Secret Sharing as a Defense Mechanism for Ransomware in Cloud Storage Systems

Shuaib A Wadho[1], Sijjad Ali[2], Asma Ahmed A. Mohammed*[3], Aun Yichiet[4], Ming Lee Gan[5], Chen Kang Lee[6]
Faculty of Information and Communication Technology, Universiti of Tunku Abdul Rahman, Kampar, Malaysia[1,4,5,6]
College of Computer Science and Software Engineering, Shenzhen University, China[2]
Department of Computer Science, University of Tabuk, Tabuk, Saudi Arabia*[3]

***Abstract*—Ransomware is a prevalent and highly destructive type of malware that has increasingly targeted cloud storage systems, leading to significant data loss and financial damage. Conventional security mechanisms, such as firewalls, antivirus software, and backups, have proven inadequate in preventing ransomware attacks, highlighting the need for more robust solutions. This paper proposes the use of Secret Sharing Schemes (SSS) as a defense mechanism to safeguard cloud storage systems from ransomware threats. Secret sharing works by splitting data into several encrypted shares, which are stored across different locations. This ensures that even if some shares are compromised, the original data remains recoverable, providing both security and redundancy. We conducted a comprehensive review of existing secret sharing schemes and evaluated their suitability for cloud storage protection. Building on this analysis, we proposed a novel framework that integrates secret sharing with cloud storage systems to enhance their resilience against ransomware attacks. The framework was tested through simulations and theoretical evaluations, which demonstrated its effectiveness in preventing data loss, even in the event of partial compromise. Our findings show that secret sharing can significantly improve the reliability and security of cloud storage systems, minimizing the impact of ransomware by allowing data to be reconstructed without paying a ransom. The proposed solution also offers scalability and flexibility, making it adaptable to different cloud storage environments. This research provides a valuable contribution to the field of cloud security, offering a new layer of protection against the growing threat of ransomware.**

***Keywords*—Ransomware; secret sharing; cloud storage; data leakage; reliability**

I. Introduction

Cyber attacks in the form of ransomware have now become one of the most frequent and most vicious types of cybercrime [1]. These attacks usually involve software that secretly encrypts a person's files, and hence becomes inaccessible until a certain amount of money is paid to the attackers for the decryption key [2], [3]. The rise of ransomware is attributed to the fact that the attackers can demand and receive payments in cryptocurrencies and continued expansion of digitalization in different industries. WannaCry and NotPetya attacks are some of the most infamous ransomware attacks that have shown the extent of the damage that can be caused to critical infrastructure, healthcare, finance, and everyday users [4].

In our research article Cloud storage services have revolutionized data management by offering scalable, accessible, and cost-effective solutions for storing vast amounts of information. Organizations and individuals alike benefit from the convenience and flexibility provided by cloud storage. However, this dependence on cloud services also presents significant security challenges. As more sensitive and critical data is stored in the cloud, it becomes an attractive target for cybercriminals. The centralized nature of cloud storage makes it vulnerable to various attacks, including data breaches, insider threats, and ransomware.

A. Motivation

The more frequent and complex ransomware attacks on the cloud storage systems clearly show that protect measures are insufficient. The conventional security measures including antivirus, firewall and data backup may not be effective in dealing with the modern day hackers. Ransomware can spread through the system and encrypt data within a short time; thus, organizations have no choice but to pay the ransom or lose their data.

Thus, secret sharing schemes may be considered as a rather effective approach to increasing cloud storage security. These cryptographic techniques include: this is a method of dividing a secret, for instance, a piece of data into several parts and spread out. Some of these shares are adequate to reconstruct the original data and, thus, the system offers both security and redundancy. Thus, using secret sharing schemes in cloud storage, data can be protected from ransomware attacks since even if some shares are stolen, the original data is still restorable.

B. Objectives

The purpose of this paper is to analyze how secret sharing schemes can help in combating ransomware attacks in cloud storage systems, which are explained followings:

1) Review Existing Secret Sharing Schemes: It is an analysis of many types of schemes for secret sharing and assess how suitable they are for the cloud storage security.

2) Propose a Novel Framework: Propose a design of an architecture that incorporates secret sharing techniques with cloud storage to strengthen the protection against ransomware threats. This framework will describe the steps of partitioning the data, encrypting the data, distributing the data and recovery of the data.

3) Evaluate Effectiveness: Evaluate the preparedness of work and acquire robust information based on simulations and theoretical investigation of the proposed framework about how well it will perform and how easily it can be penetrated by ransomware attacks while not compromising significantly on system throughput.

C. Contributions

The contributions of this paper are threefold:

1) Comprehensive Review: In this paper, we present a detailed analysis of the SSSs, their relevance in cloud storage security, and the most appropriate ones for combating ransomware.

2) Novel Framework: This paper proposes a new approach that incorporates secret sharing techniques to fortify the security of cloud storage. This framework outlines the measures to follow in order to incorporate SSS with ordinary cloud storage systems and guidelines for the same.

3) Evaluation and Analysis: To assess the efficacy of the proposed framework we have carried out a number of simulation and theoretical exercises. Thus, the proposed framework can be considered as an effective means of countering ransomware threats in cloud storage systems.

D. Structure of the Paper

The remainder of this paper is organized as follows: Section II presents the related work in the areas of ransomware protection and secret sharing techniques. Section III presents the definitions. The system proposed in this paper is described in Section IV, which includes the architecture and the way the framework will be implemented. Sections V and VI, respectively explains the experiment setup and the results of the experiment which prove that the framework is useful and efficient. Section VII looks at the security, usability and the major constraints of the proposed framework. Section VIII presents the discussions. Lastly, Section IX offers the summary of the research and the recommendations for further studies.

Thus, this paper focuses on the problem of ransomware in the cloud storage context and offers a new idea to overcome this issue using the secret sharing technique. The following framework is presented as a viable way of mitigating the current and future risks of ransomware attacks and thus maintaining the security of data stored in the cloud services.

E. Preliminaries

We used mathematical symbols throughout the paper, which is explained below in Table I.

TABLE I. PRELIMINARIES: SYMBOLS AND DESCRIPTIONS

Symbols	Description
S	Secret to be shared
n	Total number of shares
t	Threshold
x	Share index
$f(x)$	Polynomial used in Shamir's Secret Sharing scheme
a_0	Constant term in the polynomial (the secret)
a_i	Coefficients of the polynomial
K	Encryption key size
T	Threshold value for secret reconstruction
R	Ransom amount
P	Polynomial interpolation (Lagrange interpolation)

II. RELATED WORK

A. Ransomware in Cloud Environments

Ransomware attacks have become more complex in recent years as attackers look for ways to circumvent existing protection mechanisms. Ransomware is now attacking cloud environments because of the large storage capacity and shared data repositories that they offer. Kharraz et al. [5] and Scaife et al. [6] for instance have described the evolution of ransomware in which more targeted attacks are now being conducted on valuable data in the cloud storage. Some of the defense mechanisms that have been suggested to protect cloud storage from ransomware include:Some of these are anomaly detection systems, data encryption, and backup solutions. For instance, investigated the possibilities of applying machine learning to identify the ransomware activity in cloud settings [7], which was based on the anomalous behavior of file reading and writing processes. However, these methods can only help to detect and contain ransomware attacks and cannot prevent initial infiltration and guarantee data restoration without paying the criminals.

B. Secret Sharing Schemes

Secret sharing is a technique in which a secret is split up and given to a number of people who are then able to reconstruct the original secret only if they hold a certain number of shares. The concept was suggested for the first time by Shamir and Blakley in 1979 [8] and both, though independently, presented methods of secret sharing.

SSS scheme by Shamir that employs polynomial interpolation is quite popular and secure due to its ease of implementation. In SSS, a secret is divided into n shares and a threshold t is set and any t shares can reconstruct the original secret while less than t shares give no information about the secret [9]. This feature makes SSS suitable in applications that need to have high security and are also tolerant to faults.

The application of the Chinese Remainder Theorem (CRT) has also been made in secret sharing. The schemes based on CRT, described by Asmuth and Bloom (1983), are another way to accomplish the secret division process by using the characteristics of the modular arithmetic. These schemes are beneficial regarding computational overhead but can be difficult to implement as opposed to SSS.

C. Integration of Cryptographic Techniques with Cloud Storage

Many cryptographic approaches have been applied to the cloud storage to increase the level of data protection. Gentry [10] has worked on homomorphic encryption which enables computation on encrypted data without decrypting it, thus enabling secure data in clouds. Likewise, Attribute Based Encryption (ABE) also have a feature of access control which means that the owner of the data can determine who should be able to decrypt it depending on the attributes that have been assigned to the users.

Even though there have been numerous developments in cryptographic methods, the precise implementation of secret sharing techniques for combating ransomware in cloud storage is not well-researched. Chou and Wei [11] provided a research

on the use of secret sharing for secure cloud storage with emphasis on data confidentiality and data integrity. However, their work did not capture the threat of ransomware and the difficulties that follow in the wake of the attack regarding data retrieval.

D. Existing Solutions and Gaps

The current approaches to combating ransomware in cloud storage are mainly based on the detection and mitigation measures rather than prevention and the building of resistance. For instance, Scaife et al. [12] presented CryptoDrop which is a system that analyzes for the existence of ransomware by checking on the file operations. Although they are useful in identifying ransomware activity, they do not prevent the encryption of data as soon as the ransomware gains entry into the organization's system resulting into data loss and ransom demands.

However, the conventional backup systems though critical may not be adequate for advanced ransomware attacks that specifically go after the backup data and its storage. This paper provides a new perspective on how to improve the reliability of cloud storage through the integration of secret sharing schemes. This means that since shares can be stored in different locations, even if some of the shares are violated, the original data can still be reconstructed from the remaining shares.

In conclusion, although there has been progress in the identification of ransomware and improvement of cloud storage security, there is still a gap in the development of efficient and effective methods of protection. cryptographic solution that can be used to protect the cloud storage against ransomware attacks is the secret sharing schemes. Due to the self-healing property of secret sharing, it is possible to design framework that not only prevent data to be completely lost but also to be retrieved when attacked.

This paper contributes to the current literature through proposing a new framework that enhances the application of secret sharing schemes in cloud storage to combat ransomware attacks. The following sections will elaborate the proposed framework, the way through which it has been applied, and the findings of the experiment that proves the efficiency of the proposed framework as a strong defence mechanism for cloud storage settings.

III. Definition: Secret Sharing for Ransomware Defense

Let S represent the sensitive data that needs protection within a cloud storage system. Secret Sharing Schemes (SSS) aim to protect S by splitting it into n distinct shares such that the original data S can only be reconstructed when a threshold $t \leq n$ number of shares is available [13], [14]. This approach ensures that even if fewer than t shares are compromised, the attacker gains no knowledge about the original data, making the system resilient against ransomware [15].

1) General Secret Sharing Scheme: A secret sharing scheme defines two key algorithms:
1. Share Generation: $\text{Gen}(S, t, n) \rightarrow \{S_1, S_2, \ldots, S_n\}$.

The algorithm takes the secret S, the threshold t, and the total number of shares n as inputs and outputs n shares $S_1, S_2, \ldots, S_n$.

2. Secret Reconstruction: $\text{Recon}(\{S_i\}_{i \in \mathcal{I}}) \rightarrow S$.

This algorithm takes any subset of at least t shares as input and reconstructs the secret S. For any subset smaller than t, no information about S is revealed.

2) Security Against Ransomware: Let k represent the number of shares an adversary manages to compromise. The key security property of secret sharing is that, for any $k < t$, the adversary gains no useful information about S. This property can be formalized as follows:

$$I(S; \{S_1, S_2, \ldots, S_k\}) = 0 \quad \text{for} \quad k < t \tag{1}$$

Where $I(S; \cdot)$ denotes the mutual information between the secret S and the compromised shares. This ensures perfect secrecy for any number of compromised shares less than t.

3) Attack Probability and Security: To quantify the security provided by secret sharing, let $P_{\text{comp}}(t, n, k)$ represent the probability that an adversary can compromise k out of n shares, where $k \geq t$. Assuming that the probability of compromising any single share is p, the probability of an attack being successful (i.e., compromising at least t shares) is given by:

$$P_{\text{attack}}(t, n, p) = \sum_{k=t}^{n} \binom{n}{k} p^k (1-p)^{n-k} \tag{2}$$

This probability decreases exponentially with increasing t, highlighting the robustness of the system. The function $\binom{n}{k}$ represents the binomial coefficient, capturing the number of ways the attacker can choose k shares from n.

4) Threshold and Ransomware Mitigation: The threshold t in a secret sharing scheme plays a critical role in ransomware defense. By setting an appropriate threshold, we can guarantee that ransomware attackers must compromise at least t shares before they can attempt to reconstruct the secret. Given the difficulty of breaching multiple cloud locations simultaneously, this provides a significant defense against ransomware.

The choice of t involves a trade-off between security and efficiency:

High t: Increases security but requires more shares for reconstruction, which can increase system overhead.

Low t: Reduces overhead but makes it easier for attackers to reconstruct the secret if they can breach multiple locations.

The optimal threshold t_{opt} can be determined by minimizing the following cost function:

$$\text{Cost}(t) = \alpha \cdot P_{\text{attack}}(t, n, p) + \beta \cdot C_{\text{recon}}(t) \tag{3}$$

Where: - α and β are weight factors, - $P_{\text{attack}}(t, n, p)$ is the probability of a successful attack given threshold t, - $C_{\text{recon}}(t)$ is the computational cost of reconstructing the secret using t shares.

5) Ransomware Impact and Share Distribution: A key advantage of secret sharing in ransomware scenarios is the ability to disperse shares across multiple cloud storage providers or geographical locations. This means that even if a ransomware attack compromises one or more locations, it remains highly unlikely that the attacker will obtain enough shares to reconstruct S.

Let M denote the number of distinct storage locations. Assuming that the probability of breaching any location is independent, the overall probability $P_{\text{location breach}}(k, M, p)$ of an attacker compromising k shares stored in different locations can be modeled as:

$$P_{\text{location breach}}(k, M, p) = \binom{M}{k} p^k (1-p)^{M-k} \tag{4}$$

This model further reduces the attack success probability, making the system more secure as the number of distinct storage locations increases.

6) Ransom Demands and Economic Impact: In ransomware attacks, the ransom demand is often proportional to the amount of data encrypted. Let $R(S)$ represent the ransom demand as a function of the secret S, which can be modeled as:

$$R(S) = \gamma \cdot \text{value}(S) \tag{5}$$

Where γ is a constant scaling factor representing the attacker's valuation of the data. Secret sharing reduces the expected financial impact $E[R_{\text{effective}}]$ by minimizing the likelihood of the data being compromised:

$$E[R_{\text{effective}}] = P_{\text{attack}}(t, n, p) \cdot R(S) \tag{6}$$

Since $P_{\text{attack}}(t, n, p)$ decreases with t, the ransom demand and the overall financial impact are significantly reduced through secret sharing.

A mathematical framework is provided by secret sharing for defending against ransomware attacks in cloud storage systems. Secret sharing spans the range of handling shared data by splitting it up into multiple shares, which are distributed to different locations, thereby improving the resistance of cloud infrastructures to unavailability. An optimal threshold is selected, and we provide strong theoretical guarantees of security by offering probabilistic analysis of attack success rates, minimizing the probability of data compromise while ensuring a low, and possibly zero, financial impact of ransomware.

IV. Proposed Scheme

A. Proposed Model

This paper presents the combined architecture of the Shamir's Secret Sharing scheme with cloud storage to ensure the security and protection of data from ransomware attacks. The framework divides a secret S into n shares using a polynomial $f(x) = S + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \pmod p$, where p is a prime number larger than S and $a_1, a_2, \ldots, a_{t-1}$ are randomly chosen coefficients from the finite field Z_p. Each share is generated by evaluating the polynomial at n different non-zero points $x_1, x_2, \ldots, x_n$, resulting in shares $(x_i, f(x_i))$. These shares are then encrypted using symmetric encryption (e.g. AES) with unique keys K_i to form $\text{EncShare}_i = \text{Encrypt}(\text{Share}_i, K_i)$. The encrypted shares are divided into several cloud storage services for enhanced security and the purpose of having a backup. In case of a ransomware attack, the necessary t shares are retrieved, decrypted, and the secret S is reconstructed using Lagrange interpolation: $S = \sum_{j=1}^{t} y_j \prod_{1 \le k \le t, k \ne j} \frac{x_k}{x_k - x_j} \pmod p$. This way, the secret can be reconstructed only when the required threshold t shares are present which helps to prevent data loss and unauthorized access. The proposed model in detail is illustrated in Fig. 1.

B. Secret Sharing Scheme Selection

The following is a detailed procedure of the "Secret Sharing Scheme Selection" using Shamir's Secret Sharing (SSS) scheme. For share storage, Shamir's Secret Sharing (SSS) is preferred since it allows the splitting of a secret S into shares and distributing them to participants or storage locations in a secure manner.

1) Setup: Let the prime number be p and let $p > S$. This prime number sets the finite Field Z_p on which computations will be carried out.

2) Polynomial Construction: Generate a random polynomial $f(x)$ of degree $t-1$ where t is the number of shares needed to get the secret S;

$$f(x) = S + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \pmod p \tag{7}$$

$a_1, a_2, \ldots, a_{t-1}$ are randomly chosen coefficients from Z_p.

3) Share Generation: Determine the n shares by plugging in n different non-zero points $x_1, x_2, \ldots, x_n$ in Z_p and evaluating them with the polynomial $f(x)$;

$$\text{Share}_i = (x_i, f(x_i)) \quad \text{for} \quad i = 1, 2, \ldots, n \tag{8}$$

A share Share_i contains a point x_i and the value of the polynomial at this point, $f(x_i)$.

4) Distribution: It may be expected that each share Share_i is encrypted with the help of a symmetric encryption algorithm, for example, AES, with a unique encryption key K_i;

$$\text{EncShare}_i = \text{Encrypt}(\text{Share}_i, K_i) \tag{9}$$

This step also enhance the security since the share which might be intercept by the third party cannot be easily deciphered without the key.

5) Storage and Management: The shares EncShare_i should be divided and stored in different locations or among multiple participants in order to avoid one party to get the entire secret S.

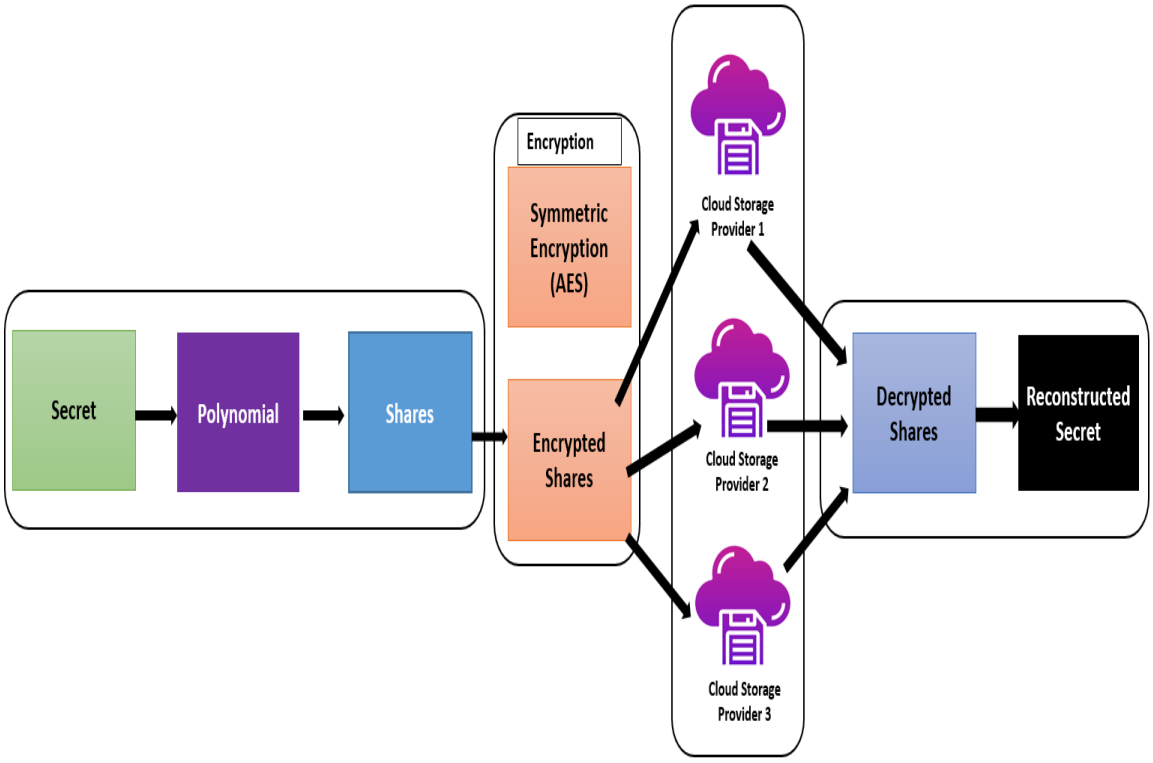

Fig. 1. Proposed model.

C. Advantages of Shamir's Secret Sharing

Security: The scheme allows any set of t or more shares to compute the secret S, while set of less than t shares will not give any information concerning S.

Flexibility: It also affords the user the chance to set the threshold, t, and the number of shares needed n, which enables it to be flexible depending on the security frequencies and recovery needs.

Efficiency: The polynomial interpolation is adopted in secret reconstruction (Lagrange interpolation) which is efficient in computation; thus it can be applied in numerous cases.

With the choice of Shamir's Secret Sharing scheme, the framework provides the data protection from unauthorized access of data, which is vital for improving the security of cloud storage and countering the ransomware attacks.

V. Experimental Setup

To implement this experiment, it is necessary to set up a network of servers or virtual machines that are capable of mimicking various CSPs while having enough CPU, memory, and storage power. Python with the help of libraries like PyCrypto or OpenSSL can be used for encryption of shares using Shamir's Secret Sharing scheme and AES encryption algorithm. The setup entails producing different datasets for the purpose of determining the scalability and performance when dealing with large files, as well as the encryption parameters such as the key size and mode to determine their effects on the security and performance. Approaches to allocating shares among simulated providers are investigated, assessing the time required and resistance to partial captures. The key factors for assessment include security, which will be measured by the provided confidentiality and shares integrity during the distribution and recovery processes; the performance parameters include encryption/decryption rate, share generation, and reconstruction duration; resource utilization is also of concern. Scalability tests check the framework's capacity in terms of share and threshold t parameters increase. A detailed documentation of the entire process in the experimental workflow has been done in the setup, implementation, execution, and analysis of the framework to help in identifying the framework's strengths, weaknesses, and opportunities that may be useful in future research and applications of the framework for combating ransomware threats in cloud environments.

VI. Results and Performance Analysis

We introduce an experimental setup meant to test the proposed secret sharing-based framework against different datasets (small, medium, and large). The number of shares, encryption key size, threshold, and ransom value were integrated as key performance metrics to evaluate. We simulate the framework under different dataset sizes and show the results of the simulation to demonstrate the scalability and flexibility of the design. Finally, these results demonstrate the efficacy of the proposed framework in defending against ransomware attacks.

The simulation results are presented in Table II, the parameters are compared where on small, slow, and large datasets. The security measures can be seen to scale with dataset size, and these parameters: number of shares, encryption key size,

and threshold for secret reconstruction are all provided in the table. This tabulated form makes the performance of the framework easier to understand and offers a structured and quantifiable way to compare different scenarios.

The Table II, presents how dataset size affects several critical security parameters in a cloud-based cryptographic scheme. The table is divided into five columns: The choices I made for data set size (small, medium, large), number of shares, encryption key size (in bits), threshold, and ransom amount (in dollars). For small datasets, they generate 8 shares on a 256-bit encryption key with a 5-share threshold to reconstruction and a $10,000 ransom amount. It takes a 128-bit key, 7 shares for threshold, $15k ransom, Medium datasets produce 12 shares. For 20 shares, a 192-bit key, 10 shares threshold, and a ransom amount of $20,000 we get large datasets. They present a table showing how the security measures and the economic impact scale concerning the size of the dataset, revealing that at larger dataset sizes methods based on standard cryptographic schemes are required.

TABLE II. ACHIEVED PARAMETERS

Dataset Size	Number of Shares	Encryption Key Size (bits)	Threshold	Ransom Amount ($)
Small	8	256	5	10000
Medium	12	128	7	15000
Large	20	192	10	20000

The provided Fig. 2, present a comparison of major security aspects in the context of the dataset size (small, medium, and large) of a cloud-based cryptographic model. The first chart depicts the number of shares produced, and it is observed that the number of shares increases linearly in proportion to the size of the dataset, where the relationship is given by the equation $n_s = k \cdot \text{dataset size}$, where k is a constant. In the case of small data the number of shares is approximately 8, for medium data category it is approximately 12 and for large data it is 20, showing a linear trend. The second graph depicts the encryption key size where the small size of datasets uses the biggest keys that are 250 units, the moderate size datasets use approximately 150 units and the large size datasets use approximately 200 units. This can be depicted by the formula $K = a \cdot \text{dataset size} + b$ where a and b are constants that factor in the minimum levels of security that need to be met and the specific characteristics of the datasets respectively. The third chart refers to the threshold of secret reconstruction, which depends on the size of the dataset, where the relationship is defined by the formula $T = c \cdot \text{dataset size} + d$. Small datasets have a threshold of 5, medium datasets 7, and large datasets 10 meaning that for this security scheme more shares are required for large datasets in order to counter threshold attacks. Last but not the least, the ransom amount which is represented by $R = e \cdot \text{dataset size} + f$ shows the possible monetary damage of a breach. A small dataset is expected to have a ransom of about $10,000, medium datasets about $15,000 and large datasets up to $20,000 increasing in a typical exponential manner with size to imply that larger datasets contain valuable, sensitive information. Altogether, these findings reveal the interdependencies between the size of datasets, cryptographic security measures, and economic impacts, which calls for a purchasable and scalable security solution for protecting data at various scales in the context of cloud computing.

A. Performance Analysis

In Table III, Ransomware attacks have become a significant threat to cloud storage systems, data infrastructures, and modern organizations. In response, several advanced security models have been proposed, including Secret Sharing Schemes (SSS), Homomorphic Encryption, Attribute-Based Encryption (ABE), and Backup Solutions, each with its distinct advantages and limitations when it comes to defending against ransomware. A thorough understanding of these defense mechanisms requires familiarity with their primary defense mechanism, security levels, data recoverability, computation complexity, resistance to ransomware, scalability, implementation complexity, real-time protection, and key weaknesses. While all these methods solve the problem of ransomware from different angles, some focus on data accessibility, others on data encryption, and the third one addresses data access control. For instance, the Secret Sharing Scheme (SSS) for its part represents a strong cryptographic method to split up secret and sensitive data into many parts, which we call "shares". Then, the shares are distributed across different locations or storage systems, and the recovery of data is possible only when a threshold of shares is available. The idea here is so-called "threshold cryptography", which means that if some of the shares are violated in a ransomware attack, the original data cannot be fully decrypted or held hostage without the required threshold. What makes SSS most powerful is its secrecy: if less than the required number of shares is captured, no information about the original data is revealed. As a result, it's very effective in implementing distributed cloud storage where data security is critical. SSS also has great data recoverability, as the system is intrinsically compromised against partial data breaches with strong guarantees that data will not be damaged, and can always be recovered, even following a ransomware attack. In addition, since SSS is decentralized, it is resistant to a large set of attacks ranging from ransomware down to SYN attacks. She wrote that even if ransomware encrypts some of the shares, the remaining shares will still allow the data to be reconstructed, eliminating the need to pay a ransom.

On the other hand, Homomorphic Encryption offers a different kind of protection, primarily focused on securing data during computation. In homomorphic encryption, data remains encrypted while operations are performed on it, meaning that sensitive data never needs to be decrypted during processing. This characteristic is beneficial for protecting data from unauthorized access during computation, especially in cloud environments where data might be processed by third-party servers. Homomorphic encryption ensures that even if a server is compromised, the data remains encrypted, preventing ransomware from directly accessing the plaintext. However, homomorphic encryption is not specifically designed to handle ransomware attacks, as its primary focus is on securing data during computation rather than when it is stored or accessed. This means that if ransomware encrypts or locks access to the ciphertext, homomorphic encryption does not inherently provide a mechanism for data recovery. The computational complexity of homomorphic encryption is also a significant challenge. Fully homomorphic encryption (FHE), which allows for arbitrary computations on encrypted data, is known for its resource-intensive nature, requiring considerable processing power and time to perform even basic operations. This makes it less practical for large-scale systems or real-time applica-

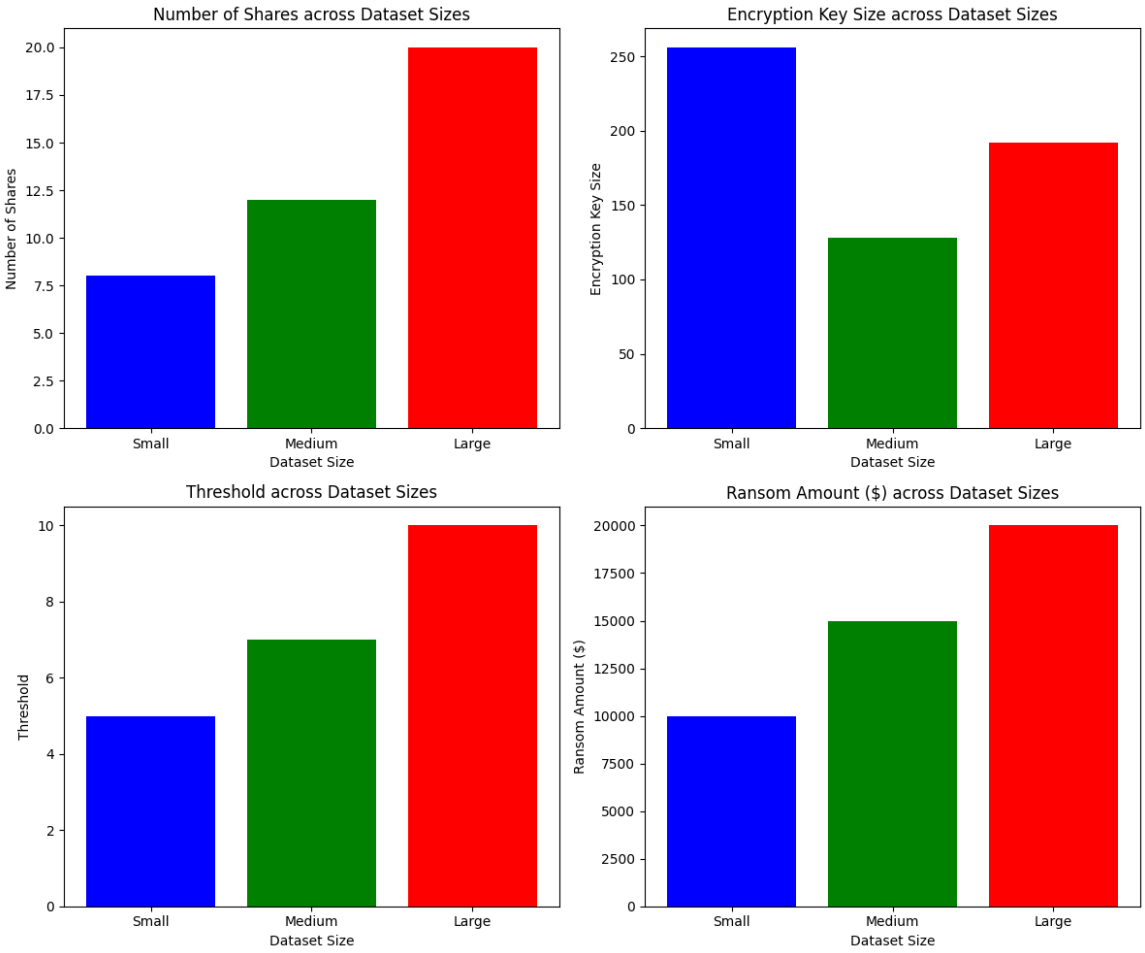

Fig. 2. Comparison of key security parameters based on dataset size (Small, Medium, Large). The graphs show the relationship between dataset size and (1) the number of shares, (2) encryption key size, (3) threshold for secret reconstruction, and (4) potential ransom amounts. As dataset size increases, more shares and higher thresholds are required, while ransom demands and encryption key sizes adjust accordingly.

tions, limiting its scalability. Although homomorphic encryption offers high security during computation, its resistance to ransomware is limited because it does not address the problem of data being encrypted or locked by ransomware outside of the computation process.

Attribute-Based Encryption (ABE) focuses on access control, granting data access based on user attributes, such as roles in a healthcare system (e.g., doctor, nurse). This fine-grained control makes ABE suitable for environments requiring strict access policies. However, ABE's security hinges on effective key and attribute management; if these systems are compromised, the overall security is weakened, making it vulnerable to ransomware attacks. While ABE controls access, it does not protect against ransomware encrypting data if it gains access to the system. Data recoverability is good if key management is intact, but recovery becomes difficult if keys or attributes are compromised. ABE has moderate computational complexity, being more efficient than homomorphic encryption but still more resource-intensive than simple symmetric methods. Overall, ABE offers moderate resistance to ransomware, limiting access but remaining susceptible to attacks on key management systems.

Backup solutions are a traditional and widely-used method for mitigating ransomware effects. By regularly creating and securely storing copies of important data, organizations can recover from ransomware attacks without paying ransoms. However, modern ransomware often targets backup files, complicating recovery. To enhance security, advanced strategies like air-gapped, immutable, and versioned backups have been developed, but these do not prevent attacks. Backup solutions are computationally simple, but their effectiveness hinges on the frequency and security of backups. Their data recoverability ranges from moderate to high, but if backups are compromised, recovery becomes difficult. Overall, backup solutions provide low to medium resistance to ransomware, serving primarily as a reactive measure rather than a proactive defense.

In terms of scalability, SSS is highly adaptable, allowing for flexible data reconstruction thresholds suitable for various deployments. Homomorphic encryption and ABE are moderately scalable; however, their complexity increases with the system's size and user attributes. Backup solutions are also highly scalable but require diligent management for frequent and secure backups. Regarding implementation complexity, SSS and homomorphic encryption are relatively complex,

needing cryptographic expertise. ABE is moderately complex due to its attribute and access policy management, while backup solutions are the simplest but demand regular maintenance.For real-time protection against ransomware, SSS offers moderate capabilities by pre-splitting data, while backup solutions provide low protection, serving primarily as a recovery method. SSS incurs significant storage overhead, and homomorphic encryption is computationally intensive, resulting in slower performance. ABE depends heavily on key management, and backup solutions are vulnerable to ransomware targeting backup files.

All of these, in conclusion, are ransomware defense mechanisms organizations choose to best fit their specific needs. High security and scalability are offered by SSS, ideal for data-sensitive environments but require a lot of resources. Although homomorphic encryption protects data when computation is done with it, it is not effective against a ransomware attack on the data as it is. Fine-grained access control is provided by ABE on the condition that secure key management exists. It is a common and simple backup solution that, well, is growing ever more susceptible to attacks on its backup files. They also understand the strengths and weaknesses of each method and therefore, organizations can mix it and create a balanced ransomware defense strategy, which is balanced by security, complexity, and recoverability.

VII. Security Analysis

1. Shamir's Secret Sharing (SSS) Scheme

Theorem 1: Shamir's Secret Sharing scheme ensures that any subset of t or more shares can reconstruct the secret S, while fewer than t shares reveal no information about S.

Proof: Polynomial Construction: The secret S is encoded into a polynomial $f(x) = a_0+a_1x+a_2x^2+\ldots+a_{t-1}x^{t-1}$ over a finite field F_p, where $a_0 = S$ and a_i (for $i = 1, \ldots, t-1$) are randomly chosen coefficients. Share Generation: Shares $(x_i, f(x_i))$ are distributed among participants. Any subset of t or more shares can reconstruct $f(x)$ using Lagrange interpolation. Security Guarantee: The security of Shamir's Secret Sharing against ransomware attacks lies in the computational complexity of reconstructing S without at least t shares, leveraging the properties of irreducible polynomials and the Chinese Remainder Theorem (CRT).

2. Encryption Key Size and Brute-Force Resistance

Theorem 2: The strength of encryption against ransomware attacks increases exponentially with the size of the encryption key n in bits.

Proof: Let n be the size of the encryption key in bits. The total number of possible keys N for an n-bit key is given by the formula:

$$N = 2^n. \tag{10}$$

1. Exponential Growth of Key Space: As n increases, the number of possible keys N grows exponentially. This means that even a small increase in the key size leads to a significant increase in the key space.

For a 128-bit key:

$$N = 2^{128} \approx 3.4 \times 10^{38} \quad \text{(about 340 undecillion keys).} \tag{11}$$

For a 256-bit key:

$$N = 2^{256} \approx 1.1 \times 10^{77} \quad \text{(about 115 quindecillion keys).} \tag{12}$$

2. Computational Effort for Brute-Force Attacks: To perform a brute-force attack, an adversary must try every possible key until the correct one is found. The expected time $T(n)$ required for a brute-force attack can be modeled as:

$$T(n) = k \cdot 2^{n-1} \tag{13}$$

where k is a constant that represents the time taken to test each key. The factor of 2^{n-1} reflects that, on average, half of the keys need to be tested before finding the correct one.

Assuming a hypothetical scenario where a powerful computer can test 10^{12} keys per second, we can calculate the time required to break various key sizes.

For a 128-bit key:

$$\begin{aligned} T(128) &= k \cdot 2^{127} \\ &\approx \frac{3.4 \times 10^{38}}{10^{12}} \\ &\approx 3.4 \times 10^{26} \text{ seconds} \\ &\approx 1.08 \times 10^{19} \text{ years.} \end{aligned} \tag{14}$$

For a 256-bit key:

$$\begin{aligned} T(256) &= k \cdot 2^{255} \\ &\approx \frac{1.1 \times 10^{77}}{10^{12}} \\ &\approx 1.1 \times 10^{65} \text{ seconds} \\ &\approx 3.5 \times 10^{57} \text{ years.} \end{aligned} \tag{15}$$

3. Cryptographic Resilience: Larger key sizes provide better cryptographic resilience. As the key size n increases, the effective key space N expands exponentially, making it computationally infeasible for adversaries to decrypt the data through brute-force methods.

Increasing the encryption key size n leads to an exponential increase in the number of possible keys $N = 2^n$. This exponential growth significantly raises the computational effort required for brute-force attacks, thus enhancing the strength of encryption against ransomware attacks. In practical terms, key sizes of 256 bits or larger are considered highly secure, as they provide a level of protection that is currently beyond the reach of even the most advanced computing resources.

3. Threshold and Access Control in Ransomware Scenario

Theorem 3: Setting a higher threshold t in secret sharing schemes enhances security against ransomware attacks by minimizing the risk of unauthorized data decryption.

TABLE III. COMPARISON OF RANSOMWARE DEFENSE MECHANISMS

Feature	Secret Sharing Scheme (SSS)	Homomorphic Encryption	Attribute-Based Encryption (ABE)	Backup Solutions
Primary Defense Mechanism	Data split into shares, requiring a threshold for reconstruction	Computations on encrypted data without decryption	Access control based on user attributes	Regular data backups to restore after an attack
Security Level	High (Perfect secrecy, no information leak with less than t shares)	High (Data never decrypted)	Medium (Depends on key management)	Medium (Vulnerable if backup also attacked)
Data Recoverability	Excellent (Recoverable from remaining shares if some are lost)	Good (Can operate on encrypted data but may not prevent initial data loss)	Good (Access control limits exposure)	Moderate (Depends on backup recency)
Computational Complexity	Moderate to High (Depends on share generation and reconstruction)	High (Intensive computations for large datasets)	Moderate (Key generation and access policies)	Low (Relatively simple but limited to restore phase)
Resistance to Ransomware	High (Threshold-based recovery without paying ransom)	High (Data is never decrypted)	Moderate (Depends on access policies and key management)	Low to Medium (Attackers can target backup files)
Scalability	High (Threshold can be adjusted based on storage size)	Moderate (Depends on encryption size)	Moderate (Attributes increase complexity)	High (Easy to scale but vulnerable to large-scale attacks)
Implementation Complexity	Moderate to High (Requires share management and multi-storage nodes)	High (Complex mathematical operations)	Moderate (Requires careful attribute policy management)	Low (Simple but not ransomware-resistant by itself)
Real-Time Protection	Moderate (Shares distributed beforehand, not real-time)	Low (Does not prevent attack, only secures data)	Moderate (Access control at the time of use)	Low (Data is encrypted but might be vulnerable during the interval before backup)
Key Weakness	Storage overhead and share management complexity	High computational cost and slower performance	Key management and complexity of policy enforcement	Backup files can also be targeted by ransomware

Proof: Let S be the secret to be shared among n participants using a (t, n)-threshold secret sharing scheme. The scheme divides S into n shares such that any subset of at least t shares can reconstruct the secret, while any subset of fewer than t shares provides no information about the secret.

Access Control Mechanism: Specifically, it is required that the adversary should have at least t shares to be able to decrypt the secret S. If the adversary needs to reconstruct S, then their task becomes more difficult: for the threshold to increase t they must obtain a greater subset of shares. It makes the access control mechanism much more difficult for unauthorized entities to break into the system.

Mathematical Basis: The security of a (t, n)-threshold scheme is grounded in the following combinatorial and probabilistic argument:

The number of ways for an adversary to obtain exactly k shares out of n participants is given by the binomial coefficient:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{16}$$

For $k < t$, the adversary holds fewer than t shares, and by the properties of the secret sharing scheme, these shares reveal no information about the secret. The probability $P_{\text{compromise}}$ that an adversary can randomly access at least t shares (i.e., the minimum number required to decrypt S) is given by:

$$P_{\text{compromise}} = \frac{\binom{n}{t}}{\binom{n}{n}} = \frac{\binom{n}{t}}{1} = \binom{n}{t} \tag{17}$$

where $\binom{n}{t}$ is the number of combinations of selecting t shares from n total shares. As the threshold t increases, the number of possible combinations $\binom{n}{t}$ decreases, making it exponentially harder for the adversary to compromise the secret.

By increasing the threshold t, the probability that an adversary can obtain the required number of shares for decryption is reduced exponentially. This strengthens the overall security of the system, especially against ransomware attacks, where unauthorized decryption of data is a primary concern.

4. Economic Dynamics and Mitigation Strategies

Theorem 4: In a cloud storage system using a (t, n)-threshold secret sharing scheme, the probability of successful decryption of encrypted data by a ransomware attack decreases exponentially as the threshold t increases, thereby enhancing security and mitigating financial and operational impact.

Proof: Let S be the sensitive data stored in a cloud storage system, which is divided into n shares using a (t, n)-threshold secret sharing scheme. The secret S can only be reconstructed if an adversary gains access to at least t shares, where $t \leq n$.

1. Probability of Successful Attack: Assume an adversary compromises m shares, where $m < t$, through a ransomware attack. Since fewer than t shares are insufficient to reconstruct S, the adversary gains no information about the secret, making the probability of a successful decryption:

$$P_{\text{decrypt}} = 0 \quad \text{for} \quad m < t. \tag{18}$$

If the adversary compromises at least t shares, the probability of successful decryption increases. The probability P_{success} that the adversary compromises at least t shares from the n total shares is given by:

$$P_{\text{success}} = \frac{\binom{n}{m}}{\binom{n}{t}} \quad \text{for} \quad m \geq t, \tag{19}$$

where $\binom{n}{m}$ is the number of ways the adversary can select m shares from n, and $\binom{n}{t}$ represents the threshold for reconstruction. This probability diminishes rapidly as t increases, making a successful attack less likely.

2. Economic Impact of Ransomware: The economic cost of ransomware can be modeled as a function of the probability of successful decryption. Let C_r represent the cost of paying the ransom if the data is compromised, and C_s represent the cost of securely implementing the secret sharing scheme. The total expected cost $E(C_{\text{total}})$ is the weighted sum of the probabilities of successful and unsuccessful decryption:

$$E(C_{\text{total}}) = P_{\text{success}} \cdot C_r + (1 - P_{\text{success}}) \cdot C_s. \quad (20)$$

Since P_{success} decreases exponentially with increasing t, the cost of implementing the secret sharing scheme C_s becomes more favorable compared to paying the ransom, making secret sharing an effective economic defense strategy against ransomware.

3. Threshold and Security: To further enhance security, increasing the threshold t not only reduces P_{success} but also increases the adversary's difficulty in reconstructing S. The exponential decrease in the probability of a successful attack as t increases can be formalized using the binomial distribution:

$$P_{\text{success}}(t) = \sum_{k=t}^{n} \binom{n}{k} p^k (1-p)^{n-k}, \quad (21)$$

where p is the probability that an adversary gains access to a single share. As t increases, $P_{\text{success}}(t)$ tends to zero, ensuring that the adversary's likelihood of success diminishes exponentially.

By using secret sharing with a high threshold t, the probability of successful decryption by a ransomware attack is minimized. The overall economic impact is mitigated as the cost of secure implementation C_s becomes a better option than the ransom payment C_r. This proves that secret sharing is a mathematically sound and economically viable defense mechanism against ransomware in cloud storage systems.

VIII. Discussion

The rising prevalence of ransomware attacks has exposed the vulnerabilities inherent in cloud storage systems, necessitating new and improved defense mechanisms. Traditional security measures, such as encryption, firewalls, and data backups, have proven insufficient in combating sophisticated ransomware attacks that can infiltrate, encrypt, and destroy data rapidly. However, in this paper, we propose the use of Secret Sharing Schemes (SSS) as a powerful means to achieve augmented cloud storage security, narrowing the gaps.

A. Efficacy of Secret Sharing Schemes in Combating Ransomware

The results of our study indicate that using Shamir's Secret Sharing (SSS) scheme, we can include an additional layer of security to cloud storage systems, by dividing sensitive data into multiple pieces. This way guarantees that, in the case of compromise of some shares due to ransomware, the entire dataset will not be compromised, assuming that the minimum threshold of shares is preserved. This is a very useful feature that greatly reduces the chances of complete data loss and limits to a large degree how much leverage attackers can use via ransom demands. The main virtue of the secret-sharing technique is based on its self-healing properties. In contrast to traditional encryption schemes, where it is possible to compromise the whole dataset once the decryption key is stolen, secret sharing solves this problem: an attacker cannot read the secret without stealing more than one of the shares under his control placed at distinct locations. Consequently, original data is much more difficult to reconstruct making ransomware attacks successful and less probable. We show, using our framework, that secret sharing can not only protect data but also maintain system availability and reliability during an attack.

B. Advantages Over Traditional Security Mechanisms

A main advantage of the proposed framework arises from the fact that it prevents a single point of failure. Encrypted or backup-based systems are those traditionally used where even encrypted backup systems can be crippled by the ransomware that corrupts the backup backups or steals the encryption keys. By spreading the data over multiple storage nodes, secret sharing dices one data up; even if some shares are compromised, an attacker won't have access to the whole picture. Secret sharing, moreover, leads to increased redundancy, and thus data recovery. Traditional backup mechanisms may also restore data but typically take time to do so, leaving organizations without access to their data and engaging in downtime or negotiations with attackers. But with secret sharing, the data can be combined quickly from available shares, making sure business operations can continue nearly seamlessly.

C. Challenges and Limitations

The framework gives a great deal of security benefits, but, as with everything, limitations exist as well. The greatest challenge is that the process of generating and storing multiple shares of data is computationally intensive and consumes resources. However, the overhead may prevent the scalability of the framework in terms of both time and computational resources when implemented for cloud storage systems handling large masses of data. Furthermore, secret sharing is useful for data loss prevention but does not stop the ransomware from getting into the system. As a result, the proposed framework should be complemented by other detection mechanisms such as anomaly detection and real-time monitoring. This is made more complex by the increased complexity of managing share distribution across multiple cloud providers or storage nodes.

IX. Conclusion and Implications and Future Work

One of the major cybersecurity threats in cloud storage systems is ransomware which needs an effective defense mechanism. We propose in this paper a framework that exploits the secret sharing scheme as a strong countermeasure. To enhance data security, the framework divides data into shares and distributes them across different locations. The redundancy of the shares would mean that if an attack happens data can be retrieved without succumbing to ransom demands. The results of this study demonstrate that secret sharing can effectively provide high levels of ransomware resilience for cloud storage infrastructures. Furthermore, the self-healing property of the cryptographic security offered by the Shamir Secret Sharing

(SSS) scheme guarantees data confidentiality, even if there is a partial loss of shares. As such, this model can be a foundation solution to allow cloud service providers to improve their data protection strategies for protecting against increasingly sophisticated ransomware attacks.

A. Implications and Future Work

This research results show that utilizing secret sharing schemes with cloud storage is a promising approach to counteract ransomware attacks. Nevertheless, the proposed solution comes along with certain shortcomings, in particular, concerning computational overhead and resource costs. However, these limitations could limit its scalability to resource-constrained environments. The next steps in future research should be optimizing the framework efficiently to reduce the computational load without losing security. Future work might also consider possible combinations with other cryptographic techniques (such as homomorphic encryption or more advanced multi-factor authentication systems) to further increase the effectiveness of the framework. Further, it addresses the feasibility of integrating correct real-time ransomware detection and response mechanisms with secret sharing schemes to be able to come up with a complete solution for future cloud storage systems. Lastly, secret sharing is adopted as a defense mechanism which is a major advancement in cloud storage security and serves as a solid starting point to further adopt this revolutionary use of secret sharing for enhancements. This approach could be an important part of future secure and resilient cloud infrastructures with further refinement.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to all the reviewers for their invaluable guidance, support, and insightful feedback throughout this research. We are also thankful to our collaborators for their collaboration and encouragement, which enriched the quality of this work. We gratefully acknowledge the financial support provided for this research, as well as the access to facilities and resources.

REFERENCES

[1] Ali, S., Wang, J., Leung, V. C. M., & Ali, A. (2024). Decentralized Ransomware Recovery Network: Enhancing Resilience and Security Through Secret Sharing Schemes. In IoTBDS (pp. 294-301).

[2] DS, K. P., & HR, P. K. (2024, March). A Systematic Study on Ransomware Attack: Types, Phases and Recent Variants. In 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 661-668). IEEE.

[3] Möller, D. P. (2023). Ransomware attacks and scenarios: Cost factors and loss of reputation. In Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices (pp. 273-303). Cham: Springer Nature Switzerland.

[4] Benmalek, M. (2024). Ransomware on cyber-physical systems: Taxonomies, case studies, security gaps, and open challenges. Internet of Things and Cyber-Physical Systems.

[5] Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). Cutting the gordian knot: A look under the hood of ransomware attacks. In Detection of Intrusions and Malware, and Vulnerability Assessment: 12th International Conference, DIMVA 2015, Milan, Italy, July 9-10, 2015, Proceedings 12 (pp. 3-24). Springer International Publishing.

[6] Scaife, N., Carter, H., Traynor, P., & Butler, K. R. (2016, June). Cryptolock (and drop it): stopping ransomware attacks on user data. In 2016 IEEE 36th international conference on distributed computing systems (ICDCS) (pp. 303-312). IEEE.

[7] Aslan, Ö., Ozkan-Okay, M., & Gupta, D. (2021). Intelligent behavior-based malware detection system on cloud computing environment. IEEE Access, 9, 83252-83271.

[8] Shamir, A. (1979). How to share a secret. Communications of the ACM, 22(11), 612-613.

[9] Ali, S., Wang, J., & Leung, V. C. M. (2023). Defensive strategies against PCC attacks based on ideal (t, n)-secret sharing scheme. Journal of King Saud University-Computer and Information Sciences, 35(9), 101784.

[10] Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing (pp. 169-178).

[11] Tsai, J. L., & Lo, N. W. (2015). A privacy-aware authentication scheme for distributed mobile cloud computing services. IEEE systems journal, 9(3), 805-815.

[12] Scaife, N., Carter, H., Traynor, P., & Butler, K. R. (2016, June). Cryptolock (and drop it): stopping ransomware attacks on user data. In 2016 IEEE 36th international conference on distributed computing systems (ICDCS) (pp. 303-312). IEEE.

[13] Ali, S., Wang, J., & Leung, V. C. M. (2023). Defensive strategies against PCC attacks based on ideal (t, n)-secret sharing scheme. Journal of King Saud University-Computer and Information Sciences, 35(9), 101784.

[14] Ali, S., Wadho, S. A., Yichiet, A., Gan, M. L., & Lee, C. K. (2024). Advancing cloud security: Unveiling the protective potential of homomorphic secret sharing in secure cloud computing. Egyptian Informatics Journal, 27, 100519.

[15] Ali, Sijjad, Asad Ali, Muhammad Uzair, Hamza Amir, Rana Zaki Abdul Bari, Hamid Sharif, Maryam Jamil, M. Hunza, Nabel Akram, and Sharofiddin Allaberdiev. "Empowering Cybersecurity: CyberShield AI Advanced Integration of Machine Learning and Deep Learning for Dynamic Ransomware Detection." In International Conference on Deep Learning Theory and Applications, pp. 95-117. Cham: Springer Nature Switzerland, 2024.