BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **25**, No 2
Sofia • 2025 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2025-0014

Geo-Blockchain Verification Based on Proof of Secret Sharing Modification via Chialvo Map

Sarah Mohammed Shareef[1], *Rehab Flaih Hassan*[2]

[1]*University of Technology – Iraq, Department of Production Engineering and Metallurgy, Department of Computer Science, 10066 Baghdad, Iraq*
[2]*University of Technology – Iraq, Department of Computer Science, 10066 Baghdad, Iraq*
E-mails: sarah.m.ali@uotechnology.edu.iq rehab.f.hassan@uotechnology.edu.iq

Abstract: *Blockchain technology has attracted substantial attention from global organizations because of its promise as a solution to centralized system issues. This paper presents a new approach utilizing Geo-Blockchain technology, an integration between blockchain technology and Geographic Information System* (*GIS*) *based on Proof of Consensus Verification* (*PoCV*) *that can establish a decentralized and tamper-proof record of property transactions. Moreover, the paper proposes combining Modified SLIM Cryptography* (*MSLIMC*) *with the Chialvo map to verify the integrity of real estate transactions data and retrieve documents. The GIS of nodes is a condition of secret sharing by using the longitude and latitude of all nodes in the number generation that uses secret shares. The proposed system was evaluated through different metrics like latency time of encryption and decryption of MSLIMC algorithm, Geo-Blockchain built time, and using a set of tests and general rules provided by the National Institute of Standards and Technology* (*NIST*)*, which tests approximately 97% of the generated random data to ensure that it is sufficiently unpredictable and suitable for cryptographic applications.*

Keywords: *Geo-Blockchain, SLIM Algorithm, Chialvo map, PoCV, Real estate.*

1. Introduction

Geo-Blockchain is a combination of GIS and blockchain, as well as a unique distributed data storage mechanism that includes safety, traceability, and credibility. Geo-Blockchain is defined as the repeated storage of blockchain data on different dispersed nodes. If a service node is destroyed, the data is not lost; hence, the data can be identified through the entire process; it can safeguard data from unwanted change [1, 2]. Blockchain is a time-stamped, decentralized series of fixed records containing data of any quantity. It is governed by a wide network of computers distributed around the world and not held by a single entity. Every block is encrypted and connected via hashing technology, preventing it from being altered by an unauthorized user [3, 4]. Because of its decentralized structure, the blockchain cannot be hacked, significantly enhancing cybersecurity. To attack the blockchain or its smart contracts, an attacker would need to successfully break into more than half of the system's nodes [5]. In cryptography, secret sharing is a means to securely transfer chunks of critical private information within a distributed network or group. Such systems are particularly effective for preserving extremely sensitive information like private cryptographic keys or biometric data [6, 7].

In recent times, various blockchain models have been utilized to store and secure data in the networks against 51% attacks. Most of these methods work effectively, but the major limitation is that Blockchain ensures data immutability, but it cannot verify whether the original data input is true or fraudulent. If false or forged documents are added to the chain, they become permanently preserved, making early validation critical. To surpass the above limitations, the paper proposes a system that verifies nodes, each of which has a specialty in conducting real estate transactions, in integration with the three main nodes (main real estate office, fax office, and service office), so that any sub-real estate office can decode the data with the three main nodes via PoCV. Also, verify transactions; all transactions on a specific date are shared in a blockchain specific to that date. Any tampering with any part produces a different final hash. However, in the event of a matching hash, the document retrieval process occurs, which depends on the integration of subnodes with three main nodes. Thus, the algorithm unlocks the transaction code required by the user.

2. Related works

Many academics concentrate on combining blockchain with various technologies and identifying the key properties of each technology, as well as their benefits and drawbacks. The authors in [8] presented a blockchain-based secure solution for mobile phone commerce, which is an important tool for promoting social entrepreneurship and sustainable development. It can contribute to the development of reliable and long-lasting mobile commerce platforms that promote social responsibility, customer confidence, and company ethics. Researchers and developers can utilize it to enhance social commerce platforms and hence improve the effectiveness of m-commerce. The authors in [9] suggested that the technique of blockchains be used with the data encryption standard algorithm to increase the degree of security of the shared photographs by enhancing the key used in the method of encryption, as well as boosting the amount of authentication between the person who sent them and the recipient. The testing results reveal that the security of the encryption image made using the recommended technique is higher, fulfilling the goal of protecting medical image features, as evidenced by the results in Entropy, Mean Square Error (MSE), and Peak Signal-to-Noise Ratio (PSNR). The authors in [10] studied the existing scenario and related problems in recording land and real estate ownership records, particularly in developing countries, and identified the obstacles and prospects for using blockchain technology in this industry. Examined whether blockchain can bring beneficial changes and play an essential role in the real estate market. It certainly has the potential to carry a positive change and play a vital part by increasing transparency, decreasing errors and delays, and enhancing the overall efficiency of the industry. The authors in [11] highlighted current research insights on several blockchain applications in cybersecurity. The researchers primarily prioritize protecting IoT systems, networks, and data. Concluded that blockchain technology can protect data from being viewed or manipulated by intruders by employing encoded blocks that can only be accessed by authorized individuals. Mohammed and Abdul Wahab [12] proposed a system that combines a Paillier cryptosystem homomorphic platform with IoT and lightweight blockchain technology to decentralize the IoT environment and boost security. The dataset utilized in this paper was developed using the machine industry 4.0 Storage System status. The dataset utilized to create and analyze the proposed system is industrial Internet of Things data. The system is evaluated using common metrics used to measure blockchain effectiveness, time, and resources consumed, and it performs better in terms of time and power usage. Kamal and Ghani [13] supplied a secure decentralized ledger by monitoring all the moves of sending and receiving proposed transactions. Every time a transaction is transmitted or received by nodes, it is authenticated using numerous techniques, including encrypting the transaction data and producing a unique hash for it. Chains are built using SQL databases. The technology has proven useful by providing a more secure messaging system with high credibility and tamper resistance. Papantoniou and Hilton [14] created a new idea known as Geo-Blockchain, which is defined here as a solution artifact that can be used to track the geographical and spatial behaviors and trends exhibited by participants (users) using blockchain technology, transactions, and geo-locations. Private blockchains like Hyperledger Fabric and geospatial technologies like ArcGIS can be used for any Geo-Blockchain application. Seven Q-Set criteria were developed for the two Geo-Blockchain enterprise solution prototypes using the Q-methodology basics. The authors in [15] proposed an innovative Merkle tree-based technique for protecting the accuracy of student records and explained how to implement it. This architecture illustrates how learning activities based on smart contracts, or blockchain structures, can be verifiable, dependable, and traceable. It also introduces the cryptography system's framework and proposes five new dimensions of chaotic map academic records. The study used DeoxyriboNucleic Acid (DNA) sequences and operations, as well as a chaotic system, to reinforce the cryptosystem used in blockchain authentication and permission.

The main contribution of this paper is an integration of blockchain with a GIS system based on secret sharing and Modified SLIM Cryptography via Chialvo map, which decreases the time and computational demands for such a system while improving the integrity, speed, and privacy of real estate transactions from fraud, theft, and 51% attacks, as well as providing immutability using the blockchain. Generate the modified proof of secret shares algorithm is Proof of Consensus Verification (PoCV), based on the Chialvo map to ensure that each node within the network has the right to participate in the consensus decision and verify the integrity of the distributed ledger before starting the process.

3. Shamir's Secret Sharing (SSS)

Shamir's Secret Sharing Scheme is a technique first presented in 1979 by the renowned cryptographer Adi Shamir (in [16]). It is one of the cryptographic approaches used to keep personal data safe and secure, including biometric data, private keys, and any other personal information that should not be made public. It permits information to be divided into multiple shares, with just a percentage of those shares needed to reconstruct the original secret. This means that, rather than requiring all shares to reconstruct the original secret, Shamir's approach requires a set number of shares, known as the threshold. To reassemble the secret, a certain threshold must be met. If there is anything less than the threshold, the secret cannot be recovered, making Shamir's Secret Sharing secure versus a hostile attacker with unbounded computational capacity [17, 18].

4. Chialvo map

The suggested method's key generation is based on a specific chaotic map known as the Chialvo map. The model is a duplicate map, with the following equations at each time step:

(1) $$x_{n+1} = x_n^2 e^{(y_n - x_n)} + k,$$

(2) $$y_{n+1} = ay_n + bx_n + c.$$

When y is the recovery variable and x is referred to as the activation or action possibility variable. The four parameters of the sample are as follows: *a,* recovery time constant (a<1); *b*, recovery activation dependency (b<1); and *c*, offset constant. k is a time-independent additive perturbation or constant bias. The sample exhibits rich dynamics, responding to small stochastic fluctuations and exhibiting oscillatory to chaotic behavior [19, 20].

5. SLIM Cryptography (SLIMC)

SLIM is a lightweight block encryption algorithm that employs a Feistel design and a block that is 32 bits. To avoid extensive key searches, SLIM utilizes a large key length of 80 bits. SLIM uses robust four-by-four substitution boxes to assess the relationship between ciphertext and plaintext data. SLIM has shown strong resistance to the most successful linear and differential cryptanalysis techniques, it has a significant protective buffer against types of assaults [21]. The standard approach is appropriate for wireless networks, particularly wireless sensor networks and Internet of Things applications, where data streams usually fall within a specific byte range. The next equations explains the entire processing during each round, where the right portion of the entered data R_i with the sub-key is modified utilizing an XOR technique [22]:

(3) $$L_i = R_i - 1,$$

(4) $$R_i = L_{i-1} + P\left(S\left(K_i + R_{i-1}\right)\right).$$

6. Proposed method

The proposed system provides a method of updating the activity occurring on properties for a particular area by updating a period, for example, a day, a week, a month, or a year, and incorporating the constraints before and after the update in the blockchain process. This ensures that there is no fraud or subsequent manipulation process when changing these restrictions. The data set is an adapted version of the California Housing Data. The collection consists of 16 columns and 20,641 rows that provide information about residences in specific California counties as well as summary statistics [23]. The purpose and benefit of the proposed method is to protect real estate transactions from fraud and unauthorized access, as well as to obtain a secure real estate ownership document using a combination of Geospatial Information Systems (GIS), Blockchain technology, Chialvo map, Secret sharing, and encryption method (MSLIMC). Below are the steps of the proposed suggestion:

A. The steps of encrypted Real estate transactions

The proposed method includes the Modified SLIM algorithm in the field of data in blockchain; the GIS technology represented by node locations is used with Chialvo map for generating numbers that are used in secret sharing to control the system from any malicious attack (malicious node). Together, these technologies create a robust framework for executing encrypted real estate transactions, safeguarding sensitive data while ensuring transparency and trust among the parties involved.

A.1. Blockchain transactions

The transactions of the blockchain consist of records on which a change operation is carried out, such as selling or buying (changing the owner's name), for example, in the relevant department, in addition to all the details of the property, which are in the form of a single record. The process of changing ownership depends on the competent employee, as he cannot change ownership except through integration between the main nodes of the system (main real estate office, fax office, and service office) with one of the sub-nodes. The main node receives transactions from all nodes in the system. At the end of the day, all these records construct a blockchain to produce a final hash, which is used for retrieving the specific information after matching. The set of records that have been updated is converted into a string and connected in a single string to be encrypted according to the proposed algorithm, the result of which will be entered into a hash function. Fig. 1 provides a brief overview of each component.

A.2. Modified Lightweight SLIM Algorithm (MLSLIM Algorithm)

This algorithm depends on the Feistel framework, the block size was changed to 64 bits, divided into two parts, each part is 32 bits, and entered in steps close to thin, reducing the number of rounds to 16 or 8. The necessary keys are generated by the proposed Chialvo map, which gives us a series of numbers that appear random. The result of this stage enters the 2D S-box, which is generated by the Chialvo map under different conditions; this process is repeated four times. The encrypted transactions are stored within databases. The MSLIM Algorithm is shown in Fig. 2.

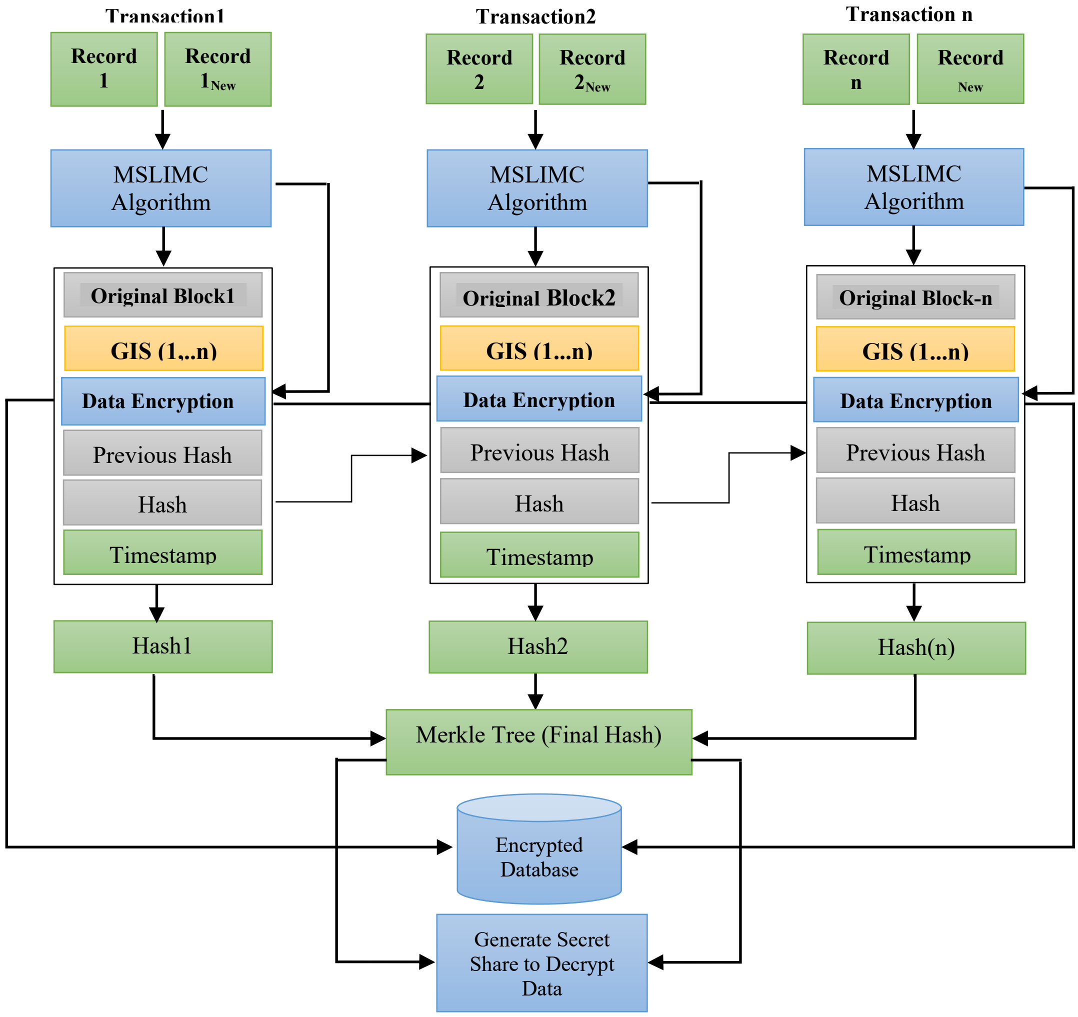

Fig. 1. Proposed method of blockchain model for encrypted transactions

A.3. Key generation based on GIS

It refers to the geographic location information of several nodes (main and sub-real estate offices) where real estate transactions are completed. The geographic location (latitude and longitude) is input to seed number generation based on the Chialvo map that is used to generate the secret sharing, as well as to retrieve the secret when three main real estate offices are identified with one of the sub-offices as the threshold limit.

A.4. Previous hash

This is the hash of the Prior block in the blockchain, which connects the current block to the previous one and guarantees the sequential order of the blocks [24]. In this proposal, the modified SHA512 is suggested to also get a number generated from the Chialvo map. This proposed hash function is used in all required places in the system. As for the other blocks, their value is present in the previous blocks.

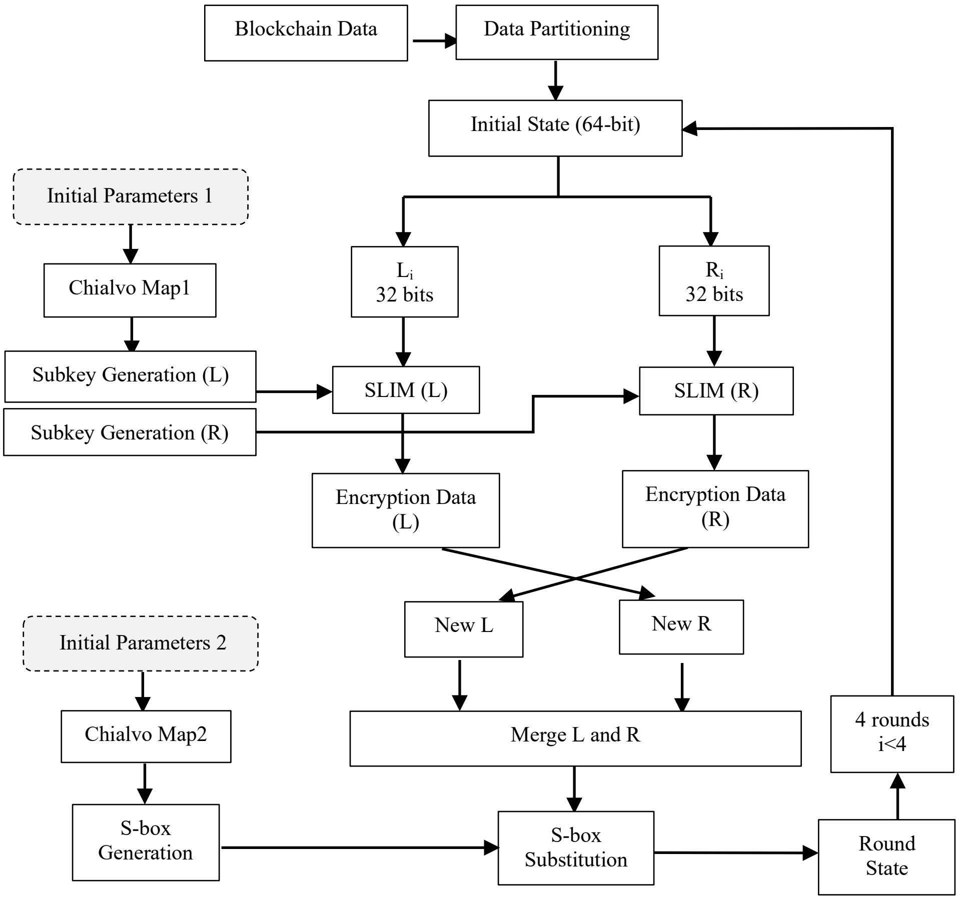

Fig. 2. The proposed method with modified lightweight SLIM and 2D S-Box cryptography

A.5. Time stamp (Date stamp)

One field of data in a transaction is a time stamp that is used to verify the value of a hash function over time. In the proposed method, used date stamp was used to join all records that were updated in day. The time stamp indicates when the block was generated or added to the blockchain.

A.6. Current hash value

The current hash of the block is put in the section block to represent the current block by applying modified SHA512 on all content of the block after merging it into a string. All previous values are summarized in a string: encrypted data, GIS of real estate offices, and a time stamp. They are entered into a hash function, resulting in a fixed hexadecimal length.

A.7. Merkle tree

All blocks within one day are merged into one blockchain structure for applying a Merkle tree, which mixes all blocks as leaves in the tree and applies Modified SHA512 in each pair of blocks. If the number of blocks is even, the final block is duplicated; if the number of blocks is odd. The same procedure is applied to the results until a root that represents the final cryptographic hash of all transactions in the block is found. It functions as a concise representation of the whole collection of transactions and contributes to the block's integrity.

A.8. Generate secret share to Decrypt data

After obtaining the final hash from the system, the secret will be created to open the data code and obtain the required transaction document. The details will be explained in step B.

B. Decrypted transactions to obtain Real estate documents

Refers to the process of accessing and retrieving real estate documents, such as deeds, titles, and contracts, by decrypting encrypted transaction data. This typically involves using cryptographic techniques to secure sensitive information during transactions, ensuring that only authorized parties can access the documents. The decrypted data can then be used for legal purposes, property transfers, or verifying ownership, enhancing transparency and security in real estate dealings.

B.1. Request for Real estate document

The user (buyer or seller) requests a real estate document related to the property from the real estate registration department, which includes information about the seller and buyer, property information, and details to verify their identities through matching the Merkle Tree by determining the date of the transaction based on it. Here's a detailed overview of each component as shown in Fig. 3.

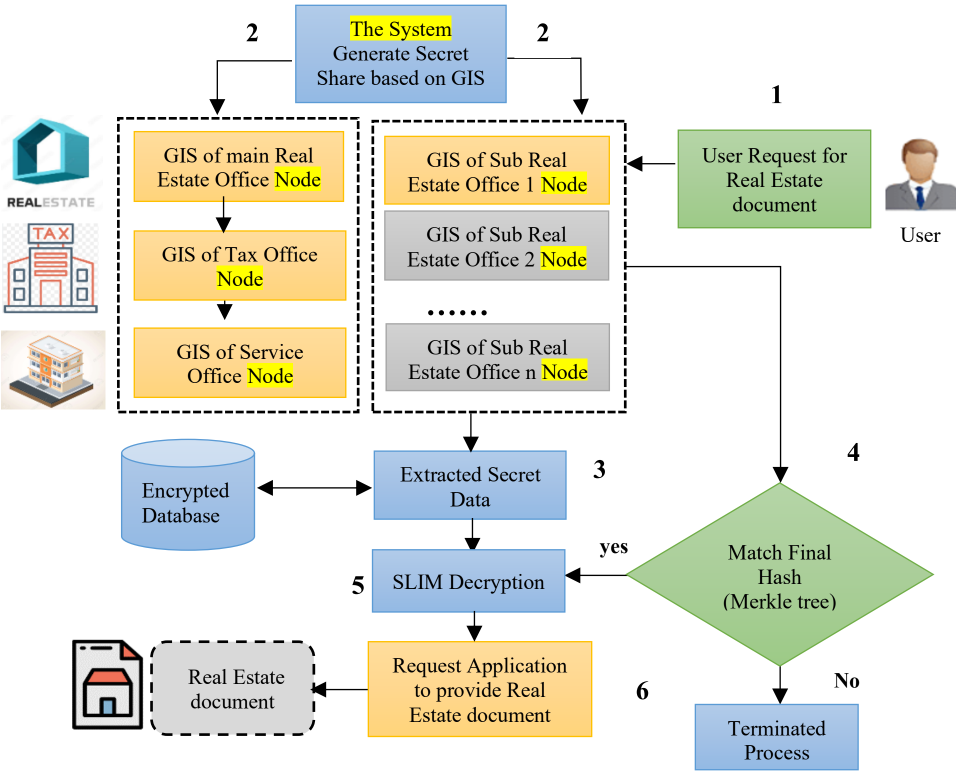
2
The System
Generate Secret
Share based on GIS
2
1
REALESTATE
TAX
GIS of main Real Estate Office Node
GIS of Tax Office Node
GIS of Service Office Node
GIS of Sub Real Estate Office 1 Node
GIS of Sub Real Estate Office 2 Node
......
GIS of Sub Real Estate Office n Node
User Request for Real Estate document
User
Encrypted Database
Extracted Secret Data
3
4
yes
Match Final Hash (Merkle tree)
5
SLIM Decryption
Real Estate document
Request Application to provide Real Estate document
6
No
Terminated Process

Fig. 3. Proposed method for real estate property document extraction process

B.2. Generate secret and share based on GIS and Chialvo map

The system is generating a secret and sharing it based on GIS and Chialvo Map. The proposed modified proof of secret shares is a Proof of Consensus Verification (PoCV) algorithm to verify the geo-location of the added data or transactions and to ensure each node within the network has the right to engage in a consensus decision, as well as before beginning the operation validate the distributed ledger's integrity.

The consensus procedure of the PoCV comprises the following phases:

Phase 1. Secret and share generation. Utilizing one of the secret-sharing algorithms (Shamir's Secret Sharing is employed in this study) to create shares that are stored under the authorization nodes throughout the network [25]. These shares can be used to check the permission of the node to be utilized within the scheme. This phase involves the following steps:

Step1. Secret generating: Select the final hash obtained from the Merkle tree that is used to generate a share using the data of location (Longitude, Latitude) and Chialvo map equal to the number of nodes in the network, *N* as the number of shares, and the threshold value, is 4 nodes to recreate the secret share.

Step 2. Share generation: Depending on the number of *N* (the number of permissioned nodes within the network), *N* number of shares are produced using data about location (Longitude, Latitude) and Chialvo map, then saved within the node as shown in Fig. 4.

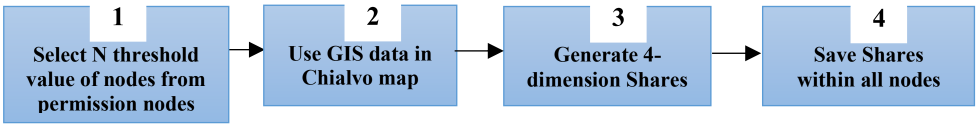
1
Select N threshold value of nodes from permission nodes
2
Use GIS data in Chialvo map
3
Generate 4-dimension Shares
4
Save Shares within all nodes

Fig. 4. Generate shares based on GIS and Chialvo map to verify the data integrity

Phase 2. Each node does the consensus phase, participates in secret share verification to guarantee that only the allowed node contributes to the operation, and includes the following two steps.

Step 1. Node Verification and Secret Calculation. All nodes within the network will verify the location of each node by generating a share using the location data (longitude and latitude) and the Chialvo map, similar to the number of nodes within the network. Then, the system chooses a threshold of four nodes (the main real estate office, fax office, service office, and one sub-real estate office among the remaining branches) to retrieve the secret. The system administrator checks if the final secret differs from the initial one; the procedure is terminated, and more research is necessary to identify the tampered node.

Step 2. If all of the involved nodes approve the node Verification, the secret recovery process is also correct, and if there is a match between the real estate transaction's final hash blockchain and the final hash of the requested transaction, then the data is decrypted by SLIM cryptography. If the hashes differ, the decryption operation is canceled.

When multiple users are dishonest in a blockchain or distributed system, the integrity of the network can be significantly compromised. Such behavior may lead to coordinated attacks or data manipulation. System security depends heavily on the design assumptions: the proportion of honest nodes, cryptographic guarantees, and the difficulty of collusion. Data integrity, trust, and consensus can be compromised without these safeguards.

B.3. Extracted secret

The verification nodes involved in the transaction use their shares of the secret to reconstruct the original secret. This step verifies the participating nodes' authenticity and integrity. The verifying process is shown in Fig. 5.

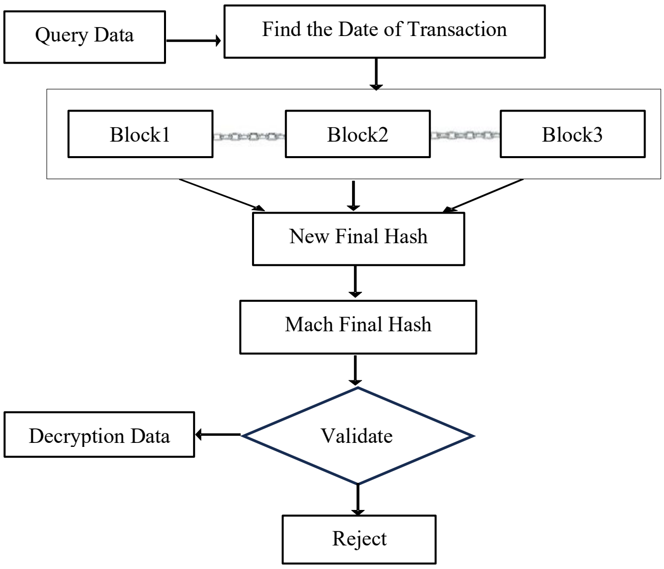

Fig. 5. Decryption of proposed

B.4. Match final hash (Merkle tree)

This step verifies the authenticity and integrity of the buyer and seller data. The system matches the final hash of the record of the requested transaction with the previous hash stored in the Merkle tree when encrypting the data, determining the date and time of the transaction based on it.

B.5. MSLIMC Decryption

The real estate transactions are decrypted after the reconstructed secret and matched hashes. MSLIMC decryption ensures that the decryption process is secure and that no single party can access the decrypted transaction alone.

B.6. Request application to provide real estate document

The user who requested the document can now access the verified and decrypted real estate document.

7. Experimental results and discussion

The experiments of the proposed model were conducted with a latency time of encryption, Blockchain built time, Merkle tree-built time, the throughput of MSLIM algorithm, the Memory usage of MSLIM Algorithm in kilobytes, and Hamming Distance of final hashes as demonstrated below.

7.1. Latency time of encryption and decryption algorithm

Refers to the delay or time it takes to encrypt or decrypt data during a transaction. Latency time is important in designing and implementing secure systems, as it balances security needs with performance requirements, as shown in Table 1.

1. Encryption Times. The encryption times increase progressively from 51.005 ms for transaction 3 to 120.909 ms for transaction 7. This trend suggests that as the transaction number increases, the time required for encryption also rises, indicating potentially increasing complexity or size of the data being encrypted.

2. Decryption Times. Similar to encryption, the decryption times also show an upward trend, starting at 35.729 ms for transaction 3 and reaching 85.076 ms for transaction 7. The decryption times are generally lower than encryption times across all transactions, which is common as decryption often requires less computational effort than encryption.

Table 1. Latency time of encryption and decryption

Transaction	Encryption time (ms)	Decryption time (ms)
3	51.005	35.729
4	68.132	47.793
5	85.560	60.347
6	102.975	72.598
7	120.909	85.076

7.2. Geo-blockchain built time

Refers to the duration it takes to create and add a new block to a blockchain. This process involves several steps, including:

Transaction verification. Validating the transactions that will be included in the block.

Consensus mechanism. Achieving agreement among network participants (validators) on the validity of the transactions.

Block creation. structuring the verified transactions into a block.

Adding to the chain. Appending the new block to the existing blockchain, as shown in Table 2.

Table 2. Geo-blockchain built time

	Geo-Blockchain time (ms)				
No	3-tansaction	4-transaction	5-transaction	6-transacion	7-transaction
1	14.467	17.698	23.581	27.476	29.019
2	14.702	20.283	22.308	24.126	28.366
3	14.657	19.110	22.573	25.976	27.995
4	14.301	17.700	19.861	26.652	30.318
5	14.674	17.470	23.068	22.520	27.564
Average	14.560	18.452	22.278	25.350	28.652

The detailed discussion analysis of the data:

Row analysis. Each row represents a Different test or instance of building the blockchain, showing some variability in times. For instance, the time for 5-transactions ranges from 23.581 ms up to 23.068 ms, indicating minor fluctuations between tests.

Average times. The average time for each transaction count is provided in the last row. Notably, the average time increases with the number of transactions: 3 transactions 14.560 ms; 4 transactions 18.452 ms; 5 transactions 22.278 ms; 6 transactions 25.35054783 ms; 7 transactions28.65296176 ms. This reinforces the observation that the time to build a blockchain grows as the number of transactions increases.

7.3. Merkle tree-build time

Refers to the time required to create a Merkle tree, which is a data format used in blockchains and cryptography to efficiently and securely confirm the integrity of big datasets. A Merkle tree is constructed by hashing pairs of data blocks (or transactions) and then combining those hashes recursively until a single hash, known as the Merkle root, is obtained. The construction time depends on the number of transactions or data blocks. It typically takes logarithmic time relative to the number of blocks, making it efficient for large datasets, as shown in Table 3.

Table 3. Merkle tree built time

	Merkle tree time (ms)				
No	3-tansaction	4-transaction	5-transaction	6-transacion	7-transaction
1	5.711	8.763	12.373	15.916	15.926
2	5.296	8.561	12.889	15.968	15.938
3	5.536	8.654	12.854	16.063	16.506
4	5.245	8.295	12.488	15.696	16.603
5	5.078	8.995	12.646	15.360	16.920
Average	5.373	8.654	12.650	15.801	16.378

Here’s a detailed discussion analysis of the data:

Row analysis. Each row represents different instances or tests of building the Merkle tree. There is noticeable variability in times across instances for the same transaction count. For instance, for 3 transactions, the times vary from 5.711 ms up to 5.078 ms, indicating some fluctuations in performance across tests.

Average times. The average time for each transaction count is calculated in the last row, showing a clear increase in time with more transactions: 3 transactions 5.373 ms; 4 transactions 8.654 ms; 5 transactions 12.650 ms; 6 transactions 15.801 ms; 7 transactions 16.378 ms. This average further supports the observation that building time increases consistently as the number of transactions grows.

7.4. Throughput of MSLIM Algorithm

Refers to the amount of encrypted data or the number of operations that the MSLIMC Algorithm can process in a given time frame. Usually calculated in units of transactions per second or items processed per second, it indicates how efficiently the algorithm can handle data, as shown in Table 4.

The detailed discussion analysis of the data:

Row analysis. Each row represents different instances of throughput for the respective number of transactions and bit sizes. For instance, for 4 transactions, the throughput values range from 1.516 up to 1.533, indicating slight variations in performance, possibly due to different testing conditions or system loads.

Average throughput. The average throughput for each configuration is provided in the last row. The averages show a general increase as the bit size and transaction count rise: 3 transactions 1.553; 4 transactions 1.555; 5 transactions 1.634; 6 transactions 1.762; 7 transactions 1.769.

The averages indicate a positive correlation between the number of transactions and throughput, particularly notable at higher bit sizes.

Table 4.Throughput of MSLIMC Algorithm

No of bits	3-transaction (1572-bit)	4-transaction (2096-bit)	5-transaction (2620-bit)	6-transacion (3144-bit)	7-transaction (3668-bit)
1	1.568	1.516	1.615	1.576	1.769
2	1.568	1.597	1.626	1.538	1.513
3	1.521	1.528	1.761	1.528	1.501
4	1.581	1.602	1.645	1.762	1.935
5	1.527	1.533	1.536	1.785	1.897
Average	1.553	1.555	1.637	1.638	1.723

7.5. Memory usage of MSLIM Algorithm in Kilobytes

Refers to the Amount of Memory (RAM) that the MSLIM algorithm consumes while it is running, measured in kilobytes (KB). Factors Influencing Memory Usage:

Data size. Larger datasets typically require more memory.

Algorithm complexity. The design of the MSLIM algorithm, including how it stores and processes data, affects memory requirements.

Implementation. Different programming languages and libraries can have varying memory overhead, as shown in Table 5.

Table 5. Memory Usage of MSLIM Algorithm in Kilobytes

No of bits	3-transaction (1572-bit)	4-transaction (2096-bit)	5-transaction (2620-bit)	6-transacion (3144-bit)	7-transaction (3668-bit)
1	5688	11,344	22632	45,152	90080
2	10,568	21,080	42,056	83,904	167,392
3	7624	15,208	30,336	60,520	120,736
4	13,360	26,656	53,176	106,088	211,648
5	9624	19,200	38,304	76416	152,448
Average	9372.8	18,697.6	37,300.8	74416	148,460.8

The detailed discussion analysis of the data:

Row analysis. Each row represents different instances of memory usage corresponding to specific transaction counts and bit sizes. For instance, for 5 transactions, memory usage ranges from 22,632 KB to 38,304 KB, demonstrating significant variability that could be attributed to different system conditions or configurations during testing.

Average memory usage. The average memory usage for each configuration is calculated in the last row. The averages show a clear increase as the number of transactions rises: 3 transactions 937.2 KB; 4 transactions 1869.67 KB; 5 transactions 74,416.8 KB; 6 transactions 76,416 KB; 7 transactions 90,080 KB.

This suggests a strong correlation between the number of transactions and memory consumption, particularly as the transaction count increases.

7.6. Hamming distance blockchain

The Hamming distance between two equal-length characters or vectors is the number of sites where their associated symbols differ. In simpler terms, it determines the minimum number of substitutions required to convert one string to a different one, as well as the minimum number of errors that may have occurred during the transformation. In a larger sense, the Hamming distance is one of several string statistics that calculate the edit distance between two strings. The HD for five real estate transactions is completed in Table 6.

Table 6. Hamming distance of final hashes of same inputs at different times

Hamming distance final hash							
	No	1	2	3	4	5	6
Final hash	1	0	254	257	238	249	241
	2	254	0	267	250	251	259
	3	257	267	0	247	264	264
	4	238	250	247	0	249	251
	5	249	251	264	249	0	230
	6	241	259	264	251	230	0

The detailed discussion analysis of the data:

Row analysis. Each row represents Hamming distances for a specific hash (0 to 6) compared to other hashes. For example, the first row shows a consistent Hamming distance of 254, 257, and 238 when compared to other hashes, indicating a moderate level of variability in hash outputs.

The distances between certain hashes show a pattern of fluctuation. For instance, hash 0 shows a Hamming distance of 0 with itself and distances of 254, 257, and others with different hashes.

The highest recorded distances (e.g., 267) suggest that environmental changes or variations in the hashing process could lead to significant differences in outputs, raising questions about the stability and reliability of the hashing algorithm over time.

7.7. NIST Test (Key generation)

A set of generated chains has been tested using the NIST global test. Table 7, titled “NIST Test of Key Generation”, presents the results of various statistical tests conducted to evaluate the randomness and quality of key generation processes. Each test is associated with a P-value, which indicates the likelihood that the observed results could occur under a random distribution.

The comparison Table 8 for the provided references addresses the element of time in their respective contexts, focusing on the efficiency of processes, timely security measures, and rapid responses to threats. The studies illustrate how blockchain technology can enhance time efficiency in various applications, from mobile commerce to educational record management, while maintaining robust security features.

Table 7. NIST Test of Key Generation

No	Test	P-value	Status		Test	P-value	Status
1	“Random excursion test”	0.938822	“Pass”	6	“Non-overlapping template matching”	0.967748	“Pass”
2	“Frequency test within a Block test”	0.876548	“Pass”	7	“Random excursion variant test”	0.802584	“Pass”
3	“The longest run of one”	0.912748	“Pass”	8	“Overlapping template matching test”	0.729638	“Pass”
4	“Frequency Monobit Test”	0.81589	“Pass”	9	“Cumulative Sums Test”	0.974789	“Pass”
5	“Run Test”	0.55856	“Pass”	10	“Serial Test”	0.94573	“Pass”

Table 8. Comparison of the proposed system with the previous work

Reference	Technology	Security focus	Key metric (s)	Proposed method	Benefits	Limitations
Jamil and Rahma [9]	Circular blockchain with Modified DES	Secure medical image access	Avgerage encryption time 6.8 ms	Optimized encryption method	Lightweight and faster encryption	Limited scalability for large datasets
Lee and Kim [10]	Blockchain as a cyber defense	Timely updates for defense	Update latency 2.1 s	Framework for integrating blockchain	Good for dynamic attack scenarios	No fine-grained access control
Uppalapu and Agarwal [11]	Advanced blockchain applications	Timely responses to threats	Detection accuracy 92%	Strategic blockchain implementation	Effective in proactive risk detection	Deployment cost is high
Mohammed and Abdul Wahab [12]	Decentralized IoT with blockchain	Timely data protection	Encryption time 5.2 ms	Quick data encryption and access	Low latency, decentralized	Not ideal for heavy computation tasks
Kamal and Ghani [13]	Blockchain for e-government authentication	Quick authentication to reduce fraud	Verification time 3.5 ms	Authentication protocol for rapid verification	Accurate and auditable	Complex contract management
Papantoniou and Hilton [14]	Geo-Blockchain for land and supply chain	Timely record updates	Avgerage update delay 1.8 s	Criteria for timely data management	Precise updates with logs	Authority-centric – single point of failure
Proposed method	Geo-blockchain	High security based on secret sharing and MSLIMC	Avgerage time 16.37 ms (7 tx), 15.8 ms (6 tx)	Integration of blockchain with GIS technology based on secret sharing	Fast, verifiable, auditable, and interpretable	Blockchain scalability

The Chialvo map was specifically chosen due to its unique properties, making it especially well-suited for modeling complex, nonlinear dynamical systems with chaotic behavior. Table 9 shows how the Chialvo map benefits the system compared to alternatives:

Table 9. Chialvo Map with other chiotic behavior

Feature	Chialvo map	Alternatives (e.g., henon, logistic)
Discrete and computationally simple	✓	✓
Exhibits complex dynamics/chaos	✓	✓
Biological relevance	✓	Limited
Low-dimensional with rich behavior	✓	✓
Easy to tune and control	✓	Varies
Integration with digital systems	✓	✓

The Proof of Consensus Verification (PoCV) was specifically chosen for the proposed system, and its characteristics benefit the use case, especially in secure, transaction-heavy domains like real estate, over alternative consensus mechanisms, as shown in Table 10.

Table 10. PoCV with alternative consensus mechanisms

Feature	PoCV		PoW / PoS / RAFT / BFT
Fast, lightweight verification	✓		PoW is slow, PoS/RAFT are complex
Low resource use	✓	Green and efficient	PoW is energy-intensive
Secure against tampering	✓	Via cryptographic proofs	PoS can be economically biased
Plug-and-play architecture	✓	Highly modular	Some are rigid or hard to integrate

8. Challenges and solutions

The integration of GIS and Blockchain innovation has the potential to improve the way we organize and transmit spatial data, but there are challenges to overcome. Addressing these challenges is critical for enabling seamless integration between these two technologies and allowing their full potential for improving spatial administration and real-world applications. This section explores these challenges:

1. Data storage:

- GIS data is often voluminous and complex, while blockchain's native storage capacity is limited;

- Storing large geospatial datasets directly on the blockchain can be inefficient and expensive.

Solutions:

- Off-chain storage: Store large datasets off-chain and only store hashes or pointers on the blockchain;

- Data compression and optimization: Reduce data size before storing it on the blockchain.

2. Scalability. Blockchain technology, especially early iterations, can struggle to handle the high transaction volumes and data throughput required for real-time GIS applications.

Solutions:

- Sharding: Partition the blockchain into smaller subchains to improve scalability;

- Improved consensus mechanisms: Explore more efficient consensus algorithms that can handle higher transaction throughput.

3. Data privacy. Ensuring the confidentiality and privacy of sensitive geospatial data within a transparent blockchain environment is a complex challenge.

Solutions:

- Homomorphic encryption allows computations on encrypted data without disclosing the original data;

- Private blockchains: Restrict access to the blockchain to authorized users only.

9. Conclusion and future scope

The integration of GIS and Blockchain innovation creates a secure and transparent platform for geographic data administration, sharing, and analysis. Combining the two technologies allows us to build a robust platform for organizing and distributing spatial data that is protected, transparent, and decentralized. Adding location to the blockchain will improve security and validity because the same transaction could not occur in two places simultaneously. The system verifies the nodes and real estate transactions by integrating one of the sub-nodes with the three main nodes (main real estate office, fax office, service office) to achieve PoCV, then decrypts the data to retrieve the document for the user submitting the transaction. The data (transaction) submitted by the user is encrypted using the SLIM algorithm, whose keys are generated by the Chialvo map. Thus, the encrypted transactions are entered into the blockchain, and a hash is issued for each transaction to be collected as a final hash (Merkle tree). When the user requests a real estate document, a request is submitted to the real estate office within a specific area. After that, a secret is generated by taking the final hash of the transactions and dividing it into shares based on the GIS coordinates of the area. When the secret matches the final hash, the required transaction data code is opened according to the date of submitting the transaction, and a real estate deed is obtained for the user. The suggested system shortens the time necessary for each operation and provides an encrypted environment. The average time to build the geoblockchain for three real estate transactions was 5.373, while the average time to execute seven transactions was 16.378. The results from the NIST tests indicate that the key generation process is largely effective in producing random keys, reaching 97% pass for the Cumulative Sums Test. For Future Work: Adaptive Proof of Secret Sharing: Develop adaptive mechanisms for proof of secret sharing that can respond to emerging threats and vulnerabilities in real time. Integration of Machine Learning: Explore the integration of machine learning algorithms to enhance decision-making processes and automate verification tasks within the geo-blockchain system.

References

1. Papantoniou, C. GeoBlockchain: The Analysis, Design, and Evaluation of a Spatially Enabled Blockchain. – M.S. Theses, CGU, Glaremont, CA, 2021. **https://dl.acm.org/doi/book/10.5555/AAI28862141**
2. Zhao, P., J. R. C. Jimenez, M. A. Brovelli, A. Mansourian. Towards Geospatial Blockchain: A Review of Research on Blockchain Technology Applied to Geospatial Data. – In: Proc. of 25th AGILE C H. Al-Hamami. Analysis and Improvement of Geographic Information Systems for Problem Solving and Decision Making. – Journal port Science Research, Vol. **6** (special), 2023, pp.107-117. DOI: 10.36371/port.
3. Kamal, Z. A., R. F. Ghani, A. K. Farhan. Blockchain-Based e-Government System Using WebSocket Protocol. – Engineering and Technology Journal, 2024, pp. 421-429.
4. Shareef, S. M., R. F. Hassan. Improved Blockchain Technique Based on Modified SLIM Algorithm for Cyber Security. – Mesopotamian Journal of CyberSecurity, Vol. **5**, 2025, No 1, pp. 147-164. DOI: 10.58496/MJCS/2025/010.
5. Swati, J., P. Nitin. Securing Decentralized Storage in Blockchain: A Hybrid Cryptographic Framework. – Cybernetics and Information Technologies, Vol. **24**, 2024, No 2, pp. 16-31.

6. Mesnager, S., A. Sınak, O. Yayla. Threshold-Based Post-Quantum Secure Verifiable Multi-Secret Sharing for Distributed Storage Blockchain. – MDPI, Mathematics, Vol. **8**, 2020, 2218. DOI: 10.3390/math8122218.

7. Kaur, M., S. Gupta, D. Kumar, C. Verma, B. Neagu, M. S. Raboaca. Delegated Proof of Accessibility (DPoAC): A Novel Consensus Protocol for Blockchain Systems. – MDPI, Mathematics, Vol. **10**, 2022, 2336. DOI: 10.3390/math10132336.

8. Riskhan, B., S. M. H. Almassri, K. Hussain, H. A. J. Safuan. Blockchain-Based Cybersecurity Proposal in Commerce Mobile Platforms for Social and Sustainability Businesses. – Metaverse, Vol. **5**, 2024, No 1, pp. 1-14. DOI: 10.54517/m.v5i1.2415.

9. Jamil, A. S., A. M. S. Rahma. Cyber Security for Medical Image Encryption Using Circular Blockchain Technology Based on Modified DES Algorithm. – International Journal of Online and Biomedical Engineering (iJOE), Vol. **19**, 2023, No 3.

10. Lee, S., S. Kim. Blockchain as a Cyber Defense: Opportunities, Applications, and Challenges. – IEEE Access, Vol. **10**, 2021, pp. 2602-2618. DOI: 10.1109/ACCESS.2021.3136328.

11. Uppalapu, V. K., A. Agarwal. Enhancing Cybersecurity through the Utilization of Blockchain Technology. – Journal of Propulsion Technology, Vol. **45**, 2024, No 1, pp. 4076-4081.

12. Mohammed, M. A., H. B. Abdul Wahab. Decentralized IoT System Based on Blockchain and Homomorphic Technologies. – Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. **23**, 2023, No 3. DOI: https://doi.org/10.33103/uot.ijccce.23.3.3.

13. Kamal, Z. A., R. F. Ghani. A Proposed Authentication Method for Documents in Blockchain-Based E-Government System. – Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. **22**, 2022, No 4. DOI: 10.33103/uot.ijccce.22.4.10.

14. Papantoniou, C., B. Hilton. Enterprise Solutions Criteria in the Age of GeoBlockchain: Land Ownership and Supply Chain. – In: Proc. of 54th Hawaii International Conference on System.

15. Balobaid, A. S., Y. H. Alagrash, A. H. Fadel, J. N. Hasoon. Modeling of Blockchain with Encryption-Based Secure Education Record Management System. – Egyptian Informatics Journal, 2023. DOI: 10.1016/j.eij.2023.100411.

16. VenkataRao, S., V. Ananth. A Hybrid Optimization Algorithm and Shamir Secret Sharing Based Secure Data Transmission for IoT-Based WSN. – International Journal of Intelligent Engineering and Systems, Vol. **14**, 2021, No 6. DOI: 10.22266/ijies2021.1231.44.

17. Oudah, M. Sh., A. T. Maolood. Lightweight Authentication Model for IoT Environments Based on Enhanced Elliptic Curve Digital Signature and Shamir Secret Share. – International Journal of Intelligent Engineering and Systems, Vol. **15**, 2022, No 5. DOI: 10.22266/ijies2022.1031.08.

18. Soni, M. Optimized Security Mechanism for Publicly Secret Key Sharing over Cloud Using Blockchain. – Journal of Engineering, Science and Mathematics, Vol. **2**, 2023, No 2, pp. 73-85.
https://jesm.in/archives

19. Salih, A. A., Z. A. Abdulrazaq, H. G. Ayoub. Design and Enhancing Security Performance of Image Cryptography System Based on Fixed Point Chaotic Maps Stream Ciphers in FPGA. – Baghdad Science Journal, 2024. DOI: 10.21123/bsj.2024.10521. P-ISSN: 2078-8665. E-ISSN: 2411-7986.

20. Pilarczyk, P., G. Graff. An Absorbing Set for the Chialvo Map. – Elsevier BV, Communications in Nonlinear Science and Numerical Simulation, 2024, 107947.

21. Sugio, N., N. Shibayama, Y. Igarashi. Higher-Order Differential Attack on Reduced-Round SLIM. – Journal of Information Processing, Vol. **32**, 2024, pp. 352-357. DOI: 10.2197/ipsjjip.32.352.

22. Aboushosha, B., R. A. Ramadan, A. D. Dwivedi, A. El-Sayed, M. M. Dessouky. SLIM: A Lightweight Block Cipher for Internet of Health Things. – IEEE Access, 2020. DOI: 10.1109/ACCESS.2020.3036589.

23. **https://www.kaggle.com/datasets/abdallahsamman/california-housing-with-name-of-counties**.

24. Mohialden, Y. M., N. M. Hussien. The Role of Blockchain Technology in Enhancing Data Integrity and Transparency Across Industries. – CyberSystem Journal (CSJ), Vol. **1**, 2024, No 1, pp. 33-41.

25. Mohammed, M. A., H. B. Abdul Wahab. Enhancing IoT Data Security with Lightweight Blockchain and Okamoto Uchiyama Homomorphic Encryption. – Computer Modeling in Engineering & Sciences (CMES), Vol. **138**, 2024, No 2.

Fast track. Received: 06.03.2025, Fist revision: 16.04.2025, Second Revision: 26.04.2025, Accepted: 07.05.2025