ISSN:2249-5789
Mahima Joshi et al, International Journal of Computer Science & Communication Networks,Vol 1(2), 171-175

SECURE CLOUD STORAGE

Mahima Joshi, Yudhveer Singh Moudgil
Computer Science and Engineering Department
Uttaranchal Institute of Technology, Dehradun
Mahima_joshi2003@yahoo.com,Yudhveer127@gmail.com

Abstract

As an emerging technology and business paradigm, Cloud Computing has taken commercial computing by storm. Cloud computing platforms provide easy access to a company's high-performance computing and storage infrastructure through web services. We consider the problem of building a secure cloud storage service on top of a public cloud infrastructure where the service provider is not completely trusted by the customer. We describe, at a high level, several architectures that combine recent and non-standard cryptographic primitives in order to achieve our goal. We survey the benefits such an architecture would provide to both customers and service providers and give an overview of recent advances in cryptography motivated specifically by cloud storage.

Keywords*-cloud computing, cloud storage, architecture, cryptographic key, token.*

1. Introduction

Cloud computing portends a major change in how to store information and run applications. Instead of running programs and data on an individual desktop computer, everything is hosted in the "cloud"—a nebulous assemblage of computers and servers accessed via the Internet. Cloud computing lets you access all your applications and documents from anywhere in the world, freeing you from the confines of the desktop and making it easier for group members in different locations to collaborate. Advances in networking technology and an increase in the need for computing resources have prompted many organizations to outsource their storage and computing needs. This new economic and computing model is commonly referred to as cloud computing and includes various types of services such as: infrastructure as a service (IaaS), where a customer makes use of a service provider's computing, storage or networking infrastructure; platform as a service (PaaS), where a customer leverages the provider's resources to run custom applications; and finally software as service (SaaS), where customers use software that is run on the provider's infrastructure. Cloud infrastructures can be roughly categorized as either private or public. In a private cloud, the infrastructure is managed and owned by the customer and located on premise (i.e., in the customers region of customer data is under its control and is only granted to parties it trusts. In a public cloud the infrastructure is owned and managed by a cloud service provider and is located on premise (i.e., in The service provider's region of control). This means that customer data is outside its control and could potentially be granted to untrusted parties.

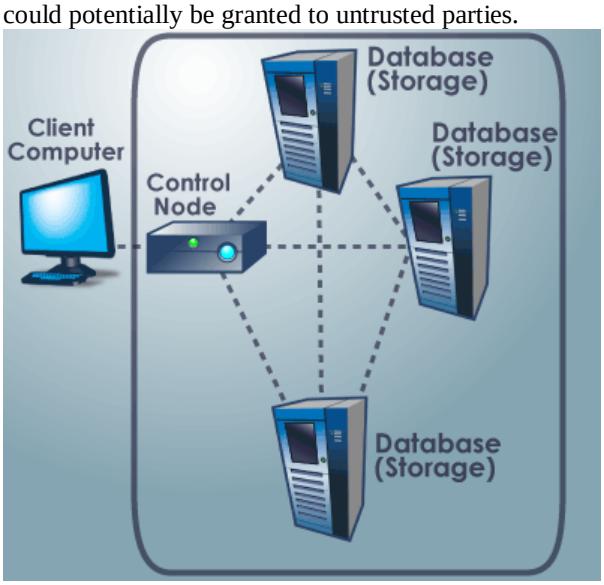

Figure 3. A typical Cloud Storage system architecture

2. Security Services

To address the concerns outlined above and increase the adoption of cloud storage, we argue for Designing a virtual private storage service based on recently developed cryptographic techniques. Such a service should aim to achieve the best of both worlds by providing the security of a private cloud and the functionality and cost savings of a public cloud.

Confidentiality: the cloud storage provider does not learn any information about customer data.

Integrity: any unauthorized modification of customer data by the cloud storage provider can be detected by the customer,while retaining the main benefits of a public storage service:

Availability: customer data is accessible from any machine and at all times

Reliability: customer data is reliably backed up.

Efficient retrieval: data retrieval times are comparable to a public cloud storage service.

Data sharing: customers can share their data with trusted parties.An important aspect of a cryptographic storage service is that the securityproperties described above are achieved based on strong cryptographic guarantees as opposed to legal, physical and access control mechanisms.

3. Architecture of a Cryptographic Storage Service

At its core, the architecture consists of three components: a data processor (DP), that processes data before it is sent to the cloud; a data verifier (DV), that checks whether the data in the cloud has been tampered with; and a token generator (TG), that generates tokens that enable the cloud storage provider to retrieve segments of customer data; and a credential generator that implements an access control policy by issuing credentials to the various parties in the system (these credentials will enable the parties to decrypt encrypted files according to the policy).

3.1. A Consumer Architecture

Consider three parties: a user Alice that stores her data in the cloud; a user Bob with whom Alice wants to share data; and a cloud storage provider that stores Alice's data. To use the service, Alice and Bob begin by downloading a client application that consists of a data processor, a data verifier and a token generator. Upon its first execution, Alice's application generates a cryptographic key. We will refer to this key as a master key and assume it is stored locally on Alice's system and that it is kept secret from the cloud storage provider. Whenever Alice wishes to upload data to the cloud, the data processor is invoked. It attaches some metadata (e.g., current time, size, keywords etc) and encrypts and encodes the data and metadata with a variety of cryptographic primitives. Whenever Alice wants to verify the integrity of her data, the data verifier is invoked. The latter uses Alice's master key to interact with the cloud storage provider and ascertain the integrity of the data. When Alice wants to retrieve data (e.g., all files tagged with keyword urgent") the token generator is invoked to create a token. The token is sent to the cloud storage provider who uses it to retrieve the appropriate (encrypted) files which it returns to Alice. Alice then uses the decryption key to decrypt the files. Data sharing between Alice and Bob proceeds in a similar fashion. Whenever she wishes to share data with Bob, the application invokes the token generator to create an appropriate token, and the credential generator to generate a credential for Bob. Both the token and credential are sent to Bob who, in turn, sends the token to the provider. The latter uses the token to retrieve and return the appropriate encrypted documents which Bob decrypts using his credential.

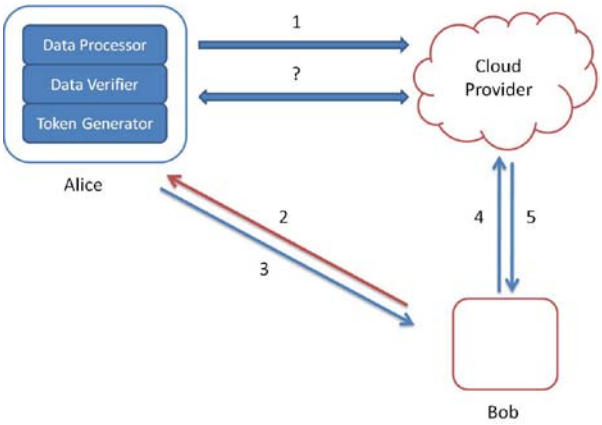

Figure 1: Alice's data processor prepares the data before sending it to the cloud. Bob asks Alice for permission to search for a keyword. Alice's token and credential generators send a token for the keyword and a credential back to Bob. Bob sends the token to the cloud. The cloud uses the token to find the appropriate encrypted documents and returns them to Bob. At any point in time, Alice's data verifier can verify the integrity of the data.

3.2. An Enterprise Architecture

In the enterprise scenario we consider an enterprise MegaCorp that stores its data in the cloud; a business partner PartnerCorp with whom MegaCorp wants to share data; and a cloud storage provider that stores MegaCorp's data. To use the service, MegaCorp deploys dedicated machines within its network. Depending on the particular scenario, these dedicated machines will run various core components. Since these components make use of a master secret key, it is important that they be adequately protected and, in particular, that the master key be kept secret from the cloud storage provider and PartnerCorp. If this is too costly in terms of resources or expertise, management of the dedicated machines (or specific components) can alternatively be outsourced to a trusted entity. In the case of a medium-sized enterprise with enough resources and expertise, the dedicated machines include a data processor, a data verifier, a token generator and a credential generator. To begin, each MegaCorp and PartnerCorp employee receives a credential from the credential generator. These credentials will reflect some relevant information about the employees such as their organization or team or role. Whenever a MegaCorp employee generates data that needs to be stored in the cloud, it sends the data together with an associated decryption policy to the dedicated machine for processing. The decryption policy specifies the type of credentials necessary to decrypt the data (e.g., only members of a particular team). To retrieve data from the cloud (e.g., all files generated by a particular employee), an employee re requests an appropriate token from the dedicated machine. The employee then sends the token to the cloud provider who uses it to find and return the appropriate encrypted files which the employee decrypts using his credentials. Whenever MegaCorp wants to verify the integrity of the data, the dedicated machine's data verifier is invoked. The latter uses the master secret key to interact with the storage provider and ascertain the integrity of the data. Now consider the case where a PartnerCorp employee needs access to MegaCorp's data. The employee authenticates itself to MegaCorp's dedicated machine and sends it a keyword. The latter verifies that the particular search is allowed for this PartnerCorp employee. If

so, the dedicated machine returns an appropriate token which the employee uses to recover the appropriate (encrypted) files from the service provider. It then uses its credentials to decrypt the file. This process is illustrated in Figure 3. Similarly to the consumer architecture, it is imperative that all components be either open-source or implemented by someone other than the cloud service provider.

In the case that MegaCorp is a very large organization and that the prospect of running and maintaining enough dedicated machines to process all employee data is infeasible, consider the following slight variation of the architecture described above. More precisely, in this case the dedicated machines only run data verifiers, token generators and credential generators while the data processing is distributed to each employee. This is illustrated in Figure 4.

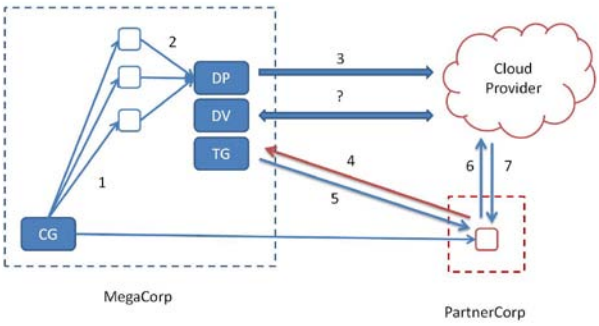

Figure 3: (1) Each MegaCorp and PartnerCorp employee receives a credential; (2) MegaCorp employees send their data to the dedicated machine; (3) the latter processes the data using the data processor before sending it to the cloud; (4) the PartnerCorp employee sends a keyword to MegaCorp's dedicated machine ; (5) the dedicated machine returns a token; (6) the PartnerCorp employee sends the token to the cloud; (7) the cloud uses the token to find the appropriate encrypted documents and returns them to the employee. More precisely, in this case the dedicated machines only run data verifiers, token generators and credential generators while the data processing is distributed to each employee.

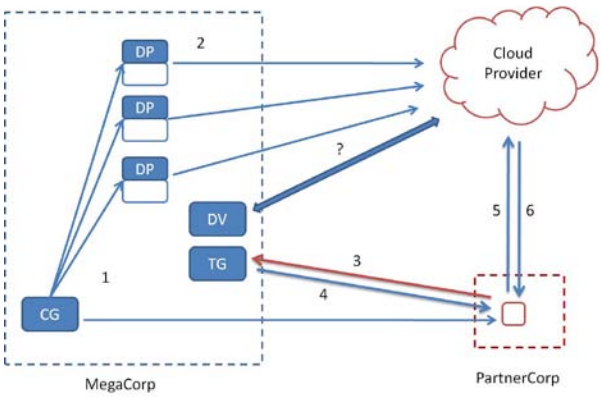

Figure 4: (1) Each MegaCorp and PartnerCorp employee receives a credential; (2) MegaCorp employees process their data using their own data processors and send them to the cloud; (3) the PartnerCorp employee sends a keyword to MegaCorp's dedicated machine; (4) the latter returns a token; (5) the employee sends the token to the cloud; (6) the cloud uses the token to find the appropriate encrypted documents and returns them to the employee. At any point in time, MegaCorp's data verifier can check the integrity of MegaCorp's data.

4. Benefits of a Cryptographically Secured Storage Service

1. *Confidentiality Assurance*

In a cryptographic storage service, the data is encrypted on-premise by the data processor(s). This way, customers can be assured that the confidentiality of their data is preserved irrespective of the actions of the cloud storage provider. This greatly reduces any legal exposure for both the customer and the provider.

2. Geographic restrictions

In a cryptographic storage service data is only stored in encrypted form so any law that pertains to the stored data has little to no effect on the customer. This reduces legal exposure for the customer and allows the cloud storage provider to make optimal use of its storage infrastructure, thereby reducing costs.

3. *Subpoenas*

In a cryptographic storage service, since data is stored in encrypted form and since the customer retains possession of all the keys, any request for the (unencrypted) data must be made directly to the customer.

4. Reducing Risk of Security Breaches

Even if a cloud storage provider implements strong security practices there is always the possibility of a security breach. If this occurs the customer may be legally responsible. In a cryptographic storage service data is encrypted and data integrity can be verified at any time. Therefore, a security breach poses little to no risk for the customer.

5. Data retention and destruction

In many cases a customer may be responsible for the retention and destruction of the data it has collected. If this data is stored in the cloud, however, it can be difficult for a customer to ascertain the integrity of the data or to verify whether it was properly discarded. A cryptographic storage service alleviates these concerns since data integrity can be verified and since the information necessary to decrypt data (i.e., the master key) is kept on-premise. Secure data erasure can be effectively achieved by just erasing the master key.

5. Implementing the Core Components

The core components of a cryptographic storage service can be implemented using a variety of techniques, some of which were developed specifically for cloud storage.

5.1. Searchable Encryption

At a high level, a searchable encryption scheme provides a way to encrypt a search index so that its contents are hidden except to a party that is given appropriate tokens. More precisely, consider a search index generated over a collection of files (this could be a full-text index or just a keyword index). Using a searchable encryption scheme, the index is encrypted in such a way that (1) given a token for a keyword one can retrieve pointers to the encrypted files that contain the keyword; and (2) without a token the contents of the index are hidden. In addition, the tokens can only be generated with knowledge of a secret key and the retrieval procedure reveals nothing about the files or the keywords except that the files contain a keyword in common.

5.1.1. Symmetric searchable encryption

SSE is appropriate in any setting where the party that searches over the data is also the one who generates it.

The security guarantees provided by SSE are, roughly speaking, the following:

1. without any tokens the server learns nothing about the data except its length.
2. given a token for a keyword w, the server learns which (encrypted) documents contain w without learning w.

5.1.2. Asymmetric searchable encryption (ASE)

ASE schemes are appropriate in any setting where the party searching over the data is different from the party that generates it.

The security guarantees provided by ASE are the following:

1. without any tokens the server learns nothing about the data except its length.
2. given a token for a keyword w, the server learns which (encrypted) documents contain w.

5.1.3. Efficient ASE (ESE)

ESE schemes are appropriate in any setting where the party that searches over the data is different from the party that generates it and where the keywords are hard to guess.

5.1.4. Multi-user SSE (MSSE)

MSSE schemes are appropriate in any setting where many parties wish to search over data that is generated by a single party.

5.2. Attribute-based Encryption

It allows the specification of a decryption policy to be associated with a cipher text. A user can then encrypt a message under a public key and a policy. Decryption will only work if the attributes associated with the decryption key match the policy used to encrypt the message. Attributes are qualities of a party that can be established through relevant credentials.

5.3. Proofs of Storage

A proof of storage is a protocol executed between a client and a server with which the server can prove to the client that it did not tamper with its data. The client begins by encoding the data before storing it in the cloud. From that point on, whenever it wants to verify the integrity of the data it runs a proof of storage protocol with the server. The main benefits of a proof of storage are that (1) they can be executed an arbitrary number of times; and (2) the amount of information exchanged between the client and the server is extremely small and independent of the size of the data. Proofs of storage can be either privately or publicly verifiable. Privately verifiable proofs of storage only allow the client (i.e., the party that encoded the file) to verify the integrity of the data. With a publicly verifiable proof of storage, on the other hand, anyone that possesses the client's public key can verify the data's integrity.

6. References

[1] G. Ateniese, S. Kamara, and J. Katz. Proofs of storage from homomorphic identification protocols. In To appear in Advances in Cryptology - ASIACRYPT '09, Lecture Notes in Computer Science. Springer, 2009.

[2] Luis M.Vaquero,Luis Rodero-Merino, Jua critical areas of focus in cloud computing. Technical report, Cloud Security Alliance, April 2009.

[5] J. Baek, R. Safavi-Naini, and W. Susilo. On the integration of public key data encryption and public key encryption with keyword search. In International Conference on Information Security (ISC '06), volume 4176 of Lecture Notes in Computer Science. Springer, 2006.

[6] J. Baek, R. Safavi-Naini, and W. Susilo. Public key encryption with keyword search revisited. In International conference on Computational Science and Its Applications, pages 1249-1259. Springer-Verlag, 2008.

[7] Q.Wang, C.Wang, J. Li, K. Ren, and W. Lou. Enabling public verifiability and data dynamics for storage security in cloud computing. In European Symposium on Research in Computer Security (ESORICS '09), volume 5789 of Lecture Notes in Computer Science, pages 355{370. Springer, 2009.

[8] D. Song, D. Wagner, and A. Perrig. Practical techniques for searching on encrypted data. In IEEE Symposium on Research in Security and Privacy, pages 44-55. IEEE Computer Society, 2000.

[9] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter. Patient controlled encryption: ensuring privacy of

electronic medical records. In ACM workshop on Cloud computing security (CCSW '09), pages 103-114. ACM, 2009.

[10] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. Skeith. Public-key encryption that allows PIR queries. In A. Menezes, editor, Advances in Cryptology - CRYPTO '07, volume 4622 of Lecture Notes in Computer Science, pages 50-67. Springer, 2007.

[11] J. W. Byun, H. S. Rhee, H.-A. Park, and D. H. Lee. Offline keyword guessing attacks on recent keyword search schemes over encrypted data. In Secure Data Management, volume 4165 of Lecture Notes in Computer Science, pages 75-83. Springer, 2006.

[12] Storage Networking Industry Association.Cloud Storage for Cloud Computing,Jun.2009.

[13] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In ACM conference on Computer and communications security (CCS '07), pages 195-203. ACM, 2007.

[14] K. Zetter. Compay caught in texas data center raid loses suit against FBI. Wired Magazine, April 2009.

[15] T. Fuhr and P. Paillier. Decryptable searchable encryption. In International Conference on Provable Security, volume 4784 of Lecture Notes in Computer Science, pages 228-236. Springer, 2007