Identity-Based Distributed Provable Data Possession in Multi-Cloud Storage

Abstract—Remote data integrity checking is of crucial importance in cloud storage. It can make the clients verify whether their outsourced data is kept intact without downloading the whole data. In some application scenarios, the clients have to store their data on multi-cloud servers. At the same time, the integrity checking protocol must be efficient in order to save the verifier’s cost. From the two points, we propose a novel remote data integrity checking model: ID-DPDP (identity-based distributed provable data possession) in multi-cloud storage. The formal system model and security model are given. Based on the bilinear pairings, a concrete ID-DPDP protocol is designed. The proposed ID-DPDP protocol is provably secure under the hardness assumption of the standard CDH (computational Diffie-Hellman) problem. In addition to the structural advantage of elimination of certificate management, our ID-DPDP protocol is also efficient and flexible. Based on the client’s authorization, the proposed ID-DPDP protocol can realize private verification, delegated verification and public verification.

Index Terms—Cloud computing, Provable data possession, Identity-based cryptography, Distributed computing, Bilinear pairings

I. INTRODUCTION

Over the last years, cloud computing has become an important theme in the computer field. Essentially, it takes advantage of information processing as a service, such as storage, computing, etc. It relieves the burden for storage management, universal data access with independent geographical locations. At the same time, it avoids of capital expenditure on hardware, software, and personnel maintenances, etc. Thus, cloud computing attracts more intention from the enterprises.

The foundations of cloud computing lie in the outsourcing of computing tasks to the third party. It entails the security risks in terms of confidentiality, integrity and availability of data and service. The issue to convince the cloud clients that their data are kept intact is especially vital since the clients do not store these data locally. Remote data integrity checking is a primitive to address this issue. For the general case, when the client stores his data on multi-cloud servers, the distributed storage and integrity checking are indispensable. On the other hand, the integrity checking protocol must be efficient in order to make it suitable for capacity-limited end devices. Thus, based on distributed computation, we will study distributed remote data integrity checking model and present the corresponding concrete protocol in multi-cloud storage.

A. Motivation

We consider an ocean information service corporation Cor in the cloud computing environment. Cor can provide the following services: ocean measurement data, ocean environment monitoring data, hydrological data, marine biological data, GIS information, etc. Besides of the above services, Cor has also some private information and some public information, such as the corporation’s advertisement. Cor will store these different ocean data on multiple cloud servers. Different cloud service providers have different reputation and charging standard. Of course, these cloud service providers need different charges according to the different security-levels. Usually, more secure and more expensive. Thus, Cor will select different cloud service providers to store its different data. For some sensitive ocean data, it will copy these data many times and store these copies on different cloud servers. For the private data, it will store them on the private cloud server. For the public advertised ocean data, it will store them on the cheap public cloud server. At last, Cor stores its whole data on the different cloud servers according to their importance and sensitivity. Of course, the storage selection will take account in the Cor’s profits and losses. Thus, the distributed cloud storage is indispensable. In multi-cloud environment, distributed provable data possession is an important element to secure the remote data.

In PKI (public key infrastructure), provable data possession protocol needs public key certificate distribution and management. It will incur considerable overheads since the verifier will check the certificate when it checks the remote data integrity. In addition to the heavy certificate verification, the system also suffers from the other complicated certificate management such as certificates generation, delivery, revocation, renewals, etc. In cloud computing, most verifiers only have low computation capacity. Identity-based public key cryptography can eliminate the complicated certificate management. In order to increase the efficiency, identity-based provable data possession is more attractive. Thus, it will be very meaningful to study the ID-DPDP.

B. Related work

In cloud computing, remote data integrity checking is an important security problem. The clients’ massive data is outside his control. The malicious 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. [1]. 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 [2] 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 [3]. The similar work has also been done by F. Sebék et al. [4]. PDP allows a verifier to verify the remote data.
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2014.1, IEEE Transactions on Services Computing

integrity without retrieving or downloading the whole data. It is a probabilistic proof of possession by sampling random set of blocks from the 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 [5]. At the same time, Zhu et al. proposed the cooperative PDP in the multi-cloud storage [6].

Following Atienese et al.’s pioneering work, many remote data integrity checking models and protocols have been proposed [7], [8], [9], [10], [11], [12]. In 2008, Shamch presented the first proof of retrievability (POR) scheme with provable security [13]. In POR, the verifier can check the remote data integrity and retrieve the remote data at any time. The state of the art can be found in [14], [15], [16], [17]. 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 [18], [19], [20], [21]. 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.

C. Contributions

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. We propose the new remote data integrity checking model: ID-DPDP. The system model and security model are formally proposed. Then, based on the bilinear pairings, the concrete ID-DPDP protocol is designed. In the random oracle model, our ID-DPDP protocol is provably secure. On the other hand, our protocol is more flexible besides the high efficiency. Based on the client’s authorization, the proposed ID-DPDP protocol can realize private verification, delegated verification and public verification.

D. Paper Organization

The rest of the paper is organized as follows. Section II formalizes the ID-DPDP model. Section III presents our ID-DPDP protocol with a detailed performance analysis. Section IV evaluates the security of the proposed ID-DPDP protocol. Finally, Section V concludes the paper.

II. SYSTEM MODEL AND SECURITY MODEL OF ID-DPDP

The ID-DPDP system model and security definition are presented in this section. An ID-DPDP protocol comprises four different entities which are illustrated in Figure 1. We describe them below:

1) Client: an entity, which has massive data to be stored on the multi-cloud for maintenance and computation, can be either individual consumer or corporation.

2) CS (Cloud Server): an entity, which is managed by cloud service provider, has significant storage space and computation resource to maintain the clients’ data.

3) Combiner: an entity, which receives the storage request and distributes the block-tag pairs to the corresponding cloud servers. When receiving the challenge, it splits the challenge and distributes them to the different cloud servers. When receiving the responses from the cloud servers, it combines them and sends the combined response to the verifier.

4) PKG (Private Key Generator): an entity, when receiving the identity, it outputs the corresponding private key.

First, we give the definition of interactive proof system. It will be used in the definition of ID-DPDP. Then, we present the definition and security model of ID-DPDP protocol.

Definition 1 (Interactive Proof System): [22] Let \( c, s : \mathbb{N} \rightarrow \mathbb{R} \) be functions satisfying \( c(n) > s(n) + \frac{1}{p(n)} \) for some polynomial \( p(\cdot) \). An interactive pair \((P, V)\) is called an interactive proof system for the language \( L \), with completeness bound \( c(\cdot) \) and soundness bound \( s(\cdot) \), if

1) Completeness: for every \( x \in L \), \( \Pr[c(P(x)) < 1] \geq c(|x|) \).

2) Soundness: for every \( x \notin L \) and every interactive machine \( B \), \( \Pr[c(B, V(x)) < 1] \leq s(|x|) \).

Interactive proof system is used in the definition of ID-DPDP, i.e., Definition 2.

Definition 2 (ID-DPDP): An ID-DPDP protocol is a collection of three algorithms (Setup, Extract, TagGen) and an interactive proof system (Proof). They are described in detail below.

1) Setup\((k)\): Input the security parameter \( k \), it outputs the system public parameters \( params \), the master public key \( mpk \) and the master secret key \( msk \).

2) Extract\((k, params, mpk, msk, ID)\): Input the public parameters \( params \), the master public key \( mpk \), the master secret key \( msk \), and the identity \( ID \) of a client, it outputs the private key \( sk_{ID} \) that corresponds to the client with the identity \( ID \).

3) TagGen\((sk_{ID}, F_i, P)\): Input the private key \( sk_{ID} \), the block \( F_i \) and a set of \( CS = \{CS_j\} \), it outputs the tuple \( \{\phi_i, (F_i, T_i)\} \), where \( \phi_i \) denotes the \( i \)-th record of metadata, \( (F_i, T_i) \) denotes the \( i \)-th block-tag pair. Denote all the metadata \( \{\phi_i\} \) as \( \phi \).

Fig. 1. The System Model of ID-DPDP