IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 7, 2021, accepted December 1, 2021, date of publication December 3, 2021, date of current version January 5, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3132690

Privacy-Preserving Framework for Blockchain-Based Stock Exchange Platform

HAMED AL-SHAIBANI, NOUREDDINE LASLA, (Member, IEEE), MOHAMED ABDALLAH, (Senior Member, IEEE), AND SPIRIDON BAKIRAS, (Senior Member, IEEE)
Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University (HBKU), Doha, Qatar
Corresponding author: Hamed Al-Shaibani (halshaibani@mail.hbku.edu.qa)

This work was supported by Qatar National Library.

ABSTRACT This paper presents a privacy persevering framework for a decentralized stock exchange platform, ensuring anonymity and unlinkability of the investors' accounts and their respective trading activities. The proposed framework meets these privacy requirements by (i) anonymizing both the unique account identifier (NIN) and balance information through customized data generalization and distortion techniques and (ii) making trading transactions unlinkable to their original investors by ensuring that both the NIN and balance are k-anonymous; i.e., k accounts belonging to different investors have the same balance. Moreover, to ensure long-term unlinkability, the process of anonymization is repeated at regular time intervals (every trading session). In addition to anonymity and unlinkability characteristics, the proposed framework is augmented with traceability and non-repudiation features. The simulation experiments with several market sizes and types confirm the effectiveness of the proposed framework in achieving full k-anonymity. Furthermore, to assess the overhead of the proposed privacy algorithms on the trading execution time, we conduct several experiments considering different anonymity levels k. We compare the transaction execution time of our proposed platform against a traditional non-privacy-preserving blockchain-based stock exchange. The obtained results for the worst-case scenarios show an acceptable execution time overhead.

INDEX TERMS Blockchain, privacy, smart contract, stock exchange, trading, anonymity.

I. INTRODUCTION

The growth and development of the stock market have a substantial impact on any given country's economic growth [1]. In 2016, the global market capitalization of the stock exchange was worth an estimated $70 trillion, which reflects the large size of the financial transactions and investments performed to buy shares and other securities offered by the stock market [2]. One of the primary factors affecting the stability of the stock market is the extent of guaranteed fairness that to investors to join the market and the existence of a well-established financial regulator to set market rules and ensure their implementation. As per the regulator rules, there should be no disclosure of sensitive information during trading that can negatively disturb the market and lead to price manipulation [3]. For instance, the reveal of information related to investor identity can cause a well-known stock market manipulation attack called front-running [4]. Some entities can benefit from prior access to premium market information about upcoming transactions and trades in such an attack. Similarly, by knowing the actual identities behind large buy or sell orders, other investors can plan accordingly to trade before or after such orders to benefit from the potential price movement of the traded shares. Therefore, most stock market regulators consider the investors' identity as confidential and sensitive information. According to [5] and [6], trading anonymously ensures that the investor's identity is not traceable and provides a regulated framework for fair trading.

Recently, blockchain technology has been introduced in the financial industry as an effective means towards a more secure, transparent, and decentralized financial system without relying on intermediaries [7]. For stock exchange, some recent efforts in [8] and [9] have been devoted to designing new decentralized stock exchange platforms that overcome traditional stock market limitations such as the presence of a single point of failure, the long time of financial settlements, and the limited transparency offered to the investors.

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Tsun Cheng.

Nevertheless, these aforementioned efforts have not considered the preservation of the privacy of investors' identities and that of their balances. Although some work already exists to address privacy concerns in blockchain-based solutions using data distortion and encryption techniques, none of them is designed to meet the privacy requirements of the stock exchange, including anonymity, unlinkability and traceability, and non-repudiation.

In blockchain, data is replicated across different participants, and for platforms like the stock exchange, sensitive information such as investors' identities should only be accessible by authorized participants. The investor's identity in the stock exchange platform consists of a unique identifier (NIN) and a balance of cash or share associated with it. The NIN is being used to enter orders into the platform to match other orders and generate trades. The investor's identity should be kept anonymous and unauthorized participants should not be able to identify the original owner of the entered order and the generated trade. Given the issue of privacy discussed above, we propose in this paper a privacy-preserving blockchain-based stock exchange framework using data distortion and encryption techniques that ensure privacy. The proposed framework creates a set of anonymous accounts for each investor that they can use during a single trading session. The original balance of an investor's account is also split into different amounts and assigned to the set of created anonymous accounts. To ensure unlinkability and prevent an attacker from tracking and linking the different anonymous accounts to their original investor's account, a data distortion technique based on the k-anonymity model is employed. Each anonymous account is assigned a balance with an amount that exists in at least $k-1$ other accounts in the market. Moreover, to make it even harder for an attacker to build trading patterns from the anonymous accounts and infer helpful information during the trading, the distortion technique is reapplied at the beginning of each new trading session. The proposed framework also allows authorized entities to perform auditing on transactions by tracing the anonymous accounts to their original owners.

The k-anonymization process is made off-chain by the Central Securities Depository (CSD) to make it possible to track back the anonymous accounts by authorized authorities for accounting and traceability purposes.

The contribution of the paper can be summarized as follows:

- Analyze and define the privacy requirement of a decentralized blockchain-based stock exchange platform.
- Develop a novel distortion technique that anonymizes both the investor's unique identifier (NIN) and balance.
- Ensure long-term unlinkability by performing repeated anonymization before the start of each trading session.
- Provide tracing functionality to authorized entities to link the investor's anonymous accounts to their original accounts.
- Assess the additional overhead by comparing the execution time with and without the decentralized stock exchange platform's privacy preserving framework.

The remaining of this paper is organized as follows: Section II discusses blockchain privacy methods and related work. Section III provides an overview of the consortium blockchain-based stock exchange platform, privacy limitations, and how to address them. Section IV presents our privacy-preserving framework that addresses the decentralized stock exchange platform's privacy requirements. In section V, we provide the performance evaluation of our solution and study the overhead added by the privacy-preserving algorithms. Section VI concludes the paper.

II. RELATED WORK

In blockchain, privacy is not provided by design, given that data are replicated across different participants. Several solutions have been proposed to preserve privacy in blockchain and can be classified into two main categories; (i) data distortion and (ii) data encryption-based approaches [10]. In the following, we provide an overview and discuss the main techniques for each category.

A. DATA DISTORTION

Data distortion is a technique that provides anonymity by making it challenging to link some sensitive information, such as geographical locations and user identities, to their actual data [10], [11]. In the trading context, this approach seeks to hide the identity of a buyer/seller transaction without altering the transaction structure or affecting its execution [10], [12]. Several distortion approaches are used to achieve this goal, such as mixing and generalization techniques [13].

Mixing techniques have been applied to cryptocurrency transactions to increase privacy. Mixing transactions allow the users to move digital coins from one user address to another without a direct link between the addresses. Mixing can be implemented as a centralized or decentralized service to provide anonymity for cryptocurrencies. In centralized mixing, [10], a third party performs the mixing by requesting users to send their coins to a mixing service address which sends the mixed coins received to different output addresses highlighted by each user [14]. However, the dependency on a third party to perform the mixing makes the system vulnerable to a single point of failure and can raise a severe privacy concern if the third party gets compromised or acts maliciously [15], [16]. Also, as studied in [13], the anonymity achieved by centralized mixing is proportional to the size of the addresses pool, where the larger the size of the pool, the better it is for the mixing service to be able to provide anonymity. CoinMixer [17] is a mixing service that requires, on average, between one to six hours to complete mixing transactions. Another service is MixCoin [18] which requires to have access to the actual mappings between input and output addresses for each transaction. Storing such information by the server can cause serious security issues if the mixing server is compromised.

It is important to highlight that centralized mixing solutions, in general, require several hours to be completed [13].

To address the previous concerns, decentralized mixing has been proposed [15] where mixing participants who would like to transact with the same amount of coins join together and exchange input and output addresses with each other. To construct a transaction, one of the participants gathers the exchanged input and output addresses with the appropriate amount of coins in a transaction that is then exchanged with the other participants to collect their signatures. Nevertheless, the level of anonymity provided by this approach is limited and depends on the number of participants in a mixing group [10], [11]. The protocol also has the challenge of finding enough participants who would like to transact with the same amount. Moreover, participants can have access to the input and output addresses, which is a severe privacy breach that some users cannot accept. Also, it has been proven that the protocol is vulnerable to Denial-of-Service (DoS) attacks in cases where a participant refuses to sign the transaction during the signature collection phase or when they spend the coins in the exchanged input address. This leads the decentralized mixed transaction to be rejected by the network, and instead, it could be treated as a double-spending transaction [19], [20]. Coinjoin [21] is an example of a decentralized mixing service that aims to break the link between input and output addresses in a transaction by mixing and joining input and output addresses of different transactions. Maurer *et al.* [22] have introduced some changes to the CoinJoin protocol by splitting the output of a single transaction before mixing it with outputs of other transactions and joining them in a single transaction. CoinParty [23] is another decentralized mixing service that relies on mixing peers to ensure reliable mixing even if one or more mixing peers acts maliciously. The performance of mixing peer solutions is affected by the heavy cryptography operations required for each transaction, and the multiple rounds of mixing the transaction has to undergo to ensure a better anonymity degree.

Generalization is another data distortion technique that provides anonymity. It refers to the process of replacing the actual data value with a less accurate one while preserving both its value and consistency as data and its privacy, as highlighted in [24]. Data generalization such as the K-anonymity model aims to ensure that for each data record, there exist at least (k-1) other records that are indistinguishable from it [24], [25]. To achieve that, attributes are classified into two main categories: Explicit Identifiers (EI) and Quasi Identifiers (QI). Explicit identifiers refer to attributes that hold private information which can uniquely distinguish a record in the data set, such as name or personal ID. Quasi-identifiers are attributes that don't enable identifying an individual in a data set separately, but when these attributes are combined, they can be used to identify individuals [25], [26]. To achieve anonymity, explicit identifiers are removed, and QI is generalized. Each specific value is replaced with a more general value or a range of values in numeric attributes. It ensures that even if an adversary knows the values of QI attributes for targeted records of data, it will not be possible to determine the record owner's actual identity since there are (k-1) other similar records in the data set. However, the main drawback of this approach is the loss of some information due to data distortion as part of the generalization process. In some cases, it can lead to losing the integrity of the shared data and its potential importance. In [27], authors have proposed the use of a generalization technique based on Heuristic K-anonymity in combination with a private blockchain platform in order to provide a higher level of confidentiality and anonymity. Similarly, Qiu *et al.* [28] proposed the use of a generalization technique based on k-anonymity to anonymize a user's actual geographical location before sending it to the network for processing. However, this solution suffers from several performance limitations. For instance, for each query, a user needs to send to the network $k - 1$ additional false queries, where k is chosen according to the desired level of anonymity for a given data set. Moreover, in order to validate a query and reply to users, the system needs at least 5 seconds.

B. DATA ENCRYPTION

Data encryption is another means to protect the privacy of transaction data in a blockchain network setting [10], [11]. The used encryption-based methods to achieve privacy in blockchain include: confidential ring signature [15], [29], [30], confidential transaction [31]–[33], Zero-Knowledge Proofs [30], [33]–[35], and homomorphic encryption [10], [15], [33].

Confidential ring signature refers to the process of signing a transaction by a group of participants, and hence, a verifier cannot determine which group member has produced the transaction's signature [30]. In other words, the public key that creates the transaction is hidden among other public keys within the same group. A group is constituted in such a way that all its members share the same amount of cryptocurrency and help in protecting the actual member who sends a particular transaction [29]. A vital advantage of this approach over the mixing technique is that the owner of the public key can construct the transaction without relying on third parties, which provides better security and anonymity. However, its main downside is that to construct a group signature for a transaction to send a specific amount of coins; there should exist other users that share the same amount of coins, which cannot always be guaranteed. Another limitation is that it is not possible to identify the signer of a transaction in case of a dispute since no group manager takes the role of assigning new members to the group or revoking group membership and managing conflicts [32]. A version of this technique is used by Monero [36]. Monero is a decentralized cryptocurrency that focuses on ensuring privacy and anonymity by hiding the transacted amount and addresses of both senders and receivers. However, since transactions in Monero are untraceable, in case the cryptocurrency is used for illegal activities, it is not possible to identify the parties involved in such activities.

Confidential transaction refers to transactions on a public blockchain network with a hidden content. A sidechain, known as Elements Alpha [31], uses such technique when attached to a parent blockchain like Bitcoin to make the transacted value hidden except to the entities involved in the transaction [33]. This protocol is mainly used as an extension to the bitcoin network. However, during the exchange of funds between the sidechain and the parent blockchain, the privacy can no longer be preserved [37]. Other solutions that use this technique include Monero and Mimblewimble [38], two blockchain protocols that use confidential transaction technique to hide the content of a transaction. Such a technique can be utilized along with other techniques such as mixing. For instance, ValueShuffle [22] is a mixing protocol based on CoinJoin that uses this technique to hide the payment values in transactions.

Zero-Knowledge Proofs (ZKP) is a cryptographic protocol that allows a prover to prove to the verifier that a particular statement is valid without revealing any additional information except the proof itself [33]. A form of ZKP known as Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zkSNARK) is used in some blockchain applications as it performs validation in a short time and is a non-interactive protocol. This means that there is no need for a synchronized or live communication between the prover and verifier since a message from the prover is sufficient to be verified by any verifier offline, as highlighted by [33]. In terms of performance and efficiency, this protocol requires high computational time to generate and validate the proof, which cannot be tolerated in some applications that require immediate response, such as the stock exchange. Also, this kind of scheme requires some sort of centralization where trusted parties need to generate common reference strings (CRS) and act as random string generators [34]. ZCash [39] is an example of a cryptocurrency that employs such a technique to provide anonymity and privacy for users transacting with it.

Homomorphic encryption is a form of encryption that allows performing a set of operations on encrypted data without revealing the actual data. These operations performed on the ciphertext yields the same result as if they were performed on the cleartext and then encrypted [32], [33]. Homomorphic encryption can be very useful to store encrypted data on the blockchain that can be later decrypted for auditing purposes. Homomorphic encryption, however, is limited by the type of operations that can be performed on the encrypted data and has limited performance compared to other privacy techniques [33]. Nevertheless, some of the existing blockchain protocols and solutions use homomorphic encryption to ensure privacy. For instance, Shrestha and Kim [40] highlighted the potential of the integration of blockchain based-IoT with homomorphic encryption to ensure the security and privacy of IoT data in a decentralized fashion. Mimblewimble [38], [41], which is a blockchain protocol, uses homomorphic encryption in addition to confidential transaction technique for better privacy and anonymity.

TABLE 1 provides a comparison between different privacy-preserving techniques by listing the advantages and disadvantages of each one. In summary, the existing privacy-preserving techniques cannot fulfill the stock exchange's privacy requirements if applied in their current forms for several reasons. First, some techniques such as mixing and confidential ring signature can only be applied to transactions with the same input or output. Second, the use of techniques that are based on confidential transactions relies on using a side-chain to achieve anonymity in a public blockchain network. Since our focus is on a private blockchain with a shared ledger that needs to be fully visible to everyone for transparency purposes, using side-chains is not a valid option. Third, techniques such as ZKP require a long time to validate transactions because of the used cryptography primitives. However, time-sensitive applications, including the stock exchange, cannot tolerate such an overhead. Moreover, techniques based on homomorphic encryption have limitations on operations applied to the encrypted data. They are computationally expensive for platforms such as the stock exchange, where transactions continuously update the investors' balances and generate trades. Finally, standard distortion techniques ensuring k-anonymity provide anonymity to numerical attributes by replacing them with a range of other numbers. However, this type of distortion cannot be directly used in the stock exchange since attributes such as shares or cash balances need to be anonymized while still maintaining their actual values for trading. In section IV, we introduce our privacy-preserving framework for a decentralized stock exchange platform. The framework addresses the privacy requirements of the stock exchange by using data distortion and encryption techniques to achieve anonymity, unlinkability, traceability, and non-repudiation.

III. PRIVACY REQUIREMENT OF BLOCKCHAIN-BASED STOCK EXCHANGE SYSTEM

In this section, we discuss and highlight the privacy requirements of the decentralized blockchain-based stock exchange platform. We first provide an overview of the architectural design of a blockchain-based decentralized stock exchange by referring to our previous work in [9]. The proposed architecture in [9] achieves full decentralization without introducing significant changes to the existing stock exchange trading logic, neither eliminating any of the traditional involved organizations. We, therefore, discuss the general privacy requirements in the stock exchange concerning regulation and then translate them to specific requirements by considering the decentralized stock exchange architecture in [9].

A. DECENTRALIZED CONSORTIUM BLOCKCHAIN-BASED STOCK EXCHANGE PLATFORM

The stock market is defined as a platform composed of financial and governmental organizations that participate in

TABLE 1. Comparision between different privacy-preserving techniques.

Privacy Technique	Advantage	Disadvantage	Examples
Centralized Mixing	1. Hides the links between transactions input and output addresses	1. Single point of failure 2. The level of anonymity is proportional to the address pool	CoinMixer [13], Mixcoin [18]
Decentralized Mixing	1. No dependency on mixing services 2. Addresses limitations of centralized mixing such as single point of failure	1. The level of anonymity is proportional to the number of participants 2. Participants know the input and output addresses 3. Needs participants that transact in the exact same amount 4. Vulnrable to DoS attacks	CoinJoin [22], CoinParty [23]
Generalization	1. Data anonymization 2. Ensures that there are at least k-1 similar records in the data set	1. Information loss due to the applied distortion 2. Can lead to losing the integrity and importance of information	Research work in [27] and [28]
Confidential ring signature	1. Hides the identity of real transaction signer using group signature 2. Does not rely on third parties to construct the transaction	1. Difficulty in finding participants who share the same amount of coins 2. It is not possible to find the signer of the transaction within the group 3. Difficult to manage in case of disputes and conflicts	Monero [29], [36]
Confidential Transaction	1. Only entities involved in a transaction can see transacted value	1. This technique can only be used with bitcoin 2. No privacy during exchanged between sidechain and mainchain	Elements [29] and Mimblewimble [38]
Zero-Knowledge Proofs (ZKP)	1. Proves statement in zero knowledge	1. suffers from high computational time and storage size	Zcash [39]
Homomorphic encryption	1. Arithmetic calculations on ciphertext is converted to calculations in cleartext	1. Limitations in homomorphic operations	MimbleWimble [38], [41], [40]

exchanging shares, bonds, or other securities in a transaction known as a trade [42]. A trade is generated as a result of a match between a buy and a sell order entered by individuals known as traders or investors. Each order consists of the investor's unique identifier (NIN), the type of the order (buy/sell), the type of traded shares, their quantities, and the bid or offers price. The main organizations/entities involved in the stock exchange include investors, Listed company, Broker, Custodian, Stock Exchange (SE), Central Securities Depository (CSD), Government, Financial Market Authority (FMA), and Central Bank (CB). For more details about the role of each entity, we refer the reader to [9]. Among them, FMA is responsible for regulating the market by monitoring the investor's trading accounts in case of any suspicious trades and ensuring that all the involved entities are following the regulation. The trading accounts are maintained by CSD, which takes the responsibility of creating and updating the investor trading accounts in terms of cash and share balances.

In [9], the traditional stock exchange has been transformed into a decentralized blockchain-based platform using a consortium (permissioned) Ethereum network. The proposed architecture in FIGURE 1 ensures high availability by distributing the ledger containing the trade-related transactions across all the network participants that are known and trusted. FMA is responsible for managing and deploying the smart contract and defining the permissions of each participant and what data it can see. It acts as an administrator for the overall platform. The shared ledger stores information related to the trading transactions that consists of the investor trading account, order types, and trades generated among the authorized entities participating in the consortium network. Since this is a permissioned based network, only authorized entities selected by FMA at the time of deploying the smart contract acts as nodes and are able to interact with the smart contract according to their defined roles. Each entity is identified using its public key address that it uses to send transactions to the network. To ensure entities are interacting with the smart contract as per their authorized role, we have ensured to add a condition in the smart contract for functions that can only be executed by a specific participant. For instance, the functionality of adding a new NIN account can only be executed by CSD as per its authorized role. By defining the participants and what function each can execute, we ensure that only authorized entities can interact with the smart contract following the predefined privileges.

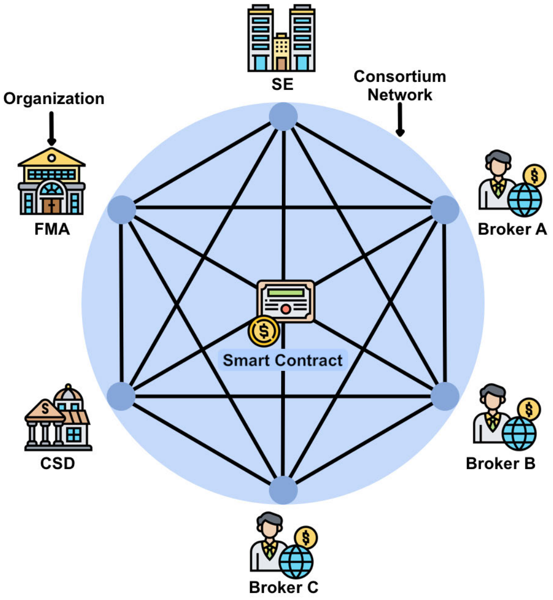

FIGURE 1. System architecture [9].

It is worth noting that not all entities in the blockchain network should have access to all the information in the shared ledger, and thus, we classify and analyze the information that needs to remain in the ledger with the necessary privacy and anonymity controls. Each order consists of the investor's unique identifier (NIN), the order type, the company name, number of shares, and bid or offer price. This information is shared across all the involved entities including, FMA, SE, CSD, and brokers. Once relevant orders are matched and trade is generated, the smart contract broadcasts it to the entities. This solution is aligned with the regulations of the stock market in terms of maintaining the financial and governmental organizations that are already involved in the traditional stock exchange platform. However, not all entities sharing the ledger are authorized to read all shared information, especially the investor's unique identifier (NIN). Therefore, it is necessary to ensure the required level of privacy is introduced to meet the stock exchange requirements in that regard.

B. PRIVACY REQUIREMENT

The decentralized architecture proposed in [9] is based on permissioned blockchain network in which the participants are known and trusted. The use of permissioned blockchain guarantees that only the known set of participants can access the ledger. However, if specific data within the ledger need to be only visible to a restricted subset of participants, the blockchain cannot ensure such a requirement. For instance, sensitive information such as the investor's identity should be recognized only by authorized participants such as FMA and CSD. To identify the sensitive information that should not be visible to all known and trusted participants, we first identify the information that needs to be executed on-chain. On-chain information must exist in the shared ledger to ensure the correct trading execution on the smart contract. In contrast, off-chain information can exist outside of the consortium network without impacting the platform's integrity. According to [33], performing a proper information classification maintains the efficiency in terms of storage management and execution time of the smart contract since only critical information needed by the smart contract remains in the shared ledger.

In TABLE 2 we perform the information analysis and define the information access and authorization. Each participant is granted access only to the allowed information as per their defined roles in the smart contract. For instance, Financial Market Authority (FMA) requires access to information related to all investors, brokers, companies, orders, and generated trades. However, information such as investor's data (Investor NIN account and its associated balance) should not be accessible by participants such as the Stock Exchange (SE) and the broker unless anonymity is applied. This anonymity should also be applied to the investor's data that is part of the buy/sell orders and trades. Unless the investor's identity cannot be identified, the information should not be accessible to brokers and the Central Securities Depository (CSD) in case of buy/sell orders.

After identifying what information each participant is authorized to access, we can classify the information hosted in our smart contract into on-chain and off-chain information. From the proposed classification, as shown in TABLE 3, the only information that needs to be anonymized is the investor's trading and orders information, as they need to be used on-chain and at the same time need to be private. Note here that only the investor's NIN and its associated balance need to be anonymized from the order information. Therefore, other order information (order type, the buyer or seller NIN, the company's name offering the shares, the number of shares to buy or sell, the bid or offer prices) can be made public as per the market regulation. However, if the identity of the investor is not hidden or encrypted, hence the NIN is not encrypted, it will allow unauthorized entities to trace all orders performed by the investor, identify the total shares and cash they own and analyze the investor's trading strategy including the profit or loss they make in each trading session. It will also be possible to trace the investor across different trading sessions to build a holistic mapping leading to stock price manipulations and attacks such as front-runner. Therefore, we strive in this paper to preserve the privacy of decentralized stock-exchange by fulfilling the following requirements:

TABLE 2. Data permissions by participants.

Access to Information	FMA	CSD	SE	Broker
Investor Account	Allowed	Allowed	Not Allowed	Not Allowed
Broker Information	Allowed	Allowed	Allowed	Allowed
Company Information	Allowed	Allowed	Allowed	Allowed
Buy Orders	Allowed	Not Allowed	Allowed	Not Allowed
Sell Orders	Allowed	Not Allowed	Allowed	Not Allowed
Trades	Allowed	Allowed	Allowed	Not Allowed

TABLE 3. Information classification.

Information	Classification	Privacy
Company Information	On-chain	Public
Investor Personal Information	Off-chain	Private
Investor Trading Information	On-chain	Private
Broker Information	Off-chain	Public
Orders Information	On-chain	Private

- **Anonymity:** The investor's identity is hidden such that entered orders and trades do not reveal the identity of the involved investor/s.
- **Unlinkability:** Trades and orders cannot be linked across different trading sessions.
- **Traceability:** Authorized participants can trace anonymous account addresses and link them to their original addresses.
- **Nonrepudiation:** Orders entered by the investor using anonymous addresses are signed by a temporarily private key only the investor can generate. Hence, investors cannot deny, withdraw, or cancel entered transactions.

IV. PRIVACY-PRESERVING FRAMEWORK FOR BLOCKCHAIN-BASED STOCK EXCHANGE PLATFORM

In this section, we present our privacy-preserving decentralized stock exchange framework. The proposed framework preserves the investors' trading accounts' privacy by ensuring that investors' private data are never shared. We define privacy for stock exchange as the *ability to ensure that any entered trading transaction is unlinkable to its trader's true account (NIN, balance) during and after a trading session.* The proposed framework meets this privacy requirement through *repeated anonymization* of investor accounts in such a way to make it infeasible to link trading transactions to their original investors. The framework consists of three phases: (1) accounts' balance anonymization phase; (2) anonymous account's identifier (NIN) generation phase; and (3) ledger updating phase.

The investor's unique identifier (NIN) is an *explicit identifier* that must be anonymized to hide the actual identity of the account's owner. The NIN information is anonymized by replacing the actual NIN with an anonymous one through an anonymous account-generation process described in Section IV-B. Moreover, to ensure long-term unlinkability between the actual NIN and the anonymous one, as well as to prevent the ability to infer any useful information by observing trading transactions performed by the anonymous NIN, the anonymization process needs to be repeated at regular intervals, i.e., before the start of each trading session.

By doing so, if an investor's anonymous address in one session was revealed, no information can be obtained to link these anonymous accounts with other anonymous accounts in any past or future trading sessions. Therefore, new anonymous accounts (NIN) need to be assigned to each investor for each new trading session.

The account's balance is another critical information and is considered to be a *quasi-identifier* attribute that may, if not anonymized, reveal the link between a set of anonymous NINs, belonging to the same investor by simply observing their balances. Possible techniques to anonymize a quasi-identifier are to use k-anonymity-based generalization or suppression methods, which involve applying changes to the values of the quasi-identifier attributes [24]. However, these types of anonymization cannot be applied to the balance kind of information, which is sensitive to changes and need to be kept unchanged to ensure a viable trading system. Therefore, we propose a new form of k-anonymity model that is based on data splitting. In this model, the total balance of an investor is split into a set of amounts and assigned to several anonymous accounts (anonymous NINs). There will also be a minimal amount that a splitted account can have to keep the account useful for trading, as will be explained in the next section. To ensure k-anonymity, the total balance is split so that each of the newly created anonymous accounts will be assigned a balance that should exist in at least $k - 1$ other accounts in the system. This prevents an adversary from linking an anonymous account with its original owner since $k - 1$ other accounts share the same balance. Moreover, because anonymization is repeatedly performed before each trading session, long-term unlinkability is also guaranteed.

It is worth noting that, for privacy purposes, the anonymization process cannot be performed on the blockchain. For this reason, we exclusively delegate this process to the CSD or FMA, which are the only organizations that have the right to access all investors' accounts.

A. K-ANONYMITY-BASED ACCOUNTS ANONYMIZATION

An investor account consists of two types of information; the NIN and the balance. To anonymize the investor's account, the actual investor's NIN is replaced by an anonymous one and the balance is split into multiple amounts. To satisfy k-anonymity property wherein at least k investor accounts are indistinguishable from one another, the balance of one investor is split into multiple amounts and assigned to different anonymous NIN in such a way each amount exists in at least $k - 1$ other (split) investors' accounts.

To illustrate the account splitting process, let's consider the following simple example. Assuming a set of six investor accounts A of NINs from 1 to 6 that are sorted in a descending order of balance, such that $A = \{(1, 793), (2, 661), (3, 618), (4, 475), (5, 465), (6, 462)\}$. To ensure k-anonymity, where $k = 3$, because the 3^{rd} (k^{th}) account (NIN$=$ 3) of balance 618 is smaller than all the first $k - 1$ accounts; i.e., 793 and 661 Thus, each of the two first accounts can be split into two new accounts. The first one has a balance equal to the 3^{rd} account balance; i.e., 618, and the second one contains the remaining balance. Therefore, the first account of NIN$=$ 1 becomes $(1_1, 618), (1_2, 175)$, and the second account of NIN$=$ 2 becomes $(2_1, 618), (2_2, 43)$. The three accounts of anonymous NINs 1_1, 2_1 and 3 with same balance 618 are now 3-anonymous. To ensure k-anonymity of the entire accounts in A, the same process is repeated for the remaining accounts including the new non k-anonymous accounts, i.e., $(1_2, 175)$ and $(2_2, 43)$. The new list of accounts is first sorted in a descending order of balance, and the process is repeated until no further splitting is possible. The final splitting result of the initial set of accounts A is as follows:

$$\begin{aligned}
&(1, 793)- > \{(1_1, 618), (1_2, 175)\}\\
&(2, 661)- > \{(2_1, 618), (2_2, 43)\}\\
&(3, 618)- > \{(3_1, 618)\}\\
&(4, 475)- > \{(4_1, 462), (4_2, 13)\}\\
&(5, 465)- > \{(5_1, 462), (1_2, 3)\}\\
&(6, 462)- > \{(6_1, 462)\}.
\end{aligned}$$

In the provided example, the new split accounts can still reveal some information about the actual balance of some accounts. For instance, it is clear from the previous splitting that at least one account has an actual total balance of 618 and another with a balance of 462. To further improve the accounts' privacy and hide this information, we slightly modify the previous algorithm. Let's first assume that the maximum share price in the market is *max_price*. The first step in the previous example consists of splitting each of the $k - 1$ accounts into two accounts, one with the k^{th} account's balance, i.e,. 618, and the other contains the remaining balance. In the new algorithm, the balance of the first split account will contain the result of dividing the amount of the k^{th} balance by *max_price* rounded down to the nearest integer using the floor function. The result is then multiplied by *max_price*, i.e., $\lfloor 618/max\_price \rfloor \times max\_price$. The final splitting result, of the previous example, using the improved algorithm when $max\_price = 100$ is as follows:

$$\begin{aligned}
&(1, 793)- > \{(\{(1_1, 600), (1_2, 193)\}\\
&(2, 661)- > \{(2_1, 600), (2_2, 61)\}\\
&(3, 618)- > \{(3_1, 600), (3_2, 18)\}\\
&(4, 475)- > \{(4_1, 400), (4_2, 75)\}\\
&(5, 465)- > \{(5_1, 400), (1_2, 65)\}\\
&(6, 462)- > \{(6_1, 400), (6_2, 62)\}.
\end{aligned}$$

Among the generated new anonymous accounts, some of them are not k-anonymous. These non k-anonymous accounts do not reveal any information about the actual original accounts. They will unlikely be used during the trading as they have only a small amount (remaining) of balance. The complete splitting process is given in Algorithm 1.

Algorithm 1 K-Anonymity-Based Accounts Splitting Algorithm

```
init :
    A; {sorted list of actual investor accounts (nin, balance)}
    S; {split accounts result}
    $N$; {the total # of actual investor accounts}
    k; {anonymity parameter)}
    $max\_price$; {the highest share price of the last session}
while len(A) >= k do
    to_remove ← [] {accounts to remove from A}
    to_add ← [] {accounts to add to A}
    for (nin, balance) in A do
        if balance >= A[k-1] then
            amount ← $max\_price$ * ⌊A[k-1]/$max\_price$⌋
            reminder ← balance - amount
            S.append( (getAnonNin(nin), amount) )
            to_add.append( (getAnonNin(nin), reminder) )
            to_remove.remove( (nin, balance) )
        end if
    end for
    A.remove(to_remove)
    A.append(to_add)
    A ← sort(A)
end while
for (nin, balance) in A do
    S.append(getAnonNin(nin), balance)
end for
```

B. GENERATING ANONYMOUS INVESTOR ACCOUNTS

In this section, we describe the process of generating anonymous NIN for each of the investor's split balances. Since the investor has a unique NIN used to interact with the decentralized stock exchange platform, anonymous NIN needs to be used instead to prevent linkability, which could lead to the reveal of confidential information such as the investor's identity. However, the anonymous NIN should allow authorized participants to trace and identify the original investor for auditing and market surveillance purposes. Therefore, to generate an anonymous NIN, Algorithm 2 is used.

Initially, each investor generates a public and private key denoted as pk and x, respectively. The investor also generates a random number r_0 and a secret key sk and shares these sets of information with authorized entities such as CSD and FMA. Based on this initially shared data, both the investor and authorized entities can independently generate anonymous NIN for the investor. The i^{th} anonymous NIN of the investor is generated using equation 1 shown below. It is important to highlight that an anonymous NIN represents a temporary public key of the investor to send buy/sell orders.

$$anon_{NIN_i} = r_i * pk \tag{1}$$

Here r_i is generated based on the previous r_{i-1} and the shared secret key sk using equation 2, as follows:

$$r_i = Enc(sk, r_{i-1}) \tag{2}$$

where $i >= 1$, and Enc is an encryption function.

Algorithm 2 Anonymous NIN Generation Algorithm

```
init :
    $x$; {The investor's private key}
    $pk = x * P$; {The investor's public key, where $P$ is a point on elliptic curves}
    $sk$; {secret key the investor shares with CSD and FMA}
    $r_i$; {random number, the initial value is denoted as $r_0$}
$r_i \leftarrow Enc\ (sk, r_i - 1)$
$anon_{NIN_i} \leftarrow r_i * pk$
$x_i \leftarrow r_i * x$
```

The corresponding private key of the i^{th} anonymous NIN is denoted as x_i and can only be generated by the investor using the initial private key x as illustrated in equation 3 below:

$$x_i = r_i * x \tag{3}$$

Since these keys are generated using elliptic curve cryptography, managing the keys and storing them do not require high storage or computational cost. In terms of storage, pk, x, sk, and r_i are 32 bytes each. This means that for 1 million investors, the total storage required for the generated keys is 128 Megabytes. Note that, it is not necessary to store the old keys that have been created for each investor. All the keys can be generated (if needed) from scratch, using pk, x, and r_0. In terms of computational cost, each key is generated with a single point-scalar multiplication, which is not expensive.

C. PRIVACY-PRESERVING FRAMEWORK

In this section, we present the main phases of the proposed framework to preserve the decentralized stock exchange's privacy. As illustrated by the sequence diagram in FIGURE 2, there are three main phases: (i) Initialization, (ii) Pre-trading and (iii) Trading. A detailed description of each of the three phases is included in the section.

1) INITIALIZATION PHASE

The steps in this phase occur when the platform is set up for the first time and whenever a new investor is created.

1) The investor generates public and private keys denoted as pk and x, respectively, a secret key sk, and a random variable r_0.
2) The investor shares sk and r_0 with CSD and FMA so both can independently generate new anonymous NINs for this investor using Algorithm 2

2) PRE-TRADING PHASE

The steps in this phase occur before the start of each trading session.

3) CSD generates a list of anonymous NINs for each investor and splits the balance across them using Algorithm 1.
4) CSD posts the list of all anonymous NINs with their associated balances into the smart contract.

5) Each investor generates a list of anonymous NINs using Algorithm 2 that match the investor's anonymous NINs posted into the smart contract by CSD.
6) For each generated anonymous NIN $anon_{NIN_i}$, the investor generates a private key denoted as x_i, which is used to sign buy/sell orders entered using the $anon_{NIN_i}$ address.

3) TRADING PHASE

The steps in this phase take place during the trading session. resume

7) the investor uses one of the generated anonymous NINs to enter a buy or sell order.
8) FMA traces entered orders to their original investors to ensure that market regulations are respected. Tracing entered orders can also be launched after the trading session.

D. THREAT MODEL

The proposed privacy-preserved stock exchange framework is designed for permissioned blockchain in which all participants are known and trusted. We consider the investor's original NIN account and the associated balance as private information that should not be revealed during and after trading even to the smart contract that holds the trading logic nor to the blockchain participants. However, very special entities, such as FMA and CSD will be authorized to view the original NINs and their associated balances, basically for monitoring and auditing purposes. The proposed framework ensures that investor information consisting of NIN and balances are anonymized using our privacy-preserving framework before sharing it in the ledger accessed by all participants. Since the blockchain is permissioned with known and trusted participants, no external entities can join the network, meaning that an adversary can only join the network by compromising one of the blockchain participants. We also assume honest but curious blockchain participants who might attempt to learn more about the investors' activities and balances. The objective of the adversary or curious participant is to de-anonymize the NIN accounts and their associated balances by trying to link different trading transactions with their original balances and NIN accounts. One of the motivations for an adversary to de-anonymize the NIN accounts is to monitor the trading activities, which in the end can lead to market manipulation. Several external pieces of information such as the total number of investors and statistical data about the percentage of investors based on nationality, gender, and age exists outside the shared ledger, hence could be also used by an adversary. We also consider that adversaries might collude with each other to share trading-related information. We consider all adversaries to be computationally bounded, i.e., it is infeasible for them to break the underlying cryptographic protocols.

E. STOCK EXCHANGE SMART CONTRACT

The stock exchange smart contract in [43] showcases the main functionalities that apply our privacy persevering framework in which investors NIN and balances are anonymized. The function addAnonymousNin() is used by CSD only every new session to add all investor's anonymous NINs with their associated balances to the smart contract so that investor's can use their anonymous NIN accounts to enter buy or sell orders by using BuyShares() and SellShares() functions. Before adding the new anonymous accounts, CSD need to revoke all previous accounts by calling lockAnonymousNin(). The function doMatch() is triggered whenever a buy or sell order is entered, and it searches all queued orders to find a match. If a match is found, a trade is generated and both matched orders are removed from the buy and sell queues. The stock exchange smart-contract is implemented using Solidity language, the de facto language for developing Ethereum smart-contract, and the entire code can be found in [43].

Our proposed framework ensures that investors can enter trades anonymously using the anonymous addresses provided, that their transactions cannot be traced across different trading sessions since CSD splits the accounts before the start of each new session. Also, the framework ensures that entities such as CSD and FMA can trace the anonymous addresses to their original addresses, which is one of the stock market regulations requirements. Moreover, nonrepudiation is ensured since each order entered by an anonymous address is signed by the investor's private key and validated by the smart contract. This validation has a tolerable overhead since the main steps that consume more time, such as splitting and assigning balances to the addresses, take place in the pre-trading phase.

The proposed framework achieves privacy by ensuring the following properties:

1) ANONYMITY

The investor's identity is anonymized by generalizing the investor's unique identity (NIN) and by splitting the investor balance. During a trading session, the account address and balance cannot reveal the actual account's investor.

2) UNLINKABILITY

Trades and orders cannot be linked across different trading sessions by performing repeated anonymity such that in each trading session, an investor has a new list of anonymous accounts and a new distribution of the balance among the anonymous addresses. It ensures that an adversary is not able to link anonymous addresses together across different trading sessions.

3) TRACEABILITY

Authorized participants can trace anonymous accounts addresses and link them to their original addresses. Our anonymous account-generation functionality allows each authorized participant such as FMA and CSD to independently generate anonymous addresses of investors and, hence, link generated anonymous addresses with the original investor's trading account.

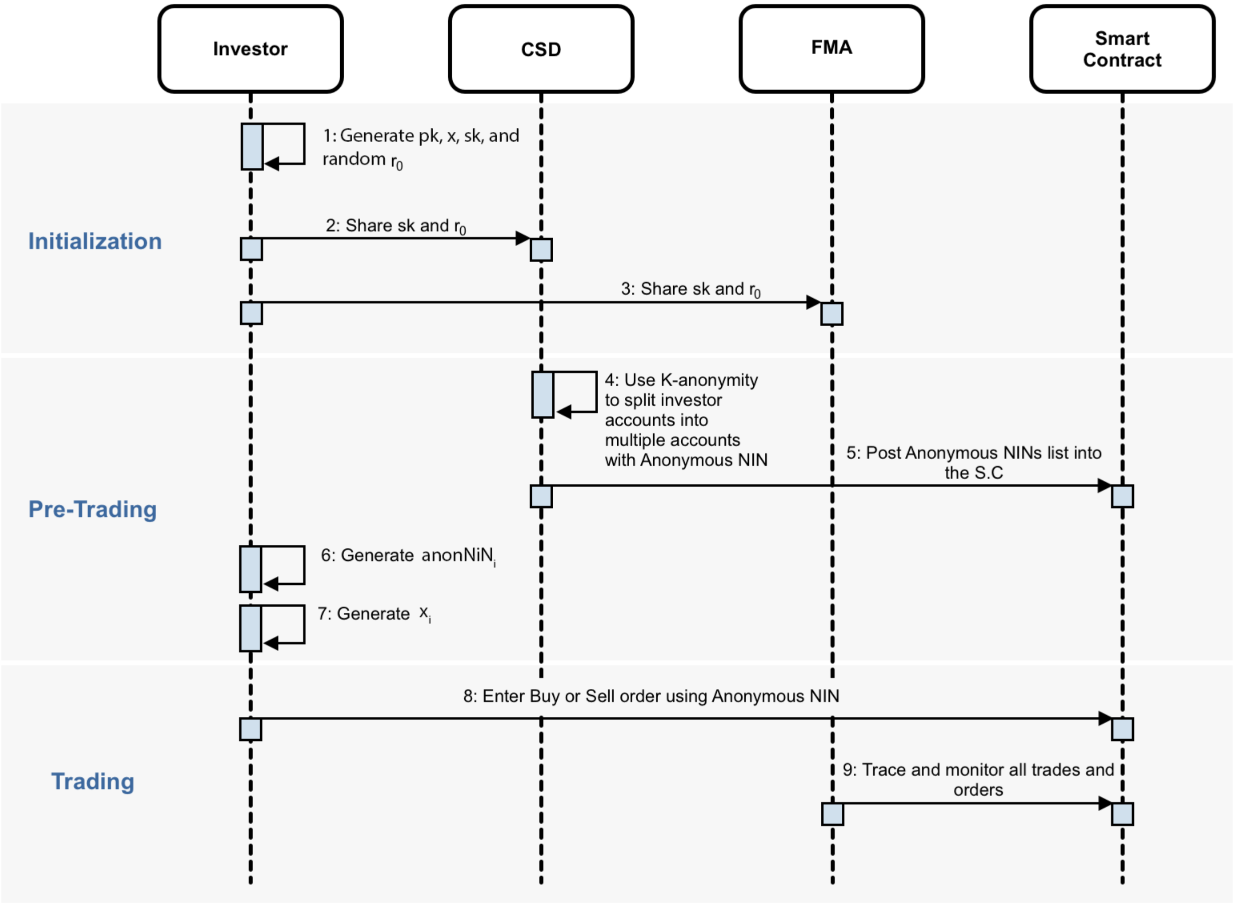
Investor
CSD
FMA
Smart Contract
Initialization
1: Generate pk, x, sk, and random r_0
2: Share sk and r_0
3: Share sk and r_0
Pre-Trading
4: Use K-anonymity to split investor accounts into multiple accounts with Anonymous NIN
5: Post Anonymous NINs list into the S.C
6: Generate anonNiN$_i$
7: Generate x_i
Trading
8: Enter Buy or Sell order using Anonymous NIN
9: Trace and monitor all trades and orders

FIGURE 2. Sequence diagram between participating entities and the StockExchange smart contract.

4) NON-REPUDIATION

Entered orders using anonymous NIN addresses are ensured to be executed by the investor and cannot be denied or withdrawn as the orders are validated by the smart contract and recorded in the shared ledger. The anonymous order is signed by a temporarily private key derived from the investor's permanent private key, and hence, only the investor can sign these anonymous NIN transactions.

F. SECURITY ANALYSIS

In this section, we analyze the security of the proposed framework against major attacks that target k-anonymity such as homogeneity and background knowledge attacks. **Homogeneity attack** causes the disclosure of private information when all the values of a sensitive attribute withing a same k-anonymity group of data are similar.

In **background attack**, even if the attributes' values are different, the knowledge of some external information can help in disclosing sensitive and private data [44], [45].

The proposed framework presented in this paper aims to ensure privacy by protecting both (i) *identity* and (ii) *attribute* disclosure. In stock exchange, the investor's unique identifier (NIN) is the *explicit identifier* that our proposed framework protects during trading by replacing the actual NIN with a new generated anonymous account, as explained in Section IV-B. The *attribute information* that is required during trading and at the same time needs to be protected against disclosure, in stock exchange, is the investors' balances. For this purpose, the proposed framework makes use of k-anonymity to generalize investors' balances using the proposed splitting algorithm given in Section IV-A. The splitting is made in such a way that each individual account's balance in the market cannot be distinguished from at least k-1 other accounts. Moreover, to ensure long-term unlinkability between the actual NIN and the anonymous NIN, the anonymization process is repeatedly performed before each new trading session. We define unlinkability as the unfeasibility of an adversary or a curious and legit participant to link between the investor's original NIN and any of the anonymous NINs generated for that investor. The process of generating anonymous NIN accounts relies on using discrete logarithm algorithm using the following equation:

$$anon_{NIN_i} = r_i * pk \tag{4}$$

It is infeasible to perform the inverse for the above equation to determine the original NIN from any anonymous NIN.

For **homogeneity attack**, the proposed framework is secure as the data are grouped based on the splitted balances and not the whole actual ones. Therefore, even if all the balances within the same group are identical, this will not be helpful for an attacker to disclose the actual identity of the investors behind those balances. For **background attack**, because the ledger contains only the anonymous NINs and splitted balances, no external information can be used to help in disclosing or inferring the actual identities of the different ledger entries. For such an attack to happen, an attacker generally relies on the existence of at least one actual additional quasi-identifier attribute. In summary, the fact that (i) the shared ledger contains only two attributes that both of them are distorted through anonymization, for the NIN, and splitting, for balance, and (ii) the framework ensures long-term unlinkability for different trading sessions by performing repeated anonymization before the start of each trading session, make the system robust against the two previous attacks and ensures that an adversary cannot link information from different past or future trading sessions together.

V. PERFORMANCE EVALUATION

This section evaluates the performance of the proposed privacy-preserving framework for a decentralized stock exchange platform. The evaluation is based on calculating the additional number of anonymous NINs to achieve different anonymity levels, k, and the corresponding execution time for processing buy/sell order transactions entered using all the generated anonymous NINs. The execution time is compared to the time it takes to process the same orders entered using only the original NINs to assess the generated overhead when the proposed privacy-preserving framework is used. We also study how the allocation of balances between original investor accounts affects the splitting process and the total number of generated anonymous NINs to achieve a targeted level of anonymity.

A. EXPERIMENT

To evaluate the total number of generated anonymous accounts when applying the splitting algorithm, we conducted several simulation experiments under different (i) stock-market sizes, (ii) anonymity levels, and (iii) allocations of balances between the different investors' accounts. In the experiment, we evaluate the total number of generated accounts by considering four stock-market sizes from small to large with 100,000, 250,000, 500,000, and 1,000,000 investors. Four anonymity levels are considered by varying k : 2, 3, 4, 5. To distribute the balances between the different investors accounts, both uniform and Gaussian distributions are employed. For the Gaussian distribution, different standard deviation values are used to set variation or dispersion of the investors' balances. We chose four different values for the standard deviation to make small to large dispersion between balances. All the experiments are based on a fixed max-price: 100 and averaged over 100 runs.

B. RESULTS

FIGURE 3 illustrates the impact of the market size and anonymity level (k) on the average number of the additional generated accounts to ensure k-anonymity, where the investors' balances are uniformly distributed between \$100 and \$10, 000, 000. The plot shows the following:

- When k increases, the percentage of the newly generated accounts also increases.
- When the market size increases, the percentage of the newly generated accounts decreases.

For instance, when $k = 5$ and for a market size of 250, 000 investors, the total number of accounts after splitting is 358, 493, representing an increase of 43.4%. For bigger market size, such as 500, 000 investors and for the same $k = 5$, the total number of accounts after splitting is 592, 000, representing a growth in the number of accounts by only 18.4%. It is because when k increases, more splitting is required to ensure that there are at least $k - 1$ other accounts with the same balance for each account. However, when the market size increases, i.e., more investors, less splitting is required as the probability of having accounts with similar or close balances increases.

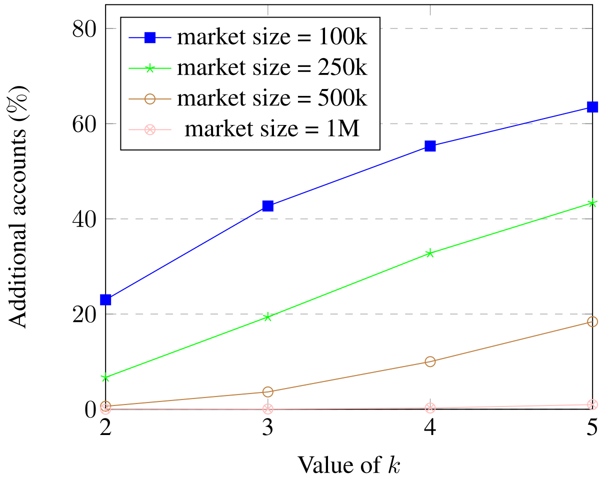

FIGURE 3. Percentage of additional accounts in order to achieve k-anonymity for different market sizes.

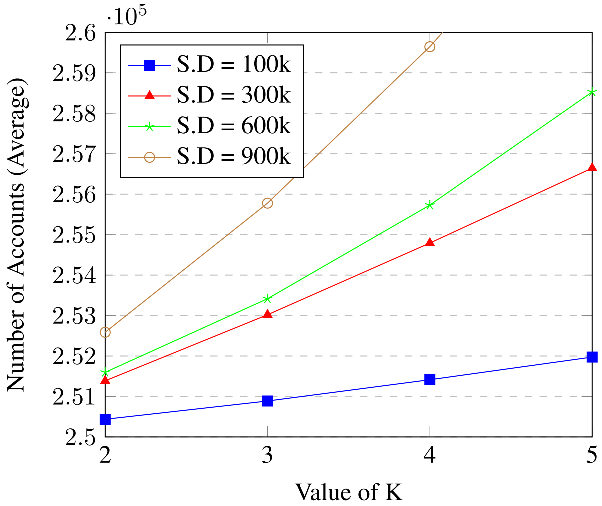

FIGURE 4. Average number of anonymous accounts generated for a market size of 250k with different standard deviation values for the balances.

TABLE 4. Percentage of execution time growth in comparison to original accounts (market size = 250,000, execution time without K-anonymity is 2,142 s).

Value of K	K-anony. Accounts (Average)	Total number of transactions	Total Execution Time	Time Difference (%)
2	266,772	800,316	2,286.6 s	6.7%
3	298,492	895,476	2,558.5 s	19.4%
4	332,065	996,195	2,846.2 s	32.8%
5	358,439	1,075,317	3,073.3 s	43.4%

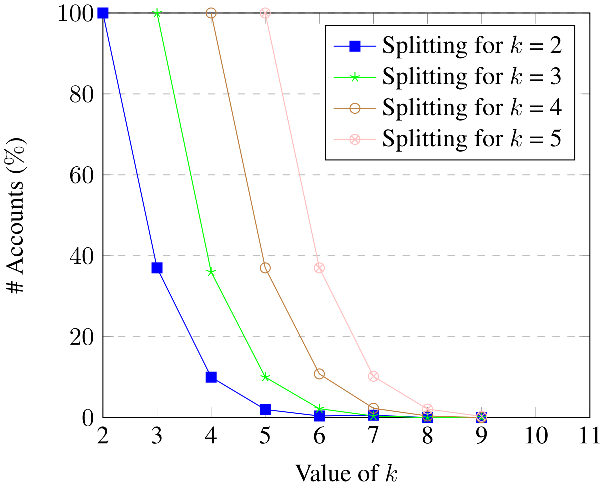

FIGURE 5. The anonymity level of accounts for different values of k for a market size = 100,000.

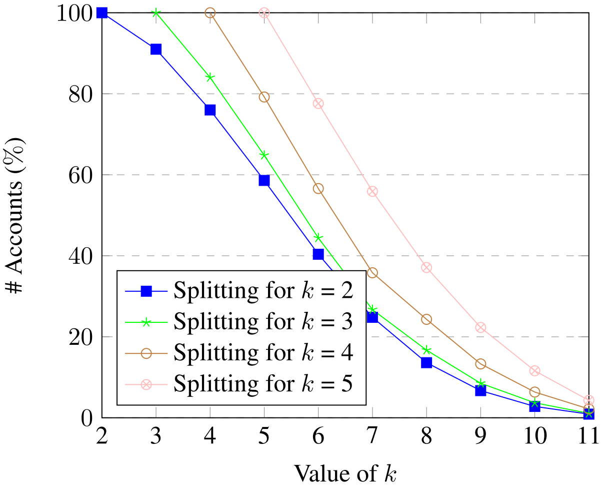

FIGURE 6. The anonymity level of accounts for different values of k for a market size = 500,000.

To further validate these results, we run the same experiment by considering that the balances among the investors follow a Gaussian distribution rather than a Uniform distribution. We set the mean of the Gaussian distribution to \$1, 000, 000, and consider four different values for the standard deviation σ: 100,000, 300,000, 600,000, 900,000. As shown in FIGURE 4, for a market size of 250, 000 investors, when the dispersion level among the investors' balances increases, i.e., standard deviation increases, the number of newly generated accounts also increases.

We also assess in Table 4 the impact of the newly generated accounts on the transactions execution time for a market size of 250, 000 investors and for different anonymity levels (k). For each selected k anonymity level, we consider the worst case scenario where each k-anonymous account will be used to enter exactly two trading transactions (buy and sell orders) and the two orders will be used to sell (resp. buy) all investors available shares (resp. balances). Therefore, to calculate the total number of generated transactions (buy, sell and trade), the number of anonymous accounts is multiplied by 3. We compare the result with the generated transaction execution time from the original 250, 000 initial accounts without splitting, which is 2,142 seconds. As per Table 4 and FIGURE3, we conclude the following:

- The execution time increases with the increase of the number of the newly generated anonymous accounts and the anonymity level k.
- For a selected anonymity level k, the increase in the percentage of execution time is equivalent to the increase in the percentage of the newly generated accounts.
- The execution time decreases as the market size increases.

For instance, when a high anonymity-level such as $k = 5$ is selected, the time needed to process all the transactions generated by anonymous accounts is 3, 073.3 seconds, representing an increase of about 43.4% compared to the time it takes to process transactions generated by the original account which is 2, 142 seconds. The 43.4% growth in execution time is also equivalent to the percentage of accounts growth shown in FIGURE 3. For a lower anonymity level, $k = 2$, the growth in the execution time and generated accounts is about 6.7%. As the size of the market increases, the execution overhead decreases. For instance, in a market with 500,000 initial accounts, the growth in accounts and execution time is 18.4% when $k = 5$ and 0.06% when $k = 2$ as shown in FIGURE 3. The execution time overheads for the market sizes presented in FIGURE 3 are acceptable by the stock exchange requirements with a worst case scenario of 63.5% growth in execution time when the market size is small and consists of only 100, 000 accounts, for $k = 5$.

During the experiments, we noticed that the anonymity level of some accounts is higher than the predefined one. For instance, if the target anonymity level $k = i$, there will be some accounts with an anonymity level $k > i$. To evaluate the ratio of these accounts, we measured the anonymity level of each account after the splitting for two market sizes: small with only 100,000 investors and large with 500,000 investors. The results are reported in FIGURE 5 and FIGURE 6, respectively. As shown in the figures, the ratio of accounts with an anonymity level that exceeds the predefined one increases with the market size growth. For instance, for a market with 100, 000 investors, as shown in FIGURE 5, when the targeted anonymity level during the splitting is $k = 2$, there are about 37% of the accounts with anonymity $k = 3$. However, for a market with 500, 000 investors, as shown in FIGURE 6, there are more than 91% of the accounts with anonymity $k = 3$. The same observation also applies for the other anonymity levels when $k = 3, 4$ and 5. This also confirms our previous results regarding the effect of balances distribution on the required number of new accounts to reach a desired level of anonymity, i.e., the closer the stocks between the investors, the higher the level of anonymity that can be reached with less overhead.

VI. CONCLUSION

This paper has presented a privacy-preserving framework that meets the requirements of a blockchain-based stock exchange platform in terms of privacy. In this framework, the privacy of investors' accounts (NIN) and balance is preserved by ensuring that all accounts are k-anonymous. This is achieved through applying repeated anonymity for both the NIN and balance. New anonymous accounts are generated, and balances are split and distributed among the new anonymous accounts in such a way to ensure that at least k accounts have the same balance. Furthermore, to ensure long-term unlinkability, this process is repeated every new trading session. We authorize blockchain ledger updates with new anonymous accounts used only by approved entities (e.g., CSD). For this purpose, we defined a non-interactive protocol between the investors and the authorized entities to create anonymous accounts without any communication overhead. By relying on the authorized entity to update the ledger, the proposed framework also ensures tractability and non-repudiation properties for trading transactions. To validate the effectiveness of the proposed framework on achieving the desired privacy, we conducted several experiments by considering different market sizes, anonymity levels, and the distribution of balances among investors. The results demonstrated the solution's efficiency where 100% anonymity can be achieved with acceptable transactions execution time overhead.

REFERENCES

[1] M. S. Nazir, M. M. Nawaz, and U. J. Gilani, "Relationship between economic growth and stock market development," *Afr. J. Bus. Manage.*, vol. 4, no. 16, pp. 3473–3479, 2010.

[2] *The Role of Stock Exchanges in Fostering Economic Growth and Sustainable Developmnt*. Accessed: Jan. 21, 2021. [Online]. Available: https://unctad.org/system/files/official-document/WFE_UNCTAD_2017_en.pdf

[3] P. Zhong, Q. Zhong, H. Mi, S. Zhang, and Y. Xiang, "Privacy-protected blockchain system," in *Proc. 20th IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2019, pp. 457–461.

[4] C. Chaturvedula, N. P. Bang, N. Rastogi, and S. Kumar, "Price manipulation, front running and bulk trades: Evidence from India," *Emerg. Markets Rev.*, vol. 23, pp. 26–45, Jun. 2015.

[5] C. Comerton-Forde, T. J. Putniš, and K. M. Tang, "Why do traders choose to trade anonymously?" *J. Financial Quant. Anal.*, vol. 46, no. 4, pp. 1025–1049, Aug. 2011.

[6] V. Mavroudis, "Market manipulation as a security problem," 2019, *arXiv:1903.12458*.

[7] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: Challenges and directions," *IEEE Security Privacy*, vol. 16, no. 4, pp. 38–45, Jul./Aug. 2018.

[8] C. Pop, C. Pop, A. Marcel, A. Vesa, T. Petrican, T. Cioara, I. Anghel, and I. Salomie, "Decentralizing the stock exchange using blockchain an ethereum-based implementation of the bucharest stock exchange," in *Proc. IEEE 14th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2018, pp. 459–466.

[9] H. Al-Shaibani, N. Lasla, and M. Abdallah, "Consortium blockchain-based decentralized stock exchange platform," *IEEE Access*, vol. 8, pp. 123711–123725, 2020.

[10] B. Li and Y. Wang, "RZKPB: A privacy-preserving blockchain-based fair transaction method for sharing economy," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1164–1169.

[11] B. Li, Y. Wang, P. Shi, H. Chen, and L. Cheng, "FPPB: A fast and privacy-preserving method based on the permissioned blockchain for fair transactions in sharing economy," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1368–1373.

[12] Y. Jiang, C. Wang, Y. Wang, and L. Gao, "A privacy-preserving E-commerce system based on the blockchain technology," in *Proc. IEEE Int. Workshop Blockchain Oriented Softw. Eng. (IWBOSE)*, Feb. 2019, pp. 50–55.

[13] T. de Balthasar and J. Hernandez-Castro, "An analysis of bitcoin laundry services," Sep. 2017, doi: 10.1007/978-3-319-70290-2_18.

[14] Q. Wang, X. Li, and Y. Yu, "Anonymity for bitcoin from secure escrow address," *IEEE Access*, vol. 6, pp. 12336–12341, 2018.

[15] K. Singh, N. Heulot, and E. B. Hamida, "Towards anonymous, unlinkable, and confidential transactions in blockchain," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1642–1649.

[16] H. Halpin and M. Piekarska, "Introduction to security and privacy on the blockchain," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Apr. 2017, pp. 1–3.

[17] *Coinmixer*. Accessed: 2021. [Online]. Available: https://coinmixer.online

[18] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. Kroll, and E. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," Mar. 2014, pp. 486–504, doi: 10.1007/978-3-662-45472-5_31.

[19] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, *Sybil-Resistant Mixing for Bitcoin*. New York, NY, USA: Association for Computing Machinery, 2014, doi: 10.1145/2665943.2665955.

[20] J. H. Ziegeldorf, R. Matzutt, M. Henze, F. Grossmann, and K. Wehrle, "Secure and anonymous decentralized bitcoin mixing," *Future Gener. Comput. Syst.*, vol. 80, pp. 448–466, Mar. 2018.

[21] *Coinjoin*. Accessed: 2021. [Online]. Available: https://coinjoin.io/en

[22] F. K. Maurer, T. Neudecker, and M. Florian, "Anonymous CoinJoin transactions with arbitrary values," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 522–529.

[23] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle, "CoinParty: Secure multi-party mixing of bitcoins," in *Proc. 5th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2015, pp. 75–86, doi: 10.1145/2699026.2699100.

[24] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient k-anonymization using clustering techniques," in *Proc. DASFAA*, in Lecture Notes in Computer Science, vol. 4443, Feb. 2007, pp. 188–200, doi: 10.1007/978-3-540-71703-4_18.

[25] X. He, H. Chen, Y. Chen, Y. Dong, P. Wang, and Z. Huang, "Clustering-based k-anonymity," May 2012, pp. 405–417, doi: 10.1007/978-3-642-30217-6_34.

[26] H. J. Thathagar and V. B. Vaghela, "Privacy preserving by anonymization approach," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 5, no. 11, pp. 86–90, 2017.

[27] B. Sowmiya and E. Poovammal, "A heuristic K-anonymity based privacy preserving for student management hyperledger fabric blockchain," *Wireless Pers. Commun.*, vol. 2021, May 2021, doi: 10.1007/s11277-021-08582-1.

[28] Y. Qiu, Y. Liu, X. Li, and J. Chen, "A novel location privacy-preserving approach based on blockchain," *Sensors*, vol. 20, no. 12, p. 3519, Jun. 2020.

[29] S. Noether and A. Mackenzie, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–18, Dec. 2016.

[30] M. Chase and A. Lysyanskaya, "On signatures of knowledge," Aug. 2006, pp. 78–96, doi: 10.1007/11818175_5.

[31] A. Poelstra, A. Back, M. Friedenbach, G. Maxwell, and P. Wuille, "Confidential assets," in *Proc. FC Int. Workshops, BITCOIN, VOTING, WTSC*, Curaçao, Jan. 2019, pp. 43–63, doi: 10.1007/978-3-662-58820-8_4.

[32] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–34, 2019.

[33] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019.

[34] J. Duan, L. Gu, and S. Zheng, "ARCT: An efficient aggregating ring confidential transaction protocol in blockchain," *IEEE Access*, vol. 8, pp. 198118–198130, 2020.

[35] T. Koens, C. Ramaekers, and C. Van Wijk. (2018). *Efficient Zero-Knowledge Range Proofs in Ethereum*. [Online]. Available: ING,blockchain@ing.com

[36] Y. Li, G. Yang, W. Susilo, Y. Yu, M. H. Au, and D. Liu, "Traceable monero: Anonymous cryptocurrency with enhanced accountability," *IEEE Trans. Depend. Sec. Comput.*, vol. 18, no. 2, pp. 679–691, Mar. 2021.

[37] D. Yang, J. Gavigan, and Z. Wilcox-O'Hearn, "Survey of confidentiality and privacy preserving technologies for blockchains," *R3 Res.*, vol. 1, pp. 1–32, Nov. 2016.

[38] G. Fuchsbauer, M. Orrù, and Y. Seurin, "Aggregate cash systems: A cryptographic investigation of mimblewimble," Apr. 2019, pp. 657–689, doi: 10.1007/978-3-030-17653-2_22.

[39] G. Fuchsbauer, "Subversion-zero-knowledge SNARKs," in *Proc. IACR Int. Workshop Public Key Cryptogr.*, Springer, 2018, pp. 315–347.

[40] R. Shrestha and S. Kim, "Integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities," in *Role of Blockchain Technology in IoT Applications* (Advances in Computers), S. Kim, G. C. Deka, and P. Zhang, Eds. Amsterdam, The Netherlands: Elsevier, 2019, vol. 115, pp. 293–331. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0065245819300269

[41] Y. Zheng, H. Ye, P. Dai, T. Sun, and V. Gelfer, "Confidential assets on mimblewimble," Cryptol. ePrint Arch., Tech. Rep. 2019/1435, 2019. [Online]. Available: https://ia.cr/2019/1435

[42] V. V. Bhandarkar, A. A. Bhandarkar, and A. Shiva, "Digital stocks using blockchain technology the possible future of stocks?" *Int. J. Manage.*, vol. 10, no. 3, pp. 44–49, 2019.

[43] (2021). *Stock Exchange Smart-Contract*. [Online]. Available: https://github.com/noureddinel/Privacy-Preserving-Stock-Exchange-Smart-Contract

[44] Q. Wang, Z. Xu, and S. Qu, "An enhanced K-Anonymity model against homogeneity attack," *J. Softw.*, vol. 6, no. 10, pp. 1945–1952, Oct. 2011.

[45] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, p. 3, 2007.

HAMED AL-SHAIBANI received the B.Sc. degree (Hons.) in computer science from Qatar University, Doha, Qatar, in 2010, and the M.Sc. degree in strategic business unit management from HEC Paris, Doha, in 2016. He is currently pursuing the Ph.D. degree in computer science and engineering with Hamad Bin Khalifa University, Doha. His main research interests include blockchain, cybersecurity, and networking.

NOUREDDINE LASLA (Member, IEEE) received the B.Sc. degree in computer science from the University of Science and Technology Houari Boumediene (USTHB), in 2005, the M.Sc. degree in computer science from the Superior Computing National School (ESI), in 2008, and the Ph.D. degree in computer science from USTHB, in 2015. He is currently a Postdoctoral Research Fellow with the Division of Information and Computing Technology, Hamad Bin Khalifa University, Qatar, with expertise in distributed systems, network communication, and cyber security.

MOHAMED ABDALLAH (Senior Member, IEEE) received the B.Sc. degree from Cairo University, in 1996, and the M.Sc. and Ph.D. degrees from the University of Maryland at College Park, in 2001 and 2006, respectively. From 2006 to 2016, he held academic and research positions at Cairo University and Texas A&M University at Qatar. He is currently a Founding Faculty Member with the rank of an Associate Professor at the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). He has published more than 150 journals and conferences and four book chapters. He has co-invented four patents. His current research interests include wireless networks, wireless security, smart grids, optical wireless communication, and blockchain applications for emerging networks. He was a recipient of the Research Fellow Excellence Award at Texas A&M University at Qatar, in 2016, the Best Paper Award in multiple IEEE conferences, including IEEE BlackSeaCom 2019 and the IEEE First Workshop on Smart Grid and Renewable Energy, in 2015, and the Nortel Networks Industrial Fellowship for five consecutive years (1999–2003). His professional activities include the Track Co-Chair of the IEEE VTC Fall 2019 Conference, the Technical Program Chair of the Tenth International Conference on Cognitive Radio Oriented Wireless Networks, the technical program committee member of several major IEEE conferences, and an Associate Editor for IEEE Transactions on Communications and IEEE Open Journal of the Communications Society.

SPIRIDON BAKIRAS (Senior Member, IEEE) received the B.S. degree in electrical and computer engineering from the National Technical University of Athens, in 1993, the M.S. degree in telematics from the University of Surrey, in 1994, and the Ph.D. degree in electrical engineering from the University of Southern California, in 2000. He is currently an Associate Professor with the College of Science and Engineering, Hamad Bin Khalifa University, Qatar. Before that, he held teaching and research positions at Michigan Technological University, The City University of New York, The University of Hong Kong, and The Hong Kong University of Science and Technology. His current research interests include security and privacy, applied cryptography, mobile computing, and spatiotemporal databases. He is a member of ACM. He was a recipient of the U.S. National Science Foundation (NSF) CAREER Award.