Security Improvements for Enterprise File Synchronization and Sharing System

Muhammad Ihsan Haikal Sukmana

Universitätsdissertation
zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

in der Wissenschaftsdisziplin
Internet-Technologien und Systeme

eingereicht an der
Digital-Engineering-Fakultät
der Universität Potsdam

Betreuer

Prof. Dr. Christoph Meinel
Hasso Plattner Institute, University of Potsdam

Prof. Dr. Marian Margraf
Freie Universität Berlin, Germany

Prof. Heru Suhartanto, Drs, M.Sc., Ph.D.
University of Indonesia, Depok, Indonesia

Published online on the
Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-54999
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-549996

Abstract

With the fast rise of cloud computing adoption in the past few years, more companies are migrating their confidential files from their private data center to the cloud to help enterprise's digital transformation process. Enterprise file synchronization and share (EFSS) is one of the solutions offered for enterprises to store their files in the cloud with secure and easy file sharing and collaboration between its employees. However, the rapidly increasing number of cyberattacks on the cloud might target company's files on the cloud to be stolen or leaked to the public. It is then the responsibility of the EFSS system to ensure the company's confidential files to only be accessible by authorized employees.

CloudRAID is a secure personal cloud storage research collaboration project that provides data availability and confidentiality in the cloud. It combines erasure and cryptographic techniques to securely store files as multiple encrypted file chunks in various cloud service providers (CSPs). However, several aspects of CloudRAID's concept are unsuitable for secure and scalable enterprise cloud storage solutions, particularly key management system, location-based access control, multi-cloud storage management, and cloud file access monitoring.

This Ph.D. thesis focuses on CloudRAID for Business (CfB) as it resolves four main challenges of CloudRAID's concept for a secure and scalable EFSS system. First, the key management system is implemented using the attribute-based encryption scheme to provide secure and scalable intra-company and inter-company file-sharing functionalities. Second, an Internet-based location file access control functionality is introduced to ensure files could only be accessed at pre-determined trusted locations. Third, a unified multi-cloud storage resource management framework is utilized to securely manage cloud storage resources available in various CSPs for authorized CfB stakeholders. Lastly, a multi-cloud storage monitoring system is introduced to monitor the activities of files in the cloud using the generated cloud storage log files from multiple CSPs.

In summary, this thesis helps CfB system to provide holistic security for company's confidential files on the cloud-level, system-level, and file-level to ensure only authorized company and its employees could access the files.

Zusammenfassung

Mit der raschen Verbreitung von Cloud Computing in den letzten Jahren verlagern immer mehr Unternehmen ihre vertraulichen Dateien von ihren privaten Rechenzentren in die Cloud, um den digitalen Transformationsprozess des Unternehmens zu unterstützen. Enterprise File Synchronization and Share (EFSS) ist eine der Lösungen, die Unternehmen angeboten werden, um ihre Dateien in der Cloud zu speichern und so eine sichere und einfache gemeinsame Nutzung von Dateien und die Zusammenarbeit zwischen den Mitarbeitern zu ermöglichen. Die schnell wachsende Zahl von Cyberangriffen auf die Cloud kann jedoch dazu führen, dass die in der Cloud gespeicherten Unternehmensdateien gestohlen werden oder an die Öffentlichkeit gelangen. Es liegt dann in der Verantwortung des EFSS-Systems, sicherzustellen, dass die vertraulichen Dateien des Unternehmens nur für autorisierte Mitarbeiter zugänglich sind.

CloudRAID ist ein Forschungsprojekt für sichere persönliche Cloud-Speicher, das die Verfügbarkeit und Vertraulichkeit von Daten in der Cloud gewährleistet. Es kombiniert Lösch- und Verschlüsselungstechniken, um Dateien in Form von mehreren verschlüsselten Datei-Blöcken bei verschiedenen Cloud-Service-Providern (CSPs) sicher zu speichern. Mehrere Aspekte des CloudRAID-Konzepts sind jedoch für sichere und skalierbare Cloud-Speicherlösungen für Unternehmen ungeeignet, insbesondere das Schlüsselverwaltungssystem, die standortbasierte Zugriffskontrolle, die Verwaltung mehrerer Cloud-Speicher und die Überwachung des Zugriffs auf Cloud-Dateien.

Diese Doktorarbeit konzentriert sich auf CloudRAID for Business (CfB), da es die vier wichtigsten Herausforderungen des CloudRAID-Konzepts für ein sicheres und skalierbares EFSS-System löst. Erstens wird das Verwaltungssystem der kryptografischen Schlüssel unter Verwendung des attributbasierten Verschlüsselungsschemas implementiert, um sichere und skalierbare unternehmensinterne und -übergreifende Dateifreigabefunktionen bereitzustellen. Zweitens wird eine internetbasierte Dateizugriffskontrolle eingeführt, um sicherzustellen, dass der Zugriff auf Dateien nur an vorher festgelegten vertrauenswürdigen Standorten möglich ist. Drittens wird ein einheitlicher Rahmen für die Verwaltung

von Multi-Cloud-Speicherressourcen verwendet, um die in verschiedenen CSPs verfügbaren Cloud-Speicherressourcen für autorisierte CfB-Akteure sicher zu verwalten. Schließlich wird ein Multi-Cloud-Storage-Monitoring-System eingeführt, um die Aktivitäten von Dateien in der Cloud anhand der von mehreren CSPs generierten Cloud-Storage-Protokolldateien zu überwachen.

Zusammenfassend lässt sich sagen, dass diese Arbeit dem CfB-System hilft, ganzheitliche Sicherheit für vertrauliche Unternehmensdateien auf Cloud-, System- und Dateiebene zu bieten, um sicherzustellen, dass nur autorisierte Unternehmen und ihre Mitarbeiter auf die Dateien zugreifen können.

Acknowledgments

First, I would like to express my deep gratitude to my supervisor, Prof. Christoph Meinel, for allowing me to do the Ph.D. study at Hasso Plattner Institute and for the trust, guidance, and support throughout my Ph.D. study. I would also like to thank Prof. Felix Naumann as my second supervisor for the discussion and the advice during my Ph.D. study.

I am incredibly grateful to my parents, my families, my wife, and my friends for their constant supports and encouragement throughout my study. Without them, I would not be where I am today.

I am deeply indebted to Kennedy Torkura and Hendrik Graupner, who welcome me since the beginning of my Ph.D. study and give me invaluable discussions, inputs, and helps throughout the CloudRAID project that help to shape my research direction.

I am incredibly grateful to Dr. Feng Cheng for the insights, suggestions, and supports during my Ph.D. study. I am also grateful to the former and current members of the Security Engineering group, and Ra'ad bin Tareaf for their help and advice during my Ph.D. study.

I would like to thank Bundesdruckerei GmbH and neXenio GmbH, particularly Dr. Maxim Schnjakin, Jörg Kirchhof, Philipp Berger, Patrick Hennig, Andrina Mascher, and Daniel Moritz, for their supports in the CloudRAID project.

Finally, many thanks to Sezi Prasetyo, Kai-Oliver Kohlen, Justus Cöster, Marvin Petzolt, Ferdiansyah Dolot, and other Bachelor and Master seminar students for their help to research and develop several of my research topics.

Eidesstattliche Erklärung

Hiermit versichere ich, dass meine Arbeit “Security Improvements for Enterprise File Synchronization and Sharing System” selbständig verfasst wurde und dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt wurden. Diese Aussage trifft auch für alle Implementierungen und Dokumentationen im Rahmen dieses Projektes zu.

Potsdam, den 06.10.2021,

(Muhammad Ihsan Haikal Sukmana)

Contents

1 Introduction

1.1 The Need for Secure Enterprise Cloud Storage Solutions

Cloud computing has revolutionized the enterprise computing paradigm in the past ten years, where more companies have migrated their data and infrastructure from their private data center to the cloud. Gartner forecasts the worldwide public cloud services market to grow to $354.6 billion in 2022, where 60% of the organizations will use external cloud service providers [Gar19]. Cloud computing offers various advantages that help to accelerate digital transformation for the companies, such as guaranteed availability, high scalability, cost-saving, and less maintenance than private data center [Sal21].

With the fast adaptation rate of cloud computing, an increasing amount of enterprise data is now stored in the cloud. In 2019, 48% of corporate files were stored in the cloud, which is a 13% increase in the last three years [Tha19]. Meanwhile, The COVID-19 pandemic also helps accelerate the data migration to the public cloud due to the need to digitalize their services or operations. According to Flexera's State of the Cloud 2021 report [Fle21], 46 percent of the organization's data is stored in the public cloud, and more than half of company respondents consider moving their data to the cloud. It is predicted that 100 zettabytes of data will be stored on the cloud by 2025 [Ste20].

With more companies and employees becomes more even dependent on the cloud, the cybercriminals are launching more cyberattacks to the cloud that caused the number of data breach incidents to be increasing over the years. Therefore, there is an increasing need for enterprise file synchronization and share (EFSS) to provide easy and secure file storage on the cloud for enterprise usage, primarily due to the COVID-19 pandemic that pushes enterprise's digital transformation. According to MarketsandMarkets [Mar21], EFSS market size is projected to grow from USD 6.1 billion in 2021 to USD 20.5 billion by 2026 with a Compound Annual Growth Rate of 27.4%.

CloudRAID[1] is a research project collaboration between Hasso Plattner Institute gGmbH[2] and Bundesdruckerei GmbH[3] focusing on providing secure storage solution in the cloud. It combines encryption techniques and the Redundant Array of Inexpensive Disks (RAID) concept to securely store users' files as multiple encrypted file chunks in various cloud service providers (CSPs). CloudRAID user's files will still be available if one or several CSPs are inaccessible, e.g., due to cloud outage, where no entity, i.e., the CSPs and CloudRAID, except authorized users, could access the files stored in the cloud.

Since there is an increasing need for enterprise cloud storage solutions to store the company's confidential files in the cloud securely, CloudRAID for Business is then developed based on CloudRAID's mechanisms to answer the market's need. It aims to provide secure and highly available cloud file storage using the cryptographic and erasure methods by storing the company's confidential files as multiple encrypted chunks across various CSPs.

However, since CloudRAID is initially developed for personal usage, its mechanisms and architecture are not suitable for enterprise usage since the requirements for enterprise cloud storage solutions differ from personal cloud storage solutions. It would require significant improvements on several areas of CloudRAID for Business to provide a secure cloud enterprise file synchronization and share system.

1.2 Thesis Contributions and Research Questions

This thesis aims to resolve several challenges faced by CloudRAID for Business to provide secure cloud storage solution for enterprise usage. Although there are various CloudRAID's areas that could be improved to support company's operations related to accessing its confidential files, this thesis only focus on four main areas as its scope: key management system, file access control, multi-cloud storage resource management, and cloud file access monitoring.

This thesis aims to answer four main research questions as follows:

1. How could CloudRAID for Business provide secure and scalable intra-company and inter-company file-sharing functionalities in the system?

[1] https://hpi.de/meinel/security-tech/secure-cloud/secure-cloud-storage.html

[2] https://hpi.de/

[3] https://www.bundesdruckerei.de/

   CloudRAID's key management system utilizing cryptographic methods based on AES and RSA algorithms is used to secure CloudRAID user's files on the cloud and the system. However, the current cryptographic techniques generate multiple encrypted file keys per encrypted file, creating scalability and access control issues that make it unsuitable for enterprise file-sharing. CloudRAID also does not provide file access control based on the employee's role in the company, which is essential for secure enterprise file-sharing functionality.

   A new key management system based on attribute-based encryption schemes is introduced for CloudRAID for Business to provide secure and scalable intra-company and inter-company file-sharing functionalities. Due to attribute-based encryption's "one-to-many" property, it generates one encrypted file key per CfB user's file for multiple users and their devices in the system. It can enforce file access control within and across the company's domains where only the authorized CfB users with the correct attributes in the system that fulfill the file-sharing specification can decrypt the encrypted file key. Finally, it allows the company to securely and scalably manage its confidential files and employees as CfB users in the system. For example, revoke access to the shared file for the company's ex-employee or ensure the new employee could access the shared file.

2. How could CloudRAID for Business provide secure and reliable access control for company's confidential files based on the employee's location?

   CloudRAID provides file access control to ensure only authorized CloudRAID users and their devices could access the files. A company might require a different access control mechanism to secure its confidential files that could only be accessed at specific trusted locations, which CloudRAID does not yet provide. CloudRAID should be able to calculate the users' location and verify whether the submitted location is at the pre-determined locations and not manipulated.

   Location-based access control functionality is implemented for CloudRAID for Business to ensure the company's confidential files can only be accessed in the allowed location. It utilizes Internet-based location as the location input inferred from the information provided or retrieved from Internet-connected devices, such as IP address, delay measurement result with known landmark servers, and surrounding Wi-Fi access points. It utilizes third-party open-source intelligence, delay-based geolocation algorithms, and landmark servers deployed in the multiple CSPs in the European region and randomly selected from the Speedtest network to determine the location for the devices with minimum or no self-geolocation capability and verify whether the users have manipulated it.

3. How could CloudRAID for Business securely manage its multi-cloud storage environment for its stakeholders while ensuring secure enterprise cloud storage solution?

   CloudRAID utilizes the cloud object storage services from multiple CSPs and erasure technique to store CloudRAID user's files as multiple chunks across multiple CSPs to ensure the files to be available when one of the CSPs is unavailable. With CloudRAID transitions itself to be an enterprise cloud storage solution, it is responsible for securing the files stored in various CSPs from unauthorized entities, especially with the increasing number of cyber-attacks on the cloud infrastructure in the past few years. CloudRAID also needs to securely manage its multi-cloud storage environment for different stakeholders that require limited and secure access while resolving the challenge of CSP heterogeneity.

   A unified multi-cloud storage resource management framework is proposed for CloudRAID for Business to manage the cloud storage resources in various CSPs securely. The framework consists of the unified cloud storage resource model to resolve the challenge of the various data model of cloud storage resources in multiple, the unified multi-cloud storage resource management platform for automated and centralized cloud storage resource management, and guidelines and instructions to ensure secure cloud storage access management for CfB stakeholders.

4. How could CloudRAID for Business monitor and analyze CfB user's file activities happening in multiple cloud service providers for suspicious or malicious activities?

   CloudRAID does not actively monitor CloudRAID user's files stored across multiple CSPs to ensure a zero-knowledge policy within the system. However, with the increasing number of data breaches over the years, an enterprise needs to be aware of the latest state and the activities of its confidential files stored on the cloud. It could help companies to ensure the files are only accessed by authorized employees.

A multi-cloud storage monitoring system is introduced for CloudRAID for Business to monitor the activities of the CfB user's files stored in the cloud. It collects, processes, and analyzes the cloud storage log files generated from multiple CSPs with different log formats and quality information. The log files, which record the events happening on the cloud object storage services, are also correlated with CfB system log entries to detect suspicious or malicious activities in the cloud and the system.

1.3 Thesis Structure

The structure of this thesis is as follows:

Chapter 2 explains the technologies and concepts used by CloudRAID and how CloudRAID provides a secure cloud storage solution for personal usage by combining erasure and encryption techniques for storing CloudRAID user's files in multiple CSPs. It also describes CloudRAID for Business and its challenges and requirements that need to be satisfied to provide a secure enterprise cloud storage solution.

Chapter 3 addresses the issue of scalability and access control issues of the CloudRAID's key management system based on RSA-based public key infrastructure for secure file-sharing between CloudRAID users, which is not suitable for enterprise usage. Attribute-based encryption is proposed for CloudRAID for Business' key management system to provide secure and scalable intra-company and inter-company file-sharing functionalities for the companies as the CfB customers. The chapter is based on two published papers:

- Muhammad Ihsan Haikal Sukmana, Kennedy Aondona Torkura, Christoph Meinel, and Hendrik Graupner. 2017. **Redesign CloudRAID for Flexible and Secure Enterprise File Sharing over Public Cloud Storage**. *In Proceedings of the 10th International Conference on Security of Information and Networks (SIN) 2017.* ACM [Suk+17].

- Muhammad Ihsan Haikal Sukmana, Marvin Petzolt, Kennedy Aondona Torkura, Hendrik Graupner, Feng Cheng, and Christoph Meinel. 2019. **Secure and Scalable Multi-Company Management in Enterprise**

Cloud Storage Broker System. *In Proceedings of the 17th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA) 2019.* IEEE [Suk+19a].

Chapter 4 describes the need for access control for the files based on the location of the CfB users to ensure the files are only accessible in certain locations set by the companies. The location-based access control mechanism is proposed using the location inferred from the CfB user's Internet-connected devices based on its IP address, surrounding Wi-Fi access points, and the delay measurement result between the devices and the active landmarks spread across Europe. The chapter is based on two publications:

- Muhammad Ihsan Haikal Sukmana, Kennedy Aondona Torkura, Hendrik Graupner, Ankit Chauhan, Feng Cheng, and Christoph Meinel. 2019. **Supporting Internet-based Location for Location-Based Access Control in Enterprise Cloud Storage Solution**. *In Proceedings of the 33th International Conference on Advanced Information Networking and Applications (AINA) 2019.* Springer [Suk+19b].
- Muhammad Ihsan Haikal Sukmana, Kai-Oliver Kohlen, Carl Gödecken, Pascal Schulze, Christoph Meinel. 2021. **Are You There, Moriarty? Feasibility Study of Internet-based Location for Location-Based Access Control Systems**. *In Proceedings of the 18th International Conference on Security and Cryptography (SECRYPT) 2021.* SciTePress [Suk+21b].

Chapter 5 proposes a unified multi-cloud storage resource management framework to resolve the challenges of securely managing the multi-cloud storage environment used by CfB. The framework allows for automated and centralized cloud storage resource management across multiple CSPs for authorized CfB stakeholders while ensuring confidentiality and availability of CfB services and the company's confidential files stored in the cloud. The chapter is based on two publications:

- Muhammad Ihsan Haikal Sukmana, Kennedy Aondona Torkura, Hendrik Graupner, Feng Cheng, and Christoph Meinel. 2019. **Unified Cloud Access Control Model for Cloud Storage Broker**. *In Proceedings of the 33rd International Conference on Information Networking (ICOIN) 2019.* IEEE [Suk+19c].

- Muhammad Ihsan Haikal Sukmana, Kennedy Aondona Torkura, Sezi Dwi Sagarianti Prasetyo, Feng Cheng, and Christoph Meinel. 2020. **A Brokerage Approach for Secure Multi-Cloud Storage Resource Management**. *In Proceedings of the 16th EAI International Conference on Security and Privacy in Communication Networks (SecureComm) 2020.* Springer [Suk+20].

Chapter 6 introduces the mechanism to monitor the activities of CfB user's files stored in multiple CSPs. Cloud storage log files, which record the events happening on the cloud object storage services, are collected, processed, and correlated with the CfB system log entries to detect any suspicious or malicious activities happening in the multi-cloud storage environment. The chapter is based on three published papers:

- Muhammad Ihsan Haikal Sukmana, Kennedy Aondona Torkura, Feng Cheng, Christoph Meinel, and Hendrik Graupner. 2018. **Unified Logging System for Monitoring Multiple Cloud Storage Providers in Cloud Storage Broker**. *In Proceedings of the 32nd International Conference on Information Networking (ICOIN) 2018.* IEEE [Suk+18].
- Muhammad Ihsan Haikal Sukmana, Kennedy Aondona Torkura, Sezi Dwi Sagarianti Prasetyo, Feng Cheng, and Christoph Meinel. 2020. **A Brokerage Approach for Secure Multi-Cloud Storage Resource Management**. *In Proceedings of the 16th EAI International Conference on Security and Privacy in Communication Networks (SecureComm) 2020.* Springer [Suk+20].
- Muhammad Ihsan Haikal Sukmana, Justus Cöster, Wenzel Puenter, Kennedy Aondona Torkura, Feng Cheng, and Christoph Meinel. 2021. **A Feasibility Study of Log-based Monitoring for Multi-Cloud Storage Systems**. *In Proceedings of the 35th International Conference on Advanced Information Networking and Applications (AINA) 2021.* Springer [Suk+21a].

Finally, Chapter 7 concludes the thesis and proposes ideas for further work on the topics proposed in the thesis.

There are papers and journals published during the Ph.D. study that are not included in this thesis because of topic differences:

- Muhammad Ihsan Haikal Sukmana and Christoph Meinel. 2016. **e-Government and Security Evaluation Tools Comparison for Indonesian e-Government System**. *In Proceedings of the 4th International Conference on Information and Network Security (ICIN) 2016.* ACM [SM16].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Feng Cheng, and Christoph Meinel. 2017. **Leveraging Cloud Native Design Patterns for Security-as-a-Service Applications**. *In Proceedings of the 2nd IEEE International Conference on Smart Cloud (SmartCloud) 2017.* IEEE [Tor+17].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, and Christoph Meinel. 2017. **Integrating Continuous Security Assessments in Microservices and Cloud Native Applications**. *In Proceedings of the 10th International Conference on Utility and Cloud Computing (UCC) 2017.* ACM [TSM17].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Michael Meinig, Feng Cheng, Christoph Meinel, and Hendrik Graupner. 2018. **A Threat Modeling Approach for Cloud Storage Brokerage and File Sharing Systems**. *In Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS) 2018.* IEEE [Tor+18d].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Michael Meinig, Anne VDM Kayem, Feng Cheng, Hendrik Graupner, and Christoph Meinel. 2018. **Securing Cloud Storage Brokerage Systems Through Threat Models**. *In Proceedings of the 32nd IEEE International Conference on Advanced Information Networking and Applications (AINA) 2018.* IEEE [Tor+18e].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Feng Cheng, and Christoph Meinel. 2018. **CAVAS: Neutralizing Application and Container Security Vulnerabilities in the Cloud Native Era**. *In Proceedings of the 14th EAI International Conference on Security and Privacy in Communication Systems (SecureComm) 2018.* Springer [Tor+18b].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Tim Strauss, Hendrik Graupner, Feng Cheng, and Christoph Meinel. 2018. **CSBAuditor: Proactive Security Risk Analysis for Cloud Storage**

Broker Systems. *In Proceedings of the 17th IEEE International Symposium on Network Computing and Applications (NCA) 2018.* IEEE [Tor+18f].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Anne VDM Kayem, Feng Cheng, and Christoph Meinel. 2018. **A Cyber Risk Based Moving Target Defense Mechanism for Microservice Architectures**. *In Proceedings of the 16th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA) 2018.* IEEE [Tor+18c].

- Johannes Sianipar, Muhammad Ihsan Haikal Sukmana, and Christoph Meinel. 2018. **Moving Sensitive Data Against Live Memory Dumping, Spectre and Meltdown Attacks**. *In Proceedings of the 26th International Conference on Systems Engineering (ICSEng) 2018.* IEEE [SSM18].

- Michael Meinig, Muhammad Ihsan Haikal Sukmana, Kennedy Aondona Torkura, and Christoph Meinel. 2019. **Holistic Strategy-Based Threat Model for Organizations**. *In Proceedings of the 10th International Conference on Ambient Systems, Networks and Technologies (ANT) 2019.* Elsevier [Mei+19].

- Hendrik Graupner, Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, and Christoph Meinel. 2019. **Secure Deduplication on Public Cloud Storage**. *In Proceedings of the 4th International Conference on Big Data and Computing (ICBDC) 2019.* ACM [Gra+19].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Feng Cheng, and Christoph Meinel. 2019. **SlingShot - Automated Threat Detection and Incident Response in Multi Cloud Storage Systems**. *In Proceedings of the IEEE 18th International Symposium on Network Computing and Applications (NCA) 2019.* IEEE [Tor+19c].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Feng Cheng, and Christoph Meinel. 2019. **Security Chaos Engineering for Cloud Services: Work In Progress**. *In Proceedings of the IEEE 18th International Symposium on Network Computing and Applications (NCA) 2019.* IEEE [Tor+19b].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Feng Cheng, and Christoph Meinel. 2020. **CloudStrike: Chaos Engineering for Security and Resiliency in Cloud Infrastructure**. *IEEE Access Volume 8*. IEEE [Tor+20].

- Kennedy Aondona Torkura, Muhammad Ihsan Haikal Sukmana, Feng Cheng, and Christoph Meinel. 2021. **Continuous Auditing and Threat Detection in Multi-Cloud Infrastructure**. *Computers & Security Volume 102*. Elsevier [Tor+21].

2 CloudRAID: Secure Cloud Storage Solution

2.1 Background

2.1.1 Cloud Object Storage Services

Cloud object storage service is an Infrastructure-as-a-Service that provides long-term, highly available, and cheap storage for unstructured data on the cloud. It provides a key-value store interface to store arbitrary objects [Hua+15]. Cloud customers could store any amount or size of the data in the cloud where the data is stored as **objects** or **blobs** in the **buckets** or **containers**.

Generally, cloud object storage services employ the pay-per-use model. This means cloud customers are charged for the size of the objects stored in the cloud, the requests made to the buckets and objects, the amount of bandwidth used for the operations, and the type of storage options used, e.g., availability regions, encryption, or retention period [Ama20b; Goo20c].

There are three authorized methods to access the buckets and objects in the cloud storage service:

- **CSP management dashboard**: Cloud customers and entities in the Identity and Access Management (IAM) services, such as service account or user, could log in to the CSP management dashboard using their credentials to access the buckets and its objects.
- **Application Programming Interface (API)**: Buckets and their objects could be accessed through the API endpoints provided by the CSP. It requires CSP's software development kit (SDK), command-line interface, or HTTP(S) requests with CSP credentials, such as access key or username and password, to send authenticated requests to the API endpoints.
- **Signed URL**: Signed URL is a Uniform Resource Locator (URL) that provides limited temporary access to a specific resource in the cloud without revealing the CSP credentials [Gra+15]. It appends authentication information in the form of a security token or signature to the query string of

the URL, which is generated by signing the HTTP request using the CSP credentials [Ama21f; Goo21c].

Once the signed URL has been requested in the specified valid duration, the signature in the signed URL is verified by the CSP using the credentials stored in the CSP. If the signed URL is executed outside of the specified duration or has mismatched signature, then the request is denied [Gra+15].

This thesis focuses on the cloud object storage services provided by the three biggest CSPs on the market: Amazon Web Services Simple Storage Service (AWS S3)[4], Google Cloud Platform Storage Service (GCP Storage)[5], and Microsoft Azure Blob Storage Service (Azure Blob)[6].

Amazon Web Services Simple Storage Service

AWS provides its object storage service through Simple Storage Service (S3). It offers several storage class types, such as S3 Intelligent-Tiering for data with unknown or changing access patterns, S3 Standard-Infrequent Access for less frequently accessed and long-lived data, and Amazon S3 Glacier for long-term data archive [Ama21a].

AWS users could upload an unlimited amount of objects, and data volume with an individual object's size could be up to 5 terabytes. The objects are stored in the bucket with a unique name on the entire AWS S3's namespace [Ama21a]. The buckets and the objects are accessible using the CSP management dashboard, API, or the pre-signed URL. The pre-signed URL could be created with customized parameters with a validity period of up to 7 days [Ama20g].

Google Cloud Platform Storage Service

GCP provides its object storage service through the Storage service. There are four storage classes available: Standard Storage for frequently accessed data, Nearline Storage for infrequently accessed data, Coldline Storage for infrequently accessed data with lower availability and lower storage cost than Nearline Storage, and Archive Storage for data archiving, online backup and disaster recovery with the lowest cost [Goo21d].

[4] https://aws.amazon.com/s3/

[5] https://cloud.google.com/storage

[6] https://azure.microsoft.com/services/storage/

GCP users could upload an unlimited amount of objects, and data volume with an individual object's size could be up to 5 terabytes. The objects are stored in the bucket with a unique name on the entire GCP Service's namespace [Goo21f]. The buckets and the objects are accessible using the CSP management dashboard, API, or the signed URL, where it could be created with customized parameters and valid up to 7 days [Goo20e].

Microsoft Azure Blob Storage Service

Azure provides its object storage service through the Blob Storage (Blob) service. A storage account could contain all types of data objects, including the blobs stored in the containers in a storage account. The storage account's name needs to be unique within the Azure's namespace [Mic21a].

There are three types of blobs supported by Azure Blob: Block blob for a large amount of data up to 190.7 terabytes, Page blob for data with random read and write operations that support s up to 8 terabytes, and Append blob for data optimized for append operations with up to 195 gigabytes [Mic21b]. The storage account, containers, and blobs are accessible using the CSP management dashboard, API, and shared access signatures (SAS). Azure Blob does not allow for SAS creation with customized parameters [Mic20d].

2.1.2 Multi-Cloud Storage Approach

More companies are migrating their data from private data centers to the cloud following the increased adoption rate of cloud computing in the past few years. Storing the data in the cloud provides various advantages for the companies, such as guaranteed data availability, higher scalability, and less maintenance than private data center [Pet13; Sal21]. However, if the cloud storage service is inaccessible, e.g., due to cloud outage or bankruptcy, it could affect the data's availability and the service's reliability. For example, in 2017, the AWS S3 service went down for 4 hours due to a maintenance issue that brought down several servers in the Northern Virginia (US-EAST-1) region and caused several web services to be unavailable, and massive financial loss [Ama17].

The multi-cloud storage approach refers to the usage of different cloud storage services to store the data in the cloud [Raf+17]. It utilizes data redundancy techniques, e.g., erasure code, fragmentation, or replication, to store the data fragments or copies in cloud storage services from multiple public CSPs, private data centers or CSPs, or a hybrid combination of both [MTB18; Nac+17]. The approach ensures the data can still be retrieved from the cloud in case one or several CSPs are inaccessible and avoids vendor lock-in issue [Pet13].

2.2 CloudRAID as Secure Personal Cloud Storage Solution

CloudRAID is a research project collaboration between Hasso Plattner Institute gGmbH and Bundesdruckerei GmbH focusing on providing a secure storage solution in the cloud for personal usage. Dr. Maxim Schnjakin first researched it during his PhD. research to resolve the uncertainty and the privacy issues of storing data in the cloud [SAM10; Sch+13; SM13a; SM13b; SMM13]. It is then continuously developed and researched by Philipp Berger, Kennedy Torkura, Hendrik Graupner, and Muhammad Sukmana.

CloudRAID applies cryptographic and erasure techniques to the CloudRAID user's files on the client-side to generate multiple encrypted chunks, which are stored in cloud object storage services across various CSPs, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure (Azure). It provides data availability and confidentiality for users' files in the cloud as users' files can still be retrieved from the cloud as long as there are enough encrypted chunks retrieved from the CSPs even though one or several chunks might be irretrievable due to CSP outage. Only authorized CloudRAID users can decrypt multiple encrypted chunks using its corresponding file key, which is stored encrypted at rest in CloudRAID and can only be decrypted using the user's cryptographic keys stored on the user's device.

2.2.1 Actors and Threat Model

CloudRAID consists of three main entities:

- **Cloud service provider (CSP)**: CSP is a third-party entity that provides different types of cloud services for its customers. CloudRAID utilizes the object storage service provided by multiple CSPs to store multiple encrypted chunks of the user's files.
- **CloudRAID service**: CloudRAID provides a secure cloud storage service for personal usage. It is responsible for securely managing its multi-

cloud storage environment and user files stored on the cloud on behalf of CloudRAID users. All user's file access requests to the various CSPs have to be authorized by CloudRAID first before the requests are granted.

- **CloudRAID user**: CloudRAID user is the subscriber or customer of CloudRAID. The user utilizes the client application installed in the user's device to upload, download, delete, and share the files stored in the cloud.

The CSPs and CloudRAID are assumed to be honest-but-curious entities where it will follow the protocol and execute submitted requests; however, it might attempt to learn any information regarding the request and its contents.

CloudRAID users will only be capable of accessing their authorized files owned by them or shared by other users. However, the users could be malicious as they might try to access unauthorized files owned by other users.

2.2.2 Requirements

There are several requirements that CloudRAID needs to fulfill to provide secure personal cloud storage solution:

- **Multi-cloud storage management**: CloudRAID is responsible for managing the used cloud services and resources in multiple CSPs and ensure CloudRAID user's files are securely stored in various CSPs.

- **Zero-knowledge policy**: No other entities except the authorized CloudRAID users could access the files stored encrypted in the cloud.

- **Data availability**: CloudRAID user's files should always be available from the cloud.

- **Request brokerage**: All file access requests from the CloudRAID users to the cloud must be authorized by CloudRAID first.

- **Secure file-sharing and file-synchronization**: CloudRAID user's files must be securely shared with other CloudRAID users and synchronized to authorized CloudRAID user's device(s).

- **File access revocation**: CloudRAID users whose file access is revoked will no longer be able to access the files.

2.2.3 Erasure Coding

CloudRAID applies the principle of RAID (Redundant Arrays of Inexpensive Disks) to distribute user's files across multiple CSPs. It utilizes the erasure coding technique to create multiple file chunks for each file, ensuring the files can be recreated in case one or several file chunks are missing due to CSP outage or corrupted file chunk [SAM10; Sch+13; SM13a; SM13b; SMM13].

The erasure coding technique essentially divides a file into k equally-sized data chunks where the chunks are then used to calculate m parity chunks. The encoding process generates $n = k + m$ chunks where each chunk contains a number of words of length w. The original file can be recovered from any arbitrary k out of n chunks, whether it's all data chunks or a combination of both data and parity chunks [Pla+09; Sch+13].

CloudRAID is using the Cauchy-Reed-Solomon algorithm for the erasure coding due to two main reasons [Sch+13]: First, according to Plank et al. [Pla+09], the algorithm performs better compared to other erasure algorithms, such as classic Reed-Solomon, Row Diagonal Parity, and EVENODD. Second, it allows for customizable k and m parameters that can be configured following a number of CSPs available and managed by CloudRAID. However, the erasure coding leads to storage overhead of factor $\frac{m}{k}$.

CloudRAID users using the client application installed on the device will encode the file into multiple encrypted file chunks and upload all file chunks across various CSPs. When the user wants to access the authorized file, the client application will retrieve a sufficient number of encrypted file chunks from multiple CSPs to decode back into an encrypted file. If the retrieved number of encrypted file chunks is insufficient for the decoding process, the file access is then failed, and the client application could try again at later times [SAM10; Sch+13; SM13a; SM13b; SMM13].

2.2.4 Cryptographic Methods

CloudRAID utilizes encryption techniques to ensure data confidentiality for the user's files in the cloud from the CSPs, unauthorized users, and CloudRAID itself. It uses two types of the cryptographic algorithm: symmetric and asymmetric cryptography algorithms.

Symmetric encryption is a cryptographic algorithm that uses the same key for the encryption and decryption processes. CloudRAID uses AES-256 as the symmetric cryptographic algorithm to encrypt and decrypt a user's file using the cryptographic key called the **file key**.

Asymmetric encryption is a cryptographic algorithm that uses a cryptographic keypair consisting of a public key to encrypt the plaintext and a private key to decrypt the ciphertext. CloudRAID uses RSA in Electronic Codebook (ECB) mode with PKCS 1 Padding as the asymmetric cryptographic algorithm to encrypt and decrypt the file key to secure file key transport between CloudRAID users and devices and CloudRAID. Only authorized CloudRAID users and their devices can decrypt the encrypted file key, ultimately allowing for access to the file.

All cryptographic operations, i.e., key generation, encryption, and decryption, are happening on the client application installed in the user's device. Any sensitive information regarding the user and the file is sent in encrypted form through encrypted communication.

CloudRAID is responsible for securely managing all cryptographic keys used for all file operations in the system with its key management system. This includes storing various cryptographic key types in plaintext or encrypted form, distributing the cryptographic keys only to authorized CloudRAID users, and revoking the cryptographic keys for unauthorized CloudRAID users. There are four different types of the cryptographic key used in CloudRAID:

- **File key**: The symmetric cryptographic key used to encrypt and decrypt the user's file and derived from the user's file hash result using SHA-256. The different file key is used for each file for its cryptographic process. The file key is encrypted first with the account public key, account recovery key, and device public key before being stored in the CloudRAID.

- **Account keypair**: The asymmetric cryptographic keypair created using RSA when a CloudRAID user registers to the system. It is used to synchronize access to user's files across different CloudRAID user's devices. The keypair consists of an account public key and a private account key encrypted with an account recovery key using AES-256. Account public key and encrypted account private key are then stored in CloudRAID.

- **Account recovery key**: The symmetric cryptographic key derived from the hash result of CloudRAID user's password using SHA-256 hash algorithm. It allows the user to recover access to their files if their device is inaccessible, e.g., stolen or missing, by decrypting a private account

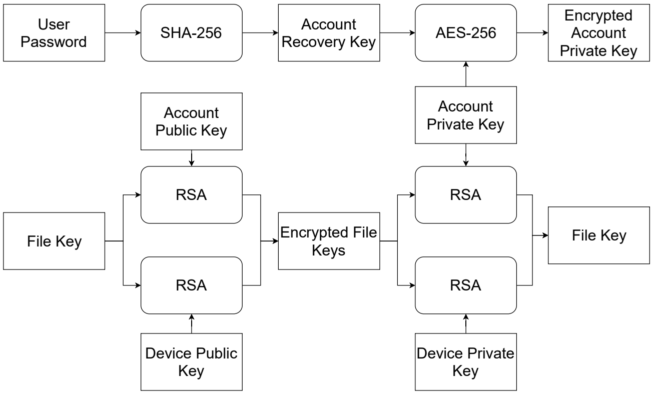


Figure 2.1: CloudRAID's cryptographic key hierarchy

key stored encrypted in CloudRAID. The encrypted account recovery key is stored in CloudRAID. If the users forget their CloudRAID's account password, they will not be able to access their files.

- **Device keypair**: The asymmetric cryptographic keypair created using RSA for each device owned by a CloudRAID user. It is used to access the user's encrypted file key specifically for the device. It consists of a device public key and a private key where the keypair is stored securely in the CloudRAID user's device.

2.2.5 Multi-Cloud Storage Management

CloudRAID follows the cloud brokerage approach [ST13] to implement a multi-cloud storage approach to provide a secure and highly available personal cloud storage solution. It acts as a third-party intermediary entity between the cloud object storage service from multiple CSPs, which CloudRAID subscribes to as the cloud customers, and the CloudRAID users. CloudRAID users do not need to manage their files stored in the cloud by themselves as CloudRAID already manages it on behalf of the users, where it stores multiple encrypted file chunks for each file across various CSPs.

There are two main challenges faced by CloudRAID to securely manage CloudRAID user's files stored across multiple CSPs [Gra+15]. First, CloudRAID users need to execute the file operations on their side, including directly interacting with the files stored on the cloud to ensure CloudRAID will not learn any information regarding user's files. Second, it needs to provide the user the access to upload, download directly, and delete encrypted file chunks on the cloud without any interference from CloudRAID and giving away the root credentials of the CSPs. The CSP root credentials should be stored securely by CloudRAID and not be used and given away to anyone. Suppose the CSP root credentials fall to unauthorized users, e.g., stolen or used by malicious CloudRAID employees. In that case, the credentials could be misused to unauthorized access CloudRAID user's files, which could ultimately interrupt the CloudRAID service.

Therefore, CloudRAID generates an access key from each CSP root credential, which is a security credential of an Identity and Access Management (IAM) account in the CSP that is used to authenticate the programmatic API calls to the CSP [Pla20a; Ser20b]. The access keys are more flexible than the CSP root credential. It could be revoked, rotated, and replaced without affecting the root credential, which could only be updated by changing its password. The access keys then allow CloudRAID to access the object storage services in multiple CSPs without directly using the CSP root credentials, which are stored securely and not used directly by CloudRAID [Gra+15].

The access keys also allow CloudRAID to implement the signed URL method to the file operations to grant CloudRAID users limited temporary access to their files in the cloud [Gra+15]. CloudRAID generates a collection of signed URLs using the access keys to allow the users to upload, download, and delete the encrypted chunks stored across multiple CSPs. The file operation process in CloudRAID using the signed URL method is as follow:

1. CloudRAID user first sends a file operation request to CloudRAID that includes the information of the file, its multiple encrypted file chunks, and file operation's type, i.e., file upload, file download, file delete.

2. Based on the user's file operation request, CloudRAID then generates a signed URL for each encrypted chunk in each CSP as follow:

    a) CloudRAID collects the necessary request parameters for each encrypted chunk, such as chunk name, bucket name where the chunk is stored, the validity duration, and the HTTP request type, e.g., PUT request for file upload.

b) Using the access key provided, these parameters are then concatenated and then signed with a signature algorithm, such as RSA with SHA-256 for GCP Storage, and HMAC-SHA256 for AWS S3 [Pla20b; Ser20a].

c) The parameters and the signature are then assembled to create the signed URL for each encrypted file chunk.

3. The collection of signed URLs are then sent to the CloudRAID user, where the user then executes the signed URLs to access the encrypted chunks in multiple CSPs.

4. As the signed URL is executed by the user, each CSP then verifies the request's validity by verifying the appended signature in the URL with the public key of the access key and checking the time limit specified in the URL. If the signature verification returns true and the execution time of the URL is still under the specified duration, then the CSP will allow the request. Else, if either the signature's verification is false or the signed URL is executed outside the specified duration, the request is then denied.

2.2.6 File Operations

CloudRAID combines the cryptographic and erasure methods for four file operations executed by the CloudRAID users: file upload, file download, file synchronization, and file share with different users and devices.

File Upload Process

When a CloudRAID user wants to upload a file, the file is first encrypted using AES-256 with a file key derived from the file's hash result using SHA-256. The file key is then encrypted using RSA with the collection of user's public keys, which consists of the user's account public key and device public key(s). Encrypted file keys are later stored in the CloudRAID's main server. Meanwhile, the encrypted file is fragmented into multiple encrypted chunks and then transmitted to various CSPs. Figure 2.2 shows the overview of the file upload process.

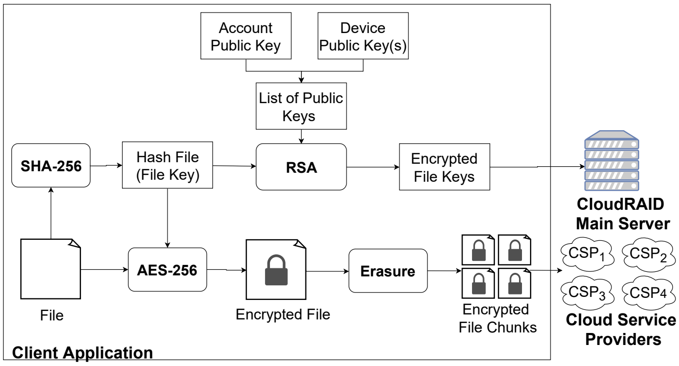


Figure 2.2: Overview of file upload process in CloudRAID [Suk+17]

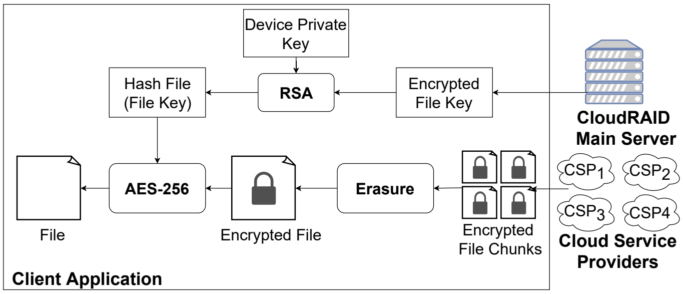


Figure 2.3: Overview of file download process in CloudRAID [Suk+17]

File Download Process

When an authorized CloudRAID user wants to access a file, a sufficient number of encrypted chunks are downloaded from various CSPs and reconstructed to an encrypted file. Encrypted file key for the particular user's device is then fetched from the CloudRAID main server and decrypted using RSA with the device private key. Finally, the file key is then used to decrypt the encrypted file using AES-256. Figure 2.3 shows the overview of the file download process.

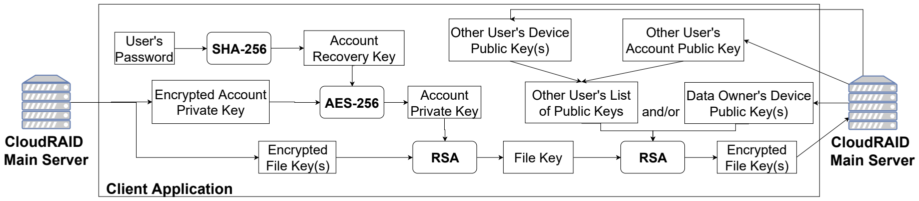


Figure 2.4: CloudRAID's scheme to support multi-user and multi-device scenarios by re-encrypting encrypted file key with new device public key or other user's public keys [Suk+17]

File Synchronization and Group File Sharing Processes

When a CloudRAID user uploads a file and shares a directory of the files with other CloudRAID users in a group, CloudRAID automatically generates additional encrypted file keys to ensure the files could be accessed by other authorized users CloudRAID users and their devices.

First, the data owner needs to download and decrypts the encrypted file keys of the files in the directory using RSA with their private account key. Once the encrypted file keys have been decrypted, or the file key of a new file has been generated, the file key(s) is then encrypted with the device public keys and account public keys of the other CloudRAID users in the group and data owner's device public key(s). The additional generated encrypted file keys are later stored in the CloudRAID main server. Finally, other CloudRAID users and their devices in the group or the other devices owned by the data owner will download and decrypt the encrypted file key(s) to access the files. Figure 2.4 illustrates the encrypted file key re-encryption process for multi-user and multi-device scenarios.

2.3 CloudRAID for Business as Secure Enterprise Cloud Storage Solution

With the demand for a secure cloud storage solution is increasing over the years, there is a strong need for enterprises to have a secure and scalable enterprise file synchronization and share (EFSS) system for companies and their employees. CloudRAID could fulfill the gap in the EFSS market as it could provide data confidentiality and availability for user's files stored in the cloud using the combination of cryptographic and erasure techniques.

CloudRAID for Business (CfB) is then introduced as an enterprise cloud storage solution based on CloudRAID concept to provide secure and scalable EFSS with Software-as-a-Service (SaaS) deployment model. It utilizes cryptography and erasure methods to provide data confidentiality and availability for the companies in the cloud by storing multiple encrypted file chunks across different CSPs.

2.3.1 Actors and Threat Model

CloudRAID for Business consists of five main actor types:

- **Cloud service provider (CSP)**: CSP is a third-party entity that provides different types of cloud services for its customers. CfB utilizes the Object Storage and Identity and Access Management (IAM) services provided by the CSP to store encrypted chunks generated from user's files.

- **CloudRAID for Business (CfB) service**: CfB service provides a secure cloud storage service for enterprise usage. It mediates the relationship between multiple CSPs and the companies and their employees as CfB customers and users, respectively.

- **CfB customer**: CfB customer is a company that subscribes to CfB to securely store its confidential files in the cloud. The company's administrator is the main actor responsible for managing the company's confidential files and the employees in the system using dedicated CfB customer's administrator dashboard.

- **CfB user**: CfB user is an employee of the company that subscribes to CfB. The user utilizes the client application installed in the user's device to upload, download, delete, and share the files stored in the cloud.

- **CfB employee**: CfB employees support the operational of the CfB service ensuring it works perfectly. There are three types of CfB employee considered in this thesis:

    - *CfB Administrator*: CfB administrator is responsible for managing CfB customers and employees in the system using administrator dashboard. One of the tasks of CfB administrator is to provision necessary cloud resources, its configurations, and access to authorized CfB stakeholders.

- *CfB Developer*: CfB developer's responsibility is to develop new features for CfB and test the system to ensure it is bug and error free.
- *CfB Security Auditor*: CfB security auditor's responsibility is to assess cloud resources owned by CfB in multiple CSPs.

CSPs and CfB are assumed to be honest-but-curious entities where it will follow the protocol and execute submitted requests; however, they might attempt to learn any information regarding the request and its content. CfB employees could be malicious as they might try to access the company's confidential files stored on the cloud or interrupt the CfB 's continuity and availability.

The company as a CfB customer is allowed to access the confidential company files that are uploaded to the cloud by its employees. The company's administrator is a trusted actor. They will not collide with the company's employees to generate unauthorized credentials that are not suitable to the employee's status in the company. However, a company may try to access other company's confidential files unauthorizedly.

The company's employees as CfB users will only be capable of accessing their authorized files owned by them or shared by other users depending on their role and status in the company. However, the users could be dishonest and may collide with other company employees to unauthorizedly access the company's confidential files. Also, an ex-company employee, or ex-CfB user, could try to access the files stored on the cloud unauthorizedly.

2.3.2 Requirements

There are several additional requirements on top of the CloudRAID's requirements mentioned previously that must be fulfilled by CfB to provide secure enterprise cloud storage service for its customers:

- **Secure multi-cloud storage management**: CfB is responsible for managing the used services and resources in multiple CSPs and ensure CfB customer's files are securely stored in multiple CSPs.
- **Central observer**: CfB should become a central authority entity observing the activities happening in the system and the cloud.

- **Zero-knowledge policy**: CfB and CSP should not be able to decrypt CfB customer's files stored on the cloud, even if they are colluding.
- **Self-sovereign authority**: Each company should have the authority to manage its files and employees without intervention from CfB.
- **Scalable key management system**: CfB should generate one encrypted file key per file for multiple users and devices in the company.
- **Revocation mechanism**: CfB should provide a scalable and fine-grained revocation mechanism where the revoked entity could not decrypt the encrypted file key.
- **Fine-grained file access control**: CfB should ensure the encrypted file and its encrypted file key can only be decrypted by its authorized CfB user and its devices following their roles and status in the company.
- **Flexible file-sharing**: CfB users should be able to share the files flexibly with other CfB users.
- **Collusion resistance**: Malicious CfB users must not be able to combine their secret keys to gain higher decryption power to decrypt unauthorized encrypted files and their encrypted file keys.
- **Backward security**: Revoked CfB user must not be able to decrypt the encrypted file key using their secret key.
- **Forward security**: New CfB user could decrypt the previously encrypted file key using their secret key.
- **Multi-cloud resource separation**: Each CfB stakeholder type should have its separate cloud resources across multiple CSPs according to the roles in the CfB system.
- **Multi-cloud access separation**: Each CfB stakeholder type should only be capable of accessing its authorized cloud resources across multiple CSPs according to the roles in the CfB system.
- **Location-based access control**: CfB customer's confidential files should be capable to be accessed only at certain pre-determined locations.

- **File activity monitoring**: CfB customer should be able to monitor the activities of its confidential files stored on the cloud.

2.3.3 Challenges

Although CloudRAID's mechanisms and architecture could provide a secure personal cloud storage solution, it could not fulfill the requirements needed by CloudRAID for Business to provide a secure cloud storage solution for enterprise usage. Therefore, four main challenges are identified in this thesis that needs to be resolved by CfB to provide a secure enterprise cloud storage solution.

Unscalable and Insecure Key Management System

As it has been explained previously, CloudRAID implements public key infrastructure using RSA cryptographic algorithm for its key management system (KMS). This allows for secure and quite efficient file sharing and synchronization between CloudRAID users and their devices. Although the mechanism is suitable for personal usage, it does not fulfill the requirements needed for enterprise file synchronization and share system due to several reasons.

EFSS system should guarantee secure and scalable file sharing and collaboration between the employees in the company. Only authorized employees listed in the file-sharing specification can access the company's confidential files [CLY17]. However, CloudRAID does not offer enterprise file-level access control based on the company's organizational structure, such as employee's roles and status, since it assumes that every CloudRAID user is equal in the system.

The KMS is not scalable for enterprise usage since it generates multiple encrypted file keys for each file stored in the cloud to ensure all authorized CloudRAID users and their devices can access the file. With a large number of files will be stored and managed by the CfB, an enormous number of encrypted file keys will be generated that will increase its management complexity level. The number of managed cryptographic keys should be kept to a minimum while ensuring all related key management processes are secure.

CloudRAID's KMS also would not allow the company to manage its confidential files and employees in the system since the cryptographic methods ensure the zero-knowledge policy where only authorized CloudRAID users can access the files. EFSS system should guarantee administrative oversight capability to the companies to access and manage its files and employees. Meanwhile, it should not be able to intervene with the company's activities in the system and learn any information regarding the company, its employees, and files to ensure the zero-knowledge policy in the system.

Lack of Enterprise File Access Control Enforcement

EFSS system should provide file access control functionalities needed by enterprises to ensure the company's confidential files could only be accessed by authorized employees under certain conditions. However, CloudRAID currently does not offer any enterprise file access control features since it assumes every CloudRAID user is equal in the system.

One of the file access control functionalities often offered by the EFSS system is ensuring the files could only be accessed at pre-determined trusted locations. User's location is first calculated using various methods, such as Global Positioning System (GPS) or IP address, and sent to EFSS system during file access request. The EFSS system then grants or denies the user's file access request by verifying the user's location to be in the trusted locations.

Insufficient Secure Multi-Cloud Storage Management Strategy for Enterprise Usage

CloudRAID utilizes a cloud brokerage approach that manages the relationship between CloudRAID users and multiple CSPs to store the encrypted file chunks of CloudRAID user's files in a single bucket of each CSP. It also utilizes an access key generated from the root credential of each CSP to generate the collection of signed URLs for the users to temporarily access their files stored across multiple CSPs with limited action. Although the approach employed by CloudRAID provides an efficient multi-cloud data storage strategy for personal usage, it might not be suitable for enterprise purposes for several reasons.

CloudRAID for Business (CfB) is responsible to securely manage the cloud resources across multiple CSPs for authorized CfB stakeholders, including the company's confidential files. Suppose the access keys generated directly from the CSP root credentials are stolen. In that case, it could be misused to gain complete access control to all cloud resources across multiple CSPs that threatens the confidentiality of the company's confidential files and the reliability of CfB .

CfB does not have a multi-cloud storage strategy for different CfB stakeholders yet without relying on the root access keys. Each CfB stakeholder should have its cloud resources with limited actions allowed to do for the resources across multiple CSPs depending on the role in the system. Meanwhile, the encrypted file chunks owned by the companies as CfB customers should not be stored in a single bucket of each CSP as it could increase the risk of the company's unauthorized access to other company's confidential files [Fac+13].

Absence of File Cloud Access Oversight

CloudRAID offers a zero-knowledge policy for its users where it could not learn any information of the CloudRAID users and their files. This includes the user's file activities in the cloud where CloudRAID does not actively monitor the activities across multiple CSPs. However, CloudRAID for Business should be aware of the user's file activities in the cloud for various reasons.

EFSS system must ensure the company that its confidential files are stored securely from unauthorized entities, especially with the increasing number of data breaches over the years. CfB then needs to actively monitor the activities happening on multiple CSPs to detect any suspicious or malicious activity happening. The company also needs to be aware of the activities happening by its employees in the CfB system ensuring only authorized employees could access the confidential files to reduce the risk of insider threat.

2.3.4 Competitors

Several enterprise cloud storage and enterprise file sync and share solutions mentioned in this thesis have similar characteristics and functionalities with CloudRAID for Business. The competitors help CfB determine the state-of-the-art mechanisms and functionalities offered by these solutions to provide a secure cloud storage solution for enterprise usage.

Dropbox Business

Dropbox Business[7] is the enterprise version of Dropbox[8], one of the most prominent personal cloud storage solutions on the market today. It provides one solution for all the company's needs for secure enterprise file storage and collaboration on the cloud. It supports various productivity and security services

[7] https://www.dropbox.com/business

[8] https://www.dropbox.com

and products for seamless collaboration anytime, anywhere, such as Microsoft Office, Slack, Zoom, or Trello. The company's files, users, and devices could be easily managed with complete visibility through the admin console. It also provides enterprise-grade security management to ensure secure file sharing and collaboration within the company's domain by providing single sign-on integration, audit logs with file event tracking as several of its features, and unlimited API access to security platform partners [Dro21e].

Tresorit

Tresorit[9] is a secure cloud collaboration platform for individual and enterprise usages based in Switzerland. It utilizes cryptographic methods and various techniques to provide end-to-end security and zero-knowledge privacy of files stored on the cloud and the device, such as remote data wipes for mobile devices and advanced link tracking. Microsoft Azure is used as the cloud backend storage to provide high data availability for its customers. For enterprise usage, Tresorit offers various functionalities that would allow companies to manage their confidential data, users securely, and devices through admin center dashboard, storage and sharing policies, and activity monitoring on the company domain [Tre20c; Tre21c].

Boxcryptor

Boxcryptor[10] is a German-based secure file storage solution for individual and enterprise purposes. It works on top of cloud storage providers used by the customers, which supports more than 30 cloud storage providers, to provide additional security layers to the files stored on the cloud, which might not be encrypted on the server-side. The customers are required to install cloud storage provider's software on the devices to ensure that would allow Boxcryptor to encrypt the files on the customer's side and stores the encrypted files on the used cloud storage providers [Box21a]. Boxcryptor provides Teams as the enterprise version that would allow companies to securely manage their files on their preferred on the cloud [Box21c].

[9] https://www.tresorit.com/

[10] https://www.boxcryptor.com/

3 Improving Scalability and Security of Key Management System

3.1 Introduction

Enterprise file synchronization and share (EFSS) systems are responsible for securely managing the company's confidential files stored on the cloud on behalf of the companies. Secure file-sharing and access control are two of the main functionalities that EFSS systems should provide for their users [CLY17]. The company's employees, as the EFSS users, should be able to securely share a file stored on the cloud with other authorized users in the system. Only the authorized users listed in the file-sharing specification can access the shared files in the company's domain.

Cryptographic operations could resolve these challenges for EFSS systems by encrypting the files before uploading them to the cloud and granting access to the cryptographic keys, which are used to decrypt the encrypted files, only to the authorized users. However, the management of cryptographic keys is critical and challenging due to a large amount of the managed files and the number of generated cryptographic keys to secure the files stored on the cloud [CIC14]. A key management system (KMS) is a system that is responsible for the lifecycle of the cryptographic keys, from its creation to deletion, and the usage of the keys for cryptographic operations in the system [FL93]. It is an essential component of EFSS systems that ensures data confidentiality in the cloud and secure file-sharing and access control in the system.

KMS plays an essential role in CloudRAID of managing the cryptographic keys used in the system to ensure secure file storage, sharing, and synchronization operations for CloudRAID users and their devices. CloudRAID utilizes the combination of AES and RSA algorithms for its key management system to secure the files stored on the cloud and the access to the files by encrypting the files and the cryptographic keys used to encrypt the files, or file keys, as explained in Chapter 2.2.4. However, as CloudRAID is shifting the focus from personal cloud storage solutions to enterprise cloud storage solutions, the cryptographic methods implemented in CloudRAID create several main challenges that make CloudRAID to be unsuitable for enterprise usage: unscalable key management system, company-level file access control, and the absence of company's administrative oversight over its confidential files.

In this chapter, a new key management system is implemented to resolve the challenges faced by CloudRAID for Business to provide secure EFSS system for the companies and their employees as CfB customers and users, respectively. It implements two attribute-based encryption (ABE) schemes [But18; Li+17] to replace the RSA-based KMS by proposing two system architectures. The ABE schemes provide scalable and secure KMS with attribute-based file access control since it generates one encrypted file key per file for multiple CfB users and their devices where only the authorized CfB users have the correct attributes that fulfill them the file-sharing specifications could decrypt the encrypted file key.

3.2 Related Works

3.2.1 Research Works

There are several researches focusing on solving the access control and scalability issues in the key management system for secure file-sharing.

Kallahalla et al. [Kal+03] introduced Plutus, a cryptographic storage system built on untrusted storage that provides highly secure and scalable key management. It groups files into a filegroup that allows file keys to be shared among files by generating a file-lockbox key for each filegroup. The file-block key is used to encrypt the file-block key(s) Each file is encrypted using the file-block keys, which are then encrypted using the file-block key. When the user's access to the filegroup is revoked, Plutus uses a lazy revocation scheme by generating a new file-lockbox key and re-encrypting the file-block keys with the latest file-lockbox. Shu et al. [SSX14] utilized a hierarchical key organization developed based on a variant of Merkle Hash Tree to reduce the complexity of managing three types of the symmetrical encryption key used in the Shield, a secure cloud storage system. The system also utilizes a proxy server) to manage access control and distribute secret keys used to encrypt the files to authorized users. It also supports concurrent file writing by using the virtual root hash linked list.

Chu et al. [Chu+14] proposed key-aggregate encryption for scalable data sharing in the cloud storage that aggregates multiple secret keys into a single constant-sized key that can be used to decrypt multiple encrypted files. [Bjo+18] tackled the issue of scalability in centralized KMS for cloud storage by using an untrusted distributed storage system with a key wrapping hierarchy scheme, where the key is encrypted with its parent or master key, to provide key rotation and secure data deletion. Li et al. [Li+10] introduced fine-grained and scalable access control for patient-centric electronic health record systems using multi-authority ABE (MA-ABE) and key-based policy ABE schemes implemented for two different security domains. Yang et al. [Yan+13] proposed an MA-ABE scheme called Data Access Control for Multi-Authority Cloud Storage Systems (DAC-MACS) to ensure secure data access control in a multi-authority cloud storage system.

[Li+14] solved the problem of the enormous key generated by convergent key encryption, which is used to provide secure deduplication by applying deduplication and secret sharing techniques to the convergent keys for secure and scalable key distribution across multiple servers. Kwon et al. [Kwo+17] presented a convergent encryption key management based on pairing-based cryptography for a secure deduplication scheme to reduce the number of keys generated after the files have been deduplicated. The convergent encryption key is divided into three components, and the file owner holds a common secret used for masking the key component distribution. The three key components could be transmitted publicly without the adversaries combining the convergent encryption key.

3.2.2 Competitors

Dropbox Business

Dropbox Business only offers encryption-at-rest for the company's files stored in the cloud using AES-256. The file encryption keys are created, stored, and managed by Dropbox on behalf of the users in a distributed manner to remove management complexity and enable advanced product features. Companies could integrate third-party digital rights management services to protect better company's confidential data, including client-side encryption [Dro20].

Dropbox client application sends the files to Dropbox Business using a TLS connection to provide secure data transit protection. Once Dropbox Business has received the files, block servers then process the uploaded files by Dropbox client applications by splitting each file into blocks and encrypting each file block using AES-256. Only file blocks that have been modified are synchronized to the cloud [Dro20].

Tresorit

Tresorit utilizes symmetric and public-key cryptographic methods to ensure secure zero-knowledge file sharing in the system. Each Tresorit user generates all cryptographic keys on the client-side, including the keypair where the private key is stored encrypted on the user's profile and the public key is distribute automatically by Tresorit using anonymized PKI certificates to protect the privacy of Tresorit user [Tre20b].

If a Tresorit user wants to upload a file, the file is encrypted using 32 bytes key called file key using AES-256 in CFB mode. The encrypted file is then added to the remote directory structure in the cloud, which is the exact copy of the structure of the client-side directory following an oriented tree graph. The file key for each file is stored in the Key Lock Box (KLB) that represents a node in the graph, whereas Master KLB represents the root. A file can only be uploaded to the cloud directory if the authorized user has access to the Master KLB [SBL17; Tre20c].

When a Tresorit user shares a tresor with another user, which is an encrypted sharable folder, the invitation is done following the Interactive Connectivity Establishment protocol [KHR18] to authenticate the invited user as the group member. After the invited user joins the group file-sharing, the inviter needs to share the symmetric key, which is used to encrypt the tresor that contains the folders, files, and its corresponding file keys [Tre20c]. There are two types of agreement module to share the key to the newly added group member [Tre20c]:

- **RSA-based Agreement Module**: The agreement module contains a set of the public certificates of all Tresorit users in the group file-sharing and a set of pre-master secrets where each secret is encrypted with the RSA public key of each group member. The invited user then decrypts the encrypted pre-master secret with their RSA private key. The user then derives the symmetric key by calculating the hash of all user's certificates as the input and the pre-master secret as the key using HMAC.

- **Tree-based Group Diffie-Hellman (TGDH)-based Agreement Module**: The module contains the user's Diffie-Hellman certificates instead of encrypted pre-master secrets. The symmetric key is then calculated using the Invitation TGDH (ITGDH) scheme by Szebeni et al. [SB+12] that uses a binary key-tree where the root node represents the established group key, and the leaf represents the group member. The ITGDH scheme allows a group member to invite a new member by using "shadow nodes" to make the binary key-tree balanced when the group membership changes. The inviter generates a temporal key for a shadow node and updates the group's public key by using the private key of the sibling node of the joining or leaving node. The joining member then can calculate the node's private key using key refresh operation, and the temporal key is then discarded. Then the group member can generate the group key, which is the symmetric key, using the private key of the leaf [SB+12].

When a user leaves the group file-sharing or is removed from the group, the revoked user's certificate is deleted from the Agreement Module with the root directory is re-encrypted with a new symmetric key generated by the user who invokes the removal process. And finally, the encrypted files in the tresor are re-encrypted using a lazy re-encryption principle using a new file key by encrypting the documents that have not been encrypted since the last group membership changes to reduce the computational cost [Tre20c].

Boxcryptor

Boxcryptor utilizes RSA and AES encryption schemes on the user's side to provide secure zero-knowledge file sharing. Each Boxcryptor user has an RSA-4096 keypair and an additional AES-256 key. The RSA private key is encrypted using AES with the password key derived from the password using the key derivation function PBKDF2 with HMAC-SHA512 with 10000 iterations and 24 bytes salt. If a user wants to upload a file, the file is first encrypted using AES with a randomly generated file key. The file key is then encrypted with the user's RSA public key, which is attached to the encrypted file. The encrypted file key could be decrypted using the user's RSA private key [Box20a; Box20b].

Boxcryptor users could share files between other users in the same company as a group for enterprise usage. When a user creates a new group file-sharing, each group has an RSA-4096 keypair, an AES-256 key, and a randomly generated membership key used to manage group membership. As a new user joins group sharing, the membership key will be encrypted with the new user's public RSA key and the group's AES key to speed up the sign-in process. The new user then decrypts the encrypted membership key using the user's private RSA key to decrypt the group's RSA private key. For the file-sharing in the group, the file key is encrypted with the group's RSA public key in addition to the file key encryption with the user's RSA public key, where the encrypted file key could be decrypted with the group's RSA private key [Box20a; Box20b].

Each company generates an RSA-4096 master keypair during company registration that protects against loss of access to the company's files. It could decrypt the private keys of all users in the company to gain access to the company's files or reset the user's password. The company's RSA public key is used to encrypt user's AES password key, which is used to encrypt user's RSA private key, allowing the company to change the user's password and access the user's files by decrypting the user's encrypted password key using the company's RSA private key. Only authorized company's administrator can access the company's master keypair, where its RSA private key is encrypted with the company's password key derived from the company's password using AES [Box20a; Box20b].

Boxcryptor's cryptographic methods used for secure intra-company file sharing are very similar to CloudRAID's cryptographic methods used for file sharing between a group of users and their devices. Although Boxcryptor could achieve zero-knowledge property and data confidentiality for Boxcryptor users, it could face the same key management system 's scalability issue faced by CloudRAID. It needs to store and manage multiple encrypted file keys per file for multiple Boxcryptor users in a group file-sharing for the enterprise usage scenario where the file key is encrypted multiple times with other users' RSA public key and group's membership key. It also uses at least seven different key types and stores different types of encrypted keys to ensure secure access in the system, making key management very complex, such as encrypted user's RSA private keys, encrypted group's RSA, and encrypted password keys. It could also affect the user experience of Boxcryptor as the Boxcryptor users are required to encrypt, decrypt, and transmit the encrypted keys to and from Boxcryptor.

3.2.3 Thesis Contribution

The work proposed in this chapter is different from the research community and competitors to provide a secure and scalable key management system for enterprise file synchronization and share system. CfB utilizes an innovative attribute-based encryption scheme for its key management system for secure and scalable file-sharing with one encrypted file key per file for multiple CfB users and their devices. Only the authorized CfB users with the correct attributes that fulfill the file-sharing specification could decrypt the encrypted file key. Based on the ABE-based key management system, two CloudRAID for Business system architectures are proposed to provide secure intra-company and inter-company file-sharing functionalities for companies and their employees.

3.3 Key Management Issues in CloudRAID

As explained in Chapter 2.2.4, CloudRAID utilizes public key infrastructure implemented in its KMS using the combination of RSA and AES algorithms to provide secure file storage and sharing for CloudRAID users and their devices. It encrypts the file using the file key using AES, which is generated from the file's hash value. The file key is then encrypted with the account public key(s) and device public key(s) owned by the CloudRAID user(s) and their device(s) using RSA during file encryption and file synchronization and sharing processes. Only authorized CloudRAID user(s) and their device(s) could decrypt the encrypted file keys using account private key and device private key and ultimately decrypt the encrypted file stored in multiple CSPs.

However, the cryptographic methods implemented in CloudRAID creates scalability and access control issues in its key management system (KMS) that make it unsuitable for enterprise usage. The implemented cryptographic method generates multiple encrypted file keys that grow linearly for multiple CloudRAID users and their devices involved in the file-sharing. The file owner needs to generate multiple encrypted file keys generated by the data owner for all authorized users and their devices that would create processing overhead and require more bandwidth to send the keys to CloudRAID [PYJ14]. This creates storage overhead as CloudRAID needs to store multiple encrypted file keys which are essentially the same file key used for encrypting and decrypting an encrypted file [LCH13]. For example, if Alice has 50 files and 3 devices then CloudRAID will store 200 encrypted file keys for all Alice's files. If Alice wants to share a file with 3 other users where each user has 2 devices, then it will store additional 9 file keys to ensure other users can access the files in their devices.

File access revocation is also an important aspect that must be provided by EFSS system so that the revoked user could no longer access the file shared by the file owner. CloudRAID enforces system-level file access revocation where the revoked user's access to the shared file owned by other user is deleted, such as removing the association between the revoked user and the shared file in the database and deleting the affected file in the revoked user's device(s). However, if the revoked user somehow is able to retrieve the encrypted shared file(s) from multiple CSPs and its encrypted file key(s) from CloudRAID, the user could still access the encrypted file(s) by decrypting encrypted file keys using the account and device keypairs stored in the device.

Another concern is that the security of the file key depends on the strength of the user's password, making it the weakest link in CloudRAID. The file key encrypted with the account public key could be decrypted with the private account key, which is encrypted by the hash result of the user's password using AES, to ensure file synchronization for CloudRAID user's devices. Assuming the attackers somehow could get access to the user's encrypted file keys and encrypted account private key, they could brute force the encrypted account private key by guessing the user's password where it depends on the length and complexity of the password [Kas20]. If the brute force attack is successful, the attacker could decrypt encrypted file keys to access the user's files.

Overall, an EFSS system should guarantee secure and scalable file sharing and collaboration between the employees in the company. Only authorized employees listed in the file-sharing specification can access the company's confidential files [CLY17]. Due to the large number of files stored and managed by the system, it should be able to keep the number of managed keys per file to a minimum while enforcing file-based access control using cryptographic methods while ensuring it does not affect the performance and the user experience in the system [CIC14; LCH13]. Each company should be able to access and manage its files and employees, while the system should not access the company's confidential files stored in the cloud. It also should guarantee that there will not be cross-company data leakage in the system where the company could unauthorizedly access other company's confidential data.

3.4 Attribute-based Encryption

Attribute-based encryption (ABE) is a public key cryptography scheme first proposed by Sahai and Waters as an extension of identity-based encryption called Fuzzy IBE scheme [SW05]. The ABE scheme views the identity as a collection of attributes where multiple entities could share the same attributes, such as employment status, sex, and age. It then utilizes the attributes to encrypt or decrypt the message instead of using identity used by identity-based encryption, e.g., e-mail address or name.

In general ABE scheme, the attribute is used for various purposes [LCH13; PYJ14]. Attribute authority (AA) entity utilizes the attribute to generate and distribute the keypair for the user to encrypt the message using public attribute key or decrypt the ciphertext using secret attribute key. The attributes are also used in the access policy or policy that consists of a set of attributes and logic gates, e.g. AND-gate, OR-gate, ==. The policy determines which entity can decrypt the ciphertext, i.e., only the entity with the correct attributes that fulfill the ciphertext's policy, otherwise, the ciphertext could not be decrypted. This allows ABE to achieve "one-to-many" and attribute-based access control (ABAC) properties where one ciphertext can be decrypted by multiple keys with the correct attributes.

Based on the policy usage, ABE could be categorized into key-based policy ABE (KP-ABE) and ciphertext-based policy attribute-based encryption (CP-ABE). KP-ABE is first introduced by Goyal et al. [Goy+06] where the policy is attached in the user's secret key that dictates which ciphertext can be decrypted where the message is encrypted using a set of attributes. Meanwhile, CP-ABE is first introduced by Bethencourt et al. [BSW07] where it utilizes the policy during the encryption process by attaching the policy in the ciphertext, while the user's secret key consists of the set of attributes owned by the user.

Essentially, ABE scheme consists of four processes [BSW07; Goy+06]:

- **Initialization**: A trusted attribute authority sets up the ABE scheme by generating a master key and public parameters based on the randomness of the pairing-based cryptography. The public parameters are shared with all participating entities to allow further ABE processes, such as key generation and encryption. The master key must be stored securely by the AA as it can be used to generate the secret key for the users.

- **Secret Key Generation**: The attribute authority generates a secret key for the users using the master key and the public parameters that would allow the users to decrypt the ciphertext. In the CP-ABE scheme, the secret key generation process requires the set of attributes owned by the user. Meanwhile, the KP-ABE scheme requires the access policy of the user to generate the secret key.

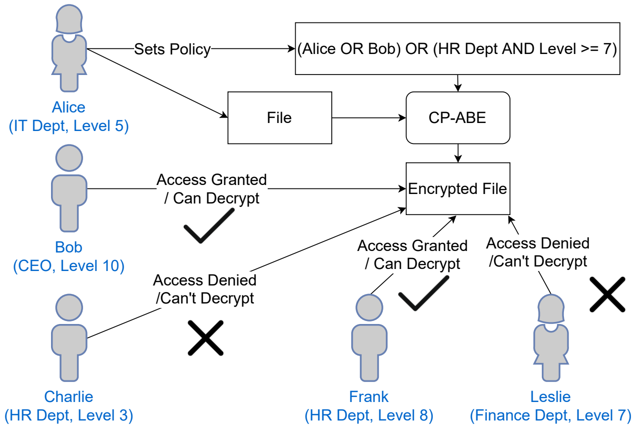


Figure 3.1: Example of how ciphertext-based policy attribute-based encryption (CP-ABE) scheme works

- **Encryption**: The user encrypts the plaintext using the public parameters and the specified access policy for the CP-ABE scheme or the set of attributes for the KP-ABE scheme to generate a ciphertext.

- **Decryption**: The ciphertext could only be decrypted when the secret key's set of attributes fulfills the ciphertext's policy for the CP-ABE scheme. For the KP-ABE scheme, the decryption is successful when the secret key's policy meets the ciphertext's set of attributes.

ABE can be categorized into single-authority ABE and multi-authority ABE based on the number of AAs in the scheme. The single-authority ABE scheme is first proposed by Sahai and Waters in the Fuzzy IBE scheme [SW05] where it only utilizes an AA in the system to set up the scheme, register the user in the system, and attribute and keypair creation and distribution to the user. However, they raised the question in the system whether it is possible to construct an ABE scheme where multiple independent AAs manage their own set of attributes and its keys instead of relying on a single authority.

Multi-authority attribute-based encryption (MA-ABE) scheme is first introduced by Chase [Cha07] to answer the question in [SW05]. A user could have the attributes assigned from different attribute authorities linked with the user's global identifier (GID) in the system. Central authority is responsible for setting up the scheme and registers a new user and attribute authority to the system where it could not issue attributes and keys for the user [Cha07; Li+17; Yan+13]. However, this scheme has a security vulnerability as it requires the user to fully trust the central authority where it holds a global decryption power that could unauthorizedly decrypt the ciphertext [CC09; LW11].

Lin et al. [Lin+08] proposed an extension of [Cha07] to construct the initial MA-ABE without a central authority entity scheme where multiple attribute authorities are working together to set up the system. Each user needs to register themselves, and their GID to each authority to be able to receive attributes and access the files [CC09; Lin+08; LW11]. By removing the central authority, it helps to prevent a single entity from having global decryption power and reduce the computational and communication costs [PYJ14].

3.5 Attribute-based Encryption for CloudRAID for Business

ABE scheme could resolve the key management system's scalability and access security issue faced by CloudRAID for Business to provide a secure and scalable enterprise file synchronization and share system. It allows CfB to generate and store one encrypted file key per file for multiple users and devices in the company instead of generating an encrypted file key per file for each user and each device due to the "one-to-many" ciphertext property. CfB could save a lot of storage by storing fewer encrypted file keys in the long run, where the number of users and files could significantly affect the number of encrypted file keys due to file-sharing between the users.

It also provides attribute-based file access control for CfB that is necessary for enterprise file access control [LCH13; PYJ14]. Only authorized CfB users with the correct attributes satisfy the file-sharing restriction imposed by the file owner could decrypt the encrypted file key. The attributes used in the ABE scheme could be derived from the company's organizational structure, such as department and job title.

The ABE scheme affects the description and responsibilities of the three main entities in the CfB system's architecture as shown in Figure 3.2:

- **CloudRAID for Business (CfB) main server**: CfB main server, or CfB,

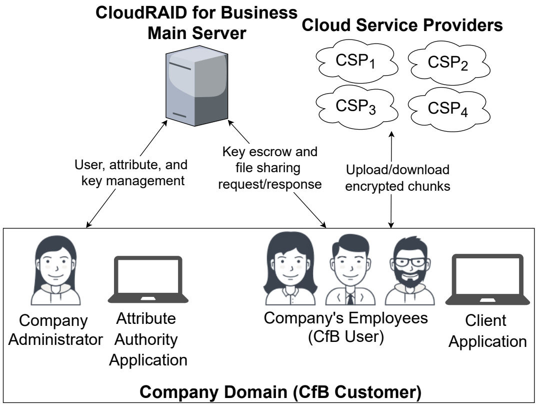

CloudRAID for Business Main Server
Cloud Service Providers
CSP$_1$
CSP$_2$
CSP$_3$
CSP$_4$
User, attribute, and key management
Key escrow and file sharing request/response
Upload/download encrypted chunks
Company Administrator
Attribute Authority Application
Company's Employees (CfB User)
Client Application
Company Domain (CfB Customer)

Figure 3.2: Overview of CloudRAID for Business' architecture

is the central server that handles the registration of the company and its employees as the CfB customers and users. It is also responsible for storing users' information, devices, and files to ensure file-sharing functionality in the system.

- **AA application**: Each company as CfB customer has its attribute authority application operated by a trusted company's administrator through the administrator dashboard. It is responsible for managing the registration and revocation of the company's employees, devices, and attributes.
- **Client application**: A company employee as a CfB user is using a client application installed in the user's device to upload, download, and share the company file with other CfB users.

3.6 Single Company Usage for CloudRAID for Business

Single-Authority CloudRAID for Business (SA-CfB) system is proposed to provide secure and scalable file-sharing between the employees in the company or intra-company file-sharing. The EFSS system must be able to provide file-level access control in the company ensuring only the authorized employees could access the company's confidential files.

Ciphertext-based policy attribute-based encryption (CP-ABE) scheme is chosen instead of the key-based policy attribute-based encryption (KP-ABE) scheme for CfB because CP-ABE allows the data owner to impose the file access control by encrypting the file key with the policy that determines which user can decrypt the encrypted file key [BSW07]. With the policy that could change over time, CP-ABE allows CfB to enforce simpler file access control management than the KP-ABE scheme since the policy attached to the encrypted file key is easier to manage than the policy attached to the user's key.

The ABE scheme is implemented for the CfB's key management system to replace CloudRAID's KMS as explained in Chapter 2.2.4. The RSA algorithm is still used by CloudRAID for Business (CfB)'s cryptographic methods for the secure key delivery mechanism by key wrapping the device's secret attribute key with the device's RSA public key. It ensures that the encrypted device's secret attribute key could only be accessed by authorized CloudRAID for Business (CfB) user's device using the device's RSA private key.

The jTR-ABE library developed by Artjom Butyrtschik [But18] is chosen for Single-Authority CloudRAID for Business (SA-CfB) system to provide secure and scalable intra-company file-sharing functionality. The library implements single-authority CP-ABE schemes *Practical Attribute-Based Encryption: Traitor Tracing, Revocation, and Large Universe scheme* by Liu and Wong [LW16] and *A Framework and Compact Constructions for Non-monotonic Attribute-Based Encryption* by Yamada et al. [Yam+14].

The jTR-ABE library supports a large attribute universe and requires the limited number of users to be specified during the setup process throughout the system's lifetime [Zic+16]. This allows the library to determine the malicious user that creates a decryption blackbox using its private key using special iterated encryption and decryption challenges outside the trusted attribute authority [LW16; Zic+16]. It utilizes the direct revocation mechanism where the list of revoked keys is included during the encryption process. If a key is listed in the list of revoked keys, although the key's attributes fulfill the ciphertext's policy, the key will not be able to decrypt the ciphertext [But18]. It provides a simple and expressive policy with threshold or boolean formulas and numerical, and geolocation attributes due to non-monotonic access structure [But18; Yam+14]. It also provides a fully collusion-resistant property that prevents users from

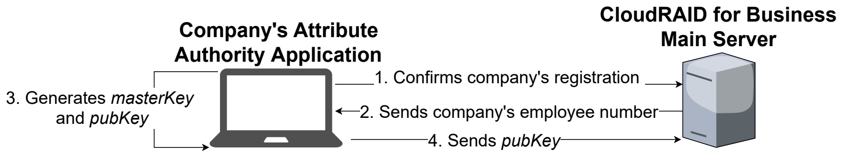

Company's Attribute Authority Application
CloudRAID for Business Main Server
1. Confirms company's registration
2. Sends company's employee number
3. Generates *masterKey* and *pubKey*
4. Sends *pubKey*

Figure 3.3: Company registration and system setup processes in the SA-CfB system [Suk+17]

colliding and combining their keys to fulfill the ciphertext policy using the identification number embedded in the key.

The SA-CfB system follows the system architecture and assumptions explained in Chapter 3.5. It consists of five processes: company registration and system setup, user and device registrations, file upload, file download, and user key revocation.

3.6.1 Company Registration and System Setup

When a company registers as CfB's new customer, the company first is required to specify the number of employees that will use the CfB system. Based on the number of the company's employees, the company's attribute authority (AA) application generates a master key ($masterKey$) and a public key ($pubKey$). The $masterKey$ is securely stored locally in the application that is used to generate the secret key and secret component necessary for CfB users and their devices. The $pubKey$ is used by CfB users to encrypt the file key, which is then sent to CfB main server to be distributed to the company's employees as CfB users. Figure 3.3 shows the company registration and system setup processes in the CfB system.

3.6.2 User and Device Registration

The company's administrator using its AA application registers a company's employees as a new CfB user to the CfB. CfB confirms the registration by sending a new user ID specifically for the company ($userID$). When a CfB user starts using the client application on their new device, the device generates the RSA key pair ($RSAKeyPair$). It then sends a new device registration request and device's

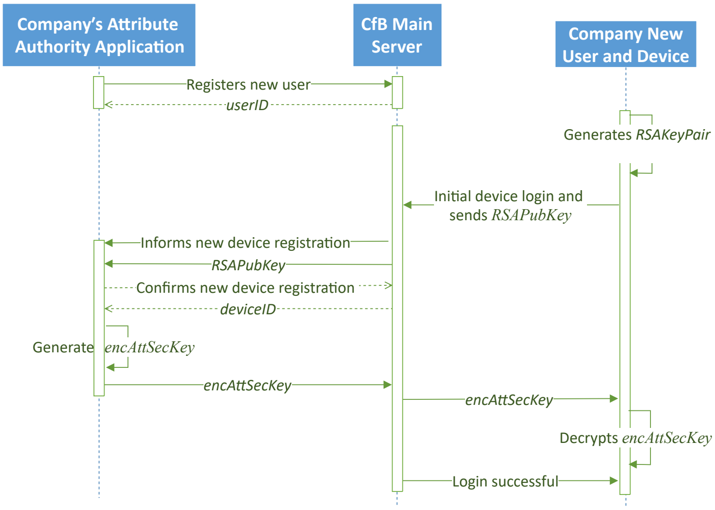

Company's Attribute Authority Application
CfB Main Server
Company New User and Device
Registers new user
userID
Generates RSAKeyPair
Initial device login and sends RSAPubKey
Informs new device registration
RSAPubKey
Confirms new device registration
deviceID
Generate encAttSecKey
encAttSecKey
encAttSecKey
Decrypts encAttSecKey
Login successful

Figure 3.4: Sequence diagram of user and device registration processes for the SA-CfB system [Suk+17]

RSA public key (*RSAPubKey*) to the company's administrator for registration confirmation.

When the company's administrator has confirmed the registration, CfB generates the device ID (*deviceID*) and sends it to the company's AA application. The AA application then utilizes the company's *masterKey*, the employee's list of attributes (*userAtts*), and *userID* to generate device's attribute secret key (*attSecKey*) and secret component (*secComp*), which is stored securely in the AA application to generate additional *attSecKey*.

Device's *attSecKey* is later encrypted with the device's *RSAPubKey* as encrypted attribute secret key (*encAttSecKey*) and sent to the new device with the company's authority public information. The client application then decrypts *encAttSecKey* with the user's RSA private key (*RSAPrivKey*) to recover *attSecKey* and store it securely in the CfB user's new device. Figure 3.4 shows the sequence diagram of the user and device registration processes.

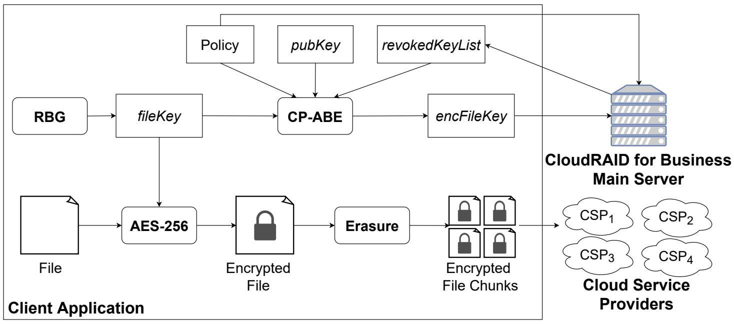


Figure 3.5: File upload process in the SA-CfB system [Suk+17]

3.6.3 File Upload

When a CfB user as the file owner wants to upload a file, 32 bytes file key ($fileKey$) is first generated with a random bit generator (RBG) that is used to encrypt the file with AES-256. The encrypted file ($encryptedFile$) is then processed with erasure coding to generate multiple encrypted file chunks, which will be uploaded to multiple CSPs. Meanwhile, the list of revoked keys ($revokedKeyList$) is retrieved from CfB main server that will determine which CfB user could not decrypt the ciphertext. $fileKey$ is then encrypted using CP-ABE with the policy that determines which CfB user could decrypt the ciphertext, $pubKey$, and $revokedKeyList$ to generate encrypted file key ($encFileKey$). Lastly, $encFileKey$ and policy are then sent to CfB. Figure 3.5 shows the file upload operation.

3.6.4 File Download

When a CfB user wants to download a file, CfB first sends $encFileKey$ to the user to be decrypted with CP-ABE using the device's $attSecKey$. If the device's $attSecKey$ is not listed in the $revokedKeyList$ and its attributes fulfill the $encFileKey$'s policy, $encFileKey$ is then encrypted to retrieve $fileKey$. Finally, $fileKey$ is then used to decrypt $encFile$ with AES-256, which is assembled from multiple file chunks retrieved from multiple CSPs. Figure 3.6 illustrates the file download operation.

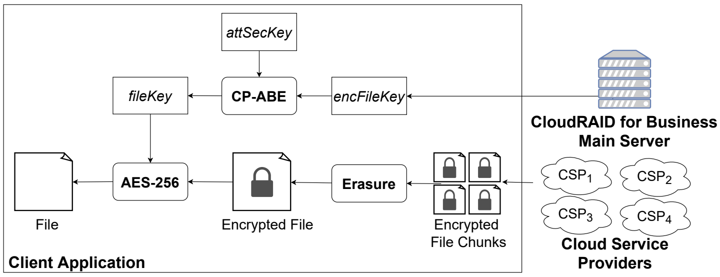


Figure 3.6: File download process in the SA-CfB system [Suk+17]

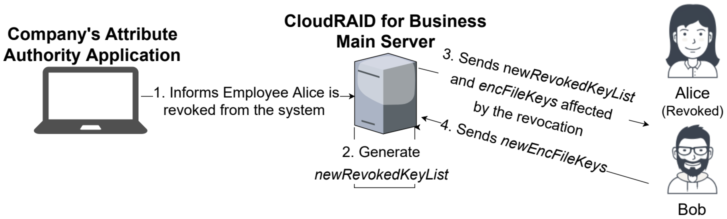


Figure 3.7: Example of Alice's user revocation process from the SA-CfB system by the company administrator [Suk+17]

3.6.5 User Key Revocation

When a company wants to revoke a CfB user from the system, e.g., due to the employee's departure from the company, the company informs CfB that the user is revoked from the system. CfB then generates the new revoked key list of the company (*newRevokedKeyList*) and lists all *encFileKeys* that are encrypted with the same attributes owned by the revoked user. It then distributes *newRevokedKeyList* to all non-revoked users. All *encFileKeys* that can be decrypted by the revoked user are sent to non-revoked users authorized to decrypt it. Later, *encFileKeys* are re-encrypted with the *newRevokedKeyList* to generate new encrypted file keys (*newEncFileKeys*) and sent back to CfB. Figure 3.7 shows an example of the user revocation process.

3.7 Multi-Company Usage for CloudRAID for Business

There is a demand for multiple companies subscribing to CfB to collaborate and share their files, or inter-company file-sharing. Only authorized users of the collaborating companies could access the shared files. However, the SA-CfB implementation in Chapter 3.6 is not suitable to provide secure and scalable inter-company file sharing due to several reasons.

The single-authority CP-ABE scheme by Liu and Wong [LW16] used in the jTR-ABE library [But18] runs on a separate large attribute universe for each company that acts as the attribute authority entity managing its users and attributes in its domain. This affects CfB users of the company to be unable to access the attributes and encrypted file keys from other companies. It also forces CfB to store an encrypted file key per file for each company participating in the file-sharing to ensure authorized companies and their employees could access the shared file. This could affect the CfB's key management system and attribute-based file access control enforcement for multiple companies to be more complex and not very scalable.

Another option to provide scalable KMS for SA-CfB system is to allow company's attribute authority (AA) application to generate attributes and keys for other companies. Although this enables CfB to store an encrypted file key per file for multiple companies, it does not give the participating companies the authority to manage the given attributes and keys where it could be revoked by the inviting company anytime.

The SA-CfB system could be in a situation where there is no longer an available key in the system as the employees have already used up all the available keys. If this situation happens, this will force SA-CfB to reject new CfB user registration in the company. Another alternative is to force the company to re-setup the system with an enormous number of keys. This causes all encrypted file keys of the company's encrypted files to be re-encrypted using the newly generated public key. The process affects the system's continuity and risks leaking the file key to unauthorized users during the re-encryption process.

A multi-authority attribute-based encryption (MA-ABE) scheme could solve the scalability issues of file-sharing between multiple companies explained above. It allows multiple attribute authority entities to co-exist in the same attribute universe while giving each AA entity to manage its attributes and keys. A user could have attributes assigned by multiple AA entities to access the data encrypted with the policy with the attributes from various AA entities [Cha07; Li+17; Yan+13].

Two-Factor Data Access Control With Efficient Revocation for Multi-Authority Cloud Storage Systems (TFDAC-MACS) scheme by Li et al. [Li+17] is then chosen to provide secure and scalable inter-company file-sharing functionality in the CfB system. It is a multi-authority CP-ABE type that requires the central authority to set up the system and register the attribute authority and the user to the system. Each AA manages the attributes and the users in its domain, where each user could have attributes assigned by multiple AAs. It could generate one encrypted file key that could be accessed by the users from multiple companies acting as the AAs, given that the user fulfills the ciphertext's policy. It supports the AND_m access policy, which is using only AND-gate on the list of attributes to generate constant-size ciphertext with a small computational cost. It provides a double-level indirect revocation mechanism as the AA entity updates the user's secret key who shares the same attribute(s) as the revoked user. The ciphertexts, which are encrypted with the attributes of the revoked user, will be re-encrypted using the update key that an untrusted party can do without revealing any information of the plaintext. The revoked user will not receive the secret key update from AA, thus unable to decrypt the newly re-encrypted ciphertext [Li+17].

The TFDAC-MACS scheme is chosen since it could help CfB's KMS be more scalable and attribute-based file access control across companies easier to enforce for different companies and their employees. The scheme does not allow the central authority to have global decryption power as it is only responsible for initializing the system and registering attribute authority and the users in the system. This is because attribute keypair and authority keypair generation processes are done in the AA side, where the private keys are stored securely. It is suitable for CfB where it will act as the central authority entity to set up the system without the capability of accessing the company's confidential files. Each company as CfB's customer acts as an attribute authority entity that manages the users, files, and attributes in the company domain. The scheme also provides a unique functionality of two-factor authorization for the ciphertext that requires a user to have the correct attributes and an additional authorization key generated by the file owner to decrypt the ciphertext. If a user fulfills the ciphertext's policy but does not possess the authorization key, the user would not be able to decrypt the ciphertext [Li+17].

As TFDAC-MACS scheme is implemented for the CfB, the scheme is not fully compatible with CfB's mechanisms and requirements as mentioned in Chapter 2.3.3. Therefore, three modifications to the scheme are proposed and implemented into a library called Practical Applied Distributed-TFDAC-MACS (PAD-TFDAC-MACS) [Pet19a] to fit the CfB's mechanisms and requirements without affecting the scheme's security proof [Pet19b]:

- **Disjunctive normal form access policy extension**: TFDAC-MACS scheme uses the AND_m access policy for encrypting a message by using only AND-gate on multiple attributes, e.g., "IT Department AND Senior Employee". This allows for constant-size ciphertext with small computational cost [Li+17]. However, it lacks the policy's expressiveness that affects the file-sharing in the system as the user can only decrypt the ciphertext where all attributes of its policy must be satisfied.

  The AND_m access policy can be trivially extended to support the disjunctive normal form (DNF) policy where it allows the OR-gate in the access policy to provide the n-of-m threshold. If the policy contains the OR-gate, the policy is split into multiple parts based on the number of node children and encrypts the message with the policy parts to generate multiple ciphertexts. CfB user only needs to decrypt one of the ciphertexts to recover the message due to the properties of OR-gate.

  The modification affects the ciphertext's size, depending on the number of policy parts used during the encryption process. It also creates a small overhead during the decryption process as it is trying to find the correct ciphertext out of all ciphertexts that the user's keys can decrypt. Unfortunately, the DNF policy extension only works for the policy with the OR-gate located at the root policy node.

  For example, suppose a message is encrypted with the policy "IT Department OR HR Department". In that case, the policy is then split into two parts: "IT Department" and "HR Department". Each part is then used to encrypt the message to generate two ciphertexts. The user then tries to decrypt both ciphertexts until one of the ciphertexts is decrypted.

- **Dynamic attribute keypair generation**: TFDAC-MACS scheme requires the attribute authority entity to define its attribute domain during its setup process to generate the public key and master secret key, which consist of public and private attribute keys, respectively [Li+17]. However, the attribute key pair, i.e., public attribute key and private attribute key, does not need to be generated during the attribute authority setup process as attribute authority's other concurrent processes do not use it. The process limits the dynamic of the attributes in the attribute authority's domain where the number of attributes could change over time, e.g., due to the change in the company's organizational structure.

Therefore, the attribute keypair generation process is made to be independent of the attribute authority's setup process to make the attribute domain in the attribute authority to be more dynamic. If CfB user's attribute keypair request contains unknown attributes not listed in the attribute domain, attribute authority generates a new random $y_{aid_i,j} \in Z_p^*$ for each unknown attribute $v_{aid_i,j}$. Attribute authority then can generate for each attribute public attribute key $UPK_{v_{aid_i,j}} = g^{y_{aid_i,j}}$ and private attribute key $USK_{v_{aid_i,j}} = y_{aid_i,j}$. Finally, the public attribute key is sent to the central authority to be added to the list of public attribute keys while private attribute key is used to generate user's secret attribute key.

- **Optional two-factor authorization constraint**: TFDAC-MACS scheme offers a unique security property of two-factor authorization for the ciphertext that ensures only the user with the correct attributes and corresponding authorization key generated by the data owner can decrypt the file. The file owner is required to specifically generate and securely send an authorization key for each user listed in the file sharing's specification [Li+17]. However, this security property could make the key management system less scalable and complicate file sharing between users since it increases the number of keys that need to be managed in the system to ensure secure file sharing between users.

  The two-factor authorization functionality is made to be an optional feature in the PAD-TFDAC-MACS library by removing the two-factor component α used in the encryption, decryption, and ciphertext update processes. This allows the CfB user to encrypt the message with or without the two-factor authorization functionality that will affect the subsequent decryption and ciphertext update processes. The modification does not affect the security of the TFDAC-MACS scheme as it is proven as follow:

  - **Encryption**: Only the C_3 component of the ciphertext is updated that contains the two-factor component α. The original C_3 component:

$$C_3 = \Big(\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}} \Big)^{s+\alpha}$$

, which consists of a randomly chosen value $s \in \mathbf{Z}_p^*$, a randomly chosen value $y_{aid_i,j} \in \mathbf{Z}*_p$ for each attribute value $v_{aid_i,j} \in S_{aid,i}$ by attribute authority AA_{aid}, is adapted to new $\widehat{C_3}$ component:

$$\widehat{C_3} = \Big(\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}} \Big)^{s}$$

The data owner ID *aid* component is removed from the ciphertext description, which is only needed for authentication key update process.

- **Decryption**: The user does not need authorization key $SK_{uid,oid}$ and public key component UPK_W anymore to decrypt a ciphertext. The decryption equation is then updated to

$$m = \frac{C_1 \cdot e(H(uid), \widehat{C_3})}{e(C_2, SK_W)}$$

The new decryption equation above does not threaten the original TFDAC-MACS' security scheme [Li+17] as it will resolve to the original decryption equation as follow [Pet19b]:

$$\begin{aligned} m &= \frac{C_1 \cdot e(H(uid), C_3)}{e(C_2, SK_W) e(SK_{uid,oid}, UPK_W)} \\ &= \frac{C_1 \cdot e\Big(H(uid), \Big(\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}} \Big)^{s+\alpha} \Big)}{e(C_2, SK_W) e(H(uid)^{\alpha}, \prod_{v_{aid_i,j} \in W} UPK_{v_{aid_i,j}})} \\ &= \frac{C_1 \cdot e\Big(H(uid), \Big(\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}} \Big)^{s+\alpha} \Big)}{e(C_2, SK_W) e(H(uid)^{\alpha}, \prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}})} \end{aligned} \quad (3.1)$$

$$
\begin{aligned}
&= \frac{C_1 \cdot e\Big(H(uid), \Big(\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}} \Big)\Big)^{s+\alpha}}{e(C_2, SK_W) e(H(uid), \prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}})^{\alpha}} \\
&= \frac{C_1 \cdot e\Big(H(uid), \Big(\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}} \Big)\Big)^{s}}{e(C_2, SK_W)} \\
&= \frac{C_1 \cdot e\Big(H(uid), \Big(\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}} \Big)^{s}\Big)}{e(C_2, SK_W)} \\
&= \frac{C_1 \cdot e(H(uid), \widehat{C}_3)}{e(C_2, SK_W)}
\end{aligned}
$$

- **Attribute revocation**: The original the ciphertext update key CUK [Li+17] could be updated to

  $$\widehat{CUK}^{ID_W}_{v_{aid_i,j}} = (g^s)^{y'_{aid_i,j} - y_{aid_i,j}}$$

  where $y_{aid_i,j}, y'_{aid_i,j} \in \mathbf{Z}_p^*$ are the current and new master secret keys for the attribute value $v_{aid_i,j}$, respectively. The $\widehat{C}_3$ component of the new ciphertext will not affect the decryption process with the same calculation as in the Equation 3.1 as proven as follows [Pet19b]:

  $$
  \begin{aligned}
  \widehat{C}'_3 &= \widehat{C}_3 \cdot \widehat{CUK}^{ID_W}_{v_{aid_i,j}} \cdot \Big(\prod_{v_{aid_t,j} \in W, v_{aid_t,j} \neq v_{aid_i,j}} g^{y_{aid_i,j}} \Big)^{r} \\
  &\cdot (g^{y'_{aid_i,j}})^{r} \\
  &= \Big(\prod_{v_{aid_t,j} \in W, v_{aid_t,j} \neq v_{aid_i,j}} g^{y_{aid_i,j}} \Big)^{s+r} \cdot (g^{y'_{aid_i,j}})^{s+r}
  \end{aligned}
  \tag{3.2}
  $$

Multi-Authority CloudRAID for Business (MA-CfB) system is then developed using PAD-TFDAC-MACS library to provide secure and scalable inter-company file sharing functionality. The system follows the system architecture and assumptions as explained previously, where CfB main server has two additional roles to ensure secure and scalable file sharing for multiple companies:

- **Central Authority (CA)**: CfB main server initially sets up the system by generating and distributing initial necessary parameters to be used for all authorized entities in the system.

- **Central Key Repository (CKR)**: CfB main server stores public authority key and list of public attribute keys of each company to facilitate inter-company file-sharing functionality. It also distributes encrypted file keys and the device's encrypted secret attribute keys to the authorized CfB user and their device(s).

The MA-CfB system consists of seven main processes: system setup, company registration, user and device registrations, file upload, file download, attribute, and user revocations, and authorization key revocation.

3.7.1 System Setup

CfB main server initiates PAD-TFDAC-MACS' setup process to generate global public parameters (GPP). The parameters are later distributed to CfB's customers and users.

3.7.2 Company Registration

As a company registers to be a new CfB customer, it first receives GPP and the company identifier ($companyID$) from the CfB main server. The company's administrator then uses its attribute authority (AA) application to generate authority key pair ($authKeyPair$) and list of attribute key pairs ($attKeyPairs$) from $GPP, companyID$, and the list of attributes managed by the company ($companyAtts$), which could be retrieved from the company's database or prompted by the administrator. $authKeyPair$ and $attKeyPairs$ are stored securely in the AA application while the list of attribute public keys ($attPubKeys$), authority public key ($authPubKey$), and $companyAtts$ are sent to CfB.

3.7.3 User and Device Registration

The company's administrator using its AA application to register a company's employees as a new CfB user to the CfB. CfB confirms the registration by sending a new user global ID ($userID$). When a CfB user starts using the client application on their new device, the device generates the RSA key pair ($RSAKeyPair$) and ownership key pair ($ownerKeyPair$) using GPP. It then sends a new device

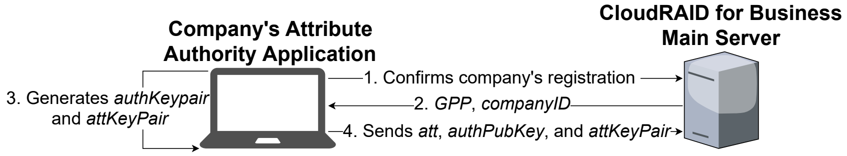

Company's Attribute Authority Application
CloudRAID for Business Main Server
1. Confirms company's registration
2. GPP, companyID
3. Generates authKeypair and attKeyPair
4. Sends att, authPubKey, and attKeyPair

Figure 3.8: Company registration process in the MA-CfB system [Suk+19a]

registration request, which consists of the device's RSA public key ($RSAPubKey$) and ownership public key ($ownerPubKey$), to the company's AA application.

When the company's administrator has confirmed the registration, CfB generates a new device global ID ($deviceID$) and sends it to the company's AA application. After that, the AA application generates the device's attribute secret key ($attSecKey$) based on the employee's list of attributes ($userAtts$), $companyID$, $userID$, $deviceID$, and attribute private key ($attPrivKey$).

Device's $attSecKey$ is later encrypted with the device's $RSAPubKey$ as encrypted $attSecKey$ ($encAttSecKey$) and sent to the new user's device together with the company's authority public information. Upon receiving the $encAttSecKey$, the client application then decrypts it with the user's RSA private key ($RSAPrivKey$) to recover $attSecKey$. Figure 3.9 shows the sequence diagram of the user and device registration processes.

3.7.4 File Upload

When a CfB user as the file owner wants to upload a file, 32 bytes file key ($fileKey$) is first generated with a random bit generator (RBG) that is used to encrypt the file with AES-256 algorithm. The encrypted file is then processed with erasure coding to generate multiple encrypted file chunks, which will be uploaded to multiple CSPs. Meanwhile, $fileKey$ is encrypted with the access policy that determines who can decrypt the ciphertext, file access specification that consists of a list of attributes and its $attPubKey$ used in the policy, and $authPubKeys$ of the involved companies in the file sharing, to generate encrypted file key ($encFileKey$).

If the file owner enables two-factor authorization protection for the file, ownership private key ($ownerPrivKey$) is also used during file key encryption. Later,

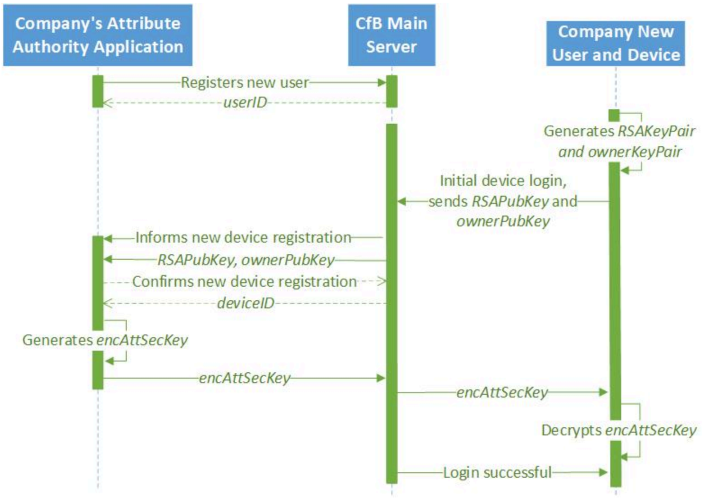

Company's Attribute Authority Application
CfB Main Server
Company New User and Device
Registers new user
userID
Generates RSAKeyPair and ownerKeyPair
Initial device login, sends RSAPubKey and ownerPubKey
Informs new device registration
RSAPubKey, ownerPubKey
Confirms new device registration
deviceID
Generates encAttSecKey
encAttSecKey
encAttSecKey
Decrypts encAttSecKey
Login successful

Figure 3.9: Sequence diagram of user and device registration processes in the MA-CfB system [Suk+19a]

the file owner generates authorization key ($authZKey$) for each authorized user using $ownerPrivKey$. $authZKey$ is then encrypted with the collection of the user's device *RSAPubKey* to get encrypted authorization keys (*encAuthZKeys*). If the file owner does not impose it, then $ownerPrivKey$ is not used during file key encryption, and $authZKey$ is not generated. Lastly, $encFileKey$, policy, file access specification, and a collection of optional encrypted $authZKey$ are then sent to CfB. Figure 3.10 shows the diagram of the file upload process.

3.7.5 File Download

When a CfB user wants to download a file, CfB first sends $encFileKey$ and optional $encAuthZKey$ to the user, which will be decrypted using the device's *RSAPrivKey* to retrieve $authZKey$, if the file has the two-factor authorization restriction. $encFileKey$ is then decrypted with the device's $attSecKey$ and $authZKey$, if available, to retrieve $fileKey$. If the user does not have the correct attributes that fulfill the $encFileKey$'s policy or the optional $authZKey$, the decryption process is failed. Finally, $fileKey$ is then used to decrypt the encrypted file, which is assembled from multiple file chunks retrieved from multiple CSPs. Figure 3.11 shows the diagram of file download operation.

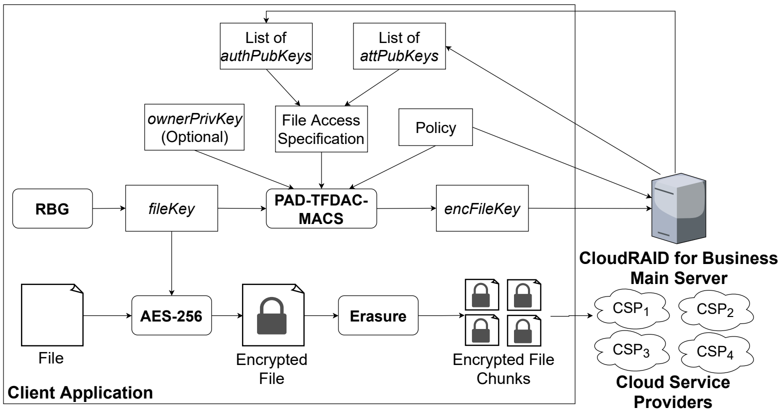


Figure 3.10: File upload process in the MA-CfB system [Suk+19a]

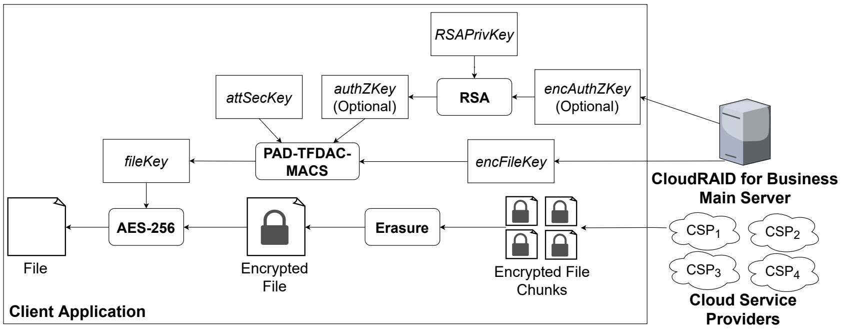


Figure 3.11: File download process in the MA-CfB system [Suk+19a]

3.7.6 Attribute and User Revocation

There are several steps involved when an attribute (att) of a CfB user is revoked by the company to ensure the revoked attribute ($revokedAtt$) could no longer be used. First, the company's AA application calculates the new attribute key pair ($newAttKeyPair$) for $revokedAtt$. It informs the CfB that att is revoked from the user while sending the new attribute public key ($newAttPubKey$). Meanwhile, it gathers the list of users who share the same $revokedAtt$ as the affected user has. It computes and sends attribute update key ($attUpdateKey$) to the affected non-revoked users and its devices to update their device's $attSecKey$ who share the same $revokedAtt$.

Furthermore, the AA application requests the C_2 component of $encFileKeys$ from the CfB main server that is encrypted with $revokedAtt$ contained in its policy. If $encFileKeys$ has two-factor authorization restriction, it also collects the file owner's $ownerPubKey$ of the device used to encrypt the affected $fileKey$ and. It then calculates a ciphertext update key $cipherUpdateKey$ based on the C_2 component of each affected $encFileKey$ and optional $ownerPubKey$. Finally, $cipherUpdateKey$ are sent to CfB main server to update the affected $encFileKeys$.

If a CfB user is revoked from the system, it runs the same procedure multiple times depending on the number of the attributes owned by the revoked user. Other users who share the same attributes as the revoked user will receive $attUpdateKeys$ to update their device's $attSecKey$ and the user's $encryptedFileKeys$ will be updated using $cipherUpdateKey$ by the CfB. Figure 3.12 shows an example of the attribute revocation process in CfB.

3.7.7 Authorization Key Revocation

If a CfB user as the file owner wants to revoke another CfB user's $authZKey$ to access the file, the user first informs CfB that the revoked user does not have access to the file anymore. The file owner then generates a new $ownerKeyPair$ that is used to calculate $cipherUpdateKey$ for the affected file and authorization update key ($authZUpdateKey$) for each non-revoked user who still has authorized access to the file.

The file owner later sends $cipherUpdateKey$ and $authZUpdateKey$ to the CfB. CfB will use $cipherUpdateKey$ to update the $encFileKey$ of the affected file and sends $authZUpdateKey$ to non-revoked users. Finally, the non-revoked

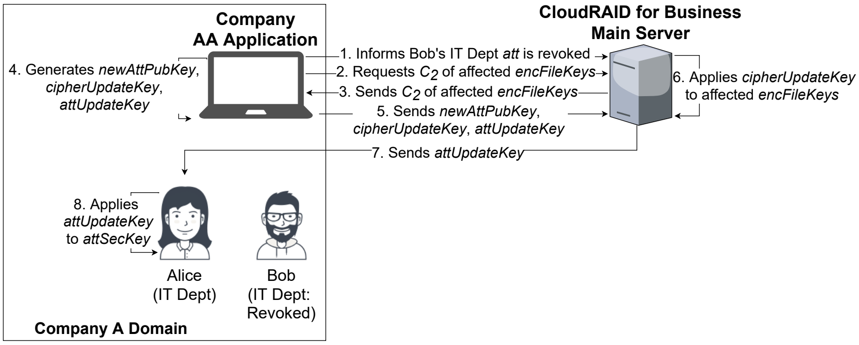

Company
AA Application
CloudRAID for Business
Main Server
1. Informs Bob's IT Dept *att* is revoked
2. Requests C_2 of affected *encFileKeys*
3. Sends C_2 of affected *encFileKeys*
4. Generates *newAttPubKey*, *cipherUpdateKey*, *attUpdateKey*
5. Sends *newAttPubKey*, *cipherUpdateKey*, *attUpdateKey*
6. Applies *cipherUpdateKey* to affected *encFileKeys*
7. Sends *attUpdateKey*
8. Applies *attUpdateKey* to *attSecKey*
Alice (IT Dept)
Bob (IT Dept: Revoked)
Company A Domain

Figure 3.12: Example of Bob's IT Department attribute revocation process by the company administrator in the MA-CfB system [Suk+19a]

user will use it to update their *authZKey* to continue having access to the file. Figure 3.13 shows an example of authorization key revocation process.

3.8 Evaluation and Discussion

The performance of key management systems of CloudRAID, Single-Authority CloudRAID for Business, and Multi-Authority CloudRAID for Business are evaluated. Later, the security of the three systems are discussed to determine which system is suitable for EFSS system.

3.8.1 Performance Evaluation

The elapsed time and the size of keypair and ciphertext generated during the key management system's main processes are evaluated for all CloudRAID, SA-CfB, and MA-CfB systems: setup, key pair generation, file key encryption, and encrypted file key decryption. The size of the file key used during the evaluation is 32 bytes. File operations (encryption, decryption, erasure, upload, download) are not relevant in this evaluation. The performance evaluation was run on Intel i5-8400 @ 2.8GHz with 16GB RAM.

The scenario of file-sharing is between the employees of two companies for our performance evaluation: Company A consists of the IT Department and Management Department of Company B, which consists of the HR Department,

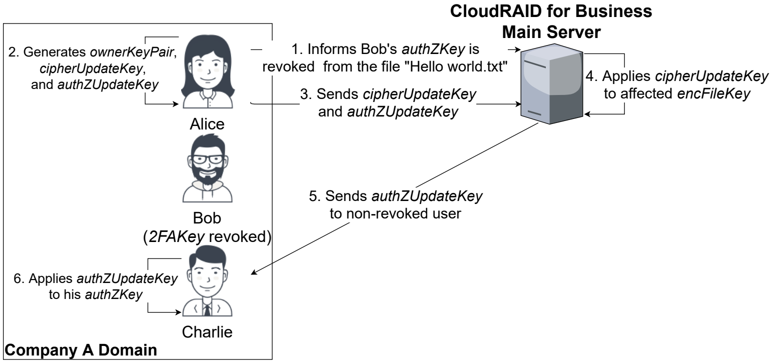

CloudRAID for Business
Main Server
1. Informs Bob's *authZKey* is revoked from the file "Hello world.txt"
2. Generates *ownerKeyPair*, *cipherUpdateKey*, and *authZUpdateKey*
3. Sends *cipherUpdateKey* and *authZUpdateKey*
4. Applies *cipherUpdateKey* to affected *encFileKey*
5. Sends *authZUpdateKey* to non-revoked user
6. Applies *authZUpdateKey* to his *authZKey*
Alice
Bob
(*2FAKey* revoked)
Charlie
Company A Domain

Figure 3.13: Example of Bob's authorization key revocation process by Alice as the file owner in the MA-CfB system [Suk+19a]

Finance Department, and the Accounting Department. Each department is assumed to have ten employees, with each employee only has a single device.

CfB will use the company's departments as the attributes for the file-sharing scenario. SA-CfB system will use one AA entity shared by two companies with 100 keys set during the setup process. For the MA-CfB system, one AA entity is used for each company, where it only uses the encryption method without two-factor authorization functionality. Both CfB systems did not use native pairing-based cryptography C library[11] to improve their performance.

As seen in Table 3.1 and 3.2, CloudRAID does not need a setup phase as it utilizes RSA in ECB mode with PKCS1 Padding for its KMS. SA-CfB's setup process is slightly faster than Multi-Authority CfB's since MA-CfB needs to run the setup process and initialize two attribute authority entities for two companies. The size of SA-CfB's public key is 67 times bigger than each MA-CfB's authority public key and attribute public key. SA-CfB's secret master keypair is up to 482 times bigger than each MA-CfB's authority private key and attribute private key.

CloudRAID takes the most time for the user keypair generation process, with overall the biggest keypair's size compared with the two CfB systems. It needs to generate two keypairs for each user to ensure secure file-sharing between the two companies: RSA account keypair and RSA device keypair. The size of the user's attribute secret key generated by SA-CfB and MA-CfB depends on the

[11] https://crypto.stanford.edu/pbc/

number of attributes owned by the user. MA-CfB requires less time to generate the keypair up to 3 times faster and 84 times smaller keypair's size than SA-CfB.

The encryption process in SA-CfB and MA-CfB requires more time compared to CloudRAID since CloudRAID is using RSA that is "much lighter" compared with ABE schemes utilizing pairing-based cryptography. SA-CfB and MA-CfB generate a single encrypted file key for multiple users sharing different attributes in the policy with almost constant size, except for MA-CfB's encryption process with the OR-gate policy, which is affected by the number of attributes in the OR-gate policy. Meanwhile, CloudRAID generates the biggest encrypted file key size as it generates two encrypted file keys per file as the file key is encrypted with each user's account public key and device public key.

The duration of the encrypted file key decryption process for SA-CfB and MA-CfB is faster than CloudRAID's decryption process since each CloudRAID user needs to decrypt one of two encrypted file keys generated explicitly for that particular user. MA-CfB's decryption duration for encrypted file key with AND-gate policy is almost constant while SA-CfB's decryption duration increases with the number of attributes in the policy. The duration for MA-CfB's encrypted file key with OR-gate policy decryption is similar to its counterpart because MA-CfB user only needs to decrypt one of the encrypted file key parts that are satisfied by their attribute secret key. The duration of MA-CfB's decryption process is overall the fastest compared to CloudRAID and SA-CfB systems due to the encrypted file key's small size.

The file key encryption and encrypted file key decryption processes with AND-gate policy in MA-CfB prove TFDAC-MACS' claim of constant-sized ciphertext with a small computational cost that affects the encryption's and decryption's duration to be relatively constant [Li+17]. Meanwhile, SA-CfB's duration of all processes and the size of the generated keypairs and encrypted file key are affected by the number of keys specified during the setup process.

Based on the comparisons above, MA-CfB and SA-CfB systems provide more scalable key management systems than CloudRAID as both can generate one encrypted file key per user's file that can be decrypted by multiple users who are sharing the same attributes. The MA-CfB system provides the best overall performance result compared with ClouRAID and SA-CfB systems as it requires the least amount of time needed for all key management system processes and generates the smallest size encrypted file key and attribute keypairs.

Processes		**CloudRAID**	**SA-CfB**	**MA-CfB**
Setup and Attribute Authority Generation		-	235	147
Key Pair Generation	1 attribute	3838	183	50
	2 attributes	7353	216	100
	3 attributes	10650	241	146
	4 attributes	13102	268	193
	5 attributes	18606	294	245
Encryption	1 attribute	2	1937	23
	2 attributes	5	AND-Gate: 1983 OR-Gate: 1987	AND-Gate: 22 OR-Gate: 43
	3 attributes	7	AND-Gate: 2031 OR-Gate: 2039	AND-Gate: 23 OR-Gate: 65
	4 attributes	9	AND-Gate: 2035 OR-Gate: 2066	AND-Gate: 23 OR-Gate: 79
	5 attributes	11	AND-Gate: 2108 OR-Gate: 2142	AND-Gate: 23 OR-Gate: 102
Decryption	1 attribute	72	62	33
	2 attributes	135	AND-Gate: 75 OR-Gate: 61	AND-Gate: 32 OR-Gate: 33
	3 attributes	202	AND-Gate: 91 OR-Gate: 60	AND-Gate: 33 OR-Gate: 34
	4 attributes	268	AND-Gate: 104 OR-Gate: 62	AND-Gate: 32 OR-Gate: 30
	5 attributes	335	AND-Gate: 117 OR-Gate: 60	AND-Gate: 33 OR-Gate: 32

Table 3.1: Elapsed time comparison in milliseconds of selected key management processes in the CloudRAID, SA-CfB, and MA-CfB systems [Suk+19a]

Cryptographic Keys		CloudRAID	SA-CfB	MA-CfB
Public Key		588	8651	Authority: 128 Attribute: 128
Secret Master Key / Private Key		2432	9653	Authority: 20 Attribute: 20
Secret Attribute Key	1 attribute	-	10786	128
	2 attributes		11102	256
	3 attributes		11419	384
	4 attributes		11736	512
	5 attributes		12053	640
Encrypted File Key	1 attribute	5120	24978	419
	2 attributes	10240	AND-Gate: 25438 OR-Gate: 25438	AND-Gate: 455 OR-Gate: 851
	3 attributes	15360	AND-Gate: 25890 OR-Gate: 25890	AND-Gate: 490 OR-Gate: 1258
	4 attributes	20480	AND-Gate: 26342 OR-Gate: 26342	AND-Gate: 526 OR-Gate: 1678
	5 attributes	25600	AND-Gate: 26794 OR-Gate: 26794	AND-Gate: 562 OR-Gate: 2098

Table 3.2: Length comparison of the generated keys and ciphertexts in bytes by the CloudRAID, SA-CfB, and MA-CfB systems [Suk+19a]

3.8.2 Security Discussion

The proposed SA-CfB and MA-CfB systems fulfill the requirements for providing a secure enterprise file synchronization and share system due to attribute-based encryption schemes. Using the ABE schemes, both CfB systems could generate and store one encrypted file key per file for multiple users and devices in the company due to the scheme's "one-to-many" property. It is different from CloudRAID's cryptographic methods that generate multiple encrypted file keys per file for each user and device. The number of the encrypted file key depends on the number of CloudRAID users and their devices involved in the group file sharing. This allows CfB to store less encrypted file keys that could potentially save more storage spaces and have better performance than CloudRAID.

Both CfB systems provide attribute-based access control for the encrypted file keys where only authorized CfB users with correct attributes that fulfill the file-sharing specification's access policy could decrypt it. The security of the encrypted file keys also does not depend any longer on the strength of the CfB user's password, as in the case of CloudRAID.

The CfB system provides file-level and system-level file access revocation using the ABE scheme's revocation mechanisms, whereas CloudRAID only provides system-level file access revocation. Revoked CfB users will no longer decrypt the encrypted file key even though their attributes fulfill its policy. The indirect revocation mechanism provided by the PAD-TFDAC-MACS library allows for file-level file access revocation for inter-company file-sharing compared with the direct revocation mechanism provided by the jTR-ABE library.

It also gives the company the authority to manage its users and files in the system without any interference from CfB thus providing a zero-knowledge policy in CfB. Companies could generate and store the necessary attribute and authority keypairs locally on its attribute authority (AA) application where only the public keys and information of the company's attributes and authority are sent to CfB. CfB will be unable to generate the company's private or secret key of the attributes and the authority due to the random elements, therefore unable to decrypt CfB user's encrypted file keys and encrypted files.

However, the single authority ABE scheme provided by the jTR-ABE library used by the SA-CfB system could not offer optimal scalable and secure inter-company file sharing functionality. This is due to the SA-CfB system runs on a limited number of keys that needs to be specified during the setup phase due to the CP-ABE scheme [LW16] that utilizes a limited large attribute universe to trace a malicious user who uses the private key to create a decryption blackbox. It could also potentially generate multiple encrypted file keys per file for sharing with multiple companies since each company through its AA will generate public keys for the attributes used in the policy during file key encryption.

Therefore, the MA-CfB system is the preferred system architecture for CloudRAID for Business as it could provide secure and scalable file-sharing for within and between companies. As a CfB customer, each company has the authority to manage and share the files with other companies where it will generate one encrypted file key per file for multiple users and their devices across various companies. The attribute-based access control for the encrypted file keys will still be enforced without any additional mechanisms, unlike the SA-CfB system.

3.9 Conclusion and Future Works

In this chapter, two CloudRAID for Business system architectures based on attribute-based encryption schemes are proposed to resolve the scalability and security challenges of CloudRAID to provide a secure enterprise file synchronization and share system: Single-Authority CloudRAID for Business (SA-CfB) and Multi-Authority CloudRAID for Business (MA-CfB). Both system architectures generate one encrypted file key per user's file for multiple users and devices in the company and provide attribute-based access control to the encrypted file key where only authorized CfB users with correct attributes fulfill the file-sharing specification's access policy could decrypt it. Based on the performance and security evaluation and comparison of two proposed CfB systems and the CloudRAID system, Multi-Authority CfB is the preferred CfB system architecture. It provides better security, scalability, and performance for a key management system that is more suitable for intra-company and inter-company file-sharing functionalities.

The file-sharing in the CfB system still requires more work to be more secure, scalable, and efficient for all authorized CfB customers and users. Improving the expressiveness of the TFDAC-MACS scheme's access policy using a linear secret sharing scheme or non-monotonic policy for the PAD-TFDAC-MACS library could be helpful to make file-sharing in the system to be more flexible. Another interesting research topic is scalable system-level organizational-based file access control for file-sharing between multiple companies to avoid cross-company data leakage. System-level attribute-based access control using eXtensible Access

Control Markup Language (XACML) could also be implemented to complement file-level attribute-based access control.

4 Enforcing Location-Based File Access Control

4.1 Introduction

As enterprise file synchronization and share systems allow for files to be always available, the company's employees can remotely access files from anywhere around the world as long as they have Internet access. Especially during the COVID-19 pandemic, the employees are working remotely and accessing the company's confidential files from their remote workplace across the globe that helps to increase the collaboration and productivity of the company.

However, this raises the challenges for companies to enforce physical access control to their confidential files. The employees could access the company's confidential files at insecure locations, where employee's laptops or mobile phones could be stolen, or an attacker could view the sensitive information by looking over the employee's shoulder [SS16]. The companies might opt to only allow their confidential files to be accessed at certain trusted locations to ensure the files are securely accessed by authorized employees, such as employee's homes or the company's office building.

Location-based access control (LBAC) could provide necessary physical access control for the company's confidential files managed by CloudRAID for Business. It utilizes the location information of the CfB users provided by Global Positioning System (GPS), mobile network, sensors, or other location determination technologies and determines whether the users are authorized to access the files based on the [Che+17; Cho+16; Dec08].

However, there are several challenges to implement location-based access control functionality for the CfB system. The location information provided by CfB users might not be 100% accurate that would tell the actual true location of the users. For example, the location information provided from the GPS consists of a latitude-longitude coordinate and accuracy radius where the user's true location might fall anywhere within the location's circle. The location information could also be manipulated to trick the CfB system into gaining unauthorized access to the location-restricted files using fake GPS applications or Virtual Private

Network (VPN) or proxy services. Therefore, the CfB system needs to provide a location-based access control scheme capable of calculating user's location, dealing with the uncertainties of the location information submitted by the users, and determining whether the users are granted or denied access to the company's confidential files.

In this Chapter, a new location-based access control scheme is proposed for CloudRAID for Business called **Internet-based location access control** (ILAC) to ensure authorized CfB users located at the pre-determined location could only access the location-restricted files. It utilizes the Internet-based location where the location information is collected from the Internet-connected device used by the CfB user. The Internet-based location could be used by the CfB system to provide an alternative location determination method and a verification method whether the CfB user is truly at the pre-determined location and does not manipulate the submitted location.

4.2 Related Works

4.2.1 Research Works

Several works focus on developing a location-based access control model necessary that would allow the underlying systems to enforce the resources to be only accessed at certain locations.

Ardagna et al. [Ard+06] proposed a location-based access control model that specifies how location-based access control policy could be expressed, evaluated, and enforced by integrating location-based conditions with the access control model. Zickau et al. [Zic+14] introduced location-based policies for healthcare cloud computing environment using location-based services that support various location positioning technologies, eXtensible Access Control Markup Language (XACML), and Geospatial XACML (GeoXACML) while complying with the data protection regulations. Hsu and Ray [HR16] demonstrated the usage of LBAC to protect personal information in the social networks by extending NIST Policy Machine to correlate user's location inferred from the IP address with user behavior characterized by geographic metadata using the combination of role-based access control and GeoXACML. Baracaldo et al. [BPJ15] presented a role-based access control model called Geo-social-RBAC to include the location history and the social contexts of the requester as part of the access control policy to determine if the requester is authorized to access the resources. Ulltveit-Moe and Oleshchuk [UO16] implemented location-aware role-based access control policies on top of GeoXACML for mobile security where the type of access to the resources is determined by the location of the requester. Singh et al. [SC21] introduced a location-based access control model for an e-Healthcare system called LoBAC to ensure the requesters could only access confidential health data in certain locations.

Several works propose different methods to calculate and verify the requester's location to be at the pre-determined location. Lu et al. [Lu+19] proposed a crowdsourcing method for location-aware Wi-Fi access control called LaSa to restrict the Wi-Fi access only in a certain area by analyzing received signal strength, channel state information, and coarse angle of arrival data from the users with one-class Support Vector Machine algorithm. Nosouhi [Nos+18] proposed the SPARSE scheme to provide secure and private distributed location proof systems for mobile users using time-limited non-distance bounding protocol. [Yam+19] utilized images generated from the camera in wireless local area network (WLAN) to create a geo-fenced area to enforce accurate access control based on the user's location around the WLAN coverage region. Choi et al. [Cho+16] implemented an LBAC system in a local area by broadcasting one-time passwords only to clients in the vicinity of a Bluetooth Low Energy beacon. Yang et al. [Yan+18] introduced a secure location verification protocol suitable for fog computing that protects the privacy of the requester's location based on secret sharing broadcast with a bounded retrieval model.

Other works are utilizing cryptography methods to provide a location-based functionality system to ensure that encrypted data could only be accessed once the requester is at the allowed location. Scott and Denning [SD03] proposed Geo-Encryption algorithm by utilizing symmetric and asymmetric cryptography algorithms with GeoLock functionality to ensure the receivers fulfill the position, velocity, and time restrictions to be fulfilled. Xue et al. [Xue+16] combined ciphertext-based policy attribute-based encryption scheme and LBAC system for the cloud storage system to ensure the ciphertext can only be accessed when the user satisfies the policy in the ciphertext and receives the location token from the corresponding location server. Baseri et al. [BHC18] introduced a privacy-preserving location-based access control scheme to provide dynamic anonymous and unforgeable location verification for mobile cloud by integrating multi-authority attribute-based encryption and proxy re-encryption.

4.2.2 Competitors

Dropbox Business

Dropbox Business provides location-based access control functionality in the form of geo-blocking, where access to the resources or services is denied based on the user's location. Dropbox users in Crimea, North Korea, Syria, and other regions or countries listed in the government's trade sanctions and embargo requirements are then unable to access the service [Dro21a]. Although it is not explicitly mentioned in [Dro21a], it is assumed that Dropbox filters the IP address of the users to determine their locations and determine if they are allowed to access the service.

Dropbox restricts access to its production environment to a limited number of IP addresses associated with the corporate network or approved Dropbox personnel. It also enforces the same access restriction functionality on its AWS environment used for processing and storage processes of Dropbox user's files in the cloud, where AWS provides IP filters functionality on several services that would block or allow specific IP address ranges of the requester [Dro20].

Tresorit

Tresorit provides location-based access control functionality for the subscribing companies to limit access to confidential data only in certain locations. It utilizes external IP geolocation services, e.g., DB-IP[12], where the Tresorit user's location is estimated based on the IP address [Tre19].

The company's administrator could set up multiple allowed locations where Tresorit will filter the user's IP address and IP-based location to the known IP addresses and locations. When the Tresorit users attempt to log in from unknown locations, the users will then be unable to log in, and the company's administrator will be notified through the Admin Center [Tre19; Tre21e].

Boxcryptor

Boxcryptor allows the organization to set up access and location policies to ensure Boxcryptor users could only access the service at certain locations [Box21c]. The access policy will restrict the access based on the specific list of countries

[12] https://db-ip.com/

and IP addresses with different conditions, such as restrict users based on the country/IP address the user initially signed in. The organization could specify the maximum number of locations configured at the same time and whether the user might or must use the specified locations on the location policy. If the Boxcryptor user's country location and IP address violate the access policy, the user will be unable to use the Boxcryptor until the user is at the correct location or utilize the correct IP address.

4.2.3 Thesis Contribution

The work proposed in this chapter is different than the research works and the competitors to provide location-based access control functionality. An Internet-based location access control system is proposed for CloudRAID for Business utilizing the information inferred from the Internet-connected devices of the CfB's user. The Internet-based location can be used to determine the location of the users with minimum or no self-geolocation capability, verify the received user's location whether it has been manipulated, and determine if the user is granted or denied access to the company's confidential files. This could help to resolve the challenge faced by most of the EFSS systems that solemnly rely on the IP address to determine a user's location since the user could manipulate the location by using VPN or proxy services.

4.3 Location-based Access Control for Enterprise File Synchronization and Share Systems

The definition of the location used in this chapter is first formalized. A location is a geographical position on earth represented by a latitude-longitude coordinate. It could also be a set of coordinates forming a polygon representing a certain region on earth, such as a country or city. The elevation element of the location is ignored as it only considers the two-dimensional plane of the location.

Location-based access control (LBAC) is an access control model type that determines the access privileges based on the physical location of the entity [Ard+08; Dec08]. Only an entity located at the pre-determined location set by the resource's access control restriction is then authorized to access the resources; else, the entity is denied access to the resources.

LBAC has been used by several services to determine if the users are granted or denied access to the resources based on their locations, particularly location-based services. Location-based services are services that deliver the information depending on the location of the devices and the users [Rap+07]. It becomes more popular in recent years due to the high rate of Internet adoption across the world and rapid advances in mobile technology that would allow everyone to access the resources from anywhere in the world [Hua+18].

Location-based services need to enforce location-based access control as the resources might only be available or unavailable to the users in a specific location due to several reasons. Several services might not be available in different countries due to government regulations or embargoes, as exemplified previously. Audio or video content might not have global licenses that affect the data to be available only at the countries or regions where a license has been obtained [Red20]. For example, in 2016, 80% of films from the United States of America were only available as Video on Demand in 11 European Union countries or fewer [Gre16]. Companies could also allow their confidential data to be accessed only at certain locations.

The common location input used by the location-based services to determine whether the user is at the allowed location is the location information provided by the global navigation satellite system (GNSS). Global Position System (GPS) is one of the most widely used GNSS to provide latitude-longitude coordinates since it is available worldwide for civilian purposes in the early 2000s [Hua+18]. The GPS coordinate could be calculated by a GPS-enabled device by constantly listening to the time information broadcasted from the navigation satellites around the earth and determine the time difference between the time information received with the broadcast time [Fed20]. Assisted-GNSS and Cloud-GNSS could transmit additional data via mobile networks or the Internet that would help the device to improve the location calculation's performance, and accuracy [Che+17].

Location-based services could also utilize the location information inferred from the signals from non-GNSS or sensors to provide efficient and accurate locations for indoors, such as cellular signals, WiFi access points, and Bluetooth Low Energy. The location could be determined using different parameters of the signals received by the device, e.g., the angle and time of signal arrival [Ard+06]. It could also be calculated using the location fingerprinting approach where the received signal strength is compared against the database or a machine learning model of signal strength [Che+17; Li+19].

The IP address could also be used by location-based services to determine if the user is located at the allowed location with the assumption that the IP address could uniquely identify the device [Pad+16]. Location information of public IP address could be inferred based on the DNS LOC record, or WHOIS registration gathered from the regional Internet registries, such as American Registry for Internet Numbers (ARIN)[13] or Réseaux IP Européens Network Coordination Centre (RIPE NCC) [Gha+17; Liv+20]. IP geolocation services could also provide the location information of the IP address by collecting the information available of the IP address, measuring the Internet network, and collecting other external resources, such as WiFi access points or GPS coordinates [KVR17].

However, several challenges are faced by location-based services while processing user's submitted locations.

The GNSS location information is "accurate" under a certain radius, e.g., 2 or 100 meters, depending on the received signal strength, which means the user's true location could lie anywhere in the location circle [ZB11]. The accuracy and performance of the location depend on the signal strength that could be affected by many factors, such as earth's atmosphere and space weather or radio frequency interference [Che+17]. This also makes the GNSS positioning to be less accurate for indoor positioning.

Also, since GNSS location information is calculated on the device using the received signals, the satellites could not actively verify the device's location; therefore, verifying GNSS location information could be a challenge for location-based services [Che+17]. The users could spoof their locations using fake GPS applications installed on their device to make location-based services believe that the users at the allowed location.

Signals from non-GNSS or sensors are also prone to several vulnerabilities. If the device could not respond to the signal from the surrounding non-GNSS or sensors due to missing access nodes or the environment, the location-based services then would not be able to calculate the device's location. Training received signal strength database used for the location fingerprinting approach might have error or wrong signal strength or location entries thus it could return the wrong location information [Che+17].

The IP address also has several weaknesses to be used as a location input for location-based services. Location-based services might assume that the user will have the same IP address for a long period of time; however, IP address assigned to a device could dynamically change for various reasons, such as network

[13] https://www.arin.net/

outage, lost connection, or session time limit set by the Internet Service Provider (ISP) [Pad+16]. Meanwhile, the location information inferred from the IP address could only provide city and country information of where or to whom the IP address is assigned to. The location information provided by IP geolocation service might not be very accurate as it can only provide good country-level accuracy, but bad city-level accuracy with the possibility of inconsistent and outdated location information [Gha+17; KVR17].

The IP address could also be masked using virtual private network (VPN) or proxy services to deceive the location-based services. The communication between the user's device and the location-based services is encrypted and tunneled through the VPN/proxy server, where it is located at the allowed location. This will affect location-based services to assume the location of the user based on the VPN server's IP address instead of the actual user's IP address.

EFSS systems in recent years have implemented LBAC functionality to ensure the services and company's confidential files could only be accessed by EFSS users located at trusted or certain locations as shown in the previous section. It should calculate the user's location even if the user's devices do not have self-geolocation capability. It should also be capable of verifying the user's submitted location whether it has been manipulated to trick EFSS system giving access to the company's confidential files. Finally, it must grant or deny the user's file access request if the user's submitted location and calculated location are at the pre-determined and trusted locations set by the companies.

4.4 Internet-based Location

Internet-based location is a location inferred from the information provided or retrieved from Internet-connected devices. It could be used by location-based services as an alternative location input instead of the conventional location inputs as explained previously to determine if the user is at the allowed location.

There are three location information inputs that can be used to infer the user's Internet-based location:

- **IP Address**: IP address could provide the country- and/or city-level accuracy location information of the user's Internet-connected device. Its location information could be retrieved from its DNS LOC record or WHOIS registration or using IP geolocation services [KVR17; Liv+20].

- **Latency/delay measurement result**: The **latency** is the amount of time taken to send a packet round-trip from a sender to a receiver. The **delay** is the amount of time taken to send a packet one-way from a sender to a receiver or half of the latency. The location of Internet-connected devices can be referred from the delay or the latency between two end-points as it follows an assumption that there is a strong correlation between the Internet delay and geographic distance of two end-points [AMV17].

  A landmark server, which is a public server with known locations, measures the latency or delay with the Internet-connected device and other available landmark servers to generate the measurement result. The delay/latency measurement result between the Internet-connected device with the landmark servers could be mapped to a geographical location using a delay-based geolocation algorithm, and the delay measurement result between the landmark servers [AKK10b; Gue+06].

- **WiFi access point's signal strength**: If the device is connected to the Internet using WiFi access points (APs), the signal strength of surrounding WiFi APs in dBm could be used to locate the Internet-connected devices in the surrounding area. Several services, e.g., OpenWifi[14] or WiGLE[15], provide the mapping between WiFi APs and its locations using the basic service set identifiers or media access control (MAC) address and the received signal strength by the Internet-connected device.

The IP address and the delay measurement result are regarded as the **primary location inputs** for Internet-based location, while WiFi AP's signal strength and other conventional location inputs are regarded as the **secondary location inputs**, such as GPS coordinate or mobile network. This is due to IP address, and delay measurement result will always be retrievable from the Internet-connected devices where the delay-based services will be able to calculate its Internet-based location, while other location inputs depend on the device's functionality or the environment it is in.

The Internet-based location could be used as the location determination method for devices with none or minimum self-geolocation capability where it might require additional hardware or the environment to calculate its location.

[14] http://www.openwlanmap.org
[15] https://wigle.net/

The Internet-connected device could actively communicate with other entities to calculate its Internet-based location inputs as **proactive location approach**, e.g., delay measurement result and WiFi AP's signal strength. The location input of an Internet-connected device could also be queried or calculated from the device itself as **reactive location approach**, such as GPS and IP address.

It could also be used to verify the device's submitted location inputs to be authentic and intersect with the allowed location. Although Internet-based location inputs could return different accuracy levels, each location input could be used to verify whether other location inputs have not been manipulated. If there is a location input that returns different location information from other location inputs, it is most probable that the device manipulates its location.

4.5 Internet-based Location Accesss Control for CloudRAID for Business

This Section describes how CloudRAID for Business system implements location-based access control functionality using Internet-based location as the location input to ensure that the company's confidential files can only be accessed in the allowed location. As CfB users need to be connected to the Internet while requesting access to the file, the CfB system will be able to calculate their Internet-based location and determine if the CfB users are at the allowed location. CfB system could verify CfB user's submitted location against the CfB user's Internet-based location while determining if the CfB user is at the allowed location to allow or deny CfB user's file access request.

4.5.1 Architecture

CloudRAID for Business system with Internet-based location access control functionality consists of four main entities as can be seen in Figure 4.1:

- **CfB Client Application**: CfB users could share files with other CfB users using CfB client application while setting certain restrictions on the file, including location restrictions to ensure CfB users could only access it at certain locations.

  CfB users could request access to the files shared by other CfB users to the CfB main server. If the file has location restriction enabled, the client

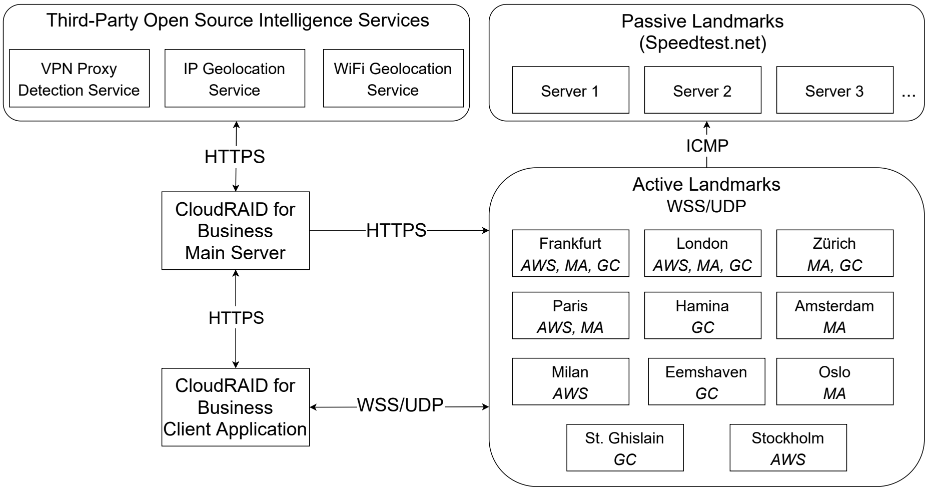


Figure 4.1: CloudRAID for Business' architecture overview with Internet-based location access control [Suk+21b]

application will collect available location information inputs depending on the device's capabilities and the surrounding environment and send it to CfB main server.

- **CfB Main Server**: The CfB main server is responsible for ensuring only authorized CfB users could access the files at the allowed location. It intermediates the connections between the CfB users using CfB client application, landmark servers, and third-party open source intelligence services. Based on the location inputs provided by the CfB users and the location information from the third-party open sources intelligence services, it then grants or denies the CfB user's file access requests.

- **Landmark Server**: The landmark server is a publicly accessible server with a known geographic location that is used to measure the delay with the target in millisecond (ms). There are two types of landmark servers available used by the CfB system to obtain the delay measurement results needed for calculating CfB user's location:
  - *Active Landmark Server*: The active landmark server is capable of both sending and receiving ping requests. It is controlled by the CfB

main server to measure the delay with other landmarks and the CfB users that will be further explained in Chapter 4.5.2.

  - *Passive Landmark Server*: The passive landmark server is only capable of responding to the ping requests and is used to get more diverse delay measurement results for more accurate location calculation using delay-based geolocation algorithms. CfB system uses the servers from the Speedtest network[16], which is a platform to test the speed and the performance of the Internet connection, where the servers are only publicly accessible by ping request. **200 servers** are randomly selected from the Speedtest network located in several northern and western European countries as the passive landmarks, e.g., Germany, United Kingdom, and Denmark.

- **Third-Party Open Source Intelligence Services (OSINT)**: The CfB main server utilizes third-party OSINT services to gain additional location information about the CfB users necessary for deciding if the CfB users are authorized to access location-restricted files. It will be further explained in Chapter 4.5.4.

The CfB client application connects to the CfB main server using HTTPS to request access to the CfB user's files and receive the response whether the CfB user is authorized to access the file or not. The CfB main server manages its active landmarks across Europe using HTTPS to trigger delay measurement between the active landmark and other active landmarks and passive landmarks. It also sends the CfB user's location information inputs to the third-party OSINT services to ensure the CfB user's location inputs could not be eavesdropped on by unauthorized entities.

The CfB system utilizes the **delay** instead of the latency to calculate and verify the CfB user's location. The delay measurement process between the active landmarks and the CfB user's client application utilizes WebSocket Secure (WSS) or User Datagram Protocol (UDP) where it is intermediated by the CfB main server. The active landmarks could not launch the delay measurement directly with the CfB users due to the firewall or NAT (network address translation) of the ISP that might block the connection from the unknown and unestablished IP address. This issue could be potentially solved using the NAT traversal

[16] https://www.speedtest.net/

techniques, such as hole punching, that would allow the active landmarks to launch the delay measurement to the CfB client application using the established connection between the client application and CfB main server. However, the success of the NAT traversal techniques depends on the NAT types used by the ISP of the CfB user, where the hole punching technique's success rate could be 64% (TCP) or 82% (UDP) [FSK05]. Therefore, the CfB client application first establishes the WSS/UDP connection with the active landmarks. Once the connection is established, the active landmarks could start measuring the delay with the client application.

The delay between the active landmarks is measured using WSS or UDP protocol to match the delay measurement between the client application and the active landmarks. Meanwhile, the delay between active and passive landmarks is measured using Internet Control Message Protocol (ICMP) since the Speedtest's server is only reachable via ICMP ping request. The difference of delay measurement result using ICMP and WSS has **a maximum of 1 ms**, which is still an acceptable error for calculating CfB user's Internet-based location.

4.5.2 Landmark Servers

The delay measurement used in several delay-based geolocation algorithm papers is typically done using Internet measurement testbed platforms, such as PlanetLab[17] or RIPE Atlas[18]. The platforms consist of a global network of multiple probes or nodes that act as the landmark servers to measure the delay to the targets. However, these platforms are not scalable for CfB systems as the probes are only capable of limited preset activities and might not be able to measure the delay to the requesters behind the network address translation system of the Internet service provider. Therefore, Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure (Azure) are chosen to support the Internet-based location access control for CfB systems due to several reasons. It allows for scalable development as new instances could be created across various regions. It is also generally cheaper and easier to launch delay measurements than using Internet measurement platforms.

Serverless computing services were the first option that could be used to

[17] https://planetlab.cs.princeton.edu/
[18] https://atlas.ripe.net/

Virtual Machine	**AWS EC2**	**GCP CE**	**Azure VM**
Locations	Frankfurt (DE) Stockholm (SE) Milan (IT) London (GB) Paris (FR)	Hamina (FI) St. Ghislain (BE) London (GB) Frankfurt (DE) Eemshaven (NL) Zürich (CH)	Frankfurt (DE) Amsterdam (NL) Oslo (NO) Paris (FR) Zürich (CH) London (GB)
Specifications	**t3.nano** 2 vCPUs 0.5 GB vRAM 8 GB disk	**e2-micro** 2 vCPUs 1 GB vRAM 10 GB disk	**B1s** 1 vCPUs 1 GB vRAM 4 GB disk

Table 4.1: Location of active landmarks deployed in AWS Elastic Cloud Computing, GCP Compute Engine, and Azure Virtual Machine [Suk+21b]

deploy the landmark servers, such as AWS Lambda[19] or GCP Functions[20]. It does not require the actual server to run the system and only costs based on the number of delay measurements executed. However, it does not allow low-level network protocol access needed for our delay measurements processes, such as UDP or ICMP. Meanwhile, delay measurement process using a high-level network protocol, e.g., HTTPS, resulted in inconsistent results due to related CSP's infrastructure and protocol overhead, which could affect the CfB user's possible location to be inaccurate.

Seventeen virtual machines (VMs) are then deployed as the active landmark servers using AWS Elastic Cloud Computing (EC2)[21], GCP Compute Engine (CE)[22], and Azure Virtual Machines (VM)[23]. The deployed VMs have the minimum computing specification and are located across Europe following the availability region of the CSPs, as can be seen in Table 4.1. The city center's coordinate of the data center is chosen as the representative location of each landmark since the CSPs do not disclose the exact coordinate of their data centers.

[19] https://aws.amazon.com/lambda/

[20] https://cloud.google.com/functions

[21] https://aws.amazon.com/ec2/

[22] https://cloud.google.com/compute

[23] https://azure.microsoft.com/services/virtual-machines/

4.5.3 Delay-based Geolocation Algorithms

A delay-based geolocation algorithm is a method of binding an Internet-connected device to a geographic location based on the observed network delay between Internet-connected device and a set of landmark servers [AMV17]. In general, different delay-based geolocation algorithms utilize the delay, which is the duration of a one-way packet traveling from sender to receiver, or the latency, which is the round-trip-time (RTT) of the packet traveling from the sender to the receiver to the sender again where it is assumed is double of the delay.

Therefore, CloudRAID for Business implements two basic delay-based geolocation methods using the delay between the CfB users, active landmarks, and passive landmarks to calculate the CfB user's location and verify the CfB users to be at the allowed location: Constraint-based geolocation (CBG) [Gue+06], and GeoWeight [AKK10a].

Constraint-based Geolocation

Constraint-based geolocation (CBG) [Gue+06] is one of the first delay-based geolocation measurement algorithms introduced by Gueye et al.. It establishes a dynamic relationship between network delay or latency with geographic distance between the landmarks and the Internet host as the target.

CBG algorithm utilizes the absolute physical lower bound called "baseline" as the assumption where the packet's travel speed will not exceed 2/3 of the lightspeed or 1 ms round trip-time per 100 km of cable. Each landmark L_i needs to compute a "bestline" using the delay d_{ij} and geographical distance g_{ij} between the landmarks where $i \neq j$ [Gue+06].

The landmark first needs to calculate the possible bestline possibilities where it should be closest to and below all data points but above the baseline that can be calculated using [Gue+06]:

$$y - \frac{d_{ij} - b_i}{g_{ij}} x - b_i \geq 0, \forall i \neq j \tag{4.1}$$

where the intercept for the landmark b_i and the gradient $m_i = (d_{ij} - b_i)/g_{ij}$. It then determines the bestline by selecting the line equation with positive gradient equals or bigger than the gradient of the baseline and non-negative intercept using the objective function [Gue+06]:

$$\min_{\substack{b_i \geq 0 \\ m_i \geq m}} \left(\sum_{i \neq j} y - \frac{d_{ij} - b_i}{g_{ij}} x - b_i \geq 0 \right) \tag{4.2}$$

The landmark calculates the geographic distance of the target τ based on its delay measurement result to the target using the bestline equation. The estimated geographic distance between the landmark and the target $\hat{g}_{i\tau}$ could then derived using the equation [Gue+06]:

$$\hat{g}_{ir} = \frac{d_{ir} - b_i}{m_i} \tag{4.3}$$

Each landmarks's estimated geographic distance to the target is then as the radius for the landmark's circle $C_{i\tau}$ where its center is the coordinate of the landmark. Finally, CBG calculates the target's possible location by finding the intersection region of all circles of the landmarks where its centroid is considered as the user's possible coordinate [Gue+06].

CBG algorithm requires a "brute-force" approach to calculate the landmark's bestline by finding the most suitable bestline candidate out of two points nearest to the baseline, and below all data points as can be seen in Equation 4.2 [Gue+06]. The number of landmark's bestline calculations could be reduced by excluding the points below the baseline, points with zero kilometers, and points with identical distances but at a lower time. If the approach could not calculate a valid bestline, the bestline is then set through the origin and the optimal points. If the calculated bestline is below the baseline, the baseline is then set as the landmark's bestline. Figure 4.2 illustrates the calculation optimization for CBG's landmark bestline.

GeoWeight

GeoWeight [AKK10b] is a delay-based geolocation algorithm proposed by Arif et al. that aims to improve the accuracy of Internet geolocation methods utilizing maximum and minimum bounds of the distance to latency relationship, e.g., CBG [Gue+06]. It considers possible variability of distance-latency relationship where a certain latency could result in multiple possible geographical distances. As each landmark measures the latency with other landmarks, an observable latency

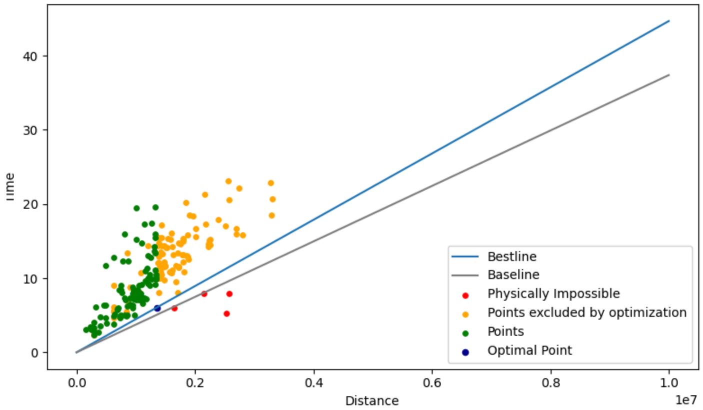


Figure 4.2: Landmark bestline's calculation optimization implemented for Constraint-based Geolocation algorithm

of the landmark t_x might have maximum possible distance $Dmax$ and minimum possible distance $Dmin$ with T_{min} and T_{max} be the minimum and maximum observed latencies. The distance range and time range are then divided into N_d and N_t of equal-sized distance bin [AKK10b].

The weight w_{ij} is assigned for each c_{ij} data point corresponding to the i-th time bin and j-th distance bin that represent the probability of the latency and distance correlation. Each landmark needs to calculate the normalized distance-latency $NR_{i,j}$ by dividing the number of correlated latency-distance by the number of measurements for the particular distance bin using [AKK10b]:

$$NR_{i,j} = c_{i_j} / \sum_{i=1}^{N_t} c_{i_j}$$

The weight for each correlated distance-latency is then computed by normalizing the latency bins using [AKK10b]:

$$w_{i,j} = NR_{i_j} / \sum_{i=1}^{N_d} NR_{i_j}$$

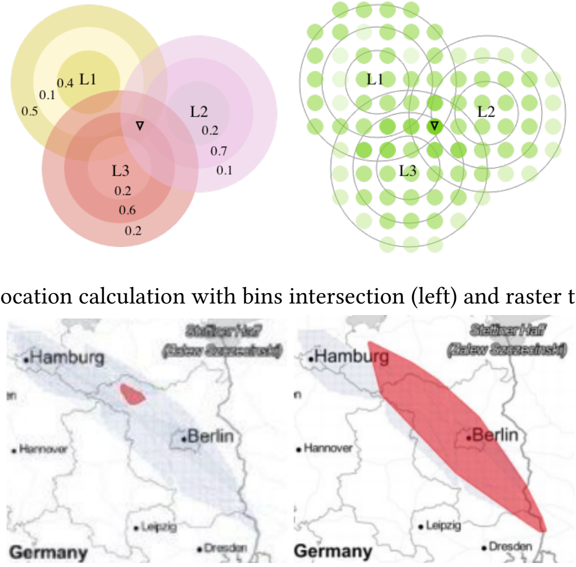
The landmarks then measure the latency with the target user, where the measurement result generates intersection regions. The user's latency measurement is then mapped to each landmark's latency and distance bin, where the intersection regions are then computed by adding the weights of the overlapping bins. The intersection region with the highest weight is then considered as the user's predicted location region [AKK10b].

There are several issues with the GeoWeight algorithm to calculate CfB user's location. The location calculation with the GeoWeight algorithm is computationally expensive since it requires to dynamically calculate the distance and latency bins and multiple intersection regions of the bins around the landmarks with their weights. The size of the user's calculated location with the GeoWeight algorithm is very small and might not include the user's actual location. This could create a "hit-or-miss" access control decision since the CfB system might wrongly determine the CfB user to be or not to be in the allowed location; thus, it could incorrectly deny or authorize CfB user's file access request. Therefore, several modifications are implemented to the GeoWeight algorithm to optimize the user's location calculation for the CfB system.

GeoWeight utilizes a fixed number of bins where the bin's range is derived from the minimum, and maximum value of the distance and the delay [AKK10b]. A fixed range and number for the bins are then set, where the distance bins have a size of around 112 kilometers (km), and the delay bins have a duration of 5 milliseconds (ms). This means that the bins are always similarly precise, no matter how big the area is covered.

CfB system utilizes a more efficient GeoWeight's location calculation with **raster technique** instead of using the intersection region with the highest weight of the bins from many landmarks. It establishes a rectangular grid of points for each landmark where each point covers an area of around 225 kilometer squares (km^2) and applies the weight to all points on the overlapping bins where the CfB user's location region can be calculated by finding the polygon closest to the points with the maximum weight. Using the raster technique, the CfB system then could calculate the CfB user's location with runtime from $O(n \cdot m)^2$ to $O(n \cdot m)$ depending on the resolution of the point, where n is the number of landmarks and m is the number of rings.

The location calculation with the raster technique allows the CfB system to customize the calculated location's size by considering all points above a certain percentage of the points with the maximum weight. The original GeoWeight is considered as GeoWeight with a 100% threshold and the bigger the percentage threshold is, the smaller the calculated CfB user's location will be. Figure 4.3



(a) GeoWeight's location calculation with bins intersection (left) and raster technique (right)

(b) Calculated location using original GeoWeight (left) and GeoWeight with 95% threshold (right)

Figure 4.3: Example of the location calculation difference between the original GeoWeight (left) and the optimized GeoWeight (right) [Suk+21b]

shows the difference between the original GeoWeight algorithm and the modified GeoWeight algorithm implemented for the CfB system.

4.5.4 Third-Party Open Source Intelligent Services

With the CfB system enforces location-based access control with Internet-based location for restricting access to the CfB user's files only at the allowed location, it requires a vast amount of location information to ensure only the authorized and correct CfB users to access the files. It then needs a lot of effort and resources to constantly gather and maintain the location information to be up-to-date; otherwise, it could lead to false file access control decisions of authorized CfB users to be unable or unauthorized CfB users to be able to access the file.

The CfB system utilizes third-party open-source intelligence (OSINT) services where CfB user's submitted location inputs are sent to the OSINT services via the CfB main server to receive the location information about the CfB users for two main reasons. First, the OSINT services are considerably cheaper and easier to use than the CfB system's self-maintaining location information knowledge base as it constantly updates and manages the location information. Second, the CfB system could use multiple different OSINT services for each location information inputs to increase the confidence of the location result. If the OSINT services return different locations results for one location information input, the CfB system could decide the final location result based on the consensus of the location information or the location information provided by the priority OSINT services.

The public IP address could reveal the location of CfB user by mapping it to a geographical location using IP geolocation services, such as ipinfo[24], ipgeolocation[25], and GeoLite2 City database[26]. The services will return the presumed user's geographic coordinate where it is converted to city and country location using Google Geocoding API[27].

The list of surrounding Wi-Fi APs collected by the CfB client application, if the CfB user's device supports the functionality, is sent to Wi-Fi AP geolocation services, such as Google's Geolocation API[28] or WiGLe[29]. The services then presumed user's location of a geographic coordinate with its accuracy radius.

The CfB users might utilize VPN or proxy services to mask their IP address to deceive the CfB system that they are at the allowed location since the IP geolocation services will return the location of the VPN or proxy server's IP address. Therefore, the CfB user's public IP address is then checked against VPN and proxy detection services or databases to ensure it is not manipulated, e.g., proxycheck[30] and IPHub[31]. Since the VPN detection services might be unable to detect if the CfB users are using the company's private VPN or proxy servers, the CfB system could manually add the known IP address of the company's private VPN servers to the IP address whitelist.

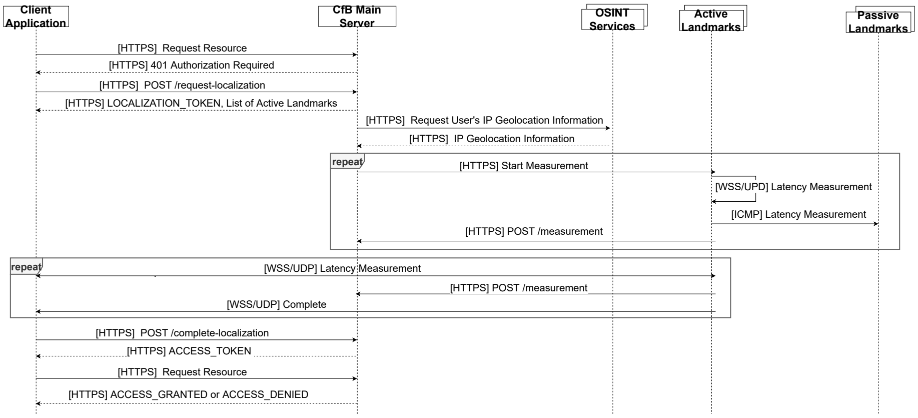

Client Application
CfB Main Server
OSINT Services
Active Landmarks
Passive Landmarks
[HTTPS] Request Resource
[HTTPS] 401 Authorization Required
[HTTPS] POST /request-localization
[HTTPS] LOCALIZATION_TOKEN, List of Active Landmarks
[HTTPS] Request User's IP Geolocation Information
[HTTPS] IP Geolocation Information
repeat
[HTTPS] Start Measurement
[WSS/UPD] Latency Measurement
[ICMP] Latency Measurement
[HTTPS] POST /measurement
repeat
[WSS/UDP] Latency Measurement
[HTTPS] POST /measurement
[WSS/UDP] Complete
[HTTPS] POST /complete-localization
[HTTPS] ACCESS_TOKEN
[HTTPS] Request Resource
[HTTPS] ACCESS_GRANTED or ACCESS_DENIED

Figure 4.4: Sequence diagram of CfB user's client application requests a file with Internet-based location access control restriction enabled [Suk+21b]

4.5.5 Internet-based Location Access Control Decision

The CfB system provides Internet-based location access control for file sharing access restriction to ensure that the authorized CfB users could only be accessed in a certain trusted location or **allowed location**. Figure 4.4 shows the overview of how the CfB system grants or denies CfB user's file access requests based on their Internet-based location.

First, the CfB users request file access to the CfB main server using the CfB client application. If the shared file has a location restriction that can only be accessed in the allowed location, then the CfB main server will deny the request and ask the users to prove that they are at the allowed location. The CfB users then send the available location information inputs to the CfB main server, e.g., IP address and list of surrounding WiFi access points. The LBAC main server sends a localization token and a list of active landmarks to be contacted.

[24] https://ipinfo.io/
[25] https://ipgeolocation.io/
[26] https://dev.maxmind.com/geoip/geoip2/geolite2/
[27] https://developers.google.com/maps/documentation/ geocoding
[28] https://developers.google.com/maps/documentation/geolocation
[29] https://wigle.net/
[30] https://proxycheck.io/
[31] https://iphub.info/

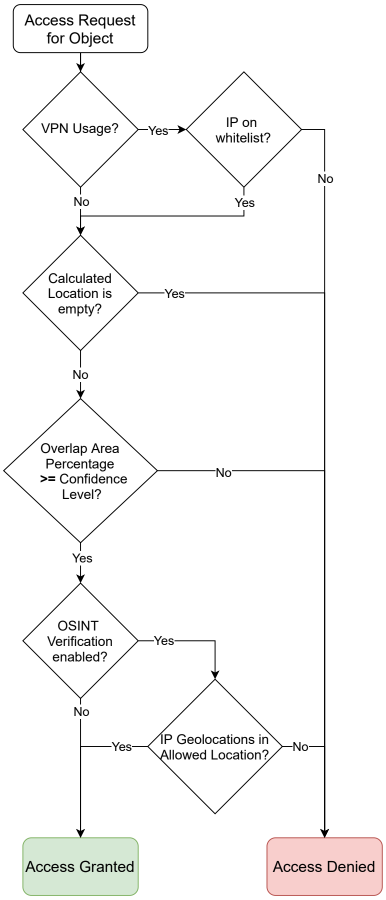

Access Request for Object
VPN Usage?
Yes
IP on whitelist?
No
No
Yes
Calculated Location is empty?
Yes
No
Overlap Area Percentage >= Confidence Level?
No
Yes
OSINT Verification enabled?
Yes
No
Yes
IP Geolocations in Allowed Location?
No
Access Granted
Access Denied

Figure 4.5: CfB system's Internet-based location access control decision flowchart [Suk+21b]

The CfB main server forwards the CfB user's location information inputs to the third-party OSINT services. The OSINT services will return the IP geographical locations and the information if the user is utilizing VPN or proxy services. It also triggers each active landmark **every 5 minutes** to measure the delay with other active and passive landmarks. The active landmarks then regularly send the delay measurement results to the CfB main server.

The CfB client application establishes simultaneous connections with the list of active landmarks using the localization token to circumvent the connection reachability problem. Once the connections have been established, the active landmarks then measure the delay with the client application. Te active landmarks send the delay measurement result with the CfB users to the CfB main server and inform the client application that the measurements are complete. The client application then informs the CfB main server that it has completed all required steps.

Figure 4.5 illustrates how the CfB system determines if the CfB users are granted or denied access to the shared files based on their Internet-based location by considering six parameters:

- **Allowed location**: A set of geographical coordinates describing the administrative boundaries of a city or country in which the file access request should be granted.

- **Calculated location**: A set of geographical coordinates describing the CfB user's location calculated using delay-based geolocation algorithms.

- **VPN usage**: The information gathered from OSINT's VPN or proxy detection services whether the CfB users utilize VPN or proxy services when requesting access to the shared file.

- **Overlap percentage**: The percentage of the overlap region between the allowed location and the calculated location as illustrated in Figure 4.6. There are two methods to calculate the overlap percentage:
  - *Overlap region to calculated location* divides the overlap region's area by the calculated location's area. This method is set by default.
  - *Overlap region to allowed location* divides the overlap region's area by the allowed location's area.

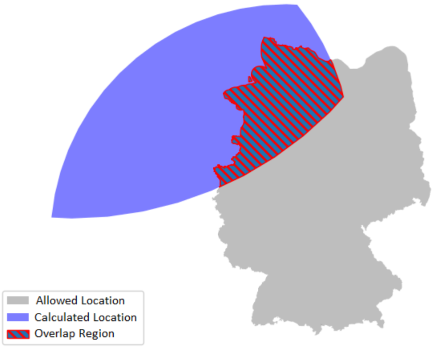


Figure 4.6: Illustration of how overlap region is calculated based on the allowed location and CfB user's calculated location [Suk+21b]

- **Confidence level**: The minimum overlap percentage required to authorize the request with the value ranges between 0.4 to 0.9.
- **IP geolocation verification**: CfB user's location information retrieved from OSINT's IP geolocation services must also fall into the allowed location. The parameter is set to false by default.

The CfB main server checks if the CfB users utilize the VPN or proxy services while requesting access to the shared file by checking the response of OSINT's VPN detection services based on the public IP address of the CfB user and the IP address whitelist. If the VPN detection services return that the CfB users are presumed to use the VPN services, and the IP address is not listed in the whitelist, then the CfB user's file access request is denied.

It then calculates the CfB user's location based on the delay measurement results between the CfB user and the active landmarks and the active landmark's delay measurement results with other landmarks using the delay-based geolocation algorithm. If the delay-based geolocation algorithm fails to calculate the requester's location, the CfB user's calculated location is empty, and the file access request is denied.

The CfB main server then calculates the overlap area percentage between the allowed location and the CfB's calculated location. If the overlap area percentage is less than the specified confidence level, the CfB system determines there is a low probability that the CfB users are in the allowed location; thus, the file access request is then denied.

Suppose the IP geolocation verification is enabled for the Internet-based location access control, the CfB main server verifies if the CfB user's location information provided by OSINT's IP geolocation services also falls in the allowed location to verify the CfB user's calculated location to match the IP geolocation information. If the IP geolocation verification returns false, the file access request is denied, else the access request is granted. If the IP geolocation verification is disabled, the file access request is then automatically granted.

Finally, the CfB main server sends an access token to the CfB users that are later used to request the shared file again. The CfB main server then grants or denies the CfB user's file access request based on the submitted access token.

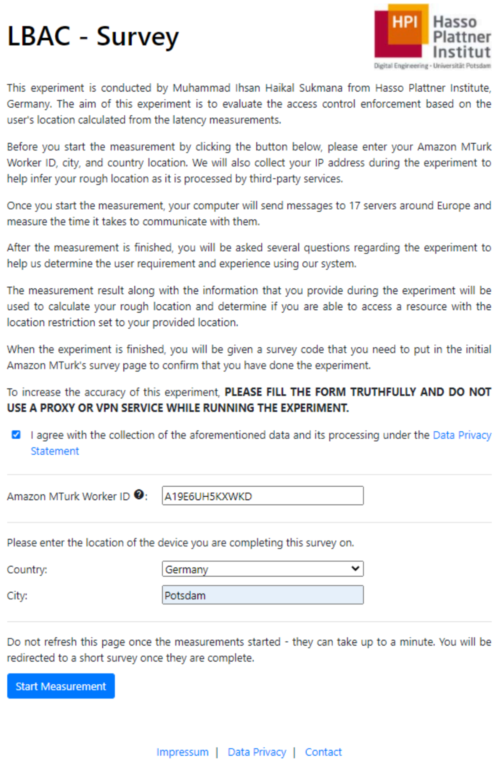
4.6 Evaluation

In this Section, the accuracy, feasibility, and the performance of Internet-based location access control implemented for CloudRAID for Business system's file access sharing restriction is evaluated using Amazon Mechanical Turk (MTurk)[32], which is a crowdsourcing platform that allows individuals and businesses to outsource their jobs to a distributed workforce.

A simple website was created to evaluate the CfB system's Internet-based location access control with Amazon MTurk platform, as can be seen in Figure 4.7. This is due to the Amazon MTurk platform does not allow the MTurk workers to download and install "unknown application" to complete the task, therefore the evaluation process needs to be done with a web browser.

A task was created on Amazon MTurk platform for 100 MTurk workers from central, western, and northern Europe, such as Germany, United Kingdom, Sweden, and Switzerland, where the workers will act as the CfB users. The workers could only access the CfB system's evaluation website once as the website will launch the delay measurement process with the active landmarks and gather necessary worker's information IP address, worker ID, and the delay measurement results with the active landmarks. The workers were required as well to enter the city and country information where the worker resides and answer several survey questions to collect information and expectation regarding location-based access control functionality. Once the evaluation is finished, the website will generate a survey code where the worker will submit the code to the Amazom MTurk's platform to indicate that the evaluation is finished.

[32] https://www.mturk.com/



LBAC - Survey

This experiment is conducted by Muhammad Ihsan Haikal Sukmana from Hasso Plattner Institute, Germany. The aim of this experiment is to evaluate the access control enforcement based on the user's location calculated from the latency measurements.

Before you start the measurement by clicking the button below, please enter your Amazon MTurk Worker ID, city, and country location. We will also collect your IP address during the experiment to help infer your rough location as it is processed by third-party services.

Once you start the measurement, your computer will send messages to 17 servers around Europe and measure the time it takes to communicate with them.

After the measurement is finished, you will be asked several questions regarding the experiment to help us determine the user requirement and experience using our system.

The measurement result along with the information that you provide during the experiment will be used to calculate your rough location and determine if you are able to access a resource with the location restriction set to your provided location.

When the experiment is finished, you will be given a survey code that you need to put in the initial Amazon MTurk's survey page to confirm that you have done the experiment.

To increase the accuracy of this experiment, **PLEASE FILL THE FORM TRUTHFULLY AND DO NOT USE A PROXY OR VPN SERVICE WHILE RUNNING THE EXPERIMENT.**

☑ I agree with the collection of the aforementioned data and its processing under the Data Privacy Statement

Amazon MTurk Worker ID: A19E6UH5KXWKD

Please enter the location of the device you are completing this survey on.

Country: Germany

City: Potsdam

Do not refresh this page once the measurements started - they can take up to a minute. You will be redirected to a short survey once they are complete.

Figure 4.7: A screenshot of Internet-based location access control evaluation website accessed by Amazon MTurk workers

The worker's IP address and city and country information are then parsed to third-party OSINT services, i.e., ipinfo, ipgeolocation, GeoLite2 City database, proxycheck, IPHub, and Google Geolocation, to gather more location information about the worker as explained in Section 4.5.4. Based on the worker's received location inputs, the CfB system will calculate the worker's location and determine if the worker is authorized to access as described in Section 4.5.5.

However, there are several modifications made to the CfB system's Internet-based location access control functionality and its evaluation website to comply with Amazon MTurk's guideline. The delay measurement processes of the MTurk workers using the website with active landmarks and between the active landmarks are done using WebSocket Secure connection only. The worker's location calculation and verification processes do not consider surrounding WLAN access points and GPS coordinate. This is due to the evaluation website developed with JavaScript does not have system-level access to send ICMP or UDP packets to measure the delay with the active landmarks and collect the device's surrounding WLAN access points and GPS coordinate. The evaluation website only records necessary information of the workers during the evaluation process needed to locate and verify the worker's location where some of the personal information of the workers are anonymized following the Amazon MTurk's guideline and European Union's General Data Protection Regulation (GDPR), such as IP address or worker ID.

4.6.1 Internet-based Location Access Control Evaluation Parameters

The location information of city and country voluntary provided by the MTurk worker is used as the **ground truth** to evaluate if the worker's calculated Internet-based location fits the submitted location. There are six parameters used to evaluate the performance and the accuracy of the CfB system's Internet-based location access control functionality:

- **Centroid deviation**: The geographic distance in kilometers between the centroid of worker's submitted city/country information and calculated location from delay-based geolocation algorithm. The lower the value is, the better performance it has.

- **Calculated location size**: The size of the worker's calculated region in square kilometers. The lower the value is, the better performance it has.

- **Update time**: The time delay-based geolocation algorithm takes to initialize the model using the landmark's delay measurement result. The lower the value is, the better performance it has.
- **Calculation time**: The time delay-based geolocation algorithm takes to calculate worker's location using the delay measurement result in second. The lower the value is, the better performance it has.
- **Centroid in the worker's city/country**: The confidence level of whether the worker's calculated location is inside the administrative boundaries of the worker's submitted city/country information. The value could be a boolean (true/false) or a percentage of the average of multiple results. The higher the value is, the better performance it has.
- **Overlap percentage**: The percentage of the worker's calculated location overlapping with the worker's submitted city/country. The higher the value is, the better performance it has.

4.6.2 Virtual Private Network/Proxy Usage Detection

7 out of 100 MTurk workers were excluded from the evaluation process since they are presumed to use a VPN or proxy service during the experiment by one or both VPN/proxy detection services; therefore, they could lie about their submitted locations. These workers were also observed to have unusual high latencies during the evaluation.

Meanwhile, one worker was also excluded from the evaluation since the worker is detected to have high delay measurement result to active landmarks, consistently above 75 ms. This could be caused by a technical issue on the worker's side, such as a bad or unstable Internet connection.

The accuracy of the VPN/proxy detection services used by the CfB system could not be calculated since the worker's voluntarily provided city and country location information used as the ground truth location for the evaluation might be false and could not be verified. The VPN/proxy detection services might also be unable to detect the workers' private or unknown VPN/proxy services. The workers might not disclose the information during the evaluation process.

Another option to detect if the workers are suspected of using VPN/proxy services while requesting access to the shared files is to analyze the latency/delay of the workers [Riv+20]. The worker's connection to the active landmarks with

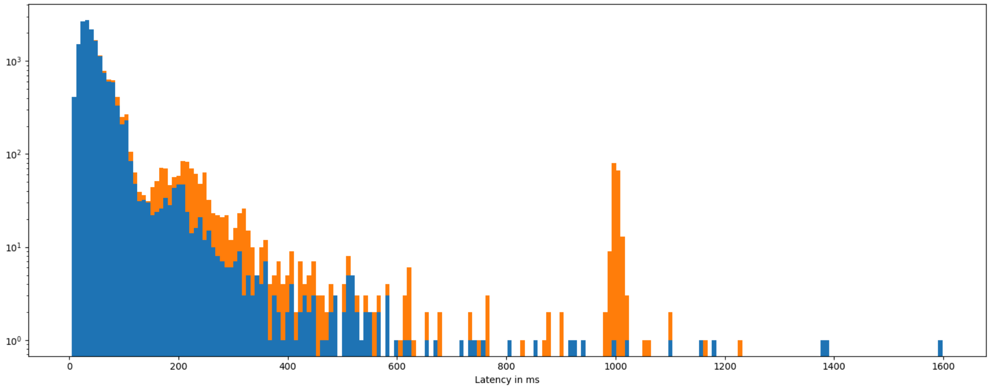


Figure 4.8: Latency comparison between experiment participants presumed to not use VPN or proxy service (blue) with experiment participants suspected to use VPN or proxy service (orange) in logarithmic scale [Suk+21b]

VPN/proxy service is tunneled through the VPN/proxy server, where there will be added delays to the connection depending on the physical network trajectory connection between the workers and the active landmarks. The VPN/proxy connection might add significant delay overhead as the packets need to be encrypted and decrypted as the packets are transmitted between the workers and the active landmarks [Par+10]. Therefore, in theory, the workers suspected of using VPN/proxy service should have a higher latency than the non-suspected workers from the same city and country.

Figure 4.8 shows the histogram of the delay measurement results of all MTurk workers to the active landmarks. The delay measurement result of the workers suspected to use VPN/proxy service is higher than non-suspected workers. However, several non-suspected workers with high latency could not be distinguished from the suspected workers, which a bad or unstable Internet connection might cause. Although the VPN detection method analyzing the delay/latency has a promising potential for the CfB system, it would require more delay measurement results with and without the VPN/proxy services and various landmarks to evaluate the effectiveness and accuracy of the method.

Source	Average (km)	Maximum (km)	90th Percentile (km)
ipinfo.io	88.02	1480.08	254.92
ipgeolocation.io	245.06	1478.09	542.47
GeoLite2 City	99.98	1598.41	339.24
All	144.35	1598.41	473.08

Table 4.2: Comparison of the difference between the location information provided by MTurk workers and OSINT's IP geolocation services with worker's IP address [Suk+21b]

4.6.3 IP Geolocation Services Accuracy

The CfB system utilizes OSINT's IP geolocation services of ipinfo.io, ipgeolocation.io, and GeoLite2 City to gather the location information of the workers based on their IP address. The accuracy of the location information provided by OSINT's IP geolocation services is essential as it could affect the CfB system's access control decision to deny or authorize CfB users wrongly. It could be evaluated by calculating the difference of location information provided by the workers against the IP geolocation services based on the participant's IP address, assuming the worker's submitted location information and IP address are not manipulated.

Table 4.2 shows the statistic of the distance of location information between the worker's provided location and OSINT's IP geolocation services based on the worker's IP address where the lower the distance, the more accurate the OSINT's IP geolocation services. ipinfo.io provides the "most accurate" location information of the workers where over 60% of the result has 0 to 25 km of deviation. Meanwhile, ipgeolocation.io provides the least accurate location information where 21% of the result has 0 to 25 km of deviation.

The location information provided by all IP geolocation services could be combined to increase the confidence of the worker's location information. The combined location information could provide "accurate" location information of the workers, however it also has a large margin of error up to 1598 km difference following GeoLite2 City's maximum location information difference.

From	To	Method	Mean	Median	Min	Max
MTurk Workers	AWS Frankfurt	WSS	28.82	22.51	17.98	71.44
	GCP Frankfurt	WSS	20.18	19.48	17.39	25.57
	Azure Frankfurt	WSS	23.01	22.25	19.05	34.23
AWS London	AWS Frankfurt	WSS	15.03	14.86	13.03	20.23
	GCP Frankfurt	WSS	14.32	14.32	13.18	16.17
	Azure Frankfurt	WSS	15.53	15.4	14.63	17.25
	Speedtest Frankfurt	ICMP	15.29	13.4	12.3	23
GCP London	GCP Frankfurt	WSS	18.76	17.95	13.14	35.02
	AWS Frankfurt	WSS	16.94	16.57	14.36	27.367
	Azure Frankfurt	WSS	17.23	16.81	14.98	25.19
	Speedtest Frankfurt	ICMP	21.26	17.65	12.2	38.3
Azure London	Azure Frankfurt	WSS	16.43	16.41	14.65	24.03
	GCP Frankfurt	WSS	15.68	15.71	14.63	17.26
	AWS Frankfurt	WSS	15.01	14.91	14.02	16.74
	Speedtest Frankfurt	ICMP	16.18	14.9	14.6	20.4

Table 4.3: Overview latency measurement results of MTurk workers in London, United Kingdom with active landmarks in Frankfurt, Germany and active landmarks in London with active and passive landmarks in Frankfurt [Suk+21b]

4.6.4 Latency Measurement Result Comparison

The evaluation generated 1700 latency measurements between MTurk workers and virtual machines (VMs) hosted in AWS, GCP, and Azure as the active landmarks. There are 3,946,782 latency measurements generated during the evaluation and review phase between the active landmarks and active landmarks to 200 servers in the Speedtest network as the passive landmarks. Throughout the evaluation, 15 of the passive landmarks were unreachable, possibly due to servers were down. Another four passive landmarks have less than 75% unusually low number of measurements, where two of them were found unreachable during the review phase.

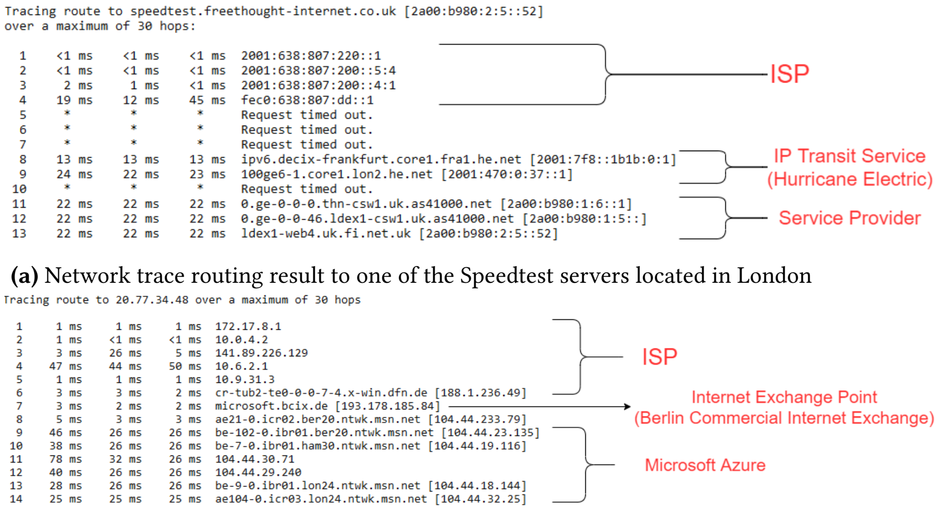
The latency measurement results are analyzed to determine the behavior of the Internet connection from each entity, where it is later divided by two to obtain the delay, which is then used to calculate the worker's location using the delay-based geolocation algorithms. Table 4.3 shows the overview of the latency measurement results between Frankfurt, Germany, and London, the United Kingdom, between the entities.

The latency measurement between the active landmarks and active landmarks with passive landmarks using WSS and ICMP generates almost similar results with the difference of maximum around 1 ms that proves the assumption in the Section 4.5.1. The passive landmarks are only reachable via ICMP or ping request that works in Open Systems Interconnection (OSI) model's network layer (layer 3). Meanwhile, the active landmarks are using WSS to measure the latency with the workers and other active landmarks that work in the application layer (layer 7) of the OSI model. 1 ms is still a tolerable error rate for the CfB system's implemented delay-based geolocation algorithms to calculate the worker's location where the calculated location size is quite large.

The delay-based geolocation algorithms assume that there is a connection between the distance and latency, where the farther the distance between two entities is, the higher latency the connection between two entities should have, and vice-versa [AMV17]. However, the Internet connection does not use a direct line between two entities but rather through various intermediate nodes that affect the latency between the entities. The latency could also be affected by several factors, e.g., transmission delay, queue in the routing path, or delayed acknowledgment [Høi+16]. This makes the latency measurement results between the entities vary, as can be seen in Table 4.3.

In general, there is a high discrepancy between the delay measurement results of MTurk workers to active landmarks with active and passive landmarks. The workers are assumed to use residential or mobile Internet connections with various bandwidths and best-effort connections to the active landmarks while doing the evaluation, where they possibly do not have any control over some of the factors to improve the latency measurement result [Bri+14; Cos20]. Suppose the workers have a slow or unstable Internet connection while doing the evaluation. In that case, it could generate high latency measurement results to the active landmarks as shown in Figure 4.8.

Meanwhile, active landmarks and passive landmarks hosted in the CSPs or service providers utilize commercial Internet connection where it provides symmetrical high bandwidth connection, guaranteed service level agreements, and static IP addresses for reliable and secure connection [Cos20]. This would make the active landmarks have a low latency connection with other active landmarks and passive landmarks.



```
Tracing route to speedtest.freethought-internet.co.uk [2a00:b980:2:5::52]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  2001:638:807:220::1
  2    <1 ms    <1 ms    <1 ms  2001:638:807:200::5:4
  3     2 ms     1 ms    <1 ms  2001:638:807:200::4:1
  4    19 ms    12 ms    45 ms  fec0:638:807:dd::1
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8    13 ms    13 ms    13 ms  ipv6.decix-frankfurt.core1.fra1.he.net [2001:7f8::1b1b:0:1]
  9    24 ms    22 ms    23 ms  100ge6-1.core1.lon2.he.net [2001:470:0:37::1]
 10     *        *        *     Request timed out.
 11    22 ms    22 ms    22 ms  0.ge-0-0-0.thn-csw1.uk.as41000.net [2a00:b980:1:6::1]
 12    22 ms    22 ms    22 ms  0.ge-0-0-46.ldex1-csw1.uk.as41000.net [2a00:b980:1:5::]
 13    22 ms    22 ms    22 ms  ldex1-web4.uk.fi.net.uk [2a00:b980:2:5::52]
```

(a) Network trace routing result to one of the Speedtest servers located in London

```
Tracing route to 20.77.34.48 over a maximum of 30 hops

  1     1 ms     1 ms     1 ms  172.17.8.1
  2     1 ms    <1 ms    <1 ms  10.0.4.2
  3     3 ms    26 ms     5 ms  141.89.226.129
  4    47 ms    44 ms    50 ms  10.6.2.1
  5     1 ms     1 ms     1 ms  10.9.31.3
  6     3 ms     3 ms     2 ms  cr-tub2-te0-0-0-7-4.x-win.dfn.de [188.1.236.49]
  7     3 ms     2 ms     2 ms  microsoft.bcix.de [193.178.185.84]
  8     5 ms     3 ms     3 ms  ae21-0.icr02.ber20.ntwk.msn.net [104.44.233.79]
  9    46 ms    26 ms    26 ms  be-102-0.ibr01.ber20.ntwk.msn.net [104.44.23.135]
 10    38 ms    26 ms    26 ms  be-7-0.ibr01.ham30.ntwk.msn.net [104.44.19.116]
 11    78 ms    32 ms    26 ms  104.44.30.71
 12    40 ms    26 ms    26 ms  104.44.29.240
 13    28 ms    26 ms    26 ms  be-9-0.ibr01.lon24.ntwk.msn.net [104.44.18.144]
 14    25 ms    25 ms    25 ms  ae104-0.icr03.lon24.ntwk.msn.net [104.44.32.25]
```

(b) Network trace routing result to Microsoft Azure VM service's IP address London region

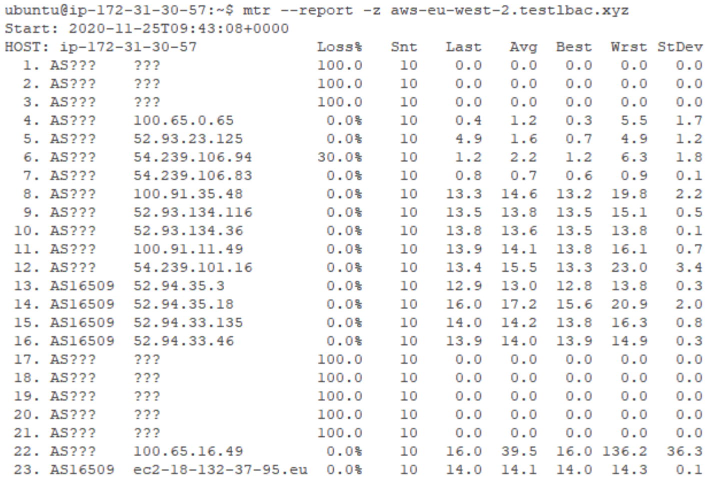
Figure 4.9: Network trace routing results from Potsdam, Germany to landmark servers in London, United Kingdom that show the usage of Internet exchange point and IP transit service

The CSPs and the service providers also utilize Internet/cloud exchange points and IP transit services to provide optimized routing and lower latency connection with the requesters, e.g., Berlin Commercial Internet [33] and Hurricane Electric[34]. This could be seen by analyzing the output of network trace routing result as can be seen in Figure 4.9.

The servers from the Speedtest network utilize IP transit services to facilitate the connection between two entities through best-effort public Internet with added benefits, such as lower latency, fewer hops, and high bandwidth up to 100 Gbps or more [Ahm+17; Hur21]. In Figure 4.9 (a), the packet from Potsdam went through to Hurricane Electric's Frankfurt and London transit points before being sent to the Speedtest server in London.

Each CSP has the peering policy to optimize the traffic exchange between the data centers spread across the world and provide the best connection for its customers with an optimized routing path and low latency, including direct

[33] https://www.de-cix.net/
[34] https://he.net/



```
ubuntu@ip-172-31-30-57:~$ mtr --report -z aws-eu-west-2.testlbac.xyz
Start: 2020-11-25T09:43:08+0000
HOST: ip-172-31-30-57            Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. AS???    ???                100.0    10    0.0   0.0   0.0   0.0   0.0
  2. AS???    ???                100.0    10    0.0   0.0   0.0   0.0   0.0
  3. AS???    ???                100.0    10    0.0   0.0   0.0   0.0   0.0
  4. AS???    100.65.0.65         0.0%    10    0.4   1.2   0.3   5.5   1.7
  5. AS???    52.93.23.125        0.0%    10    4.9   1.6   0.7   4.9   1.2
  6. AS???    54.239.106.94      30.0%    10    1.2   2.2   1.2   6.3   1.8
  7. AS???    54.239.106.83       0.0%    10    0.8   0.7   0.6   0.9   0.1
  8. AS???    100.91.35.48        0.0%    10   13.3  14.6  13.2  19.8   2.2
  9. AS???    52.93.134.116       0.0%    10   13.5  13.8  13.5  15.1   0.5
 10. AS???    52.93.134.36        0.0%    10   13.8  13.6  13.5  13.8   0.1
 11. AS???    100.91.11.49        0.0%    10   13.9  14.1  13.8  16.1   0.7
 12. AS???    54.239.101.16       0.0%    10   13.4  15.5  13.3  23.0   3.4
 13. AS16509  52.94.35.3          0.0%    10   12.9  13.0  12.8  13.8   0.3
 14. AS16509  52.94.35.18         0.0%    10   16.0  17.2  15.6  20.9   2.0
 15. AS16509  52.94.33.135        0.0%    10   14.0  14.2  13.8  16.3   0.8
 16. AS16509  52.94.33.46         0.0%    10   13.9  14.0  13.9  14.9   0.3
 17. AS???    ???                100.0    10    0.0   0.0   0.0   0.0   0.0
 18. AS???    ???                100.0    10    0.0   0.0   0.0   0.0   0.0
 19. AS???    ???                100.0    10    0.0   0.0   0.0   0.0   0.0
 20. AS???    ???                100.0    10    0.0   0.0   0.0   0.0   0.0
 21. AS???    ???                100.0    10    0.0   0.0   0.0   0.0   0.0
 22. AS???    100.65.16.49        0.0%    10   16.0  39.5  16.0 136.2  36.3
 23. AS16509  ec2-18-132-37-95.eu 0.0%    10   14.0  14.1  14.0  14.3   0.1
```

Figure 4.10: Network trace routing result between the landmarks hosted in AWS EC2 from Frankfurt region (eu-central-1) to London region (eu-west-2)

connection to the resources in the CSP [Ahm+17; Ama21d; Mic20e]. It utilizes Internet/cloud exchange points and Autonomous Systems (AS) for direct and secure layer 2 and 3 connectivity across different network domains to bypass the public Internet [Yeg+19; Yeg+20].

For example, AWS provides 25 regions as by 2021 that covers more than 80 cities and 200 Points of Presence. To support that, AWS has the autonomous system AS16509 and 98 public Internet exchange points that would allow dedicated connections with optimized direct routing and high bandwidth between the requester and AWS [Ama21d]. It also supports virtual private interconnection that enables the requesters to connect to AWS without the need to have AS by connecting to the private peering facilities [Yeg+19]. The peering policy could be observed in the network trace routing result between the VMs hosted in AWS EC2 from different regions in Figure 4.10 where the hops have the IP addresses belong to AS16509 or owned by the AWS.

4.6.5 GeoWeight Threshold Percentage Variants

The CfB system's implementation of GeoWeight algorithm [AKK10b] allows for the threshold percentage to be specified that affects the size of calculated location size as explained in Section 4.5.3. The threshold percentages of 50, 60, 70, 75, 80, 90, and 100 (the original GeoWeight) are then evaluated to determine which percentage should be used for the GeoWeight algorithm for further evaluation.

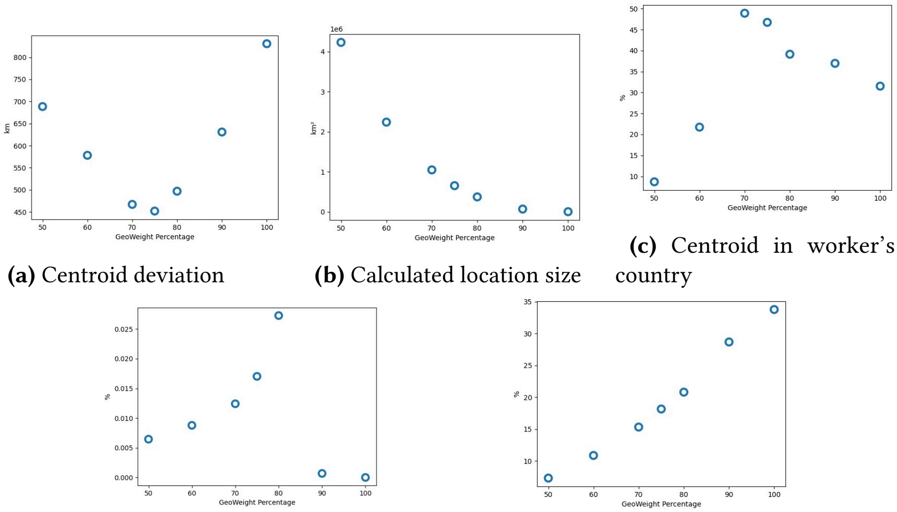
As can be seen in Figure 4.11, GeoWeight with 70%, 75%, and 80% thresholds provide good performance for centroid deviation, centroid in the worker's country, and overlap calculated location parameters. Original GeoWeight with 100% threshold performs the best for calculated location size and overlap calculated location with the worker's country parameters. In contrast, it performs poorly for centroid deviation and centroid in worker's country parameters. All percentage threshold variants take almost similar time of around 55 milliseconds to initialize and 4 to 4.5 seconds to calculate the worker's location. All percentages perform poorly for centroid in the worker's city parameter.

The GeoWeight's threshold percentage affects the size of the calculated location that determines the possibility of the calculated location in the worker's submitted location. However, it might be better for the calculated location to intersect with the worker's country than its city. **GeoWeight with 75% threshold** is then concluded as the best overall performing GeoWeight's percentage threshold. Geoweight with 75% and 100% threshold are used for further evaluation.

4.6.6 Landmarks Comparison

The CfB system utilizes the active landmarks hosted in AWS, GCP, and Microsoft Azure and the passive landmarks of 200 servers from the Speedtest network to establish the baseline needed to calculate the worker's location using the delay-based geolocation algorithms. The effect of calculating the worker's location using the delay-based geolocation algorithms with the delay measurement results from only active landmarks and active landmarks with passive landmarks as "all landmarks" is evaluated, as can be seen in Figure 4.12.

The original GeoWeight with 100% threshold benefits from including the delay measurement results from passive landmarks where all key figures improve by at least 29% compared with relying on only the delay measurement results from active landmarks. For GeoWeight with 75%, the delay measurement from all landmarks affects the calculated location's centroid to have a lower deviation to the worker's submitted country information and a higher percentage of it in the submitted country. However, the size of the calculated location is also 397% larger than the average. Finally, the delay measurements from all landmarks somehow affect the CBG to have a bigger calculated location size that slightly



(a) Centroid deviation

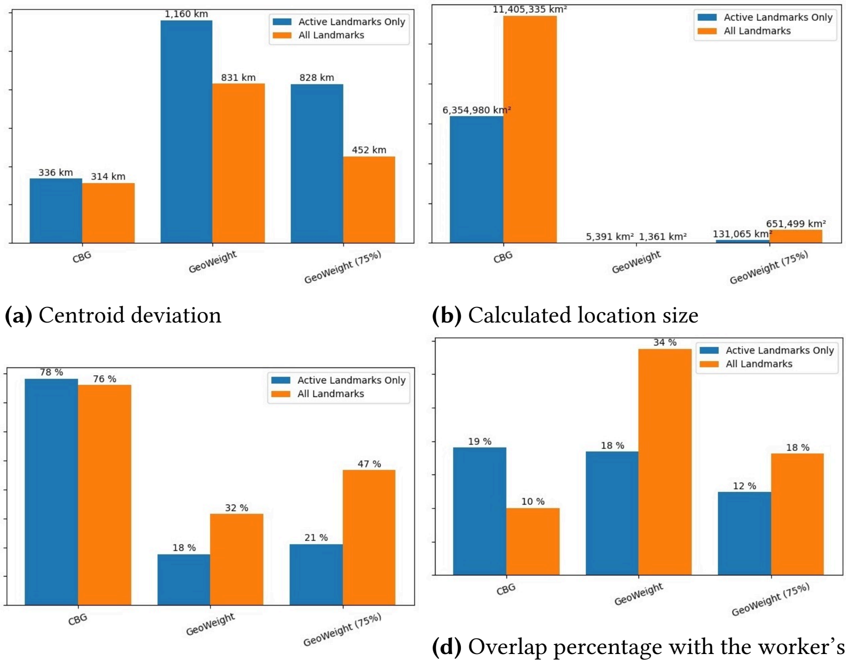
(b) Calculated location size

(c) Centroid in worker's country

(d) Overlap calculated location with worker's city

(e) Overlap percentage with worker's country

Figure 4.11: Performance comparison of GeoWeight with various threshold percentages [Suk+21b]



(a) Centroid deviation

(b) Calculated location size

(c) Centroid in the worker's country

(d) Overlap percentage with the worker's country

Figure 4.12: Comparison of location calculation using only active landmarks against active and passive landmarks (all landmarks) [Suk+21b]

improves the centroid deviation and centroid to be in the worker's country. However, it makes the overlap percentage to the country to be lower than the delay measurements from only active landmarks.

The delay-based geolocation algorithms using the delay measurement results from all landmarks perform better than the delay measurement results only from active landmarks. The landmark's delay measurement results impact the algorithms' initialization, thus influencing the accuracy of the calculated location. CBG algorithm utilizes the landmark's delay measurement result to calculate the landmark's bestline equation. If the delay measurement points from all landmarks contain outliers, the landmark could have the incorrect bestline equation. Meanwhile, the GeoWeight could benefit from the delay measurement results from all landmarks to calculate the weight matrix of each landmark that requires many measurement points to fill the cells of the weight matrix evenly.

4.6.7 Delay-based Geolocation Algorithms Performance

The performance of CBG and GeoWeight algorithms with 75% and 100% thresholds (original GeoWeight) is evaluated using the latency measurement results of the landmarks and MTurk workers, which are divided by two, to calculate the worker's location. The worker's calculated location is then compared with the supplied location to evaluate its accuracy.

The CBG algorithm performs the best for centroid deviation and centroid in the worker's country parameters. However, the size of the calculated location using CBG is enormous that it could encapsulate multiple European countries or even the European continent. The calculated location has low accuracy as the worker's actual location could lie anywhere inside the large calculated location.

Meanwhile, the original GeoWeight with a 100% threshold performs the best for calculated location size and overlap of calculated location with the worker's country. This is because GeoWeight's calculated location has the smallest size compared to other algorithms, although it performs the worst for the centroid deviation parameter.

GeoWeight with 75% threshold seems to be the most balanced delay-based geolocation algorithm where it comes in second place in every evaluation parameter. All delay-based geolocation algorithms require up to **1.3 seconds to initialize and 4 seconds to calculate the worker's location**. The algorithms have **bad city-level accuracy** since the worker's calculated location and its centroid do not intersect with the worker's submitted city information.

4.6.8 Access Control Decision Result

The accuracy of the CfB system's Internet-based access control decision based on the worker's Internet-based location as explained in Chapter 4.5.5 is evaluated. **Italy** is selected as the allowed location where 49 out of 92 MTurk workers are located. Only the workers that are confirmed to be at the allowed location will be authorized to access the shared CfB user's files. Several test scenarios were created using the combination of three main configurable access control decision configurable parameters to determine the effect of the parameters to the access control decision's result, particularly confidence level, overlap percentage calculation method, and OSINT verification parameters.

Each test scenario generates a confusion matrix consisting of four possible results that could be generated from the CfB system's Internet-based location

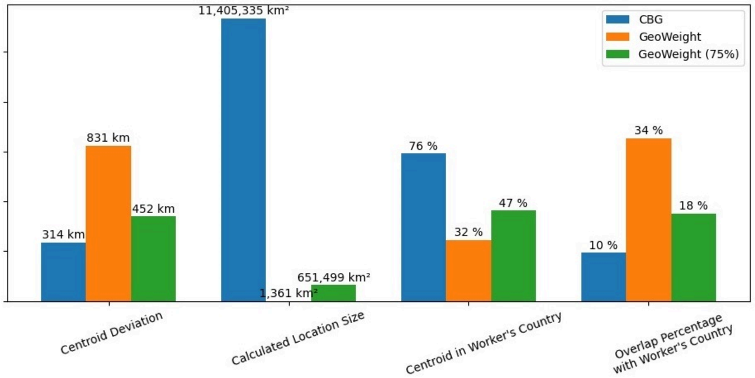
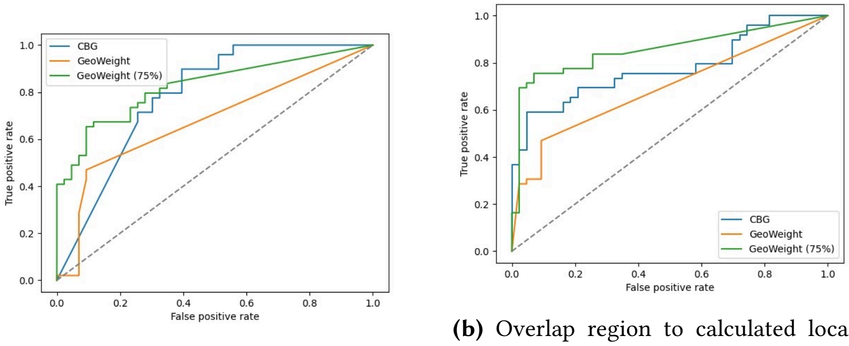


Figure 4.13: Delay-based geolocation algorithm's performance comparison between CBG and GeoWeight with 75% and 100% (original GeoWeight) thresholds [Suk+21b]

access control decision process: **true-positive** if the worker is correctly determined to be in the allowed location; **true-negative** if the worker is correctly determined to be outside of the allowed location; **false-positive** if the worker is wrongly determined to be in the allowed location; and **false-negative** if the worker is wrongly determined to be outside of the allowed location. The objective is to minimize the number of false-positive and false-negative results while maximizing the number of true-positive and true-negative results.

The accuracy of the calculated location and the method for calculating the overlap region with the allowed location play essential roles in the Internet-based location access control decision. As the confidence level parameter increases, the number of true-positive and false-positive results decrease. This is because the overlap region needs to satisfy the confidence level set, which indicates a high probability of the requester's calculated location or includes the allowed location. If the overlap region could not fulfill the confidence level, then the request is denied.

The CBG algorithm's calculated location might include multiple European countries, which could create high true-positive and false-positive results when overlap percentage is calculated using *overlap region to allowed location* method. Meanwhile, the calculated location using the original GeoWeight is relatively small. If it falls into the allowed location, the overlap percentage calculated using the *overlap region to calculated location* method could lead to a decent amount of true-positive and false-negative results. Suppose the overlap percentage is



(b) Overlap region to calculated location

(a) Overlap region to allowed location method method

Figure 4.14: ROC curves for CfB system's Internet-based location access control decision of two overlap percentage calculation methods [Suk+21b]

calculated using *overlap region to allowed location* method. In that case, it will return a small percentage that will not satisfy the confidence level parameter, thus increasing the false-negative results. GeoWeight with a 75% threshold generates the calculated location where most of its parts are outside the allowed region compared with other delay-based geolocation algorithms, which leads to the confusion matrix generating only negative results.

The OSINT verification parameter affects the access control decision to be more restrictive. It requires the location information provided by the OSINT's IP geolocation services to confirm the calculated location to be in the allowed location. This could reduce false-positive results as two different location information are regarded as negative results.

Receiver operating characteristic (ROC) curves are generated based on true-positive and false-positive results of the delay-based geolocation algorithms with various confidence levels between 0 to 100% as shown in Figure 4.14. The random guess line is plotted as a gray dotted line that represents a 50% chance that the worker is randomly allowed or denied access to the files. Overall, **GeoWeight with 75%** provide the best access control decision result among the delay-based geolocation algorithms for Internet-based location access control.

4.7 Discussion

The Internet-based location, in general, has a good potential as a location input for a location-based access control scheme for CfB system with **country-level accuracy**. It could be used to determine the location of CfB users and verify if the CfB user is at the allowed location before giving access to the files with location restrictions. However, several points are raised during the implementation and evaluation of Internet-based location access control for the CfB system.

However, there are several weaknesses of Internet-based location to be a location input for location-based services. The CfB users are required to respond to the delay/latency measurement inquiry process by the active landmarks where the users should have a fast and stable Internet connection; otherwise, it will generate high delay/latency. Meanwhile, malicious CfB users could delay the response sent to the landmarks to increase the delay or latency to the landmarks. High delay/latency with the landmarks could cause inaccurate, false, or even empty calculated locations and increase the false-negative and false-positive access control decision results. The CfB system could set up a maximum threshold of the delay/latency to the landmarks to mitigate this issue. If the CfB users have the delay/latency exceeding the threshold, their file access requests will be denied, and they will be asked to try requesting access to the file later with a better Internet connection.

The CfB system relies on the third-party OSINT services to provide additional information of the CfB user's submitted Internet-based location inputs. However, the CfB users could fake their IP address using VPN/proxy services and submit a list of fake surrounding WiFi APs by gathering the WiFi APs in the allowed location from WiFi AP databases, such as WiGLE. IP geolocation services might provide good country-level accuracy with bad city-level accuracy. However, there is a possibility that the location information provided could be inconsistent or outdated [Gha+17; KVR17]. VPN or proxy detection services and WiFi geolocation services could also provide outdated, false, or empty information as they might not cover the CfB user's submitted IP address or list of surrounding WiFi APs. This could generate false-positive and false-negative access control decisions and affect the CfB system to allow/deny access to the requester incorrectly. The CfB system could maintain a list of allowed/denied IP addresses and WiFi APs to reduce the possibility of manipulated IP-based location inputs.

The CfB system utilizes 17 VMs deployed in the three largest public CSPs on the market in the European region as the active landmarks to measure the delays between the entities. It is considerably cheaper and easier to launch delay measurements than Internet measurement testbed platforms, such as RIPE Atlas. However, the number of active landmarks available is still too small, requiring Speedtest's servers to be used as the passive landmarks to ensure the delay-based geolocation algorithms have enough dataset to calculate the location better, as proven previously. If the CfB system wants to provide Internet-based location access control on a larger scale with better location accuracy, the CfB system would require local CSPs across Europe to host more active landmarks.

Based on the previous evaluation, GeoWeight with a 75% threshold performs the best among other delay-based geolocation algorithms implemented by the CfB system to calculate and verify CfB user's location. Overall, the implemented delay-based geolocation algorithms could generate the location with good country-level accuracy and bad city-level accuracy. It is recommended to use one or several countries as the allowed location for the CfB system to accommodate the CfB user's a calculated location with delay-based geolocation algorithms. For example, depending on the location restriction use case, the CBG algorithm could verify whether the CfB user is in the European Union 1region as the size of the calculated location could cover several European countries due to Europe's geographical characteristics.

The discrepancy of delay/latency measurement results between the CfB users to active landmarks and between the landmarks could affect the accuracy of the CfB user's location calculated using delay-based geolocation algorithms that could affect the accuracy of the CfB user's calculated location. Other non-controllable factors could affect the delay/latency measurements as mentioned in Chapter 4.6.4, such as the routing path, the usage of Internet exchange points or IP transit service, or transmission delay. This could increase the number of false-positive or false-negative access control decision results as the CfB users might be wrongly allowed or denied access to the shared files. The CfB systems would require more diverse measurement results with realistic latency/delay that reflect residential or mobile Internet connection or other delay-based geolocation algorithms that could ignore the discrepancy of delay/latency measurement results to increase the accuracy of the CfB user's calculated location.

The CfB system's Internet-based location access control decision utilizes multiple inputs and customizable parameters as explained in Chapter 4.5.5. The access control decision is based on the overlap region between the allowed location and the CfB user's multiple Internet-based location information based on the submitted location inputs. It increases the confidence that the CfB user is actually at the allowed location to reduce the number of false-positive and false-negative access control decision results. The CfB system could set the restrictiveness of the access control decision by fine-tuning its parameters depending on the use case as proven from the evaluation. If the access control decision is set to be lenient, it would generate more positive results and allow more CfB users to access the shared files. If it is set to strict, it would generate more negative results and deny more CfB users access to the resources.

CfB system could calculate the MTurk worker's location using delay-based geolocation algorithms up to 5 seconds as mentioned in Chapter 4.6.7, in which the total should be less than 10 seconds to calculate and determine if the workers are at the allowed location. This is because the delay-based geolocation algorithms have been initialized using the active landmark's delay measurement results with other landmarks that would enable the requester's calculation to be "instant" based on the delay measurement result with the active landmarks.

4.8 Conclusion and Future Works

In this Chapter, CloudRAID for Business system implements a new location-based access control model utilizing the location inferred from the CfB user's Internet-connecting devices to enforce location-restricted files to be only accessible at the allowed location. The Internet-based location access control utilizes two delay-based geolocation algorithms and third-party open-source intelligence services deployed in the European region of AWS, GCP, and Azure. Based on the evaluation, Internet-based location could be used as a location information input for LBAC to determine the CfB user's location and verify whether the CfB user is at the allowed location with country-level accuracy.

Other delay-based geolocation algorithms could be implemented for the CfB system to improve the performance, accuracy, and effectiveness of the Internet-based location access control, such as Posit by Eriksson et al. [Eri+12], or Octant by Wong et al. [WSS07]. The Internet-based location could utilize the sensor-based location information input to determine and verify the location of the CfB users inside the company's building, such as using Wi-Fi access points or Bluetooth Low Energy [Che+17; Li+19]. The CfB system could also utilize

Geospatial eXtensible Access Control Markup Language (GeoXACML) to store the location restriction of the file's access control policy.

5 Secure Multi-Cloud Storage Environment Management

5.1 Introduction

Cloud object storage service is one of the most used cloud computing services where individuals and enterprises as cloud customers could store unlimited data as objects in the cloud. It provides cheaper data storage and better data availability and scalability compared to in-house data storage, which would require constant maintenance [MTB18]. It is predicted the global market size of the cloud object storage services could reach USD 13.65 Billion with compound annual growth rate (CAGR) of 13.6% in 2028 [Eme21].

Although the cloud service provider (CSP) guarantees that the uptime of cloud object storage services will be up to 99.99% [Ama21b; Goo21e], it could still be susceptible to outage. If the cloud object storage services are unavailable, cloud customers will be unable to access the data stored in the cloud which could affect the availability of dependent services or applications, ultimately creating financial or reputation loss [MTB18]. It could also create vendor lock-in situation where it could be complicated for cloud customers to switch to other CSPs as they rely solemnly on single CSP for storing their data [APW10; OST14].

To resolve the challenges above, more cloud customers are then using cloud object storage services from multiple CSPs to store their data in the cloud, or commonly known as **multi-cloud storage approach** [Raf+17]. The approach utilizes data redundancy techniques, such as erasure code or replication, to store the data or its fragments in various CSPs [Nac+17]. It provides better data availability and service reliability than using cloud object storage services from a single CSP as the data could still be accessed in case one or several CSPs are inaccessible due to outage [MTB18; Nac+17]. The approach becomes more prevalent where according to IDG Communication's Cloud Computing Survey 2020, 47% of small and medium businesses and 66% of large enterprises utilize multiple public clouds for their operations [IDG20].

CloudRAID for Business utilizes cloud object storage services from multiple CSPs following cloud brokerage approach to provide data availability, integrity, and confidentiality for the company's confidential files stored on the cloud as explained in Chapter 2. CfB is then responsible to securely manage the used cloud resources on various CSPs from unauthorized entities following cloud computing's shared responsibility model [Ama20h], including the files stored on the cloud. However, due to heterogeneity of the CSPs in terms of API, data model, and service implementation and the lack of cross-CSP collaboration force CfB to securely manage its cloud resources on its own where the complexity is growing with the number of CSPs subscribed by the cloud customers [RR18].

In this Chapter, a unified multi-cloud storage resource management framework is proposed for CfB for secure, centralize, and automated cloud storage resource management in Amazon Web Services and Google Cloud Platform. The unified cloud storage resource model provides an abstraction model to solve the heterogeneity of the data model of cloud storage resources and access control from different CSPs. A unified multi-cloud storage resource management platform implements the unified resource model built on top of the CSP's native APIs to provide secure cloud storage resource lifecycle management in a multi-cloud storage environment in a single interface. Introduction and guidelines are used during resource management processes to securely manage cloud storage resources and its access for authorized CfB stakeholders.

5.2 Related Works

5.2.1 Research Works

Several works have been proposed throughout the years to manage cloud storage resources in a multi-cloud environment.

Abu-Libdeh et al. [APW10] described Redundant Array of Cloud Storage (RACS), a multi-cloud storage proxy that stands between various CSPs and cloud customers to avoid vendor lock-in and reduce the cost of switching providers. It mimics the interface and data model of AWS S3 service to store the data across various CSPs using the RAID-5 technique. Bessani et al. [Bes+13] implemented DEPSKY, a cloud-of-clouds system called DEPSKY implemented on top of AWS S3, Azure Blob, Nirvanix CDN, and Rackspace Files cloud storage services. The system utilizes erasure code, Byzantine quorum system protocols, Shamir secret sharing scheme, and symmetric cryptography algorithm to provide data availability, integrity, and confidentiality on the cloud. It also proposed a data model with three abstraction levels that provide detailed information of the data and how the data is stored across different CSPs. Hill and Humprey [HH10] presented a CSP vendor-agnostic cloud storage abstraction layer (CSAL) that allows an application to access Blob, Table, and Queue storage services in the multiple CSPs. It utilizes a single namespace across all storage services to maintain the metadata of each storage entity.

Rafique et al. [Raf+17] introduced an adaptive middleware platform for (semi-)autonomous storage architecture management across multiple CSPs for three different scenarios: performance optimization, peak-load condition, and evolving pricing scheme. It continuously monitors the storage system's metrics that allow for identifying the changing condition of the system and optimizing the multi-cloud data placement strategy. Krotsiani and Spanoudakis [KS14] proposed a certification model for non-repudiation in the cloud storage services to ensure neither data owner nor CSP could deny the activities happening in the CSP. It uses a non-repudiation mechanism based on the fair multi-party non-repudiation scheme and continuous monitoring and assessment to detect the anomaly and suspicious behavior. [EFP17] developed a multi-cloud storage broker API to provide portability and easier migration between different CSPs. It is based on a layered ontological framework to map and abstract common functionalities of cloud object storage services.

Celesti et al. [Cel+16; Cel+19] introduced an abstract storage layer on top of Dropbox, Google Drive, and Copy combined with a redundant residue number system (RRNS) algorithm to achieve a reliable hybrid multi-cloud storage environment. Di Pietro et al. [Di +17] implemented Secure Storage in Multi-Cloud Environment (SSME) architecture to provide confidentiality and integrity for a distributed multi-cloud storage system. The architecture utilizes an SSME middleware server that is responsible for secure file management between client applications and multiple cloud storage services, such as file fragmentation and recovery. Tchernykh et al. [Tch+18] introduced a multi-cloud-based storage system called WA-RRNS using a weighted access scheme based on redundant residue number algorithm and Mignotte secret sharing scheme. It provides failure detection and recovery mechanisms to avoid data loss, distortion, corruption, or denial of access. Junghanns et al. [JFE16] presented a cloud gateway system for secure storage in a multi-cloud architecture. It integrates CP-ABE encryption, public key infrastructure, and threshold secret sharing scheme, which consists of Shamir secret sharing scheme, Rabin's information dispersal algorithm, and

Krawczyk secret sharing scheme, to provide data confidentiality and increased share availability on the cloud.

5.2.2 Competitors

Dropbox Business

Dropbox utilizes the hybrid cloud infrastructure consisting of Amazon Web Services and its on-premise data centers to provide distributed data processing, storage, and recovery across the world [Ama20d]. It utilizes a hybrid software stack consisting of various services as a unified set of interfaces and tools to provision, operate, and manage both AWS and on-premise infrastructures and services. The machine management service provides system management capabilities for both AWS and on-premise infrastructure, such as device inventory, service discovery, and remote command execution. The compute, storage, and database hybrid services are capable of provisioning, managing, and operating infrastructure on the hybrid cloud.

Tresorit

Tresorit utilizes Microsoft Azure's datacenters in Ireland and the Netherlands to host the service and store the customer's data and 12 secure datacenters across the world where the customers could choose where the files are stored [Tre20a; Tre21d]. However, no information is found on how Tresorit manages its Azure infrastructure and other datacenters to store the user's files.

Boxcryptor

Boxcryptor allows its users to add the account of more than 30 cloud file storage services to provide an additional security layer of the files storage on the cloud [Box21b]. Boxcryptor could automatically detect and add most cloud file storage services as locations in the Boxcryptor drive, such as by detecting Google Drive's Backup and Sync already installed on the computer. Boxcryptor users could manually add other cloud file storage services as a custom location if it is not detected automatically, although it is not clear how the user could perform that action based on the available information. Boxcryptor also supports other cloud providers and local storage that utilize WebDAV protocol.

5.2.3 Thesis Contribution

The work proposed in this chapter is different from the research community and the competitors to securely manage the multi-cloud storage environment used by CloudRAID for Business to store the company's confidential files across various CSPs. A unified multi-cloud storage resource management framework is proposed to securely and automatically manage cloud resources in multiple CSPs in a single interface. The unified framework resolves the heterogeneity of the data model of cloud storage resources and access control from different CSPs and allows CfB to securely manage cloud resources and their access for authorized CfB stakeholders.

5.3 Multi-Cloud Storage Management for CloudRAID for Business

Cloud computing utilizes complex and distributed hardware and software infrastructure stacks to handle the requests from the clients and the events beyond the control of the clients and the CSP [Lia+17b]. Therefore, the cloud resource management process is important where it manages and allocates available resources in the cloud for the requiring entity to fulfill its requirements and objectives [JS15]. It is part of the shared responsibility model in cloud computing where the CSP and the cloud customers subscribing to the CSP are responsible for managing the security and compliance of the cloud resources [Ama20h].

The objectives of the cloud resource management process vary on the three entities involved in the process:

- **Cloud service provider (CSP)**: The CSP is responsible to manage its hardware and software infrastructure to provide necessary cloud services and resources for its subscribing customers using virtualization techniques [JS15]. It is also responsible to achieve cloud customers' expected level of services based on the agreed Service Level Agreement (SLA). The objective of the cloud resource management process is to optimize the resource utilization to be efficient and effective while saving energy and cost based on the resource usage [Asl+17; MS14].

- **Cloud customer**: Cloud customers subscribe to the CSP to use its services and resources as they expect the CPS to fulfill the level of services based on the agreed SLA. They aim to optimize resource usage efficiently and effectively while saving costs. They are also responsible for "security in the cloud" due to the shared responsibility model, which means they need to correctly and securely configure the owned cloud resources to disallow access from unauthorized entities [Ama20h; Clo19].

Cloud customers might "lease" the owned cloud resources to be used by their customers, or the cloud end-users. They are then responsible to manage their cloud resources to achieve the objectives set in the SLA agreed with the cloud end-user that might not be covered by the SLA from the CSP, e.g., data availability or security [JS15].

- **Cloud end-user**: Cloud end-users are entities that require certain access to the cloud resources provided by the cloud customer to do its job, e.g., applications, services, or persons. It demands the level of services to be fulfilled according to the agreed SLA with the cloud customers [JS15].

CloudRAID for Business as a cloud storage broker entity is responsible to manage the relationship between object storage services offered by multiple CSPs and companies and their employees as CfB customers and users for a secure EFSS system. Its main objective of the cloud resource management process is to manage the security and compliance of the cloud resources across multiple CSPs for authorized CfB stakeholders due to the shared responsibility model implemented by the CSP [Ama20h; Clo19].

CfB is required to keep track of the global information state of the owned cloud resources and its latest state across multiple CSPs as part of the shared responsibility model [Lia+17b]. It also needs to orchestrate necessary cloud resources for authorized CfB stakeholders, including the buckets where the files are stored, the CSP credential(s) to access the services, and the configuration of the resources to determines who has what kind of access to the buckets and its stored files [TJA10]. Any accidents that happened due to configuration mistakes on the cloud resources will be CfB's faults and responsibility while the CSPs will not be accountable for it. For example, if confidential files are publicly accessible to anyone on the Internet it could cause massive data leak incidents.

CfB also needs to provide access to the cloud resources for the CfB stakeholders as the cloud end-users following their roles and requirements in the CfB environment:

- **CfB customer**: CfB is responsible to manage confidential files of the companies as the CfB customers are stored on the cloud to be always secure and available. Access to the company's files is only granted through the CfB using the file's signed URL to its authorized employees and CfB users outside of the company's domain. CfB employees, other CfB customers, and anonymous Internet users should not be allowed to access the company's files under all circumstances.

- **CfB user**: Authorized company's employees as CfB users must only be able to temporarily access the files stored on the cloud using the file's signed URL generated by the CfB. Non-authorized CfB users should be unable to reuse the file's signed URL that has been used previously by authorized CfB users to access the files.

- **CfB administrator**: CfB administrator is responsible to provide necessary cloud storage resources for the CfB stakeholders by creating or deleting the cloud storage resources and correctly configure the cloud resources to ensure that the authorized CfB stakeholders have access to their cloud storage resources.

- **CfB developer**: CfB developer requires full access to the specific developer bucket in each CSP only.

- **CfB security analyst**: CfB security analyst is responsible to assess the configuration of cloud storage resources across various CSPs ensuring that the resources are correctly configured following the requirements of CfB stakeholders.

However, several challenges of cloud resource and access control management in cloud storage brokerage and multi-cloud storage need to be solved by CfB to provide a comprehensive secure cloud storage service to its customers.

Each CSP utilizes various hardware and software infrastructure to provide its services for its customers where it is not required to comply with cloud computing standards available on the market. This causes the heterogeneity of cloud computing as different CSPs could have various custom mechanisms and implementations of the same cloud services, including its API and data model [PLS15; TJA10]. Meanwhile, the CSP lacks cloud interoperability and cross-collaboration functionalities, which would allow cloud customers to manage owned cloud resources and utilize cloud services in multiple CSPs using a single CSP management platform [TCD20].

CfB is then responsible to manage the cloud resources across multiple CSPs on its own by collecting and processing the precise global information state of cloud storage resources across different CSPs and provisioning necessary resources to fulfill the requirements of CfB stakeholders explained in Chapter 2.3.1 [Lia+17b]. It is also responsible to ensure that the owned cloud resources follow cloud security best practices and standards available, such as Center for Internet Security's benchmarks[35] or German Federal Office for Information Security's cloud computing compliance criteria catalog (BSI C5)[36].

To manage the cloud resources across multiple CSPs, it needs to solve the heterogeneity of the CSPs in terms of API, data model, and service implementations while there is a lack of cloud interoperability and cross-collaboration for cloud resource management process between CSPs [RR18]. Meanwhile, CfB is required to have the necessary cloud management knowledge and skills since the cloud resource management process are prone to error, which could cause unwanted cyber incidents, such as confidential data leak or identity theft [Clo19; Lia+17b]. This results in the cloud resource management complexity to be increasing with the number of CSPs used by the CfB.

Although CfB could manage the cloud resources in each CSP using the available management portal, the CSP management portal might not provide a global outlook of the cloud resources owned across different services or even regions, which could create limited visibility of cloud resources and their usage in multiple CSPs [Clo19; VB18]. CfB also needs to manually access each cloud service and manage the cloud resources in each CSP where it might be prone to inefficient resource management and misconfigured cloud resources [Clo19; FTA16; Tor+18f]. Certain critical CfB's processes, such as CfB customers on-boarding and resource monitoring, would take CfB a lot of time if done manually using multiple CSP management portals.

There are several multi-cloud APIs and services able to provide cloud interoperability and multi-cloud orchestration that can be used by CfB to manage the resources in the multi-cloud scenario, such as jclouds[37], Terraform[38] or Lib-

[35] https://www.cisecurity.org/benchmark

[36] https://bit.ly/3y0cNqf

[37] https://jclouds.apache.org/

[38] https://www.terraform.io/

clouds[39]. However, the multi-cloud APIs and services do not provide full CSP native functionality for all cloud services, e.g., Identity and Access Management (IAM) cloud services or bucket configuration. It is also capable only of limited resource provisioning process where other cloud resource management processes, such as resource monitoring and resource discovery, require separate multi-cloud services specifically for a particular cloud resource management process. Despite the availability of multi-cloud APIs and services, CfB still needs to provide an abstraction layer and a unifying environment to achieve multi-cloud resource management necessary for enterprise cloud storage solution [VB18].

CfB provides system-level and limited cloud-level security and access control for the company's confidential files through encryption and erasure methods. The attacker needs to collect a sufficient number of encrypted file chunks from multiple CSPs to recover the CfB user's encrypted file and without the encryption key, which is stored encrypted in CfB, it still could not be decrypted. But it still does not guarantee that the files stored in the cloud to be only accessible by its authorized CfB users. It then needs to securely configure the cloud resources on multiple CSPs while solving the challenges of enterprise file lifecycle management.

With the virtualization of the CSP's infrastructure that allows multi-tenancy in cloud computing, CSP's cloud resources owned by CfB share the same physical hardware with a similar data-storage mechanism as other CSP customers [MS14]. This might pose the threats of side-channel attack and unauthorized information flow where data could be accessed by other CSP customers [Alm+11]. CfB customer's confidential files are also stored in the same CSP's resources owned by CfB that could raise the risk of cross-client data leakage as CfB customers could potentially access other company's confidential files [Fac+13].

Insider threat is a big issue that needs to be handled by the CfB to ensure the company's confidential files are not accessed by unauthorized entities. According to Ponemon Institute's 2020 Cost of Insider Threats Global Report, an insider theft incident could cost each organization an average of $755,760 [Pon20]. The insider threat is mainly caused by over-privileged access given to unauthorized entities with malicious intent or low awareness and the theft of privileged entity's credentials. Unauthorized company employees and the CfB employees could try to directly access the company's confidential files stored in multiple clouds using the file's generated signed URLs or the CSP credentials. Thus, CfB needs to

[39] https://libcloud.apache.org/

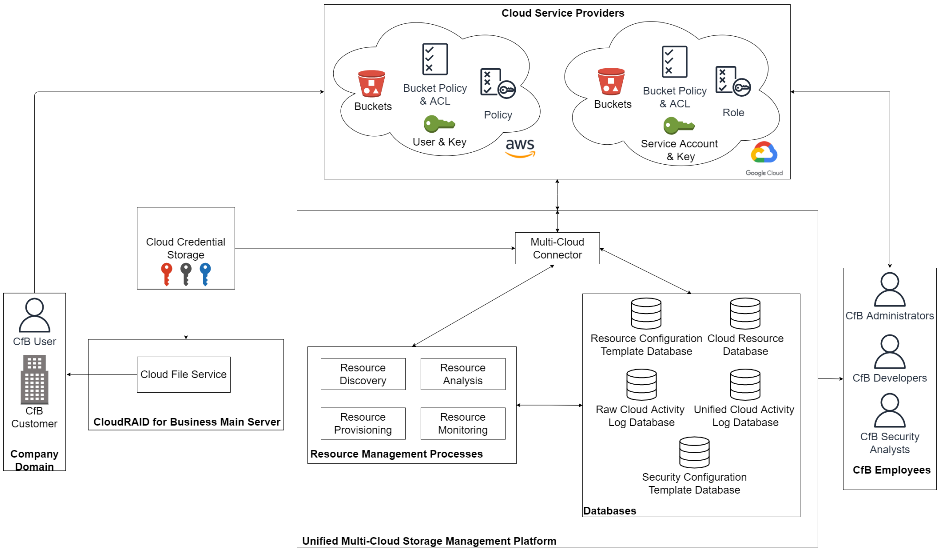


Figure 5.1: Overview of unified multi-cloud storage resource management framework for CloudRAID for Business implemented for Amazon Web Services and Google Cloud Platform [Suk+20]

delegate necessary access for authorized CfB stakeholders to the cloud resources across multiple CSPs without giving away the CSP root credentials owned by CfB. This is to avoid credential theft where unauthorized users can get the CSP root credentials and use them for malicious activities, such as intentional company's confidential files deletion to disrupt the company's activities.

5.4 Unified Multi-Cloud Storage Resource Management Framework

A unified multi-cloud storage resource management framework is proposed for CloudRAID for Business to securely manage cloud storage resources across multiple CSPs for the CfB stakeholders and ensure secure enterprise file synchronization and share system, as can be seen in Figure 5.1. The framework consists of a unified cloud storage resource model, a unified multi-cloud storage resource management platform, and a set of instructions and guidelines implemented in the unified platform and the CfB system.

5.4.1 Unified Cloud Storage Resource Model

Unified cloud storage resource model is proposed to solve the challenges of managing the information of cloud storage resources with different data models from each CSP faced by the CfB. It combines the data model of cloud storage resources and cloud access control available from each CSP's API perspective. The cloud access control utilized by the CSPs follows the role-based access control (RBAC) model to grant entities or cloud services access with defined permissions based on the assigned role to the cloud resources [Ama21h].

The unified model could be used to store and manage the global information state of cloud storage resources across multiple CSPs in a single format. The state of the cloud storage resource could be analyzed to discover the relationship between the cloud storage resources and the entities that have access to it.

It is currently implemented to manage cloud storage resources available in AWS Simple Storage Service (S3), AWS Identity and Access Management (IAM)[40], GCP Storage, GCP IAM[41], and GCP Cloud Resource Manager (CRM)[42]. It could be extended for Object Storage and IAM services from other CSPs employing the RBAC model for cloud storage resources. The unified cloud storage resource model consists of nine entities, as can be seen in Figure 5.2.

- **Object**: Object is the logical abstraction of the data or an object stored in the Bucket. It represents Object in AWS[43] and GCP[44].
- **Bucket**: Bucket is a logical abstraction of object storage container where the Objects are stored in the CSP. It represents Bucket both in AWS[45] and GCP[46].
- **Account**: Account is the identity of an entity created in the IAM service to interact with the CSP, including cloud resources and services. It consists of User[47] in AWS and Service Account[48] in GCP.

[40] https://aws.amazon.com/iam/
[41] https://cloud.google.com/iam/
[42] https://cloud.google.com/resource-manager/
[43] https://docs.aws.amazon.com/en_pv/AmazonS3/latest/dev/UsingObjects.html
[44] https://cloud.google.com/storage/docs/json_api/v1/objects
[45] https://docs.aws.amazon.com/en_pv/AmazonS3/latest/dev/UsingBucket.html
[46] https://cloud.google.com/storage/docs/json_api/v1/buckets
[47] https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html
[48] https://cloud.google.com/iam/docs/service-accounts

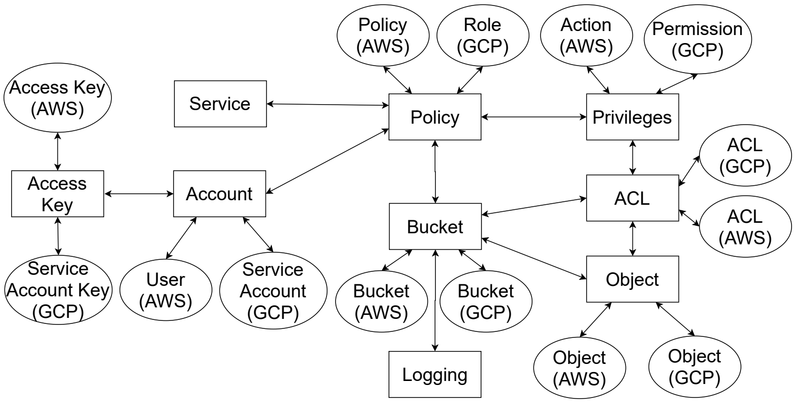


Figure 5.2: Unified cloud storage resource model and its implementation on top of Object Storage and Identity and Access Management services in AWS and GCP [Suk+20]

- **Service**: Service represents the identity of a cloud service of the CSP.
- **Privilege**: Privilege is the possible action/permission in the cloud services and resources. It consists of Action[49] in AWS and Permission in GCP[50].
- **Policy**: Policy is a set of Privileges and its state (allow/deny) that regulates cloud-level access control between the entity in the CSP and the cloud resources or services. Policy is represented as Policy[Ama21e] in AWS and Role[51] in GCP. In general, there are two types of how Policy can be assigned:

    1. *IAM-level Policy*: Policy is attached to IAM entities or cloud service that allows or denies access to cloud services and resources. In AWS, Policy can be assigned directly to User, Group, or Role [Ama21e]. In GCP, Role can be assigned to Service Account, Google account and group, G Suite domain, and cloud identity domain [Goo21b].
    2. *Resource-level Policy*: Policy is assigned to a cloud resource that determines who is authorized to access the resource. In AWS, Policy

[49] https://docs.aws.amazon.com/en_pv/IAM/latest/UserGuide/reference_policies_actions-resources-contextkeys.html

[50] https://cloud.google.com/storage/docs/access-control/using-iam-permissions

[51] https://cloud.google.com/iam/docs/understanding-roles

can be assigned to Bucket by specifying the IAM entities or AWS service accessing it [Ama21e]. In GCP, a Role can be assigned to Service Account, Google account and group, G Suite domain, and cloud identity domain to the Bucket[52].

- **Access Control List (ACL)**: ACL is a list of access permission to Buckets and/or its Object that defines the entity and its type of access. It is a legacy access control mechanism that predates IAM-level access control via Policy for Object Storage services. It represents ACL both in AWS[53] and GCP[54]. GCP introduces uniform bucket-level access where the cloud customers could optionally disable Bucket's ACL and assign Roles to the entity for access to the Bucket and its Objects [55].
- **Access Key**: Access Key is the credential of Account used for authentication and allowing programmatic calls to the CSP consisting of the access key ID and secret key, which is similar to username and password combination. The privileges of Access Key follow the Policy granted to the Account to only access its authorized cloud resources. It represents Access Key[56] in AWS and Service Account Key[57] in GCP.
- **Logging**: Logging is the representation of Bucket's logging configuration[58][59] where all activities on the monitored Bucket are logged and delivered to the target Bucket. The detailed usage of Logging will be explained further in Chapter 6.

5.4.2 Unified Multi-Cloud Storage Management Platform

The unified multi-cloud storage resource management platform provides centralized multi-cloud management and holistic visibility for the cloud storage resources in various CSPs for CfB. It follows centralized state information collection centric technique [Asl+17] and cloud brokerage approach [HLV16; TCB14]

[52] https://cloud.google.com/storage/docs/access-control/iam-roles
[53] https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html
[54] https://cloud.google.com/storage/docs/access-control/lists
[55] https://cloud.google.com/storage/docs/uniform-bucket-level-access
[56] https://docs.aws.amazon.com/en_pv/IAM/latest/UserGuide/id_credentials_access-keys.html
[57] https://cloud.google.com/iam/docs/creating-managing-service-account-keys
[58] https://docs.aws.amazon.com/AmazonS3/latest/dev/ServerLogs.html
[59] https://cloud.google.com/storage/docs/access-logs

to collect and pre-processes the information of the cloud storage resources and manage the relationship between CfB stakeholders and multiple CSPs.

The unified platform aims to reduce the decision complexity taken by the CfB to manage the cloud storage resources and their configurations access across multiple CSPs for CfB stakeholders [HLV16]. It is utilized by the CfB employees to execute necessary automated cloud resource management processes in a single platform instead of utilizing each CSP's management platform and API individually. It implements the unified cloud storage resource model to simplify the information asymmetry of cloud storage resources for managing and provisioning cloud storage resources and their access control for the CfB stakeholders.

The unified multi-cloud storage resource management framework incorporates cloud credential store, multi-cloud-connector, and several databases as it focuses on four main resource management processes: resource discovery, resource orchestration, resource assessment, and resource monitoring.

Multi-Cloud Connector

The multi-cloud connector is the gateway between the unified multi-cloud storage management platform with Amazon Web Services and Google Cloud Platform. All cloud resource commands made by the unified platform, such as Bucket creation or assigning Policy to Service Account, are translated into CSP's native API commands by the multi-cloud connector.

It utilizes an abstraction layer built on top of CSP's native APIs of AWS S3, AWS IAM, GC Storage, GCP IAM, and GC CRM services to ensure that the platform can access the full functionalities of the Object Storage and IAM services. The abstraction layer is implemented by finding the similar functionalities and data model from the API's perspective needed based on the unified cloud storage resource model explained previously. Currently, the abstraction layer is based on the Java SDK of the Object Storage and IAM services from AWS and GCP.

It also consists of **Log Collector application** that automatically checks and downloads new log files generated in the **Log Sink Buckets** across various CSPs, such as Logstash[60] or Fluentd[61]. The usage of Log Collector application will be explained in later part and Chapter 6.

[60] https://www.elastic.co/logstash
[61] https://www.fluentd.org/

Cloud Credential Storage

Cloud credential storage is an entity responsible to securely store the Access Key of each CSP for each CfB stakeholder. The Access Keys are generated by the unified platform from the Accounts with different Policies following the CfB stakeholder's access requirements in the multi-cloud storage environment, which will be explained in further subsection. Vault[62] or Thycotic Secret Server[63] are several services that could be used as secure cloud credential storage.

Cloud credential storage could only be accessed by a unified multi-cloud storage resource management platform and CfB main server to retrieve the necessary Access Keys. When the unified platform issues a request to a CSP on behalf of the CfB stakeholders, the multi-cloud connector first requests the required Access Key to cloud credential storage before it is used by the unified platform to send the request to the respective CSP. Meanwhile, the CfB main server's cloud file service will request the Access Keys of multiple CSPs owned by the CfB customers to cloud credential storage to generate the signed URLs for authorized CfB users, which are used to access the files stored in the cloud.

Resource Discovery Process

Resource discovery is the process to detect and register all available created resources for each service in the CSPs. The unified multi-cloud storage management platform provides a resource discovery process by automatically gathering the information of all cloud storage resources and their configurations across multiple CSPs. It helps CfB to provide holistic visibility of the cloud storage resources across multiple CSPs in a single interface instead of manually checking each cloud service using the CSP management dashboard. When the unified platform runs the resource discovery process for the first time, it does not require prior knowledge of cloud storage resources created previously by the CfB.

Figure 5.3 shows an overview of the resource discovery process in the unified multi-cloud storage management platform. The multi-cloud connector first sends a request to each CSP service to retrieve the information of all available cloud storage resources and their configurations, e.g., name, type, Policy, and ACL. The information of cloud storage resources that could not be collected during the discovery process due to the limitation of the CSP's API could be added

[62] https://www.vaultproject.io/

[63] https://thycotic.com/products/secret-server/

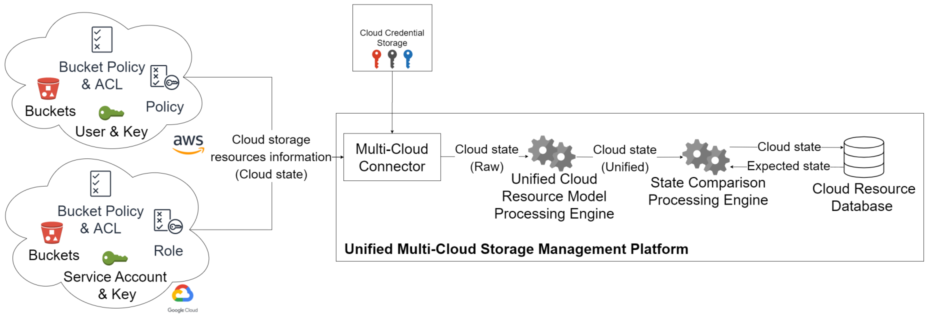


Figure 5.3: Overview of unified multi-cloud storage management platform's resource discovery process [Suk+20]

manually later by the CfB administrator. For example, the secret key of Access Key is only available once it is newly generated.

The cloud storage resource's raw information is then processed by the Unified Cloud Resource Model Processing engine to parse the information with different data models from various CSPs to the proposed unified cloud storage resource model. The processed cloud storage information is then stored in the Cloud Resource database. An example of the AWS S3 bucket's information in the unified format:

```
{
   "name":"exampleBucket",
   "type":"Bucket",
   "csp":"AWS",
   "creationDate":"2019-01-02T21:27:04.000+0000",
   "location":"eu-central-1:Frankfurt",
   "bucketConfiguration":{
      "logging":{
         "enabled":false
      },
      "accessors":[
         {
            "name":"TestUser",
            "effect":"Allow",
            "type":"ACL",
            "entity":"User Grantee",
            "privileges":[
                    "s3:ListBucket",
                    "s3:PutObject",
                    "s3:DeleteObject"
                ]
            }
        ]
    },
    "deleted":false
}
```

The state transition model is incorporated into the resource discovery process to track the changes happening to the cloud storage resource [Tor+19c]. When the resource discovery process runs for the first time, the cloud storage resource information in a unified format stored in the Cloud Resource database is regarded as the **expected state**. After the initial resource discovery process, the retrieved information of cloud storage resource is then regarded as the **cloud state**. These states are then compared using the State Comparison Processing engine. If the states are different, CfB could decide whether to store the cloud state in the Cloud Resource database as the expected state or retain the expected state by reversing any changes in the cloud storage resources across multiple CSPs.

CfB could then associate the information of cloud storage resources and their configurations with the information of CfB stakeholders. It also runs the resource discovery process periodically in the background to monitor any changes in the cloud storage resources and be alerted if there are changes to the resources not executed by the unified platform, which will be explained later.

Resource Orchestration Process

Resource orchestration is the process of allocating cloud storage resources across multiple CSPs. It aims to reduce the possibility of misconfiguration and unwanted changes in cloud storage resources across multiple CSPs due to human error while orchestrating cloud storage resources manually using the CSP's management dashboard. CfB could then automatically create, delete, and modify the cloud storage resources and their configurations across multiple CSPs using the unified multi-cloud storage resource platform.

Figure 5.4 shows an overview of the resource orchestration process in the unified multi-cloud storage management platform. The resource orchestration process utilizes **cloud storage resource specification** that specifies the neces-

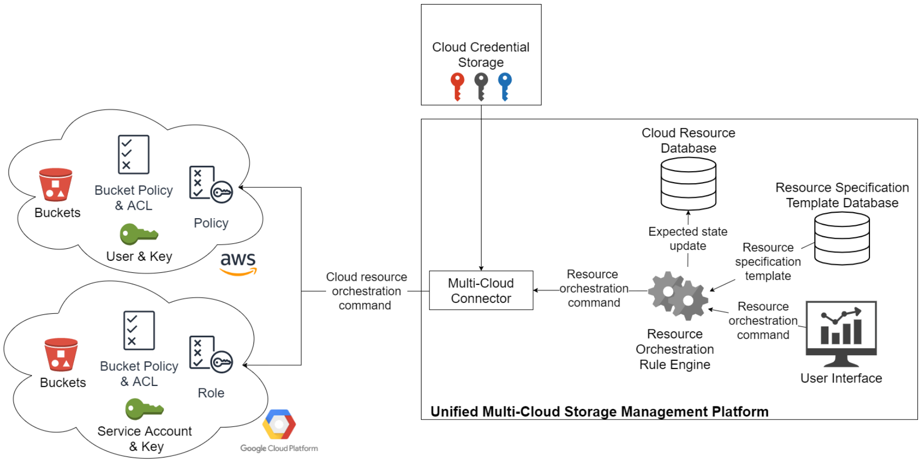

Cloud Credential Storage
Buckets
Bucket Policy & ACL
Policy
User & Key
aws
Buckets
Bucket Policy & ACL
Role
Service Account & Key
Google Cloud Platform
Cloud resource orchestration command
Multi-Cloud Connector
Resource orchestration command
Cloud Resource Database
Expected state update
Resource Specification Template Database
Resource specification template
Resource orchestration command
Resource Orchestration Rule Engine
User Interface
Unified Multi-Cloud Storage Management Platform

Figure 5.4: Overview of unified multi-cloud storage management platform's resource orchestration process [Suk+20]

sary cloud storage resources and its configurations and access for authorized entities following the proposed unified cloud storage resource model. CfB could orchestrate the cloud storage resources by using the cloud storage resource specification template stored in the **Resource Specification Template** database or execute other resource orchestration commands through the user interface from the CfB administration platform. The resource orchestration command and resource specification template are then processed by **Resource Orchestration Rule** engine to consolidate the resource orchestration command. The resource orchestration command is then translated by the multi-cloud connector to the specific CSP's API commands, where it is then executed to the respective CSP. Finally, it updates the expected state with the information of created, deleted, or modified cloud storage resources in the Cloud Resource database.

Resource Monitoring Process

Resource monitoring is the process of monitoring the events happening on the cloud storage resources across multiple CSPs. The aim of the resource monitoring process is to have full oversight of the activities and the usage on the CfB's multi-cloud storage environment and detect any suspicious or malicious events on the cloud storage resources. This is due to the information provided by the CfB system and its unified multi-cloud storage resource management platform might not be enough to give full oversight of activities in CfB's multi-cloud storage environment.

The resource monitoring process then utilizes the log files generated by AWS CloudTrail[64] and Google Cloud Logging[65] services, or **cloud activity log files**, to monitor the activities happening in cloud storage resources across multiple CSPs. The cloud activity log files, which are generated by using cloud logging and monitoring services, contain detailed information of the events happening in the cloud resources and services used and owned in the CSP environment.

However, there are several challenges in processing cloud activity log files from different CSPs. In general, the cloud activity log entries could only be viewed and processed using the cloud logging and monitoring services provided by the CSP where the log files could be automatically deleted after a certain period [Ama21g; Goo20f]. Each CSP also has its log format structure and information quality for the cloud activity log file. For example, AWS CloudTrail log files provide more information and better data structure consistency compared to GCP Logging log files. CfB is then responsible to actively retrieve, store, and process the cloud activity log files from multiple CSPs to gain necessary information about the events happening on the cloud storage resources.

The resource monitoring process follows the data warehouse method [Hu+14], which consists of extraction, transformation, and loading steps, to transform cloud activity log files in JSON format as semi-structured data to the structured data format, such as CSV. **Unified cloud activity log format** is proposed to normalize different log formats and information quality from various CSPs to a single format as can be seen in Table 5.1. The necessary information needed from the available cloud activity log fields is first selected where the values, which are in different formats or could contain information for multiple log fields, are then normalized. Finally, the information from cloud activity log files from multiple CSPs is combined to give an overview of the events happening to the cloud storage resources in multiple CSPs.

Figure 5.5 shows the overview of the resource monitoring process. First, cloud logging and monitoring services in the CSPs are configured to record the events happening in the CSP environments where the cloud activity log files are delivered into a specific **Cloud Activity Log Sink Bucket** in each CSP that

[64] https://aws.amazon.com/cloudtrail/
[65] https://cloud.google.com/logging

Unified Cloud Activity Log	**AWS CloudTrail**	**GCP Logging**
eventId	eventID	-
timestamp	eventTime	timestamp
csp	"AWS"	"GCP"
service	eventSource	protoPayload.serviceName
resourceName	requestParameters	protoPayload.request Parameter or protoPayload. resourceName
resourceType	requestParameters	protoPayload.request
resourceLocation	awsRegion	resource.label.location
method	eventName	protoPayload.methodName
ipAddress	sourceIPAddress	protoPayload.request Metadata.callerIP
userAgent	userAgent	protoPayload.request Metadata.caller SuppliedAgent
responseCode	errorCode	protoPayload.status.code
responseMessage	errorMessage	protoPayload.status.message
requesterCredential	userIdentity	protoPayload.authentication Info.principalEmail

Table 5.1: Unified cloud activity log format and the parsing from AWS CloudTrail and GCP Logging [Suk+20]

provides inexpensive and long-term storage for the log files. Depending on the CSP, the cloud activity log file could be delivered to the Bucket every 5 minutes up to one hour [Ama21c; Goo20d]. The multi-cloud connector through its Log Collector application then routinely checks and downloads the cloud activity log files from the Log Sink Buckets in multiple CSPs.

After the cloud activity log files have been downloaded, it is then stored into **Raw Cloud Log Activity database** while it is processed by the **Unified Cloud Activity Log Parser** to parse cloud activity log files into the proposed unified log format and store it in **Unified Cloud Log Activity database**. The raw and unified cloud activity log entries are then pushed into the **Log Correlation engine** that will be correlated with the log entries stored in the **CfB System Log**

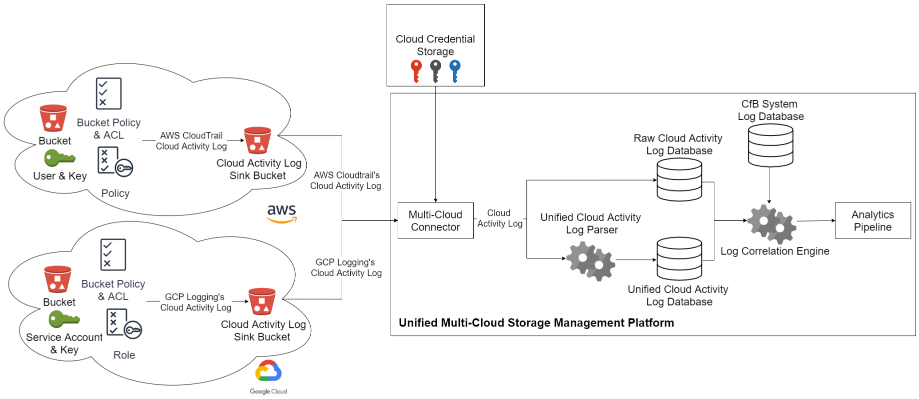

Cloud Credential Storage
Bucket Policy & ACL
Bucket
User & Key
Policy
AWS CloudTrail Cloud Activity Log
Cloud Activity Log Sink Bucket
aws
AWS Cloudtrail's Cloud Activity Log
Bucket Policy & ACL
Bucket
Service Account & Key
Role
GCP Logging's Cloud Activity Log
Cloud Activity Log Sink Bucket
Google Cloud
GCP Logging's Cloud Activity Log
Multi-Cloud Connector
Cloud Activity Log
Unified Cloud Activity Log Parser
Raw Cloud Activity Log Database
CfB System Log Database
Unified Cloud Activity Log Database
Log Correlation Engine
Analytics Pipeline
Unified Multi-Cloud Storage Management Platform

Figure 5.5: Overview of unified multi-cloud storage management platform's resource monitoring process [Suk+20]

database, which record the activities happening in the CfB system, including the activities on the unified platform. The Log Correlation engine will analyze if the cloud activity log entries have the corresponding CfB system log entries and detect any suspicious or malicious events. Finally, the insights are then forwarded to Analytics Pipeline for further processing.

Resource Assessment Process

Resource assessment is the process of evaluating the cloud storage resources and its configurations against the cloud storage resource specifications set by the CfB. The goal is to ensure the cloud storage resources are correctly configured and only be accessed by its authorized entities and detect any unauthorized changes happening to the resources.

Figure 5.6 shows the overview of the resource assessment process. The raw information of cloud storage resources and their configurations, or cloud state, is first retrieved periodically and parsed with the Unified Cloud Resource Model Processing engine to follow the proposed unified cloud storage resource model. The **Resource Assessment engine** compares the cloud state in the unified model, or unified cloud state, with the expected state stored in the Cloud Resource database to detect if there are any unauthorized modifications on the cloud storage resources.

The Resource Assessment engine also evaluates the unified cloud state and

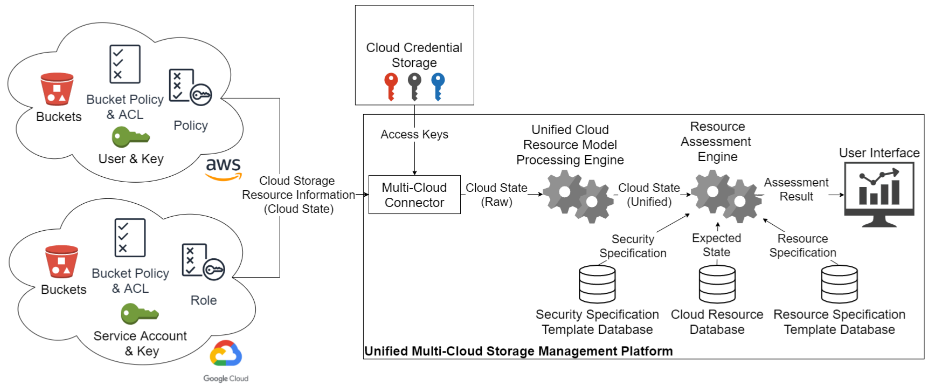


Figure 5.6: Overview of unified multi-cloud storage management platform's resource assessment process [Suk+20]

the expected state against the security specifications and the resource specifications that are fetched from the **Security Specification Template database** and the Resource Specification Template database, respectively. The security specification template could be derived from cloud security best practices and standards available on the market.

If there are unauthorized modifications to the cloud state or the cloud storage resource configurations do not comply with the security and resource specification templates, the Resource Assessment engine will generate an assessment result that summarizes the alerts of the non-compliant cloud storage resources that violate the security and resource specification templates and the recommended actions to be taken to address the alerts. CfB administrators and CfB security reviewers could take necessary actions to improve the cloud storage resources' configurations to ensure that the resources are secure and can only be accessed by authorized cloud end-users.

5.4.3 Instructions and Guidelines

The set of instructions and guidelines are implemented based on the cloud security standards and best practices to ensure the CfB's cloud storage resources are securely configured. It is implemented into the various cloud resource management processes in the unified multi-cloud storage resource management platform to assign necessary cloud storage resources with correct configurations and access privileges to CfB stakeholders. It also dictates how the CfB system should provide secure access to the files stored in multiple CSPs only for authorized CfB users.

Cloud Storage Resources and Access Control for CfB Stakeholders

CfB needs to provide necessary cloud storage resources with correct configurations to ensure only the authorized CfB stakeholders could access the cloud storage resources with limited actions following their roles in the CfB system. The instructions and guidelines are then implemented in the unified multi-cloud storage management platform's resource orchestration process as a cloud storage resource specification template as follows:

1. All Buckets owned by the CfB are configured to deny access outside of the authorized CfB stakeholders by correctly assigning the ACL and Policy to the Buckets.

2. A randomized string is added to Bucket's name during the Bucket creation process since the Bucket needs to be unique in the entire CSP's bucket namespace. It also helps to avoid brute-force enumeration attacks on the cloud bucket where the attacker enumerates the list of bucket names using the supplied wordlist and checks if the bucket exists and is publicly accessible.

3. The Access Key of all Accounts managed by CfB should be stored centrally in Cloud Credential Storage.

4. The Access Key of the Account should be regularly rotated, for example, every 90 days, to reduce the chance of the Access Key associated with expired or compromised Account or stolen Access Key to be used to unauthorizedly access the cloud storage resources. The rotated Access Keys could then no longer be used to access the CSPs.

5. The cloud storage resources for the CfB stakeholders are orchestrated based on the least privilege principle and privilege separation concept [PFH03] and cloud security best practices and standards. This is to ensure the CfB stakeholders only have limited access to the authorized cloud storage resources following their roles in the CfB environment. This is

to avoid insider threats from CfB employees or over-privileged access that could unauthorizedly access the CfB customer's files on the cloud or interrupt CfB's service continuity.

- **CfB Customer**: CfB is responsible to securely store the company's confidential files as multiple encrypted chunks across multiple CSPs and delegate access to the files stored on the cloud only to authorized CfB customers and users. CfB also needs to monitor and report to the company for any activities happening to the company's files stored in the cloud.

  When a new company registers for CfB, for each CSP, CfB creates a **File Bucket** for storing the encrypted data chunks and **File Account** for accessing the File Bucket, i.e., upload, download, delete, and list the encrypted data chunks in File Bucket. The Policy is then assigned to the company's File Account and File Bucket to ensure only the corresponding company's File Account is allowed to access its File Bucket and avoid data leakage across other CSP tenants and CfB customers. In AWS, a custom Bucket Policy is assigned to the company's File Bucket that only allows File Account to access its corresponding File Bucket. Meanwhile, in GCP the company's File Account is assigned Storage Object Admin role to its corresponding File Bucket.

  All File Buckets of the CfB customers in each CSP are configured to record the events happening in the monitored File Bucket and its the encrypted data chunks as storage access log files and store the log files in a centralized **Log Bucket**. A **Log Account** is created that is capable of only listing and downloading storage log files from Log Bucket. In AWS, a custom Bucket Policy is assigned to the Log Bucket that only allows Log Account to access its corresponding Log Bucket. Meanwhile, the Log Account in GCP is assigned Storage Object Viewer role to its corresponding Log Bucket. Finally, the Access Keys of File Accounts and Log Accounts are then stored securely in Cloud Credential Storage.

- **CfB Employees**: Three types of CfB employees would require access to the CfB's multi-cloud storage environment via the unified multi-cloud storage resource management platform to ensure the continuity of CfB services. For each CfB employee's role, the CfB employee's Account is assigned to a Group attached to the Policies in the AWS. As for GCP, the appropriate Role is assigned to the CfB employee's Account in the GCP IAM service. No CfB employee can access the company's confidential files in any circumstance.

- *CfB Administrator*: CfB administrator is responsible to manage cloud storage resources and its configurations and assign necessary access to authorized CfB stakeholders. **Administrator Account** is created for each CfB administrator and assigned custom Policy and Role to create, delete, and modify the cloud storage resources in AWS and GCP, respectively.
- *CfB Developer*: A **Developer Bucket** is created where the CfB developers could use it to develop new features for the CfB system. Each CfB developer is given a **Developer Account** where the Policy is applied to the Developer Bucket to ensure only the CfB developer's Account could fully access (upload, download, list, delete) the Developer Bucket. In AWS, a custom Bucket Policy is assigned to the Developer Bucket that only allows the Developer Account to access its corresponding Developer Bucket. Meanwhile, in GCP the Developer Account is assigned Storage Object Admin role to its corresponding Developer Bucket [Goo21a].
- *CfB Security Auditor*: CfB Security Auditor is responsible to assess the cloud storage resources and their configurations ensuring that they are compliant with the cloud storage resource specification template. **Security Auditor Account** is created for each CfB security auditor and assigned Security Audit Policy[66] in AWS and custom Role in GCP to list the cloud storage resources and their configurations.

The cloud storage resource specification template could also be used for the resource assessment process to verify if the cloud storage resources are correctly and securely configured for the particular CfB stakeholders. The resource orchestration process could then modify the misconfigured cloud storage resources

[66] https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_job-functions.html#jf_security-auditor

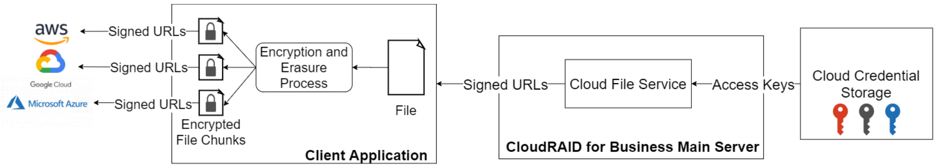


Figure 5.7: Example of Cloud File service's implementation of signed URL method following the unified multi-cloud storage resource management framework

following the cloud storage resource specification template to fix the insecure and incorrect cloud storage resource configuration.

Cloud File Service

CfB customers and CfB users do not have direct access to the company's confidential files stored in the cloud as both do not possess the necessary Access Keys from various CSPs. Instead, the access to the files is managed by the CloudRAID for Business' Cloud File service using the signed URL method as explained in Chapter 2.2.5. The signed URL method allows CfB to delegate temporary access to the files stored in multiple CSPs to authorized CfB users without giving away the CSP root credentials. With each company having a File Account credential in each CSP, the signed URL generation process by the Cloud File service is affected to provide secure access to the files stored multiple CSPs within the company's domain.

The signed URLs are generated using the Access Keys of the File Accounts stored on the Cloud Credential Storage. CfB's Cloud File service first requests the necessary Access Keys of the company that owns the requested file to Cloud Credential Storage. It then generates the signed URLs to the authorized CfB users using the Access Keys by signing the file operation request and appending the signature in the URL. Finally, the CfB users could upload, download, or delete the file on the cloud by executing the received signed URLs. Using this method, CfB could also provide secure inter-company file sharing functionality by distributing the signed URLs generated by the corresponding company's File Account to the CfB users outside of the company's domain, assuming the user is authorized to access the file.

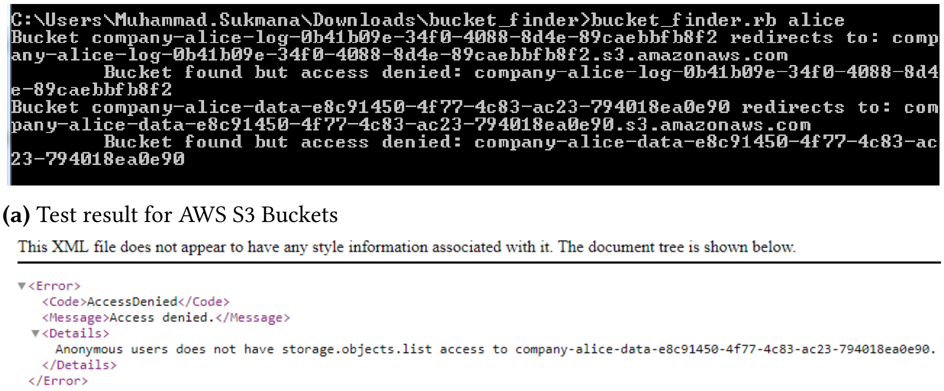
By default, the signed URL could be valid up to 7 days if the validity period is not specified when it is generated [Ama20g; Goo20e]. This means anyone having access to the valid signed URL could unauthorizedly access the files on the cloud, such as replacing the original file with malicious file or downloading confidential file. To protect signed URL from being shared by the CfB user to be used by unauthorized user, the signed URL's validity period is limited to minimum time needed to execute the request, such as 1 minute. Using this method, once the CfB user has finished executing the request, the signed URLs are no longer valid for other unauthorized users. If unauthorized user modifies the signed URL to gain access to the file by changing its validity period or the file's name, the request is then denied by the CSP as it requires valid signature generated by the corresponding Access Key of company's File Account. Figure 5.7 shows an example of Signed URL implementation for file upload operation.

5.5 Evaluation

In this Section, the security of the proposed unified multi-cloud storage resource management framework is evaluated for the CfB system. The company's confidential files stored in multiple CSPs are evaluated whether it is secure against cross-client data leakage, insider threat, and unauthorized data access. For this evaluation, four attackers are trying to steal the files owned by Alice company as new CfB customers: malicious Bob company, malicious CfB employees, malicious Alice company's employees, and anonymous Internet users.

Malicious Bob company tried to download a file from Alice company's File Bucket in AWS S3 using Bob company's File Account. Malicious CfB employees also tried to access Alice company's files by generating unauthorized Access Keys from their Accounts. Since Bob company's File Account and CfB employee's Accounts are not listed in the Alice company's File Bucket Policy, the unauthorized requests are denied and logged into the CfB's centralized Log Bucket.

Alice company's malicious employee then tried to access unauthorized files by capturing the file's signed URLs executed by the authorized employees. Since the malicious employee requested the signed URLs past their validation time, the file access request is then denied by the CSPs. Even if the malicious employee tried to execute the signed URL with modified validation time or the object name on the signed URL, it will still return denied file access responses. This is due to the signatures on the signed URLs do not match with the signatures calculated by the CSPs based on the file access request's information and the access keys of the requester, which are owned by the Alice company's File Account.



```
C:\Users\Muhammad.Sukmana\Downloads\bucket_finder>bucket_finder.rb alice
Bucket company-alice-log-0b41b09e-34f0-4088-8d4e-89caebbfb8f2 redirects to: comp
any-alice-log-0b41b09e-34f0-4088-8d4e-89caebbfb8f2.s3.amazonaws.com
        Bucket found but access denied: company-alice-log-0b41b09e-34f0-4088-8d4
e-89caebbfb8f2
Bucket company-alice-data-e8c91450-4f77-4c83-ac23-794018ea0e90 redirects to: com
pany-alice-data-e8c91450-4f77-4c83-ac23-794018ea0e90.s3.amazonaws.com
        Bucket found but access denied: company-alice-data-e8c91450-4f77-4c83-ac
23-794018ea0e90
```

(a) Test result for AWS S3 Buckets

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access denied.</Message>
  ▼<Details>
     Anonymous users does not have storage.objects.list access to company-alice-data-e8c91450-4f77-4c83-ac23-794018ea0e90.
   </Details>
 </Error>
```

(b) Test result for GCP Storage Bucket

Figure 5.8: Bucket public accessible test result for Alice company's File Buckets in AWS S3 and GCP Storage services [Suk+19c]

Finally, anonymous Internet users tried to access the files stored on the Alice company's File Buckets in multiple CSPs by accessing directly from the browser without authentication to the CSP or using bucket wordlist brute-force enumeration tool, such as Bucket Finder[67], to check if the Buckets are publicly accessible. Figure 5.8 shows that Alice company's Bucket and its files could not be accessed without proper authentication and authorization.

The results of the evaluation above show that CfB could secure the cloud storage resources, especially CfB customer's confidential files, from malicious CfB customers, CfB users, and anonymous Internet users. The proposed unified cloud storage resource model helps CfB to provide secure logical storage separation and role-based access control for CfB stakeholders and cloud storage resources across multiple CSPs.

5.6 Discussion

The unified cloud storage resource model helps to normalize various data and cloud access control models of cloud storage resources from Object Storage and IAM services in AWS and GCP as both CSPs employ quite a similar cloud access

[67] https://digi.ninja/projects/bucket_finder.php

control model. CfB utilizes the unified model to easily manage the information of cloud storage resources in a single data model and determine the relationship between the cloud storage resources with authorized CfB stakeholders. This allows CfB to provide secure and automated multi-cloud management processes and role-based access control assignments for CfB stakeholders accessing the authorized cloud storage resources with limited allowed actions.

The unified multi-cloud storage resource management platform provides a secure and automated cloud storage management process for CfB where cloud storage resources could be managed in a single interface instead of using multiple CSP's management dashboards. The unified platform utilizes the abstraction layer built on top of CSP's native APIs to access the full functionality of the IAM and Object Storage cloud services, which are not provided by the multi-cloud APIs and services available on the market. It provides holistic visibility of CfB's multi-cloud storage environment as it regularly collects the latest state of cloud storage resources in a centralized environment using the proposed unified cloud resource model. It could also determine if any unauthorized changes are happening on the cloud storage resources across multiple CSPs and revert the changes to the expected state of cloud storage resources. It also monitors the activities happening in the CfB's multi-cloud storage environment to detect if there are any suspicious or malicious activities happening on the cloud unsanctioned by the unified platform. The file activity monitoring process on the company's File Buckets will be explained further in Chapter 6.

Although CloudRAID for Business utilizes Microsoft Azure as one of the CSPs for its multi-cloud storage environment, the unified multi-cloud storage resource management framework currently does not cover the cloud storage resource management for Microsoft Azure. This is due to the different data model, access control, and API of cloud storage resources in the Blob Storage and Active Directory[68] services are quite different compared to Storage and IAM services in AWS and GCP, as explained previously in Chapter 2.1.1.

Blob Storage service utilizes the concept of the storage account where it could contain multiple containers and an unlimited number of blobs. Two pairs of access keys are automatically generated for each storage account that would allow full access to the storage account. If an unauthorized entity could obtain the storage account's access keys, the access key could be used to uauthorizedly

[68] https://azure.microsoft.com/en-us/services/active-directory/

access company's file chunks and modify storage account's configuration, which could jeopardize the confidentiality of the company's files.

Azure Active Directory service provides User as one of the identities of an entity in the Organization to interact with available Azure services. Each User in the Azure environment could use its username and password to gain access to the Azure Portal, CLI, and API. The usage of username and password could create security vulnerability in the CfB environment since it would give CfB stakeholders unnecessary access to the CSP's management dashboard to access the cloud storage resources and other unauthorized services. This is different than the User in AWS IAM and Service Account in GCP IAM services as both utilize only Access Key that only supports API access to the CSPs.

The Java SDKs provided by Microsoft Azure also do not provide several core full functionalities and complete documentation of the Azure services. This affects several CfB's main operations on the cloud storage resources available in Azure to be done manually using Azure Portal or CLI. For example, to enforce data separation between companies, each CfB customer will have a Storage Account to store its confidential files and log files containing the activities happening on the Storage Account. However, a Storage Account previously could only be created using the Azure Portal, Azure CLI, or the REST API[69]. This would require CfB to manually create the Storage Account for each company and assign necessary access configurations for each company, which is prone to human error and could cause misconfiguration on the cloud resources.

5.7 Conclusion and Future Works

In this Chapter, a unified multi-cloud storage resource management framework is proposed for CloudRAID for Business to solve the challenges of managing its multi-cloud storage environment for various CfB stakeholders across multiple CSPs and ensure secure enterprise file synchronization and share solution. The unified cloud storage resource model tackles different data models of various CSPs to determine and manage the state of cloud storage resources in a single model. The unified multi-cloud storage resource management platform implements the proposed unified model to automatically and centrally discover, create, delete, modify, evaluate, and monitor the cloud storage resources and

[69] https://docs.microsoft.com/en-us/rest/api/storagerp/storage-accounts/create

their configurations centrally across multiple CSPs. The guidelines and instructions are implemented on the CfB's multi-cloud storage environment based on the cloud security best practices and standards, least privilege principle, and privilege separation concept to ensure that the cloud storage resources and CfB stakeholder's access to the cloud are correctly and securely configured.

Other cloud management processes, such as billing and SLA monitoring, could be implemented to the unified framework to provide CfB a centralized and complete cloud management view across various CSPs. The proposed unified framework could be expanded by incorporating other CSPs, such as Microsoft Azure or Openstack, and cloud resource types, e.g., databases, virtual machines, or containers, to provide centralized, automated, and secure cloud resource management for CfB and other multi-cloud environments.

Security chaos engineering technique [Tor+20; Tor+21] could be implemented in the unified multi-cloud storage resource management framework to verify the security and resilience of CfB's cloud storage resources against possible cloud attacks and ensure only authorized CfB stakeholders could access the resources. Security controllable faults, or **chaos**, are injected into the cloud resources to mimic the cloud security attacks where the effect to the cloud resources are then observed to generate the report. Based on the report, CfB could determine if the cloud resources are secure and resilient against possible cloud attacks and improve the configurations of cloud resources, if necessary.

6 Monitoring File Activities in Multi-Cloud Storage Systems

6.1 Introduction

Enterprise file synchronization and sharing systems allow enterprises to store their confidential files on the cloud with high data availability and service reliability. However, several challenges are faced by the companies using the EFSS system to ensure their files are only accessed by their authorized employees. Companies effectively relinquish the physical control of their files to the CSPs once the files are stored on the cloud [Wan+10]. Malicious CSP administrators or even the EFSS systems could unauthorizedly access or modify the files on the cloud without the company's knowledge [KE16].

Files stored in the cloud object storage services could also be publicly accessible if the ACL or the policy of the buckets or the files are misconfigured. It could result in data breach incidents where anonymous Internet users could access the confidential files by requesting the URL of the bucket or the files, such as https://testbucket.s3.amazonaws.com/targetFile [Con+18; Tor+18a; Wan+10]. Insider threat is another issue faced by the companies where unauthorized employees could try accessing the files or sharing the files with unauthorized entities. According to Ponemon Institute's 2020 Cost of Insider Threats Global Report [Pon20], 62% of the insider incidents happened due to employee's negligence, and 23% is caused by insiders with criminal or malicious intents.

CloudRAID for Business is responsible to manage the company's files stored in the cloud on behalf of the companies following the CSP's shared responsibility model [Ama20h] as mentioned in Chapter 5. One of CfB's responsibilities is to monitor and audit the activities of CfB user's files and the cloud storage services [Car+17]. CfB also needs to inform the companies of the file activities happening in the system and its multi-cloud storage environment to ensure only the authorized company's employees could access the company's files on the cloud. It could monitor its multi-cloud storage systems by collecting, processing, and analyzing log files generated by cloud object storage services from multiple CSPs or **cloud storage log files**. The cloud storage log files provide information on the events happening in the cloud objects storage services, such as the requester's information, request types, and response information [Kha+16; Tor+19a].

In this chapter, a multi-cloud file storage monitoring system is proposed for CloudRAID for Business system to monitor activities of the files stored in multiple CSPs. It automatically collects, processes, and correlates cloud storage log files generated by AWS S3, GCP Storage, and Azure Blob with the CfB system log entries. The collected and correlated information is then furtherly analyzed for various use cases, such as detecting suspicious activities and monitoring file activities. The current state of cloud object storage services and their logging functionality in AWS S3, GCP Storage, and Azure Blob is also investigated to determine the feasibility of using the generated cloud storage log files for monitoring file activities on multiple CSPs.

6.2 Related Works

6.2.1 Research Works

Several works have proposed various monitoring systems for cloud storage services.

De Marco et al. [DFK15] utilized the AWS S3 server access log to monitor and detect the violations in the service level agreement (SLA) between AWS with cloud customers. Garion et al. [Gar+17] analyzed large amounts of cloud object storage service's log entries using Apache Spark to monitor the service's performance, estimate the potential for archiving the storage services, and detect security threats and anomalies of customer behavior. [DS19] developed a system to monitor Infrastructure-as-a-Service storage service's usage and analyze the file access patterns based on several files' parameters, e.g., access frequency, size, and replication. Torkura et al. [Tor+19a] proposed a cloud threat detection and incident response for multi-cloud storage systems called Slingshot by aggregating and analyzing cloud logs from AWS CloudTrail and GCP Logging with the cloud security assessment alerts. de Carvalho et al. [Car+17] proposed a monitoring and auditing mechanism for cloud storage services to verify the security properties of the data in the cloud and detect possible security violations using attestation elements of cloud transactions. Van Landuyt et al. [Van+19] evaluated continuous and client-centric trust monitoring for cloud storage services based on the statistical correlation between black-box performance metrics and reported white-box metrics.

[WTM17] collected network logs and user application logs periodically from the guest virtual machines (VMs) in Hadoop Distributed File System (HDFS). Then attack features are extracted using graph-based event correlation and MapReduce where it is analyzed using two-step machine learning algorithms to determine potential attack presence and path. [Li+16] analyzed the dataset of 350 million HTTP request logs from mobile cloud storage services to understand mobile user access behavior patterns and data transmission performance of the service. Berger et al. [Ber+16] introduced Cloud Security Intelligence that collects, processes, and analyzes data from various components of OpenStack to detect malicious activities and misconfiguration in the cloud, including audit trail functionality for OpenStack Swift to record data access activities using Apache Hadoop and Spark. Devarajan and SudalaiMuthu [DS19] proposed a cloud storage monitoring system that monitors the file storage usage and analyzes its information and access patterns using the K-Mean algorithm to rank the files stored on an IaaS platform. Based on the file ranking, the monitoring system could recommend the best solutions for users to optimize storage usage.

6.2.2 Competitors

Dropbox Business

Dropbox's monitoring services provide unified metrics and logging service for the on-premise and AWS cloud infrastructures and the Dropbox user's files stored on the cloud. It utilizes network security monitoring and intrusion detection systems to ensure only authorized non-malicious traffic could reach its infrastructure [Ama20d; Dro20].

DropBox Business also provides a monitoring functionality that records all events happening in the system and user and admin actions in the team's activity feed [Dro21d]. It is also capable to monitor Dropbox user's devices of Windows and Macintosh for malicious events. The security logs are then collected in a centralized location for forensic and incident response purposes [Dro20]. The team admin or user management admin of the organization could monitor how the users interact with the system using the admin console, such as how files are shared between the users, the number of internal and external file sharing events that happened, and how many devices are active perusers [Dro21b]. The activity feed can be exported as a downloadable report in CSV format and directly into third-party security information and event management solutions [Dro20].

If there are suspicious behavior, risky activity, or potential data leaks detected on the system, Dropbox Business will raise security alerts to the team admin [Dro21c]. The team admin then could inform or suspend the suspicious user based on the alert using the admin console. Several examples of events that might trigger the alerts are a large amount of data is deleted or moved by a user over a short period or files labeled as personal information are shared externally to people outside of the team.

Tresorit

Tresorit allows the organization administrator to monitor the activities of Tresorit users using the admin center dashboard [Tre21b]. It also allows for generating user activity report that contains detailed insights of how the users utilize the system, such as link sharing activities or tresor summary [Tre21a]. The organization's Tresorit account and the operational data could be exported as a CSV file for further analysis. It is not clear how Tresorit monitors the activities on the infrastructure and the files stored in Microsoft Azure.

Boxcryptor

Boxcryptor allows the company's administrator to monitor the activities of Boxcryptor users by recording the events related to users, devices, groups, and policies in the system [Box21c]. However, it is not clear whether it also provides monitoring functionality on the company's cloud storage services used with Boxcryptor.

6.2.3 Thesis Contribution

The work proposed in this chapter is different from the research community and the competitors to monitor file activities happening on the cloud storage services. The cloud object storage services' logging functionality and the generated cloud storage log files from different CSPs are compared and analyzed to investigate the feasibility of cloud storage monitoring using cloud storage log files. A multi-cloud storage monitoring system is then proposed for the CfB system that automatically collects, processes, and analyzes cloud storage log files from various CSPs with the CfB system log entries to monitor activities of the files stored on the cloud and detect suspicious or malicious activities on the cloud object storage services.

6.3 Multi-Cloud Storage Monitoring for CloudRAID for Business

Cloud monitoring is a process of tracking the latest state of the cloud infrastructure and measuring its key metrics, such as performance and availability. It is an important cloud management task for both the CSP and cloud customer for different reasons [Sye+17]. The CSP is responsible to install monitoring agents and services over its complex hardware and software infrastructure to gather information necessary for the cloud monitoring process [Fat+14; WB14]. The gathered information of the cloud infrastructure is then collected and processed for further use cases, such as ensuring efficient resource utilization, enforcing the agreed service level agreement with cloud customers, or troubleshooting apparent issues [Ace+13; Sye+17]. Cloud customers monitor the owned cloud resources based on the information gathered by the CSP's monitoring process using the dedicated monitoring view or service. Several use cases of cloud monitoring are detecting violations based on the agreed SLA, calculating the cost based on the resource usage, and managing the cloud resources [Ace+13; Sye+17].

CloudRAID for Business is responsible to securely manage and store the company's confidential files on the cloud on behalf of the companies. One of CfB's cloud data management tasks is monitoring the activities happening of cloud object storage services in multiple CSPs, including the files owned by the companies [Car+17; Sye+17]. This is due to the information provided by the CfB system is not enough for companies and CfB to oversee the complete activities in the entire environment. CfB could then monitor its multi-cloud storage environment by collecting, processing, and analyzing the generated **cloud storage log files**. The log files provide information on the events happening in the cloud object storage services, such as the requester's information and response information [Kha+16; Tor+19a]. The information from the cloud storage log files could be processed and correlated with the information from the CfB system for further analysis, e.g., forensic investigation or data analytics [PLS15].

CfB could utilize the information from cloud storage monitoring for multiple purposes. The main objective of cloud storage monitoring is to provide data provenance, which summarizes the history and the information of the data from its creation to deletion [LAA15]. Data provenance could be used to provide the activity timeline of the company's files, e.g., who has accessed the files and how the files are shared between the company employees. It helps the CfB to provide accountability of the files stored in the cloud as only authorized CfB users should be able to access the files where the file access permission must be generated by the CfB system [Lia+17a]. It could also be used to identify suspicious activities both in the CSPs and the system where it could be used as the evidence for forensic investigation [Alq+16]. It also helps CfB to calculate storage usage cost in multiple CSPs [Gar+17] and discovers malicious activities and misconfiguration in the cloud object storage services [Tor+19a]. Monitoring the events happening could also help the companies to detect and investigate possible insider threat incidents.

However, several challenges are faced by the CfB to monitor the file activities in multiple CSPs using cloud storage log files as explained in Chapter 5.

Cloud storage monitoring is a complex process and to simplify this process for cloud customers, each CSP provides cloud log management and monitoring services to monitor the activities on the cloud object storage services, such as AWS CloudWatch or GCP Logging. However, these services lack cross-CSP collaboration functionality CSPs where cloud storage log entries might only be viewed and processed using these services. Meanwhile, the cloud storage log files could be automatically deleted after a certain period [Ama21g; Goo20f]. This would require CfB to collect, process, and analyze the generated cloud storage log files to monitor the storage activities in multiple CSPs on its own due to the shared responsibility model, which increases the complexity of monitoring its multi-cloud storage environment.

As it processes the cloud storage log files from various CSPs, CfB then needs to resolve the heterogeneity of the logging functionality of cloud object storage services from multiple CSPs to monitor the file activities stored on the cloud. The generated cloud storage log files could also have a different format, structure, and information quality from various CSPs [PLS15]. Meanwhile, CfB could not influence how the cloud storage services would behave, including how the cloud storage log files are delivered to CfB and what information is recorded on the log files based on the events in the cloud object storage services. Finally, CfB also needs to correlate the cloud storage log files with the CfB system log entries to obtain the complete information of the CfB user's file activities on the system and the cloud. This would require extensive skills and knowledge on the cloud object storage services to monitor file activities happening on the cloud.

6.4 The State of Logging Functionality of Cloud Object Storage Services

Cloud service providers provide the logging functionality that records the various event types happening to the cloud object storage services, the buckets, and the objects as **cloud storage log files**. This could be a built-in functionality or configuration in the cloud object storage services or a separate cloud logging or monitoring service that record all types of events happening in the CSP environment. Enabling logging functionality for the monitored buckets could be free or have an additional small cost to the cloud customers, however, the CSP will charge cloud customers for the storage and transfer of the actual cloud storage log files.

There are two types of cloud storage log provided by the CSPs:

- **Storage access log**: Storage access log records the events happening in a selected monitored bucket, e.g. object download or bucket accessed. It provides a structured and simple data format, such as CSV. Storage access log files are then delivered to a target bucket. Cloud customers need to enable the logging configuration for each monitored bucket or the entire cloud storage services.

- **Cloud activity log**: Cloud activity log records the events in the cloud resources and services used and owned in the CSP environment using cloud logging and monitoring services. It provides semi-structured data commonly in JSON format to accommodate different data models of the cloud resources. It could be used to monitor activities happening in the cloud object storage services or a specific bucket. Generally, the cloud activity log has more detailed information than the storage access log.

  Although the CSP's cloud logging and monitoring services might already record the events happening in the cloud environment as cloud activity log entries by default, the log entries could only be viewed and processed using the cloud logging and monitoring services. Cloud customers are then required to store or export the cloud activity log files since the log entries might be deleted after a certain period [Ama21g; Goo20f].

The cloud storage log files record the information of the executed requests and the successful or failed responses sent by the cloud object storage services depending on the ACL or policy configuration of the buckets and their objects. There are four types of requests in the cloud object storage services:

- **Unauthenticated request**: The request is executed by anonymous actors that are not authenticated to the CSP. Depending on the configuration of the bucket and its objects, the request could return a successful or failed response. The actors could request the URL of the bucket or the objects stored in the bucket to check if the bucket exists or the bucket and its objects are publicly accessible [Con+18].
- **Authenticated request**: The request is executed by CSP customers, CSP services, or other entities that exist in the authorized cloud customer's CSP domain, such as service account or virtual machine, using valid CSP credentials. Depending on the configuration of the bucket and its objects, the request could return a successful or failed response.
- **Authorized request**: The request is executed by actors who have the correct or sufficient privileges fulfilling the configuration of the bucket and its objects where the cloud storage services return a successful response.
- **Unauthorized request**: The request is executed by actors who have insufficient or incorrect privileges to fulfill the configuration of the bucket and its objects where the cloud storage services returns failed response.

Cloud storage log entries are expected to record at least the information of the event happening in the cloud object storage services as follows:

- **Timestamp**: The time the request is received by the CSP.
- **IP address**: The Internet address of the requester.
- **Requester information**: The information that could be used to identify the requester, e.g., IAM entity, cloud customer, a CSP service, or anonymous user.

- **Request ID**: The request identifier generated by the CSP.
- **Bucket**: The bucket specified in the request.
- **Object**: The object specified in the request.
- **Request URI**: The uniform resource identifier (URI) of the cloud resource listed in the request.
- **Request method**: The method listed in the request.
- **Request length**: The length of the request.
- **Response length**: The length of the response.
- **Duration**: The time it took by the CSP to process the request.
- **Response code**: The status code of the response.
- **Response message**: The message of the response code.
- **User agent**: The user agent used to send the request.

Cloud customers, including CfB, could expect several characteristics of the cloud storage log files to monitor the events in cloud object storage services:

- **Reliable log delivery**: Cloud storage log must be delivered after the event is happening in cloud storage services.
- **Record all requests**: Cloud storage log must record all requests made to cloud storage services, i.e., authorized, unauthorized, and unauthenticated requests.
- **Record requester's information**: Cloud storage log must record the information of the entity that executes the request.
- **Record request information**: Cloud storage log must record the information of the request sent by the requester.
- **Record response information**: Cloud storage log must record the information of the response sent by the CSP.

- **Consistent log values**: Cloud storage log should have consistent log values for different request types.

- **Consistent log fields**: Cloud storage log should have consistent log fields for different request types.

6.4.1 Amazon Web Services Simple Storage Service (S3)

AWS S3 provides the **server access log** [Ama20a] by enabling the bucket logging option in the monitored AWS S3 bucket. Events in the monitored bucket will be periodically collected and written as log objects by AWS S3 Log Delivery group to the target bucket. The log files are delivered on a best effort basis where they can be delivered within a few hours of the time, the activity happened [Ama20a].

AWS CloudTrail can be used to provide **cloud activity log** for AWS S3 services. It continuously monitors and logs all events in the AWS infrastructure, including AWS S3's bucket-level and object-level API calls in the monitored bucket or AWS S3 account [Ama20e]. The log entries are only available for 90 days in the CloudTrail console and a trail needs to be enabled to deliver the log files every 5 minutes to the specified target bucket or AWS CloudWatch to actively persist the log entries [Ama20c; Ama21g].

6.4.2 Google Cloud Platform Storage Service

GCP Storage provides the **usage log** to monitor activities happening in the GCS bucket [Goo20a]. The logging functionality on the monitored bucket could be enabled using gsutil, JSON API, or XML API. After the bucket logging option in the monitored GCP Storage bucket has been enabled, all events happening in the monitored bucket will be periodically collected and written as log files by the GCP Storage Analytics group to the target bucket. The log files are delivered hourly approximately 15 minutes after the end of the hour. Duplicate log entry could exist for the log files created in the same hour that could be detected by checking the *s_request_id* field [Goo20a] .

GCP Logging generates **cloud audit log** that records activities in a GCP project, folder, or organization. For Google Cloud Storage service, it generates two types of cloud audit log: Admin Activity log that records the configuration or metadata modification of project, bucket, or object, and Data Access log that records the activities of the project, bucket, or object [Goo20b]. CfB system utilizes only the **data access audit log** from the cloud audit log to record the events in the monitored GCP Storage bucket. Since log entries are available up to 30 days on the Logging console, a log sink needs to be created to deliver the log files around every hour to the target bucket in GCP Storage, GCP Pub/Sub[70], or GCP BigQuery[71] [Goo20d; Goo20f].

6.4.3 Microsoft Azure Storage Blob Service

Azure Storage Blob (Blob) provides **storage analytics log** [Mic20b] that records detailed events happening in the storage account on the best effort basis. The log files are delivered to a container named `$logs` up to every hour. Duplicate log entries may exist for the log files created in the same hour and could be detected by checking the duplicate values in *RequestId* field and value more than 0 for *Operation* fields [Mic20b]. CfB system utilizes storage analytics log version 2.0 to monitor the Blob service.

Storage resource log generated by Azure Monitor[72] is used by the CfB system as the cloud activity log to provide detailed diagnostic and auditing information of the events within the Azure infrastructure, including the storage account [Mic20c]. The log files are delivered every hour to Azure Event Hubs[73] or the storage account [Mic20a].

6.5 Multi-Cloud Storage File Monitoring Systems for CloudRAID for Business

A multi-cloud storage file monitoring system is proposed for CloudRAID for Business to monitor the file activities happening in multiple CSPs as can be seen in Figure 6.1. It follows the data warehouse method [Hu+14], which consists of extraction, transformation, and loading steps, to provide a unified multi-cloud storage activity monitoring view rather than utilizing different cloud logging and monitoring services from various CSPs.

The monitoring system collects, processes, and analyzes generated cloud storage log files from various CSPs. It also correlates the cloud storage log files

[70] https://cloud.google.com/pubsub/

[71] https://cloud.google.com/bigquery/

[72] https://azure.microsoft.com/en-us/services/monitor/

[73] https://azure.microsoft.com/en-us/services/event-hubs/

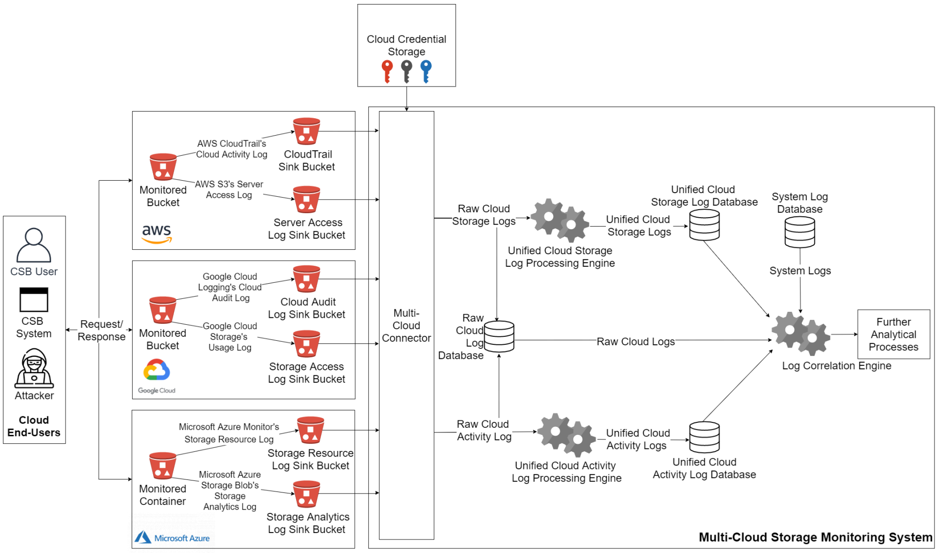

Cloud Credential Storage
CSB User
CSB System
Attacker
Cloud End-Users
Request/ Response
Monitored Bucket
AWS CloudTrail's Cloud Activity Log
CloudTrail Sink Bucket
AWS S3's Server Access Log
Server Access Log Sink Bucket
aws
Monitored Bucket
Google Cloud Logging's Cloud Audit Log
Cloud Audit Log Sink Bucket
Google Cloud Storage's Usage Log
Storage Access Log Sink Bucket
Google Cloud
Monitored Container
Microsoft Azure Monitor's Storage Resource Log
Storage Resource Log Sink Bucket
Microsoft Azure Storage Blob's Storage Analytics Log
Storage Analytics Log Sink Bucket
Microsoft Azure
Multi-Cloud Connector
Raw Cloud Storage Logs
Unified Cloud Storage Log Processing Engine
Unified Cloud Storage Logs
Unified Cloud Storage Log Database
System Log Database
System Logs
Raw Cloud Log Database
Raw Cloud Logs
Log Correlation Engine
Further Analytical Processes
Raw Cloud Activity Log
Unified Cloud Activity Log Processing Engine
Unified Cloud Activity Logs
Unified Cloud Activity Log Database
Multi-Cloud Storage Monitoring System

Figure 6.1: Overview of architecture CloudRAID for Business' multi-cloud file storage monitoring system [Suk+21a]

with the CfB system log entries to obtain complete information of the activities of confidential files owned by the companies as the CfB customers. Based on the correlated log information, the monitoring system could perform further analysis, e.g., file-sharing tracking or suspicious activity detection.

The CfB first sets up three-bucket types in each CSP as follows:

- **File bucket** stores the files of each company separately as explained in Chapter 5.
- **Storage access log bucket** stores the storage access log files generated by the cloud storage services.
- **Cloud activity log bucket** stores the cloud activity log files generated by the cloud monitoring services.

The CfB sets a storage access log bucket and a cloud activity log bucket to store all cloud storage log files for all the companies. Each company's file bucket is configured where the activities on the file buckets will be logged as the storage access log files and cloud activity log files and delivered to the respective buckets.

As explained in Chapter 5.4, **Multi-Cloud Connector's Log Collector application** continuously checks and fetches for newly generated cloud storage log files from the cloud storage log buckets in multiple CSPs to the CfB's multi-cloud file storage monitoring system, which could take between 5 minutes to a couple of hours depending on the logging functionality of cloud object storage services. The Log Collector application utilizes one Access Key for each CSP retrieved from the **Cloud Credential Storage** to list and download the cloud storage log files. It then pre-processes the cloud storage logs for easier storage and further processing, such as parsing AWS S3's server access log's space-delimited format to CSV format. The fetched storage access log files and cloud activity log files are then stored on the **raw cloud storage log database**.

Unified storage access log format (see Table 6.1) and unified cloud activity log format (see Table 6.2) are proposed to solve the heterogeneity of cloud storage log from multiple CSPs [PLS15]. Information necessary for monitoring activities on cloud storage services from the cloud storage log of various CSPs is selected and pre-processed to a single unified log format, where the value could be in a different format or contain information for multiple unified log fields. The unified log format also helps to transform cloud activity log entries from semi-structured data in JSON format to structured data, e.g., in CSV format.

The multi-cloud file storage monitoring system parses the raw cloud storage log files following the proposed unified storage access log format and unified cloud activity log format. Since the cloud activity log format is designed to monitor various types of cloud services and resources, the unified cloud activity log format is modified slightly to accommodate the information needed to monitor the activities on cloud storage services. The parsed unified cloud storage log entries are then stored on the **Unified Cloud Storage Log database**.

Meanwhile, every CfB user's file activity using the CfB client application to a particular CSP is recorded as a CfB system log entry and stored in the **CfB System Log database**. Since CfB users utilize signed URL requests to access the files on the cloud, the cloud object storage services return response code and optional response message values, which will be forwarded to the CfB system and written in the system log entry. An example of a CfB system log entry for file chunk download using signed URL method is as follow:

```
{
   "userId":"d3dbb8a1-9044-4f1f-b43d-6d66a79e1a6d",
   "userIp":"141.89.221.247",
```

Unified Storage Access Log	**AWS S3 Server Access Log**	**GCP Storage Usage Log**	**Azure Blob Storage Analytics Log**
provider	aws_s3	gcp_storage	azure_blob
requestID	Request-ID	s_request_id	request-id-header
timestamp	Time	time_micros*	request-start-time
ipAddress	Remote-IP	c_ip	requester-ip-address*
bucket	Bucket	cs_bucket	requested-object-key*
objectKey	Key	cs_object	requested-object-key*
requestMethod	Operation	cs_operation	operation-type
statusCode	HTTP-status	sc_status	http-status-code
statusMessage	Error-Code	-	request-status
requestURI	Request-URI	cs_uri	request-url
userAgent	User-Agent	cs_user_agent	cs_user_agent
referrer	Referrer	cs_referer	referrer-header
requestLength	-	cs_bytes	request-packet-size
responseLength	Bytes-Sent	sc_bytes	response-packet-size
operationTime	Total-Time	time_taken_micros*	end-to-end-latency-in-ms

Table 6.1: Unified storage access log format for AWS S3's server access log, GCP Storage's usage log, and Azure Blob's storage analytics log. Log field with asterisk (*) sign requires further normalization [Suk+18]

Unified Cloud Activity Log	**AWS CloudTrail Cloud Activity Log**	**GCP Logging Cloud Activity Log**	**Azure Monitor Resource Log**
requestID	requestID	-	correlationId
timestamp	eventTime	timestamp	time
provider	aws_s3	gcp_storage	azure_blob
bucketName	requestParameters.bucket	protoPayload.*resourceName	uri*
objectName	requestParameters.key	protoPayload.*resourceName	uri*
region	awsRegion	resource.label.location	location
requestMethod	eventName	protoPayload.methodName	operationName
ipAddress	sourceIPAddress	protoPayload.requestMetadata.callerIP	callerIpAddress
userAgent	userAgent	protoPayload.requestMetadata.callerSuppliedAgent	properties.userAgentHeader
responseCode	errorCode	protoPayload.status.code	statusCode
responseMessage	errorMessage	protoPayload.status.message	statusText
requester	userIdentity.arn	protoPayload.authenticationInfo.principalEmail	-
requestLength	additionalEventData.bytesTransferredIn	-	properties.requestBodySize
responseLength	additionalEventData.bytesTransferredOut	-	properties.responseBodySize

Table 6.2: Unified cloud activity log format specifically to monitor cloud object storage services for AWS CloudTrail's cloud activity log, GCP Logging's cloud activity log, and Azure Monitor's resource log. Log field with asterisk (*) sign requires further normalization [Suk+20]

```
   "date":"2020-10-30T14:33:44.841603100Z",
   "fileName":"testfile\_k01",
   "fileStorageType":"ERASURE",
   "fileAccessType":"URL",
   "fileActionType":"DOWNLOAD",
   "csp":"AWS",
   "fileSize":524800,
   "httpStatus":200,
   "url":"https://testbucket.s3.eu-central-1.amazonaws.com/testfile_k01?X-Amz-Alg
   orithm=AWS4-HMAC-SHA256&X-Amz-Date=20201030T143344Z&X-Amz-SignedHeaders=host&
   X-Amz-Expires=60&X-Amz-Credential=************%2F20201030%2Feu-central-1%2Fs3
   %2Faws4_request&X-Amz-Signature=54627f70dd2e41ac7df6e98eb68edccb8101637ffd61e
   16edc855dae5b893d16"
}
```

Once the cloud storage log files have been collected from multiple CSPs, preprocessed, and stored in the database, the unified and/or raw storage access log entries are correlated with cloud activity log entries using **log Correlation engine** to fill the information gap of both log types. The Log Correlation engine also correlates unified and/or raw cloud storage log entries with CfB system log entries to provide contextual information of the cloud storage log entries from the point of view of CfB.

The log correlation engine works in a batch-processing manner [Hu+14] where it waits for cloud storage log entries to be fetched and stored first before processing it with CfB system log entries. The engine utilizes unique and identifiable value that exists on both logs. If it fails, the correlation could be done using the values on common log fields available on both logs. Duplicate information on the correlated log information is then removed to simplify the information.

Finally, the CfB system log, storage access log, and cloud activity log, both in a raw and unified format, as well as the correlated log are then processed for further analysis, such as storage usage tracking, suspicious user behavior, and tracking file activities on the CfB system and the cloud. CfB administrator and company's administrator could then monitor the information related to the files stored in multiple CSPs through the administrator dashboard.

6.6 Evaluation

In this section, the CfB's multi-cloud file storage monitoring system is evaluated to ensure how feasible it is to monitor the activities of the files stored in the cloud using the generated cloud storage log files from multiple CSPs. The logging functionality of cloud object storage services of AWS, GCP, and Azure are investigated to evaluate the behavior and the information quality of the cloud storage log files. Finally, the cloud storage log files from various CSP are then correlated with the CfB system log entries to obtain complete activities of CfB user's files stored in the cloud.

File upload, download, and delete activities of 10 CfB users are simulated using the signed URLs generated by the CfB system where the users are from the same organization with the same public IP address. Several unauthorized and unauthenticated download file activities done by the **attacker**, which represents anonymous Internet users, malicious CfB users, and malicious CfB employees following the threat model in [Tor+18a], are also simulated using signed URLs and API to gain unauthorized access to the company's confidential files stored in the CfB's file buckets as follows:

- Attacker requests the object storage URLs.
- Attacker accesses the objects using their CSP credential via API.
- Attacker requests object signed URLs generated using their CSP credential via API.
- Attacker requests expired object authorized signed URLs.
- Attacker requests modified object authorized signed URLs.

The cloud storage log files from AWS S3, GCP Storage, and Azure Blob services, and CfB system log entries generated from the evaluation scenario are then collected, processed, and analyzed using the CfB's multi-cloud file storage monitoring system.

6.6.1 Cloud Storage Log Comparison and Correlation

Category	AWS S3 Server Access Log	AWS CloudTrail Cloud Activity Log	GCP Storage Usage Log	GCP Logging Cloud Activity Log	Azure Blob Analytics Log	Azure Monitor Resource Log
Log file delivery	Best effort [Ama20a]	Every 5 minutes [Ama20c]	Every hour [Goo20a]	Every hour [Goo20d]	Every hour [Mic20b]	Every hour [Mic20a]
Log format type	Space delimited	JSON	CSV	JSON	CSV	JSON
Log format selection	✗	✗	✗	✗	✓	✗
Duplicate entry	✗	✗	✓	✓	✓	✓
Timestamp	✓	✓	✓	✓	✓	✓
IP address	✓	✓	✓	✓	✓	✓
Requester information	✓	✓	✓	✓	✓	✓
Request ID	✓	✓	✓	✗	✓	✓
Bucket	✓	✓	✓	✓	✓	✓
Object	✓	✓	✓	✓	✓	✓
Request URI	✓	✗	✓	✗	✓	✓
Request method	✓	✓	✓	✓	✓	✓

Request length	✓	✓	✓	✓	✓	✓
Duration	✓	✓	✓	✓	✓	✓
Response length	✓	✓	✓	✓	✓	✓
Response code	✓	✓	✓	✓	✓	✓
Response message	✓	✓	✗	✓	✓	✓
Signed URL signature	✓$^{+}$	✗	✓	✗	✓$^{+}$	✓$^{+}$
Object access method difference	✓	✓	✓	✗	✓	✓
Consistent number of log fields	✓	✗	✓	✗	✓	✓
Consistent log values	✓	✓	✗	✓	✓	✓
Authorized API request	✓	✓	✓	✓*	✓	✓
Authorized signed URL request	✓	✓	✓	✓*	✓	✓

Unauthenticated object storage URL request	✓	✓	✓	✓*	✗	✗
Unauthorized API request	✓	✓	✓	✓	✗	✗
Unauthorized signed URL request	✓	✓	✓	✓*	✗	✗
Expired signed URL request	✓	✗	✓	✗	✗	✗
Modified signed URL request	✓	✗	✓	✗	✗	✗

Table 6.3: Comparison of cloud storage logs from Amazon Web Services, Google Cloud Platform, and Microsoft Azure. $^{+}$= Signature anonymized, *= Using custom user agent [Suk+21a]

Table 6.3 shows the overview of the data quality and the behaviour of storage access log files and cloud activity log files from AWS S3, GCP Storage, and Azure Blob generated from the evaluation scenario. The cloud storage log files are then correlated to complete the missing information gap of each log.

Amazon Web Services Simple Storage Service

AWS S3's server access log provides a summary of the events in the monitored bucket. Meanwhile, AWS CloudTrail's cloud activity log provides more detailed information about the events in the AWS S3 environment, such as the requester identity and the request parameters.

Cloud activity log does not have *Request URI* field, which exists only in the server access log. It also does not have a consistent number of log fields affected by requester types (authenticated or unauthenticated actors) and the response status (authorized or denied). The values on both logs are consistent for almost all fields, except for the request method in the *Operation* field in the server access log and *eventName* field in the cloud activity log. For example, object upload method is logged as `REST.PUT.OBJECT` (server access log) and `PutObject` (cloud activity log).

The server access log files are observed to be delivered around **20 to 40 minutes** after the event recorded in the CfB system log, while the cloud activity log files are delivered about **10 minutes** after the event. The log files consist of one or multiple log entries where the files could be generated **unsorted** regardless of when the requests happened. The timestamp in the *Time* field (server access log) and *eventTime* field (cloud activity log) are almost identical with **1 second** maximum difference. The requester could be identified using the *Requester* field (server access log) and the *userIdentity* field (cloud activity log). The request made using API or pre-signed URL could be differentiated using *Authentication Type* field (server access log) and *authenticationMethod* field (cloud activity log).

The signature embedded in the pre-signed URL in AWS S3 is anonymized in the *Request-URI* field of the server access log, such as:

```
/CkSPr1lT25_k02?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20200802T095
536Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-Credential=<creden
tial>%2F20200802%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Signature=XXXX
```

The server access log records all of the attacker's unauthenticated and unauthorized requests, while cloud activity log **does not log** object access requests using expired signed URL and modified signed URL. This is because AWS CloudTrail records failed authorization requests and unauthenticated requests, but it does not record the request with failed authentication [Ama20e].

There were several attempts to access the CfB's file bucket by unknown actors outside of the simulated attacker requests were detected in the storage access log files during the evaluation phase, as can be seen in Figure 6.2. Several anonymized authorized `GetBucketLocation` requests to determine in which region the bucket resides in and unauthorized `HeadBucket` requests from AWS Config service[74] to determine if the bucket exists and the requester has permission to access it were recorded. Unauthorized and unknown AWS accounts also launched crawling attempts to the CfB's file bucket using `GetBucket` (list the content of the bucket), `GetBucketACL` (retrieve the bucket's ACL configuration), and `HeadBucket` requests. Most of the detected requests from unknown IP addresses outside of the attacker scenario are trying to determine if the CfB's AWS file bucket and its objects are publicly accessible due to misconfiguration.

The server access log could be correlated with the cloud activity log using the *Request ID* fields in both log types to generate **one-to-one correlation result**. However, several server access log entries of several unauthenticated and unauthorized requests could not be correlated with the cloud activity log entries since the AWS CloudTrail does not record these requests.

Google Cloud Platform Storage Service

GCP Storage's storage usage log provides a summary of events in the monitored bucket with a consistent number of log fields for different requests. Although GCP Logging's cloud audit log contains more fields than the storage usage log, it could have a different number of fields depending on the request types.

The storage usage log includes *Response code* in *sc_status* field however it does not include *Response message* information. The cloud audit log does not have the *Request ID* and *Request URI* log fields. The values on both log types are almost consistent except for the request method in the *cs_method* field in the storage usage log and *methodName* field in the cloud audit log, e.g., object upload method is logged as `PUT_Object` and `storage.objects.insert` (server access log) and `storage.objects.create` (cloud activity log).

The request status code for cloud audit log in *status* field utilizes Google API's

[74] https://aws.amazon.com/config/

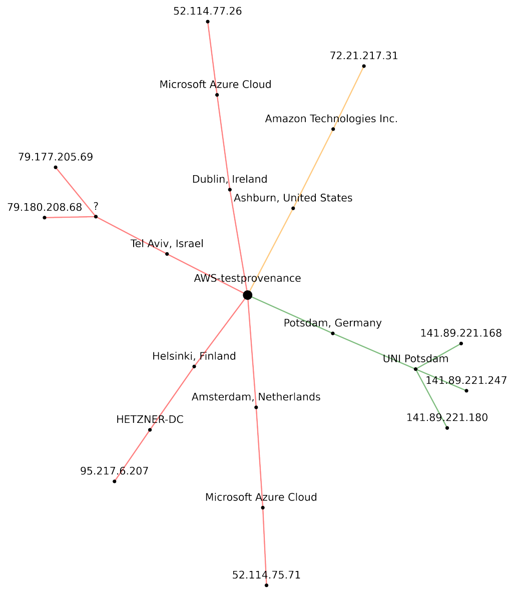


Figure 6.2: Overview of the requester's location based on the IP address accessing monitored CfB file bucket. Green line = activities from known authorized requesters. Yellow line = activities from the AWS. Red line = activities from unknown requesters.

remote procedure call status codes[75] where successful request returns null status code. Meanwhile, the *Response code* field in storage usage log utilizes HTTP status codes defined in RFC 7231[76].

The storage usage log files and cloud audit log files are delivered between **1 to 2 hours** after the event recorded in the CfB system log entries. The delivered log files record one or multiple log entries where the files could be generated **unsorted** regardless of when the requests happened.

The timestamp recorded in the storage usage log's *time_micros* field is stored as microseconds since Unix epoch would require parsing to the conventional date. The cloud audit log has two timestamp fields: *receiveTimestamp*, which

[75] https://github.com/googleapis/googleapis/blob/master/google/rpc/code.proto
[76] https://tools.ietf.org/html/rfc7231

describes when the log entry is received by Google Logging, and *timestamp* that stores the time when the request is received by GCP Storage and comes earlier than *receiveTimestamp* field. The timestamp recorded in the cloud audit log is earlier compared to the timestamp in the storage usage log with the observable differences of between **4 to 140 milliseconds**.

The storage usage log records the request URI in the *cs_uri* field with the signature embedded in the signed URL is recorded in an exact form while the cloud audit log does not record the signature. The request from signed URL and API could be differentiated in the storage usage log by checking if `X-Goog-Signature` exists in the *cs_uri* field or checking the values in *cs_method* field, for example, object download operation is logged as *GET_Object* (signed URL) or *storage.objects.get* (API). However, the request from signed URL and API in the cloud audit log could not be differentiated. Therefore, the evaluation scenario is done using custom user-agents to help identify the requests made by the CfB users and the attackers on the cloud audit log.

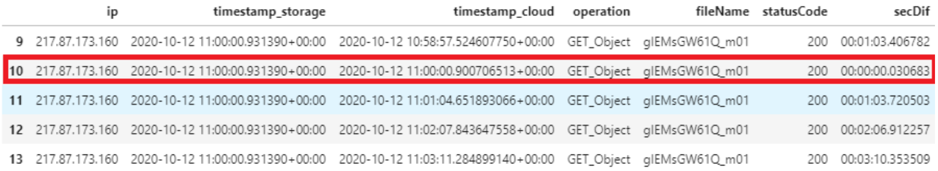
The storage usage log records all of the attacker's unauthenticated and unauthorized requests to the CfB's bucket and its objects. However, the cloud audit log **does not record** the attacker's unauthenticated access request with expired and modified signed URLs. A suspicious request from unknown IP addresses launching a HEAD HTTP operation was discovered on the storage usage log to list the objects contained in the bucket or get the configuration of the bucket.

The storage usage log could not be directly correlated with the cloud audit log since the cloud audit log does not have the *Request ID* field. Although both logs could be correlated using the similar existing fields that could be used to identify the request, e.g., *IP address*, *User agent*, *Object*, *Request method*, and *Response code*, the values of log fields need to be pre-processed and uniform to ensure the correlation works, such as response code and request method fields.

The log correlation process could result in **one-to-many** result, where one storage usage log entry is correlated with multiple cloud audit log entries. A possible solution to resolve this issue is to select the correlated log entry with the minimum timestamp difference between *time_micros* field (storage usage log) and *timestamp* field (cloud audit log) as shown in Figure 6.3.

Microsoft Azure Blob Storage Service

Azure Blob's storage analytics log provides more comprehensive information about the event happening in the storage account than Azure Monitor's storage



	ip	timestamp_storage	timestamp_cloud	operation	fileName	statusCode	secDif
9	217.87.173.160	2020-10-12 11:00:00.931390+00:00	2020-10-12 10:58:57.524607750+00:00	GET_Object	glEMsGW61Q_m01	200	00:01:03.406782
10	217.87.173.160	2020-10-12 11:00:00.931390+00:00	2020-10-12 11:00:00.900706513+00:00	GET_Object	glEMsGW61Q_m01	200	00:00:00.030683
11	217.87.173.160	2020-10-12 11:00:00.931390+00:00	2020-10-12 11:01:04.651893066+00:00	GET_Object	glEMsGW61Q_m01	200	00:01:03.720503
12	217.87.173.160	2020-10-12 11:00:00.931390+00:00	2020-10-12 11:02:07.843647558+00:00	GET_Object	glEMsGW61Q_m01	200	00:02:06.912257
13	217.87.173.160	2020-10-12 11:00:00.931390+00:00	2020-10-12 11:03:11.284899140+00:00	GET_Object	glEMsGW61Q_m01	200	00:03:10.353509

Figure 6.3: An example of determining one-to-one correlation between GCP Storage's storage usage log and GCP Logging's cloud audit log by selecting the correlated log entry with the minimum timestamp difference (red box).

resource log. Since both logs record the events in the whole storage account, the evaluation only focuses on the events happening on the CfB's monitored container by applying the filter on *request-url* field in the storage analytics log and *uri* field in the storage resource log.

The values and the number of fields on both log types are consistent for all request types. Although no duplicate log entries are detected for the storage analytics log, there is a possible duplicate log entry detected for the storage resource log by checking the duplicate value for *correlationId* field.

The successful request from the shared access signature and API in both logs could be differentiated by checking the *authentication-type* field in the storage analytics log and *type* field in the storage resource log. The signature embedded in the shared access signature is anonymized in *request-url* field (storage analytics log) and *uri* field (storage resource log), such as:

```
https://<storageAccount>.blob.core.windows.net:443/<bucketName>/RPDrRJKukd_k01?
sig=XXXXX&amp;st=2020-08-21T09%3A49%3A28Z&amp;se=2020-08-21T09%3A59%3A28Z&amp;sv=
2019-02-02&amp;sp=r&amp;sr=b
```

The request is logged in the order of arrival of the request to Azure Blob service. The storage analytics log files and storage resource log are delivered around **1 hour** after the event recorded in the CfB system log entry. Both log files consist of one or multiple log entries where the files could be generated **sorted** based on the request's timestamp. The timestamp difference between the storage analytics log and storage resource log is observed between **1 millisecond to 24 seconds**, which corresponds to *end-to-end-latency-in-ms* field in the storage analytics log. The timestamp difference may vary as it is affected by the duration the Azure Blob takes to process the request, especially for file upload requests with different file sizes.

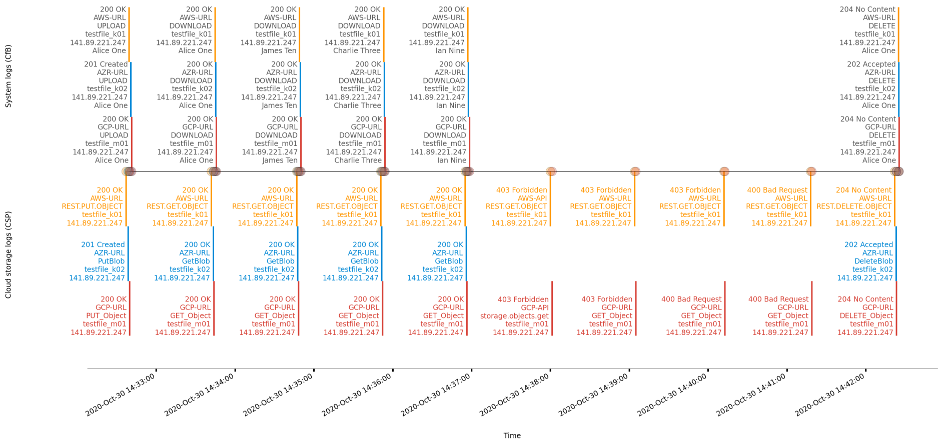


Figure 6.4: Example of the file activity timeline based on the correlated cloud storage log files from AWS S3 (yellow), GCP Storage (red), and Azure Blob (blue) services with CfB system log entries.

Both storage analytics log and storage resource log **do not record** any requests made by the attackers to the bucket and its objects. This is due to Azure only records the anonymous request, e.g., server errors or timeout errors, but it does not log failed unauthenticated and unauthorized requests to the storage account [Mic20b]. This affects unauthorized or unauthenticated requests to CfB's container and blobs outside the attacker scenario to be undetected.

Both logs could be correlated using *correlationId* field in the storage resource log and *requestIdHeader* field in the storage analytics log to generate **one-to-one log correlation result**.

6.6.2 CfB System Log Entries with Cloud Storage Log Entries Correlation

CfB's multi-cloud monitoring system should be able to correlate CfB system log entries with either storage cloud storage log files from multiple CSPs to provide full monitoring information of the activities happening on the cloud object storage services and the CfB user's files stored in the cloud. However, there are several challenges faced by the CfB's multi-cloud monitoring system to correlate these different logs.

The executed file requests, either using API or signed URL, do not return the request identifier (ID) value on the response received. It is due to the CSP not returning the request identifier of the executed request in the response information as it may not be propagated correctly in various CSP components [Yu+16]. The CfB system log entries then do not contain the request ID value that could help direct correlation with the cloud storage log entries for a one-to-one log correlation result. Meanwhile, the response information of the executed request might also contain an empty response code and response message, although it could be circumvented for a signed URL by using the response information from the HTTP request.

The CfB system log entries could still be correlated with cloud storage log entries using the log fields exist on both logs, such as *Timestamp*, *IP address*, *User agent*, *Bucket*, *Object*, *Request method*, *Response code*, and *Request URI* (if available). The cloud storage log files then need to be pre-processed through several steps before correlated with the CfB system log entries.

First, Azure Blob's cloud storage log files need to be filtered to only show events from the monitored bucket. Any duplicate log entries on the cloud storage log files need to be removed as explained previously. Certain values in cloud storage log files need to be uniform following the CfB system log format, for example, file upload request is logged as `UPLOAD` (CfB system), `REST.PUT.OBJECT` and `PutObject` (AWS S3), `storage.objects.insert` and `PUT_Object` (GCP Storage), and `PutBlob` (Azure Blob).

The proposed unified storage access log format and unified cloud activity log format help multi-cloud file storage monitoring system to pre-process the cloud storage log files from various CSPs with different log formats and information quality to a single simple format. It would allow the multi-cloud file storage monitoring system to simplify the information from the cloud storage log files for further processes. However, there might be a loss of information in the unified log format due to only necessary log fields are persisted.

The log correlation could generate **one-to-many result** where one CfB system log entry is correlated with multiple cloud storage log entries. This is due to the CfB system log entry's timestamp is slightly late compared to the timestamp in cloud storage log entries as the CfB waits on the response sent from the cloud object storage services of the executed request to write the corresponding system log entry. One-to-one log correlation could be achieved by selecting the correlated log entry with the smallest time difference between the CfB system log entry and cloud storage log entry.

Since all file requests to multiple CSPs should be authorized by the CfB system, the CfB system log is then used as the **source of truth** for the evaluation as explained previously. This means a CfB system log entry should have corresponding cloud storage log entries from the three CSPs, and vice-versa. If there is a cloud storage log entry that is not correlated with the CfB system log entry, two explanations are possible, depending on the response information in cloud storage log entry:

- If the cloud storage log entry shows successful request code and message, such as 200 or `SASSuccess` (Azure Blob's analytics log), there is an unaccounted authorized request outside the CfB system's behavior that might be executed by entities in the CfB system, anonymous user, or the CSP. It could also mean the buckets where the CfB user's files are stored are misconfigured, which allows unauthorized entities to access it.

- If the cloud storage log entry shows failed request code and message, such as 403 or `AccessDenied` (AWS CloudTrail's cloud activity log), the request is made by unauthorized entities and the buckets are correctly configured.

The uncorrelated cloud storage log entries would require further investigations to determine the nature of the requests to the cloud object storage services or what causes the request to be authorized, such as checking the configuration of the bucket and the objects. CfB administrator will be alerted of the uncorrelated cloud storage log entries where it will be further investigated to determine the nature of the requests to the cloud object storage services. Meanwhile, the company administrator will be notified if possible data leaks are happening based on the successful uncorrelated log entries.

After the information from CfB system log entries and cloud storage log files from multiple CSPs have been correlated, the information could be processed for further analysis. A possible use case of processing the correlated log entries is data provenance. Figure 6.4 shows an example of the timeline of a CfB user's file stored in AWS S3, GCP Storage, and Azure Blob services from its initial storage to the final file deletion on the cloud. The timeline shows the files were accessed by multiple CfB users and the attacker following the evaluation scenario where several cloud storage log entries of failed object access attempts do not have the corresponding CfB system log entries. This indicates the buckets are correctly configured to deny unauthorized file access requests outside the CfB system.

6.7 Discussion

Based on the evaluation conducted previously, it is feasible for CfB to monitor file activities happening in a multi-cloud storage system consisting of AWS S3, GCP Storage, and Azure Blob using the generated cloud storage log files. CfB could use either a cloud activity log, storage access log, or the correlated cloud storage log to monitor the events in a multi-cloud storage environment depending on the use case. The storage access log is more suitable for monitoring cloud object storage services, while the cloud activity log is more suitable for monitoring the whole CSP environment. The correlated cloud storage log could be used to solve the missing information from each log. Nevertheless, the storage access log is the preferable log choice for CfB since it contains sufficient information needed for monitoring and analysis purposes.

However, several issues of the cloud object storage services, its logging functionality, and the cloud storage log files are faced by the CfB to monitor file activities on multiple CSPs using the multi-cloud file storage monitoring system.

The cloud storage log files from AWS, GCP, and Azure have **different information quality** as several logs do not have the log fields expected to be available to monitor activities on cloud object storage services. The log files also might contain **incomplete and inconsistent information** of the events happening in the cloud object storage services, especially AWS CloudTrail's cloud activity log and GCP Logging's cloud activity log. There might be missing information from either the cloud activity log or the storage access log that would require both logs to be correlated to fill the information gap needed for further investigations and processes.

The cloud storage log files from multiple CSPs are delivered **inconsistently** and **unpredictably** where the log files might be available **up to 2 hours** after the actual events happened on the cloud object storage services, depending on the CSP. This could make real-time multi-cloud monitoring and analysis using cloud storage log files to be infeasible as the CfB needs to regularly check and retrieve newly generated log files on the sink buckets in various CSPs before processing and analyzing the log files.

The CSPs could not guarantee the completeness and the timeliness of the generated cloud storage log files to record all events in the cloud object storage services. For example, AWS S3's server access log entry might be delivered a couple of hours after the request is happening or it might not be delivered at all [Ama20a]. AWS, GCP, and Azure also intentionally **do not record several unauthorized and unauthenticated requests** as proven previously [Ama20f; Mic20b]. Meanwhile, the generated cloud storage log files could not provide non-repudiation property as it is vulnerable to tamper [YWH17]. For example, the log files could be tampered with by malicious CSP administrators to hide the evidence that they access unauthorized files stored on the cloud.

Non-existent request ID value on the CfB system log entry, which is not provided by the response of the executed requests to the CSPs, could create an **information gap** issue for CfB as it is difficult to directly correlate the information of executed requests from the CfB systems with the cloud storage log files. The correlation could still be done using multiple similar log fields available on both logs to generate a one-to-many result. Although one-to-one log correlation results could be achieved by selecting the correlated log entry with the minimum timestamp difference of both logs, the result might not be accurate especially for burst activities in a short time.

The combination of inconsistent and incomplete information in the cloud storage log and unpredictable log delivery time could create **reliability and security issues** for CfB and companies as CfB customers. An attacker could try accessing the files stored on the CfB's file buckets on multiple CSPs by exploiting the misconfiguration on the file buckets. CfB will be unable to detect the attacks in real-time due to the time gap between the actual attacks and the cloud storage log files to be retrieved by CfB's multi-cloud file storage monitoring system. Once the cloud storage log files have been collected and processed, the actual attacks might not even be recorded in the log files by the CSPs, which hinders the CfB to investigate the attack. The companies might then be unaware that there might be possible data leaks happening as the CfB does not inform the undetected attacks happening on the cloud to the companies.

Certain steps could be taken by CfB to minimize the issues with the file activity monitoring on multiple CSPs. CfB should rely on the system log entries as the source of truth to monitor activities of CfB user's files on the cloud. All possible information of the events happening on the system should be recorded to increase the possibility of accurate correlation with cloud storage log files, including the executed signed URL requests and received responses from the

CSPs. CfB also needs to generate an identifier value that appears on the cloud storage log entries to achieve a one-to-one log correlation result. There are two possible solutions that could be implemented by CfB to achieve one-to-one log correlation result between cloud storage log files with CfB system log entries:

- **Signed URL**: The signed URL generated by the CfB system could be used to correlate the cloud storage log files with CfB system log entries. The signature appended on the signed URL is logged anonymized as in the case of AWS S3's server access log and Azure Blob's cloud storage log as proven previously. CfB's multi-cloud file storage monitoring system then needs to parse and anonymize the signature of the generated signed URL to match the request URI value recorded on the cloud storage log entry.

  Another option is for the CfB to generate the signed URLs with customized parameters and values that could be used to help accurately correlate the log entries. Other available parameters on the signed URL, such as timestamp or credential, could not be used as an identifier for correlating with the CfB system log entries. CfB could add system request ID or requester's CfB user ID to the generated signed URLs that will be executed by the authorized CfB users. The cloud storage log entries record the additional customized parameters as clear text where the multi-cloud file storage monitoring system could match the request ID and/or CfB user ID values on the CfB system log entry with the cloud storage log entry.

  The signed URL is limited only to the object activities, which means it does not cover other possible request types in the cloud object storage services using API, such as bucket configuration change. Any requests logged on the cloud storage log files outside of the executed signed URL requests will be further investigated to determine whether it is malicious or benign. Nonetheless, the method only works for cloud storage log *Request URI* field, such as AWS S3's server access log, GCP Storage, and Azure Blob. Some CSPs, e.g., Azure Blob service, might not provide the customized signed URL parameter functionality, therefore, making the CfB be unable to add additional identifier values for accurate correlation.

- **Customized user agent**: CfB could set the customized user agent to the CfB system, CfB users, and authorized CfB employees while sending the requests to the CSPs. The user agent should contain identifying information that could be used to correlate the CfB system log entry with cloud storage log entries while distinguishable from other unknown or unauthorized requesters. This is due to the *User Agent* field is always available on both cloud storage log types from AWS, GCP, and Azure where the cloud storage log entry will record the full customized user agent.

This solution would require the CfB user's client application to append the user ID value to the user agent as the user executes the request to multiple CSPs. The CfB could also send the CfB system's request ID for every CfB user's file request where the client application then appends the request ID to the user agent. CfB's multi-cloud file storage monitoring system then could match the request ID and/or user ID value on the CfB system log entry with the *User Agent* value on the cloud storage log entries to generate accurate one-to-one correlation results.

The disadvantage of this method is the CfB needs to manage and ensure the client application has manually appended the user ID value to the user agent before it executes the signed URLs using the HTTP client. If the CfB decides to add the request ID value, the CfB then needs to routinely update the request ID value for every file request, which will increase the management complexity.

Although CfB could only be done so much to optimize the file activity monitoring process on multiple CSPs, the biggest obstacle lies in the CSPs as they do not fully uphold the shared responsibility model to provide a complete monitoring process on the cloud object storage services. Interestingly, the CSPs do not recommend using generated cloud storage log files for monitoring the activities of buckets and objects in the cloud object storage services as the cloud storage log files should not be used as a complete accounting of all requests to the cloud object storage services [Ama20a; Goo20a].

Therefore, several major aspects are proposed for the CSPs to improve on the cloud object storage services to allow CfB to monitor the file activities on the cloud. The CSPs must ensure the generated cloud storage log files record complete and consistent information of all request types happening on the cloud object storage services, such as recording unauthenticated and unauthorized requests. Standardized cloud logging format could be implemented to help cloud customers process and correlate log information from different CSPs, e.g., the proposed unified storage access log and unified cloud activity log format. The generated cloud storage log files should be available as soon as the CSPs have processed the requests to allow (near) real-time cloud monitoring. Finally, the CSPs should provide request ID, response code, and response message as part of the response information from the executed requests to allow easy and accurate cloud storage log correlation with the CfB system log.

6.8 Conclusion and Future Works

In this Chapter, a multi-cloud file storage monitoring system is proposed to provide file activity monitoring functionality across multiple CSPs for CloudRAID for Business. The system collects, processes, and analyzes cloud storage log files generated from the activities on AWS S3, GCP Storage, and Azure Blob services with CfB system log entries. Unified storage access log and unified cloud activity log formats are proposed to solve the heterogeneity of cloud storage log files from multiple CSPs and simplify the information to a single format. Although monitoring file activities on the CfB's multi-cloud storage system with cloud storage log files is feasible, there might be reliability and security issues arise that affecting CfB's multi-cloud storage system due to characteristics and behaviors of cloud storage log files. Cloud storage log files might be delivered unpredictably and inconsistently up to a couple of hours where it might not record several unauthenticated and unauthorized requests made to the cloud object storage services. CfB system log entries could not be directly correlated with the cloud storage log files due to inconsistent and incomplete event information that creates an information gap for CfB and the company's administrator. The CSPs then need to improve the logging functionality of the cloud object storage services to ensure that CfB and the companies as CfB customers could better monitor file activities on the cloud.

The proposed multi-cloud file storage monitoring system could be used to monitor the activities on multi-CSP environments by utilizing generated cloud activity log files from various CSPs. The information from cloud resource management processes explained in Chapter 5 could be correlated with the information gathered from cloud activity log files to provide a holistic multi-cloud management process, e.g., detect changes happening in cloud resources or investigate suspicious activities on the CSP environment. Finally, big data architecture could be implemented for multi-cloud file storage monitoring systems to improve the performance and the scalability of the system given the increasing size of cloud storage log files.

7 Conclusion

Enterprise file synchronization and share (EFSS) provides the solution for companies to store their confidential files on the cloud and easy file sharing and collaboration between the employees. With the increasing number of cyberattacks on the cloud and data breaches over the past few years, the EFSS systems are then responsible to manage the company's files stored on the cloud and ensures only authorized entities in the company's domain could access the files.

This thesis resolves some of the challenges faced by EFSS systems to provide secure and scalable enterprise cloud storage solution for companies, particularly from the perspective of CloudRAID for Business (CfB). The CfB system is developed based on the concept of CloudRAID, a secure personal cloud storage research project aiming to provide data confidentiality and availability on the cloud by combining cryptographic and erasure techniques to store the files as multiple encrypted file chunks across various cloud service providers (CSPs). It focuses on key management system, location-based file access control, multi-cloud storage resource management, and cloud file access monitoring aspects of the CloudRAID concept, which are needed by an EFSS system to securely store, manage, and share company's confidential files for its authorized employees. The contributions of this thesis could be summarized as follows.

A scalable and secure key management system based on a multi-authority attribute-based encryption (MA-ABE) scheme is introduced in Chapter 3 to replace the RSA-based key management system used to manage the cryptographic keys used for secure file operations in the system. The multi-authority attribute-based encryption scheme allows the CfB to provide secure and scalable file sharing and file-level security of the CfB user's files within the company's domain and between companies. It could generate one encrypted file key per file for multiple CfB users and their devices due to the attribute-based encryption's "one-to-many property" instead of generating multiple encrypted file keys that helps to reduce the number and size of encrypted file keys to a minimum. It also provides attribute-based file-level access control as only the authorized CfB users with the correct attributes that fulfill the file-sharing specification could decrypt the encrypted file key. The company could securely and scalable manage its confidential files and employees in its domain without any interference from the CfB system to ensure a zero-knowledge policy within the system.

Chapter 4 proposes Internet-based location access control functionality to provide system-level security by ensuring only authorized CfB users at the predetermined trusted location could access the files. The IP address, the delay measurement results with known landmark servers, and surrounding Wi-Fi access points of Internet-connected devices used by the CfB users can be used to determine and verify the user's location during the file access request. Seventeen virtual machines deployed in the various CSPs and 200 servers randomly selected from the Speedtest network across the European region are used to calculate the latency, which could be used to calculate the location of the users using Constraint-based Geolocation and GeoWeight delay-based geolocation algorithms. Third-party open source intelligent services are also used to provide additional location information of the users based on the IP address and the surrounding Wi-Fi access points. Based on the evaluation, Internet-based location could be used as a location information input for location-based access control with country-level accuracy.

A unified multi-cloud storage resource management framework is presented in Chapter 5 to provide cloud-level security for authorized CfB stakeholders by securely managing the owned cloud storage resources across multiple CSPs to be only accessible by authorized CfB stakeholders. The unified cloud storage resource model is proposed to resolve the data model heterogeneity of cloud storage resources and cloud access control model to store and manage the global state of cloud storage resources from different CSPs in a single format. A unified multi-cloud storage resource management platform allows CfB to securely, centrally, and automatically manage the cloud storage resources across multiple CSPs in a single platform. It is capable of discovering, provisioning, monitoring, and assessing the cloud storage resources to ensure the CfB system and its multi-cloud storage environment are secure and work normally. The guidelines and instructions are implemented in the unified platform and the CfB system to ensure only authorized CfB stakeholders could access the cloud storage resources following their roles in the system.

Chapter 6 explains the mechanisms to monitor and analyze the activities of CfB user's files on the cloud by automatically collecting, processing, and correlating storage access log files and cloud activity log files from various CSPs in its multi-cloud storage environment with the CfB system log entries. Although it is feasible to monitor cloud file activities using cloud storage log files, there are reliability and security concerns since the generated log files are generated unpredictable with inconsistent and incomplete information where it might not record several types of unauthenticated and unauthorized activities on the cloud object storage services. Overall, the CSPs need to improve several aspects of the cloud object storage services and their logging functionality that would help CfB to better monitor file activities on the cloud.

In summary, the proposed security approaches in this thesis allow CloudRAID for Business to achieve similar data confidentiality and data availability in CloudRAID. It provides holistic security for cloud storage resources used and owned by the CfB and the company's confidential files on the cloud and within the system. The zero-knowledge policy could be achieved by CfB for its customers and users as only the authorized companies and their employees should be able to manage and access the files stored on the cloud. Other authorized CfB stakeholders should also have limited access to the available cloud storage resources following their roles in the CfB's domain. The approaches explained in this thesis could also be used for other purposes and similar services. For example, the Internet-based location access control could be implemented for location-based services to ensure only authorized users in the designated locations could access the resources. The unified multi-cloud storage management framework could be used automatically and centrally manage the cloud resources for services relying on a multi-cloud environment.

With CfB system is utilized as Software-as-a-Service, companies interested to utilize the CfB system to manage their confidential files for their employees are only required to upload the files to the multi-cloud environment managed by the CfB system without the need to manage their files on the cloud or . They also need to upload the information of the employees and company's organizational structure, which can be retrieved from the company's internal record system (e.g., Microsoft ActiveDirectory or other Lightweight Directory Access Protocol (LDAP) systems), to ensure only authorized employees could access the files and the system.

There are several aspects of the CloudRAID for Business that could be considered for future work to secure the files on the cloud and the system.

- CfB could implement an MA-ABE scheme better than the PAD-TFDAC-MACS scheme used in its key management system to achieve more flexible, scalable, and secure file-sharing between the companies and their employees. The scheme should support a more complicated and flexible policy to generate a small-sized encrypted file key accessible across multiple company's domains. It should also allow the company to better securely manage their files and their employees with a flexible revocation scheme without any intervention from CfB.

System-level attribute-based access control (ABAC) could also be integrated with file-level attribute-based access control to achieve holistic ABAC in the CfB. ABAC system supporting eXtensible Access Control Markup Language (XACML) standard could be integrated into the CfB system where it will store and manage the file-sharing restriction used in the file key encryption and decryption processes as the policy documents. The ABAC system will evaluate the CfB user's attributes first during file access requests and grant access to the files only for authorized users with the correct attributes fulfilling the file-sharing restriction.

- Various delay-based geolocation schemes could be used by CfB to improve the accuracy, privacy, and efficiency of the user location calculation process using the delay measurement results between the users and active landmarks. Location information provided by the sensors in the surrounding area, e.g., Bluetooth Low Energy, could be used for location calculation process in indoor or limited area settings, e.g., in a building or a park. It would help the CfB to correctly determine the location of users and grant access to the files only to authorized users at the correct locations.

  Another variant of XACML called Geospatial XACML (GeoXACML) could also be implemented to support the Internet-based location access control functionality. The location information is embedded as part of the attribute in the managed policy documents where the CfB could better manage the location information to determine if the user is authorized to access the files based on the location.

- The unified multi-cloud storage management framework could be expanded by including various cloud resource types from different CSPs to securely manage cloud resources in a centralized environment, such as virtual machines and databases in Microsoft Azure and OpenStack. The unified framework could utilize a graph database with a unified cloud storage resource model to store the information of the cloud resources and their configurations, including the cloud access control model, and model the relationship between the cloud resources. The graph database could be used to compare and analyze the states of cloud resources to detect any changes or misconfigurations in the cloud resources.

Bibliography

[Ace+13] Giuseppe Aceto, Alessio Botta, Walter De Donato, and Antonio Pescapè. **Cloud monitoring: A survey**. *Computer Networks* 57:9 (2013), 2093–2115 (see page 145).

[Ahm+17] Adnan Ahmed, Zubair Shafiq, Harkeerat Bedi, and Amir Khakpour. **Peering vs. transit: Performance comparison of peering and transit interconnections**. In: *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE. 2017, 1–10 (see pages 98, 99).

[AKK10a] Mohammed Jubaer Arif, Shanika Karunasekera, and Santosh Kulkarni. **GeoWeight: Internet host geolocation based on a probability model for latency measurements**. In: *Proceedings of the Thirty-Third Australasian Conferenc on Computer Science-Volume 102*. 2010, 89–98 (see page 80).

[AKK10b] Mohammed Jubaer Arif, Shanika Karunasekera, and Santosh Kulkarni. "GeoWeight: internet host geolocation based on a probability model for latency measurements." In: *Proceedings of the Thirty-Third Australasian Conferenc on Computer Science-Volume 102*. 2010, 89–98 (see pages 74, 81–83, 99).

[Alm+11] Abdulrahman Almutairi, Muhammad Sarfraz, Saleh Basalamah, Walid Aref, and Arif Ghafoor. **A distributed access control architecture for cloud computing**. *IEEE software* 29:2 (2011), 36–44 (see page 118).

[Alq+16] Saad Alqahtany, Nathan Clarke, Steven Furnell, and Christoph Reich. **A forensic acquisition and analysis system for IaaS**. *Cluster Computing* 19:1 (2016), 439–453 (see page 146).

[Ama17] Amazon Web Services. *Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region*. https://aws.amazon.com/message/41926/. (Accessed on 06/04/2021). 2017 (see page 13).

[Ama20a] Amazon Web Services. *Amazon S3 server access logging*. https://docs.aws.amazon.com/AmazonS3/latest/dev/ServerLogs.html. (Accessed on 06/09/2020). 2020 (see pages 150, 158, 170, 172).

[Ama20b] Amazon Web Services. *Amazon S3 Simple Storage Service Pricing*. https://aws.amazon.com/s3/pricing/. (Accessed on 10/15/2020). 2020 (see page 11).

[Ama20c] Amazon Web Services. *How CloudTrail Works.* https://docs.aws.amazon.com/awscloudtrail/latest/userguide/how-cloudtrail-works.html. (Accessed on 11/16/2020). 2020 (see pages 150, 158).

[Ama20d] Amazon Web Services. *Hybrid Cloud with AWS.* https://d1.awsstatic.com/whitepapers/hybrid-cloud-with-aws.pdf?did=wp_card&trk=wp_card. (Accessed on 06/09/2021). 2020 (see pages 113, 143).

[Ama20e] Amazon Web Services. *Logging Amazon S3 API calls using AWS CloudTrail.* https://docs.aws.amazon.com/AmazonS3/latest/dev/cloudtrail-logging.html. (Accessed on 06/18/2020). 2020 (see pages 150, 162).

[Ama20f] Amazon Web Services. *Logging with Amazon S3.* https://docs.aws.amazon.com/AmazonS3/latest/dev/logging-with-S3.html. (Accessed on 09/15/2020). 2020 (see page 170).

[Ama20g] Amazon Web Services. *Share an object with others.* https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html. (Accessed on 12/07/2020). 2020 (see pages 12, 135).

[Ama20h] Amazon Web Services. *Shared Responsibility Model.* https://aws.amazon.com/compliance/shared-responsibility-model/. (Accessed on 11/19/2020). 2020 (see pages 111, 114, 115, 141).

[Ama21a] Amazon Web Services. *Amazon S3 FAQs.* https://aws.amazon.com/s3/faqs/. (Accessed on 08/12/2021). 2021 (see page 12).

[Ama21b] Amazon Web Services. *Amazon Simple Storage Service (S3).* https://aws.amazon.com/s3/faqs/. (Accessed on 06/04/2021). 2021 (see page 110).

[Ama21c] Amazon Web Services. *AWS CloudTrail FAQs.* https://aws.amazon.com/cloudtrail/faqs/. (Accessed on 07/14/2021). 2021 (see page 129).

[Ama21d] Amazon Web Services. *Network Peering.* https://aws.amazon.com/peering/. (Accessed on 01/26/2021). 2021 (see page 99).

[Ama21e] Amazon Web Services. *Policies and permissions in IAM.* https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html. (Accessed on 07/01/2021). 2021 (see pages 121, 122).

[Ama21f] Amazon Web Services. *Sharing an object with a presigned URL.* https://docs.aws.amazon.com/AmazonS3/latest/userguide/ShareObjectPreSignedURL.html. (Accessed on 06/16/2021). 2021 (see page 12).

[Ama21g] Amazon Web Services. *Viewing Events with CloudTrail Event History.* https://docs.aws.amazon.com/awscloudtrail/latest/userguide/view-cloudtrail-events.html. (Accessed on 05/11/2021). 2021 (see pages 128, 146, 148, 150).

[Ama21h] Amazon Web Services. *What is ABAC for AWS?* https://docs.aws.amazon.com / IAM / latest / UserGuide / introduction _ attribute - based - access - control.html. (Accessed on 07/01/2021). 2021 (see page 120).

[AMV17] AbdelRahman Abdou, Ashraf Matrawy, and Paul C Van Oorschot. **Accurate manipulation of delay-based internet geolocation**. In: *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 2017, 887–898 (see pages 74, 80, 97).

[APW10] Hussam Abu-Libdeh, Lonnie Princehouse, and Hakim Weatherspoon. **RACS: a case for cloud storage diversity**. In: *Proceedings of the 1st ACM symposium on Cloud computing*. 2010, 229–240 (see pages 110, 111).

[Ard+06] Claudio A Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. **Supporting location-based conditions in access control policies**. In: *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. 2006, 212–222 (see pages 67, 71).

[Ard+08] Claudio Agostino Ardagna, Marco Cremonini, S De Capitani di Vimercati, and Pierangela Samarati. "Privacy-enhanced location-based access control." In: *Handbook of Database Security*. Springer, 2008, 531–552 (see page 70).

[Asl+17] Sidra Aslam, Saif ul Islam, Abid Khan, Mansoor Ahmed, Adnan Akhundzada, and Muhammad Khurram Khan. **Information collection centric techniques for cloud resource management: Taxonomy, analysis and challenges**. *Journal of Network and Computer Applications* 100 (2017), 80–94 (see pages 114, 122).

[Ber+16] Stefan Berger, Shelly Garion, Yosef Moatti, Dalit Naor, D Pendarakis, Alexandra Shulman-Peleg, Josyula R Rao, Enriquillo Valdez, and Yaron Weinsberg. **Security intelligence for cloud management infrastructures**. *IBM Journal of Research and Development* 60:4 (2016), 11–1 (see page 143).

[Bes+13] Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando André, and Paulo Sousa. **DepSky: dependable and secure storage in a cloud-of-clouds**. *Acm transactions on storage (tos)* 9:4 (2013), 1–33 (see page 111).

[BHC18] Yaser Baseri, Abdelhakim Hafid, and Soumaya Cherkaoui. **Privacy preserving fine-grained location-based access control for mobile cloud**. *computers & security* 73 (2018), 249–265 (see page 68).

[Bjo+18] Mathias Bjorkqvist, Christian Cachin, Felix Engelmann, and Alessandro Sorniotti. **Scalable Key Management for Distributed Cloud Storage**. In: *2018 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE. 2018, 250–256 (see page 31).

[Box20a] Boxcryptor. *Boxcryptor Key Management explained*. https://www.boxcryptor.com/en/key-management/. (Accessed on 03/25/2020). 2020 (see pages 34, 35).

[Box20b] Boxcryptor. *Technical Overview – How Boxcryptor's Encryption Works*. https://www.boxcryptor.com/en/technical-overview/. (Accessed on 03/06/2020). 2020 (see pages 34, 35).

[Box21a] Boxcryptor. *Introduction*. https://www.boxcryptor.com/en/help/introduction/windows/. (Accessed on 08/18/2021). 2021 (see page 29).

[Box21b] Boxcryptor. *Manage Your Clouds and Locations in Boxcryptor*. https://www.boxcryptor.com/en/help/manage-cloud-providers-and-locations/windows/. (Accessed on 06/09/2021). 2021 (see page 113).

[Box21c] Boxcryptor. *Teams*. https://www.boxcryptor.com/en/help/teams/windows/. (Accessed on 05/27/2021). 2021 (see pages 29, 69, 144).

[BPJ15] Nathalie Baracaldo, Balaji Palanisamy, and James Joshi. **Geo-social-RBAC: A location-based socially aware access control framework**. In: *International Conference on Network and System Security*. Springer. 2015, 501–509 (see page 67).

[Bri+14] Bob Briscoe, Anna Brunstrom, Andreas Petlund, David Hayes, David Ros, Jyh Tsang, Stein Gjessing, Gorry Fairhurst, Carsten Griwodz, and Michael Welzl. **Reducing internet latency: A survey of techniques and their merits**. *IEEE Communications Surveys & Tutorials* 18:3 (2014), 2149–2196 (see page 97).

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. **Ciphertext-policy attribute-based encryption**. In: *2007 IEEE symposium on security and privacy (SP'07)*. IEEE. 2007, 321–334 (see pages 38, 42).

[But18] Artjom Butyrtschik. *jTR-ABE*. https://github.com/TU-Berlin-SNET/jTR-ABE. 2018. (Visited on 01/14/2020) (see pages 31, 42, 47).

[Car+17] Carlos André Batista de Carvalho, Nazim Agoulmine, Miguel Franklin de Castro, and Rossana Maria de Castro Andrade. **How to improve monitoring and auditing security properties in cloud storage?** In: *Anais do XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC. 2017 (see pages 141, 142, 145).

[CC09] Melissa Chase and Sherman SM Chow. **Improving privacy and security in multi-authority attribute-based encryption**. In: *Proceedings of the 16th ACM conference on Computer and communications security*. ACM. 2009, 121–130 (see page 40).

[Cel+16] Antonio Celesti, Maria Fazio, Massimo Villari, and Antonio Puliafito. **Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems**. *Journal of Network and Computer Applications* 59 (2016), 208–218. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2014.09.021. URL: https://www.sciencedirect.com/science/article/pii/S1084804514002288 (see page 112).

[Cel+19] Antonio Celesti, Antonino Galletta, Maria Fazio, and Massimo Villari. **Towards hybrid multi-cloud storage systems: Understanding how to perform data transfer**. *Big Data Research* 16 (2019), 1–17 (see page 112).

[Cha07] Melissa Chase. **Multi-authority attribute based encryption**. In: *Theory of Cryptography Conference*. Springer. 2007, 515–534 (see pages 39, 40, 48).

[Che+17] Liang Chen, Sarang Thombre, Kimmo Järvinen, Elena Simona Lohan, Anette Alén-Savikko, Helena Leppäkoski, M Zahidul H Bhuiyan, Shakila Bu-Pasha, Giorgia Nunzia Ferrara, Salomon Honkala, et al. **Robustness, security and privacy in location-based services for future IoT: A survey**. *IEEE Access* 5 (2017), 8956–8977 (see pages 66, 71, 72, 108).

[Cho+16] Min Choi, Jungha Lee, Sungho Kim, Young-Sik Jeong, and Jong-Hyuk Park. **Location based authentication scheme using BLE for high performance digital content management system**. *Neurocomputing* 209 (2016), 25–38 (see pages 66, 68).

[Chu+14] Cheng-Kang Chu, Sherman SM Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H Deng. **Key-aggregate cryptosystem for scalable data sharing in cloud storage**. *IEEE transactions on parallel and distributed systems* 25:2 (2014), 468–477 (see page 31).

[CIC14] Ramaswamy Chandramouli, Michaela Iorga, and Santosh Chokhani. “Cryptographic key management issues and challenges in cloud services.” In: *Secure Cloud Computing*. Springer, 2014, 1–30 (see pages 30, 37).

[Clo19] Cloud Security Alliance. *Top Threats to Cloud Computing: The Egregious 11*. 2019. URL: https://cloudsecurityalliance.org/download/artifacts/top-threats-to-cloud-computing-egregious-eleven/ (visited on 06/06/2019) (see pages 115, 117).

[CLY17] Long Cheng, Fang Liu, and Danfeng Yao. **Enterprise data breach: causes, challenges, prevention, and future directions**. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7:5 (2017), e1211 (see pages 26, 30, 37).

[Con+18] Andrea Continella, Mario Polino, Marcello Pogliani, and Stefano Zanero. **There's a Hole in that Bucket! A Large-scale Analysis of Misconfigured S3 Buckets**. In: *Proceedings of the 34th Annual Computer Security Applications Conference*. 2018, 702–711 (see pages 141, 148).

[Cos20] Samantha Cossick. *What's the Difference Between Residential and Business Internet?* https://www.allconnect.com/blog/residential-or-business-internet-for-small-business. (Accessed on 04/05/2021). 2020 (see page 97).

[Dec08] Michael Decker. **Requirements for a location-based access control model**. In: *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*. 2008, 346–349 (see pages 66, 70).

[DFK15] Lucia De Marco, Filomena Ferrucci, and Tahar Kechadi. **SLAFM: A Service Level Agreements Formal Model for Cloud Computing**. In: *The 5th International Conference on Cloud Computing and Service Science (CLOSER 2015), Lisbon, Portugal, 20-22 May 2015*. 2015 (see page 142).

[Di +17] Riccardo Di Pietro, Marco Scarpa, Maurizio Giacobbe, and Antonio Puliafito. **Secure storage as a service in multi-cloud environment**. In: *International Conference on Ad-Hoc Networks and Wireless*. Springer. 2017, 328–341 (see page 112).

[Dro20] Dropbox. *Dropbox Business Security Whitepaper*. https://aem.dropbox.com/cms/content/dam/dropbox/www/en-us/business/solutions/solutions/dfb_security_whitepaper.pdf. (Accessed on 03/11/2021). 2020 (see pages 32, 33, 69, 143, 144).

[Dro21a] Dropbox. *I'm not able to access Dropbox in Crimea, North Korea, or Syria. Why, and what should I do?* https://help.dropbox.com/accounts-billing/security/restricted-access-region-country. (Accessed on 03/11/2021). 2021 (see page 69).

[Dro21b] Dropbox. *Monitor team sharing activity*. https://help.dropbox.com/teams-admins/admin/monitor-sharing-activity. (Accessed on 04/30/2021). 2021 (see page 143).

[Dro21c] Dropbox. *Security alerts for Dropbox Business*. https://help.dropbox.com/teams-admins/admin/security-alerts. (Accessed on 05/26/2021). 2021 (see page 144).

[Dro21d] Dropbox. *View team activity in the admin console.* https://help.dropbox.com/teams-admins/admin/view-activity. (Accessed on 05/26/2021). 2021 (see page 143).

[Dro21e] Dropbox Business. *Dropbox plan comparison.* https://www.dropbox.com/business/plans-comparison. (Accessed on 08/17/2021). 2021 (see page 29).

[DS19] A Augustus Devarajan and T SudalaiMuthu. **Cloud storage monitoring system analyzing through file access pattern**. In: *2019 International Conference on Computational Intelligence in Data Science (ICCIDS).* IEEE. 2019, 1–6 (see pages 142, 143).

[EFP17] Divyaa Manimaran Elango, Frank Fowley, and Claus Pahl. **An ontology-based architecture for an adaptable cloud storage broker**. In: *European Conference on Service-Oriented and Cloud Computing.* Springer. 2017, 86–101 (see page 112).

[Eme21] Emergen Research. *Cloud Object Storage Market Size to Reach USD 13.65 Billion.* https://www.globenewswire.com/fr/news-release/2021/05/25/2235619/0/en/Cloud-Object-Storage-Market-Size-to-Reach-USD-13-65-Billion-in-2028-Durability-Scalability-Compliance-Security-Faster-Data-Retrieval-are-Some-Factors-that-will-Drive-Industry-Growt.html. (Accessed on 06/04/2021). 2021 (see page 110).

[Eri+12] Brian Eriksson, Paul Barford, Bruce Maggs, and Robert Nowak. **Posit: a lightweight approach for IP geolocation**. *ACM SIGMETRICS Performance Evaluation Review* 40:2 (2012), 2–11 (see page 108).

[Fac+13] Michael Factor, David Hadas, Aner Harnama, Nadav Har'El, Elliot K Kolodner, Anil Kurmus, Alexandra Shulman-Peleg, and Alessandro Sorniotti. **Secure logical isolation for multi-tenancy in cloud storage**. In: *2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST).* IEEE. 2013, 1–5 (see pages 28, 118).

[Fat+14] Kaniz Fatema, Vincent C Emeakaroha, Philip D Healy, John P Morrison, and Theo Lynn. **A survey of cloud monitoring tools: Taxonomy, capabilities and objectives**. *Journal of Parallel and Distributed Computing* 74:10 (2014), 2918–2933 (see page 145).

[Fed20] Federal Aviation Administration. *GPS - How It Works.* https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks. (Accessed on 01/07/2021). 2020 (see page 71).

[FL93] Walter Fumy and Peter Landrock. **Principles of key management**. *IEEE Journal on selected areas in communications* 11:5 (1993), 785–793 (see page 30).

[Fle21] Flexera. *2021 State of the Cloud Report*. https://info.flexera.com/CM-REPORT-State-of-the-Cloud. (Accessed on 07/22/2021). 2021 (see page 1).

[FSK05] Bryan Ford, Pyda Srisuresh, and Dan Kegel. **Peer-to-Peer Communication Across Network Address Translators.** In: *USENIX Annual Technical Conference, General Track*. 2005, 179–192 (see page 78).

[FTA16] Massimo Ficco, Luca Tasquier, and Rocco Aversa. **Intrusion detection in federated clouds**. *International Journal of Computational Science and Engineering* 13:3 (2016), 219–232 (see page 117).

[Gar+17] Shelly Garion, Hillel Kolodner, Allon Adir, Ehud Aharoni, and Lev Greenberg. **Big data analysis of cloud storage logs using spark**. In: *Proceedings of the 10th ACM International Systems and Storage Conference*. 2017, 1–1 (see pages 142, 146).

[Gar19] Gartner. *Worldwide Public Cloud Revenue to Grow 17% in 2020*. https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020. (Accessed on 07/29/2021). 2019 (see page 1).

[Gha+17] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. **A look at router geolocation in public and commercial databases**. In: *Proceedings of the 2017 Internet Measurement Conference*. 2017, 463–469 (see pages 72, 73, 106).

[Goo20a] Google Cloud Platform. *Access logs & storage logs*. https://cloud.google.com/storage/docs/access-logs. (Accessed on 06/05/2020). 2020 (see pages 150, 158, 172).

[Goo20b] Google Cloud Platform. *Cloud Audit Logs with Cloud Storage*. https://cloud.google.com/storage/docs/audit-logs. (Accessed on 06/23/2020). 2020 (see page 150).

[Goo20c] Google Cloud Platform. *Cloud Storage pricing*. https://cloud.google.com/storage/pricing. (Accessed on 10/15/2020). 2020 (see page 11).

[Goo20d] Google Cloud Platform. *Exporting logs with the Google Cloud Console*. https://cloud.google.com/logging/docs/export/configure_export_v2. (Accessed on 11/16/2020). 2020 (see pages 129, 151, 158).

[Goo20e] Google Cloud Platform. *Signed URLs*. https://cloud.google.com/storage/docs/access-control/signed-urls. (Accessed on 12/07/2020). 2020 (see pages 13, 135).

[Goo20f] Google Cloud Platform. *Storing logs*. https://cloud.google.com/logging/docs/storage. (Accessed on 12/07/2020). 2020 (see pages 128, 146, 148, 151).

[Goo21a] Google Cloud Platform. *IAM roles for Cloud Storage*. https://cloud.google.com/storage/docs/access-control/iam-roles. (Accessed on 07/12/2021). 2021 (see page 134).

[Goo21b] Google Cloud Platform. *Overview Cloud IAM Documentation*. https://cloud.google.com/iam/docs/overview. (Accessed on 07/01/2021). 2021 (see page 121).

[Goo21c] Google Cloud Platform. *Signed URLs*. https://cloud.google.com/storage/docs/access-control/signed-urls. (Accessed on 06/16/2021). 2021 (see page 12).

[Goo21d] Google Cloud Platform. *Storage classes*. https://cloud.google.com/storage/docs/storage-classes. (Accessed on 08/12/2021). 2021 (see page 12).

[Goo21e] Google Cloud Platform. *Storage classes*. https://cloud.google.com/storage/docs/storage-classes. (Accessed on 06/04/2021). 2021 (see page 110).

[Goo21f] Google Cloud Platform. *Uploads and downloads*. https://cloud.google.com/storage/docs/uploads-downloads. (Accessed on 08/12/2021). 2021 (see page 13).

[Goy+06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. **Attribute-based encryption for fine-grained access control of encrypted data**. In: *Proceedings of the 13th ACM conference on Computer and communications security*. Acm. 2006, 89–98 (see page 38).

[Gra+15] Hendrik Graupner, Kennedy Torkura, Philipp Berger, Christoph Meinel, and Maxim Schnjakin. **Secure access control for multi-cloud resources**. In: *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*. IEEE. 2015, 722–729 (see pages 11, 12, 18, 19).

[Gra+19] Hendrik Graupner, Kennedy A Torkura, Muhammad IH Sukmana, and Christoph Meinel. **Secure Deduplication on Public Cloud Storage**. In: *Proceedings of the 2019 4th International Conference on Big Data and Computing*. 2019, 34–41 (see page 9).

[Gre16] Christian Grece. **How do films circulate on VOD services and in cinemas in the European Union**. *Observatoire européen de l'audiovisuel* (2016) (see page 71).

[Gue+06] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. **Constraint-based geolocation of internet hosts**. *IEEE/ACM Transactions On Networking* 14:6 (2006), 1219–1232 (see pages 74, 80, 81).

[HH10] Zach Hill and Marty Humphrey. **CSAL: A cloud storage abstraction layer to enable portable cloud applications**. In: *2010 IEEE Second International Conference on Cloud Computing Technology and Science*. IEEE. 2010, 504–511 (see page 112).

[HLV16] Leonard Heilig, Eduardo Lalla-Ruiz, and Stefan Voß. **A cloud brokerage approach for solving the resource management problem in multi-cloud environments**. *Computers & Industrial Engineering* 95 (2016), 16–26 (see pages 122, 123).

[Høi+16] Toke Høiland-Jørgensen, Bengt Ahlgren, Per Hurtig, and Anna Brunstrom. **Measuring latency variation in the internet**. In: *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies.* 2016, 473–480 (see page 97).

[HR16] Andy Chunliang Hsu and Indrakshi Ray. **Specification and enforcement of location-aware attribute-based access control for online social networks**. In: *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control.* 2016, 25–34 (see page 67).

[Hu+14] Han Hu, Yonggang Wen, Tat-Seng Chua, and Xuelong Li. **Toward scalable systems for big data analytics: A technology tutorial**. *IEEE access* 2 (2014), 652–687 (see pages 128, 151, 156).

[Hua+15] Wei Huang, Afshar Ganjali, Beom Heyn Kim, Sukwon Oh, and David Lie. **The state of public infrastructure-as-a-service cloud security**. *ACM Computing Surveys (CSUR)* 47:4 (2015), 1–31 (see page 11).

[Hua+18] Haosheng Huang, Georg Gartner, Jukka M Krisp, Martin Raubal, and Nico Van de Weghe. **Location based services: ongoing evolution and research agenda**. *Journal of Location Based Services* 12:2 (2018), 63–93 (see page 71).

[Hur21] Hurricane Electric Internet Services. *IP Transit.* https://he.net/ip_transit.html. (Accessed on 04/06/2021). 2021 (see page 98).

[IDG20] IDG Communications. *2020 IDG Cloud Computing Executive Summary.* https://resources.idg.com/download/2020-cloud-computing-executive-summary-rl. (Accessed on 06/07/2021). 2020 (see page 110).

[JFE16] Philipp Junghanns, Benjamin Fabian, and Tatiana Ermakova. **Engineering of secure multi-cloud storage**. *Computers in Industry* 83 (2016), 108–120 (see page 112).

[JS15] Brendan Jennings and Rolf Stadler. **Resource management in clouds: Survey and research challenges**. *Journal of Network and Systems Management* 23:3 (2015), 567–619 (see pages 114, 115).

[Kal+03] Mahesh Kallahalla, Erik Riedel, Ram Swaminathan, Qian Wang, and Kevin Fu. **Plutus: Scalable Secure File Sharing on Untrusted Storage.** In: *Fast.* Vol. 3. 2003, 29–42 (see page 31).

[Kas20] Kaspersky. *What Is a Brute Force Attack?* https://www.kaspersky.com/resource-center/definitions/brute-force-attack. (Accessed on 03/09/2020). 2020 (see page 37).

[KE16] Muhammad Kazim and David Evans. **Threat modeling for services in cloud**. In: *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. IEEE. 2016, 66–72 (see page 141).

[Kha+16] Suleman Khan, Abdullah Gani, Ainuddin Wahid Abdul Wahab, Mustapha Aminu Bagiwa, Muhammad Shiraz, Samee U Khan, Rajkumar Buyya, and Albert Y Zomaya. **Cloud log forensics: Foundations, state of the art, and future directions**. *ACM Computing Surveys (CSUR)* 49:1 (2016), 1–42 (see pages 142, 145).

[KHR18] Ari Keranen, Christer Holmberg, and Jonathan Rosenberg. *RFC 8445 - Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal*. https://tools.ietf.org/html/rfc8445. (Accessed on 03/27/2020). 2018 (see page 33).

[KS14] Maria Krotsiani and George Spanoudakis. **Continuous certification of non-repudiation in cloud storage services**. In: *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE. 2014, 921–928 (see page 112).

[KVR17] Dan Komosny, Miroslav Voznak, and Saeed Ur Rehman. **Location Accuracy of Commercial IP Address Geolocation Databases**. *Information Technology and Control* 3:46 (2017), 334 (see pages 72, 73, 106).

[Kwo+17] Hyunsoo Kwon, Changhee Hahn, Dongyoung Koo, and Junbeom Hur. **Scalable and reliable key management for secure deduplication in cloud storage**. In: *2017 IEEE 10th international conference on cloud computing (CLOUD)*. IEEE. 2017, 391–398 (see page 32).

[LAA15] Brian Lee, Abir Awad, and Mirna Awad. **Towards secure provenance in the cloud: a survey**. In: *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*. IEEE. 2015, 577–582 (see page 146).

[LCH13] Cheng-Chi Lee, Pei-Shan Chung, and Min-Shiang Hwang. **A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments.** *IJ Network Security* 15:4 (2013), 231–240 (see pages 36–38, 40).

[Li+10] Ming Li, Shucheng Yu, Kui Ren, and Wenjing Lou. **Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings**. In: *International conference on security and privacy in communication systems*. Springer. 2010, 89–106 (see page 32).

[Li+14] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick PC Lee, and Wenjing Lou. **Secure deduplication with efficient and reliable convergent key management**. *IEEE transactions on parallel and distributed systems* 25:6 (2014), 1615–1625 (see page 32).

[Li+16] Zhenyu Li, Xiaohui Wang, Ningjing Huang, Mohamed Ali Kaafar, Zhenhua Li, Jianer Zhou, Gaogang Xie, and Peter Steenkiste. **An empirical analysis of a large-scale mobile cloud storage service**. In: *Proceedings of the 2016 Internet Measurement Conference.* 2016, 287–301 (see page 143).

[Li+17] Xiaoyu Li, Shaohua Tang, Lingling Xu, Huaqun Wang, and Jie Chen. **Two-factor data access control with efficient revocation for multi-authority cloud storage systems**. *IEEE Access* 5 (2017), 393–405 (see pages 31, 40, 48–52, 60).

[Li+19] Ziwei Li, Ke Xu, Haiyang Wang, Yi Zhao, Xiaoliang Wang, and Meng Shen. **Machine-learning-based positioning: A survey and future directions**. *IEEE Network* 33:3 (2019), 96–101 (see pages 71, 108).

[Lia+17a] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla. **Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability**. In: *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID).* IEEE. 2017, 468–477 (see page 146).

[Lia+17b] Misbah Liaqat, Victor Chang, Abdullah Gani, Siti Hafizah Ab Hamid, Muhammad Toseef, Umar Shoaib, and Rana Liaqat Ali. **Federated cloud resource management: Review and discussion**. *Journal of Network and Computer Applications* 77 (2017), 87–105 (see pages 114, 115, 117).

[Lin+08] Huang Lin, Zhenfu Cao, Xiaohui Liang, and Jun Shao. **Secure threshold multi authority attribute based encryption without a central authority**. In: *International Conference on Cryptology in India.* Springer. 2008, 426–436 (see page 40).

[Liv+20] Ioana Livadariu, Thomas Dreibholz, Anas Saeed Al-Selwi, Haakon Bryhni, Olav Lysne, Steinar Bjørnstad, and Ahmed Elmokashfi. **On the Accuracy of Country-Level IP Geolocation**. In: *Proceedings of the Applied Networking Research Workshop.* 2020, 67–73 (see pages 72, 73).

[Lu+19] Bingxian Lu, Lei Wang, Jialin Liu, Wei Zhou, Linlin Guo, Myeong-Hun Jeong, Shaowen Wang, and Guangjie Han. **LaSa: location aware wireless security access control for IoT systems**. *Mobile Networks and Applications* 24:3 (2019), 748–760 (see page 68).

[LW11] Allison Lewko and Brent Waters. **Decentralizing attribute-based encryption**. In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2011, 568–588 (see page 40).

[LW16] Zhen Liu and Duncan S Wong. **Practical attribute-based encryption: traitor tracing, revocation and large universe**. *The Computer Journal* 59:7 (2016), 983–1004 (see pages 42, 47, 63).

[Mar21] MarketsandMarkets. *Enterprise File Sharing and Synchronization (EFSS) Market Size, Share and Global Market Forecast to 2026*. https://www.marketsandmarkets.com/Market-Reports/enterprise-file-sharing-and-synchronization-market-149308334.html. (Accessed on 08/17/2021). 2021 (see page 1).

[Mei+19] Michael Meinig, Muhammad IH Sukmana, Kennedy A Torkura, and Christoph Meinel. **Holistic strategy-based threat model for organizations**. *Procedia Computer Science* 151 (2019), 100–107 (see page 9).

[Mic20a] Microsoft Azure. *Azure resource logs*. https://docs.microsoft.com/en-us/azure/azure-monitor/platform/resource-logs. (Accessed on 11/16/2020). 2020 (see pages 151, 158).

[Mic20b] Microsoft Azure. *Azure Storage Analytics logging*. https://docs.microsoft.com/en-us/azure/storage/common/storage-analytics-logging?tabs=dotnet. (Accessed on 06/09/2020). 2020 (see pages 151, 158, 166, 170).

[Mic20c] Microsoft Azure. *Create diagnostic settings to send platform logs and metrics to different destinations*. https://docs.microsoft.com/en-us/azure/azure-monitor/platform/diagnostic-settings. (Accessed on 09/18/2020). 2020 (see page 151).

[Mic20d] Microsoft Azure. *Grant limited access to data with shared access signatures (SAS)*. https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview. (Accessed on 12/07/2020). 2020 (see page 13).

[Mic20e] Microsoft Azure. *Microsoft peering policy*. https://docs.microsoft.com/en-us/azure/internet-peering/policy. (Accessed on 04/07/2021). 2020 (see page 99).

[Mic21a] Microsoft Azure. *Storage account overview*. https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview. (Accessed on 06/29/2021). 2021 (see page 13).

[Mic21b] Microsoft Azure. *Understanding block blobs, append blobs, and page blobs*. https://docs.microsoft.com/en-us/rest/api/storageservices/understanding-block-blobs--append-blobs--and-page-blobs. (Accessed on 08/12/2021). 2021 (see page 13).

[MS14] Sunilkumar S Manvi and Gopal Krishna Shyam. **Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey**. *Journal of network and computer applications* 41 (2014), 424–440 (see pages 114, 118).

[MTB18] Yaser Mansouri, Adel Nadjaran Toosi, and Rajkumar Buyya. **Data storage management in cloud environments: Taxonomy, survey, and future directions**. *ACM Computing Surveys (CSUR)* 50:6 (2018), 91 (see pages 14, 110).

[Nac+17] Rekha Nachiappan, Bahman Javadi, Rodrigo N Calheiros, and Kenan M Matawie. **Cloud storage reliability for big data applications: A state of the art survey**. *Journal of Network and Computer Applications* 97 (2017), 35–47 (see pages 14, 110).

[Nos+18] Mohammad Reza Nosouhi, Shui Yu, Marthie Grobler, Yong Xiang, and Zuqing Zhu. **SPARSE: privacy-aware and collusion resistant location proof generation and verification**. In: *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2018, 1–6 (see page 68).

[OST14] Justice Opara-Martins, Reza Sahandi, and Feng Tian. **Critical review of vendor lock-in and its impact on adoption of cloud computing**. In: *International Conference on Information Society (i-Society 2014)*. IEEE. 2014, 92–97 (see page 110).

[Pad+16] Ramakrishna Padmanabhan, Amogh Dhamdhere, Emile Aben, KC Claffy, and Neil Spring. **Reasons dynamic addresses change**. In: *Proceedings of the 2016 Internet Measurement Conference*. 2016, 183–198 (see pages 72, 73).

[Par+10] Shihyon Park, Bradley Matthews, Danny D'Amours, and William J McIver Jr. **Characterizing the impacts of VPN security models on streaming video**. In: *2010 8th Annual Communication Networks and Services Research Conference*. IEEE. 2010, 152–159 (see page 94).

[Pet13] Dana Petcu. **Multi-Cloud: expectations and current approaches**. In: *Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds*. 2013, 1–6 (see pages 13, 14).

[Pet19a] Marvin Petzolt. *PAD-TFDAC-MACS*. https://github.com/Anroc/PAD-TFDAC-MACS. 2019. (Visited on 01/14/2020) (see page 49).

[Pet19b] Marvin Petzolt. **Practical Attribute-Based Encryption for Secure Cloud Storage Systems**. MA thesis. Technical University Berlin, 2019 (see pages 49, 51, 52).

[PFH03] Niels Provos, Markus Friedl, and Peter Honeyman. **Preventing Privilege Escalation**. In: *USENIX Security Symposium*. 2003 (see page 132).

[Pla+09] James S Plank, Jianqiang Luo, Catherine D Schuman, Lihao Xu, Zooko Wilcox-O'Hearn, et al. **A Performance Evaluation and Examination of Open-Source Erasure Coding Libraries for Storage.** In: *Fast*. Vol. 9. 2009, 253–265 (see page 16).

[Pla20a] Google Cloud Platform. *Creating and managing service account keys*. https://cloud.google.com/iam/docs/creating-managing-service-account-keys. (Accessed on 02/09/2020). Feb. 2020 (see page 19).

[Pla20b] Google Cloud Platform. *V4 signing process with your own program*. https://cloud.google.com/storage/docs/access-control/signing-urls-manually. (Accessed on 03/04/2020). 2020 (see page 20).

[PLS15] Ameer Pichan, Mihai Lazarescu, and Sie Teng Soh. **Cloud forensics: Technical challenges, solutions and comparative analysis**. *Digital investigation* 13 (2015), 38–57 (see pages 116, 145, 146, 153).

[Pon20] Ponemon Institute. *2020 Cost of Insider Threats Global Report*. https://www.observeit.com/wp-content/uploads/2020/04/2020-Global-Cost-of-Insider-Threats-Ponemon-Report_UTD.pdf. (Accessed on 06/15/2021). 2020 (see pages 118, 141).

[PYJ14] Liaojun Pang, Jie Yang, and Zhengtao Jiang. **A survey of research progress and development tendency of attribute-based encryption**. *The Scientific World Journal* 2014 (2014) (see pages 36, 38, 40).

[Raf+17] Ansar Rafique, Dimitri Van Landuyt, Vincent Reniers, and Wouter Joosen. **Towards an adaptive middleware for efficient multi-cloud data storage**. In: *Proceedings of the 4th Workshop on CrossCloud Infrastructures & Platforms*. 2017, 1–6 (see pages 13, 110, 112).

[Rap+07] Jonathan Raper, Georg Gartner, Hassan Karimi, and Chris Rizos. **A critical evaluation of location based services and their potential**. *Journal of Location Based Services* 1:1 (2007), 5–45 (see page 71).

[Red20] Julia Reda. "Geoblocking: At Odds with the EU Single Market and Consumer Expectations." In: *Digital Peripheries*. Springer, Cham, 2020, 81–99 (see page 71).

[Riv+20] Esteban Rivera, Lizzy Tengana, Jesús Solano, Alejandra Castelblanco, Christian López, and Martín Ochoa. **Risk-based Authentication Based on Network Latency Profiling**. In: *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*. 2020, 105–115 (see page 93).

[RR18] Pethuru Raj and Anupama Raman. "Multi-cloud management: Technologies, tools, and techniques." In: *Software-Defined Cloud Centers*. Springer, 2018, 219–240 (see pages 111, 117).

[Sal21] Salesforce. *12 Benefits of Cloud Computing and Its Advantages*. https://www.salesforce.com/products/platform/best-practices/benefits-of-cloud-computing/. (Accessed on 07/29/2021). 2021 (see pages 1, 13).

[SAM10] Maxim Schnjakin, Rehab Alnemr, and Christoph Meinel. **A security and high-availability layer for cloud storage**. In: *International Conference on Web Information Systems Engineering*. Springer. 2010, 449–462 (see pages 14, 16).

[SB+12] Szilveszter Szebeni, Levente Butty'n, et al. **Invitation-oriented TGDH: Key management for dynamic groups in an asynchronous communication model**. In: *2012 41st International Conference on Parallel Processing Workshops*. IEEE. 2012, 269–276 (see page 34).

[SBL17] Szilveszter Szebeni, Levente Buttyán, and István Lám. *Method and system for handling of group sharing in a distributed data storage, particularly in P2P environment*. US Patent 9,563,783. 2017 (see page 33).

[SC21] Ashish Singh and Kakali Chatterjee. "LoBAC: A Secure Location-Based Access Control Model for E-Healthcare System." In: *Advances in Machine Learning and Computational Intelligence*. Springer, 2021, 621–628 (see page 68).

[Sch+13] Maxim Schnjakin, Dimitri Korsch, Martin Schoenberg, and Christoph Meinel. **Implementation of a secure and reliable storage above the untrusted clouds**. In: *2013 8th International Conference on Computer Science & Education*. IEEE. 2013, 347–353 (see pages 14, 16).

[SD03] Logan Scott and Dorothy E Denning. **A location based encryption technique and some of its applications**. In: *Proceedings of the 2003 National Technical Meeting of The Institute of Navigation*. 2003, 734–740 (see page 68).

[Ser20a] Amazon Web Services. *Authenticating Requests: Using the Authorization Header (AWS Signature Version 4)*. https://docs.aws.amazon.com/AmazonS3/latest/API/sigv4-auth-using-authorization-header.html. (Accessed on 03/04/2020). 2020 (see page 20).

[Ser20b] Amazon Web Services. *Managing Access Keys for IAM Users - AWS Identity and Access Management*. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html. (Accessed on 02/09/2020). Feb. 2020 (see page 19).

[SM13a] Maxim Schnjakin and Christoph Meinel. **Evaluation of cloud-raid: A secure and reliable storage above the clouds**. In: *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2013, 1–9 (see pages 14, 16).

[SM13b] Maxim Schnjakin and Christoph Meinel. **Scrutinizing the state of cloud storage with Cloud-RAID: A secure and reliable storage above the clouds**. In: *2013 IEEE Sixth International Conference on Cloud Computing*. IEEE. 2013, 309–318 (see pages 14, 16).

[SM16] Muhammad Sukmana and Christoph Meinel. **e-Government and Security Evaluation Tools Comparison for Indonesian e-Government System**. In: *Proceedings of the 4th International Conference on Information and Network Security*. 2016, 96–103 (see page 8).

[SMM13] Maxim Schnjakin, Tobias Metzke, and Christoph Meinel. **Applying erasure codes for fault tolerance in cloud-raid**. In: *2013 IEEE 16th International Conference on Computational Science and Engineering*. IEEE. 2013, 66–75 (see pages 14, 16).

[SS16] Murugiah Souppaya and Karen Scarfone. **Guide to enterprise telework, remote access, and bring your own device (BYOD) security**. *NIST Special Publication* 800 (2016), 46 (see page 66).

[SSM18] Johannes Sianipar, Muhammad Sukmana, and Christoph Meinel. **Moving sensitive data against live memory dumping, spectre and meltdown attacks**. In: *2018 26th International Conference on Systems Engineering (ICSEng)*. IEEE. 2018, 1–8 (see page 9).

[SSX14] Jiwu Shu, Zhirong Shen, and Wei Xue. **Shield: A stackable secure storage system for file sharing in public storage**. *Journal of Parallel and Distributed Computing* 74:9 (2014), 2872–2883 (see page 31).

[ST13] National Institute Standards and Technology. *NIST Cloud Computing Standards Roadmap*. https://www.nist.gov/system/files/documents/itl/cloud/NIST_SP-500-291_Version-2_2013_June18_FINAL.pdf. (Accessed on 02/14/2020). 2013 (see page 18).

[Ste20] Morgan. Steve. *The World Will Store 200 Zettabytes Of Data By 2025*. https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/. (Accessed on 07/22/2021). 2020 (see page 1).

[Suk+17] Muhammad IH Sukmana, Kennedy A Torkura, Christoph Meinel, and Hendrik Graupner. **Redesign cloudraid for flexible and secure enterprise file sharing over public cloud storage**. In: *Proceedings of the 10th International Conference on Security of Information and Networks*. ACM. 2017, 3–10 (see pages 5, 21, 22, 43–46).

[Suk+18] Muhammad IH Sukmana, Kennedy A Torkura, Feng Cheng, Christoph Meinel, and Hendrik Graupner. **Unified logging system for monitoring multiple cloud storage providers in cloud storage broker**. In: *2018 International Conference on Information Networking (ICOIN)*. IEEE. 2018, 44–49 (see pages 7, 154).

[Suk+19a] Muhammad IH Sukmana, Marvin Petzolt, Kennedy A Torkura, Hendrik Graupner, Feng Cheng, and Christoph Meinel. **Secure and Scalable Multi-Company Management in Enterprise Cloud Storage Broker System**. In: *Proceedings of the 17th IEEE International Symposium on Parallel and Distributed Processing with Applications*. IEEE. 2019 (see pages 6, 54–56, 58, 59, 61, 62).

[Suk+19b] Muhammad IH Sukmana, Kennedy A Torkura, Hendrik Graupner, Ankit Chauhan, Feng Cheng, and Christoph Meinel. **Supporting Internet-Based Location for Location-Based Access Control in Enterprise Cloud Storage Solution**. In: *International Conference on Advanced Information Networking and Applications*. Springer. 2019, 1240–1253 (see page 6).

[Suk+19c] Muhammad IH Sukmana, Kennedy A Torkura, Hendrik Graupner, Feng Cheng, and Christoph Meinel. **Unified Cloud Access Control Model for Cloud Storage Broker**. In: *2019 International Conference on Information Networking (ICOIN)*. IEEE. 2019, 60–65 (see pages 6, 137).

[Suk+20] Muhammad IH Sukmana, Kennedy A Torkura, Sezi DS Prasetyo, Feng Cheng, and Christoph Meinel. **A Brokerage Approach for Secure Multi-Cloud Storage Resource Management**. In: *16th EAI International Conference on Security and Privacy in Communication Networks (SecureComm) 2020*. Springer. 2020 (see pages 7, 119, 121, 125, 127, 129–131, 155).

[Suk+21a] Muhammad I. H. Sukmana, Justus Cöster, Wenzel Puenter, Kennedy A. Torkura, Feng Cheng, and Christoph Meinel. **A Feasibility Study of Log-Based Monitoring for Multi-cloud Storage Systems**. In: *Advanced Information Networking and Applications*. Ed. by Leonard Barolli, Isaac Woungang, and Tomoya Enokido. Cham: Springer International Publishing, 2021, 458–471. ISBN: 978-3-030-75075-6 (see pages 7, 152, 160).

[Suk+21b] Muhammad I. H. Sukmana., Kai-Oliver Kohlen., Carl Gödecken., Pascal Schulze., and Christoph Meinel. **Are You There, Moriarty? Feasibility Study of Internet-based Location for Location-based Access Control Systems**. In: *Proceedings of the 18th International Conference on Security and Cryptography - SECRYPT,* INSTICC. SciTePress, 2021, 111–124. ISBN: 978-989-758-524-1. DOI: 10.5220/0010541101110124 (see pages 6, 76, 79, 84, 86, 87, 89, 94–96, 101, 102, 104, 105).

[SW05] Amit Sahai and Brent Waters. **Fuzzy identity-based encryption**. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2005, 457–473 (see pages 37, 39).

[Sye+17] Hassan Jamil Syed, Abdullah Gani, Raja Wasim Ahmad, Muhammad Khurram Khan, and Abdelmuttlib Ibrahim Abdalla Ahmed. **Cloud monitoring: A review, taxonomy, and open research issues**. *Journal of Network and Computer Applications* 98 (2017), 11–26 (see page 145).

[TCB14] Adel Nadjaran Toosi, Rodrigo N Calheiros, and Rajkumar Buyya. **Interconnected cloud computing environments: Challenges, taxonomy, and survey**. *ACM Computing Surveys (CSUR)* 47:1 (2014), 1–47 (see page 122).

[TCD20] Orazio Tomarchio, Domenico Calcaterra, and Giuseppe Di Modica. **Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks**. *Journal of Cloud Computing* 9:1 (2020), 1–24 (see page 117).

[Tch+18] Andrei Tchernykh, Mikhail Babenko, Vanessa Miranda-López, Alexander Yu Drozdov, and Arutyun Avetisyan. **WA-RRNS: Reliable data storage system based on multi-cloud**. In: *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE. 2018, 666–673 (see page 112).

[Tha19] Thales. *2019 Global Cloud Security Study*. https://cpl.thalesgroup.com/cloud-security-research. (Accessed on 10/19/2020). 2019 (see page 1).

[TJA10] Hassan Takabi, James BD Joshi, and Gail-Joon Ahn. **Security and privacy challenges in cloud computing environments**. *IEEE Security & Privacy* 8:6 (2010), 24–31 (see pages 115, 116).

[Tor+17] Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel. **Leveraging cloud native design patterns for security-as-a-service applications**. In: *2017 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE. 2017, 90–97 (see page 8).

[Tor+18a] K. A. Torkura, M. I. H. Sukmana, M. Meinig, A. V. D. M. Kayem, F. Cheng, H. Graupner, and C. Meinel. **Securing Cloud Storage Brokerage Systems Through Threat Models**. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. 2018, 759–768. DOI: 10.1109/AINA.2018.00114 (see pages 141, 157).

[Tor+18b] Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel. **Cavas: Neutralizing application and container security vulnerabilities in the cloud native era**. In: *International Conference on Security and Privacy in Communication Systems*. Springer. 2018, 471–490 (see page 8).

[Tor+18c] Kennedy A Torkura, Muhammad IH Sukmana, Anne VDM Kayem, Feng Cheng, and Christoph Meinel. **A cyber risk based moving target defense mechanism for microservice architectures**. In: *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications (ISPA)*. IEEE. 2018, 932–939 (see page 9).

[Tor+18d] Kennedy A Torkura, Muhammad IH Sukmana, Michael Meinig, Feng Cheng, Christoph Meinel, and Hendrik Graupner. **A threat modeling approach for cloud storage brokerage and file sharing systems**. In: *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2018, 1–5 (see page 8).

[Tor+18e] Kennedy A Torkura, Muhammad IH Sukmana, Michael Meinig, Anne VDM Kayem, Feng Cheng, Hendrik Graupner, and Christoph Meinel. **Securing Cloud Storage Brokerage Systems Through Threat Models**. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. IEEE. 2018, 759–768 (see page 8).

[Tor+18f] Kennedy A Torkura, Muhammad IH Sukmana, Tim Strauss, Hendrik Graupner, Feng Cheng, and Christoph Meinel. **CSBAuditor: Proactive Security Risk Analysis for Cloud Storage Broker Systems**. In: *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2018, 1–10 (see pages 9, 117).

[Tor+19a] K. A. Torkura, M. I. H. Sukmana, F. Cheng, and C. Meinel. **SlingShot - Automated Threat Detection and Incident Response in Multi Cloud Storage Systems**. In: *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. 2019, 1–5. DOI: 10.1109/NCA.2019.8935040 (see pages 142, 145, 146).

[Tor+19b] Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel. **Security chaos engineering for cloud services: Work in progress**. In: *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2019, 1–3 (see page 9).

[Tor+19c] Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel. **SlingShot-Automated Threat Detection and Incident Response in Multi Cloud Storage Systems**. In: *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2019, 1–5 (see pages 9, 126).

[Tor+20] Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel. **Cloudstrike: Chaos engineering for security and resiliency in cloud infrastructure**. *IEEE Access* 8 (2020), 123044–123060 (see pages 10, 140).

[Tor+21] Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel. **Continuous auditing and threat detection in multi-cloud infrastructure**. *Computers & Security* 102 (2021), 102124 (see pages 10, 140).

[Tre19] Tresorit. *How to improve geo-location accuracy?* https://support.tresorit.com/hc/en-us/articles/216114447-How-to-improve-geo-location-accuracy-. (Accessed on 05/27/2021). 2019 (see page 69).

[Tre20a] Tresorit. *Data storage.* https://support.tresorit.com/hc/en-us/articles/360017472540-Data-storage. (Accessed on 06/09/2021). 2020 (see page 113).

[Tre20b] Tresorit. *Security and Encryption.* https://tresorit.com/security/encryption. (Accessed on 03/29/2020). 2020 (see page 33).

[Tre20c] Tresorit. *White Paper.* https://tresorit.com/files/tresoritwhitepaper.pdf. (Accessed on 03/06/2020). 2020 (see pages 29, 33, 34).

[Tre21a] Tresorit. *Export user activity reports.* https://support.tresorit.com/hc/en-us/articles/360024340453-Export-user-activity-reports. (Accessed on 05/27/2021). 2021 (see page 144).

[Tre21b] Tresorit. *Introduction to Admin Center.* https://support.tresorit.com/hc/en-us/articles/360000980474-Introduction-to-Admin-Center. (Accessed on 05/27/2021). 2021 (see page 144).

[Tre21c] Tresorit. *Secure File Sharing & Encrypted Cloud Storage.* https://tresorit.com/features. (Accessed on 05/27/2021). 2021 (see page 29).

[Tre21d] Tresorit. *Third party services.* https://support.tresorit.com/hc/en-us/articles/216114397-Third-party-services. (Accessed on 06/09/2021). 2021 (see page 113).

[Tre21e] Tresorit. *Tresorit policy settings explained.* https://support.tresorit.com/hc/en-us/articles/360020362094-Tresorit-policy-settings-explained. (Accessed on 03/10/2021). 2021 (see page 69).

[TSM17] Kennedy A Torkura, Muhammad IH Sukmana, and Christoph Meinel. **Integrating continuous security assessments in microservices and cloud native applications**. In: *Proceedings of the 10th International Conference on Utility and Cloud Computing.* 2017, 171–180 (see page 8).

[UO16] Nils Ulltveit-Moe and Vladimir Oleshchuk. **Enforcing mobile security with location-aware role-based access control**. *Security and Communication Networks* 9:5 (2016), 429–439 (see page 68).

[Van+19] Dimitri Van Landuyt, Luuk Raaijmakers, Ansar Rafique, and Wouter Joosen. **Continuous and Client-centric Trust Monitoring in Multicloud Storage.** In: *CLOSER*. 2019, 100–110 (see page 142).

[VB18] Blesson Varghese and Rajkumar Buyya. **Next generation cloud computing: New trends and research directions**. *Future Generation Computer Systems* 79 (2018), 849–861 (see pages 117, 118).

[Wan+10] Cong Wang, Kui Ren, Wenjing Lou, and Jin Li. **Toward publicly auditable secure cloud data storage services**. *IEEE network* 24:4 (2010), 19–24 (see page 141).

[WB14] Jonathan Stuart Ward and Adam Barker. **Observing the clouds: a survey and taxonomy of cloud monitoring**. *Journal of Cloud Computing* 3:1 (2014), 1–30 (see page 145).

[WSS07] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. **Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts.** In: *NSDI*. Vol. 7. 2007, 23–23 (see page 108).

[WTM17] Thu Yein Win, Huaglory Tianfield, and Quentin Mair. **Big data based security analytics for protecting virtualized infrastructures in cloud computing**. *IEEE Transactions on Big Data* 4:1 (2017), 11–25 (see page 143).

[Xue+16] Yingjie Xue, Jianan Hong, Wei Li, Kaiping Xue, and Peilin Hong. **LABAC: A location-aware attribute-based access control scheme for cloud storage**. In: *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2016, 1–6 (see page 68).

[Yam+14] Shota Yamada, Nuttapong Attrapadung, Goichiro Hanaoka, and Noboru Kunihiro. **A framework and compact constructions for non-monotonic attribute-based encryption**. In: *International Workshop on Public Key Cryptography*. Springer. 2014, 275–292 (see page 42).

[Yam+19] Go Yamanaka, Takayuki Nishio, Masahiro Morikura, Koji Yamamoto, Yuichi Maki, Shin-ichiro Eitoku, and Takuya Indo. **Geo-Fencing in Wireless LANs with Camera for Location-Based Access Control**. In: *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2019, 1–4 (see page 68).

[Yan+13] Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, and Ruitao Xie. **DAC-MACS: Effective data access control for multiauthority cloud storage systems**. *IEEE Transactions on Information Forensics and Security* 8:11 (2013), 1790–1801 (see pages 32, 40, 48).

[Yan+18] Rupeng Yang, Qiuliang Xu, Man Ho Au, Zuoxia Yu, Hao Wang, and Lu Zhou. **Position based cryptography with location privacy: A step for fog computing**. *Future Generation Computer Systems* 78 (2018), 799–806 (see page 68).

[Yeg+19] Bahador Yeganeh, Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. **How Cloud Traffic Goes Hiding: A Study of Amazon's Peering Fabric**. In: *Proceedings of the Internet Measurement Conference*. 2019, 202–216 (see page 99).

[Yeg+20] Bahador Yeganeh, Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. **A First Comparative Characterization of Multi-cloud Connectivity in Today's Internet**. In: *International Conference on Passive and Active Network Measurement*. Springer. 2020, 193–210 (see page 99).

[Yu+16] Xiao Yu, Pallavi Joshi, Jianwu Xu, Guoliang Jin, Hui Zhang, and Guofei Jiang. **Cloudseer: Workflow monitoring of cloud infrastructures via interleaved logs**. *ACM SIGARCH Computer Architecture News* 44:2 (2016), 489–502 (see page 167).

[YWH17] Zhen Yang, Wenyu Wang, and Yongfeng Huang. **Ensuring reliable logging for data accountability in untrusted cloud storage**. In: *2017 IEEE International Conference on Communications (ICC)*. IEEE. 2017, 1–6 (see page 170).

[ZB11] Paul A Zandbergen and Sean J Barbeau. **Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones**. *The Journal of Navigation* 64:3 (2011), 381–399 (see page 72).

[Zic+14] Sebastian Zickau, Dirk Thatmann, Tatiana Ermakova, Jonas Repschläger, Rüdiger Zarnekow, and Axel Küpper. **Enabling location-based policies in a healthcare cloud computing environment**. In: *2014 IEEE 3rd International Conference on Cloud Networking (Cloudnet)*. IEEE. 2014, 333–338 (see page 67).

[Zic+16] Sebastian Zickau, Dirk Thatmann, Artjom Butyrtschik, Iwailo Denisow, and Axel Küpper. **Applied attribute-based encryption schemes**. In: *19th International ICIN Conference-Innovations in Clouds, Internet and Networks-March*. 2016, 1–3 (see page 42).

List of Publications

Articles in Refereed Journals

[1] **Holistic strategy-based threat model for organizations**. *Procedia Computer Science* 151 (2019), 100–107. Joint work with Michael Meinig, Muhammad IH Sukmana, Kennedy A Torkura, and Christoph Meinel.

[2] **Cloudstrike: Chaos engineering for security and resiliency in cloud infrastructure**. *IEEE Access* 8 (2020), 123044–123060. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel.

[3] **Continuous auditing and threat detection in multi-cloud infrastructure**. *Computers & Security* 102 (2021), 102124. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel.

Articles in Refereed Conference Proceedings

[4] **Secure Deduplication on Public Cloud Storage**. In: *Proceedings of the 2019 4th International Conference on Big Data and Computing*. 2019, 34–41. Joint work with Hendrik Graupner, Kennedy A Torkura, Muhammad IH Sukmana, and Christoph Meinel.

[5] **e-Government and Security Evaluation Tools Comparison for Indonesian e-Government System**. In: *Proceedings of the 4th International Conference on Information and Network Security*. 2016, 96–103. Joint work with Muhammad Sukmana and Christoph Meinel.

[6] **Moving sensitive data against live memory dumping, spectre and meltdown attacks**. In: *2018 26th International Conference on Systems Engineering (ICSEng)*. IEEE. 2018, 1–8. Joint work with Johannes Sianipar, Muhammad Sukmana, and Christoph Meinel.

[7] **Redesign cloudraid for flexible and secure enterprise file sharing over public cloud storage**. In: *Proceedings of the 10th International Conference on Security of Information and Networks.* ACM. 2017, 3–10. Joint work with Muhammad IH Sukmana, Kennedy A Torkura, Christoph Meinel, and Hendrik Graupner.

[8] **Unified logging system for monitoring multiple cloud storage providers in cloud storage broker**. In: *2018 International Conference on Information Networking (ICOIN).* IEEE. 2018, 44–49. Joint work with Muhammad IH Sukmana, Kennedy A Torkura, Feng Cheng, Christoph Meinel, and Hendrik Graupner.

[9] **Secure and Scalable Multi-Company Management in Enterprise Cloud Storage Broker System**. In: *Proceedings of the 17th IEEE International Symposium on Parallel and Distributed Processing with Applications.* IEEE. 2019. Joint work with Muhammad IH Sukmana, Marvin Petzolt, Kennedy A Torkura, Hendrik Graupner, Feng Cheng, and Christoph Meinel.

[10] **Supporting Internet-Based Location for Location-Based Access Control in Enterprise Cloud Storage Solution**. In: *International Conference on Advanced Information Networking and Applications.* Springer. 2019, 1240–1253. Joint work with Muhammad IH Sukmana, Kennedy A Torkura, Hendrik Graupner, Ankit Chauhan, Feng Cheng, and Christoph Meinel.

[11] **Unified Cloud Access Control Model for Cloud Storage Broker**. In: *2019 International Conference on Information Networking (ICOIN).* IEEE. 2019, 60–65. Joint work with Muhammad IH Sukmana, Kennedy A Torkura, Hendrik Graupner, Feng Cheng, and Christoph Meinel.

[12] **A Brokerage Approach for Secure Multi-Cloud Storage Resource Management**. In: *16th EAI International Conference on Security and Privacy in Communication Networks (SecureComm) 2020.* Springer. 2020. Joint work with Muhammad IH Sukmana, Kennedy A Torkura, Sezi DS Prasetyo, Feng Cheng, and Christoph Meinel.

[13] **A Feasibility Study of Log-Based Monitoring for Multi-cloud Storage Systems**. In: *Advanced Information Networking and Applications.* Ed. by Leonard Barolli, Isaac Woungang, and Tomoya Enokido. Cham:

Springer International Publishing, 2021, 458–471. ISBN: 978-3-030-75075-6. Joint work with Muhammad I. H. Sukmana, Justus Cöster, Wenzel Puenter, Kennedy A. Torkura, Feng Cheng, and Christoph Meinel.

[14] **Are You There, Moriarty? Feasibility Study of Internet-based Location for Location-based Access Control Systems**. In: *Proceedings of the 18th International Conference on Security and Cryptography - SECRYPT,* INSTICC. SciTePress, 2021, 111–124. ISBN: 978-989-758-524-1. DOI: 10.5220/0010541101110124. Joint work with Muhammad I. H. Sukmana., Kai-Oliver Kohlen., Carl Gödecken., Pascal Schulze., and Christoph Meinel..

[15] **Leveraging cloud native design patterns for security-as-a-service applications**. In: *2017 IEEE International Conference on Smart Cloud (SmartCloud).* IEEE. 2017, 90–97. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel.

[16] **Cavas: Neutralizing application and container security vulnerabilities in the cloud native era**. In: *International Conference on Security and Privacy in Communication Systems.* Springer. 2018, 471–490. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel.

[17] **A cyber risk based moving target defense mechanism for microservice architectures**. In: *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications (ISPA).* IEEE. 2018, 932–939. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Anne VDM Kayem, Feng Cheng, and Christoph Meinel.

[18] **A threat modeling approach for cloud storage brokerage and file sharing systems**. In: *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium.* IEEE. 2018, 1–5. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Michael Meinig, Feng Cheng, Christoph Meinel, and Hendrik Graupner.

[19] **Securing Cloud Storage Brokerage Systems Through Threat Models**. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA).* IEEE. 2018, 759–768. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Michael Meinig, Anne VDM Kayem, Feng Cheng, Hendrik Graupner, and Christoph Meinel.

[20] **CSBAuditor: Proactive Security Risk Analysis for Cloud Storage Broker Systems**. In: *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2018, 1–10. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Tim Strauss, Hendrik Graupner, Feng Cheng, and Christoph Meinel.

[21] **Security chaos engineering for cloud services: Work in progress**. In: *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2019, 1–3. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel.

[22] **SlingShot-Automated Threat Detection and Incident Response in Multi Cloud Storage Systems**. In: *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2019, 1–5. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, Feng Cheng, and Christoph Meinel.

[23] **Integrating continuous security assessments in microservices and cloud native applications**. In: *Proceedings of the 10th International Conference on Utility and Cloud Computing*. 2017, 171–180. Joint work with Kennedy A Torkura, Muhammad IH Sukmana, and Christoph Meinel.

List of Figures

List of Tables

Acronyms

AA attribute authority.

ABAC attribute-based access control.

ABE attribute-based encryption.

ACL access control list.

AP access point.

API application programming interface.

AWS Amazon Web Services.

Azure Microsoft Azure.

CBG Constraint-based Geolocation.

CfB CloudRAID for Business.

CLI command-line interface.

CP-ABE ciphertext-based policy attribute-based encryption.

CSP cloud storage provider.

ECSB enterprise cloud storage broker.

EFSS enterprise file synchronization and share.

GCP Google Cloud Platform.

GeoXACML Geospatial eXtensible Access Control Markup Language.

IaaS Infrastructure-as-a-Service.

IAM Identity and Access Management.

ILAC Internet-based location access control.

KMS key management system.

KP-ABE key-based policy attribute-based encryption.

LBAC location-based access control.

MA-ABE multi-authority attribute-based encryption.

MA-CfB Multi-Authority CloudRAID for Business.

MTurk Mechanical Turk.

OSINT open source intelligence.

PAD-TFDAC-MACS Practical Applied Distributed-TFDAC-MACS.

RBG random bit generator.

revokedKeyList list of revoked keys.

RSA Rivest–Shamir–Adleman.

RTT round-trip-time.

S3 Simple Storage Service.

SA-CfB Single-Authority CloudRAID for Business.

SaaS Software-as-a-Service.

SLA service level agreement.

TFDAC-MACS Two-Factor Data Access Control with Efficient Revocation for Multi-Authority Cloud Storage Systems.

VM virtual machine.

VPN virtual private network.

WLAN wireless local area network.

XACML eXtensible Access Control Markup Language.