DECENTRALIZED ACCESS CONTROL WITH ANONYMOUS AUTHENTICATION OF DATA STORED IN CLOUDS

Najma.K¹, J. Stanly Jayaprakash²

¹(Department of CSE, PG Scholar, Mahendra Institute of Technology, Namakkal-637503, Tamilnadu, najmak20@gmail.com)
²(Department of CSE, HOD, Mahendra Institute of Technology, Namakkal-637503, Tamilnadu, hodcse@mahendratech.org)

Abstract—Data sharing is an important functionality in cloud storage. In this paper, 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.

Keywords—constant-size ciphertexts; security analysis; the first public-key patient-controlled encryption; data filtering mechanism

1. INTRODUCTION

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 owner’s 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 re-lying 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.

Assume that Alice puts all her private photos on Dropbox, and she does not want to expose her photos to everyone. Due to various data leakage possibility Alice cannot feel relieved by just relying on the privacy protection mechanisms provided by Dropbox, so she encrypts all the photos using her own keys before uploading. One day, Alice’s friend, Bob, asks her to share the photos taken over all these years which Bob appeared in. Alice can then use the share function of Dropbox, but the problem now is how to delegate the decryption rights for these photos to Bob. A possible option Alice can choose is to securely send Bob the secret keys involved. Naturally, there are two extreme ways for her under the traditional encryption paradigm. Encryption keys also come with two flavors — symmetric key or asymmetric (public) key. Using symmetric encryption, when Alice wants the data to be originated from a third party, she has to give the encryptor her secret key; obviously, this is not always desirable. By contrast, the encryption key and decryption key are different in public-key encryption. The use of public-key encryption gives more flexibility for our applications. For example, in enterprise settings, every employee can up-load encrypted data on the cloud storage server without the knowledge of the company’s master-secret key.

Stage 2: Model building and validation: This stage involves considering various models and choosing the best one based on their predictive performance (i.e., explaining the variability in question and producing stable results across samples). This may sound like a simple operation, but in fact, it sometimes involves a very elaborate process. Stage 3:
Deployment. That final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

The concept of Data Extraction is becoming increasingly popular as a business information management tool where it is expected to reveal knowledge structures that can guide decisions in conditions of limited certainty. Recently, there has been increased interest in developing new analytic techniques specifically designed to address the issues relevant to business Data Mining (e.g., Classification Trees), but Data Mining is still based on the traditional principles of statistics including the traditional Exploratory Data Analysis (EDA) and modeling and it shares with them both some components of its general approaches and specific techniques.

2. LITERATURE SURVEY

In cloud computing, remote data integrity checking is an important security problem. The clients’ massive data is outside his control. The malicious cloud Cloud Server may corrupt the clients’ data in order to gain more benefits. Many researchers proposed the corresponding system model and security model. In 2007, provable data possession (PDP) paradigm was proposed by Ateniese et al.. In the PDP model, the verifier can check remote data integrity with a high probability. Based on the RSA, they designed two provably secure PDP schemes.

After that, Ateniese et al. proposed dynamic PDP model and concrete scheme although it does not support insert operation. In order to support the insert operation, in 2009, Erway et al. proposed a full-dynamic PDP scheme based on the authenticated flip table. The similar work has also been done by F. Seb’ e et al. PDP allows a verifier to verify the remote data integrity without retrieving or downloading the whole data. It is a probabilistic proof of possession by sampling random set of blocks from the Cloud Server, which drastically reduces I/O costs. The verifier only maintains small metadata to perform the integrity checking. PDP is an interesting remote data integrity checking model. In 2012, Wang proposed the security model and concrete scheme of proxy PDP in public clouds. At the same time, Zhu et al. proposed the cooperative PDP in the multi-cloud storage. Following Ateniese et al.’s pioneering work, many remote data integrity checking models and protocols have been proposed.

In 2008, Shacham presented the first proof of retrievability (POR) scheme with provable security. In POR, the verifier can check the remote data integrity and retrieve the remote data at any time. On some cases, the client may delegate the remote data integrity checking task to the third party. It results in the third party auditing in cloud computing. One of benefits of cloud storage is to enable universal data access with independent geographical locations. This implies that the end devices may be mobile and limited in computation and storage. Efficient integrity checking protocols are more suitable for cloud clients equipped with mobile end devices.

In identity-based public key cryptography, this paper focuses on distributed provable data possession in multi-cloud storage. The protocol can be made efficient by eliminating the certificate management.

3. EXISTING METHOD

In the Existing system we use Iolus approach proposed the notion of hierarchy subgroup for scalable and secure multi-cloud. In this method, a large communication group is divided into smaller subgroups. Each subgroup is treated almost like a separate multi-cloud group and is managed by a trusted group security intermediary (IB-DPDP). IB-DPDP connect between the subgroups and share the subgroup key with each of their subgroup members. IB-DPDPs act as message relays and key translators between the subgroups by receiving the multi-cloud messages from one subgroup, decrypting them and then re-multi-clouding to the next subgroup after encrypting them by the subgroup key of the next subgroup. The IB-DPDP are also grouped in a top-level group that is managed by a group security controller (GSC).

When a group member joins or leaves only affect subgroup only while the other subgroup will not be affected. It has the drawback of affecting data path. This occurs in the sense that there is a need for translating the data that goes from one subgroup, and thereby one key, to another. This becomes even more problematic when it takes into account that the IB-DPDP has to manage the subgroup and perform the translation needed. The IB-DPDP may thus become the bottleneck.

4. PROPOSED SYSTEM

The advantages over the existing system are, we use an identity tree instead of key tree in our scheme. Each node in the identity tree is associated with an identity. The leaf node’s identity is corresponding to the user’s identity and the intermediate node’s identity is generated by its children’s identity. Hence, in an identity tree, an intermediate node represents set users in the sub tree rooted at this node. We propose a novel multi-cloud Authentication protocol, namely IBE, including two schemes.

The basic scheme (IBE) eliminates the correlation among data’s and thus provides the perfect resilience to data security, and it is also efficient in terms of latency, computation, and communication overhead due to an efficient cryptographic primitive called batch signature, which supports the authentication of any number of Entities simultaneously. We also present an enhanced scheme IBE-E, which combines the basic scheme with a data filtering mechanism to alleviate the DoS impact while preserving the perfect resilience to data security. The keys used in each subgroup can be generated by a group of IBE on Multi cloud storage Key Generation centers (IBE) in parallel.

All the members in the same subgroup can compute the same subgroup key though the keys for them are generated by different KGCs. This is a desirable feature especially for the large-scale network systems, because it minimizes the problem of concentrating the workload on a single entity.
5. EXPERIMENTAL SETUP

Protecting privacy is an alarming issue in the distributed systems. Henceforth, our multi-attribute based technique in key generation concept can be used as a sound solution to construct privacy preserved data transfer and access control concepts in distributed systems. Users can obtain keys from authorities. As a result, our scheme can provide the following important properties: 1. access tree; 2) Users cannot be impersonated as they can obtain keys.

6. ENHANCED IDENTITY BASED ENCRYPTION:

Identity-based encryption (IBE) is an exciting alternative to public-key encryption. The senders using an IBE do not need to look up the public keys and the corresponding certificates of the receivers, the identities (e.g. emails or IP addresses) of the latter are alone enough to encrypt. Any setting, PKI- or identity-based, must provide a means that the users must be revoked from the system.

The efficient revocation is a studied problem in the traditional IBE setting. However, in the setting of IBE, there is a need for the study of revocation mechanisms. The most important practical solution requires the senders to also use time periods while encrypting, and all the receivers (regardless of whether their keys have been compromised or not) need to update their private keys regularly by contacting the trusted authority. We note that this solution does not scale well as the number of users in the system increases, the work on key updates becomes a bottleneck. Any setting, IBE- or identity-based, must provide the means for revocation of the users from the system. IBE scheme that significantly improves key-update efficiency on the side of the trusted party (from linear to logarithmic in the number of users), while staying efficient for the users. Our schemes should be particularly useful in the settings where a large number of users is involved and here scalability is an issues.

7. CONCLUSION

We focused on the problem of verifying if an untrusted server stores a client’s data. We introduced a model for provable data possession, in which it is desirable to minimize the file block accesses, the computation on the server, and the client-server communication. Our solutions for PDP fit this model: They incur a low (or even constant) overhead at the server and require a small, constant amount of communication per challenge. They allow verifying data possession without having access to the actual data file. Experiments show that our schemes, which offer a probabilistic possession guarantee by sampling the server’s storage, make it practical to verify possession of large data sets. Previous schemes that do not allow sampling are not practical when PDP is used to prove possession of large amounts of data. Our experiments show that such schemes also impose a significant I/O and computational burden on the server.

REFERENCES


