Security Enhancement In Cloud Computing

1Dr. Rajanarayanan S , 2 Rohit Menon
Associate Professor, Final Year Cse, Cse
Selvam College Of Technology,
Namakkal, Tamilnadu, India

Abstract: In this paper, we propose a two-factor data security protection mechanism with factor revocability for cloud storage system. Our system allows a sender to send an encrypted message to a receiver through a cloud storage server. The sender only needs to know the identity of the receiver but no other information (such as its public key or its certificate). The receiver needs to possess two things in order to decrypt the ciphertext. The first thing is his/her secret key stored in the computer. The second thing is a unique personal security device which connects to the computer. It is impossible to decrypt the ciphertext without either piece. More importantly, once the security device is stolen or lost, this device is revoked. It cannot be used to decrypt any ciphertext. This can be done by the cloud server which will immediately execute some algorithms to change the existing ciphertext to be un-decryptable by this device. This process is completely transparent to the sender. Furthermore, the cloud server cannot decrypt any ciphertext at any time. The security and efficiency analysis show that our system is not only secure but also practical.

Objectives:
- IBE-Identity-based encryption
- It provides two-factor data encryption protection.
- our system, for the first time, provides security device (one of the factors) revocability. Once the security device is stolen or reported as lost, this device is revoked. That is, using this device can no longer decrypt any ciphertext (corresponding to the user) in any circumstance. The cloud will immediately execute some algorithms to change the existing ciphertext to be un-decryptable by this device.
- The cloud server cannot decrypt any ciphertext at any time.

INTRODUCTION

CLOUD storage is a model of networked storage system where data is stored in pools of storage which are generally hosted by third parties. There are many benefits to use cloud storage. The most notable is data accessibility. Data stored in the cloud can be accessed at any time from any place as long as there is network access. Storage maintenance tasks, such as purchasing additional storage capacity, can be offloaded to the responsibility of a service provider. Another advantage of cloud storage is data sharing between users. If Alice wants to share a piece of data (e.g., a video) to Bob, it may be difficult for her to send it by email due to the size of data. Instead, Alice uploads the file to a cloud storage system so that Bob can download it at anytime. Despite its advantages, outsourcing data storage also increases the attack surface area at the same time.

For example, when data is distributed, the more locations it is stored the higher risk it contains for unauthorized physical access to the data. By sharing storage and networks with many other users it is also possible for other unauthorized users to access your data. This may be due to mistaken actions, faulty equipment, or sometimes because of criminal intent. A promising solution to offset the risk is to deploy encryption technology. Encryption can protect data as it is being transmitted to and from the cloud service. It can further protect data that is stored at the service provider. Even there is an unauthorized adversary who has gained access to the cloud, as the data has been encrypted, the adversary cannot get any information about the plaintext. Asymmetric encryption allows the encryptor to use only the public information (e.g., public key or identity of the receiver) to generate a ciphertext while the receiver uses his/her own secret key to decrypt. This is the most convenient mode of encryption for data transition, due to the elimination of key management existed in symmetric encryption.

ENHANCED SECURITY PROTECTION.

In a normal asymmetric encryption, there is a single secret key corresponding to a public key or an identity. The decryption of ciphertext only requires this key. The key is usually stored inside either a personal computer or a trusted server, and may be protected by a password. The security protection is sufficient if the computer/server is isolated from an opening network. Unfortunately, this is not what happens in the real life. When being connected with the world through the
Internet, the computer/server may suffer from a potential risk that hackers may intrude into it to compromise the secret key without letting the key owner know. In the physical security aspect, the computer storing a user decryption key may be used by another user when the original computer user (i.e., the key owner) is away (e.g., when the user goes to toilet for a while without locking the machine). In an enterprise or college, the sharing usage of computers is also common. For example, in a college, a public computer in a copier room will be shared with all students staying at the same floor.

In these cases, the secret key can be compromised by some attackers who can access the victim’s personal data stored in the cloud system. Therefore, there exists a need to enhance the security protection. An analogy is e-banking security. Many e-banking applications require a user to use both a password and a security device (two factors) to login system for money transfer. The security device may display a one-time password to let the user type it into the system or it may be needed to connect with the computer (e.g., through USB or NFC). The purpose of using two factors is to enhance the security protection for the access control. As cloud computing becomes more mature and there will be more applications and storage services provided by the cloud, it is easy to foresee that the security for data protection in the cloud should be further enhanced. They will become more sensitive and important, as if the e-banking analogy. Actually, we have noticed that the concept of two-factor encryption, which is one of the encryption trends for data protection,1 has been spread into some real-world applications, for example, full disk encryption with Ubuntu system, AT&T two factor encryption for Smartphones, 2 electronic vaulting and druva—cloud-based data encryption.3 However, these applications suffer from a potential risk about factor revocability that may limit their practicability. Note we will explain it later. A flexible and scalable two-factor encryption mechanism is really desirable in the era of cloud computing.

EXISTING SYSTEM

Data sharing is an important functionality in cloud Storage. For example, bloggers can let their friends view a subset of their private pictures; an enterprise may grant her employees access to a portion of sensitive data. The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. However, finding an efficient and secure way to share partial data in cloud storage is not trivial. Transferring these secret keys inherently requires a secure channel, and storing these keys requires rather expensive secure storage. Now days the data sharing can be accessible with the asymmetric key encryption only Like Private key and public Key mechanism.

Disadvantages of Existing System:

1. Secret keys are required tamper-proof memory which is very expensive.
2. Sharing the partial data with others is not possible.

PROPOSED SYSTEM

we propose a novel two-factor security protection mechanism for data stored in the cloud. Our mechanism provides the following nice features: 1) Our system is an IBE (Identity-based encryption) based mechanism. That is, the sender only needs to know the identity of the receiver in order to send an encrypted data (ciphertext) to him/her. No other information of the receiver (e.g., public key, certificate etc.) is required. Then the sender sends the ciphertext to the cloud where the receiver can download it at anytime. 2) Our system provides two-factor data encryption protection. In order to decrypt the data stored in the cloud, the user needs to possess two things. First, the user needs to have his/her secret key which is stored in the computer. Second, the user needs to have a unique personal security device which will be used to connect to the computer (e.g., USB, Bluetooth). It is impossible to decrypt the ciphertext without either piece. 3) More importantly, our system, for the first time, provides security device (one of the factors) revocability. Once the security device is stolen or reported as lost, this device is revoked; the corresponding ciphertext will be updated automatically by the cloud server without any notice of the data owner.

Advantages of Proposed System:

1. Our solution not only enhances the confidentiality of the data, but also offers the revocability of the device so that once the device is revoked; the corresponding ciphertext will be updated automatically by the cloud server without any notice of the data owner.
2. The cloud server cannot decrypt any ciphertext at any time
LITERATURE REVIEW

Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage

Data sharing is an important functionality in cloud storage. In this article, we show how to securely, efficiently, and flexibly share data with others in cloud storage. We describe new public-key cryptosystems which produce constant-size ciphertexts such that efficient delegation of decryption rights for any set of ciphertexts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of ciphertext set in cloud storage, but the other encrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We also describe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.

Cloud storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Nowadays, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 25GB (or a few dollars for more than 1TB). Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication, which means any unexpected privilege escalation will expose all data. In a shared-tenancy cloud computing environment, things become even worse.

Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM co-resident with the target one. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without compromising the data owners anonymity. Likewise, cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality. A cryptographic solution, with proven security relied on number-theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. These users are motivated to encrypt their data with their own keys before uploading them to the server. Data sharing is an important functionality in cloud storage. For example, bloggers can let their friends view a subset of their private pictures; an enterprise may grant her employees access to a portion of sensitive data. The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. However, finding an efficient and secure way to share partial data in cloud storage is not trivial. Below we will take Dropbox1 as an example for illustration.

How to protect users’ data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this article, we consider how to “compress” secret keys in public-key cryptosystems which support delegation of secret keys for different ciphertext classes in cloud storage. No matter which one among the power set of classes, the delegatee can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges.

Dynamic Audit Services for Outsourced Storages in Clouds

In this paper, we propose a dynamic audit service for verifying the integrity of an untrusted and outsourced storage. Our audit service is constructed based on the techniques, fragment structure, random sampling, and index-hash table, supporting provable updates to outsourced data and timely anomaly detection. In addition, we propose a method based on probabilistic query and periodic verification for improving the performance of audit services. Our experimental results not only validate the effectiveness of our approaches, but also show our audit system verifies the integrity with lower computation overhead and requiring less extra storage for audit metadata.

Rajanarayanan, Rohit Menon .... (IJ0SER) April–2017

(p)-1204-213
CLOUD computing provides a scalable environment for growing amounts of data and processes that work on various applications and services by means of on-demand self-services. Especially, the outsourced storage in clouds has become a new profit growth point by providing a comparably low-cost, scalable, location-independent platform or managing clients’ data. The cloud storage service (CSS) relieves the burden for storage management and maintenance. However, if such an important service is vulnerable to attacks or failures, it would bring irretrievable losses to the clients because their data or archives are stored in an uncertain storage pool outside the enterprises. These security risks come from the following reasons: First, the cloud infrastructures are much more powerful and reliable than personal computing devices, but they are still susceptible to internal threats (e.g., via virtual machine) and external threats (e.g., via system holes) that can damage data integrity; second, for the benefits of possession, there exist various motivations for cloud service providers (CSP) to behave unfaithfully toward the cloud users; furthermore, disputes occasionally suffer from the lack of trust on CSP because the data change may not be timely known by the cloud users, even if these disputes may result from the users’ own improper operations. Therefore, it is necessary for CSP to offer an efficient audit service to check the integrity and availability of stored data [4]. Security audit is an important solution enabling traceback and analysis of any activities including data accesses, security breaches, application activities, and so on. Data security tracking is crucial for all organizations that should comply with a wide range of federal regulations including the Sarbanes-Oxley Act, Basel II, HIPAA, and so on.1 Furthermore, compared to the common audit, the audit services for cloud storages should provide clients with a more efficient proof for verifying the integrity of stored data. Unfortunately, the traditional cryptographic technologies, based on hash functions and signature schemes, cannot support for data integrity verification without a local copy of data. In addition, it is evidently impractical for audit services to download the whole data for checking data validation due to the communication cost, especially for large-size files. Therefore, following security and performance objectives should be addressed to achieve an efficient audit for outsourced storage in clouds:

In this paper, we presented a construction of dynamic audit services for untrusted and outsourced storages. We also presented an efficient method for periodic sampling audit to enhance the performance of TPAs and storage service providers. Our experiments showed that our solution has a small, constant amount of overhead, which minimizes computation and communication costs

Privacy-Preserving Public Auditing for Secure Cloud Storage

Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.

CLOUD Computing has been envisioned as the next-generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced to the Cloud. From users’ perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenance, etc. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users’ outsourced data. Since
cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user’s ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Secondly, there do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been or is rarely accessed, or even hide data loss incidents so as to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale data storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the cloud architecture.

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users’ fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient.

Distributed, concurrent, and independent access to encrypted cloud databases

Placing critical data in the hands of a cloud provider should come with the guarantee of security and availability for data at rest, in motion, and in use. Several alternatives exist for storage services, while data confidentiality solutions for the Database as a Service paradigm are still immature. We propose a novel architecture that integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data. This is the first solution supporting geographically distributed clients to connect directly to an encrypted cloud database, and to execute concurrent and independent operations including those modifying the database structure. The proposed architecture has the further advantage of eliminating intermediate proxies that limit the elasticity, availability and scalability properties that are intrinsic in cloud-based solutions. The efficacy of the proposed architecture is evaluated through theoretical analyses and extensive experimental results based on a prototype implementation subject to the TPC-C standard benchmark for different numbers of clients and network latencies.

In a cloud context, where critical information is placed in infrastructures of untrusted third parties, ensuring data confidentiality is of paramount importance. This requirement imposes clear data management choices: original plain data must be accessible only by trusted parties that do not include cloud providers, intermediaries, Internet; in any untrusted context data must be encrypted. Satisfying these goals has different levels of complexity depending on the type of cloud service. There are several solutions ensuring confidentiality for the storage as a service paradigm, while guaranteeing confidentiality in the database as a service (DBaaS) paradigm is still an open research area. In this context, we propose SecureDBaaS as the first solution that allows cloud tenants to take full advantage of DBaaS qualities, such as availability, reliability, elastic scalability, without exposing unencrypted data to the cloud provider. The architecture design was motivated by a threefold goal: to allow multiple, independent, and geographically distributed clients to execute concurrent operations on encrypted data, including SQL statements that modify the database structure; to preserve data confidentiality and consistency at the client and cloud level; to eliminate any intermediate server between the cloud client and the cloud provider. The possibility of combining availability, elasticity, and scalability of a typical cloud DBaaS with data confidentiality are demonstrated through a prototype of SecureDBaaS that supports the execution of concurrent and independent operations to the remote encrypted database from many geographically distributed clients as in any unencrypted DBaaS setup. To achieve these goals, SecureDBaaS integrates existing cryptographic schemes, isolation mechanisms, and novel strategies for management of encrypted metadata on the untrusted cloud database. This paper contains a theoretical discussion about solutions for data consistency issues due to concurrent and independent client accesses to encrypted data. In this context, we cannot apply fully
homomorphic encryption schemes because of their excessive computational complexity.

We propose an innovative architecture that guarantees confidentiality of data stored in public cloud databases. Unlike state of the art approaches, our solution does not rely on an intermediate proxy that we consider a single point of failure and a bottleneck limiting availability and scalability of typical cloud database services. A large part of the research includes solutions to support concurrent SQL operations (including statements modifying the database structure) on encrypted data issued by heterogeneous and possibly geographically dispersed clients. The proposed architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS, such as the experimented PostgreSQL Plus Cloud Database, Windows Azure and Xeround. There are no theoretical and practical limits to extend our solution to other platforms and to include new encryption algorithms.

It is worth to observe that experimental results based on the TPC-C standard benchmark show that the performance impact of data encryption on response time becomes negligible because it is masked by network latencies that are typical of cloud scenarios. In particular, concurrent read and write operations that do not modify the structure of the encrypted database cause negligible overhead. Dynamic scenarios characterized by (possibly) concurrent modifications of the database structure are supported, but at the price of high computational costs. These performance results open the space to future improvements that we are investigating.

Security as a Service Model for Cloud Environment

Cloud computing is becoming increasingly important for provision of services and storage of data in the Internet. However there are several significant challenges in securing cloud infrastructures from different types of attacks. The focus of this paper is on the security services that a cloud provider can offer as part of its infrastructure to its customers (tenants) to counteract these attacks. Our main contribution is a security architecture that provides a flexible security as a service model that a cloud provider can offer to its tenants and customers of its tenants. Our security as a service model while offering a baseline security to the provider to protect its own cloud infrastructure also provides flexibility to tenants to have additional security functionalities that suit their security requirements. The paper describes the design of the security architecture and discusses how different types of attacks are counteracted by the proposed architecture. We have implemented the security architecture and the paper discusses analysis and performance evaluation results.

CLOUD computing has become an important technology where cloud services providers provide computing resources to their customers (tenants) to host their data or perform their computing tasks. Cloud computing can be categorized into different service deliver models such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Virtualisation is one of the key technologies used in the IaaS cloud infrastructures. For instance, virtualisation is used by some of the major cloud service providers such as Amazon and Microsoft in the provision of cloud services. We will use the term tenant to refer to cloud customers who wish to access services from cloud providers. Tenants can themselves be using their virtual machines to provide services to their own customers; we will refer to customers (or users) as those who use the services of the tenants. Hence customers in our architecture are the customers of the tenants. In general, the tenants in the cloud can run different operating systems and applications in their virtual machines. As the operating systems and applications of the tenants can be potentially large and complex, they may contain security vulnerabilities. Furthermore, there can be several tenants on the same physical platform sharing resources in a cloud infrastructure. The vulnerabilities in operating systems and applications can be potentially exploited by an attacker to generate different types of attacks. These attacks can be targeted against the cloud infrastructure as well as against other virtual machines belonging to other tenants. So there is a need to design security architecture and develop techniques that can be used by the cloud service provider for securing its infrastructure and tenant virtual machines.

In this paper we have proposed a security architecture that provides a security as a service model that a cloud provider can offer to its multiple tenants and customers of its tenants. Our security as a service model while offering a baseline security to the provider to protect its own cloud infrastructure also provides flexibility to tenants to have additional security functionalities that suit their security requirements. The paper described the design of the security architecture and discussed how different types of attacks are counteracted by the proposed architecture. We have described the implementation of the security architecture and gave a detailed analysis of the security mechanisms and performance evaluation results.
**Toward Secure and Dependable Storage Services in Cloud Computing**

Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users’ physical possession of their outsourced data, which inevitably poses new security risks toward the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, we propose in this paper a flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server. Considering the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

SEVERAL trends are opening up the era of cloud computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the Software as a Service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers. Moving data into the cloud offers great convenience to users since they don’t have to care about the complexities of direct hardware management. The pioneer of cloud computing vendors, Amazon Simple Storage Service (S3), and Amazon Elastic Compute Cloud (EC2) are both well-known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers (CSP) for the availability and integrity of their data. On the one hand, although the cloud infrastructures are much more powerful and reliable than personal computing devices, broad range of both internal and external threats for data integrity still exist. Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time. On the other hand, since users may not retain a local copy of outsourced data, there exist various incentives for CSP to behave unfaithfully toward the cloud users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion. Similarly, CSP may even attempt to hide data loss incidents so as to maintain a reputation. Therefore, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, its lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users.

In this paper, we investigate the problem of data security in cloud data storage, which is essentially a distributed storage system. To achieve the assurances of cloud data integrity and availability and enforce the quality of dependable cloud storage service for users, we propose an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, we can almost guarantee the simultaneous identification of the misbehaving server(s). Considering the time, computation resources, and even the related online burden of users, we also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the integrity checking tasks to third-party auditors and be worry-free to use the cloud storage services. Through detailed security and extensive experiment results, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

**NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds**

To provide fault tolerance for cloud storage, recent studies propose to stripe data across multiple
cloud vendors. However, if a cloud suffers from a permanent failure and loses all its data, we need to repair the lost data with the help of the other surviving clouds to preserve data redundancy. We present a proxy-based storage system for fault-tolerant multiple-cloud storage called NCCloud, which achieves cost-effective repair for a permanent single-cloud failure. NCCloud is built on top of a network-coding-based storage scheme called the functional minimum-storage regenerating (FMSR) codes, which maintain the same fault tolerance and data redundancy as in traditional erasure codes (e.g., RAID-6), but use less repair traffic and, hence, incur less monetary cost due to data transfer. One key design feature of our FMSR codes is that we relax the encoding requirement of storage nodes during repair, while preserving the benefits of network coding in repair. We implement a proof-of-concept prototype of NCCloud and deploy it atop both local and commercial clouds. We validate that FMSR codes provide significant monetary cost savings in repair over RAID-6 codes, while having comparable response time performance in normal cloud storage operations such as upload/download.

CLOUD storage provides an on-demand remote backup solution. However, using a single-cloud storage provider raises concerns such as having a single point of failure and vendor lock-ins. As suggested in, a plausible solution is to stripe data across different cloud providers. By exploiting the diversity of multiple clouds, we can improve the fault tolerance of cloud storage. While striping data with conventional erasure codes performs well when some clouds experience short-term transient failures or foreseeable permanent failures, there are real-life cases showing that permanent failures do occur and are not always foreseeable. In view of this, this work focuses on unexpected permanent cloud failures. When a cloud fails permanently, it is necessary to activate repair to maintain data redundancy and fault tolerance. A repair operation retrieves data from existing surviving clouds over the network and reconstructs the lost data in a new cloud. Today’s cloud storage providers charge users for outbound data, so moving an enormous amount of data across clouds can introduce significant monetary costs. It is important to reduce the repair traffic (i.e., the amount of data being transferred over the network during repair), and hence, the monetary cost due to data migration. To minimize repair traffic, regenerating codes have been proposed for storing data redundantly in a distributed storage system (a collection of interconnected storage nodes). Each node could refer to a simple storage device, a storage site, or a cloud storage provider. Regenerating codes are built on the concept of network coding, in the sense that nodes perform encoding operations and send encoded data. During repair, each surviving node encodes its stored data chunks and sends the encoded chunks to a new node, which then regenerates the lost data. It is shown that regenerating codes require less repair traffic than traditional erasure codes with the same fault-tolerance level.

We present NCCloud, a proxy-based, multiple-cloud storage system that practically addresses the reliability of today’s cloud backup storage. NCCloud not only provides fault tolerance in storage, but also allows cost-effective repair when a cloud permanently fails. NCCloud implements a practical version of the FMSR codes, which regenerates new parity chunks during repair subject to the required degree of data redundancy. Our FMSR code implementation eliminates the encoding requirement of storage nodes (or cloud) during repair, while ensuring that the new set of stored chunks after each round of repair preserves the required fault tolerance. Our NCCloud prototype shows the effectiveness of FMSR codes in the cloud backup usage, in terms of monetary costs and response times.
In this paper, we introduced a novel two-factor data security protection mechanism for cloud storage system, in which a data sender is allowed to encrypt the data with knowledge of the identity of a receiver only, while the receiver is required to use both his/her secret key and a security device to gain access to the data. Our solution not only enhances the confidentiality of the data, but also offers the revocability of the device so that once the device is revoked, the corresponding ciphertext will be updated automatically by the cloud server without any notice of the data owner. Furthermore, we presented the security proof and efficiency analysis for our system.

REFERENCES


Rajanarayanan, Rohit Menon .... (IJOSE) April– 2017
(p)-1204-213