Implementing Blockchain for Enhancing Security and Authentication in Iraqi E-Government Services

Huda Kamil Abdali
Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq
pgs.huda.kamel@uobasrah.edu.iq

Mohammed Abdulridha Hussain
Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq
mohammed.abdulridha@uobasrah.edu.iq (corresponding author)

Zaid Ameen Abduljabbar
Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq
zaid.ameen@uobasrah.edu.iq

Vincent Omollo Nyangaresi
Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science and Technology, Bondo 40601, Kenya | Department of Applied Electronics, Saveetha School of Engineering, SIMATS, Chennai, Tami lnadu, 602105, India
vnyangaresi@jooust.ac.ke

Received: 27 August 2024 | Revised: 19 September 2024 | Accepted: 22 September 2024

 DOI: https://doi.org/10.48084/etasr.8828

ABSTRACT

E-Government is used to provide various services to citizens via an online portal and is currently available in many countries. Current e-government technology is supported by an extensive, centrally controlled database and a collection of applications linked to it through web interfaces. However, e-government depends too much on centralization. E-government services store sensitive data about citizens, making them particularly vulnerable to cyberattacks, data breaches, and access control. Therefore, alternative techniques should be developed to protect sensitive data and ensure secure storage in e-government platforms. This study proposes a safe and distributed electronic system for e-government based on blockchain technology to protect sensitive data from breaches. This system uses advanced encryption methods, including Lightweight Encryption Device (LED) and Elliptic-Curve Cryptography (ECC), to protect transmitted data. The proposed system employs a two-layer encryption approach to secure user data. The first layer utilizes the LED algorithm with a randomly generated key, and the second employs the ECC algorithm with a public key obtained from the blockchain server to enhance user data security and privacy. The proposed system allows data to be disseminated across many networks, retrieves and synchronizes data in case of unauthorized changes, and restores them to their original form. Experimental results showed that the proposed system takes an average of 0.05 seconds to complete the login process for five successful login attempts, confirming the effectiveness of the proposed approach in the execution of login procedures. The effectiveness of this system in resisting different attack types was verified through formal and informal security analyses and simulations based on the Scyther tool.

Keywords-e-government; blockchain technology; centralized; decentralized; LED; ECC

I. INTRODUCTION

Electronic administrative procedures simplify conventional activities, thus reducing the costs associated with outdated manual techniques and helping to establish an efficient, transparent, and customer-centric administrative ecosystem [1]. Digital government platforms improve connectivity by facilitating interactions through cost-effective communication methods, even when these interactions take place over long distances. The relationships between citizens, companies, and government are all being transformed by digitalization in the public sector, which also enhances the accessibility and quality of public services. By optimizing resource utilization and reducing the demand for physical input, this change in public organizations not only facilitates a shift to preserving energy sources but also makes the audience feel the environmental benefits of digitization [2].

E-Government is a global endeavor that involves providing services over the Internet or other digital platforms using Information and Communication Technology (ICT) to enhance public engagement and promote open innovation [3]. E-Government is used around the world to reform public services and improve communication between governments and the public [4]. The use of digital technology in public administration promotes the growth of inclusive institutions and contributes positively to society's general advancement [5]. E-government has been proven to offer significant benefits to its stakeholders, such as eliminating corruption, improving public service quality and administrative effectiveness, and encouraging e-democracy, openness, and a citizen-centered approach [6].

E-Government is a powerful tool for providing government services to individuals and businesses promptly, effectively, and efficiently. However, most existing e-government systems, often centralized and utilizing redundant servers and databases, may have vulnerabilities, making them susceptible to potentially devastating cyberattacks [7]. A centralized e-government public service delivery platform is a unique administrative center that typically offers a variety of public services in a unified manner. In other words, citizens, businesses, and non-residents can access a diverse selection of information products from a single digital location. The objective is to establish a single digital access point for all public services of government agencies [8]. However, traditional centralized systems are susceptible to challenges associated with privacy, lack of user control, and data breaches, which require secure and user-centric solutions [9].

Transactions amongst parties in centralized systems require the participation of a trusted third party but result in a single point of failure and hefty transaction costs. Blockchain technology overcomes these challenges by allowing untrusted actors to engage in a distributed manner without the involvement of a trustworthy third party. A blockchain is a database that keeps track of all transactions performed on a network [10]. This distributed technology protects a system against the risk of a single point of failure, as it provides enhanced security by storing data in a distributed database instead of a centralized one [11]. Blockchain also has the potential to revolutionize established sectors due to its fundamental features, including decentralization, immutability, anonymity, persistence, and auditability [12]. Since its first use in 2008, blockchain has undergone several design improvements, such as using a hash function to timestamp blocks, obviating the need for document signatures [13]. A blockchain can also be described as an assemblage of data (blocks) that are linked together through a hash function. Each block in a blockchain contains the previous block's hash value, a timestamp, and the transaction data [14]. Blockchain networks are governed through peer-to-peer (P2P) networks, where nodes collaborate to communicate and verify new blocks [15]. This technology possesses the following attributes that make it suitable for the implementation and administration of e-government systems:

- Decentralization: Decentralization indicates that no third party governs or controls e-government services on a blockchain [16]. A blockchain uses a collection of nodes and disperses data and code amongst them to maintain the network, making it decentralized [17]. However, network partners must approve any changes made to the blockchain [18].
- Enhancing security: Blockchains' decentralized architecture guarantees information security and immutability. Any information stored on the blockchain cannot be easily modified [19]. The use of hash values further secures the system [20].
- P2P communication: Through direct browser interaction with the blockchain network, users can easily connect with peers in real-time without going through a third party. The transition from centralized to decentralized blockchain applications has revolutionized data access, storage, and business operations through increased transparency, security, and flexibility [21].
- Distributed ledger: The distributed ledger contains all the details of a transaction and its participants [22]. As it is stored in the network, all other users of the system can maintain and see its contents [23].
- Consensus: Consensus algorithms are the foundation of blockchain systems. These systems use consensus algorithms to establish unanimous agreement among nodes on a certain transaction [24]. The proof of work and proof of stake methods are examples of consensus algorithms [25-26].

LED is a symmetric-key encryption method specifically designed for lightweight cryptography. The LED encryption technique operates by processing data in 64-bit blocks and provides key sizes of 64 and 128 bits. LED technology is characterized by a compact size, low power consumption, and elevated security levels [27].

This study presents the design of a blockchain-based e-government system for electronic government authentication to improve security and safeguard sensitive government data. The encryption algorithms SHA-256, Lightweight Encryption Device (LED), and Elliptic Curve Cryptography (ECC) encrypt user data sent from a node to the server to increase security.

This system also has a biometric security feature that uses cryptographic hashes of user fingerprints as an additional authentication element. This system consists of two applications: a mobile-based application and a web application for creating the web UI page. The contributions of this study are summarized as follows:

- Designs and implements an e-government system based on blockchain technology to enhance security and protect sensitive government data.
- The mobile and web applications require users to undergo two-factor authentication before accessing the system. Upon registration, users must provide a username and password and three fingerprint scans of the same finger to ensure eligibility.
- The test results show that the registration process time for five blocks, each consisting of five users, takes 0.37 seconds on average. Similarly, the login procedure for five successful logins takes 0.05 seconds on average. These reasonable timeframes confirm the efficient execution of the registration and login processes of the proposed system.
- The robustness of the system was tested and verified via formal (Scyther) and informal security analyses of various attack types. The results unequivocally show that the system is highly resistant to these attacks, providing a secure and reliable environment for e-government operations.

II. PROBLEM STATEMENT

The problem statement for electronic government in Iraq revolves around centralization, security, and authentication. Dependence on centralized databases introduces single points of failure, leaving the entire system exposed to cyberattacks. Centralization increases the risk of data breaches, illegal access, and single points of failure, leaving sensitive information vulnerable to cyberattacks and exploitation. Cyberattacks can pose a security vulnerability. Centralized e-government systems are attractive targets for hackers. Data breaches can result in the disclosure of sensitive personal information, weakening the faith in government services. Furthermore, current authentication mechanisms, such as simple passwords, are out of date and insufficiently secure. This can result in unauthorized access to sensitive information.

III. RELATED WORK

Many studies have recently focused on assessing and analyzing blockchain technology due to its crucial role in advancing new algorithms. In [28], a two-layer authentication was proposed to secure the login process to the KRG e-government system. This system was essentially a web portal for different government services. This method was based on two factors: authentication, using a username and password, and a mobile SMS to authorize legal users to log in and prevent attacks, such as replay attacks, by generating different passcodes for each access. The disadvantage of such a system based on a single website is the single point of failure of the centralized login and pool of service. In [29], blockchain security features were incorporated into the e-government system of the Ministry of Interiors Iraq and recommended using blockchain-managed digital IDs and one-time passwords to reduce instances of fraud and impersonation. However, neither the implementation of the framework nor the creation of the digital ID were described in detail.

In [30], the use of electronic smart contracts was proposed to build an organization's distributed autonomous system. This system maintains security and integrity by providing registered users with certificates containing public and private keys. However, the generated certificate is prone to manipulation or modification by unauthorized users. Sending the first messages in plain text makes the system's certificate dependence vulnerable. This vulnerability allows attackers to abuse the system's communication channel, which can lead to cyberattacks, data interception, tampering, credential theft, and privacy violations. Transmission of sensitive information in plain text jeopardizes the confidentiality, integrity, and authenticity of the message, possibly leading to unauthorized access and data breaches.

In [31], a private-blockchain-based e-voting system was proposed, which required voters or candidates to present their local civil ID upon registration. This ID was only issued locally, in a traditional manner, and each ID could only be registered once in the system through an application. A successfully registered user would receive a private key that he needed to save. Users of this system can vote through the application, where the data are stored in a private blockchain in a distributed manner. However, there is the question of whether civil IDs are synchronized with the blockchain to prevent the use of fake IDs. In addition, private keys should be sent to users in ciphertext to avoid attacks. In [32], a system was proposed that combined biometric authentication with a blockchain-based method to store and preserve sensitive government data. This system used SHA-256 encryption and Schnorr digital signatures to enhance the security of data transmission from the node to the server. However, given that SHA-256 is a one-way hash function that is not reversible, it is computationally infeasible to recover the original input from the hash value. SHA-256 prioritizes safety over speed. The comparatively longer processing time of this hashing method compared to others can be a drawback in settings where efficiency is critical. While this is an essential feature of hash functions, it does not offer the same degree of security as genuine encryption methods.

In [10], a safe and distributed e-government system was proposed based on blockchain technology. This system involves several entities, organizations, and nodes that must reach consensus when making decisions. Users of this system have the authority to start a transaction or submit a request. However, this system has several drawbacks, such as its implementation complexity due to the requirement for a consensus amongst nodes. Moreover, increasing transaction volume leads to scalability issues, while smart contract validation and digital signatures create security risks. Lack of encryption or authentication in digital signatures can allow attackers to corrupt or falsify data. In [33], a system was designed to ensure security and monitor communication among numerous nodes. This system stores communication in a

decentralized database to protect distributed ledger transactions and prevent fraud and tampering when these transactions are shared among several parties. Its primary disadvantage is its dependence on the blockchain's inherent authentication, meaning that it neither establishes authentication nor delineates the data included inside the transaction.

In [34], a framework for a decentralized P2P e-government system was proposed using blockchain technology to ensure data confidentiality and privacy while boosting public sector confidence. A theoretical and qualitative analysis of this system's security and privacy implications was conducted, and a prototype was designed. However, using key management imposes on users the additional obligation to create backups and securely store their private keys. Some users may lack the technical expertise to handle their private keys effectively. Moreover, some users may find it tedious and error-prone to generate a new blockchain address and transfer their information if they lose their private keys.

The proposed e-government system aims to address the disadvantages mentioned above by using the LED and ECC algorithms, two-factor authentication, and the inherent security advantages of blockchain. These integrated security technologies can result in a robust and secure system.

IV. PROPOSED SYSTEM

The proposed system consists of the administrator, the users, and the blockchain server.

1. Administrator: The person who supervises and administers the blockchain network, adds new servers, and verifies chain synchronization.
2. Users: Citizens are the backbone of the e-government system. Their attempts to access government services and provide their personal data are crucial. These data are recorded, verified, and transferred to the blockchain, highlighting the significant role they play in the system.
3. Blockchain servers: These servers act as intermediaries between government services and the requesting users. The system verifies a user's identity by using the stored data in the network. The process involves recording the data in the blockchain system, distributing them to all servers within the blockchain network, verifying the validity of the data, confirming the user's registration, and authenticating the user to access government services.

Figure 1 shows the registration process in the proposed system. First, the user registers with the blockchain network by entering essential data, such as his/her username and password, and three fingerprint scans of the same finger. After three fingerprints are taken, the features are stored in various positions, and the fingerprint is extracted in different orientations. The system saves these three fingerprints to help identify the user when he tries to log in. Second, the user device generates an LED key and uses it to encrypt his/her data. The LED key is an 8-byte symmetric encryption key that is randomly generated. Third, the user requests a randomly generated ECC public key from the blockchain server and uses it to encrypt the encrypted data in the previous step before sending it to the blockchain server. The blockchain uses an internal validation procedure to confirm the chain's integrity and consistency. This process verifies the validity of the blockchain and starts the decryption process, which involves decrypting data by ECC private key output (LED encrypted data, LED key), and using the LED key to decrypt the LED encrypted data to get the user data (username, password, finger hash).

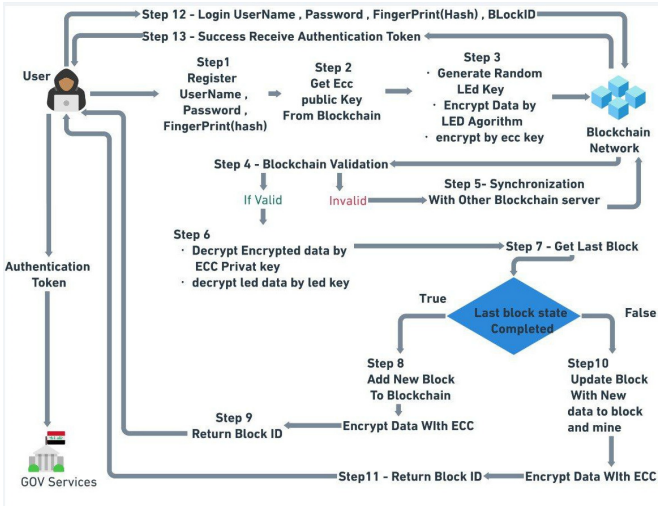

Fig. 1. The proposed system.

Suppose that the chain is discovered to be invalid. In this case, the system will synchronize the data with other blockchain servers to ensure that they are consistent throughout the network. After verifying the authenticity of the blockchain, the system receives the most recent block, which is the last block in the chain, and then starts the transaction process. When the previous retrieved block has N registered users inside, the state of this block is marked completed, and a new block is created to allow the blockchain structure to continue growing. Afterward, the ECC encryption algorithm generates new ECC random keys. The ECC then generates public and private keys that are synchronized to the blockchain network, and the public key encrypts the data. The newly created block includes the user's registration data (username, password, fingerprint_hash, current_hash, previous_hash, nonce, and timestamp) and is added to the blockchain. The user's fingerprints are securely stored for future verification. The system then returns a block ID to the registered user that he/she can use for future login attempts, reducing the number of data searches that he/she needs to perform within the blockchain and saving valuable time.

However, if the last block in the chain is incomplete, the ECC public and private keys are retrieved, and then the private key is used to decrypt the old block data. The system then updates the relevant old block data by appending new user registration data, re-encrypting the data using the public key, starting the mining operation, and replacing the block in the blockchain. After this process, the block ID is returned to the user, who then uses this ID to log into the blockchain network along with his/her username, password, and fingerprint scan. If

the user is successfully identified within the blockchain, a token is generated to authenticate his/her access, and he/she will be redirected to the requested government services. Otherwise, the system will tell the user to try again because it cannot verify his/her identity using his/her login information.

The proposed system's main phases include the blockchain initialization, registration, login, synchronization, and validation phases.

A. Blockchain Initialization Phase

When creating a blockchain, the first check involves whether there are blocks. If there are no blocks, the genesis block is created automatically. The process starts by verifying the chain's emptiness. If the blockchain is empty, the user data (i.e., username, password, and fingerprint hash) will be assigned to the genesis block. These data are then used to instantiate a new block object, with the block's characteristics (e.g., data, nonce, previous hash, and timestamp) being assigned accordingly. The genesis block's completion state is then denoted as true, signifying its ultimate conclusion. The block's hash is determined by applying the SHA-256 algorithm to the combined values of the username, password, fingerprint hash, previous hash, nonce, and timestamp. This hash serves as a distinct identification for this block.

The genesis block is then mined via the following iterative approach: all the steps are repeated until the block has the necessary number of leading zeros. First, the block's current hash is determined by concatenating its contents, nonce, previous hash, timestamp, and fingerprint hash. Second, the SHA-256 algorithm is used to obtain the hash value. Third, the prefix of the hash is compared to the predefined amount of zeros to assess the created hash against the mining difficulty. The current hash is assigned to the block's hash attribute, signifying its validity if it satisfies the mining difficulty. As a result of the mining process, the mined block is returned and added to the chain. If the present hash fails to satisfy the mining difficulty, then a new candidate is generated by increasing the block's nonce value. The mined block sequence is preserved when the modified block is added to the chain. The altered chain is ultimately preserved to guarantee the blockchain's durability and consistency.

B. Registration Phase

Upon registration, users must enter a username and password and three fingerprint scans of the same finger using a sensor, as shown in Figure 2. This phase proceeds as follows:

- Step 1: Enter username, password, and fingerprint hash on the user's device.
- Step 2: The user's device generates an LED random key and encrypts the data.
- Step 3: The ECC key is requested from the blockchain server, is encrypted, and sent to the blockchain server.
- Step 4: The blockchain server decrypts the data using the ECC private key outputs (LED encrypted data and LED key).
- Step 5: The blockchain server decrypts the LED-encrypted data using the LED key (username, password, and fingerprint hash).
- Step 6: The last block is obtained from the blockchain.
- Step 7: The last block completed state is checked as follows:

  For True completed state:

  - Generate public and private ECC keys and encrypt the data using the generated public key.
  - Sync keys with the blockchain network.
  - Generate a new block (id, data, encrypted_data, prev_Hash, current_Hash, nonce, timestamp).
  - Mine the block (data, prev_hash, time, nonce).
  - Add the mined block to the chain.
  - Save the chain.

  For False completed state:

  - Retrieve the generated public and private keys and decrypt the data using the private key.
  - Concatenate the last block data with the new user data and then encrypt the data using the public key.
  - Mine the block with the new data.
  - Set the state to complete True when the block user data count is N.
  - Replace the block in the chain.
  - Save the chain.
- Step 8: Return the block ID to the user.

During the registration procedure, the timestamp fulfills many significant functions, including:

- Key generation: The timestamp is used in the key generation process as a unique name to help the system determine the ECC public and private keys. This ensures that the keys are distinct and associated with the particular occurrence of the registration procedure.
- Preventing replay attacks: The timestamp confirms that a request is recent. By rejecting requests that are out of date, the server can determine if the timestamp falls within an acceptable time limit.

```
Algorithm 1: Registration
1.   Input: username, password, fingerprint
2.   Output: new block added to the chain
3.   user request():
4.     ledKey = GenerateLedkey();
5.     encryptedDataByLed = led.encrypt(username,
       password, finger_hash, ledKey)
6.     eccPublicKey =
       getEccKeyFromBlockchain(timeStamp)
7.     encryptedByEcc = ecc.encrypt(
       encryptedDataByLed, ledKey, eccPublicKey)
8.     sendDataToBlockchainServer(encryptedByEcc)
```

```
9.   GetECCPublicKey(timeStamp):
10.    ECCPublicKey, ECCPrivateKey=ECC.generateKeys(
       timestamp)
11.    Return ECCPublicKey
12.  Register(request):
13.    if request.method == 'POST':
14.      data = json.loads(request.body)
15.    eccEncryptedtext= data.get('encryptedByEcc')
16.    ECCPrivateKey =
       CC.getPrivateKey(eccEncryptedtext)
17.    ledEncryptedText, LedKey =
       ECC.Decrypt(eccEncryptedtext,ECCPrivateKey)
18.    username, password, fingerprint =
       Led.decrypt(ledEncryptedText,LedKey)
19.    if len(userName) < 3 or len(password) < 3 or
       len(finger_print) < 64:
20.      Return JsonResponse("Invalid Data",
         safe=False)
21.    block_chain = BlockChain()
22.    block_chain.chain = list(Block.objects.all())
23.    last_block = block_chain.chain[-1]
24.    if  last_block. completed == True :
25.      newBlock = Block()
26.      newBlock.data = userName + "," + password +
         "," + finger_print
27.      newBlock.time = datetime.now()
28.      publickey,privaeKey =
         ECCGenerateKeys(newBlock.timeStamp)
29.      newBlock.encData = ECCEncrypt(newBlock.data)
30.      newBlock.finger_print_hash = finger_print
31.      newBlock.completed =False
32.      bockAfterMining = block_chain.mine(newBlock)
33.      block_chain.add(bockAfterMining)
34.      blockchain.save()
35.    if last_block. completed == False :
36.      last_block.data += userName + "," + password
         + "," + finger_print
37.      publicKey = ECCGetKey(last_block.timeStamp)
38.      last_block.encData =
         ECCEncrypt(last_block.data)
39.      updated_block = block_chain.mine(last_block)
40.      block_chain.remove(last_block)
41.      block_chain.add(updated_block)
42.      blockchain.save()
43.  Return JsonResponse(block.id, safe=False)
```

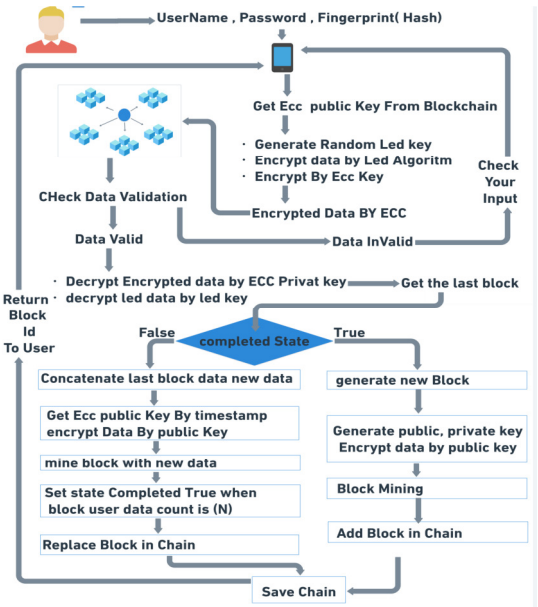

Fig. 2. The register phase.

C. *Login Phase*

To log into the system, users must provide their username, password, fingerprint hash, and ID, as shown in Figure 3. This phase proceeds as follows:

- Step 1: Check if the data is valid.
- Step 2: Check if the user sends the block ID.
- Step 3: If no block ID is sent, it is retrieved based on the username entered.
- Step 4: If found block with id or found block by name.
- Step 5: Get blockchain and validate it.
- Step 6: Get the user block by block ID.
- Step 7: Decrypt data with ECC private key.
- Step 8: Check if block data == user data:
  - If equal, return login success and user ID.
  - Otherwise, return login failed.

```
Algorithm 2: Login
1.   Input: username, password, fingerprint, block ID
2.   Output: singin status (Success, Failed)
3.   def SignIn(request):
4.     if request.method == 'POST':
5.       data = json.loads(request.body)
6.       userName = data.get('userName')
7.       password = data.get('password')
8.       fingerprint = data.get('fingerprint')
9.       blockID = data.get('blockID')
10.    if len(userName) < 3 or len(password) < 3:
11.      return JsonResponse("Invalid Sign-in Data",
         safe=False)
12.    oldBlock = FoundBlockByID(blockID)
13.    if oldBlock:
14.      block = allBlock.getBlockByID(blockID)
15.      Private_key =
         ECC.getPrivatKey(block.timeStamp)
16.      DecryptedData = ECC.decrypt(Private_key,
         block.data)
17.    else if userFoundInBlock(userName, password):
18.      authentication_token =
         generate_authentication_token()
19.      return JsonResponse({"message": "Sign-in
         successful", "token": authentication_token},
         safe=False)
20.    else:
21.      return JsonResponse("Invalid credentials",
         safe=False)
```

D. *Synchronization Phase*

The synchronization process begins when the chain is found to be invalid. As shown in Figure 1, the system will synchronize the data with other blockchain servers on the network to ensure that they are consistent throughout the network. The system initially sends a request to all other blockchain servers in the network asking for their copies of the blockchain. After receiving the blockchain data from the network participants, the system thoroughly validates each of the received blockchains to ensure their integrity and authenticity. The system proceeds with the synchronization procedure if the majority (i.e., above 51%) of the verified distant blockchains are consistent and accurate. The blocks in

the verified distant blockchains are then compared by the system to ensure that they are equal and consistent throughout the network. The verified distant blockchains are matched to the local copy of the blockchain stored in the government's server. Suppose that the local blockchain is inconsistent or distinct from most confirmed distant blockchains. In this case, the system will update and synchronize the local blockchain with the proper validated version from the network. During this process, any missing or changed blocks are added to ensure that all nodes have the same, most up-to-date view of the blockchain.

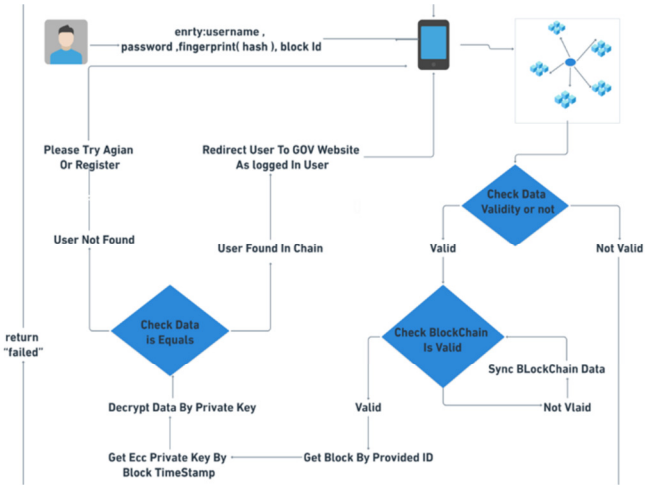

Fig. 3. The login phase.

E. Validation Phase

The blockchain also performs an internal validation procedure to confirm the chain's integrity and consistency. The validation process starts by initializing a 'chain' variable to hold the blockchain data obtained from the database. The system then verifies if the blockchain is empty or consists only of the genesis block. Both are considered valid cases. The validation procedure is repeated for every block in the chain, beginning with the second block. For each block, the system obtains the 'previous hash' value of the current block and computes the preceding block's hash. Afterward, the system confirms if the prior hash of the current block matches the computed hash of the previous block and checks if the hash of the previous block satisfies the set difficulty level, which is indicated by a string of '0' characters whose length is determined by the 'self.diff' property. If discrepancies or incorrect hashes emerge throughout this process, the blockchain is deemed invalid. In this case, the system gives a False value to signal the invalidity of the blockchain and then terminates the validation process. Conversely, if the validation loop runs without encountering problems or discrepancies, the blockchain is deemed valid, and the system generates a True value to signify its integrity.

V. RESULTS

The proposed system has two applications: a mobile-based application that uses Google's open-source Flutter UI toolkit and the DART programming language and a web application that creates the web UI page using HTML, CSS, and JavaScript. The operations in the proposed system start with the formation of the blockchain and fingerprint servers, which are programmed using the Django 4.0.3 rest framework, MySQL DBMS, SQL, and Python 3.10.2. The proposed system is designed to operate on the Windows 10 platform. In the first phase, the first (genesis) block is generated and appended to the blockchain. This block contains an initial value but does not have a previous hash. This phase is performed only once. Afterward, the registration step begins by entering user data (username, password) using the graphical interface of the mobile application and taking three fingerprint scans of the same finger using a sensor fingerprint type (ZKTECO), as seen in Figure 4. After the registration step, the user's information is stored in the database. The user may access the blockchain system via the login interface by entering his/her login credentials, including username, password, fingerprint, and block ID (Figure 5).

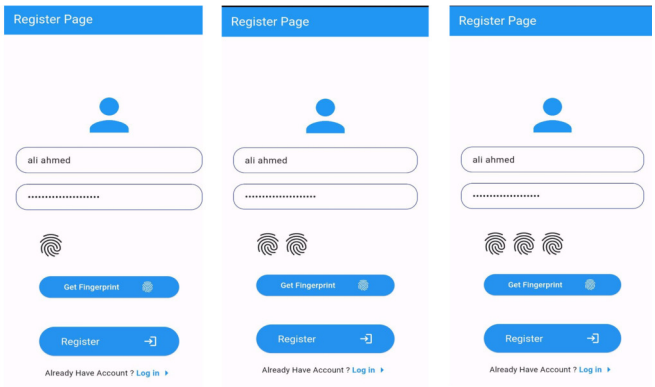

Fig. 4. The register page.

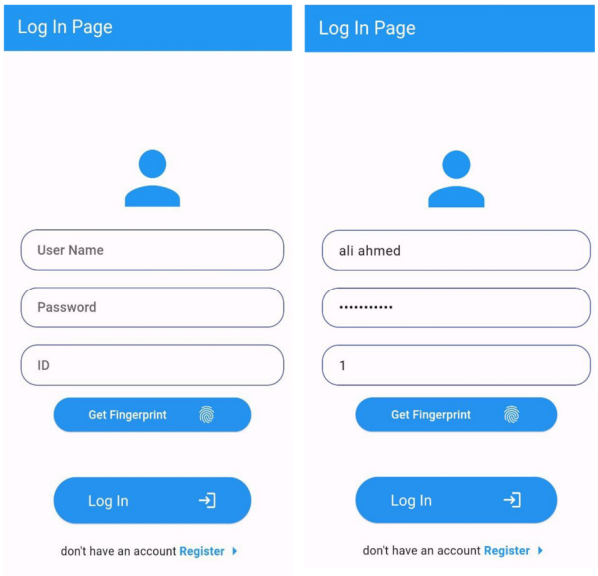

Fig. 5. The login page.

The registration process is divided into one new block and four block updates for each block, each consisting of five users. As shown in Figure 6, both the new block operation and the block update require time. This result is based on 5 blocks,

where each one contains 5 users. Testing suggested that the test should be conducted on a block containing five users. Several tests were conducted throughout the recording process. Specifically, five blocks were recorded, each including five users, totaling 25 users who entered the system. Each block comprises one new block procedure and four block updates, each with its designated time frame.

```
**********   Register ******************
time_difference New Block  0:00:00.465495
[25/May/2024 21:43:23] "POST /register HTTP/1.1" 200 1
<Response [200]>
**********   Register ******************
time_difference update block  0:00:00.313744
[25/May/2024 21:44:45] "POST /register HTTP/1.1" 200 1
<Response [200]>
**********   Register ******************
time_difference update block  0:00:00.262329
[25/May/2024 21:45:19] "POST /register HTTP/1.1" 200 1
<Response [200]>
**********   Register ******************
time_difference update block  0:00:00.289704
[25/May/2024 21:46:07] "POST /register HTTP/1.1" 200 1
<Response [200]>
**********   Register ******************
time_difference update block  0:00:00.456127
[25/May/2024 21:47:07] "POST /register HTTP/1.1" 200 1
<Response [200]>
```

Fig. 6. Times for a new block and four update blocks for Block 1.

The Average Register Time (ART) for Block 1 is calculated by:

$$Total\ sum = \sum_{user=1}^{5} time\_difference\ New\ Block\ for\ user \quad (1)$$

$$Total\ sum = 0.465495 + 0.313744 + 0.262329 + 0.289704 + 0.456127 = 1.787399 \text{ s}$$

$$ART = Total\ sum / Number\ of\ registered\ users \quad (2)$$

$$ART = 0.3574798 \text{ s}$$

So the average of the five registers is 0.3574798 s for block 1. Similarly to calculating the ART for Block 1, the ART for Blocks 2 to 5 were calculated, and the results were:

- ART for Block 2: 0.3546746 s
- ART for Block 3: 0.3812598 s
- ART for Block 4: 0.3828840 s
- ART for Block 5: 0.3847402 s.

The average time for the registration operation of 5 blocks, containing 25 users was calculated as:

$$Total\ Sum = 1.8610384 \text{ s}$$

$$Art\ of\ 5\ blocks = 0.37220768 \text{ s}$$

Therefore, the average time for the registration operation for 5 blocks, containing 25 users, takes 0.37220768 seconds. The time for the login operation was measured by recording the average time for five successful logins, with each operation having a login time, as shown in Figure 7.

```
**********   Login ******************
time_difference Login Success  0:00:00.108717
[27/May/2024 15:53:37] "POST /login HTTP/1.1" 200 58
<Response [200]>
```

Fig. 7. Time difference login success.

The time differences of successful login operations from 1 to 5 were:

- Time difference for login success 1: 0.108717 s
- Time difference for login success 2: 0.043019 s
- Time difference for login success 3: 0.063456 s
- Time difference for login success 4= 0.029207 s
- Time difference for login success 5: 0.030968 s

$$Total\ sum = \sum_{login\ success=1}^{5} time\_difference\ login\ success \quad (3)$$

$$AverageTimeforOperationLoginSuccess = TotalSum/5 \quad (4)$$

$$AverageTimeforLoginSuccessOperation = 0.055073\ s$$

Each block contains specific information as shown in Table I.

TABLE I. DATA IN BLOCK

No	Field	Type	Size (Bytes)
1	Block_id	Integer field	4
2	Time_stamp	Float field	8
3	UserData(username, password, hash(finger_print))	Char field	80
4	Previous_hash	Char field	64
5	Current_hash	Char field	64
6	Nonce	Integer field	4
7	Completed_block	Boolean field	1
8	Encryption_data	Char field	175
Size of one user in the block			400

VI. DISCUSSION

The registration process takes an average of 0.37220768 seconds to complete, calculated based on the data of five blocks, with each block comprising five users. This time is deemed reasonable, which may be ascribed to the system's efficient utilization of the blockchain transaction processing capabilities. The system can record a variable number of users, ranging from 1 to N, in each transaction on the blockchain. This system also increases the number of users registered in each block as required. Optimizing system-level transaction processing raises the capacity of blockchain technology. If the block size can expand to meet user demand, then the registration process becomes asymptotic and manageable, even for a large number of users. The proposed system achieves a subsecond average registration time by intelligently utilizing the blockchain's transaction processing capabilities. The system dynamically alters the number of users in each block to fit the current demand.

Users need an average of 0.055073 seconds to log into the system. This value was calculated based on the observations of five successful operations. It is generally regarded as a reasonable result that can be ascribed to the system's use for storing data on the blockchain. This indexing method facilitates the login process by efficiently retrieving the required data from the database based on block_ID (block number) without

complex computations or time-consuming tasks. Using block_ID also enhances the efficiency of the search process, allowing for the retrieval of the required information in under 0.05 seconds. This temporally expeditious search period is a reasonable result for the system's performance objectives.

As shown in Table I, the data size for a single user in the proposed system is 400 B (80 B for single user data and 320 B for block const data). However, uploading a complete block that includes data for five users by default results in a block size of 720 B. This outcome depends on the amount of user data (password, username, and fingerprint hash), with the remaining data being constant. This block size (720 B = 5 users × 80 B + 320 B for block data) is less than 1 KB, which is highly acceptable. If each block incorporates 1 KB of information and there are 1,000 blocks inside the system, then the overall information size may reach 1 MB. The proposed system has an ability of 1 MB, thus limiting itself to 5,000 users at most. The 400 B per user data size is an efficient and scalable design choice that allows the system to accommodate a substantial user base inside a modest 1 MB storage envelope.

In addition, the proposed system offers a higher security level by implementing various essential security measures. First, the decentralized and distributed design of the blockchain, which is widely acknowledged to provide better security compared to centralized systems, characterizes the system's intrinsic security. Second, the proposed system adopts decentralized data distribution, where data are stored across various sites in a redundant or duplicated manner, thus improving overall security and dependability. This system is designed with decentralized and redundant data storage, ensuring data integrity and system reliability. In addition, it applies multilayer encryption using the LED and ECC algorithms and features a biometric security element that uses cryptographic hashes of user fingerprints. These measures, combined with the inherent security advantages of the blockchain, provide a high degree of security.

VII. SECURITY ANALYSIS

A. Informal Security Analysis

The proposed system can effectively protect itself from different types of attacks, such as the following.

1) Reply Attacks

The proposed system prevents reply attacks by utilizing a timestamp generated by the user device and sent to the blockchain server, which verifies if the time is now or before 1 second. The system also employs two layers of encryption, where the first layer relies on an LED encryption algorithm with a randomly generated key, whereas the second layer employs ECC and a public key from the blockchain server. This procedure generates a precise timestamp for each data recording, allowing the system to verify the freshness of the received data and distinguish it from replayed data.

2) Stripping Attacks

The proposed system avoids SSL stripping attacks by using HTTPS. This protocol maintains secure connections between users and blockchain servers through SSL/TLS encryption. This encryption ensures that potential hackers cannot intercept and decode critical information, such as logins, passwords, and personal information.

3) Spoofing Attacks

The proposed system employs ECC algorithms to encrypt data on the blockchain. Even if attackers successfully capture user data, they would be incapable of deciphering it, because of the absence of the private key produced using the symmetric ECC technique. The LED key also offers an additional layer of protection.

4) 51% Attack Prevention

Using blockchain technology to design an e-government system and giving the e-government authority control over the mining process can effectively prevent 51% attacks and minimize the risk of external attacks and network takeovers. Specifically, when the e-government authority controls a large amount of mining power, attackers cannot gather enough computing power to reach 51%.

5) SQL Injection Attacks

The proposed system utilizes Django's Object-Relational Mapping (ORM) functionality, which automatically converts SQL queries into parameterized queries and effectively protects the system against direct user input of SQL code. Abstracting and sanitizing user input significantly reduces the risk of SQL injection vulnerabilities and thus effectively protects the system from SQL injection attacks.

6) Eavesdropping Attacks

The proposed system prevents eavesdropping attacks by using encryption and decryption methods. Specifically, this system applies a dual-layer encryption approach, in which the initial layer utilizes an LED encryption technique with a randomly generated key. The second layer utilizes ECC and a public key obtained from the blockchain server. The ECC private key on the receiving end is used to decrypt blockchain servers. This procedure produces LED-encrypted data and an LED key. The blockchain server decrypts the LED-encrypted data using the LED key and retrieves user data. This dual-layer encryption approach increases system complexity, significantly impeding eavesdroppers from intercepting and decrypting data.

B. Formal Security Analysis with Scyther

1) Logging into the Proposed System without Encryption

In conventional systems, the user enters his/her username, password, and fingerprint and then submits his/her request. These data are then transmitted to the blockchain as plain text without encryption. When receiving these data, the blockchain matches those with the username, password, and fingerprint (hash) stored in the database. In this instance, an attacker can seize, alter, or reroute the data to the server, allowing him/her to penetrate the system using the acquired data, as shown in Figure 8. The Scyther software simulates this login procedure, sending the data without encryption.

2) Logging into the Proposed System with Encryption

The proposed system is equipped with many security mechanisms that significantly increase the difficulty for attackers to breach the system. These mechanisms include timestamp-based authentication, which confirms that the login request is not just a replay of a previously intercepted message. In double encryption, the user initially uses the LED encryption function and a random LED key to encrypt his/her username, password, and fingerprint hash. The first encryption provides an additional level of security for sensitive data. The blockchain obtains the ECC public key and applies additional encryption to the LED-encrypted data. After receiving the encrypted credentials, the blockchain decrypts the data using the ECC private key and the LED algorithm. After receiving the encrypted user credentials, the blockchain generates and transmits a new token to the user. As a secure session identifier, this token allows users to access system resources without disclosing sensitive data. The blockchain maintains the confidentiality of the ECC private key and the decoded user credentials, thereby minimizing the risk of unauthorized access or hacking. This login protocol makes it extremely difficult for attackers to breach the system and obtain unauthorized access to the user's private data, as shown in Figure 9.

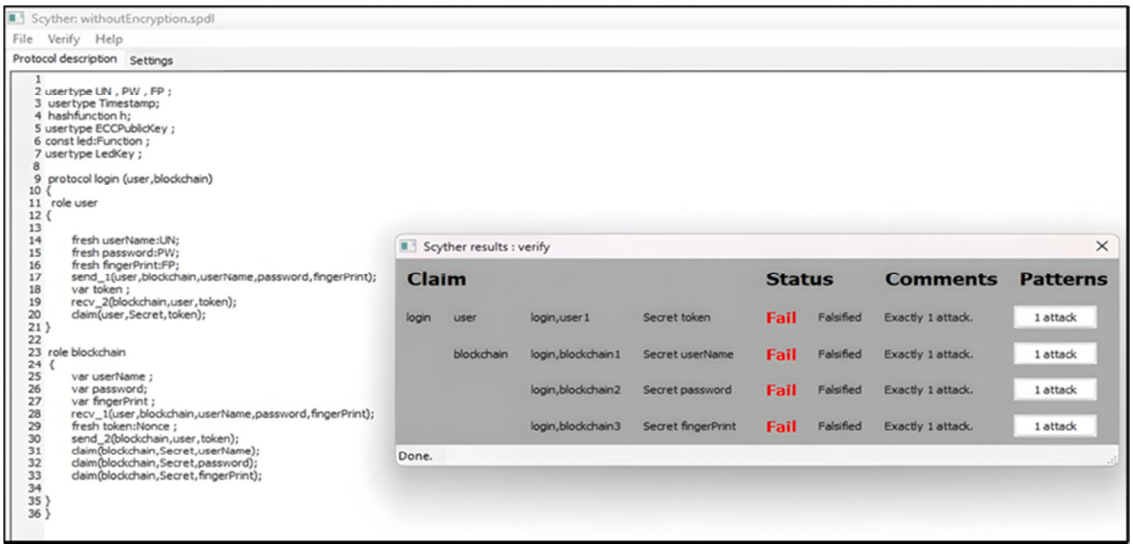

Fig. 8. Scyther results.

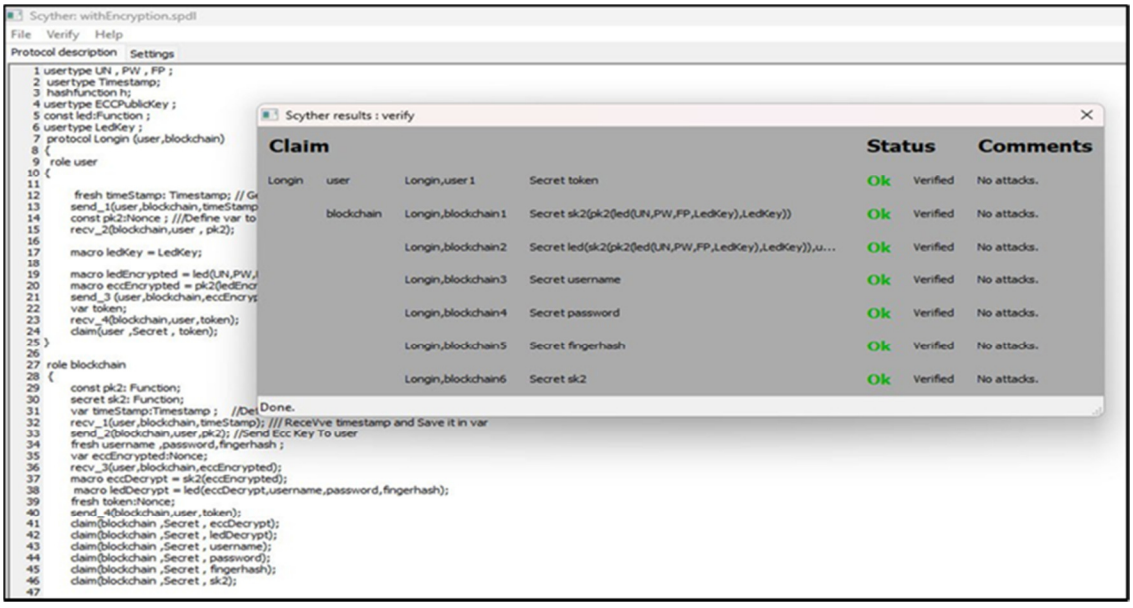

Fig. 9. Strength of the proposed login algorithm.

VIII. CONCLUSION AND FUTURE WORK

Most currently available e-government systems are centralized on servers and databases, making them vulnerable to single points of failure and cyberattacks. As a decentralized, distributed, and immutable ledger that monitors transactions and assets and fosters trust, blockchain technology offers enhanced security for these systems by storing data across multiple networks. This study proposes an e-government system that utilizes blockchain technology to maintain security and privacy. Data are stored on the blockchain using the MySQL database, and advanced encryption algorithms, such as SHA-256, LED, and ECC, are used along with biometric authentication. This system takes advantage of a decentralized and distributed blockchain-based architecture. This solution ensures the secure storage and preservation of sensitive government data, effectively addressing evolving security and privacy needs. The proposed system was also subjected to multiple tests. It reported average login and registration times of less than one second by dynamically modifying the number of users in each block and using a database design with indexing capabilities. The proposed system also achieves a high level of security by combining biometric fingerprint authentication, a two-layer encryption approach, decentralized and redundant data storage, and the inherent security advantages of the blockchain.

In the future, the proposed system can be enhanced by considering other applications, including integrating IoT devices, to expand its capabilities. For example, IoT sensors can supply real-time data for authentication, and Artificial Intelligence (AI) and Machine Learning (ML) approaches can improve several aspects of the system. ML algorithms can enhance security by detecting anomalies, preventing fraud, and analyzing user behavior.

REFERENCES

[1] N. M. Doran *et al.*, "E-Government Development—A Key Factor in Government Administration Effectiveness in the European Union," *Electronics*, vol. 12, no. 3, Jan. 2023, Art. no. 641, https://doi.org/10.3390/electronics12030641.

[2] G. Ilieva *et al.*, "Factors Influencing User Perception and Adoption of E-Government Services," *Administrative Sciences*, vol. 14, no. 3, Mar. 2024, Art. no. 54, https://doi.org/10.3390/admsci14030054.

[3] T. Thi Uyen Nguyen, P. Van Nguyen, H. Thi Ngoc Huynh, G. Q. Truong, and L. Do, "Unlocking e-government adoption: Exploring the role of perceived usefulness, ease of use, trust, and social media engagement in Vietnam," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 2, Jun. 2024, Art. no. 100291, https://doi.org/10.1016/j.joitmc.2024.100291.

[4] A. I. Alkraiji, "An examination of citizen satisfaction with mandatory e-government services: comparison of two information systems success models," *Transforming Government: People, Process and Policy*, vol. 15, no. 1, pp. 36–58, Jan. 2021, https://doi.org/10.1108/TG-01-2020-0015.

[5] M. Tokovska, V. N. Ferreira, A. Vallušova, and A. Seberíni, "E-Government—The Inclusive Way for the Future of Digital Citizenship," *Societies*, vol. 13, no. 6, Jun. 2023, Art. no. 141, https://doi.org/10.3390/soc13060141.

[6] H. Zahid, S. Ali, E. Abu-Shanab, and H. Muhammad Usama Javed, "Determinants of intention to use e-government services: An integrated marketing relation view," *Telematics and Informatics*, vol. 68, Mar. 2022, Art. no. 101778, https://doi.org/10.1016/j.tele.2022.101778.

[7] R. B. Walde and A. K. Yadav, "Blockchain Technology for e-Government," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 8, pp. 1698–1703, Aug. 2022, https://doi.org/10.22214/ijraset.2022.46487.

[8] M. Kassen, "Blockchain and public service delivery: a lifetime cross-referenced model for e-government," *Enterprise Information Systems*, vol. 18, no. 4, Apr. 2024, Art. no. 2317175, https://doi.org/10.1080/17517575.2024.2317175.

[9] A. Rodionov, "The Potential of Blockchain Technology for Creating Decentralized Identity Systems: Technical Capabilities and Legal Regulation," *International Journal of Law and Policy*, vol. 2, no. 4, pp. 19–30, Apr. 2024, https://doi.org/10.59022/ijlp.170.

[10] Z. A. Kamal and R. Fareed, "E-government based on the blockchain technology, and the evaluation of its transaction through the number of transactions completed per second," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 10, no. 1, pp. 620–631, Feb. 2022, https://doi.org/10.21533/pen.v10i1.2726.

[11] S. Umran, S. Lu, Z. Abduljabbar, and X. Tang, "A Blockchain-Based Architecture for Securing Industrial IoTs Data in Electric Smart Grid," *Computers, Materials & Continua*, vol. 74, no. 3, pp. 5389–5416, 2022, https://doi.org/10.32604/cmc.2023.034331.

[12] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, Honolulu, HI, USA, Jun. 2017, pp. 557–564, https://doi.org/10.1109/BigDataCongress.2017.85.

[13] Z. A. Al-Sulami, Z. A. Abduljabbar, V. O. Nyangaresi, and J. Ma, "Knowledge Management and its Role in the Development of a Smart University in Iraq," *TEM Journal*, pp. 1582–1592, Aug. 2023, https://doi.org/10.18421/TEM123-40.

[14] A. Razzaq *et al.*, "Use of Blockchain in Governance: A Systematic Literature Review," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019, https://doi.org/10.14569/IJACSA.2019.0100585.

[15] E. Zaghloul, T. Li, M. W. Mutka, and J. Ren, "Bitcoin and Blockchain: Security and Privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10288–10313, Jul. 2020, https://doi.org/10.1109/JIOT.2020.3004273.

[16] S. M. Umran, S. Lu, Z. A. Abduljabbar, J. Zhu, and J. Wu, "Secure Data of Industrial Internet of Things in a Cement Factory Based on a Blockchain Technology," *Applied Sciences*, vol. 11, no. 14, Jan. 2021, Art. no. 6376, https://doi.org/10.3390/app11146376.

[17] A. O. Alzahrani, M. A. Al-Khasawneh, A. A. Alarood, and E. Alsolami, "A Forensic Framework for gathering and analyzing Database Systems using Blockchain Technology," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14079–14087, Jun. 2024, https://doi.org/10.48084/etasr.7143.

[18] D. Shrier, W. Wu, and A. Pentland, "Blockchain & Infrastructure (Identity, Data Security)," Massachusetts Institute of Technology, Cambridge, MA, USA, 2016.

[19] S. M. Umran, S. Lu, Z. A. Abduljabbar, Z. Lu, B. Feng, and L. Zheng, "Secure and Privacy-preserving Data-sharing Framework based on Blockchain Technology for Al-Najaf/Iraq Oil Refinery," in *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, Haikou, China, Dec. 2022, pp. 2284–2292, https://doi.org/10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00325.

[20] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, Madrid, Spain, May 2017, pp. 468–477, https://doi.org/10.1109/CCGRID.2017.8.

[21] J. Wu and N. K. Tran, "Application of Blockchain Technology in Sustainable Energy Systems: An Overview," *Sustainability*, vol. 10, no. 9, Sep. 2018, Art. no. 3067, https://doi.org/10.3390/su10093067.

[22] S. M. Umran, S. Lu, Z. A. Abduljabbar, and V. O. Nyangaresi, "Multi-chain blockchain based secure data-sharing framework for industrial IoTs smart devices in petroleum industry," *Internet of Things*, vol. 24, Dec. 2023, Art. no. 100969, https://doi.org/10.1016/j.iot.2023.100969.

[23] Y. Yuan and F. Y. Wang, "Blockchain and Cryptocurrencies: Model, Techniques, and Applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1421–1428, Sep. 2018, https://doi.org/10.1109/TSMC.2018.2854904.

[24] H. Vu and H. Tewari, "An Efficient Peer-to-Peer Bitcoin Protocol with Probabilistic Flooding," in *Emerging Technologies in Computing*, London, UK, 2019, pp. 29–45, https://doi.org/10.1007/978-3-030-23943-5_3.

[25] S. Alsaqqa and S. Almajali, *Blockchain Technology Consensus Algorithms and Applications: A Survey*. International Association of Online Engineering, 2020, pp. 142–156.

[26] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," *IEEE Access*, vol. 7, pp. 85727–85745, 2019, https://doi.org/10.1109/ACCESS.2019.2925010.

[27] S. Boubaker, F. S. Alsubaei, Y. Said, and H. E. Ahmed, "Lightweight Cryptography for Connected Vehicles Communication Security on Edge Devices," *Electronics*, vol. 12, no. 19, Jan. 2023, Art. no. 4090, https://doi.org/10.3390/electronics12194090.

[28] S. Mirza Abdullah, M. M. Ameen, S. Asaad Ahmed, and A. Najdat Muhamad, "A Two-Layer Authentication Security Through Personal Mobile SMS for KRG-Iraq E-Government System," *Eurasian Journal of Science and Engineering*, vol. 8, no. 3, 2023.

[29] M. M. Al-Musawi, "Transforming One-Stop E-Services in Iraq: Focusing on perception of Blockchain Technology in Digital Identity System," in *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, Seattle, WA, USA, Oct. 2020, pp. 1–4, https://doi.org/10.1109/GHTC46280.2020.9342959.

[30] N. Diallo *et al.*, "eGov-DAO: a Better Government using Blockchain based Decentralized Autonomous Organization," in *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*, Ambato, Ecuador, Apr. 2018, pp. 166–171, https://doi.org/10.1109/ICEDEG.2018.8372356.

[31] B. Alothman, C. Jumaa, S. Alshammeri, and M. Khan, "Development of a Secure Preserve E-Voting System Using Private Blockchain Solutions," *International Journal of Innovative Science, Engineering & Technology*, vol. 09, no. 10, 2022.

[32] M. M. Ashor and H. M. Al-Mashhadi, "Enhanced Security of Iraqi National Card Based on Blockchain Technique," *Iraqi Journal of Intelligent Computing and Informatics (IJICI)*, vol. 2, no. 2, pp. 58–67, 2023.

[33] R. F. Ghani and Z. A. Kamal, "A Proposed Authentication Method for Document in Blockchain Based E-Government System," *Iraqi Journal of Computers, Communications, Control, and Systems Engineering*, vol. 22, no. 4, 2022.

[34] N. Elisa, L. Yang, F. Chao, and Y. Cao, "A framework of blockchain-based secure and privacy-preserving E-government system," *Wireless Networks*, vol. 29, no. 3, pp. 1005–1015, Apr. 2023, https://doi.org/10.1007/s11276-018-1883-0.