DOI: https://doi.org/10.21123/bsj.2023.8164

Secure Smart Contract Based on Blockchain to Prevent the Non-Repudiation Phenomenon

Noor Sabah Mohammed[1] * ***Omar A. Dawood***[1] ***Ali M. Sagheer***[1]
Ahmed Adil Nafea[2]

[1] College of Computer Science and IT, University of Anbar, Ramadi, Iraq
[2] Center for Artificial Intelligence Technology (CAIT), Faculty of Information Science and Technology Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor, Malaysia
*Corresponding author: noorsabah123456noor@uoanbar.edu.iq
E-mail addresses: omar-abdulrahman@uoanbar.edu.iq, ali.m.sagheer@gmail.com, ahmed.adil.nafea@gmail.com

Received 24/11/2022, Revised 13/3/2023, Accepted 14/3/2023, Published Online First 20/5/2023, Published 01/1/2024

Abstract:

Blockchain is an innovative technology that has gained interest in all sectors in the era of digital transformation where it manages transactions and saves them in a database. With the increasing financial transactions and the rapidly developed society with growing businesses many people looking for the dream of a better financially independent life, stray from large corporations and organizations to form startups and small businesses. Recently, the increasing demand for employees or institutes to prepare and manage contracts, papers, and the verifications process, in addition to human mistakes led to the emergence of a smart contract. The smart contract has been developed to save time and provide more confidence while dealing, as well as to cover the security aspects of digital management and to solve negotiation concerns. The smart contract was employed in creating a distributed ledger to eliminate the need for centralization. In this paper, a simple prototype has been implemented for the smart contract integrated with blockchain which is simulated in a local server with a set of nodes. Several security objectives, such as confidentiality, authorization, integrity, and non-repudiation, have been achieved in the proposed system. Besides, the paper discussed the importance of using the Blockchain technique, and how it contributed to the management of transactions in addition to how it was implemented in highly transparent real-estate scenarios. The smart contract was employed in creating a distributed ledger to eliminate the need for centralization. The elliptic-curve public key has been adopted as an alternative for the RSA in a signature generation/verification process and encryption protocol. For secure transactions, The Secure Socket Layer (SSL) also has been adopted as a secure layer in the web browser. The results have been investigated and evaluated from different aspects and the implementation was in a restricted environment. Experiments showed us the complexity of time and cost when using the (ECC) algorithm and using (RSA) algorithm depending on the size and length of the key. So if the size of the key in (ECC) equals (160) bits, and it corresponds to (1024) bits in (RSA), which is equivalent to 40% for (ECC) and 30% for (RSA). As a result, the (ECC) algorithm is complex, its key is smaller and the process of generating the key is faster, so it has achieved a high level of security.

Keywords: Blockchain, Cryptography, Digital Signature, Hash Function, Non-Repudiation, Smart contract.

Introduction:

On peer-to-peer networks, blockchain is considered one of the distributed ledger technologies controlled via multiple peers. This system works with no requirement for data storage management or a centralized administrator. Data is dispersed among multiple nodes, with encryption and replication ensuring data quality. The notion of blockchain was introduced on October 31, 2008, when Nakamoto published a white paper [1]. Because data is saved on each one of the nodes operating in each of the different networks, such a network is commonly indicated as a distributed registry. The hash regarding the previous block's record is used for combining a transaction group into blocks of transactions connected in the chain in blockchain networks[2]. As a result, the primary security aspect of blockchain networks is enforced as a property of immutability. The further along the chain a block is (the older it is), the more data it contains is protected against alterations [3,4]. Since it allows databases and peers in a trustworthy environment to be maintained (publicly) in a safe manner, the relevance of smart contract integration regarding blockchain technology will become a focus. Intelligent and permanent contracts are trackable. Every financial detail is used in a smart contract and implemented immediately.

The cryptography's hash function is used for encrypting transactions, which are checked with the nodes. The transaction's hash value is connected to its preceding hash value. No one may change or adjust it after the transaction has been inserted into the Blockchain; furthermore, the transaction can be publicly accessed, giving network accountability [5,6].

Technological transformations and digital features in the legal field have cast a shadow over various transactions, especially modern contracting methods such as the smart contract[7]. Currently, the security of smart contract administration is considered one of the major pressing challenges[8,9]. Blockchain technology has also emerged as the most innovative application to support smart contracts[10]. The smart contract technology aids in the management of real estate transactions by ensuring transaction reliability, transparency, and accuracy, as well as overcoming the present mechanism's pivotal capability in executing transactions via a real estate broker[11]. A secure environment for organizing, storing, exchanging, and retrieving information is provided by an efficient information management process [12,13]. Also, a robust security level and a decentralized ledger are two properties of the blockchain, whereas non-repudiation is a key information security feature [14,15]. Achieving non-repudiation through the use of a digital signature, as well as improving it, is also a possible approach. Most governmental document systems continue to rely on physical records, while others have transitioned to digital recording via a centralized manner[16].

The main contribution of this paper is to suggest a secure smart contract that depends on Blockchain with decentralized management and to prevent a non-repudiation scheme. The present work aims at providing a secure model with a more confident decentralized contract in addition to providing more effective smart contract protection through several security requirements, for instance, confidentiality, authorization, accountability, integrity, and authenticity. The essence idea focused on the used of blockchain technology with the smart contract and how to apply it in a real-estate. The real-estate scenario includes the establishment of a distributed ledger, which eliminates the need for centralization and (TTP). Where the proposed model has the ability to achieve a substantial change in the way international trade and business are performed by accelerating the transactions, finishing paperwork, and reducing cost-efficiency.

This proposed is organized into four sections in addition to the current section and the rest are outlined as follows: Section two: entitled Related Work covers the theoretical background of smart contracts and blockchain and blockchain challenges, as well as smart contract security issues. Section three entitled System Design: This section attempts to show the details of the design and applied of the proposed solutions assisted by flow charts. Section four entitled Results and Discussion involves the experimental results and the analytical discussion about the cost and security aspects of blockchain and smart contracts. Section five: entitled Conclusion and Future Works presents the conclusion of this proposed and suggestions for future works.

Related Work:

Zyskind et al. [1] presented a lightweight decentralized blockchain data management architecture to protect personal data and ensure users' ownership and control their data. Off-chain data storage was used in the proposed strategy to boost efficiency. In addition, using the protocol that transforms a blockchain into an autonomous access-control manager that does not rely on third-party trust.

Clack et al. [3] studied four fundamental topics of smart contracts: terminology, automation, enforceability, and semantics. They concluded that legal contracts are a simple framework based on two main aspects: operational and non-operative, as well as smart contract templates that can be implemented as electronic representations. This approach resulted in the creation of three separate contracts between Ricardian and Design. In this context, they also discovered the design scene to increase the parameterization of complex first-grader types and long-term work that would contribute to the acceptance of official languages in the courts.

McCorry et al. [5] a decentralized implementation and a self-voting protocol were suggested that can be implemented over the internet. The implementation was performed with a high degree of voting privacy using blockchain because it was suitable for council elections and was written as a smart Ethereum contract and was not dependent on a stable electoral privacy authority, as it was a self-adhesive protocol that makes it was possible to protect voting privacy. As for their further research, they will conduct national blockchain elections and will need a special blockchain where large numbers of transactions are stored on the chain and managed in a manner similar to the central RS-coin and examined the smart contract limitation used by Ethereum.

Karamitsos et al. [17] investigated the impact of smart contracts and their various components on implementation in the real estate market. The benefits of using smart contracts and blockchain technology for real estate are as follows: Various parties have the ability to edit the database. Untrustworthy among entities and groups. Disintermediation has the advantage of allowing transactions to be independently reviewed and automatically authenticated, and transactions have the advantage of separating transactions between parties in order to increase the efficiency of the invoicing process.

Daniel et al. [18] investigated the implementation of smart contracts utilizing cutting-edge blockchain technology, which can serve as a component technology for a computing paradigm known as service-oriented computing (SOC) on the blockchain to encourage reuse and increase cost-effectiveness. It showed how the conceptual underpinnings of this new environment were more interconnected than one might assume, and how smart contracts, to some extent, may be read as basic components, that is, services, of a blockchain-based, SOC paradigm.

Terzi et al. [19] described an application of BC technology in two real-world supply chain scenarios. The first was designed to log and trace items by employing smart contracts to identify the constituents of food products and how to uniquely identify the food product throughout their transportation from the plant to the client who purchases it. The transparency of the procedure and transportation verification are critical components of this method. The second explanation is meant to address the authentication process for users having BC IDs. A user's identity is critical for network security, enabling only authorized parties to access and process data.

Sookhak et al.[20] investigated the implementation of smart contracts using cutting-edge blockchain technology, which can serve as a component technology for a computing paradigm known as service-oriented computing (SOC) in the blockchain to promote reuse and increase cost-effectiveness and describes how the conceptual underpinnings of this new landscape are more integrated than one might expect, and how smart contracts, to some extent, can be interpreted as basic components, i.e. services, of a blockchain-based, service-oriented computing paradigm. The researchers focused on studying ways to ensure the security of patients' information and privacy when Electronic Health Records (EHRs) are used in healthcare settings, particularly during data breaches. The introduction of blockchain technology and smart contracts cleared the path for the creation of a reliable EHR access control system that supports secure client authentication, identification, and authorization. Also, researchers will look into cutting-edge blockchain-based access control approaches in healthcare. A thematic taxonomy regarding blockchain-based access control techniques is offered to emphasize the essential security needs for designing a granular access control approach and recognizing the security concerns of current approaches. They tested and investigated the contrasts and similarities of conventional access control techniques, as well as several significant and unresolved contemporary concerns as potential future directions.

For managing vaccine registration, distribution, and storage, Rotbi et al. [21] used a Blockchain-based system. They provided a compact system depending on blockchain technology and the Internet of Things (IoT) to service the Covid-19 vaccination. This study proposed a secure smart contract that depends on Blockchain with decentralized management to prevent a non-repudiation scheme. The present work aims at providing a secure model with a more confident decentralized contract in addition to providing more effective smart contract protection through several security requirements, for instance, confidentiality, authorization, accountability, integrity, and authenticity.

Research Methodology:

The Proposed System Infrastructure

The system infrastructure is divided into three parts, the client, the server, and the database as a system that is supposedly more than one client and more than one server. The client part includes many interfaces and for server contains events. The requirement of the system is consisting of:

1. Server (nodes): that contains two types of servers, the first one for major processing. While the second server is used for saving the database.
2. Client: needs to interface, may be web application or client application.
3. Database: connects with the server directly and must be safe.

Thus, the requirements of the user to establish the connection can be listed and illustrated in the following Fig 1.

A. Bank account.

B. Real estate account.

C. Key pair (public key and private key) used for the sign.

D. Wallet with the username and password for storing all the required information in addition to:

1) Approved Contract.
2) Pending Contract.

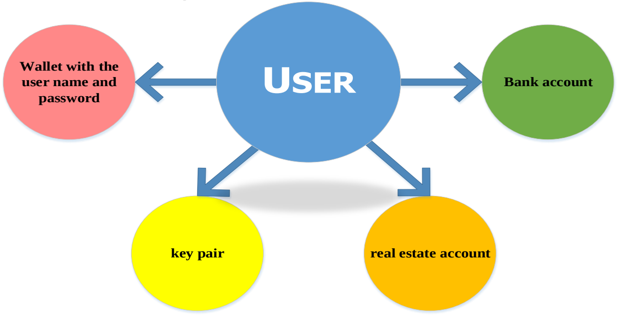


Figure 1. The requirements of the user.

The system consists of two parts client application and server application as shown in Fig 2. The client application allows user to apply the following function:

a) Logging into Wallet.

b) Register in the Wallet.

c) A new contract can sign according to the following contain:

- Seller name, public key, real estate, and signature.
- Buyer name, public key, price to pay, and signature.

d) View old contract

e) View pending contract

f) Check bank Balance

g) Check real estate. However, the client application can apply the following functions:

- o Has the user bank account?
- o Has lodger?
- o Register.
- o Create a contract to sell.
- o Login.
- o Pending contract.

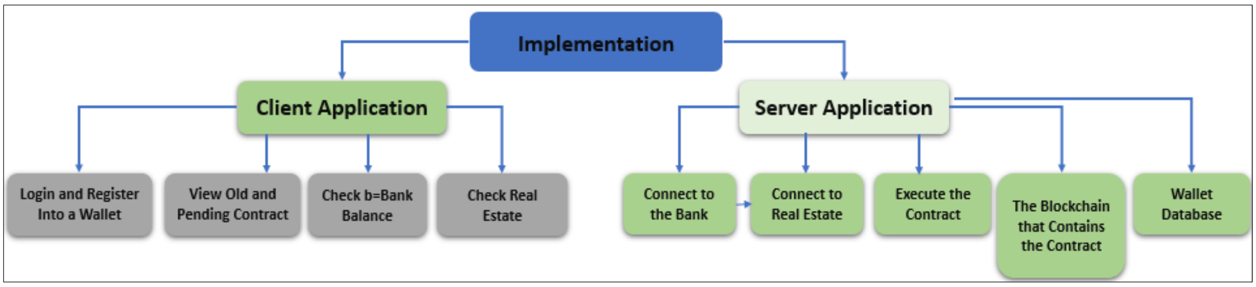


Figure 2. The implementation of client and server applications.

The server application is applied on the trusted party, and contains the following:

1) Wallet database.
2) The blockchain that contains the contract.
3) Connection to the bank.
4) Connection to real estate.
5) Execution the contract. The server application can apply the following functions:

a. Receiving the contract, in case the contract is not signed, then sent it to the parties to sign.

b. Contract signiture

c. Verifying the signatures

d. Verifying information (bank, ledger)

e. Transfering

f. Adding bank ID

g. Signing with a trusted key.

h. Saving in the chain.

Therefore, the server application connects with a set of databases as seen in Fig 3.

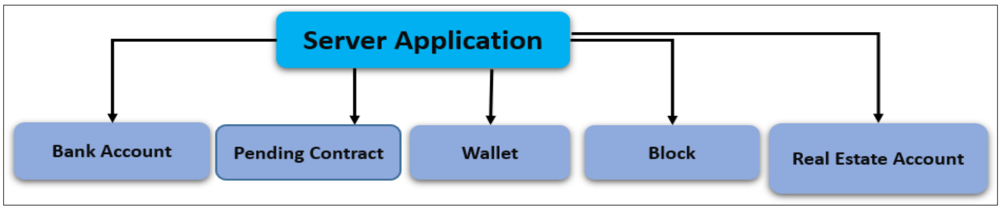


Figure 3. Connections server applications.

1- Bank account

2- Real estate account.

3- Block

4- Wallet

5- Pending contract. However, each of the block databases. Wallet, and pending contracts have to be placed in the trusted party.

The contract management stage takes place after creating a smart contract by both parties filling in the contract’s information for both parties and adding the required signatures from each party all information will be included in the signature. Suppose a person wants to sell a property, the first step requires building a contract and signing the information, adding the required price, the property offered for sale, and the other party is the buyer.

Then the contract is signed by the seller. The stage of completing the contract by the seller is complete. Now comes the role of the buyer. Then the contract is received as stated in Fig 4 where the system of smart contracts that sent to the server.

The most important step is to verify the information included in the contract verifying whether the information is intact or has been manipulated by the other party. After making sure that the information is correct and adding the signature of the seller and the buyer, the next step is to verify the information by the server. The role of the server at this stage is to verify the signatures of the two parties, the property information, and the bank account of both parties. After the verification is done, the transaction is considered completed. The information is saved and added to the Previous Block ID after which it is saved to the Blockchain network.

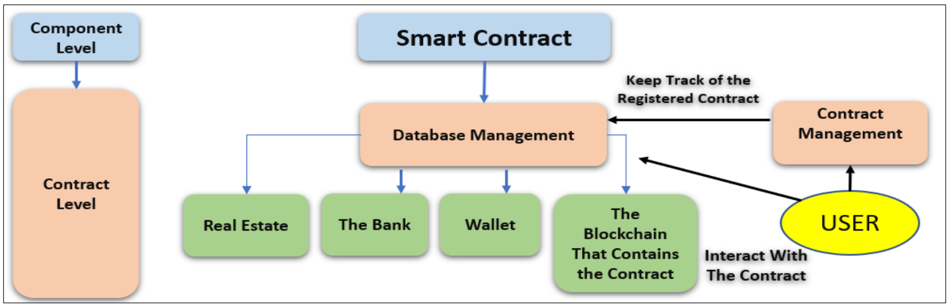


Figure 4. The smart contracts management steps.

The Proposed Prototype System Overview

The proposed system was designed and implemented with a set of compact programs that include virtual real estate and virtual banks. This means that non-repudiation has been required on the SSL layer's top, which is often utilized for securing communications between seller and buyer as well as the server. Entity authentication, confidentiality, and data authentication are all provided by SSL. Transactions must, on the other hand, be time-bound to the parties involved in order to be legally valid — non-repudiation is required. It is impossible to implement this on a layer underneath the application layer, like the SSL layer since involved parties must beware of legal binding and procedures of non-repudiation. As a result, it must be presented in the application layer itself. Hence, the main goal was to design and implement a program that simulates real estate and banks by using blockchain technology.

The Construction of the Smart Contracts with Blockchain

The proposed system will discuss a virtual prototype scenario using a real-estate department between a buyer and seller as stated in Fig 5. Where the proposed smart contract gives us the following more distinguished smart compared with traditional systems:

- The proposed system adopts a distribution scheme that allows the data to be dispersed over several nodes, allowing any fraudulent, forged, or corrupted records to be recognized and swiftly rejected by comparing the data across nodes.

- The proposed system would reduce reliance on external parties for the purposes of information verification and authentication requirements, and it will make asset exchange as easy as possible.

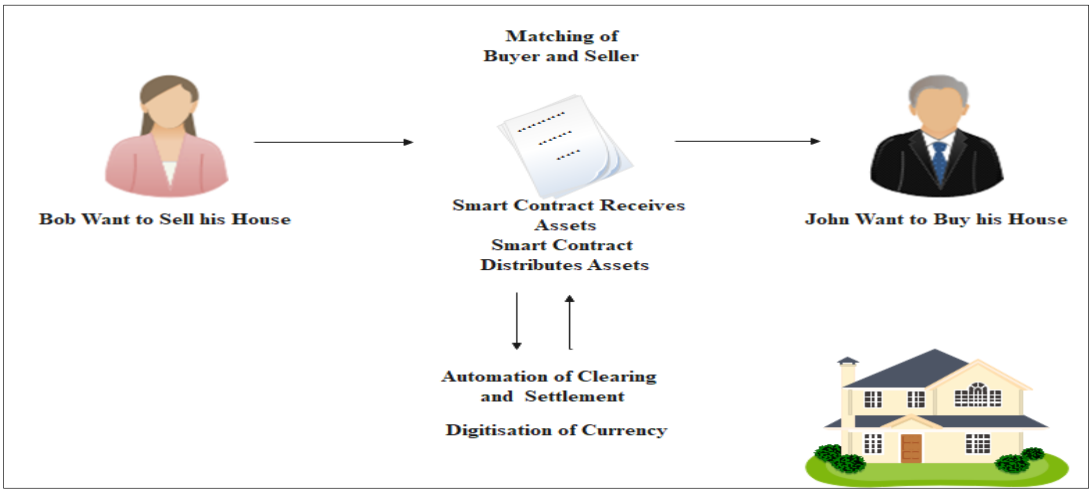

Matching of Buyer and Seller
Bob Want to Sell his House
Smart Contract Receives Assets
Smart Contract Distributes Assets
John Want to Buy his House
Automation of Clearing and Settlement
Digitisation of Currency

Figure 5. Virtual scenario between buyer and seller [22].

Through the interaction of the smart contract with other smart contracts or accounts during a transaction, a property can be sold or transferred. In this approach, building management might be automated and carried out more quickly, saving cost and time[22].

As shown in Fig 6 the proposed system has four primary levels: software system level, container level, component level, and contract level. Nodes are constructed at a variety of trustworthy sites, such as government buildings, banks, or huge corporations. A trusted node is assigned to a party that directly or indirectly benefits from maintaining safe and speedy transactions, such as banks and real estate businesses.

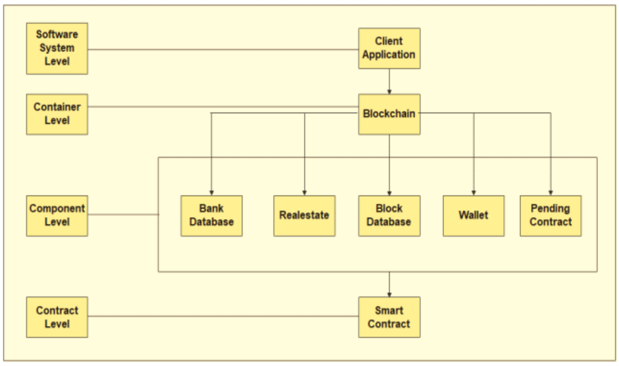


Figure 6. Overview of the proposed system.

Because they cooperate to make the system safe, all nodes will suffer and be affected if a record is damaged or fabricated. To create and sign contracts, users must communicate with nodes; everything else is handled by the node, which verifies the information provided and takes other necessary actions to ensure the accuracy of the information. In order to expedite the exchange of assets, banks and real estate offices connect with nodes.

The Proposed System Scheme

The proposed system consists of four main correlated parts that show the architecture of the proposed system with its main ingredients: Bank Application Programming Interface (API); Real-estate Ledger; Server; and Client. The system scenario consists of several sides and parties that can be illustrated in the following phases:

The following description involves several phases for the construction scenario of smart contract transactions from client to server and their permissions:

1. User Side Permissions: This aspect is concerned about the login information of the user to the system, as well as the execution regarding a few actions or processes linked to entity operations:
   - The user can create a bank account to deposit money and ensure that the amount is preserved and can be used for different types of online payments.
   - The user can create a real estate account with his money. The real estate account can give the seller a gateway entry to the platform to sell any real estate. At the same time, it gives the buyer the opportunity to browse thousands of classified ads daily and discover the best properties offered.
   - The user can get a pair of keys generated using his own content algorithms. The first is public and is accessible to everyone, and the second is a secret that is only for the user.
   - The wallet consists of (a username and password). The user must have a wallet that includes the user's name and password. The benefit of the wallet lies in the storage facilities and gives the user convenience as well as ease of payment.

2. Client API-Side Permissions: The user will register to a new wallet or log in to an existing wallet through a user interface that is called Client API, with the following steps.
   - At first login to a wallet which allows the user to access the wallet by entering the username and password
   - Then register to a new wallet the login stage includes entering the user's name, real estate account, and bank account so that the user can enter the system.
   - A new contract is generated for the user and takes place after the registration process in the system.
   - View old contracts where the user can view the list of old contracts by date and time.
   - View pending contracts, where the user can view the contracts that are not signed by others and that are in pending status.
   - Check and verify the bank account where the user can check the balance in the bank account of the other party.
   - Check real estate the user can verify the property in the real estate concerning the other party.
3. Server API-Side Permissions: This aspect relates to the contact information for the bank account and real estate. It is achieved with multiple tasks and performs the following operations or steps:
   - Containing wallet database: it consists of the wallet database, which in turn includes the user's name, password, public key, and private key, in addition to contract (Block ID) and pending actions.
   - Containing Blockchain (block) which includes the contract after it has been signed and verified.
   - Connect to the bank: server application can contact the Bank and check all the information received in the form of requests for verification requirements.
   - Connect to real estate where the server application can connect to Real-estate and check all the information received in the form of requests for verification purposes.
   - Contract Execution: the application can also implement the contract after making sure that all information is correct.
4. Block Database Side: This part includes information, such as the user's ID and contents, as well as signatures, and the following steps explain more:
   - User ID of each user who enters the system with a unique identifier.
   - Detail Content: Including the contents of the user contract.
   - Signature (content signed with trusted key) includes the procedures that are verified when the transaction is completed between the two parties to be signed and later verified by the server.
5. Wallet Database Side: This part is specific to each user who enters the system and some details are included in this side, such that:
   - Username and Password: each user must have a unique username with a secret password to maintain the user's account.
   - Key pair each user must have a key pair (public/private).
   - Contract (block ID): the id of each block through which is connected to the chain.
   - Pending Status: It includes contracts that are temporary or not approved and still in pending status.
6. Real-estate Ledger Side: This side contains user information about Real-estate and includes some sensitive information:
   - Main Database.
   - ID, Name, and Owner.
   - ID user (Owner).

7. Bank API Side: This side contains user information about Bank and includes some other details:
   - Bank Database.
   - Consist of the ID of each user registered in the system.
   - Account name: it involves the user account name to log in to the system.
   - Balance the amount of the user's amount that is added to the bank account to complete the transaction between the two parties.

The Design Paradigms of User API

The user design style involves registration steps which also known as the wallet procedure that can be seen in Fig 7 which explains how the user registers to the system:

a) The user chooses an ID and password to enable him to enter the system at any time, and the nickname must be unique that can be recognized by network subscribers.
b) Filling in the bank account information that is required by the system when entering any user into the user interface.

c) Filling real estate account information is required also by the system when entering any user.

d) Creating public or private key pairs utilizing Elliptic Curve cryptographic protocols. The key generation process took the responsibility of the server.

e) Encrypts information with Advanced Encryption Standard (AES-256) to encrypt data and provide a high security and protection level against brute force and other attacks.

f) Sending information to be saved in Blockchain nodes with hashed data, so that the user might access the file hashes through smart contacts after worker node authentication.

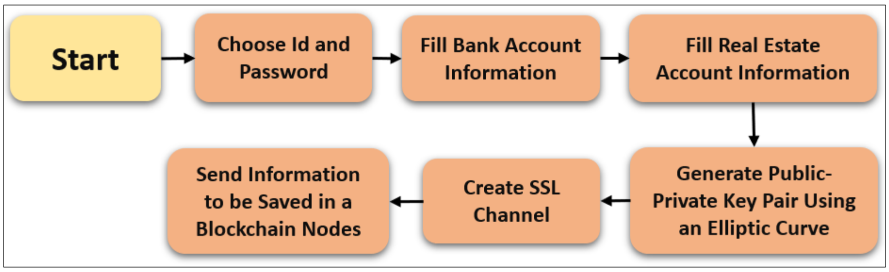

Start
Choose Id and Password
Fill Bank Account Information
Fill Real Estate Account Information
Generate Public-Private Key Pair Using an Elliptic Curve
Create SSL Channel
Send Information to be Saved in a Blockchain Nodes

Figure 7. Diagram of user registration steps

Advanced Encryption Standard

Scrambled data using the intriguing technique of encryption, making it unintelligible to unauthorized parties. A specification for the encryption of electronic data is called the Advanced Encryption Standard (AES), sometimes known as Rijndael (its original name). The practice of encrypting communications or important information so that only authorized parties may read, it is known as encryption. Although encryption does not stop interceptions by itself, it does prevent the interceptor from getting the information.

Simply put, encryption is the process of creating encrypted text that can only be decoded by a certain person. Advanced Encryption Standard is one such encryption method used to safeguard online data from AES. In most cases, symmetric key encryption techniques or public key encryption schemes are used for encryption [23].

The Design Paradigms of Client API

In this section, starting the process of client connection with the server using TCP connection protocol. Where the contract creation process will start after the process of creating a contract. Then adding the contents of the contract which involve the information of the seller and buyer. After that, the signature is verified between the buyer and the seller and then the details will be sent to the server for checking and verification. The process of verifying, signing, and saving them in the Blockchain will occur immediately.

The following steps will prepare the information to be added to the contract by the seller and buyer as shown in Fig 8 that represents the contract creation of the seller and buyer:

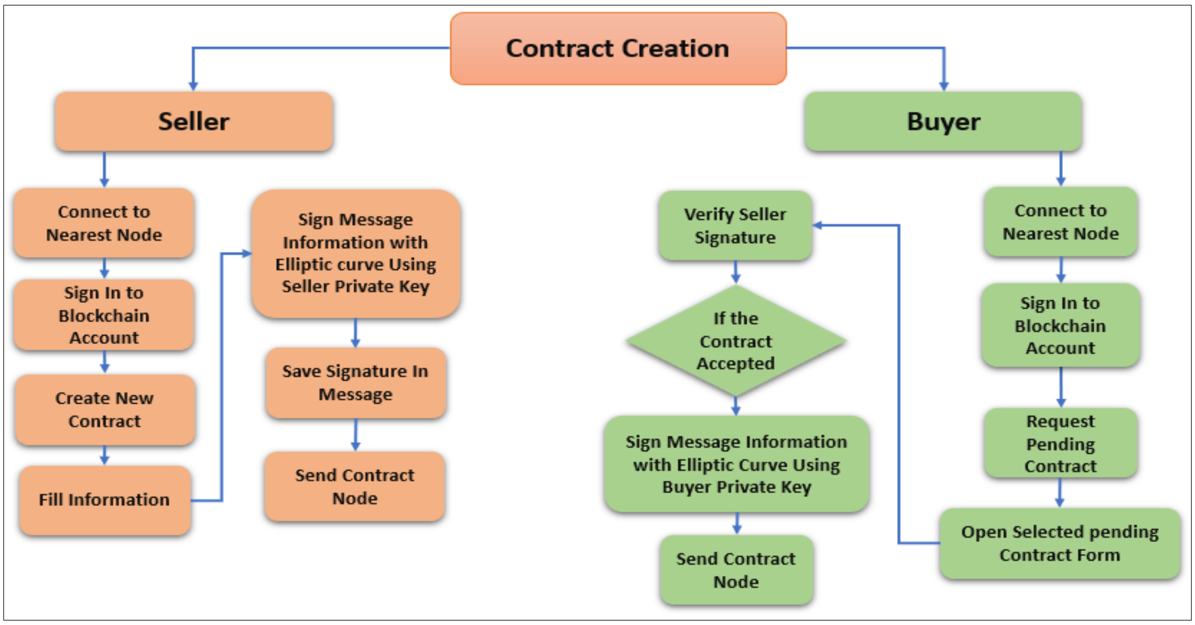

Contract Creation
Seller
Connect to Nearest Node
Sign In to Blockchain Account
Create New Contract
Fill Information
Sign Message Information with Elliptic curve Using Seller Private Key
Save Signature In Message
Send Contract Node
Buyer
Connect to Nearest Node
Sign In to Blockchain Account
Request Pending Contract
Open Selected pending Contract Form
Verify Seller Signature
If the Contract Accepted
Sign Message Information with Elliptic Curve Using Buyer Private Key
Send Contract Node

Figure 8. Contract creation for the seller and buyer.

The Design Paradigms of Server API

The local server has been addressed that is responsible for receiving the queries and requests. After completing the contract information, the verification process will start between the seller and the buyer to ensure that the information is not tampered with. The signing process is carried out by both parties and sent to the trusted node so that the verification process is done later on. During this process, the transaction data is signed and kept in the Block-chain as a signature creation and verification process as stated in Fig 9 below:

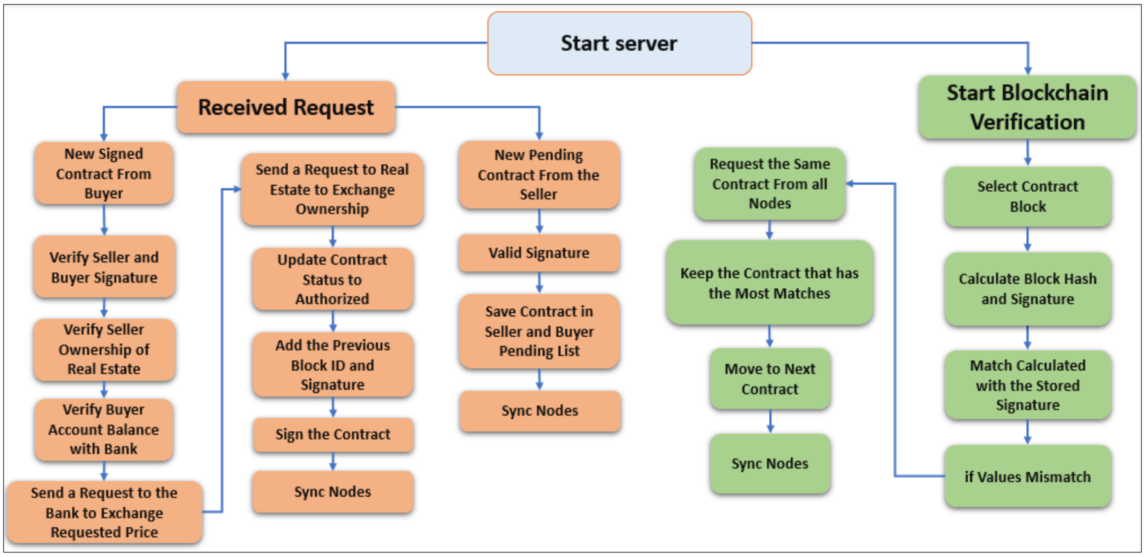

Start server
Received Request
New Signed Contract From Buyer
Verify Seller and Buyer Signature
Verify Seller Ownership of Real Estate
Verify Buyer Account Balance with Bank
Send a Request to the Bank to Exchange Requested Price
Send a Request to Real Estate to Exchange Ownership
Update Contract Status to Authorized
Add the Previous Block ID and Signature
Sign the Contract
Sync Nodes
New Pending Contract From the Seller
Valid Signature
Save Contract in Seller and Buyer Pending List
Sync Nodes
Start Blockchain Verification
Select Contract Block
Calculate Block Hash and Signature
Match Calculated with the Stored Signature
if Values Mismatch
Request the Same Contract From all Nodes
Keep the Contract that has the Most Matches
Move to Next Contract
Sync Nodes

Figure 9. Signature Creation and Verification.

The process of verifying the contents of the contract between the seller and buyer ensures the information has not been tampered with. Concerning a communications scenario, such as a contract, one of the users may subsequently deny the contents of the agreement or even that the communication occurred at all. Thus, the signing stage comes to prove the authentication for a specific transaction.

1. Contract Singing The seller or buyer signs a contract.
2. Contract Verification: To confirm the Seller's or Buyer's signature on the contract.
3. Contracts should be signed in the trusted node.
4. Timestamping is used to confirm that a contract was signed in a trusted node.
5. Contracting the Blockchain: The following connected steps must be completed in order to add a contract to the block-chain in the trusted node:
   I. Checking the contract's assets.
   II. Exchanging resources.
   III. Declaring the Contract to be authorized.

IV. Adding the contract's ID and signature from the previous block's blockchain record.
V. In the Trusted Node (_), sign a contract.
VI. Including the contract in the blockchain.
VII. The blockchain is synced with other nodes.

6. Trusted Node Block-Chain Verification: The process of checking the contract is shown in Fig 10.

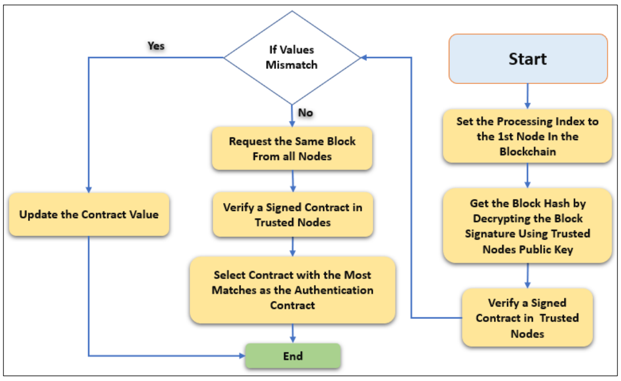

Start
Set the Processing Index to the 1st Node In the Blockchain
Get the Block Hash by Decrypting the Block Signature Using Trusted Nodes Public Key
Verify a Signed Contract in Trusted Nodes
If Values Mismatch
Yes
No
Update the Contract Value
Request the Same Block From all Nodes
Verify a Signed Contract in Trusted Nodes
Select Contract with the Most Matches as the Authentication Contract
End

Figure 10. Flowchart of blockchain verification in trusted node.

Elliptic Curve

Similar to RSA the ECC is a type of public-key cryptosystem. But it differs from RSA in that it can evolve more quickly and offers researchers working on cryptographic algorithms an appealing and different approach. Even with smaller ECC keys, the level of security offered by RSA can be achieved. As an example, the 163-bit security strength of an ECC. But could provide the 1024-bit security strength of an RSA. Other than this, ECC is particularly well suited for wireless communications, such as mobile phones and smart cards [24].

Elliptic Curve Cryptography is a relatively new type of cryptography that Koblitz independently proposed in the latter half of the 19th century. Due to the same level of security, they offer with significantly smaller key sizes than traditional public-key cryptosystems, it has since attracted a lot of attention and grown immensely popular. The ECC includes key exchange, the Elliptic Curve Digital Signature Algorithm(ECDSA), and all pertinent asymmetric cryptographic primitives.

Elliptic Curve Cryptography is a relatively new type of cryptography that Koblitz et al. [25] independently proposed in the latter half of the 19th century. Due to the same level of security they offer with significantly smaller key sizes than traditional public-key cryptosystems, it has since attracted a lot of attention and grown immensely popular [26, 27]. The ECC includes key exchange, the Elliptic Curve Digital Signature Algorithm(ECDSA), and all pertinent asymmetric cryptographic primitives [28, 29].

Elliptic Curve Digital Signature Algorithm

The EC Digital Signature Algorithm is the elliptic curve equivalent of the DSA; this protocol requires not only elliptic curve operations, such as integer multiplication, inverse operation, and modular operation, but also scalar multiplication, field multiplication, and field inverse operation. The EC Digital Signature Algorithm is the elliptic curve equivalent of the DSA; this protocol requires not only elliptic curve operations, such as integer multiplication, inverse operation, and modular operation, but also scalar multiplication, field multiplication, and field inverse operation [30].

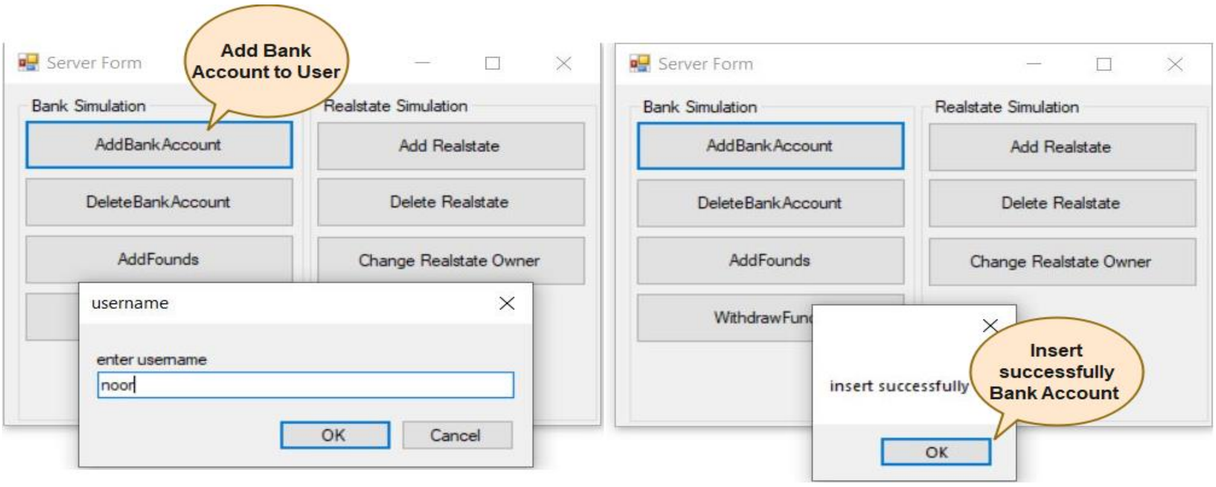
Results and Discussion

The main aim of any smart contract design must be performance, security, implementation costs, and other factors. The performance test approach determines the time required to perform implementation smart contracts operations. The design of a smart contract depends on several programming languages and several important libraries with different environments. Several tests were implemented on many platforms and in several languages in addition to the different hardware. This chapter describes the implementation and testing for a real estate transaction using smart contracts and blockchain technology. The proposed system utilized a smart contract which is utilized on a blockchain that stores a description of a rent item or a purchase and regulates transactions between the two entities. And how the smart contract code executed on the blockchain is integrated into the C# language and the implementation of the actual components. The simulation software package for the proposed models is implemented in C# and the complete environment was built in Visual Studio 2017. The ECC algorithm is selected because it gave the best results in implementation compared with RSA. Furthermore, a new additional layer of the secure web via SSL has been dedicated.

The Proposed Graphical User Interface

To implement the real estate transaction, the user has to obey the following steps: registration process, create a Real-estate account, and create a bank account initially. The Graphical User Interface (GUIs) of the prototype can be explained with some of the captured snapshots below:

If the user logs in to the system, he will need to create add bank account and real estate account as explained in Fig 11 and Fig 12.



Add Bank Account to User

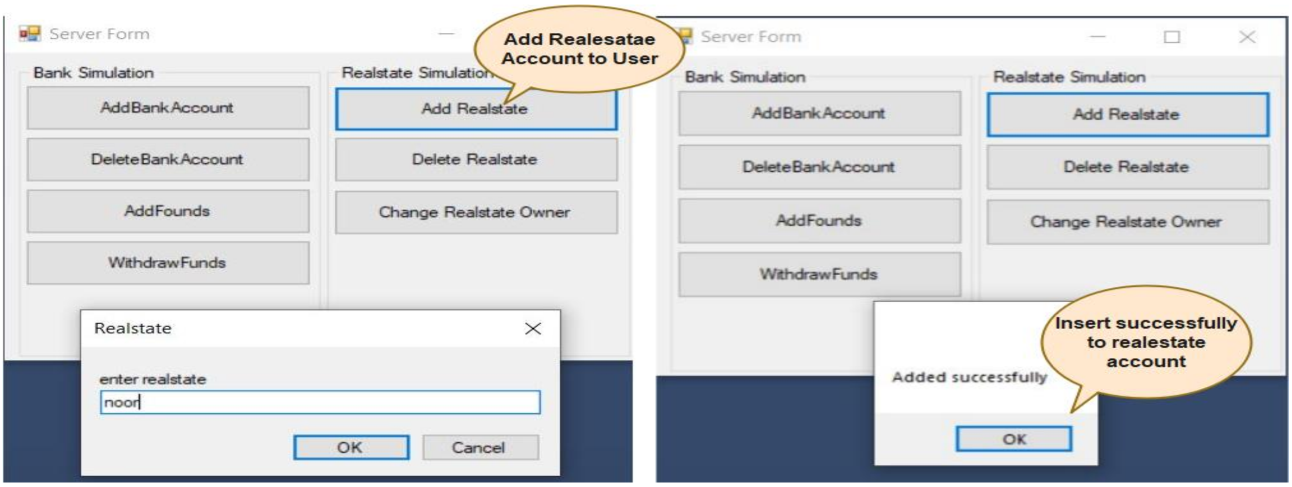
username

enter username: noor

insert successfully

Insert successfully Bank Account

Figure 11. Create a bank account.



Add Realesatae Account to User

Realstate

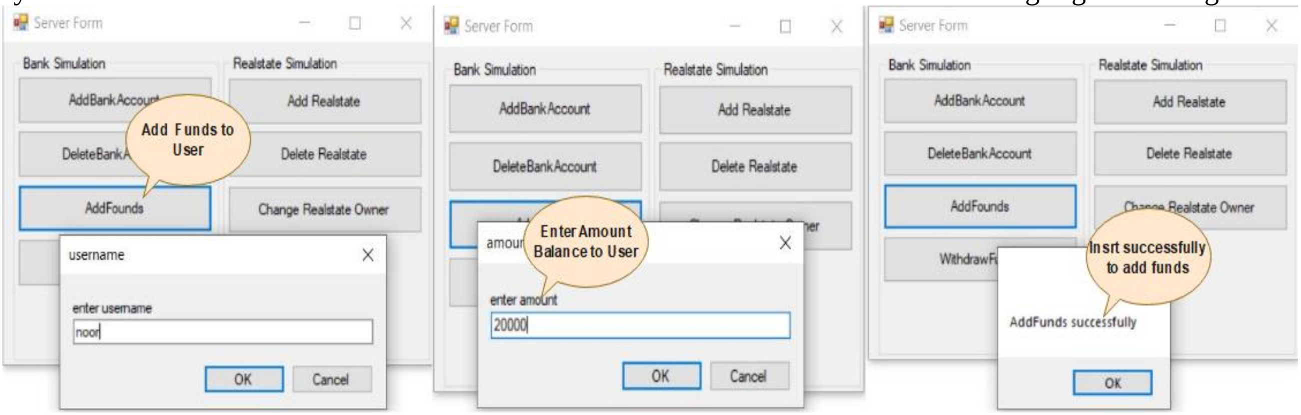
enter realstate: noor

Added successfully

Insert successfully to realestate account

Figure 12. Real-estate account.

1. Adding funds to the user and the amount of balance.
2. This step represents the user registration for the system and creates the contract.
3. The next step after the registration process involves adding the bank account and the real estate account, the contract is created as shown in the following Fig 13 and Fig 14.



Add Funds to User

username

enter username: noor

Enter Amount Balance to User

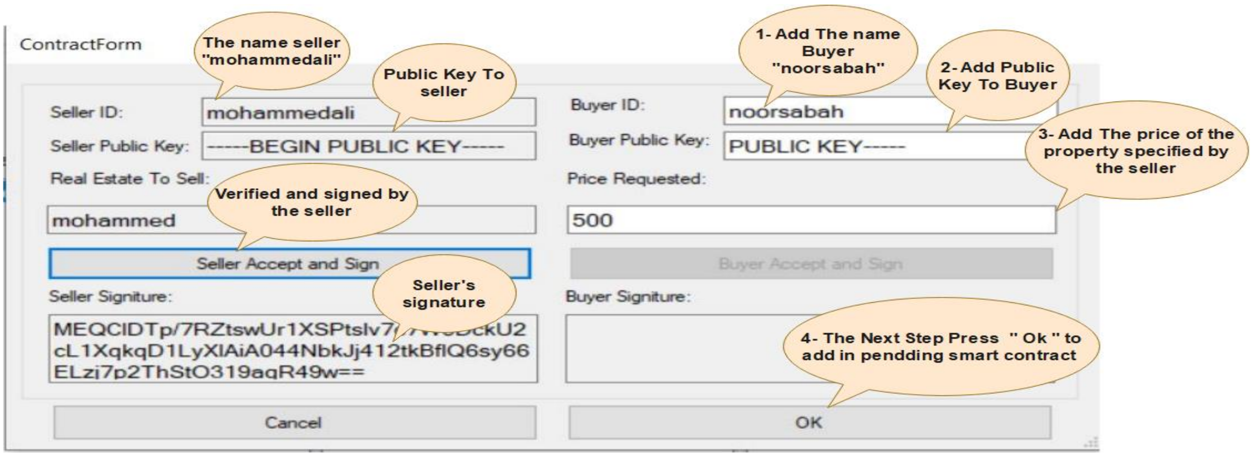
amount

enter amount: 20000

AddFunds successfully

Insrt successfully to add funds

Figure 13. Illustrates the add funds and amount balance process.



ContractForm

Seller ID: mohammedali

Seller Public Key: -----BEGIN PUBLIC KEY-----

Real Estate To Sell: mohammed

Seller Signiture: MEQCIDTp/7RZtswUr1XSPtslv7/[illegible]ckU2 cL1XqkqD1LyXIAiA044NbkJj412tkBflQ6sy66 ELzi7p2ThStO319aqR49w==

Buyer ID: noorsabah

Buyer Public Key: PUBLIC KEY-----

Price Requested: 500

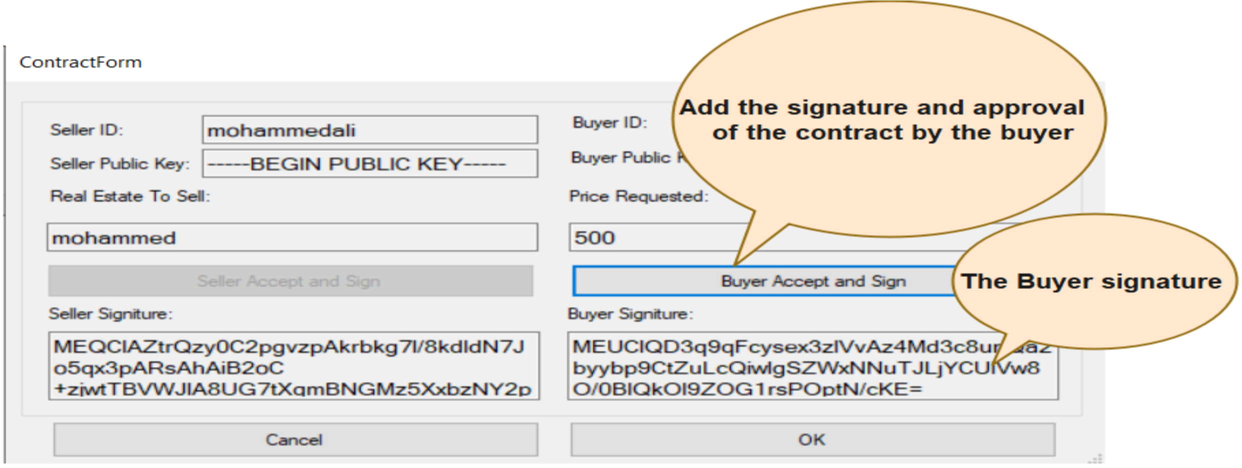
Buyer Signiture:

Seller Accept and Sign
Buyer Accept and Sign
Cancel
OK
The name seller "mohammedali"
Public Key To seller
Verified and signed by the seller
Seller's signature
1- Add The name Buyer "noorsabah"
2- Add Public Key To Buyer
3- Add The price of the property specified by the seller
4- The Next Step Press " Ok " to add in pendding smart contract

Figure 5. The process of adding seller and buyer information to the contract.

The next step is the forming of the final contract with the information of the seller and buyer and the price required to sell the property and adding the seller's signature to the information and adding in the pending smart contract. The incoming step is the process of defining the contract by the buyer, by double clicking to start the process of adding the buyer's signature and agreeing to the information sent by the seller.

The last step involves adding the signature and approval by the buyer as shown in the following Fig 15.



ContractForm

Seller ID: mohammedali

Seller Public Key: -----BEGIN PUBLIC KEY-----

Real Estate To Sell: mohammed

Seller Signiture: MEQCIAZtrQzy0C2pgvzpAkrbkg7I/8kdIdN7J o5qx3pARsAhAiB2oC +zjwtTBVWJIA8UG7tXqmBNGMz5XxbzNY2p

Buyer ID:

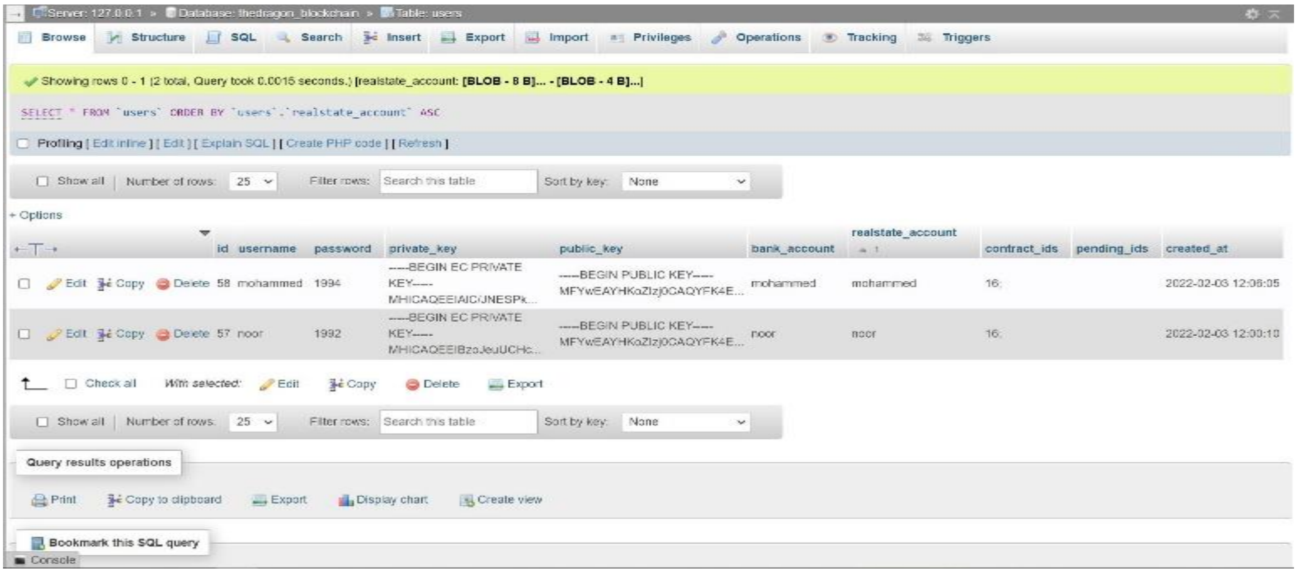
Buyer Public

Price Requested: 500

Buyer Signiture: MEUCIQD3q9qFcysex3zlVvAz4Md3c8ur[illegible]az byybp9CtZuLcQiwlgSZWxNNuTJLjYCUlVw8 O/0BIQkOI9ZOG1rsPOptN/cKE=

Figure 6. Adding the signature and approved by the buyer.

The next step includes transferring the property from the first person to the second person and transferring the balance from the first person to the second person and saving in blockchain and taking a number (16) in the list of approved contracts. Another step shows the generation of the keys for the seller and the buyer, which are public, private, time, and date of creating contracts and storing them in (Database). Finally, combining the signature of the seller and buyer into one string where as it calculated the seller and buyer signatures and combined the information into one string as shown in Fig 16.



✓ Showing rows 0 - 1 (2 total, Query took 0.0015 seconds.) [realstate_account: **[BLOB - 8 B]... - [BLOB - 4 B]...**]

```
SELECT * FROM `users` ORDER BY `users`.`realstate_account` ASC
```

	id	username	password	private_key	public_key	bank_account	realstate_account	contract_ids	pending_ids	created_at
□ Edit Copy Delete	58	mohammed	1994	-----BEGIN EC PRIVATE KEY----- MHICAQEEIAIC/JNESPk...	-----BEGIN PUBLIC KEY----- MFYwEAYHKoZIzj0CAQYFK4E...	mohammed	mohammed	16;		2022-02-03 12:06:05
□ Edit Copy Delete	57	noor	1992	-----BEGIN EC PRIVATE KEY----- MHICAQEEIBzoJeuUCHc...	-----BEGIN PUBLIC KEY----- MFYwEAYHKoZIzj0CAQYFK4E...	noor	noor	16;		2022-02-03 12:00:10

Figure 7. Date and time of the creation of the contracts for the seller and buyer.

Confidentiality is a key aspect of security, and it ensures that sensitive information is protected from unauthorized access. In the blockchain, confidentiality is achieved through the use of encryption. Data is encrypted before it is added to the blockchain, and only authorized parties with the proper decryption keys can access the data. This ensures that even if the data is intercepted by an unauthorized party, they will not be able to read it.

While authorization is another important aspect of security, as it ensures that only authorized parties are able to access and modify data. In the blockchain, authorization is achieved through the use of digital signatures. Digital signatures are a cryptographic technique that is used to confirm the identity of the parties involved in a transaction. Each party involved in a transaction will have a unique digital signature, and these signatures are used to confirm their identity. This ensures that only authorized parties are able to access and modify data on the blockchain. Where the transaction is signed by the persons present in the contract and the signature is made using the high-security (ECC) algorithm, which in turn is difficult and complex.

Data integrity refers to the reliability and trustworthiness of data. It involves the maintenance of, and the assurance of the accuracy and consistency of, data over its entire life cycle. Blockchain just might be the solution to improve data integrity to the highest standards .Blockchain ledgers are immutable meaning that data addition or transaction cannot be edited or deleted. One aspect of blockchain technology which particularly important for improved data integrity is the Merkle tree, it ensures the integrity of data in the blockchain. Merkle tree is a fundamental component of blockchain that uses a cryptographic hash function. Every block stores transaction data in the form of a tree, hashes of child nodes are combined into the parent nodes header and this technique continues iteratively until a final, or root, node is reached, this node will contain all the information contract, so it is used hash and Merkel tree in verifying the integrity of the data in the blockchain.

Non-repudiation is the concept that parties involved in a transaction cannot deny their involvement in it. In the blockchain, non-repudiation is achieved through the use of digital signatures. Digital signatures are used to confirm the identity of the parties involved in a transaction and ensure that they cannot deny their involvement. This is because digital signatures are unique to each party, and they cannot be forged or duplicated. This ensures that parties cannot deny their involvement in a transaction, even if they try to.

All these elements work together to provide a secure and tamper-proof environment for conducting transactions and storing data on the blockchain. The use of encryption, digital signatures, cryptographic hash functions, and non-repudiation ensures that data stored on the blockchain is protected from unauthorized access, tampering, and criminal involvement.

Analysis and Evaluation

This study, evaluates the prototype implementation of the present proposal and provides short security and cost analysis of the suggested blockchain-based solution to provide proof of the authenticity of the digital content.

Security Analysis

This section focus on security analysis that comprises the cryptographic algorithm and protocols like SSL encryption, RSA, and ECC.

While buying an SSL certificate, you should have a clear understanding of both of these terms. It may be utilized for the creation of smaller, more efficient, and much faster cryptography keys. Instead of using the traditional method to generate a product of very large prime numbers, it uses an elliptic curve equation to generate keys. ECC is used in the well-known cryptocurrency (i.e. Bitcoin etc). For hackers, it is really hard to crack the ECC algorithm that operates upon the Elliptic Curve Discrete Logarithm Problem (ECDLP). The ECC certificate has often smaller size because the information that is needed for exchanging for validation is less. For organizations having long-term security solutions as a primary concern, ECC may be an ideal choice.

Also, hybrid SSLs can be utilized to use ECC instead of RSA-trusted root keys. A simple comparison can be shown in terms of the length of ciphering key between the RSA and the ECC can be stated in Table 1 according to the NIST (i.e. the National Institute of Standards and Technology) [31].

Table 1. Comparison between RSA and ECC key strength[32].

No.	RSA Cipher	ECC Cipher
1	1024	160
2	2048	224
3	3072	256
4	7680	384
5	15360	521

Here is a quick comparison between RSA and ECC according to the NIST publication. It will help us to decide which is better in terms of security. From the above Table 1, this research concludes that the ratio of a key in size and its strength for the ECC compared with RSA is as follows: No.1 (1:7), No.2 (1:10), No.3 (1:12), No.4 (1:20), No.5 (1:30).

Cost Analysis

This section involves the evaluation and analysis of cost in terms of the creation of contracts and interaction activities. It is important to note that the transaction sender in the blockchain should avoid paying gas fees in reverse contracts using Etherume. However, concerning transaction cost economics, the proposed solution has been argued that, though smart contracts reduce transaction costs related to contracting parties as shown in Table 2.

This section includes a discussion of the costs of deployment and interaction that are related to the purchase and sales contracts that have been explained in the section Analysis and Evaluation.

Table 2. Transaction costs in smart contracts.

Activities	Transaction Costs Decreasing	Transaction Costs Increasing
1. Because the blockchain prevents any deviation from what contracting parties have agreed upon ex ante 1. Because the blockchain prevents any deviation from what contracting parties have agreed upon ex ante 1. Because the blockchain prevents any deviation from what contracting parties have agreed upon ex-ante 1. Because the blockchain prevents any deviation from what contracting parties have agreed upon ex-ante	Blockchain avoids opportunistic renegotiations of smart contracts	Transaction costs due to lack of legal adaptation: blockchain also prevents efficiency-enhancing
2. Because the consensus mechanism provides the blockchain with a degree of flexibility	Users can adapt the blockchain and smart contracts to new needs	Transaction costs consensus mechanism may lead to an uncertain, majority-driven

From the standpoint of transaction cost economics, though smart contracts reduce transaction costs related to contracting parties' hold-ups, they can also create or increase other costs related to it. As a result, the execution time for transaction validity according to the Proof-of-stake consensus can be calculated as a function of elapsed time as shown below:

Function of Elapsed Time

```
DateTime= 0;
ElipsedTime = 0;
EndTime = 0;
DateTime t1 = DateTime.Now;
{
Encryption Process ();
}
EndTime t2 = DateTime.Now;
Elapsed Time t = t2 - t1;
```

A simple comparison of the cost of key generation between the RSA and ECC can be shown in Fig 17 and Fig 18 below respectively.

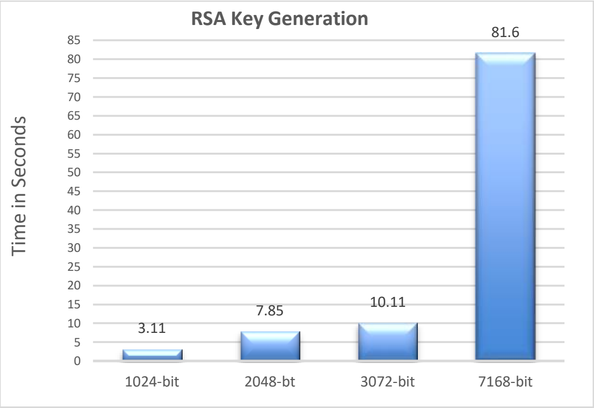


Figure 17. The time cost of RSA key generation in seconds.

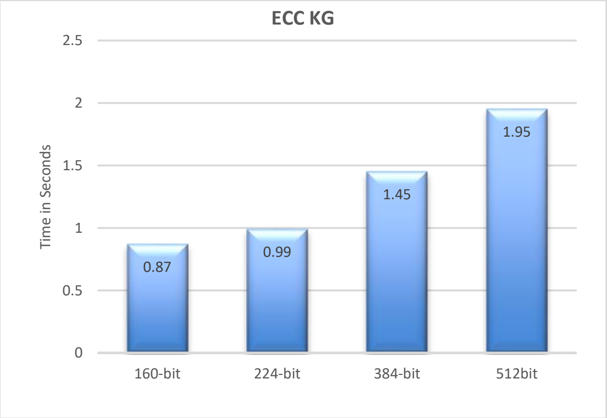


Figure 18. The time cost of ECC key generation in seconds.

Also, some empirical results for the time cost of digital signature in signing and verification processes are illustrated in Fig 19 and Fig 20 respectively.

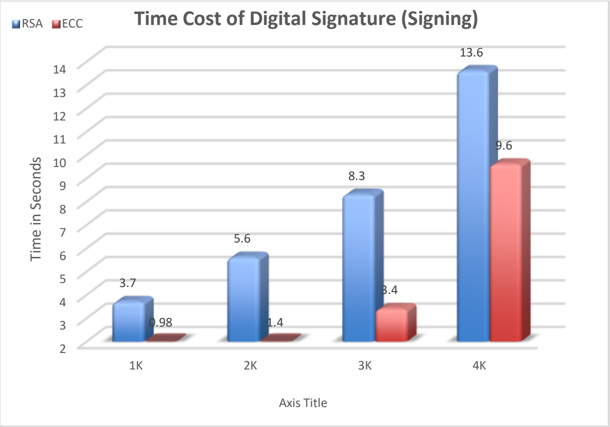


Figure 19. The time cost of digital signature (signing).

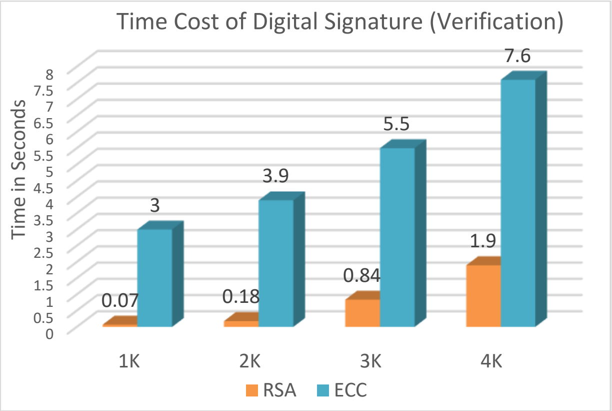


Figure 20. The time cost of digital signature (verification).

The Finding of the Study

It has been in regard to ECC, has three features that give the ECC the privilege over RSA, which are more security algorithms, smaller key size, and faster key generation processes. These features give ECC a higher level of security.

Privacy of the Transaction

This section presents a discussion of the most significant aspects of privacy, which include the security of smart contract accounts as well as transaction security. Accounts might be locally managed. The second approach should only be utilized for the purposes of testing, due to the fact that it includes unlocking the account and then sending the wallet password via a network in plaintext.

In the case of utilizing WalletAccountService, a local wallet can be decrypted with the use of the password that is given by the user for signing each one of the transactions with the secret key that belongs to the account. Utilizing WalletAccountService presents protection from eavesdropping due to the fact that the wallet password is never given to the client. In addition to that, it provides protection from the Man-in-the-Middle (MITM) attack, due to the fact that there is no ability for altering a transaction cannot once it was signed with a secret key.

There are only 2 cases left, where the attacker might get or utilize the secret key of the account:

- An attacker has root access to OS and is capable of intercepting the password from a user with the use of a key logger.
- An attacker gains physical access to a device and the account remains unlocked, where the attacker has the ability of using the account for signing the sale contract or purchasing the contracts and transmitting money to their own accounts.

The transaction backend needs to solve the blockchain computation for the chain of digests. Thus, regardless of the various computations of multiple hash functions, it can be deduced the average time for the several hashing as shown in Fig 21 below:

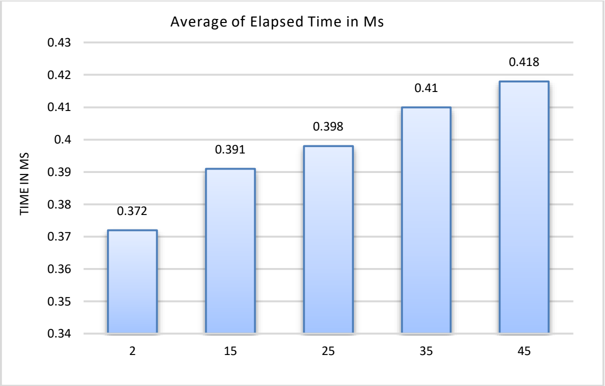


Figure 21. Time cost for the number of chained hash.

The privacy issues arising from the storing and exchanging of personal user data have been discussed in the application. It should be stated that potential issues related to privacy, may happen in the case of storing the details of the contract on blockchain and the way they may be evaded. The storing of the contract details in plaintext on the smart contract may lead to privacy problems, due to the fact that the seller and buyer addresses are stored as well on the contract, which is why they are accessible publicly. A party that is aware of an individual belonging to a certain address (for example, in the case where it exchanged the user or the contract data with that individual in the past) might look up the details of other contracts which have been signed by the individuals. For the purpose of preventing this, the option of light deployment may be utilized. In the case of utilizing the light deployment, only the hash of the details of the contract which don’t have to be stored on a blockchain is stored on smart contracts. From the result, the finding of this study was competitive with state-of-the-art studies.

Conclusion:

This study focused on achieving a trusted smart contract exploiting the features of Blockchain technology. A simulated scenario of a prototype system has been designed and implemented. The implemented scenario supposed the existence of Blockchain infrastructure that integrated with designed smart contracts effectively. The blockchain includes a set of blocks in which a block includes several transactions, in which the transaction can be described as a record or data that has been written to block-chain. Blockchains are replicated on all nodes in a network and every one of the nodes validates the content of the block, as a result, maintaining the consistency of the database by a mechanism of consensus. In this study, the investigation is directed towards using the blockchain and smart contracts in real estate property and concluded. The integration of the blockchain with smart contracts gives a decentralized system with less time implementation and with less expense. The smart contracts eliminated any necessity for intermediaries as well as contract enforcement. Which results in greatly reducing costs and simplifying the process of contract negotiation. The Blockchain technology provides a secure and tamper-proof environment for conducting transactions and storing data. The use of encryption, digital signatures, cryptographic hash functions, and non-repudiation ensures that data stored on the blockchain is protected from unauthorized access, tampering, and criminal involvement.

Acknowledgment

Special thanks are due to the Dean of the College of Computer Science and Information Technology and the staff.

Authors' Declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in University of Anbar.

Authors' Contributions Statement:

N.S.M, A.M.S, O.A.D, and A.A.N contributed to the implementation, design, and writing the research, to the analysis of the results.

References:

1. Zyskind G, Nathan O, Pentland A “Sandy.” Decentralizing Privacy: Using Blockchain to Protect Personal Data. In: 2015 IEEE Security and Privacy Workshops . IEEE; 2015. p. 180–4. https://homepage.divms.uiowa.edu/~ghosh/blockchain.pdf
2. Feng H, Wang X, Duan Y, Zhang J, Zhang X. Applying blockchain technology to improve agri-food traceability: A review of development methods, benefits and challenges. J Clean Prod. 2020; 260: 121031. https://doi.org/10.1016/j.jclepro.2020.121031

3. Clack CD, Bakshi VA, Braine L. Smart contract templates: foundations, design landscape and research directions. 2016. arXiv Prepr arXiv160800771. https://doi.org/10.48550/arXiv.1608.00771
4. Singh SK, Rathore S, Park JH. Blockiotintelligence: A blockchain-enabled intelligent IoT architecture with artificial intelligence. Futur Gener Comput Syst. 2020; 110: 721–43. https://doi.org/10.1016/j.future.2019.09.002
5. McCorry P, Shahandashti SF, Hao F. A smart contract for boardroom voting with maximum voter privacy. In: International conference on financial cryptography and data security. Springer; 2017. p. 357–75. https://doi.org/10.1007/978-3-319-70972-7_20
6. Hewa T, Ylianttila M, Liyanage M. Survey on blockchain based smart contracts: Applications, opportunities and challenges. J Netw Comput Appl. 2021; 177: 102857. https://doi.org/10.1016/j.jnca.2020.102857
7. Zheng Z, Xie S, Dai HN, Chen W, Chen X, Weng J, et al. An overview on smart contracts: Challenges, advances and platforms. Futur Gener Comput Syst. 2020; 105: 475–91. https://doi.org/10.1016/j.future.2019.12.019
8. Ibrahim R, Harby AA, Nashwan MS, Elhakeem A. Financial Contract Administration in Construction via Cryptocurrency Blockchain and Smart Contract: A Proof of Concept. Buildings. 2022; 12(8): 1072. https://doi.org/10.3390/buildings12081072
9. Balcerzak AP, Nica E, Rogalska E, Poliak M, Klieštik T, Sabie OM. Blockchain technology and smart contracts in decentralized governance systems. Adm Sci. 2022; 12(3): 96. https://doi.org/10.3390/admsci12030096
10. Hamledari H, Fischer M. Construction payment automation using blockchain-enabled smart contracts and robotic reality capture technologies. Autom Constr. 2021; 132: 103926. https://doi.org/10.1016/j.autcon.2021.103926
11. Omar IA, Jayaraman R, Debe MS, Salah K, Yaqoob I, Omar M. Automating procurement contracts in the healthcare supply chain using blockchain smart contracts. IEEE Access. 2021; 9: 37397–409. https://doi.org/10.1109/ACCESS.2021.3062471
12. Khatoon A. A blockchain-based smart contract system for healthcare management. Electronics. 2020; 9(1): 94. https://doi.org/10.3390/electronics9010094
13. Abdul-Ghani SA, Abdul-Wahhab RD, Abood EW. Securing Text Messages Using Graph Theory and Steganography. Baghdad Sci J. 2022; 19(1): 189-196. https://www.iasj.net/iasj/article/226470
14. Al-Hassani MD. A Novel Technique for Secure Data Cryptosystem Based on Chaotic Key Image Generation. Baghdad Sci J. 2022; 19(4): 905-913. https://doi.org/10.21123/bsj.2022.19.4.0905
15. Abdulameer SA, Kashmar AH, Shihab AI. A Cryptosystem for Database Security Based on TSFS Algorithm. Baghdad Sci J. 2020; 17(2): 567-574. https://doi.org//10.21123/bsj.2020.17.2.0567
16. Salim KG, Al-alak SMK, Jawad MJ. Improved image security in internet of thing (IoT) using multiple key AES. Baghdad Sci J. 2021; 18(2): 4–17. https://doi.org/10.21123/bsj.2021.18.2.0417
17. Karamitsos I, Papadaki M, Al Barghuthi NB. Design of the blockchain smart contract: A use case for real estate. J Inf Secur. 2018; 9(03): 177. https://doi.org/ 10.4236/jis.2018.93013
18. Daniel F, Guida L. A service-oriented perspective on blockchain smart contracts. IEEE Internet Comput. 2019; 23(1): 46–53. https://doi.org/10.1109/MIC.2018.2890624
19. Terzi S, Zacharaki A, Nizamis A, Votis K, Ioannidis D, Tzovaras D, et al. Transforming the supply-chain management and industry logistics with blockchain smart contracts. In: Proceedings of the 23rd Pan-Hellenic conference on informatics. 2019. p. 9–14. https://doi.org/10.1145/3368640.3368655
20. Sookhak M, Jabbarpour MR, Safa NS, Yu FR. Blockchain and smart contract for access control in healthcare: a survey, issues and challenges, and open issues. J Netw Comput Appl. 2021; 178: 102950. https://doi.org/10.1016/j.jnca.2020.102950
21. Rotbi MF, Motahhir S, Ghzizal A El. Blockchain technology for a safe and transparent covid-19 vaccination. 2021. arXiv Prepr arXiv210405428. https://doi.org/10.13052/jicts2245-800X.1022
22. Morena M, Truppi T, Pavesi AS, Cia G, Giannelli J, Tavoni M. Blockchain and real estate: Dopo di Noi project. Prop Manag. 2020; 38(2): 273-295. https://doi.org/10.1108/PM-01-2019-0005
23. Rawal S. Advanced encryption standard (AES) and it's working. Int Res J Eng Technol. 2016; 3(8): 1165–9. https://www.irjet.net/archives/V3/i8/IRJET-V3I8214.pdf
24. Yu H, Wang H. Elliptic curve threshold signature scheme for blockchain. J Inf Secur Appl. 2022; 70: 103345. https://doi.org/10.1016/j.jisa.2022.103345
25. Koblitz N. Elliptic curve cryptosystems. Math Comput. 1987; 48(177): 203–9. https://www.ams.org/journals/mcom/1987-48-177/S0025-5718-1987-0866109-5/S0025-5718-1987-0866109-5.pdf.
26. Cabrera-Gutiérrez AJ, Castillo E, Escobar-Molero A, Alvarez-Bermejo JA, Morales DP, Parrilla L. Integration of Hardware Security Modules and Permissioned Blockchain in Industrial IoT Networks. IEEE Access. 2022;10: 114331 - 114345. https://doi.org/10.1109/ACCESS.2022.3217815
27. Mäurer N, Gräupl T, Schmitt C, Rodosek GD, Reiser H. Advancing the Security of LDACS. IEEE Trans Netw Serv Manag. 2022: 1-15. https://elib.dlr.de/187783/1/TNSM3189736.pdf
28. Liu Z, Li Z. A blockchain-based framework of cross-border e-commerce supply chain. Int J Inf Manage. 2020; 52: 102059. https://doi.org/10.1016/j.ijinfomgt.2019.102059
29. Fernandez-Carames TM, Fraga-Lamas P. Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks. IEEE access. 2020; 8: 21091–116.

https://doi.org/10.1109/ACCESS.2020.2968985

30. Amara M, Siad A. Elliptic curve cryptography and its applications. In: International workshop on systems, signal processing and their applications, WOSSPA. IEEE; 2011. p. 247–50. https://doi.org/10.1109/WOSSPA.2011.5931464

31. Barker E, Roginsky A. Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. NIST Spec Publ. 2011; 800: 131A. http://www.gocs.eu/pages/fachberichte/archiv/075-sp800-131A.pdf

32. Barker E, Roginsky A. Transitioning the use of cryptographic algorithms and key lengths. National Institute of Standards and Technology. NIST Spec Publ. 2018. https://doi.org/10.6028/NIST.SP.800-131Ar2

تأمين عقد ذكي قائم على بلوكتشين لمنع ظاهرة عدم الإنكار

نور صباح محمد[1] عمر عبدالرحمن داود[1] علي مكي صغير[1] احمد عادل نافع[2]

[1] كلية علوم الحاسوب وتكنولوجيا المعلومات ، جامعة الانبار ، الرمادي ، العراق.
[2] مركز تكنولوجيا الذكاء الاصطناعي (CAIT) ، كلية علوم وتكنولوجيا المعلومات ، جامعة ماليزيا الوطنية (UKM) ، بانجي ، سيلانجور ، ماليزيا.

الخلاصة:

بلوكتشين هي تقنية مبتكرة اكتسبت اهتمامًا في جميع القطاعات في عصر التحول الرقمي حيث تدير المعاملات وتحفظها في قاعدة بيانات. مع تزايد المعاملات المالية وسرعة تطور المجتمع مع نمو الأعمال، يبحث العديد من الأشخاص عن حلم حياة مستقلة مالياً أفضل، والتي تبتعد عن الشركات والمؤسسات الكبيرة لتشكيل الشركات الناشئة والشركات الصغيرة. في الآونة الأخيرة، أدى الطلب المتزايد على الموظفين أو المعاهد لإعداد وإدارة العقود والأوراق وعملية التحقق، بالإضافة إلى الأخطاء البشرية إلى ظهور عقد ذكي. تم تطوير العقد الذكي لتوفير الوقت وتوفير المزيد من الثقة أثناء التعامل، وكذلك لتغطية الجوانب الأمنية للإدارة الرقمية وحل مشكلات التفاوض. تم استخدام العقد الذكي في إنشاء دفتر الأستاذ الموزع لإزالة الحاجة إلى المركزية. في هذا البحث، تم تنفيذ نموذج أولي بسيط للعقد الذكي المدمج مع بلوكتشين والذي تمت محاكاته في خادم محلي مع مجموعة من العقد. تم تحقيق العديد من الأهداف الأمنية، مثل السرية والتفويض والنزاهة وعدم التنصل، في النظام المقترح. إلى جانب ذلك، ناقشت الورقة أهمية استخدام تقنية بلوكتشين، وكيف ساهمت في إدارة المعاملات، بالإضافة إلى كيفية تنفيذها في سيناريوهات عقارية شديدة الشفافية. تم استخدام العقد الذكي في إنشاء دفتر الأستاذ الموزع لإزالة الحاجة إلى المركزية. تم اعتماد المفتاح العام ذي المنحنى البيضاوي كبديل لـ RSA في عملية إنشاء او التحقق من التوقيع وبروتوكول التشفير. بالنسبة للمعاملة الآمنة، تم أيضًا اعتماد طبقة المقابس الآمنة (SSL) كطبقة آمنة في متصفح الويب. تم التحقيق في النتائج وتقييمها من جوانب مختلفة وكان التنفيذ في بيئة مقيدة. توضح لنا التجارب مدى تعقيد الوقت والتكلفة. عند استخدام خوارزمية (ECC) واستخدام خوارزمية (RSA) يعتمد على حجم وطول المفتاح. لذلك إذا كان حجم المفتاح في (ECC) يساوي (160) بت ، وكان يتوافق مع (1024) بت في (RSA) ، وهو ما يعادل 40٪ لـ (ECC) و 30٪ لـ (RSA). نتيجة لذلك، فإن خوارزمية (ECC) معقدة ، ومفتاحها أصغر وتكون عملية توليد المفتاح أسرع ، لذا فقد حققت مستوى عالٍ من الأمان.

الكلمات المفتاحية: بلوكتشين، التشفير، التوقيع الرقمي، وظيفة التجزئة، عدم الإنكار، العقد الذكي.