Secure and Distributed Data Sharing with Blockchain and Shamir's Secret Sharing

Usama Habib, Ayad Mashaan Turky

***Abstract*—As a secure distributed ledger technology, block chain has attracted widespread attention from academia and industry for its decentralization, immutability and traceability. This paper explores a secure data sharing approach that leverages block chain technology and Shamir's Secret Sharing for robust key management. Data is fragmented and encrypted using Fernet, with the encryption key split and distributed using Shamir's Secret Sharing for enhanced security. While a simulated block chain showcases potential transaction data storage and retrieval, key shares are stored in separate locations, offering distributed storage and fault tolerance. This approach ensures data confidentiality, enables granular access control, and increases fault tolerance compared to traditional storage methods. Further research is needed for real-world block chain integration, scalability for large data sets, and additional security considerations.**

***Index Terms*—Block Chain, Shamir's Secret Sharing, Digital Twin and Fernet Symmetric Encryption Algorithm.**

I. Introduction

A. Security Attacks

Traditional centralized systems are vulnerable to a myriad of security attacks, ranging from data breaches to malware infections. Cybercriminals exploit weaknesses in network infrastructure, software vulnerabilities, and human error to compromise sensitive information and disrupt operations. Common attacks include phishing scams, DDoS attacks, and ransomware infections, posing significant threats to organizations and individuals alike [2]. However, block chain technology offers a promising solution to enhance security and mitigate these risks. By decentralizing data storage and transaction processing, block chain reduces the single points of failure inherent in centralized systems. Additionally, cryptographic techniques such as digital signatures and hash functions ensure data integrity and authentication, bolstering the resilience of block chain-based systems against malicious activities.

Despite its robust security features, block chain is not immune to security attacks, albeit in a different context. Threats specific to block chain ecosystems include 51% attacks, double-spending attacks, and smart contract vulnerabilities [3] . These attacks exploit weaknesses in consensus mechanisms, network protocols, and smart contract code, potentially undermining the integrity and trustworthiness of block chain networks. To address these challenges, block chain developers and researchers have implemented various mitigation strategies. Enhanced consensus algorithms, such as Proof of Stake (PoS) and Delegated Proof of Stake (DPoS), reduce the likelihood of 51% attacks by requiring validators to stake cryptocurrency as collateral. Moreover, rigorous auditing and testing of smart contracts help identify and rectify vulnerabilities before deployment, minimizing the risk of exploitation.

Furthermore, advancements in privacy-preserving technologies like zero-knowledge proofs and ring signatures enhance transaction confidentiality while maintaining transparency and audit-ability. By continuously innovating and fortifying security measures, the block chain community strives to foster a secure and trustworthy environment for decentralized applications and digital asset management.

B. Block Chain

Block chain technology operates on a decentralized ledger system, fundamentally altering traditional modes of data management and transactional processes. At its core, a blockchain is a chain of blocks, where each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. The process begins with a user initiating a transaction, whether it involves cryptocurrency exchange, smart contracts, or data transfer. This transaction is broadcasted to a network of nodes, each maintaining a copy of the ledger. Next, miners or validators on the network verify the transaction's validity through complex cryptography algorithms. Once verified, the transaction is grouped with other transactions into a block. The block is then added to the existing chain, forming an immutable record of transactions. This decentralized and transparent nature ensures that transactions are secure, tamper-proof, and traceable. Additionally, consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS) ensure agreement among network participants, further enhancing security and reliability [1].

Block chains can be broadly categorized into public, private, and consortium block chains, each with distinct characteristics and use cases. Public block chains, exemplified by Bitcoin and Ethereum, are open to anyone, allowing for permission less participation and transparency. In contrast, private block chains restrict access, typically used by enterprises for internal operations, offering enhanced privacy and control over data. Consortium block chains, also known as federated block chains, are governed by a group of organizations rather than a single entity. These block chains combine the benefits of public and private networks, enabling shared control and selective participation. Furthermore, variations in consensus mechanisms, such as Delegated Proof of Stake (DPoS) or Practical Byzantine Fault Tolerance (PBFT), influence factors like scalability, security, and decentralization. Overview of block chain process is illustrated in figure 1 Block Chain working diagram [6] (https://www.geeksforgeeks.org/how-does-the-blockchain-work/).

Monrat et al. [1] mentions applications of block chain that includes cryptocurrencies, encompassing diverse sectors such as supply chain management, healthcare, and voting systems. In supply chain management, block chain ensures transparency and traceability, enabling stakeholders to track the journey of products from manufacturer to end consumer. Similarly, in healthcare, block chain secures patient records, facilitates interoperability, and enhances data integrity. Moreover, block chain technology intersects with emerging trends like digital twins, which are virtual replicas of physical assets or processes. By integrating block chain with digital twins, businesses can ensure data authenticity, facilitate secure transactions, and enable automated decision-making. This synergy empowers industries to harness the full potential of digital transformation, fostering innovation, efficiency, and trust in an increasingly interconnected world.

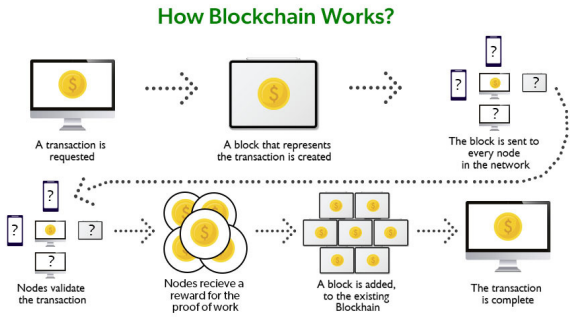

Fig. 1. Block Chain working diagram

C. Digital Twins

Digital twin technology represents a paradigm shift in how we conceptualize, design, and manage physical assets and processes. At its core, a digital twin is a virtual representation of a physical object or system, encompassing its characteristics, behavior, and interactions in real-time. This virtual counterpart enables organizations to gain deeper insights, optimize performance, and make data-driven decisions across the entire lifecycle of assets, from design and manufacturing to operation and maintenance. The working principle of a digital twin involves the seamless integration of various data sources, including sensors, IoT devices, and simulation models, to create a dynamic and holistic representation of the physical entity. These data sources continuously feed real-time information to the digital twin, allowing it to mirror the state and behavior of its physical counterpart with a high degree of fidelity. Advanced analytics techniques, such as machine learning and predictive algorithms, analyze this data to identify patterns, anomalies, and optimization opportunities, enabling proactive maintenance, predictive analytics, and scenario simulation [4].

D. Shamir's Secret Sharing

A critical aspect of secure data storage and access control involves managing encryption keys. Traditional methods often rely on a central authority to store the key, creating a single point of vulnerability. Shamir's Secret Sharing (SSS) offers a robust alternative by distributing the key into multiple fragments, or shares. These shares are then distributed to various participants in the system. The beauty of SSS lies in its ability to reconstruct the original key even if some shares are lost or compromised. This is achieved through a mathematical scheme that defines the minimum number of shares (threshold) required for successful reconstruction. For instance, a key can be split into five shares, where any three shares can be used to recover the key. This threshold value provides flexibility in controlling access and mitigating the risk of unauthorized key retrieval [5].

In essence, Shamir's Secret Sharing empowers a decentralized approach to key management. By distributing shares among multiple participants, the system eliminates the need for a central authority and enhances the overall security posture. This technique plays a vital role in the proposed block chain-based data sharing system, as we will explore in the following section.

E. Fernet Symmetric Encryption Algorithm

Fernet itself is a specific type of symmetric encryption algorithm. This means that the same secret key is used for both encryption and decryption. Fernet uses a single secret key for both encryption and decryption. This key needs to be shared securely between the sender and receiver for successful communication. Fernet is considered a secure encryption algorithm and offers a good level of protection for your data. The cryptography.fernet library provides a relatively user-friendly way to implement Fernet encryption in your Python code.

Here's an analogy to understand Fernet better: Imagine a locked box (encrypted data). You need a key (secret key) to open the box (decrypt) and see the contents. However, the same key can also be used to lock the box again (encrypt) and hide the contents from unauthorized access. It's important to note that while Fernet is a good choice for many scenarios, it's not suitable for all situations. Sharing the secret key securely is crucial for Fernet's effectiveness. If the key is compromised, the encryption can be broken. Fernet only encrypts data, it doesn't guarantee the integrity of the data (i.e., it doesn't ensure the data hasn't been tampered with during transmission).

II. Literature Review

In recent years, the intersection of block chain technology and security has been a focal point of research and innovation, driving trans-formative advancements across various industries. As the foundational technology behind cryptocurrencies like Bitcoin, block chain has evolved beyond its initial use case to revolutionize processes ranging from supply chain management to identity verification. Concurrently, the proliferation of cyber threats and attacks has underscored the critical importance of robust security measures in block chain ecosystems. This literature review aims to explore the latest advancements and techniques in block chain technology and security, encompassing developments in consensus mechanisms, privacy-preserving techniques, smart contract security, and resilience against adversarial attacks. By synthesizing insights

from academic research, industry initiatives, and real-world applications, this review seeks to provide a comprehensive understanding of the current landscape and future directions in block chain security, with implications for enhancing trust, integrity, and resilience in decentralized systems.

Huang et al. [7] proposes a blockchain-based key management system for cloud storage. To resist brute-force attacks launched by adversaries against ciphertexts, the system uses oblivious pseudorandom function (OPRF) to generate random hash keys and improve data confidentiality. Second, the system improves the reliability of concurrent key management through a secret sharding mechanism, where aggregate keys are split into key fragments and distributed for storage on the blockchain. Even if a certain number of key fragments are lost or damaged, users can still recover the complete key data using block transaction records. In addition, the system effectively supports file deletion and block-level data security. Security analysis and experimental performance evaluation show that this scheme can ensure key security and data confidentiality, and has little computational cost to generate file-level encryption keys according to this scheme. Even for a 100 MB file, the computational load required to generate encryption keys is less than 2 s, which improves computational efficiency.

Paper [8] integrates cloud storage into the framework, which allows Big Digital Twin Data (BDTD) to be encrypted and stored in the cloud, while BDTD hash and transaction records are stored on the block chain. Some of the new block generation rules are designed to improve the speed of block chain processing. An algorithm for choosing the optimal sampling rate to maximize the social utility of the participants in the BDTD distribution is presented. The simulation results show that the algorithm is better than the traditional method for maximizing social benefits. Evaluation results show that BDTD can be safely distributed multiple times per second through the framework, which shows the feasibility of the framework in supporting timely distribution of BDTD.

The paper [9] introduces a novel Block Chain-Based Federated Cooperative Learning (BL-FCL) scheme to enhance security and privacy in digital twins within mobile distributed machine learning systems. The proposed scheme combines block chain technology with federated learning to address data privacy protection challenges in distributed machine learning environments. The BL-FCL model consists of two layers: the customer end layer, where individual clients train their data sets locally using a width learning model, and the server layer, where the server aggregates model weights from clients to generate a global model. By securely aggregating model weights at the server level, the BL-FCL scheme aims to improve prediction accuracy while ensuring data privacy and security in mobile distributed federated learning with low computational costs. The integration of blockchain technology in the scheme provides a decentralized and transparent framework for secure network data sharing and privacy protection within digital twins, promoting the development and application of the digital economy.

In a simulation study to validate the performance of the BL-FCL algorithm, the results demonstrate the effectiveness of the proposed scheme in enhancing security and privacy in digital twins. The scheme successfully addresses data privacy breach risks and privacy protection in distributed machine learning environments within digital twins. The simulation experiments show that as the number of devices increases, the detection probability exhibits a rapid decrease, highlighting the scalability and efficiency of the BL-FCL model. Moreover, the prediction accuracy of the BL-FCL outperforms existing federated averaging algorithm-based schemes by 20%–60%, emphasizing the scheme's capability to deal with inaccurate training while ensuring user privacy and security. Overall, it indicate that the BL-FCL scheme offers a promising solution for ensuring the security of digital twins and advancing the development of the digital economy through secure distributed learning methodologies [9].

The paper [10] proposes a novel blockchain-IoT solution for secure smart homes, aiming to enhance user experience and address security challenges in IoT networks within smart home systems. The proposed solution integrates hyper ledger fabric and hyper ledger composer to meet security requirements such as confidentiality, integrity, authorization, and availability. This integrated solution is designed to overcome security limitations in commonly used permissioned block chain approaches by mapping smart home attributes to hyper ledger composer attributes, creating a customized security solution for IoT-based smart homes. The proposed architecture consists of four layers: Cloud storage, Hyper ledger fabric, Hyper ledger composer, and a smart home layer. By mapping smart home attributes to hyper ledger composer attributes, the solution aims to achieve enhanced security for smart homes. Experimental results with the proposed approach have verified its effectiveness in satisfying all defined smart home security requirements, leveraging block chain features such as transparency and interoperability.

Thada et al. [11] used block chain technology that allows handling of time series data, public-key encryption, and proof of work by both municipal workers and local people. This enables the verification of work through solving mathematical problems using devices managed by the client interface. During the development and testing phase, the system's cost and power consumption were evaluated for economic feasibility. The entire system was set up and deployed at a cost of 600 INR. Various corner cases were simulated to ensure the proper functioning of the system. The block chain data structure was utilized to add hierarchy to the authentication process, incorporating two-level authentication to regulate work.

The Trusted Twins for Securing Cyber-Physical Systems (TTS-CPS) framework [12] works by integrating block chain technology with Digital Twins to enhance security and trustworthiness in cyber-physical systems. Components such as Engineering Knowledge, Domain Knowledge, and Rule Generator are combined to establish trusted twins, which are virtual replicas of physical counterparts. Block chain technology is leveraged to store and retrieve Safety and Security rules, ensuring their reliability. By using block chain, the framework retains the history of modifications, preventing illegal data modification and enhancing data security. The framework includes simulation of a DT for an assembly line to demonstrate its viability. Reliable system specification

data from block chain is used to build process knowledge of DTs, ensuring trustworthiness and avoiding misuse cases. The TTS-CPS framework undergoes formal verification to validate its correctness and security properties. Bounded model checking and Satisfiability Modulo Theories Library are used to verify properties and ensure the integrity of the system. By integrating block chain with digital twins, the TTS-CPS framework enhances security by tracking modifications, ensuring data consistency, and preventing unauthorized access or modifications.

III. Initial Proposed Solution

The encryption scheme is a security technique where data is divided into fragments and encrypted with multiple keys. These encrypted fragments are then stored across geographically distributed locations. This approach offers several advantages:

1) **Enhanced Security:** Even if an attacker breaches one or more storage locations, they won't be able to decrypt the entire data set. They would need to access all the fragments and possess all the corresponding keys to decrypt the original data.
2) **Improved Tamper Detection:** Any attempt to modify data in one location would be easily detectable since the modification wouldn't match the corresponding fragments stored elsewhere.

Here's a breakdown of how block chain technology can be integrated with such encryption:

1) **Data Preparation:** The data to be stored on the block chain is first divided into smaller pieces (fragments). Each fragment is then encrypted with a unique key. These keys can be generated using various cryptographic algorithms.
2) **Key Sharing:** Each encryption key is further divided into multiple shares using a secret sharing scheme. This scheme ensures that a minimum number of key shares (e.g., 3 out of 5) are required to reconstruct the original key. These key shares are then stored on the blockchain itself.
3) **Fragment Distribution:** The encrypted data fragments are distributed across different storage locations. These locations could be on individual computers in a peer-to-peer network, cloud storage providers in various geographical regions, or even specialized hardware security modules.
4) **Block chain Integration:** The block chain acts as a tamper-proof ledger that keeps track of: The locations where the encrypted data fragments reside. The cryptographic hashes of each fragment (to ensure data integrity). The public keys used for encryption (so anyone can verify the data's authenticity). The information on how many key shares are needed and where they are stored on the block chain.
5) **Access Control:** To access the data, users would need to possess a sufficient number of key shares to reconstruct the original encryption keys. These key shares could be distributed using access control mechanisms built on the block chain itself.

IV. Methodology

The proposed block chain technique involves several intricate tasks: Accessing real-time digital twin data, this suggests the system deals with dynamic data that constantly updates, likely representing real-world processes or simulations. Utilizing multiple storage locations where data needs to be fragmented and stored across various cloud or physical storage units for security and potential redundancy. Multiple users are required to run and test the proposed block chain system, suggesting collaborative development or evaluation. To achieve the above scenario privacy and cost questions come into the loop. Most real-world data and storage resources are likely private, making it difficult to obtain access for testing or implementation. Utilizing multiple cloud storage providers or maintaining physical storage can be expensive, posing a financial barrier and time consuming as well. Figure 2 explain the overall block chain network.

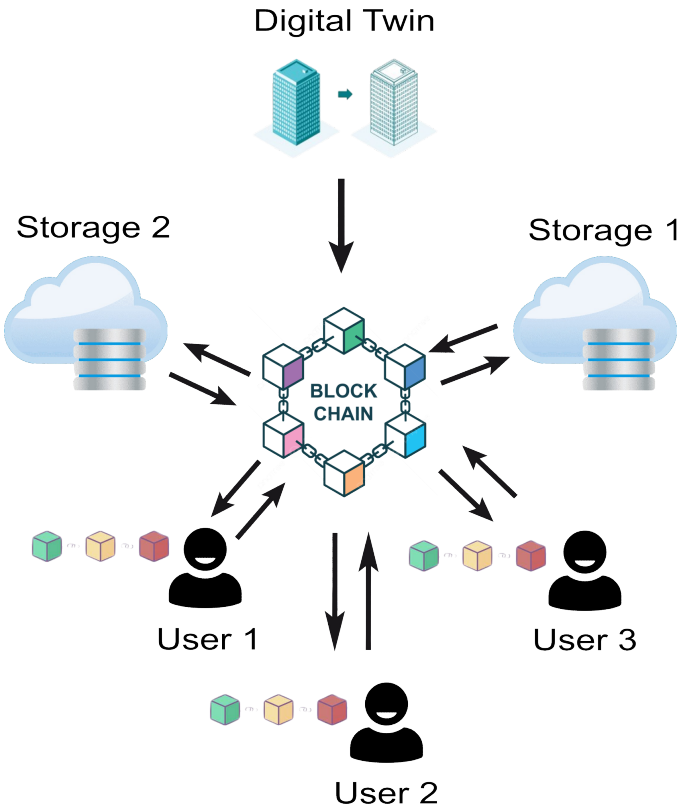

Fig. 2. Block Chain Security Network diagram

Due to the complexities and resource limitations, the development opted for a simulated version, user1, user2, digital twin, storage1 and storage2 are simulated, the simulation is built using Visual Studio Code (VS Code), a popular code editor that can be extended with various libraries. The specific libraries used provide functionalities for block chain simulation, data fragmentation, and potentially interaction with simulated storage systems. Table I list the libraries used to accomplish the task.

Methodology of the technique is divided into two parts, encryption and decryption. Following is the detailed step by step methodology for the proposed technique. For the sake

TABLE I
LIBRARIES

Name	Purpose
cryptography.fernet	Encrypts and Decrypts Data
secrets	Generates Random Secret Data
datetime	Handles Date and Time Functions
OS	Interacts with Operating System (file system)
random	Generates Random Numbers
json	Works with JSON data format
pycoin.Shamir	Implements Shamir's Secret Sharing

of explaining methodology its necessary to mention a digital twin scenario as blockchain technique is developed in context of digital twin. Let's consider a digital twin of a jet engine. This digital twin would contain a vast amount of data about the engine, including: Real-time readings from various sensors on the engine, such as temperature, pressure, vibration, and fuel flow. Historical data on engine performance metrics like thrust, fuel efficiency, and wear and tear. A detailed computer-aided design model of the engine. Information on past maintenance procedures performed on the engine.

A. Encryption

Step 1 - Data Fragment Generation:

- Sensor data is divided into time-based segments (e.g., minutes or hourly readings), Define the parameters for generating data fragments, such as the number of readings "numreadings" and time interval between readings "timeinterval".
- Function "createdatafragment(numreadings, timeinterval)" to generate simulated temperature readings at regular intervals. This function should return a string containing the formatted data fragment.

Step 2 - Data Encryption:

- Using "encryptdata(data)" function to encrypt the generated data fragment. Each data fragment is passed to the function for encryption process. Inside the function, generating a symmetric encryption key using "Fernet.generatekey()". Create a "Fernet" object with the generated key. Encrypt the data fragment using the "encrypt()" method of the "Fernet" object. The function should return the encrypted data and the encryption key. After encryption each data fragment is send to a random storage.
- Fernet, in the context you've encountered, refers to a specific Python library called "cryptography.fernet". It provides a way to encrypt and decrypt data using the Fernet symmetric encryption algorithm.

Step 3 - Key Sharing:

- Utilize the "splitkey(key, numshares, threshold)" function to shard the encryption key.
- Inside the function, Use the Shamir Secret Sharing scheme from the "pycoin" library to split the key into multiple shares. The function should return a list of key shares. Here, each key is divided into 3 shares.

Step 4 - Storage of Key Shares:

- Using the "storeshares(shares, folderpaths)" function to store the generated key shares.
- Inside the function, Iterate over the key shares and corresponding folder paths. Create folders if they don't exist and store each share in a separate folder.

Step 5 - Block chain Transaction Creation:

- Create a transaction object that contain three kind of information, which is the data fragment ID, key share, and folder location using the "createtransaction()" function. Each key share is processed with a separate transaction in block chain. The Figure 3 gives visual representation of encryption process.

B. Decryption

Step 1 - Blockchain Transaction Data Retrieval:

- To Retrieve the necessary transaction data related to the data fragment to be retrieved and key shares. Define the data fragment identifier "datafragmentidentifier" to specify the data fragment being decrypted. Get transactional data containing information about the data fragment ID and folder locations.

Step 2 - Key Reconstruction:

- To retrieve key shares to reconstruct the original encryption key the Shamir Secret Sharing scheme and the "combine()" method to reconstruct the key from the shares.

Step 3 - Encrypted Data Retrieval:

- Retrieve the encrypted data from storage based on the folder location obtained from transaction data. Read the encrypted data from the corresponding file.

Step 4 - Data Decryption:

- Using "decryptdatafragment(encrypteddata, key)" function to decrypt the encrypted data fragment.
- Inside the function, Create a "Fernet" object with the reconstructed key. Decrypt the encrypted data using the "decrypt()" method of the "Fernet" object. The function should return the decrypted data. The Figure 4 gives a visual representation of Decryption process.

This methodology provides a step-by-step guide for both encryption and decryption processes, including the functions and variables required at each stage. The encryption methodology begins by generating simulated data fragments containing temperature readings at regular intervals. These data fragments are then encrypted using a symmetric encryption key and Fernet encryption, ensuring data security. To enhance resilience and decentralization, the encryption key is shard into multiple shares using the Shamir Secret Sharing scheme. These key shares are stored in separate folders, with added random data for obfuscation and security purposes. Subsequently, blockchain transactions are simulated to store both the encrypted data and the key shares, facilitating decentralized storage and retrieval.

On the other hand, the decryption methodology involves retrieving blockchain data that contains crucial information regarding the encrypted data and its corresponding key shares.

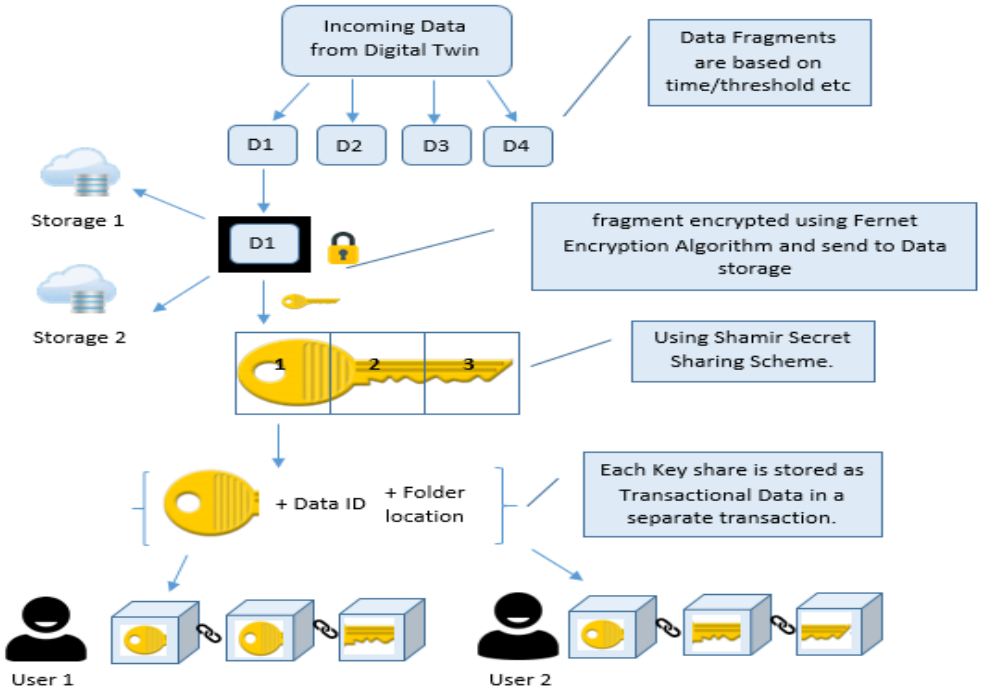

Fig. 3. Encryption Process Diagram

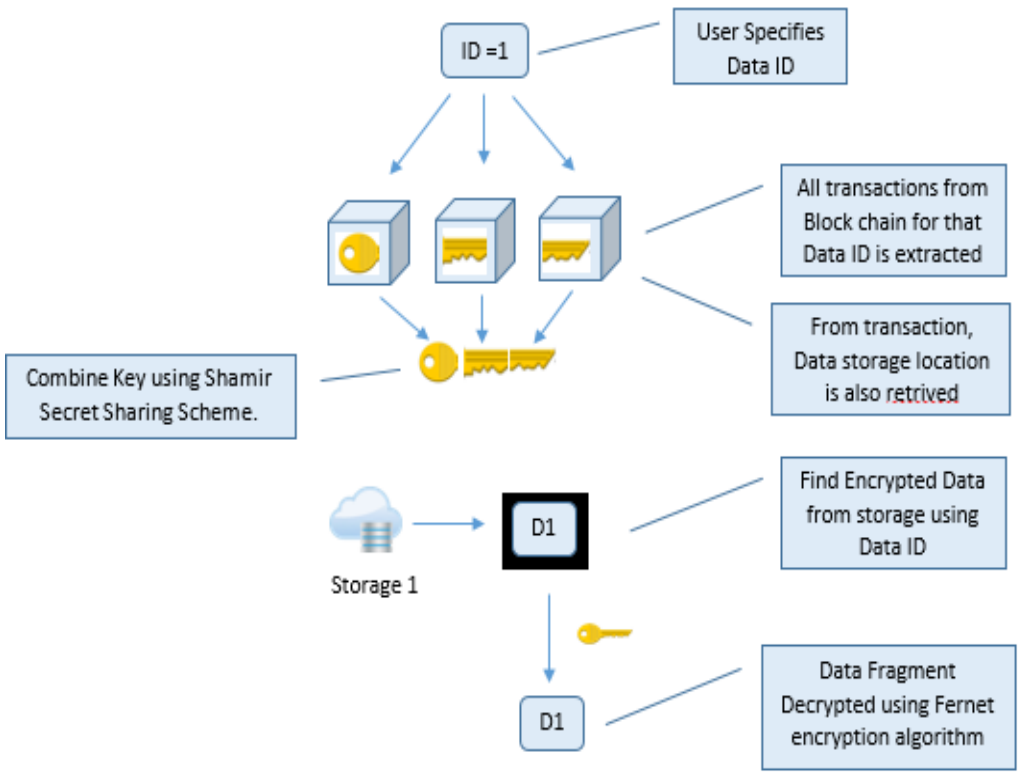

Fig. 4. Decryption Process Diagram

With this information, the original encryption key is reconstructed from the retrieved key shares using the Shamir Secret Sharing scheme. Following key reconstruction, the encrypted data is retrieved from storage based on the blockchain data, ensuring data integrity. The encrypted data fragment is then decrypted using the reconstructed key, thus recovering the original data. Finally, the decrypted data is presented or further processed for analysis or utilization, completing the decryption process.

V. Discussion

This section delves into the key advantages and considerations surrounding the proposed secure data sharing technique that leverages block chain technology and Shamir's Secret Sharing for key management.

A. Enhanced Security and Access Control:

- **Confidentiality through Encryption:** Data fragmentation and Fernet encryption ensure data remains unreadable by unauthorized parties. Fernet, a well-regarded symmetric encryption algorithm, provides a strong layer of protection for data fragments before they are distributed for storage.
- **Distributed Key Management:** Shamir's Secret Sharing fragments the encryption key generated by Fernet and distributes the shares across separate storage locations. This significantly increases the difficulty of unauthorized access compared to a single key storage approach.
- **Granular Access Control:** The threshold mechanism in Shamir's Secret Sharing allows for defining the minimum number of key shares required to reconstruct the decryption key. This enables granular control over who can access specific data fragments. For example, requiring all key shares for highly sensitive data ensures only authorized users with access to all share locations can decrypt it.

B. Improved Fault Tolerance and Efficiency:

- **Distributed Storage:** Storing key shares across multiple locations mitigates the risk of data loss due to hardware failures or security breaches in a single storage unit. Even if one location becomes unavailable, the data remains accessible as long as enough key shares are retrievable from other locations.
- **Potential for Scalability:** Fragmentation allows for efficient data storage and retrieval, particularly for large datasets. By distributing data fragments across multiple storage units, the system can potentially scale to accommodate increasing data volumes.

C. Considerations and Future Work:

- **Real-World Blockchain Integration:** The simulated blockchain environment provides a valuable proof-of-concept. However, integrating with a real blockchain network is crucial for enhanced security and scalability in practical applications. Real blockchain networks offer features like tamper-proof data storage, improved auditability, and potentially distributed consensus mechanisms for key management.
- **Performance Optimization:** While fragmentation offers scalability benefits, it's essential to investigate optimization techniques for data processing and retrieval. This could involve exploring different fragmentation strategies or optimizing communication protocols for efficient interaction with distributed storage locations.
- **Advanced Security Considerations:** While Shamir's Secret Sharing strengthens key management, additional security measures are necessary for comprehensive protection. Exploring user authentication and authorization mechanisms is vital for controlling access to the system and data fragments. Additionally, investigating potential vulnerabilities specific to the chosen real-world blockchain network would be crucial for a robust security posture.

VI. Conclusion

This paper investigated a novel approach for secure data sharing. It combined blockchain technology with Shamir's Secret Sharing for robust key management. Data fragmentation and Fernet encryption ensure data confidentiality, while distributed key storage with Shamir's Secret Sharing strengthens security even if some storage locations are compromised. The project utilized a simulated blockchain to showcase how transaction data could be stored and retrieved. Additionally, distributed storage across multiple locations improves fault tolerance. Overall, this approach demonstrates promise for secure data sharing with granular access control.

Looking ahead, the paper potential can be further explored by integrating the proposed approach with a real blockchain network. This would enhance security and scalability for real-world applications. Additionally, investigating optimizations for handling large datasets and exploring advanced security considerations like user authentication and authorization protocols would be valuable next steps. By addressing these future directions, this project's core concept can be developed into a robust and scalable solution for secure data sharing in various domains.

References

[1] A. A. Monrat, O. Schelén and K. Andersson, "A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities," IEEE Access, vol. 7, pp. 117134 - 117151, 2019. DOI: 10.1109/ACCESS.2019.2936094

[2] J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," in 2017 International Conference on I-SMAC, Palladam, 2017. DOI: 10.1109/I-SMAC.2017.8058363

[3] D. Dasgupta, J. M, Shrein and K. D. Gupta, "A survey of blockchain from security perspective," Journal of Banking and Financial Technology, vol. 3, pp. 1 - 17, 2019.

[4] C. Semeraro, M. Lezoche, H. Panetto and M. Dassisti, "Digital twin paradigm: A systematic literature review," Computers in Industry, vol. 130, 2021. https://doi.org/10.1016/j.compind.2021.103469

[5] E. Dawson and D. Donovan, "The breadth of Shamir's secret-sharing scheme," Computer and Security, vol. 13, no. 1, pp. 69-78, 1994. https://doi.org/10.1016/0167-4048(94)90097-3

[6] Sandeep Jain, "Geeksforgeeks," https://www.geeksforgeeks.org/how-does-the-blockchain-work/, accessed on [23/4/2024].

[7] J. Huang and J. Yi, "The key Security Management Scheme of Cloud Storage based on blockchain and digital twins," Huang and Yi Journal of Cloud Computing, vol. 13, no. 15, 2024.
[8] W. Shen, T. Hu, C. Zhang and S. Ma, "Secure sharing of big digital twin data for smart manufacturing based," Journal of Manufacturing Systems, pp. 338 - 350, 2021.
[9] Z. Lv, C. Cheng and H. Lv, "Blockchain based decentralized learning for security in digital twins," IEEE Internet of things Journal, vol. 10, no. 24, 2023.
[10] M. Ammi, S. Alarabi and E. Benkhelifa, "Customized blockchain-based architecture for secure smart home for lightweight IoT," Information Processing and Management, vol. 58, no. 3, 2021. https://doi.org/10.1016/j.ipm.2020.102482
[11] A. Thada, U. K. Kapur, S. Gazali, N. Sachdeva and S. Shridevi, "Custom Block Chain Based Cyber Physical System for Solid Waste Management," Procedia Computer Science, vol. 165, pp. 41-49, 2019. https://doi.org/10.1016/j.procs.2020.01.068
[12] S. Suhaila, S. U. R. Malik, R. Jurdak, R. Hussain, R. Matulevičius and D. Svetinovic, "Towards situational aware cyber-physical systems: A security-enhancing," Computers in Industry, 2022.