Secure Replica Allocation in Cloud Storage Systems with Heterogeneous Vulnerabilities

Yun Tian
Department of Computer Science
California State University, Fullerton
Fullerton, CA 92834-6870
Email: ytian@fullerton.edu

Xiao Qin
Department of Computer Science
and Software Engineering
Auburn University
Auburn, Alabama 36849-5347
http://www.eng.auburn.edu/∼xqin

Yafei Jia
IBM Research - China
Beijing, China
Email: jybj@cn.ibm.com

***Abstract*—Highly available cloud storage is often implemented with complex, multi-tiered distributed systems built on top of clusters of commodity servers and disk drives. Storage reliability, security and performance are among the top desired features when clients consider storing data on cloud storage. Although replication improves reliability and performance in cloud storage systems, data replication increases the risk of data storage in an insecure network environment. When a cloud storage scales up, storage nodes are very likely to become heterogeneous in nature. In this study, we propose a secure replica allocation scheme called SecRA to improve security, reliability, and performance of a cloud storage system where storage nodes have a wide variety of vulnerabilities. Our SecRA integrates the techniques of replication and fragmentation with secret sharing in a heterogeneous cloud system, where storage nodes are comprised of various server types in terms of vulnerability characteristics. SecRA allocates data replicas of fragments of a file to as many different types of nodes as possible. For the replicas of the same fragment, SecRA tries to allocate these replicas to the same type of nodes in the system. Data assurance is significantly improved, because the replicas of different fragments of a file are allocated to multiple types of storage nodes. To quantitatively evaluate the quality of security offered by SecRA, we develop a storage assurance model. Our analytically results show that replica allocations made by SecRA lead to enhanced security thanks to the consideration of heterogeneous vulnerabilities in cloud storage systems.**

I. Introduction

Currently, cloud computing is becoming more and more popular in a lot of application fields both in academic and industry such as bioinformatics, weather forecasting, social networks and other web-based applications. In the upcoming big data era, more and more data is easier to collect and much of the more sensitive data needs to be stored securely and accessed easily.

A. Reliability is Highly Demanded

Reliability is a storage system property that is both highly desired and yet minimally implemented. When some storage nodes in a cloud storage system are down, the entire system might grind to a halt [24]. Downtime is clearly very expensive; for example, in the on-line business world, millions of dollars per hour are lost when systems become unreliable. Replication techniques are commonly employed to enhance the reliability of data-intensive services.

Replication degree indicates the number of replicas maintained for each file/fragment in a replication mechanism. Files may have different access rates; the locations of their guests may vary greatly different. If there are too many replicas existing in a cloud storage system, the storage space will be wasted. The storage servers need extra time to search the corresponding replicas of a file. Thus storage system reliability is increased at the cost of degraded system performance. To achieve better reliability and system performance, storing different number of replicas for different files may be a better solution than storing the same number of replicas for different pieces of data. There are also some scenarios exist that most of the files have similar access rates and their users' geographic locations are also similar. In such cases, there is no need to vary the replication degree of the stored files.

B. Data Replication Increases Risk

If a cloud storage system only stores one copy for each fragment, it will be not that reliable. Data replications will increase reliability. However, it may impose security risks due to an increasing number of file fragments handled by cloud storage servers. A file is more likely to be compromised when more replications of the file are stored in a cloud storage system especially in a risky network environment.

Evidence in [20] shows that heterogeneous features can be used to improve storage security for non-replicated data. In this study, the impact of heterogeneous features on security is investigated for replicated data storage in cloud storage systems. To simplify the study, we only investigate the scenario when all fragments of a file have the same replication degree (the same number of replicas).

C. Heterogeneous Features among Storage Nodes

One of the advantages for cloud computing systems is that they can be easily scaled up by adding more nodes to the systems as requirements on processing power increase. Large data and high demand drive the sizes of cloud storage systems to increase very quickly and to become more and more scalable and heterogeneous. When the sizes of cloud storage systems grow, the heterogeneous features also grow such as available bandwidths, processor speeds, disk capacities,

978-1-4673-7891-8/15/$31.00 ©2015 IEEE

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

lifetimes, potential weaknesses, security levels, failure rates, and pattern of failures among the storage nodes and network condition. On the other hand, heterogeneous features of different applications that run on such systems are increasing at the same time. The data for different applications may have different sizes, access rates, and different levels of security and performance requirements. We believe that future security mechanisms for cloud storage systems must be aware of heterogeneous vulnerabilities.

D. Fragmentation Technique

A fragmentation technique partitions a security sensitive file into multiple fragments that are distributed across different storage servers in a cloud storage system. A lot of fragmentation schemes have been proven to be valuable tools to improve security of data stored in cloud storage systems (see, for example, [21][9]). Many fragmentation approaches aim to improve reliability and performance of cloud storage systems by applying data replication methods. For example, Dabek *et al.* developed a wide-area cooperative storage system in which they implemented a fragmentation scheme to improve reliability and facilitate load balancing [1].

In real-world cloud storage systems, the fragmentation technique is usually combined with replication to achieve better performance at the cost of increased security risk to the data stored in the systems. A practical cloud storage system normally contains multiple heterogeneous servers providing services with various vulnerabilities. Unfortunately, the existing fragmentation and replication solutions do not take heterogeneity issues into account.

E. Main Contributions

This paper focuses on how to evaluate and reduce the security risk caused by data replication in heterogeneous cloud storage systems. We proposed a secure replica allocation scheme for heterogeneous cloud storage systems called SecRA. Data fragmentation and secret sharing techniques are used to improve data security in the proposed SecRA scheme. The storage nodes are categorized into different groups based on their heterogeneous vulnerabilities. The SecRA scheme tries to distribute the replicas of a fragment into the same group of storage nodes which share similar vulnerabilities; for the replicas of different fragments of a file the SecRA scheme tries to scatter them into different groups of storage nodes which have different set of vulnerabilities. We built an assurance model to evaluate the SecRA scheme for replicated data storage in heterogeneous cloud storage systems. Experimental results show that increasing heterogeneity levels can improve data assurance in cloud storage systems where data replication is implemented.

- We address the heterogeneous vulnerability issue by logically categorizing storage nodes of a cloud storage system into different server-type groups based on their vulnerabilities. Each server-type group contains storage nodes with the similar sets of security vulnerabilities.
- We propose a secure replica allocation scheme called SecRA to improve security of a cloud storage system where storage nodes have a wide variety of vulnerabilities.
- To quantify information assurance and to evaluate the proposed SecRA scheme, we develop a storage assurance model.
- We discover principles to improve assurance levels of heterogeneous cloud storage systems where data replication is implemented to improve reliability and performance. These principles are general guidelines to help designers achieve a secure replica allocation solution for cloud storage systems.

F. The Organization of this Paper

The rest of the paper is organized as follows: In Section II, we review previous work related to this study and summarize the preliminary results. Section III presents the system model and a motivation example. Section IV describes SecRA - a secure fragment replica allocation scheme. In Section V, we develop an assurance model to evaluate the SecRA scheme. In Sections VI, we quantitatively evaluate the proposed SecRA scheme in the context of cloud storage systems. Section VII summarizes this paper and outlines our future work.

II. Prelimilaries

A. Heterogeneous Vulnerabilities

Heterogeneous cloud storage systems have been applied to security sensitive application, such as banking systems and digital government, which require new approaches to security [25]. There are a lot of factors that affect cloud storage system security both in hardware and software [13]. The traditional security techniques for cloud storage systems include access control, security threat detection, authentication, authorization, and fault tolerance et al.. The confidentiality of security-sensitive files must be preserved in modern cloud storage systems, because cloud storage systems are exposed to an increasing number of attacks from malicious users [15].

Although there exist many security techniques and mechanisms, it is quite challenging to secure data stored in a cloud storage system. In general, security mechanisms need to be built for each component in a cloud storage system. Then a secure way of integrating all the components in the system must be implemented. It is critical and important to maintain the confidentiality of files stored in a cloud storage system when malicious programs and users compromise some storage nodes in the system.

In a large-scale cloud storage system, different storage nodes have a variety of ways to protect data. The same security policy may be implemented in various mechanisms. Data encryption schemes may vary; even with the same encryption scheme, key lengths may vary across the cloud storage systems. The above mentioned factors can contribute to different vulnerabilities among storage nodes. Although security mechanisms deployed in multiple storage nodes can be implemented in a homogeneous way, different vulnerabilities may exist due to heterogeneities in computational units.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

There have been numerous amounts of work and research done on cloud storage system security [8] [5] [7]. However, little attention has been paid to security solutions designed for cloud storage systems with data replication by making use of the heterogeneous vulnerabilities. This problem motivates us to focus on heterogeneity issues concerning security mechanisms of cloud storage systems where data replication is implemented.

B. Secret sharing

Secret sharing–independently invented by Shamir and Blakley–is a method of distributing a secret among a group of participants, each of which is allocated a share of the secret. The secret can be successfully reconstructed only when a sufficient number of shares are collected and combined [14][16]. In addition to cryptographic systems, secret sharing is an approach to providing data confidentiality by distributing a file among a group of n storage nodes, to each of which a fragment of the file is allocated. The file can be reconstructed only when a sufficient number (e.g., more than k) of the fragments are available to legitimate users for the (k, n) secret sharing scheme. Attackers are unable to reconstruct a file using the compromised fragments, if a group of servers are compromised and fewer than k fragments are disclosed.

The secret sharing scheme has been extended and employed in different application domains [17]. For example, Bigrigg *et al.* proposed an architecture called PASIS for secure storage systems. The PASIS architecture integrates the secret sharing scheme with information dispersal to improve security, integrity and reliability [23][26]. In a storage system with PASIS, even if an attacker compromises a limited (i.e., fewer than the threshold) subsets of storage nodes, the confidentiality of data stored in the system is still preserved. The aforementioned secret-sharing solutions designed for cloud storage systems ignore the issue of heterogeneous vulnerabilities. This fact motivates us to extend the secret sharing scheme by considering heterogeneity of vulnerabilities in the context of cloud storage systems where data replication is implemented.

C. Preliminaries of S-FAS to Improve Assurance for Non-replicated Data

In [20], a secure fragment allocation scheme (S-FAS) was proposed to improve storage assurance. Storage nodes in a cloud storage systems are classified into a variety of different server-type groups based on vulnerability characteristics. Given a file and a cloud storage system, S-FAS allocates fragments of the file to as many different types of nodes as possible in the system. Data confidentiality is preserved because fragments of a file are allocated to multiple types of storage nodes. Analysis results show that the more heterogeneous of the storage nodes, the higher of the storage assurance can be achieved by the proposed S-FAS scheme.

If all storage nodes in the evaluated cloud storage system are identical in terms of vulnerability, the probability that fragments of a file can be compromised using one successful attack method is 1. Fig. 1 shows the impact of K, the number of storage types, on system assurance. Results plotted in Fig. 1 suggest that for a cloud storage system with homogeneous vulnerability, threshold m has no impact on system assurance. When it comes to a cloud storage system with heterogeneous vulnerabilities, the system assurance increases significantly with the increasing of K and threshold m (see Fig. 1). Such a trend implies that a high heterogeneity level of vulnerability gives rise to high confidentiality assurance.

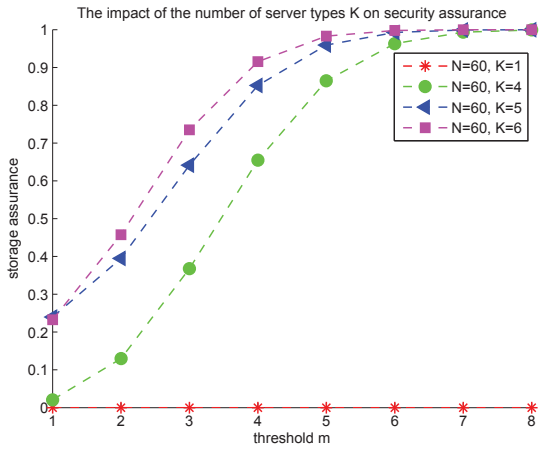

Fig. 1. *Heterogeneous system and homogeneous system using secret sharing scheme [20].* In all the four test cases, N is set to 60. K is set to 1, 4, 5, and 6, respectively. When K is 1, there is only one server group in the system.

However, in the S-FAS scheme, data replication was not considered. In this study we extend the S-FAS scheme by considering data replication to improve data reliability and system performance. In the next section, we will describe the system model and give a detailed motivation example to illustrate in what condition and how the proposed scheme SecRA can be utilized to enhance data assurance and system reliability.

III. System Model and Motivational Examples

A. System Model

The SecRA scheme is designed for a cloud storage system where each storage node is a cluster storage subsystem. Different cluster storage subsystems may be connected within some subnetworks to form a larger scale cloud storage sysytem.

Fig. 2 depicts a cluster storage subsystem, which consists of a number of storage nodes and a gateway. Considering heterogeneous vulnerability in large-scale storage systems, we classify storage nodes into different server-type groups.

B. Replication Scheme for Heterogeneous Cloud storage Systems

Replication scheme is commonly employed to enhance the reliability of cloud storage systems.There may be two cases, where the fragments of files are duplicated.

- Case 1: Each file fragment has the same number t of replicas.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

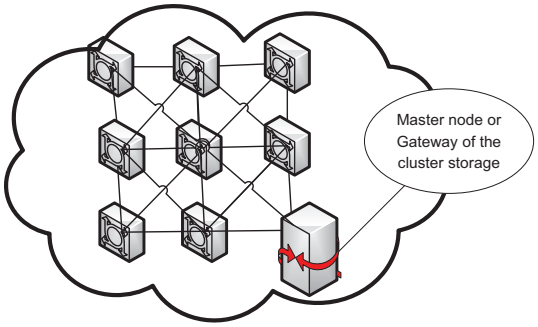

Fig. 2. A cluster storage subsystem consists of a number of storage nodes and a gateway. Storage nodes with the similar vulnerabilities are categorized into the same server-type group.

- Case 2: File fragments have various number (i.e., t_1, t_2, ..., t_x) copies of replicas.
  To simplify the analysis in this study, we provide the study result of the first case when each file fragment has the same number of replicas.

The appropriate replica placements are of growing importance to improve reliability and efficiency. Appropriate replica placements can be used to balance resource consumption for the underlying infrastructure. Both static and dynamic replica placement strategies have been investigated in the past [19] [10]. Many factors (e.g., system components, system topology, application types, clients' distribution and access patterns, et al.) have impacts on the efficiency of replica placements [4] [11]. We mainly study the impact on reliability and security caused by heterogeneous vulnabilities of a cloud storage system with replication technique.

C. A Motivation Example

Study in [20] shows that the security of the stored files can be improved by making use of the heterogeneous strategies for security on different storage nodes. In this study we analyze how a fragment replication scheme may increase storage risk by using random or hashing method replica placement in heterogeneous cloud storage systems. The following motivation example lays out the foundation for us to develop the secure replica allocation scheme.

We consider a heterogeneous cloud storage system (see Fig. 3) that contains 25 storage nodes categorized into 5 server-type groups (or server groups for short), i.e., T_1, T_2, T_3, T_4, and T_5. Storage nodes in each server group offer similar services with the same set of vulnerabilities. In this example, server group T_1 consists of nodes $r_1, r_6, r_{11}, r_{16}, r_{21}$, i.e., $T_1 = \{r_1, r_6, r_{11}, r_{16}, r_{21}\}$. Similarly, we define the other four server groups as: $T_2 = \{r_2, r_7, r_{12}, r_{17}, r_{22}\}$, $T_3 = \{r_3, r_8, r_{13}, r_{18}, r_{23}\}$, $T_4 = \{r_4, r_9, r_{14}, r_{19}, r_{24}\}$, and $T_5 = \{r_5, r_{10}, r_{15}, r_{20}, r_{25}\}$.

We assume that a file F is divided into three fragments and each fragment has two copies (i.e., f_{a1}, f_{a2}, f_{b1}, f_{b2}, f_{c1}, f_{c2}). Fig. 4 shows the replica placement decisions that do not take heterogeneous vulnerabilities into account. The decision made using a hashing function (see Eq. 11 in [12]) randomly allocates the six fragments of file F to six different nodes, each of which belongs to one of the six server sets illustrated in Fig. 4. For example, the six fragment replicas f_{a1}, f_{a2}, f_{b1}, f_{b2}, f_{c1} and f_{c2} are stored on nodes r_2, r_7, r_4, r_9, r_{18}, and r_{23}, respectively. Here r_2 and r_7 belong to storage node group 2; r_4 and r_9 belong to storage node group 4; and r_{18} and r_{23} belong to storage node group 3. This replica placement happens to be a good decision, because the replicas of a certain fragment are distributed to the storage nodes within the same storage node group. At the same time, different fragments' replicas are distributed to different storage-type groups which have various vulnerabilities. A malicious user must launch three successful attacks (one for each server group) in order to compromise all the three fragments.

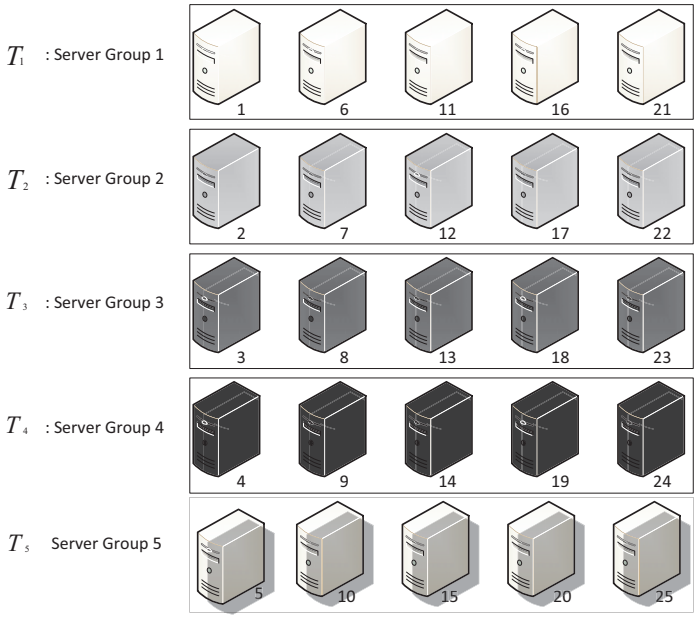

Fig. 3. *A cloud storage system* containing 25 storage nodes and the nodes are categorized into 5 server-type groups (or server groups for short), i.e., T_1, T_2, T_3, T_4, and T_5. Servers in each group have the same level of security vulnerability.

On the other hand, let us consider a case where the six fragment replicas f_{a1}, f_{a2}, f_{b1}, f_{b2}, f_{c1} and f_{c2} are stored on nodes r_{19}, r_{14}, r_9, r_4, r_5, and r_{24} respectively. Once a malicious user successfully launch server group 4, the replicas f_{a1}, f_{a2}, f_{b1}, f_{b2}, and f_{c2} can be compromised and thus the file F is accessible by the hacker. This is because r_{19}, r_{14}, r_9, r_4, and r_{24} belong to the same storage node group 4.

Except in diverse security vulnerabilities, storage nodes with different hardware configurations have different limitations, lifetimes, and environmental challenge tolerance. Distributing the replicas of a fragment into heterogeneous storage nodes tends to increase reliability, because this approach reduces the possibility that the heterogeneous nodes collapse at the same time due to some environmental causes.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

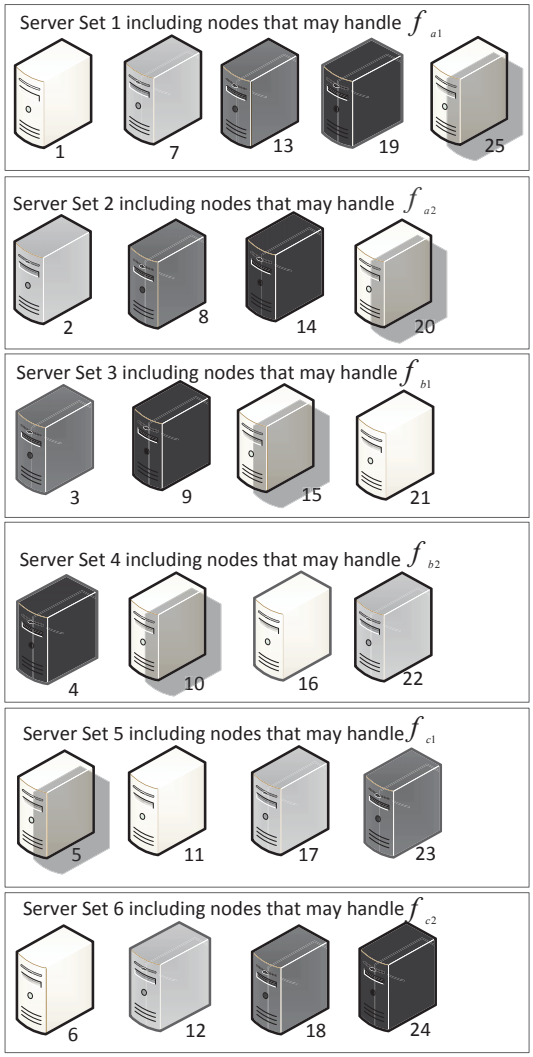

Fig. 4. *Possible insecure file fragment allocation*

Fig. 4 shows the possible insecure file fragment allocation decision made by a hashing function(see Eq. 11 in [16]). In this placement decisions, replica f_{a1} could be distributed into Server set 1, replica f_{a2} could be distributed into Server set 2, Replica f_{b1} could be distributed into Server set 3, Replica f_{b2} could be distributed into Server set 4, Replica f_{c1} could be distributed into Server set 5, Replica f_{c2} could be distributed into Server set 6. Server set 1 contains storage nodes r_1, r_6, r_{11}, r_{16}, and r_{21}; server set 2 consists storage nodes r_2, r_7, r_{12}, r_{17}, and r_{22}; and server set 3 contains storage nodes r_3, r_8, r_{13}, r_{18}, and r_{23}; server set 4 is comprised of storage nodes r_4, r_9, r_{14}, r_{19}, and r_{24}; server set 5 contains storage nodes r_5, r_{10}, r_{15}, r_{20}, and r_{25}. It may occurred that at least three different replicas of the six fragment replicas f_{a1}, f_{a2}, f_{b1}, f_{b2}, f_{c1} and f_{c2} are allocated to storage nodes that belong to the same server-type group. For example, f_{a1}, f_{a2}, f_{b1}, f_{b2}, f_{c1} and f_{c2} are stored on nodes r_{19}, r_{14}, r_9, r_4, r_5, and r_{24} respectively. Nodes r_{19}, r_{14}, r_9, r_4, and r_{24} are the member of server group 4. If a malicious user successfully attacks server group 4, replicas f_{a1}, f_{a2}, f_{b1}, f_{b2}, and f_{c2} will be compromised, making the file F is reconstructed by the hacker. In such a case, one successful attack against server group $T4$ will break the confidentiality of file F.

In this example, system reliability is improved at the cost of system security. Increasing the number of fragment replicas leads to a high risk of having the fragment compromised by hackers. Both security and reliability are among the top desired prosperities of the cloud storage service, so fragment assignment schemes that consider both reliability and security are of great demand. In the rest of this paper we propose a fragment replica assignment solution that improves both security and reliability for heterogeneous cloud storage systems.

IV. SecRA: A Secure Replica Allocation Scheme

In this section, we first outline the desired properties in our solution including security, reliability and performance. Next, we describe the corresponding policies applied in the design of our solution to improve system security, reliability, and performance.

A. Design Goal of SecRA

The design goal of SecRA is to improve desired properties including security, reliability, and performance through fragment replica assignments of files in heterogeneous cloud storage systems. Based on the above analysis, we summarize the design objectives of SecRA below:

- **Security:** Our motivation example indicates that the replication scheme decreases system security. This is because the risk that a file is compromised increases due to the increased number of replicas stored in risky cloud storage systems. In many circumstances, the risk that the storage nodes are successfully attacked is out of control. Thus, our SecRA aims at boosting security even under conditions that some of the storage nodes are compromised.

  One way to improve security is to release as little information as possible through compromised storage nodes. In a fragment replication scheme, an ideal file assignment is to store all replicas of a fragment into one server-type group of storage nodes. Different server-type groups of storage nodes take charge of the replica storage for different fragments. Thus, with one set of successful attacks to any server-type group, only one fragment of a file is compromised, meaning that hackers are unable to construct the file.
- **Reliability:**
  To improve reliability of a cloud storage system, we incorporate the fragment replication in SecRA. Each fragment has a few duplicated replicas. The replication degree (a.k.a., number of replicas for each fragment) may be different among fragments, because fragments may have different access rates [18] [22].

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

To improve reliability, the SecRA scheme attempts to make the hardware as diverse as possible within a server group. Since the scheme makes an effort to distribute the replicas of a fragment to one server group, the replicas of the fragment are assigned to the heterogeneous storage nodes in the group. Thus, SecRA can increase reliability by decreasing the possibility that the homogeneous storage nodes fail at the same time.

- **Performance:**
  A handful of studies focused on the performance improvement of data replication strategies [6] [2]. Our storage node group categorization is a logical classification in a cloud storage system. The storage nodes within the same group may physically belong to different subsystems. When fragments located in different subsystems are transferred through the network, it is risky and takes more time than the case that all needed fragments are stored in one subsystem. Thus considering system performance, firstly all replicas should be distributed to the storage nodes that are close to the clients; secondly the replicas of a file should be distributed to less number of subsystems to reduce the risk of network transferring and time cost.

B. Design of the SecRA Scheme

The design object for SecRA is a simple yet efficient approach to distributing replicas of fragments for a file into storage nodes with various vulnerabilities and at same time to keep the file more secure, with higher reliability and performance.

The storage node deployment of a cloud storage system directly and fundamentally influences system prosperities including performance, security, reliability, scalability, etc. To improve the assurance of a cloud storage system while maintaining high I/O performance, each cluster storage subsystem has to be built with high heterogeneity in vulnerability. This provides the possibility that the fragments of a file are less likely distributed across multiple storage subclusters.

Based on the previous analysis, fragmentation is one of the techniques that improve storage security; secret sharing is also an efficient technique to improve storage security; the special fragment distribution strategies are another type of key methods for security improvement. The difference between non-replication and replication schemes is that there are multiple replicas for every fragment of a file. In addition to the above mentioned security techniques for the non-replication scheme, we distribute the multiple replicas of each fragment of a file during the replica distribution phase. We illustrate our security solution for replica distribution in policies 3 to 6 in the SecRA scheme.

The strategies to improve performance for non-replication scheme and replication scheme are very similar from the following perspectives: distributing replicas or fragments into the storage nodes that are close to clients; reducing replicas or fragments transferring across different subsystems; and distributing replicas or fragments to storage nodes with high CPU process speed and reliability.

One goal in our design is to reduce the replica or fragment transfer time across different subsystems for the replication scheme. There are different approaches for the non-replication and replication schemes. For the non-replication scheme, fragments of a file shoud be allocated to storage nodes within a subcluster. If the number of nodes with different vulnerabilities cannot meet the aforementioned criterion, some fragments of the file must be allocated across multiple clusters. To illustrate how useful our high performance strategy is for the replication scheme, we define a file's complete-replica set, which includes at least one replica copy for every fragment of the file. In order to reduce replica transferring, SecRA distributes at least one complete replica set of a file to a subsystem that is close to clients. Otherwise, at least one fragment replica must be transferred between two subsystems when users issue any read or write requests for the stored files in the cloud storage systems.

Considering reliability, disasters, unexpected downtime, and normal system maintenance can make some of the storage nodes out of service. The replication scheme is integrated into a cloud storage system to improve system reliability. Each file is divided to multiple fragments and each fragment has multiple replicas. Increasing the number of replicas stored in a cloud storage system can improve the system reliability.

The SecRA scheme makes storage node deployment for cloud storage systems and replica placement decisions by following the seven policies in the four categories below:

Storage Node Deployment (The policies in this category can be ignored, if SecRA is used in existing heterogeneous cloud storage systems. SecRA can be fully used by following some policies below during the storage node deployment phase.).

- Policy 1: All the storage nodes in a cloud storage system are classified into multiple server-type groups (server group for short) based upon their various vulnerabilities. Each server group consists of storage nodes with the same set of vulnerabilities. We place servers with diversified hardware configurations into one server group to make the hardware as diverse as possible to improve the reliability while the downtime is caused by hardware.

Reliability

- Policy 2: The replication scheme is integrated with the fragment assignment module to improve system reliability. Every file is divided into multiple fragments, each of which has multiple replicas.

Security

- Policy 3 (Please skip this policy if policy 1 is already implemented): All the storage nodes in a cloud storage system are classified into multiple server-type groups (server group for short) based upon their various vulnerabilities. Each server group consists of storage nodes with the same set of vulnerabilities.
- Policy 4: To improve security of a cloud storage system, SecRA allocates fragments of a file to storage nodes, which belong to as many different server-type groups as

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

possible.

- Policy 5: Replicas of the same fragment are assigned to the same server-type group of storage nodes. The goal of this policy is to improve the storage assurance of a file by making it less likely happen to compromise enough number of fragments of a file through one set of successful attack techniques for a certain server-type group of storage nodes.
- Policy 6: The (m, n) secret sharing scheme is incorporated into the SecRA allocation mechanism.

Performance

- Policy 7: In order to reduce replica transfer overhead, SecRA assigns at least one complete replica set of a file to a subsystem that is close to clients.

If the allocation decisions for the fragment replicas of a file are guided by the above seven policies, successful attacks against less than m server-type groups have little chance to gain unauthorized accesses of files stored in a cloud storage system. In other words, if the number of compromised unique replicas of a file's fragments is less than m, attackers are unable to reconstruct the file from the fragments accessible to hackers. The SecRA scheme can improve information assurance of files stored in a cloud storage system without enhancing confidentiality services deployed in cloud storage systems, because SecRA is orthogonal to security mechanisms offering confidentiality for each server group in cloud storage systems. Thus, SecRA can be seamlessly integrated with any confidentiality service employed in cloud storage systems in order to offer enhanced security services. At the same time, the integrated fragment replication scheme, the corresponding deployment of storage nodes, and the distribution of replicas help to improve reliability, fault tolerance, performance, as well as scalability.

V. Static Assurance Model for SecRA

In this section, we developed an assurance model to quantitatively evaluate the security of a heterogeneous cloud storage system, in which SecRA handles fragment allocations. To simplify the model, we develop the static storage assurance model for the case where all fragments share the same number t of replicas.

The model includes multiplication principle, probability theory, and combinatorics. We analyze the steps in the entire process of a successful attack to a cloud storage system. Here we use the multiplication principle to describe the attacking process. To describe the success rate of each step within the attacking process, we apply the probability theory and combinatorics to evaluate the process.

In the SecRA model, X means the event that a set of storage nodes is chosen to be attacked and $P(X)$ means the probability of event X occurring. V means the event that file F is compromised using one attack method and $P(V)$ represents the probability of event V occurring.

Y is the event that if X occurs, at least m different fragments' replicas can be compromised using the same attack method. The number of compromised replicas ranges from m to $n * t$(in S-FAS [20], it ranges from m to n), because every fragment has t number of replicas. The compromised fragments may have different (e.g., 1, ..., t) copies of replicas that are stored in the compromised nodes. In the following part, we derive probability $P(Y)$, which is the ratio of the combinatorial number of Y event to the combinatorial number of distributing the $n * t$ replicas to the N nodes in a cloud storage system.

To calculate the combinatorial number of event Y, we consider a special case W, where if one picks y replicas from the x fragments' replicas, the y pieces are the replicas of the x different fragments.

We denote $T(x, y)$ as the combinatorial number of case W. In the chosen jth storage server group that includes S_j storage nodes, if event Y occurs, the number of nodes storing replicas of file F may range from m to S_j ($m <= S_j$), because there might be a portion of nodes in the jth group storing replicas of file F.

Now we can draw the combinatorial number ($CM(Y)$) of Y from the above expression as below:

$$CM(Y) = \sum_{y=m}^{S_j} [T(m, y) + T(m+1, y) + T(m+2, y) + ... + T(n, y)] \tag{1}$$

When $t >= y$, we have the following further reasoning and expression of $T(x, y)$ (we do not give the general formula when $t < y$ and it is not that hard to calculate $T(x, y)$ if we have the concrete values for all the related parameters).

$$\begin{aligned} &T(1, y) = C_t^y \\ &T(2, y) = C_{2t}^y - C_2^1 * T(1, y) = C_2^2 * C_{2t}^y - C_2^1 * C_t^y \\ &T(3, y) = C_{3t}^y - C_3^2 * T(2, y) - C_3^1 * T_1^y \\ &= C_3^3 * C_{3t}^y - C_3^2 * C_{2t}^y + C_3^1 * C_t^y \\ &... \\ &T(x, y) = C_x^x * C_{xt}^y - C_x^{x-1} * C_{(x-1)t}^y + ... \\ &= \sum_{i=0}^{x-1} (-1)^i * C_x^{x-i} * C_{(x-i)t}^y \end{aligned} \tag{2}$$

The combinatorial number $CM(n * t)$ of distributing the $n * t$ replicas of file F to the N nodes in a cloud storage system can be expressed as:

$$CM(n * t) = C_N^t * C_{N-t}^t * C_{N-2t}^t * ... * C_{N-(n-1)t}^t \tag{3}$$

Based on the definition of $P(Y)$–the rate of the combinatorial number of event Y dividing the combinatorial number of distributing the $n * t$ replicas to the N nodes, we can derive $P(Y)$ from both $CM(Y)$ and $CM(n * t)$ as:

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

$$P(Y) = \frac{CM(Y)}{CM(n*t)} = \frac{\sum_{y=m}^{S_j} [T(m,y) + T(m+1,y) + T(m+2,y) + ... + T(n,y)]}{C_N^t * C_{N-t}^t * C_{N-2t}^t * C_{N-3t}^t * ... * C_{N-(n-1)t}^t} \quad (4)$$

$P(V)$ can be derived from Eq. 4 as:

$$P(V) = \sum_{j=1}^{K} \{\frac{S_j}{N} * P(Z) * \frac{\sum_{y=m}^{S_j} [T(m,y) + T(m+1,y) + ... + T(n,y)]}{C_N^t * C_{N-t}^t * C_{N-2t}^t * ... * C_{N-(n-1)t}^t}\} \quad (5)$$

The confidentiality of file F is assured if file F is unable to be reconstructed by an attacker. Given a replica fragment assignment decision α, we can derive the storage assurance $SA(\alpha)$ for file F stored in a cloud storage system as:

$$SA(\alpha) = 1 - P(V) = 1 - \sum_{j=1}^{K} \{\frac{S_j}{N} * P(Z) * \frac{\sum_{y=m}^{S_j} [T(m,y) + T(m+1,y) + ... + T(n,y)]}{C_N^t * C_{N-t}^t * C_{N-2t}^t * ... * C_{N-(n-1)t}^t}\} \quad (6)$$

VI. Evaluation of System Assurance

In addition to the influential factors (i.e., K, N, n and m) in the S-FAS scheme [20], replication degree t also affects the assurance model for the SecRA scheme. The rest of this sub-section presents our quantitative evaluation on the impacts of these factors on the information assurance of cloud storage systems.

A. Impact of Replication Degree on Storage Assurance

Obviously, increasing the number of replicas improves the availability of a cloud storage system. With more replicas stored across multiple nodes in the system, a system provides good I/O throughput assuming load balancing is handled well. Fig. 5 shows the impacts of replication degree on the static assurance of a cloud storage system.

In this experiment, the secret-sharing threshold is increased from 1 to 4. We evaluate the static assurance of a 50-node cloud storage system, where there are five different server types. The number of replicas is set to 1, 2, and 4 respectively. Fig. 5 confirms that with the increasing value of the replication degree, the assurance of the cloud storage system is decreased. This observation suggests that we should consider limiting the boundary of replication degree to control the storage assurance at an acceptable level in a risky network environment.

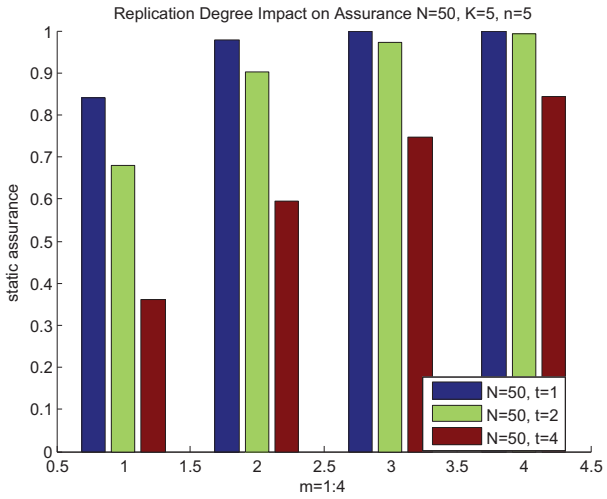

Fig. 5. *Replication Degree Impact on Assurance*

B. Impact of System Size on Storage Assurance

System size or the number of storage nodes (N) in a cloud storage system is one of the important parameters for the system. We further study the system size impact on storage assurance. In the second experiment, we keep the number of replicas to 2. The value of N is set to 30, 40, and 50. The other parameters are kept the same as those in the first experiment. Fig. 6 presents the preliminary results of the impact of system size in the SecRA replication scheme.

Fig. 6 shows that the storage assurance increases with the increasing of system size. This result indicates that for our proposed SecRA scheme, even the diversity level, here represented by K, is kept unchanged, the storage assurance can be enhanced by increasing the number of storage nodes within every server-type of storage nodes.

This trend on assurance with the replication scheme is significantly different from that of the non-replicas-based scheme in S-FAS [20], where increasing the system size has no impact on the storage assurance if the diversity level (K) is a constant. This observation demonstrates that SecRA is especially useful for large-scale cloud storage systems, where replicas are deployed to improve reliability. We can increase the storage assurance by adding more nodes to a cloud storage system even if the diversity level of the system does not increase.

We define that the distance between n and m eaquals to $n - m$. Interestingly, when the system size is increased to a certain level and the system size is increased to a certain level (here when N is greater than 40), the static assurance of the cloud storage system becomes less sensitive to the system size.

C. Impact of Number n of File Fragments on Storage Assurance

The number of fragments for each file in our integrated secret sharing scheme is another important parameter affecting storage assurance. We now study the impact of the fragment

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

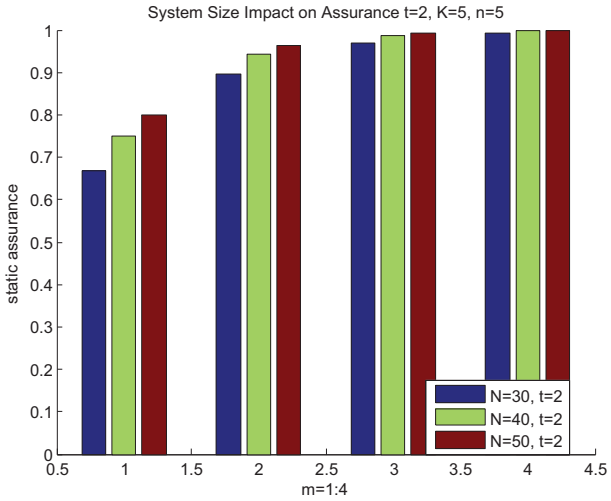

Fig. 6. *System Size Impact on Assurance*

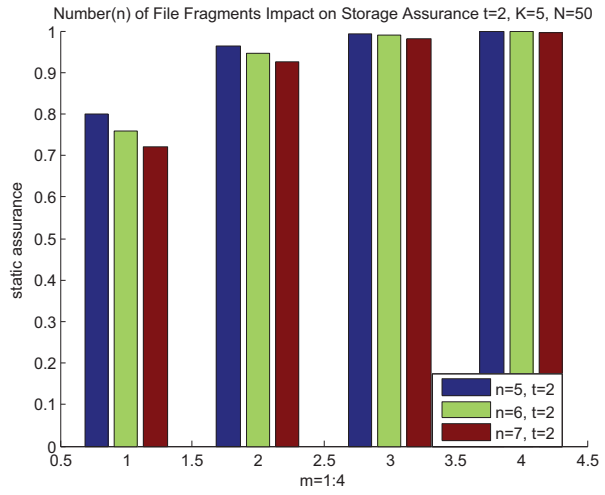

Fig. 7. *Number of Fragments Impact on Assurance*

number n on storage assurance. In the design of the experiment, the number n of fragments is increased from 5 to 7. The parameters k and N are set to 5 and 50, respectively. We also vary the threshold m from 1 to 4.

Fig. 7 shows that the system assurance is decreasing with the increasing of fragment number n. The trend of the SecRA scheme is the same as the trend of S-FAS in [20]. The reason is that, with the increasing of fragment number n, the possibility that more fragments of a file are allocated to storage nodes of the same server-type group is increased. Compared to S-FAS in [20], SecRA is more sensitive to the number n of a file's fragments. The reason is that each replica of a fragment in SecRA has the same level of influence on storage assurance as a single fragment in the S-FAS scheme in [20].

Thus, we can reach a conclusion that increasing the number of fragments for files stored in a cloud storage system reduces the storage assurance of the system.

VII. Conclusion

It is critical to maintain the confidentiality of files stored in a cloud storage system, even when some storage nodes in the system are compromised by attackers. Data replication is one of the widely used techniques to improve reliability and performance in cloud storage systems. However, data replication increases the risk of data storage in an unsecure network environment. With the increasing of size and scalability of cloud systems, the heterogeneous feature is becoming much more prevalent among the storage nodes of a cloud storage system.

In recognizing that storage nodes in a cloud storage system have heterogeneous vulnerabilities, we propose a scheme called SecRA to improve the cloud storage system security and reliability by integrating the techniques of fragmentation, data replication and secret sharing. The SecRA scheme aims to improve security, reliability and performance of the cloud storage systems by considering heterogeneous vulnerabilities and load balancing during the replica placement phase. SecRA tries to allocate fragments of a file in such a way that even if attackers compromised a number of server-type groups and fewer than k different replicas are disclosed, the file cannot be reconstructed by the attackers from the compromised fragments.

We then describe the system model and illustrate our idea by a motivation example. To quantitatively evaluate the security quality offered by SecRA when all fragments have the same replication degrees, a storage assurance model is developed. To analyze storage assurance provided by SecRA, we study the impacts caused by replication degree, system size, and the number of fragments and derive a few principles when using the proposed SecRA scheme.

More replicas stored across multiple nodes within the system provides good I/O throughput. However, our analysis shows that with the increasing value of the replication degree, the assurance of the cloud storage system is decreasing. This observation suggests that we should consider limiting the boundary of replication degree to control the storage assurance at an acceptable level. Interestingly, we find that when distance between n and m in the integrated secret sharing scheme (m, n) is big, static assurance can be improved by increasing the number of nodes in the system. This discovery suggests that in the systems with the similar diversity, the larger the system, the possible higher storage assurance may be achieved by following the SecRA replica placement.

There are three future research directions of this study. In this paper, we only focus on the static replica placement solution. Dynamic replica reallocation schemes are essential to achieve high performance and availability of cloud storage systems, especially for internet based applications and services. In a dynamic wide-area environment, client access patterns, network conditions, and service characteristics are constantly changing. Firstly we plan to study and propose a dynamic replica reallocation approach to address the heterogeneous vulnerabilities in the large scale cloud storage systems. Secondly,

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.

we will implement a cloud storage system prototype where SecRA is deployed. In this prototype, we will evaluate the performance of SecRA by traces in real-world systems. Third, we will extend SecRA and improve the performance of the scheme by using multi-threading technique in a heterogeneous cloud storage system.

REFERENCES

[1] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with cfs. In *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*, pages 202–215, New York, NY, USA, 2001. ACM.

[2] A. Elghirani, R. Subrata, and A.Y. Zomaya. Intelligent scheduling and replication in datagrids: a synergistic approach. In *Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on*, pages 179–182, 2007.

[3] Daniel Ford, Francois Labelle, Florentina Popovici, Murray Stokely, Van-Anh Truong, Luiz Barroso, Carrie Grimes, and Sean Quinlan. Availability in globally distributed storage systems. In *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation*, 2010.

[4] Lei Gao, M. Dahlin, A. Nayate, Jiandan Zheng, and Arun Iyengar. Improving availability and performance with application-specific data replication. *Knowledge and Data Engineering, IEEE Transactions on*, 17(1):106–120, 2005.

[5] C. Hannon and J.R. Rinewalt. Addressing security issues in geographically distributed systems. In *Computer Science, 2003. ENC 2003. Proceedings of the Fourth Mexican International Conference on*, pages 182–189, 2003.

[6] Hui-I Hsiao and D.J. DeWitt. A performance study of three high availability data replication strategies. In *Parallel and Distributed Information Systems, 1991., Proceedings of the First International Conference on*, pages 18–28, 1991.

[7] Pankaj Jalote. *Fault tolerance in distributed systems*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994.

[8] Vishal Kher and Yongdae Kim. Securing distributed storage: challenges, techniques, and systems. In *Proceedings of the 2005 ACM workshop on Storage security and survivability*, StorageSS '05, pages 9–25, New York, NY, USA, 2005. ACM.

[9] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. Oceanstore: an architecture for global-scale persistent storage. *SIGPLAN Not.*, 35:190–201, November 2000.

[10] Thanasis Loukopoulos and Ishfaq Ahmad. Static and adaptive distributed data replication using genetic algorithms. *Journal of Parallel and Distributed Computing*, 64(11):1270 – 1285, 2004.

[11] Najme Mansouri and GholamHosein Dastghaibyfard. Job scheduling and dynamic data replication in data grid environment. *The Journal of Supercomputing*, 64(1):204–225, 2013.

[12] A. Mei, L. V. Mancini, and S. Jajodia. Secure dynamic fragment and replica allocation in large-scale distributed file systems. *IEEE Trans. on Parallel and Distributed Systems*, 14(9):885 – 896, Sept. 2003.

[13] D.M. Nessett. Factors affecting distributed system security. *Software Engineering, IEEE Transactions on*, SE-13(2):233–248, 1987.

[14] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 129–140, London, UK, 1992. Springer-Verlag.

[15] M. Pourzandi, D. Gordon, W. Yurcik, and G. A. Koenig. Clusters and security: distributed security for distributed systems. In *Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on*, volume 1, pages 96 – 104 Vol. 1, May 2005.

[16] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[17] G. J. Simmons. How to (really) share a secret. In *CRYPTO '88: Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*, pages 390–448, London, UK, 1990. Springer-Verlag.

[18] Swaminathan Sivasubramanian, Gustavo Alonso, Guillaume Pierre, and Maarten van Steen. Globedb: autonomic data replication for web applications. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 33–42, New York, NY, USA, 2005. ACM.

[19] Ming Tang, Bu sung Lee, Xueyan Tang, and Chai kiat Yeo. The impact of data replication on job scheduling performance. In *in the Data Grid, Future Generation Computer Systems*, pages 254–268, 2006.

[20] Yun Tian, Shu Yin, Jiong Xie, Ji Zhang, Xiao Qin, M.I Alghamdi, Meikang Qiu, and Yiming Yang. Secure fragment allocation in a distributed storage system with heterogeneous vulnerabilities. In *Networking, Architecture and Storage (NAS), 2011 6th IEEE International Conference on*, pages 170–179, July 2011.

[21] M. Tu, P. Li, I-Ling Yen, B. M. Thuraisingham, and L. Khan. Secure data objects replication in data grid. *IEEE Trans. on Dependable and Secure Computing*, 7(1):50 –64, 2010.

[22] Ouri Wolfson, Sushil Jajodia, and Yixiu Huang. An adaptive data replication algorithm. *ACM Trans. Database Syst.*, 22(2):255–314, June 1997.

[23] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kiliccote, and P. K. Khosla. Survivable information storage systems. *Computer*, 33(8):61 –68, Aug 2000.

[24] Jiong Xie, Shu Yin, Xiaojun Ruan, Zhiyang Ding, Yun Tian, J. Majors, A. Manzanares, and Xiao Qin. Improving mapreduce performance through data placement in heterogeneous hadoop clusters. In *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1–9, 2010.

[25] Tao Xie and Xiao Qin. A security-oriented task scheduler for heterogeneous distributed systems. In *Proc. 13th Annual IEEE International Conf. High Performance Computing, Lecture Notes in Computer Science (LNCS 4297), ISSN 0302-9743*, pages 35–46, 2006.

[26] G. Zanin, A. Mei, and L. V. Mancini. Towards a secure dynamic allocation of files in large scale distributed file systems. In *HOT-P2P '04: Proceedings of the 2004 International Workshop on Hot Topics in Peer-to-Peer Systems*, pages 102–107, Washington, DC, USA, 2004. IEEE Computer Society.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:00:39 UTC from IEEE Xplore. Restrictions apply.