mechanism is based on mutual containment, consisting of users (**member**) with a unique ID and a cryptographic currency exchange (**exchange E**) who hosts the members in the system. The secret sharing is done by two processes: **splitting** and **reconstructing** as shown in Figure 1 and explained below:

- Every member runs the splitting process.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY MADRAS. Downloaded on November 21,2021 at 19:51:47 UTC from IEEE Xplore. Restrictions apply.

- The exchange E runs the reconstructing engine.
- Every split file is stored in storage spaces $S_n = (S_1, S_2, \ldots)$, provided other members in the system, who wishes to provide the spaces in return of reward or no-reward, based on the business model.
- If a member (M) wants to update his ledger, it requests E to initiate the process.
- E collects the required pieces from S_ns and combines them using **reconstruct** engine to reconstruct the plain-text/ message and sends back to M.
- M writes/updates the ledger and then according to a hierarchy among E and servers (hierarchy can be decided by M and must be sent to E) he divides the ledger using **split** engine i.e. $ShareGen^{\mathsf{HSS}}$ and shares the updated file/ledger with S_n.
- Once M side process is over, E combines the files using **reconstruct** from S_n and reconstructs the file.
- E checks whether the information updated by M is correct or not; and accordingly he "passes" the transaction/ update.
- Remittance from one member of one certain exchange to a member of another exchange can be done using interaction between the two exchanges in which first exchange "proves" to the second exchange that his member has enough balance for remittance.

Goals of the distributed ledger mechanism. The main goal achieved by the above technique is enlisted below:

1) **Anti-manipulation**: The mutual containment property of the mechanism achieves anti-manipulation of the ledgers.
2) **Manipulation detection**: Verification by the exchange E ensures detection of any manipulation done to the ledger.
3) **Fault tolerance**: Since the secret file is split and copied to different storage spaces as back-up files, this achieves fault tolerance, as the secret can be reconstructed using a threshold (not all) numbers of secret files.

The key entities playing significant roles are:

1) **Exchange:** Possibly a *Crypto-currency* exchange which can host many members/users.
2) **Members:** Users who are affiliated to at least one Exchange. Members have different IDs. One member can be affiliated to many Exchanges but with different IDs.
3) **Storage Providers:** Providers of storage of data. May or may not be a member, possibly untrusted entities.

• Core Mechanism

The core of the process is a hierarchical secret sharing scheme (HSS) such that the "Sharing process" (seen as a software module) is possessed by Members only. However, the " Reconstruction process" (also implemented as a software module) is possessed by Exchange only. An efficient hierarchical secret sharing scheme using *information dispersal algorithm* was proposed by Shima and Doi [30]. In Algorithm 2 we present the scheme and we may use the scheme of Roy et al. [34] on top of [30] to ensure reconstructing the correct secret value even if some of the shares are corrupted.

Algorithm 1 Protocol for updating ledger

procedure MUTUAL CONTAINMENT BETWEEN MEMBER & EXCHANGE
Member m makes request to Exchange A to reconstruct its ledger.
A fetches requisite shares from storage providers and reconstructs the ledger. Then the ledger file is sent to m.
procedure UPDATING THE LEDGER FILE
Upon receiving file from A, m updates the file.
m generates shares of this updated file by running $\mathsf{Share}^{\mathsf{HSS}}$ on updated file.
m distributes the shares among the Exchange and storage providers such that A receives "privileged" shares.
procedure VERIFICATION STEP
Upon receiving shares of the updated file from m, A reconstructs the file by using its own share and requisite shares from some storage providers.
A verifies the content of the updated file.
if, it is verified to be correct, then record "no error"
if, verification fails, then check for error or notify m to redo the splitting

Algorithm 2 Hierarchical Secret Sharing Scheme from [30]

procedure SHARE GENERATION
Input: secret $s \in \{0,1\}^{\lambda}$.
for $i \leftarrow 1$ to $k-1$:
$\vec{r}_i \leftarrow r_i \leftarrow \{0,1\}^{\lambda}$ // generating the randomness
$\vec{r}_0 \leftarrow s' \leftarrow s \oplus \{r_1 \oplus \ldots \oplus r_{k-1}\}$
$\vec{M}_r = (\vec{r}'_0 \cdots \vec{r}'_{k-1}) \leftarrow (\vec{r}_0 \cdots \vec{r}_{k-1})^T$
for $i \leftarrow 0$ to $k-1$:
$\vec{R}_i \leftarrow Share^{IDA}(\vec{r}'_i, G)$ // G is generator matrix capturing the hierarchy & IDA denotes the information dispersal algorithm.
for $i \leftarrow 0$ to $n-1$:
$w_i \leftarrow ||_{j=0}^{k-1} \vec{R}_j[i]$ // concatenating i^{th} element of the vectors $\vec{R}_j$
Output: n shares $w_0, w_1, \ldots, w_{n-1}$.
procedure RECOVERY ALGORITHM
Input: k many shares $(w_{t_0}, \ldots, w_{t_{k-1}})$
for $i \leftarrow 0$ to $k-1$:
$||_{j=0}^{k-1} \vec{R}_j[t_i] \leftarrow w_{t_i}$
for $i \leftarrow 0$ to $k-1$:
$\vec{r}'_i \leftarrow Recover^{IDA}(\vec{R}'_i, G')$
$(\vec{r}_0 \cdots \vec{r}_{k-1})^T \leftarrow \vec{M}_r = (\vec{r}'_0 \cdots \vec{r}'_{k-1})$
for $i \leftarrow 0$ to $k-1$:
$s' \leftarrow r_0$
$s \leftarrow s' \oplus \{\oplus_{j=1}^{k-1} r_j\}$
return s

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY MADRAS. Downloaded on November 21,2021 at 19:51:47 UTC from IEEE Xplore. Restrictions apply.

IV. Storing and Fair Exchange of Private Data

In this section, we demonstrate some applications of our mechanism in details. Let us consider the scenario where huge amount of private data are generated on a daily basis e.g., medical data of people (blood pressure, sugar level, amount of physical exercise, pulse rate, distance traveled everyday etc.) and also data on financial transactions taking place everyday. The data are sensitive and should not be made public unless and until the owner agrees to do so. Of course, storing and managing this kind of huge data in a blockchain is costly and time consuming. Suppose an Organization/Exchange E_1 develops a system which maintains a ledger to store private data for each member who has registered with the *exchange*. Parties who play different roles in the system are as follows:

1) *Exchange*: The exchange has only READ option from the ledger. It validates the update on ledger.
2) *Member*: Members or users have only WRITE option on the ledger. However, members need *permission* of *Exchange* to write/update the ledger. Members may or may not have permission to read.
3) *Storage Providers*: Possibly untrusted entities which offer storing space to make the ledger management scheme a distributed one.

A scheme realizing such a system and also maintaining privacy of data is described in Algorithm 1. We note that not using existing blockchain technology makes the system more efficient.

A. How to sell private data through the Broker

We present a distributed ledger management technology to store and sell private data where the users can have control over her private data. Same scheme can be achieved by integrating *SmartJudge* [6] and *FairSwap* [5]. However, we want to give an alternative solution having the following features — minimizing/ removing the usage of Blockchain technology, relaxing the necessity of making a security deposit prior to the protocol. In the following we give an overview of our procedure.

- The Premise:
  1) User (Alice) must be registered with an Exchange/ Organization with an account id.
  2) Exchange sets up an Exchange server to serve the users with locating IP address of the users, locate the IP addresses of the storage (in case a distributed storage set up is needed to maintain privacy of the data), read and verify the content of the ledger, provide verification details to recipient/ buyer side (Bob) Exchange/ Bank, collaborate with recipient side Exchange and finalize the transaction.
  3) A certification authority (optional) to manage a public master ledger file which contains all transactions of all users.
- Procedure:
  1) Storing data to the Exchange: Alice stores data in the Exchange A (encrypted) and updates with the permission of Exchange. Exchange is able to verify. For implementation one can use the proposal in Algorithm 1.
  2) If Bob wants to buy data from Alice, he sends Alice side Exchange A all his details (including the name of exchange M he is affiliated with) and informs his side Exchange/ Bank about the request. A forwards the message to Alice.
  3) Upon receiving the request from Bob, Alice sends her Exchange a notice with the following information — Alice wants to start a selling process, content of the data to be sold.
  4) Alice side Exchange A then does two things. First, it verifies the content of Alice's data. Second, it searches for Bob's location by contacting the Bob-side Exchange M.
  5) If M responds to the inquiry of A, then A does the following: (a) sends notice to M that Alice wants to sell her data, (b) IP address and ID of Alice, (c) public key of Alice and (d) remittance content of Alice is at par with Bob's request.
  6) Upon receiving the information from A, M does the following (after encrypting them with Bob's public key): (a) Alice wants to sell her personal data to Bob, (b) ID and IP address of Alice, (c) public key of Alice and (d) remittance content of Alice is at par with Bob's request.
  7) If all information checks out, Alice and Bob both signal their respective exchanges to start the transaction.
  8) A sends the data of Alice to M after encrypting the data with Bob's public key and similarly, M sends the required amount to A.
  9) Alice and Bob receive their money and data respectively from corresponding Exchanges.

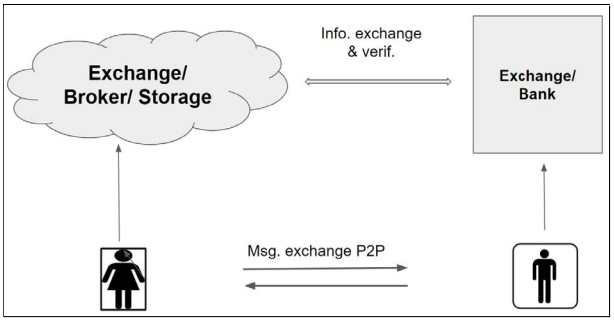

Fig. 2. High level overview of data exchange procedure. Fig. 1 contains a more detailed description of the procedure.

B. An alternative approach

We see our approach is divided into two subprotocols – the first is to store and update the ledger and the second is secure exchange of data and its price. We modify the first subprotocol, the fair exchange protocol remains the same (as above).

Blockchain (based on bitcoins) use hash chain structure and store individual transaction data as a Merkle tree. We take a

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY MADRAS. Downloaded on November 21,2021 at 19:51:47 UTC from IEEE Xplore. Restrictions apply.

simpler form used in [3] which improves the recovery cost and eases consistency verification. Following their way, we take two hash functions g, h. If U_t denote the update at time t for a party, we denote by $W_t = (H_{t-1}||g(U_t)$, the concatenation of previous hash and a hash of the current data. The hash value which is stored with the $(t+1)^{th}$ update is $H_t = h(W_t)$.

Let at time t, client Alice wants to make an update U_t to her private data. It is an addition of more data to the existing data of her. Alice requests exchange A to make an update, to which A replies back with (a) IP addresses of available storage providers/peers and (b) a random partition of the peers such that size of each set in partition is bounded by a predefined value. These sets are called *data zones* and the update from Alice is securely distributed in each zone along with the Exchange A. In Algorithm 3 we describe the process which is based on the work of [3]. We assume a private key encryption scheme $\mathbb{E}$ and an ideal hierarchical secret sharing scheme.

Algorithm 3 Protocol for uploading data

1: **procedure** MUTUAL CONTAINMENT BETWEEN ALICE & EXCHANGE
2: Alice makes request to Exchange A to update her data.
3: A sends locations of peers & a random Partition $P_t = \{zone(t)^1, \ldots, zone(t)^r\}$ of peers.
4: **procedure** UPLOADING NEW DATA
5: for each $zone(t)^i$ $i = 1, \ldots, r$, Alice generates a private key K_t^i.
6: Encrypt U_t with K_t^i to get C_t^i for every i.
7: Distribute C_t^i and store it at all peers in $zone(t)^i$
8: Alice generates shares of K_t^i by running $ShareGen^{\mathsf{HSS}}$ for every i where the hierarchical access structure contains the peers as the ordinary nodes whereas, without the shares of Exchange (and/or Alice) the secret K_t^i cannot be reconstructed.
9: Alice distributes the shares among the Exchange and peers such that E receives "privileged" shares for every *data* zone.

V. DISCUSSIONS

A. Advantage over Blockchain

The methodology presented in this paper deals with only personal data which grows at a great rate so that it becomes impossible for a user to store them. The Exchange/ Organization provides support to store them and also manage business dealings. The user side computation power and storage capacity is not sufficient. The same technology can be applied to music or video files. But in those cases, files may not grow at a great rate. We only consider one particular scenario – providing storage and making a platform for business is the main aim of this work. There is a possibility of redistribution of files by a dishonest buyer. However, that issue is out of our scope. The exchanges are assumed to be reputed so that misbehaviour is less expected. They have a reputation to maintain. Also, the users are relieved from the burden of communicating with each other and manage the business deals. The main achievement of this work is to avoid the complexities of blockchains. There are few works achieving similar functionality and in fact, more general ones using the platform of blockchain. However, the systems become typically slow as a consequence. Moreover, one main drawback of public permissionless blockchains is that the success of accepting a transaction into the ledger depends on financial incentives rather than a legislative need to create an immutable audit trail. Another issue is that in case of blockchain it is assumed that the majority of hashing power is under control of honest miners – but in reality, most miners are rational and will give preference to transactions with highest fees. To speed up the process increasing the transaction rewards for the miners can be proposed but that increases the overall cost of the entire procedure. So whether or not cryptocurrencies can be used depends on the scenario. Moreover, blockchain is suited only for small data. When the data becomes terabytes, keeping hundreds, if not thousands of identical copies and propagating new data to all copies is a slow and tedious process. The discussion on cryptocurrency using blockchain has exposed their "slowness" and "latency" nature of blockchain. The good point is that all data can be made public and yet it is tamper resistant. Privacy is non-existent in public blockchain and in fact is not the goal of blockchain. Further, power consumption for blockchain, especially when the data grow big, is an ecological issue. According to [35], the power consumption of BitCoin can be compared to the entire power used in Ireland.

B. Security Analysis

We assume that the Exchanges are trusted and reputed entities. However, the nodes/peers/storage providers can be subject to randomized denial of service attack and adaptive, targeted node capture attack. First attack may prevent data recovery whereas second type of attack may result in hampering the privacy of the stored data and possible alteration of stored data.

In the first scenario, the adversary $\mathcal{A}_1$ is unaware of system parameters. It randomly selects a subset of peers at random so that the peers do not return the data stored in them. However, this type of attack is thwarted by the system if we assume majority of the storage providers are not captured by the adversary. This type of assumption is recurrent in the literature. Setting the threshold value to be $\lceil \frac{N}{2} \rceil$ where N denotes the number of available storage providers among which the data is secret shared by the user. It is not very difficult to see that the above adjustment in the threshold parameter is not enough to prevent second type of adversary $\mathcal{A}_2$ from altering the original data. However, whatever be the value of threshold parameter, $\mathcal{A}_2$ is unable to hamper the privacy of the stored data. This is due to the reason that even if all the peers are captured by $\mathcal{A}_2$ and it learns the contents stored in the peers, it is never able to reconstruct the private data as the data can never be restored with the help of ordinary shares only. The peers contain only ordinary shares and the privileged shares are stored in the Exchange. Coming back to possible "alteration

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY MADRAS. Downloaded on November 21,2021 at 19:51:47 UTC from IEEE Xplore. Restrictions apply.

of content" attack by $\mathcal{A}_2$, we see that if the adversary changes the content of the peers then during the reconstruction process some false value can be recovered. In this scenario, there must be a checking procedure to validate the correctness of the share-values submitted by the peers. In this context, applying a robust hierarchical secret sharing scheme which guarantees correct recovery of the secret value when there is a majority of servers present (e.g. the scheme presented in [34], [36], instead of [30]), the attack can be prevented.

VI. Conclusion

In this paper, we have put forward a distributed ledger mechanism to create a platform for securely storing and selling private data. We use hierarchical secret sharing as a tool to achieve the functionality. Our methodology reduces computational cost and is energy efficient when compared to the blockchain technology. We note that the approach can be applicable also to cryptocurrencies including government controlled Digital-Cash, which was the original motivation of the white paper [27].

Acknowledgement

S. Dutta is grateful to NICT, Japan for a financial support for 2018-19 when a preliminary draft of this paper was made along with A. Paul who at the time was visiting Kyushu University, Japan. Kouichi Sakurai acknowledges a JSPS KAKENHI Grant (Number JP18H03240). The authors are grateful to the anonymous reviewers for their comments.

References

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

[2] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the national computer conference*, vol. 48, no. 313, 1979.

[3] R. K. Raman and L. R. Varshney, "Distributed storage meets secret sharing on the blockchain," in *2018 Information Theory and Applications Workshop, ITA 2018, San Diego, CA, USA, February 11-16, 2018*, 2018, pp. 1–6.

[4] Y. Kim, R. K. Raman, Y. Kim, L. R. Varshney, and N. R. Shanbhag, "Efficient local secret sharing for distributed blockchain systems," *IEEE Communications Letters*, vol. 23, no. 2, pp. 282–285, 2019. [Online]. Available: https://doi.org/10.1109/LCOMM.2018.2886016

[5] S. Dziembowski, L. Eckey, and S. Faust, "Fairswap: How to fairly exchange digital goods," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, 2018, pp. 967–984.

[6] E. Wagner, A. Völker, F. Fuhrmann, R. Matzutt, and K. Wehrle, "Dispute resolution for smart contract-based two-party protocols," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2019, pp. 422–430.

[7] G. Kellaris, G. Kollios, K. Nissim, and A. O'Neill, "Accessing data while preserving privacy," *CoRR*, vol. abs/1706.01552, 2017. [Online]. Available: http://arxiv.org/abs/1706.01552

[8] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Symposium on Security and Privacy Workshops, SPW 2015, San Jose, CA, USA, May 21-22, 2015*, 2015, pp. 180–184.

[9] ——, "Enigma: Decentralized computation platform with guaranteed privacy," *CoRR*, vol. abs/1506.03471, 2015. [Online]. Available: http://arxiv.org/abs/1506.03471

[10] C. Noyes, "Blockchain multiparty computation markets at scale," 2016.

[11] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2nd International Conference on Open and Big Data, OBD 2016, Vienna, Austria, August 22-24, 2016*, 2016, pp. 25–30.

[12] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Distributed Applications and Interoperable Systems - 17th IFIP WG 6.1 International Conference, DAIS 2017, Held as Part of the 12th International Federated Conference on Distributed Computing Techniques, DisCoTec 2017, Neuchâtel, Switzerland, June 19-22, 2017, Proceedings*, 2017, pp. 206–220.

[13] S. Gurses and J. M. del Álamo, "Privacy engineering: Shaping an emerging field of research and practice," *IEEE Security & Privacy*, vol. 14, no. 2, pp. 40–46, 2016. [Online]. Available: https://doi.org/10.1109/MSP.2016.37

[14] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017. [Online]. Available: https://doi.org/10.3390/info8020044

[15] F. Bao, R. H. Deng, and W. Mao, "Efficient and practical fair exchange protocols with off-line TTP," in *Security and Privacy - 1998 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 3-6, 1998, Proceedings*, 1998, pp. 77–85.

[16] H. Bürk and A. Pfitzmann, "Value exchange systems enabling security and unobservability," *Computers & Security*, vol. 9, no. 8, pp. 715–721, 1990. [Online]. Available: https://doi.org/10.1016/0167-4048(90)90114-9

[17] J. Zhou and D. Gollmann, "A fair non-repudiation protocol," in *1996 IEEE Symposium on Security and Privacy, May 6-8, 1996, Oakland, CA, USA*, 1996, pp. 55–61.

[18] I. Bentov and R. Kumaresan, "How to use bitcoin to design fair protocols," in *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, 2014, pp. 421–439.

[19] D. Jayasinghe, K. Markantonakis, and K. Mayes, "Optimistic fair-exchange with anonymity for bitcoin users," in *11th IEEE International Conference on e-Business Engineering, ICEBE 2014, Guangzhou, China, November 5-7, 2014*, 2014, pp. 44–51.

[20] A. R. Choudhuri, M. Green, A. Jain, G. Kaptchuk, and I. Miers, "Fairness in an unfair world: Fair multiparty computation from public bulletin boards," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, 2017, pp. 719–728.

[21] M. Chaieb, S. Yousfi, P. Lafourcade, and R. Robbana, "Verify-your-vote: A verifiable blockchain-based online voting protocol," in *Information Systems - 15th European, Mediterranean, and Middle Eastern Conference, EMCIS 2018, Limassol, Cyprus, October 4-5, 2018, Proceedings*, 2018, pp. 16–30.

[22] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES 2005, Alexandria, VA, USA, November 7, 2005*, 2005, pp. 61–70.

[23] S. Bag, F. Hao, S. F. Shahandashti, and I. G. Ray, "Seal: Sealed-bid auction without auctioneers," *IEEE Transactions on Information Forensics and Security*, 2019.

[24] R. Tso, Z.-Y. Liu, and J.-H. Hsiao, "Distributed e-voting and e-bidding systems based on smart contract," *Electronics*, vol. 8, no. 4, p. 422, Apr 2019. [Online]. Available: http://dx.doi.org/10.3390/electronics8040422

[25] S. Heiberg, I. Kubjas, J. Siim, and J. Willemson, "On trade-offs of applying block chains for electronic voting bulletin boards," *E-Vote-ID 2018*, p. 259, 2018.

[26] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[27] R. H. Ozaki and K. Yokoyama, "Secret sharing based distributed ledger management mechanism," *Real Technology Inc.*, Ver. 1.4.2, May 26, 2018 (available on request from Rocki H. Ozaki).

[28] K. Sakurai and R. H. Ozaki, "Hierarchization of computational secret sharing schemes and analysis of their access structures," *IEICE 2016 General conference A-7-17*, March, 2016.

[29] ——, "Yet another security issue around secret sharing schemes – revisiting "unfair" cryptosystems," *2016 Symposium on Cryptography and Information Security*, Kumamoto, Japan, January 19–22, 2016.

[30] K. Shima and H. Doi, "A hierarchical secret sharing scheme based on information dispersal techniques," in *Information Security and Cryptology - ICISC 2018 - 21st International Conference, Seoul, South Korea, November 28-30, 2018, Revised Selected Papers*, 2018, pp. 217–232.

[31] T. Tassa, "Hierarchical threshold secret sharing," *J. Cryptology*, vol. 20, no. 2, pp. 237–264, 2007. [Online]. Available: https://doi.org/10.1007/s00145-006-0334-8

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY MADRAS. Downloaded on November 21,2021 at 19:51:47 UTC from IEEE Xplore. Restrictions apply.

[32] K. Shima and H. Doi, "({1, 3}, n) hierarchical secret sharing scheme based on xor operations for a small number of indispensable participants," in *2016 11th Asia Joint Conference on Information Security (AsiaJCIS)*. IEEE, 2016, pp. 108–114.
[33] ——, "Xor-based hierarchical secret sharing scheme," in *International Workshop on Security*. Springer, 2018, pp. 206–223.
[34] P. S. Roy, S. Dutta, K. Morozov, A. Adhikari, K. Fukushima, S. Kiyomoto, and K. Sakurai, "Hierarchical secret sharing schemes secure against rushing adversary: Cheater identification and robustness," in *Information Security Practice and Experience - 14th International Conference, ISPEC 2018, Tokyo, Japan, September 25-27, 2018, Proceedings*, 2018, pp. 578–594.
[35] "https://www.economist.com/the-economist-explains/2018/07/09/why-bitcoin-uses-so-much-energy."
[36] G. Traverso, D. Demirel, and J. Buchmann, "Dynamic and verifiable hierarchical secret sharing," in *International Conference on Information Theoretic Security*. Springer, 2016, pp. 24–43.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY MADRAS. Downloaded on November 21,2021 at 19:51:47 UTC from IEEE Xplore. Restrictions apply.