4-2017

Identity-based data outsourcing with comprehensive auditing in clouds

Yujue WANG
Singapore Management University

Qianhong WU
Beihang University

Bo QIN
Renmin University of China

Wenchang SHI
Renmin University of China

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

See next page for additional authors

Published in IEEE Transactions on Information Forensics and Security, 2017 Feb, Volume 12, Issue 4, pp 940-952.
https://doi.org./10.1109/TIFS.2016.2646913

Identity-Based Data Outsourcing With Comprehensive Auditing in Clouds

Yujue Wang, Qianhong Wu, *Member, IEEE*, Bo Qin, Wenchang Shi,
Robert H. Deng, *Fellow, IEEE*, and Jiankun Hu

***Abstract*— Cloud storage system provides facilitative file storage and sharing services for distributed clients. To address integrity, controllable outsourcing, and origin auditing concerns on outsourced files, we propose an *identity-based data outsourcing* (IBDO) scheme equipped with desirable features advantageous over existing proposals in securing outsourced data. First, our IBDO scheme allows a user to authorize dedicated proxies to upload data to the cloud storage server on her behalf, e.g., a company may authorize some employees to upload files to the company's cloud account in a controlled way. The proxies are identified and authorized with their recognizable identities, which eliminates complicated certificate management in usual secure distributed computing systems. Second, our IBDO scheme facilitates comprehensive auditing, i.e., our scheme not only permits regular integrity auditing as in existing schemes for securing outsourced data, but also allows to audit the information on data origin, type, and consistence of outsourced files. Security analysis and experimental evaluation indicate that our IBDO scheme provides strong security with desirable efficiency.**

***Index Terms*— Cloud storage, data outsourcing, proof of storage, remote integrity proof, public auditing.**

Manuscript received April 2, 2016; revised August 8, 2016 and November 29, 2016; accepted December 10, 2016. Date of publication December 30, 2016; date of current version January 30, 2017. This work was supported in part by the Natural Science Foundation of China under Project 61672083, Project 61370190, Project 61272501, Project 61402029, Project 61472429, Project 61202465, and Project 61532021, in part by the Beijing Natural Science Foundation under Project 4132056, and in part by the Guangxi Natural Science Foundation under Project 2013GXNSFBB053005. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mauro Conti.

Y. Wang is with the School of Information Systems, Singapore Management University, Singapore 188065, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: yjwang@smu.edu.sg).

Q. Wu is with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China, and also with the State Key Laboratory of Cryptology, Beijing 100878, China (e-mail: qianhong.wu@buaa.edu.cn).

B. Qin is with the Key Laboratory of Data Engineering and Knowledge Engineering, School of Information, Ministry of Education, Renmin University of China, Beijing 100872, China, and also with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: bo.qin@ruc.edu.cn).

W. Shi is with the Key Laboratory of Data Engineering and Knowledge Engineering, School of Information, Ministry of Education, Renmin University of China, Beijing 100872, China (e-mail: wenchangg@ruc.edu.cn).

R. H. Deng is with the School of Information Systems, Singapore Management University, Singapore 188065 (e-mail: robertdengg@smu.edu.sg).

J. Hu is with the School of Engineering and Information Technology, University of New South Wales, Defence Force Academy, Canberra 2600, Australia (e-mail: j.hu@adfa.edu.au).

I. Introduction

CLOUD platform provides powerful storage services to individuals and organizations [1]. It brings great benefits of allowing on-the-move access to the outsourced files, simultaneously relieves file-owners from complicated local storage management and maintenance [2]. However, some security concerns may impede users to use cloud storage. Among them, the *integrity* of outsourced files is considered as a main obstacle [3], since the users will lose physical control of their files after outsourced to a cloud storage server maintained by some cloud service provider (CSP). Thus, the file-owners may worry about whether their files have been tampered with, especially for those of importance.

Considerable efforts have been made to address this issue. Among existing proposals, *provable data possession* (PDP) [4] is a promising approach in proof of storage (PoS). With PDP, the file-owner only needs to retain a small amount of parameters of outsourced files and a secret key. To check whether or not the outsourced files are kept intact, the file-owner or an auditor can challenge the cloud server with low communication overheads and computation costs. If some part of the file has been altered or deleted, for example, due to random hardware failures, the cloud storage server would not be able to prove the data integrity to convince the clients.

We observe two critical issues not well addressed in existing proposals. First, most schemes lack a controlled way of delegatable outsourcing. One may note that many cloud storage systems (e.g., Amazon, Dropbox, Google Cloud storage) allow the account owner to generate signed URLs using which any other designated entity can upload, and modify content on behalf of the user. However, in this scenario, the delegator cannot validate whether or not the authorized one has uploaded the file as specified or verify whether or not the uploaded file has been kept intact. Hence, the delegator has to fully trust the delegatees and the cloud server. In fact, the file-owner may not only need to authorize some others to generate files and upload to a cloud, but also need to verifiably guarantee that the uploaded files have been kept unchanged. For instance, in *Electronic Health Systems* (EHS) [5], [6], when consulting a doctor, the patient needs to delegate her doctor to generate *electronic health records* (EHRs) and store them at a remote EHRs center maintained by a CSP [7]. In another typical scenario of cloud-aided office applications, a group of engineers in different places may fulfill a task in cooperation. The group leader can create a cloud storage account and authorize

TABLE I
COMPARISON WITH EXISTING RELATED WORKS

Schemes	Delegated data outsourcing	Certificate-Freeness	Origin Auditing	Consistence Validation	Public Verifiability
Shacham and Waters [9]	×	×	×	×	√
Wang et al. [10]	×	×	×	×	×
Wang et al. [11]	×	×	×	×	√
Chen et al. [12]	×	×	×	×	√
Wang [13]	×	×	×	×	×
Shen and Tzeng [14]	×	×	×	×	×
Armknecht et al. [15]	×	×	×	×	×
Wang et al. [16]	×	√	×	×	√
Ours	√	√	√	√	√

the members with secret warrants. The behavior of the group members and the cloud server should be verifiable.

Second, existing PoS-like schemes, including PDP and *Proofs of Retrievability* (PoR) [8], do not support data log related auditing in the process of data possession proof. The logs are critical in addressing disputes in practice. For example, when the patient and doctor in EHS get involved medical disputes, it would be helpful if some specific information such as outsourcer, type and generating time of the outsourced EHRs are auditable. However, there exist no PoS-like schemes that can allow validation of these important information in a multi-user setting.

A. Our Contributions

To address the above issues for securing outsourced data in clouds, this paper proposes an *identity-based data outsourcing* (IBDO) system in a multi-user setting. Compared to existing PoS like proposals, our scheme has the following distinguishing features.

- **Identity-based outsourcing.** A user and her authorized proxies can securely outsource files to a remote cloud server which is not fully trustable, while any unauthorized ones cannot outsource files on behalf of the user. The cloud clients, including the file-owners, proxies and auditors, are recognized with their identities, which avoids the usage of complicated cryptographic certificates. This delegate mechanism allows our scheme to be efficiently deployed in a multi-user setting.
- **Comprehensive auditing.** Our IBDO scheme achieves a strong auditing mechanism. The integrity of outsourced files can be efficiently verified by an auditor, even if the files might be outsourced by different clients. Also, the information about the origin, type and consistence of outsourced files can be publicly audited. Similar to existing publicly auditable schemes, the comprehensive auditability has advantages to allow a public common auditor to audit files owned by different users, and in case of disputes, the auditor can run the auditing protocol to provide convincing judicial witnesses without requiring disputing parties to be corporative.
- **Strong security guarantee.** Our IBDO scheme achieves strong security in the sense that: (1) it can detect any unauthorized modification on the outsourced files and (2) it can detect any misuse/abuse of the delegations/authorizations. These security properties are formally proved against active colluding attackers. To the best of our knowledge, this is the first scheme that simultaneously achieves both goals.

A thorough comparison of our scheme with several related schemes is shown in Table I in terms of delegated data outsourcing, certificate-freeness, data origin auditing, data consistence validation and public verifiability. We also conduct extensive experiments on our proposed IBDO scheme and make comparisons with Shacham and Waters' (SW) PoR scheme. Both theoretical analyses and experimental results confirm that the IBDO proposal provides resilient security properties without incurring any significant performance penalties.

B. Related Work

The notion of PDP introduced by Ateniese *et al.* [4] allows an auditor to check the integrity of an outsourced file without retrieving the entire file from the cloud server; at the same time the server does not need to access the entire file for answering integrity queries. A subsequent work in [17] supports modification and deletion, but not insertion operations on the outsourced data. Yang and Jia [18] presented a scheme to support dynamic update for the outsourced data. Wang *et al.* [19] introduced a third security-mediator into PDP system to generate verifiable metadata on the outsourced files in a blind way, so that the security-mediator learns nothing about the file. Wang *et al.* [20] offloaded the burdensome exponentiations in PDP schemes at the client side by outsourcing the computations to a single computation server.

Using proxy re-signatures, Wang *et al.* [11] proposed a secure cloud storage scheme with user revocation in a multi-user setting, that is, if some user is revoked, then her outsourced data will be re-signed by the cloud storage server. Chen *et al.* [12] investigated the relationship between secure cloud storage and secure networking coding, where a systematic way is presented to construct a secure cloud storage scheme from any secure networking coding protocol. Zhu *et al.* [21] discussed multicloud storage and presented a cooperative PDP scheme which can efficiently support data migration. Wang [22] also considered the multicloud storage scenario and proposed a secure identity-based scheme. Recently, Yu *et al.* [23] studied key-exposure problem in

secure cloud storage. In [24], an identity-based PDP scheme is presented from pre-homomorphic signatures to support group-oriented applications.

Public verifiability is a preferable property for PDP schemes, which allows anyone to audit the outsourced data without knowing the private parameters of the data owner. With this property, the data owner can delegate the rights of integrity auditing to a third party auditor (TPA) without leaking private information. Aforementioned schemes [4], [11], [12], [17], [19], [21], [22] are all publicly verifiable. Jiang *et al.* [25] recently presented a publicly verifiable scheme using the vector commitment technique, which also supports secure user revocation. Fan *et al.* [26], Yu *et al.* [27], and Yu *et al.* [28] considered indistinguishability/privacy on outsourced data against the auditor in auditing integrity. Zhang and Dong's publicly verifiable data outsourcing scheme [29] is proved with tight security reduction in ID-based setting. Zhang *et al.* [30] designed a certificateless public verification scheme which provides stronger security against a malicious auditor.

PoR [8] is also related to our work, which allows the cloud server to convince the clients (file-owners and auditors) that the outsourced files can be successfully retrieved. As sentinels are used for detecting unauthorized modifications, only a limit number of integrity queries on the outsourced files are supported in [8]. Shacham and Waters [9] further presented three PoR schemes with private and public verifiability, which are the first with strict security proofs. Bowers *et al.* [31] investigated PoR in multi-server setting, which strengthes the security and availability of outsourced files in the standard PoR framework.

Another line of related works support delegatable integrity auditing on outsourced files, yet they cannot support controlled delegatable data outsourcing. Wang *et al.* [10] proposed privacy-preserving public auditing cloud storage schemes, in which a secure TPA is introduced to verify the integrity of outsourced files, while TPA can learn nothing about the file's content. In a proxy PDP scheme presented by Wang [13], if the designated auditor is unavailable, then the authorized proxy can be delegated for conducting data possession checking on behalf of the auditor. Shen and Tzeng [14] proposed a delegatable PDP scheme, where a user can delegate integrity auditing capability to a delegatee so that the delegatee can perform auditing protocol on any outsoured files of this user. Armknecht *et al.* [15] studied delegatable auditing for privately auditable PoR schemes, which simultaneously protects against collusion attacks by malicious clients, auditors and cloud servers. Based on a variant of the Schnorr signature, Wang *et al.* [16] proposed a secure data outsourcing scheme in the identity-based setting, however, their scheme also does not support delegated data outsourcing mechanism.

C. Paper Organization

We describe the IBDO system architecture, threat model and security goals in Section II. The framework of IBDO system and the corresponding security model are formalized in Section III. A detailed IBDO construction is presented in Section IV. The security and performance of our IBDO scheme are analyzed in Section V and Section VI, respectively. Finally, Section VIII concludes the paper.

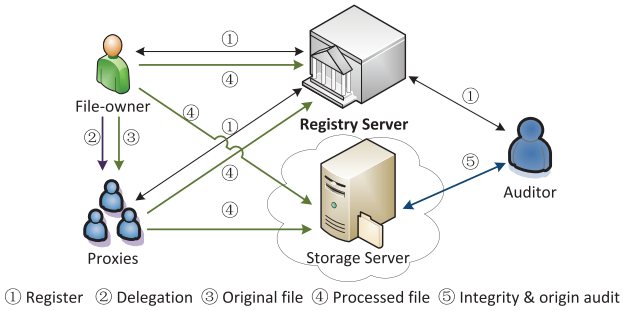


Fig. 1. The architecture of IBDO system

II. System Model

A. System Architecture

The architecture of our IBDO system is shown in Fig. 1. An IBDO system consists of five types of entities, that is, file-owners, proxies, auditors, registry server, and storage server. Generally, the file-owners, proxies and auditors are cloud clients. The registry server is a trusted party responsible for setting up the system and responding to the clients' registration, and also allows the registered clients to store the public parameters of outsourced files. The cloud storage server provides storage services to the registered clients for storing outsourced files. In real-world applications, an organization buys storage services from some CSP, and the IT department of the organization can play the role of a registry server. In this way, the registered clients (employees) can take advantage of the storage services.

The file-owner and her authorized proxies can outsource files to the cloud server. Specifically, on behalf of the owner, the authorized proxy processes the file, sends the processed results to the storage server, and uploads the corresponding public parameters of the file to the registry server. Neither the file-owner nor the proxy is required to store the original file or the processed file locally. The duty of the auditor is to check the integrity of outsourced files and their origin-like general log information by interacting with the cloud storage server without retrieving the entire file.

B. Adversary Model and Security Goals

An IBDO system confronts two types of active attacks. The cloud client may impersonate others, specifically, she may impersonate an owner or another authorized proxy, or abuse a delegation, and in this way she can process a file and outsource it to the storage server in an unwanted way. On the other hand, a malicious storage server may modify or even remove the outsourced files (for example, for saving storage space or due to hardware failures), especially for the rarely accessed files.

Taking into account the above realistic attacks, a secure IBDO system should satisfy the following requirements:

- Dedicated delegation: A delegation issued by a file-owner can only be used by the specific authorized proxy to

outsource specified files in a designated way. Even the authorized proxy cannot abuse it to outsource unspecified files, and multiple proxies cannot cooperatively deduce a valid delegation for a new warrant to outsource an unspecified file.
- Comprehensive auditing: Not only the integrity of the outsourced file, but also the log information about the origin, type and consistence of the outsourced files should be verifiable by the auditors. The integrity auditing ensures that the outsourced files have been kept intact; the other general log information auditing ensures that the file has been outsourced in the designated way. With comprehensive auditing, an IBDO system can provide convincing judicial witnesses to address disputes.

III. Definitions

A. Framework of IBDO System

Formally, an IBDO system consists of five polynomial-time computable algorithms/protocols, that is, Setup, Regst, Dlgtn, IBDOsc, and Audit.

- $\mathsf{Setup}(1^\kappa) \rightarrow (\mathsf{Para}, msk)$: on input 1^κ where κ is a security parameter, the system setup algorithm, which is run by the registry server, generates the public parameter Para for the system and a master secret key msk for the registry server itself.
- $\mathsf{Regst}(\mathsf{Para}, msk, ID_i) \rightarrow sk_i$: on input the public parameter Para, the master secret key msk and an identity ID_i, the register algorithm, which is run by the registry server, generates a private key sk_i for user ID_i. User ID_i should be able to validate sk_i and accept it as her/his private key only if it passes the validation.
- $\mathsf{Dlgtn}(\mathsf{Para}, ID_o, sk_o, ID_p) \rightarrow (W, \sigma_w)$: on input the public parameter Para, an identity ID_o (file-owner) and her private key sk_o, and another identity ID_p (proxy), the delegating outsourcing rights algorithm, which is run by a delegator ID_o, generates a pair of warrant and delegation (W, σ_w) for proxy ID_p. The proxy ID_p should be able to validate (W, σ_w) and accept the delegation only if it passes the validation.
- $\mathsf{IBDOsc}(\mathsf{Para}, W, \sigma_w, sk_p, M) \rightarrow (\tau, M^*)$: on input the public parameter Para, a pair of warrant and delegation (W, σ_w), a private key sk_p and a file M, the proxy data outsourcing algorithm, which is run by an authorized proxy ID_p, generates a file tag τ and a processed file M^* on behalf of the file-owner.
- $\mathsf{Audit}(\mathsf{Para}, \tau) \rightarrow \{0, 1\}$: on input the public parameter Para and a file tag τ, the public auditing protocol, which is jointly run by the auditor and storage server, outputs "1" if the origin and integrity of the outsourced file specified by τ can be verified as true; otherwise it outputs "0".

A secure IBDO scheme must be *sound*, that is, if all entities honestly follow the scheme, then no failure will occur at any stage during the scheme running. Formally, for a security parameter $\kappa \in \mathbb{N}$ and any $(\mathsf{Para}, msk) \leftarrow \mathsf{Setup}(1^\kappa)$, all the following conditions hold:

- For any private key $sk_i \leftarrow \mathsf{Regst}(\mathsf{Para}, msk, ID_i)$ issued by the registry server, it can be validated as true and thus accepted by user ID_i;
- For any pair of warrant and delegation $(W, \sigma_w) \leftarrow \mathsf{Dlgtn}(\mathsf{Para}, ID_o, sk_o, ID_p)$ issued by user ID_o, it can be validated as true and thus accepted by user ID_p;
- For any outsourced file $(\tau, M^*) \leftarrow \mathsf{IBDOsc}(\mathsf{Para}, W, \sigma_w, sk_p, M)$ under a valid delegation σ_w, it should be always audited as true in a round of Audit protocol, that is, $\mathsf{Audit}(\mathsf{Para}, \tau) = 1$.

B. Formal Security Definitions

We present formal security definitions to capture the security requirements described in Section II-B. Two types of probabilistic polynomial-time (PPT) adversaries are used to model the malicious clients and a dishonest storage server. The former may collude to forge or abuse the delegation with regard to some file-owner, while the latter may try to modify the stored files without being caught.

We first define the security against malicious proxies, where a PPT adversary $\mathcal{A}$ is assumed to play the following game with a challenger $\mathcal{C}$.

Setup: On input a security parameter κ, the challenger $\mathcal{C}$ generates (Para, msk) and sends public parameter Para to adversary $\mathcal{A}$.

Queries: Adversary $\mathcal{A}$ can adaptively issue the following queries to $\mathcal{C}$. The challenger maintains the corresponding query lists which are initially empty.

- *Register:* The adversary can ask for private key for any identity ID_i. The challenger generates sk_i and gives it to $\mathcal{A}$. This query means that the attacker can collude with some file-owner or proxy.
- *Delegation:* In each query, the adversary submits a warrant W to $\mathcal{C}$. Note that W contains a delegator identity ID_o and a delegatee identity ID_p. If the private key of ID_o has not been queried before, the challenger will first generate it. Then, the challenger answers with a delegation σ_w. This query implies that the attacker can obtain any normal delegations.
- *Processing file:* In each query, the adversary submits a tuple (W, M) to $\mathcal{C}$. If the private key of ID_p and the delegation in relation to warrant W have not been queried before, the challenger will first generate them. Then, the challenger answers with a processed file M^*. This query implies that the attacker can obtain any processed files.

End-Game: Eventually, the adversary $\mathcal{A}$ outputs a processed file $\widetilde{M}^*$ under $\widetilde{W}$. Note that $\widetilde{M}^*$ corresponds to an original file $\widetilde{M}$ and $\widetilde{W}$ contains a delegator identity $\widetilde{ID}_o$ and a delegatee identity $\widetilde{ID}_p$. We say the adversary $\mathcal{A}$ succeeds if the following conditions hold, no matter whether $\widetilde{M}^*$ can pass integrity checking for $\widetilde{M}$:

- Adversary $\mathcal{A}$ has not been made a registry query on identity $\widetilde{ID}_o$ to get a private key;
- Adversary $\mathcal{A}$ has not been made a delegation query on $\widetilde{W}$;
- Adversary $\mathcal{A}$ has not been made a processing file query that involves $\widetilde{W}$;

- $\tilde{\sigma}_w$ is a valid delegation by delegator $\widetilde{ID}_o$ to proxy $\widetilde{ID}_p$ under $\widetilde{W}$.

Definition 1: An IBDO scheme is secure against adaptive impersonation and misusing delegation attacks if any PPT adversary $\mathcal{A}$ who plays the above game with $\mathcal{C}$ has only negligible probability in winning the game, that is,

$$\Pr[\mathcal{A}_{win}] \leq \epsilon(\kappa)$$

where the probability is taken over all coin tosses made by $\mathcal{C}$ and $\mathcal{A}$.

We proceed to define the security of IBDO scheme against a dishonest storage server. Although the storage server may forge a delegation as it holds a mass of delegations for outsoured files, it can be essentially captured by Definition 1. Thus, the following security game focuses particularly on the integrity guarantee on outsourced files.

Setup: On input a security parameter κ, the challenger $\mathcal{C}$ generates (Para, msk) and sends public parameter Para to adversary $\mathcal{A}$. Then adversary $\mathcal{A}$ adaptively requests processed files in the same way as in Definition 1 by interacting with the challenger $\mathcal{C}$.

Queries: The challenger adaptively carries out the integrity and origin auditing protocol with the adversary $\mathcal{A}$. That is, adversary $\mathcal{A}$ plays the role of a prover, in a way it answers integrity challenges by $\mathcal{C}$ on any outsourced file that it holds.

- The challenger $\mathcal{C}$ generates a challenge for some outsourced file and sends it to $\mathcal{A}$;
- The adversary responds with a proof according to its maintained processed file;
- The challenger verifies the proof and gives the verification results to $\mathcal{A}$.

End-Game: Eventually, the challenger and the adversary carry out a final round of integrity auditing protocol. That is, the challenger sends out a challenge $\widetilde{C}$ for some processed file $\widetilde{M}^*$, and accordingly the adversary returns a proof $\widetilde{P}$. Assume $\widetilde{M}^*$ has been tampered with at the storage server side and the challenge $\widetilde{C}$ touches the corrupted parts of $\widetilde{M}^*$. We say that the adversary $\mathcal{A}$ succeeds in the game if the tuple $(\tilde{\pi}_1, \cdots, \tilde{\pi}_c, \tilde{\sigma})$ in the proof $\widetilde{P}$ is valid for the challenge $\widetilde{C}$ and processed file $\widetilde{M}^*$, while this tuple is different from that generated by $\mathcal{C}$ from the maintained processed file.

Definition 2: An IBDO scheme is secure against modification attacks on outsourced files if any PPT adversary $\mathcal{A}$ has only negligible probability in κ in winnig the above game, that is,

$$\Pr[\mathcal{A}_{win}] \leq \epsilon(\kappa)$$

where the probability is taken over all coin tosses made by $\mathcal{C}$ and $\mathcal{A}$.

IV. An IBDO Scheme

A. An Overview

It is challenging to achieve both proxy data outsourcing and comprehensive auditing functionalities in IBDO. At a first glance, it seems that if the file-owner has delegated her outsourcing rights to some proxy, then the authorized proxy can simply employ the existing PDP/PoR schemes for processing and outsourcing files. However, although this delegation has been signed by the file-owner, there exists a gap that the information of the file-owner is not bounded with the file, which leaves a vulnerability that the proxy may abuse the delegation without being caught. We fill this gap in our IBDO construction.

In our IBDO system, to delegate outsourcing rights to a proxy, the file-owner signs a dedicated *warrant* for the proxy. The warrant may specify *who* can outsource *which* kind of files during *what* time on behalf of the owner, and so on. When a file is processed, it is partitioned into blocks, so as to generate metadada for each block individually. The warrant should be embedded into every metadata, to characterize that the metadata are generated by the authorized proxy. During the execution of integrity and origin auditing protocol, except the aggregate file blocks, the auditor also requests the aggregate metadata and the signed warrant. Both the aggregate metadata and signed warrant should be audited, in this way to conclude that the file is intact and is indeed outsourced by the one as specified in the warrant.

From a technical point of view, we employ Paterson and Schuldt's identity-based signature scheme [32] as building block. The delegation is generated as an identity-based signature on a warrant by their scheme, in this way the delegation can be publicly verified in Audit protocol of IBDO system. Also, we follow the framework due to Shacham and Waters [9] to split the file blocks when generating metadata, which provides a trade-off between storage costs and communication overheads in auditing.

B. Mathematic Background

Our scheme is built on *bilinear groups*. Suppose G and G_T are two cyclic groups with prime order q, where $G = \langle g \rangle$. The group G is a symmetric bilinear group if there exists a bilinear map $\hat{e} : G \times G \to G_T$ such that: (1) *Bilinearity*: $\forall h_1, h_2 \in G$, and $\forall a, b \in Z_q^*$, $\hat{e}(h_1^a, h_2^b) = \hat{e}(h_1, h_2)^{ab}$; (2) *Non-degeneracy*: $\hat{e}(g, g) \neq 1$ is a generator of G_T; (3) *Efficiency*: the map $\hat{e}$ and group operations in G and G_T can be efficiently computed.

CDH Assumption [33]. Suppose $G = \langle g \rangle$ is a cyclic group with prime order q. Given a triplet (g, g^α, g^β) with $\alpha, \beta \in_R Z_q^*$, for any PPT algorithm $\mathcal{A}$, the probability in computing $g^{\alpha\beta}$, i.e., $\Pr\left[\mathcal{A}(g, g^\alpha, g^\beta) = g^{\alpha\beta}\right]$, is negligible.

C. System Setup: Setup

Taking as input a security parameter κ, the registry server randomly picks a cyclic group $G = \langle g \rangle$ with prime order q and a bilinear map $\hat{e} : G \times G \to G_T$. Then, randomly choose an integer $a \in_R Z_q^*$ and elements $g_2, \mu_0, \mu_1, \cdots, \mu_l, \nu_0, \nu_1, \cdots, \nu_\ell, u_1, \cdots, u_{\mathsf{B}} \in_R G$, where B is the upper bound of sector number in file blocks. Computes $g_1 = g^a$ and $msk = g_2^a$. The registry server also chooses three collision-resistant hash functions such as $H_1 : \{0,1\}^* \to \{0,1\}^l$, $H_2 : \{0,1\}^* \to \{0,1\}^\ell$ and $H_3 : \{0,1\}^* \to G$, where the bitstring lengths l and ℓ are determined by κ. The public system parameter is $\mathsf{Para} = (1^\kappa, G, G_T, \hat{e}, q, g, g_1, g_2, \mu_0, \mu_1, \cdots, \mu_l, \nu_0, \nu_1, \cdots, \nu_\ell, u_1, \cdots, u_{\mathsf{B}}, H_1, H_2, H_3)$. The master key msk is kept secret by the registry server.

D. Register: Regst

Using the master secret key msk, the registry server generates a private key sk_i for every user ID_i who may be a file-owner or a proxy in the IBDO system. In detail, the registry server computes

$$\vec{u}_i = (\mathrm{u}_{i,1}, \cdots, \mathrm{u}_{i,l}) \leftarrow H_1(ID_i), \tag{1}$$

picks a random value $t_i \in_R Z_q^*$ and generates sk_i as follows:

$$sk_i = (sk_{i,1}, sk_{i,2}) = (g_2^a \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathrm{u}_{i,j}})^{t_i}, g^{t_i}) \tag{2}$$

Finally, the registry server sends sk_i to ID_i.

The user ID_i can validate $sk_i = (sk_{i,1}, sk_{i,2})$ by computing $\vec{u}_i$ and checking whether

$$\hat{e}(sk_{i,1}, g) \stackrel{?}{=} \hat{e}(g_1, g_2) \cdot \hat{e}(\mu_0 \prod_{j=1}^{l} \mu_j^{\mathrm{u}_{i,j}}, sk_{i,2}) \tag{3}$$

If the equality holds, then ID_i *accepts* the private key sk_i; otherwise, she $fails$ to register.

E. Delegating Outsourcing Rights: Dlgtn

For authorizing a proxy ID_p, the file-owner ID_o generates a warrant W which includes her identity ID_o, the proxy's identity ID_p, the validity period of W, and may also contain some other specifics such as file type, etc. For example, $W = ID_o \| ID_p \| TimePeriod \| FileType$. With Para, the file-owner ID_o evaluates hash function

$$\vec{w} = (\mathrm{w}_1, \cdots, \mathrm{w}_\ell) \leftarrow H_2(W) \tag{4}$$

picks a random value $t_w \in_R Z_q^*$ and produces a delegation using her private key as follows:

$$\sigma_w = (g_2^a \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathrm{u}_{o,j}})^{t_o} \cdot (\nu_0 \prod_{j=1}^{\ell} \nu_j^{\mathrm{w}_j})^{t_w}, g^{t_o}, g^{t_w})$$

The warrant-delegation pair (W, σ_w) is sent to ID_p.

Upon receiving $(W, \sigma_w) = (W, (\alpha, \beta, \gamma))$ from ID_o, the proxy ID_p can validate the delegation. Specifically, he computes the hash values $\vec{u}_o \leftarrow H_1(ID_o)$ and $\vec{w}$ as shown in Equations (1) and (4), and checks whether

$$\hat{e}(\alpha, g) \stackrel{?}{=} \hat{e}(g_1, g_2) \cdot \hat{e}(\mu_0 \prod_{j=1}^{l} \mu_j^{\mathrm{u}_{o,j}}, \beta) \cdot \hat{e}(\nu_0 \prod_{j=1}^{\ell} \nu_j^{\mathrm{w}_j}, \gamma) \tag{5}$$

If the condition holds, then the proxy ID_p *accepts* the delegation from ID_o which means he is successfully authorized with the warrant W; otherwise, he $rejects$ the delegation.

F. Identity-Based Data Outsourcing: IBDOsc

Given a file $M \in \{0,1\}^*$ to be outsourced under a valid warrant-delegation pair (W, σ_w), the authorized proxy ID_p partitions the file M into blocks such that each block comprises c ($1 \le c \le \mathsf{B}$) sectors, that is, $M = (m_{i,j})_{r \times c}$ where $m_{i,j} \in Z_q$. Then, ID_p chooses a random identifier $fid \in_R Z_q^*$ and a random value $t_f \in_R Z_q^*$ for file M, computes $v_f = g^{t_f}$, and sets $\tau_0 = W \| fid \| r \| c \| g^{t_o} \| g^{t_p} \| g^{t_w} \| v_f$. The proxy ID_p proceeds to choose a (one-time) signature scheme $\mathcal{S} = \langle \mathsf{Kgen}, \mathsf{Sig}, \mathsf{Vrf} \rangle$ and generates the file tag for M as $\tau = \tau_0 \| \mathcal{S}.\mathsf{Sig}(\tau_0, tsk) \| tpk$, where $(tpk, tsk) \leftarrow \mathcal{S}.\mathsf{Kgen}(1^\kappa)$.

The proxy ID_p continues to create metadata σ_i ($1 \le i \le r$) for each file block using his private key sk_p as follows

$$\sigma_i = g_2^a \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathrm{u}_{p,j}})^{t_p} \cdot (H_3(W \| fid \| i) \cdot \prod_{j=1}^{c} u_j^{m_{i,j}})^{t_f}$$

Finally, the file tag τ is sent to the registry server, while the processed file M^* that comprises M, fid, $\{\sigma_i\}_{1 \le i \le r}$ and the first entry of delegation σ_w is uploaded to the cloud storage server and removed at the proxy ID_p's local side.

G. Auditing Outsourced Data: Audit

The auditing procedure consists of three phases: *challenge*, *response* and *verification*.

Phase 1 (Challenge): The auditor first makes sure the file tag τ is valid, that is, he retrieves τ from the registry server and performs $\mathcal{S}.\mathsf{Vrf}(\tau_0, tpk)$. If it is invalid, which means this outsourced file cannot be audited, then the protocol aborts; otherwise, the auditor parses τ_0 to obtain the total number r of file blocks. The auditor randomly picks a nonempty subset $I \subseteq [1, r]$ along with a number of random values $s_i \in_R Z_q^*$ for every $i \in I$. After that, the auditor sends the challenge $C = \{(i, s_i) : i \in I\}$ and file identifer fid to the storage server.

Phase 2 (Response): When receiving a challenge C and a file identifier fid, the storage server locates to the corresponding outsourced file M^* and computes

$$\pi_j = \sum_{i \in I} s_i m_{i,j} \bmod q \quad \text{for all } j \in [1, c], \text{ and } \sigma = \prod_{i \in I} \sigma_i^{s_i}.$$

The storage server sends to the auditor a proof P which consists of $\pi_1, \cdots, \pi_c, \sigma$ and the first part of delegation.

Phase 3: (Verification): On receiving a proof P, with Para and τ, the auditor computes $\vec{u}_o \leftarrow H_1(ID_o)$, $\vec{u}_p \leftarrow H_1(ID_p)$, and $\vec{w} \leftarrow H_2(W)$ as shown in Equalities (1) and (4). The auditor checks the validity of σ_w indicated by Equality (5) and σ as follows

$$\hat{e}(\sigma, g) \stackrel{?}{=} \hat{e}(g_1, g_2)^{\sum_{i \in I} s_i} \cdot \hat{e}(\mu_0 \prod_{j=1}^{l} \mu_j^{\mathrm{u}_{p,j}}, g^{t_p})^{\sum_{i \in I} s_i} \cdot \hat{e}(\prod_{i \in I} H_3(W \| fid \| i)^{s_i} \cdot \prod_{j=1}^{c} u_j^{\pi_j}, v_f) \tag{6}$$

If Equality (5) holds, the auditor concludes that ID_p is authorized by ID_o with W. If Equality (6) also holds, she is convinced that the challenged file is intact.

In the above three-phase auditing procedure, the auditor can also simultaneously request for the specifics (e.g., file creation/process/outsourcing time) associated with the warrant W of the audited file. In this case, the response P will additionally contain the requested data specifics, so that they can be checked by the auditor. In fact, the verification on these specifics requires no crypto techniques.

H. Discussions

As remarked in [32], each user ID_i can re-randomize his/her private key $sk_i = (sk_{i,1}, sk_{i,2})$ issued by the registry server, that is, he/she may pick a random value $t_i' \in_R Z_q^*$ and generates sk_i' as follows

$$\begin{aligned} sk_i' = (sk_{i,1}', sk_{i,2}') &= (sk_{i,1} \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{i,j}})^{t_i'}, sk_{i,2}^{t_i'}) \\ &= (g_2^a \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{i,j}})^{t_i + t_i'}, g^{t_i + t_i'}) \end{aligned}$$

In a similar way, the proxy is also able to re-randomize the received delegation. However, when processing and outsourcing a concrete file, both private keys of the file-owner and authorized proxy as well as the corresponding delegation should be fixed, although they might be different for distinct files. In fact, this is realized by inserting $g^{t_o}, g^{t_p}, g^{t_w}$ into the file tag. Also, since v_f is contained in the file tag, the cloud server cannot randomize the metadata.

Besides, although our IBDO scheme is proposed in symmetric bilinear groups, it can also be efficiently instantiated using asymmetric bilinear mapping $\hat{e} : G_1 \times G_2 \to G_T$, that is, by choosing $H_3 : \{0,1\}^* \to G_2$, and letting g be from G_1, and $g_2, \mu_0, \mu_1, \cdots, \mu_l, \nu_0, \nu_1, \cdots, \nu_\ell, u_1, \cdots, u_{\mathsf{B}}$ from G_2.

V. Soundness and Security

We show the soundness of our scheme, that is, if all the entities in the IBDO system behave honestly, then the registration information, delegation and processed file produced by the proposed IBDO scheme can be correctly audited.

Theorem 1: In a successful registration, the client always accepts the private key generated by the registry server. The proxy always accepts the delegation if the corresponding file-owner is honest. If the target outsourced file has not been tampered with, then the proof generated by the storage server will be verified as valid.

Proof: We only need to show that Equation (3), Equation (5) and Equation (6) all hold. The correctness of Equality (3) is straight-forward. Since $\vec{u}_o = (\mathfrak{u}_{o,1}, \cdots, \mathfrak{u}_{o,l})$ and $\vec{w} = (\mathrm{w}_1, \cdots, \mathrm{w}_\ell)$, it follows that

$$\begin{aligned} \hat{e}(\alpha, g) &= \hat{e}(g_2^a, g) \cdot \hat{e}((\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{o,j}})^{t_o}, g) \cdot \hat{e}((\nu_0 \prod_{j=1}^{\ell} \nu_j^{\mathrm{w}_j})^{t_w}, g) \\ &= \hat{e}(g_1, g_2) \cdot \hat{e}(\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{o,j}}, \beta) \cdot \hat{e}(\nu_0 \prod_{j=1}^{\ell} \nu_j^{\mathrm{w}_j}, \gamma) \end{aligned}$$

Therefore, Equality (5) holds.

Note that $\pi_j = \sum_{i \in I} s_i m_{i,j} \bmod q$ for all $j \in [1, c]$ and

$$\begin{aligned} \sigma &= \prod_{i \in I} (g_2^a)^{s_i} \cdot \prod_{i \in I} ((\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{p,j}})^{t_p})^{s_i} \\ &\quad \cdot \prod_{i \in I} ((H_3(W \| fid \| i) \cdot \prod_{j=1}^{c} u_j^{m_{i,j}})^{t_f})^{s_i} \\ &= (g_2^a)^{\sum_{i \in I} s_i} \cdot ((\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{p,j}})^{t_p})^{\sum_{i \in I} s_i} \\ &\quad \cdot (\prod_{i \in I} H_3(W \| fid \| i)^{s_i} \cdot \prod_{j=1}^{c} u_j^{\sum_{i \in I} s_i m_{i,j}})^{t_f} \end{aligned}$$

We have

$$\begin{aligned} \hat{e}(\sigma, g) &= \hat{e}(g_1, g_2)^{\sum_{i \in I} s_i} \cdot \hat{e}(\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{p,j}}, g^{t_p})^{\sum_{i \in I} s_i} \\ &\quad \cdot \hat{e}(\prod_{i \in I} H_3(W \| fid \| i)^{s_i} \cdot \prod_{j=1}^{c} u_j^{\pi_j}, g^{t_f}) \end{aligned}$$

Hence, Equality (6) holds. □

The following theorems guarantee the security of our IBDO scheme as defined.

Theorem 2: Suppose the CDH assumption holds in bilinear groups and the signature scheme $\mathcal{S}$ used for generating file tags is secure. The proposed IBDO scheme is secure against adaptive impersonation and misusing delegation attacks. Specifically, neither any client nor the cloud storage server can forge a valid delegation on some warrant with regard to any file-owner, any proxy and any file of his choice, and in this way outputs a valid processed file under this delegation.

The following proof follows the standard framework established in [32], [34].

Proof: Assume the adversary $\mathcal{A}$ can output a processed file that contains a forged delegation with probability ϵ, in relation to some warrant he generates. Then, we show that adversary $\mathcal{A}$ can break Paterson and Schuldt's identity-based signature scheme [32] with probability at least ϵ. In the following, algorithm $\mathcal{C}$ will perfectly simulate a challenger when interacting with adversary $\mathcal{A}$.

The challenger $\mathcal{C}$ is given a group $G = \langle g \rangle$ with prime order q and a CDH instance (g, g^a, h), whose target is to compute h^a by interacting with $\mathcal{A}$.

Setup. The challenger $\mathcal{C}$ sets $l_u = 2(q_r + q_d)$ and $l_w = 2q_d$, where q_r and q_d represent the total number of register queries and delegation queries, respectively. It is assumed that $l_u(l+1) < q$ and $l_w(\ell+1) < q$. The challenger $\mathcal{C}$ then picks a series of random values as follows

$$\begin{aligned} k_u &\xleftarrow{R} [0, l], \quad \text{and } k_w \xleftarrow{R} [0, \ell] \\ x' &\xleftarrow{R} Z_{l_u}^*, \quad \text{and } x_i \xleftarrow{R} Z_{l_u}^* \text{ for every } 1 \le i \le l \\ y' &\xleftarrow{R} Z_q^*, \quad \text{and } y_i \xleftarrow{R} Z_q^* \text{ for every } 1 \le i \le l \\ z' &\xleftarrow{R} Z_{l_w}^*, \quad \text{and } z_j \xleftarrow{R} Z_{l_w}^* \text{ for every } 1 \le j \le \ell \\ \varpi' &\xleftarrow{R} Z_q^*, \quad \text{and } \varpi_j \xleftarrow{R} Z_q^* \text{ for every } 1 \le j \le \ell \\ \theta_j, \vartheta_j &\xleftarrow{R} Z_q^* \text{ for every } 1 \le j \le c \end{aligned}$$

Then, the challenger $\mathcal{C}$ sets the public parameters as follows

$$\begin{aligned} &g_1 = g^a, \quad g_2 = h \\ &\mu_0 = g_2^{-l_u k_u + x'} g^{y'}, \quad \mu_i = g_2^{x_i} g^{y_i} \text{for every} 1 \le i \le l \\ &\nu_0 = g_2^{-l_w k_w + z'} g^{\varpi'}, \quad \nu_j = g_2^{z_j} g^{\varpi_j} \text{for every} 1 \le j \le \ell \\ &u_j = g_2^{\theta_j} g^{\vartheta_j}, \quad 1 \le j \le c \end{aligned}$$

These parameters perfectly simulate the adversary $\mathcal{A}$'s view, that is, they are computationally indistinguishable from the public parameters in Para. Note that the solution for the given CDH instance is the same as the master secret key msk.

Queries. The adversary $\mathcal{A}$ can adaptively issue the following queries to the challenger $\mathcal{C}$.

Register Queries: The adversary $\mathcal{A}$ adaptively submits identities to $\mathcal{C}$ for requesting private keys. Since the challenger $\mathcal{C}$ does not have the master secret key msk, he will answer this type of queries as follows. For each queried identity ID_i, the challenger $\mathcal{C}$ computes $\vec{u}_i$ as defined in Equality (1). For simplicity of presentation, we set

$$F(\vec{u}_i) = x' + \sum_{j=1}^{l} x_j \mathfrak{u}_{i,j} - l_u k_u, \text{ and } J(\vec{u}_i) = y' + \sum_{j=1}^{l} y_j \mathfrak{u}_{i,j}.$$

Thus, we have $\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{i,j}} = g_2^{F(\vec{u}_i)} g^{J(\vec{u}_i)}$.

If $F(\vec{u}_i) \neq 0 \mod q$, then the challenger $\mathcal{C}$ chooses a random value $t_i \in_R Z_q^*$ and computes a private key $sk_i = (sk_{i,1}, sk_{i,2})$ such that

$$sk_{i,1} = g_1^{-\frac{J(\vec{u}_i)}{F(\vec{u}_i)}} \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{i,j}})^{t_i}, \text{ and } sk_{i,2} = g_1^{-\frac{1}{F(\vec{u}_i)}} g^{t_i}.$$

If denotes $t_i' = t_i - a/F(\vec{u}_i)$, then one can check that the above private key sk_i is valid for ID_i due to the following equalities

$$\begin{aligned} sk_{i,1} &= g_2^a \cdot (g_2^{F(\vec{u}_i)} \cdot g^{J(\vec{u}_i)})^{-\frac{a}{F(\vec{u}_i)}} \cdot (g_2^{F(\vec{u}_i)} \cdot g^{J(\vec{u}_i)})^{t_i} \\ &= g_2^a \cdot (g_2^{F(\vec{u}_i)} \cdot g^{J(\vec{u}_i)})^{t_i - \frac{a}{F(\vec{u}_i)}} = g_2^a \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{i,j}})^{t_i'} \end{aligned}$$

and $sk_{i,2} = g^{t_i - a/F(\vec{u}_i)} = g^{t_i'}$. Thus, the private key sk_i generated in this way is computationally indistinguishable from the real private key in the adversary's perspective.

Otherwise, i.e., $F(\vec{u}_i) = 0 \mod q$, the algorithm aborts.

Delegation Queries: The adversary $\mathcal{A}$ can adaptively submit a warrant W to the challenger $\mathcal{C}$, where W specifies a delegator ID_o, a proxy ID_p and some other specific information. The challenger $\mathcal{C}$ will generate a delegation of ID_o for ID_p with the warrant W as follows. The challenger first computes $\vec{u}_o$ as shown in Equality (1) and evaluates $F(\vec{u}_o)$, then gets into one of the following two cases.

Case 1: $F(\vec{u}_o) \neq 0 \mod l_u$. In this case, the challenger $\mathcal{C}$ generates a private key sk_o for ID_o as discussed in register queries, and generates a delegation σ_w as described in our scheme. The challenger $\mathcal{C}$ gives σ_w to $\mathcal{A}$.

Case 2: $F(\vec{u}_o) = 0 \mod l_u$. For a given W, the challenger $\mathcal{C}$ computes $\vec{w}$ as shown in Equality (4). Similar to the discussions of register queries, for simplicity, we set

$$K(\vec{w}) = z' + \sum_{j=1}^{\ell} z_j \mathrm{w}_j - l_w k_w, \text{ and } L(\vec{w}) = \varpi' + \sum_{j=1}^{\ell} \varpi_j \mathrm{w}_j.$$

Thus, we have $\nu_0 \prod_{j=1}^{\ell} \nu_j^{\mathrm{w}_j} = g_2^{K(\vec{w})} g^{L(\vec{w})}$.

If $K(\vec{w}) = 0 \mod q$, the algorithm aborts; otherwise, the challenger $\mathcal{C}$ chooses random values $t_o, t_w \in_R Z_q^*$ and computes a delegation $\sigma_w = (\sigma_{w,1}, \sigma_{w,2}, \sigma_{w,3})$ where

$$\sigma_{w,1} = (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{o,j}})^{t_o} \cdot g_1^{-\frac{L(\vec{w})}{K(\vec{w})}} \cdot (\nu_0 \prod_{j=1}^{\ell} \nu_j^{\mathrm{w}_j})^{t_w},$$

$$\sigma_{w,2} = g^{t_o}, \quad \text{and } \sigma_{w,3} = g_1^{-\frac{1}{K(\vec{w})}} \cdot g^{t_w}.$$

If denote $t_w' = t_w - a/K(\vec{w})$, then the delegation σ_w generated in this way is computationally indistinguishable from the real delegation in the adversary's view due to the following equality

$$\sigma_w = (g_2^a \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{o,j}})^{t_o} \cdot (\nu_0 \prod_{j=1}^{\ell} \nu_j^{\mathrm{w}_j})^{t_w'}, g^{t_o}, g^{t_w'})$$

Delegated Data Processing Queries: The adversary $\mathcal{A}$ can adaptively submit a tuple (W, M) to the challenger $\mathcal{C}$, where W specifies a file owner ID_o, a proxy ID_p and some other specific information. From the proposed scheme, we know that the register query and delegation query should be submitted at first, with inputs (ID_o, W) and ID_p, respectively. If $F(\vec{u}_o) = 0 \mod l_u$ and $K(\vec{w}) = 0 \mod q$, where $\vec{u}_o \leftarrow H_1(ID_o)$ and $\vec{w} \leftarrow H_2(W)$, then $\mathcal{C}$ just aborts the game since he cannot generate a delegation for the given identities and warrant. For the case that $F(\vec{u}_o) \neq 0 \mod l_u$ or $K(\vec{w}) \neq 0 \mod q$, the challenger $\mathcal{C}$ carries out the following steps.

Step 1: The challenger generates a delegation σ_w as discussed in delegation queries.

Step 2: Computes $\vec{u}_p \leftarrow H_1(ID_p)$ as shown in Equality (1). If $F(\vec{u}_p) = 0 \mod l_u$, then the challenger aborts. Otherwise, the challenger extracts a private key sk_p for ID_p as we discuss above. The challenger $\mathcal{C}$ picks a random value $t_f \in_R Z_q^*$ and a random unique identifier fid for file M, and computes $v_f = g^{t_f}$. For each file block $\vec{m}_i = (m_{i,1}, \cdots, m_{i,c})$ where $1 \leq i \leq r$, picks random values $\hat{t}_i \in_R Z_q^*$ and sets

$$H_3(W \| fid \| i) = g^{\hat{t}_i} \Big/ (g_2^{\sum_{j=1}^{c} \theta_j m_{i,j}} \cdot g^{\sum_{j=1}^{c} \vartheta_j m_{i,j}})$$

That is,

$$(H_3(W \| fid \| i) \cdot \prod_{j=1}^{c} u_j^{m_{i,j}})^{t_f} = (g^{\hat{t}_i})^{t_f} = (g^{t_f})^{\hat{t}_i}$$

Next, the challenger computes the metadata for file block $\vec{m}_i$ as $\sigma_i = g_2^a \cdot (\mu_0 \prod_{j=1}^{l} \mu_j^{\mathfrak{u}_{p,j}})^{t_p} \cdot (g^{t_f})^{\hat{t}_i}$. It is easy to see that the metadata generated here is computationally indistinguishable from the real one in the adversary's view.

Step 3: The challenger sends the processed file M^* which includes $\{\sigma_i\}_{1 \leq i \leq r} \cup \{\sigma_w, fid\}$ to $\mathcal{A}$.

End-Game. Finally, if the challenger does not abort, then the adversary $\mathcal{A}$ will output a forgery

$$\widetilde{M}^* = (\tilde{m}_{i,j})_{\tilde{r} \times \tilde{c}} \cup \{\tilde{\sigma}_i\}_{1 \leq i \leq \tilde{r}} \cup \{\tilde{\sigma}_w, \widetilde{fid}\}$$

in relation to some warrant $\widetilde{W}$ with non-negligible probability. According to Definition 1, the adversary $\mathcal{A}$ succeeds in the game means that $\tilde{\sigma}_w$ is valid for $\widetilde{W}$. It further implies $\tilde{\sigma}_w$ is a forged signature for $\widetilde{W}$ in identity-based setting. Thus, if the adversary $\mathcal{A}$ can break the proposed scheme with probability ϵ, then it can also break Paterson and Schuldt's identity-based

signature scheme [32] (i.e., outputs h^a) with probability at least ϵ. □

Theorem 3: Suppose the CDH assumption holds in bilinear groups and the signature scheme $\mathcal{S}$ used for generating file tags is secure. The proposed IBDO scheme is secure against modification attacks. Specifically, the storage server cannot forge a valid response for any integrity auditing challenge, if the challenged file blocks in the outsourced file have been corrupted.

The following proof is inspired by a standard framework established in [9].

Proof: The proof consists of a list of security games along with respective analysis.

Game 0. In this game, both the adversary and the challenger behave in the way defined in Definition 2.

Game 1. This game is identical to Game 0, with the following difference. The challenger maintains all the processed files that have been provided to the adversary in Setup phase. In the final round of integrity auditing protocol, the adversary outputs a proof that satisfies verification equation (6) for the challenge $\widetilde{C}$, warrant $\widetilde{W}$ and processed file $\widetilde{M}^*$, while the aggregate metadata is not equal to that generated by the challenger from its maintained information.

Analysis. Suppose the adversary $\mathcal{A}$ succeeds in Game 2 with non-negligible probability. Then, we can construct an algorithm $\mathcal{C}$ to solve the CDH problem. Algorithm $\mathcal{C}$ is given a group $G = \langle g \rangle$ with prime order q and a CDH instance (g, g^α, h), whose goal is to output h^α by interacting with $\mathcal{A}$.

Algorithm $\mathcal{C}$ behaves like a challenger of Game 0, but with the following differences:

- It picks a random value $a \in_R \mathbb{Z}_q^*$, and sets $g_1 = g^a$, $g_2 = h$ and $msk = g_2^a$. Also, it randomly chooses $\theta_j, \vartheta_j \xleftarrow{R} Z_q^*$ for each $1 \le j \le c$, and computes $u_j = g_2^{\theta_j} g^{\vartheta_j}$.
- It controls the random oracle H_3 so that it maintains all the queries and answers them in a consistently manner. For the queries of the form $H_3(W \| fid \| i)$, it will respond in the following way as shown in processing file.
- For processing a file M under W, it first computes $\vec{u}_p \leftarrow H_1(ID_p)$ as defined in Equality (1), and extracts a private key sk_p for ID_p as defined in the proposed scheme (see Equality (2)). The challenger $\mathcal{C}$ picks a random unique identifier fid and a random value $\bar{a} \in_R Z_q^*$ for file M, and computes $v_f = (g^\alpha)^{\bar{a}}$ which implies $t_f = \alpha \bar{a}$. For each file block $\vec{m}_i = (m_{i,1}, \cdots, m_{i,c})$ where $1 \le i \le r$, picks a random value $\hat{t}_i \in_R Z_q^*$ and sets

$$H_3(W \| fid \| i) = g^{\hat{t}_i} \Big/ (g_2^{\sum_{j=1}^c \theta_j m_{i,j}} \cdot g^{\sum_{j=1}^c \vartheta_j m_{i,j}})$$

  That is,

$$(H_3(W \| fid \| i) \cdot \prod_{j=1}^c u_j^{m_{i,j}})^{t_f} = (g^{\hat{t}_i})^{t_f} = (v_f)^{\hat{t}_i}$$

  The challenger proceeds to compute the metadata for file block $\vec{m}_i$ as $\sigma_i = g_2^a \cdot (\mu_0 \prod_{j=1}^l \mu_j^{u_{p,j}})^{t_p} \cdot (v_f)^{\hat{t}_i}$. It is easy to see that the metadata generated in this way is computationally indistinguishable from the real metadata in the adversary's view.
- Algorithm $\mathcal{C}$ interacts with the adversary $\mathcal{A}$ to perform the integrity auditing protocol. As specified in Game 1, the game aborts if the adversary outputs an aggregate metadata $\tilde{\sigma}$ that is not equal to the expected one σ in a round of auditing.

Let $(\tilde{\pi}_1, \cdots, \tilde{\pi}_c)$ be the aggregate challenged file sectors in the proof $\widetilde{P}$. Thus, we have

$$\hat{e}(\tilde{\sigma}, g) = \hat{e}(g_1, g_2)^{\sum_{i \in I} s_i} \cdot \hat{e}(\mu_0 \prod_{j=1}^l \mu_j^{u_{p,j}}, g^{t_p})^{\sum_{i \in I} s_i} \cdot \hat{e}(\prod_{i \in I} H_3(\widetilde{W} \| \widetilde{fid} \| i)^{s_i} \cdot \prod_{j=1}^c u_j^{\tilde{\pi}_j}, v_f) \quad (7)$$

and

$$\hat{e}(\sigma, g) = \hat{e}(g_1, g_2)^{\sum_{i \in I} s_i} \cdot \hat{e}(\mu_0 \prod_{j=1}^l \mu_j^{u_{p,j}}, g^{t_p})^{\sum_{i \in I} s_i} \cdot \hat{e}(\prod_{i \in I} H_3(\widetilde{W} \| \widetilde{fid} \| i)^{s_i} \cdot \prod_{j=1}^c u_j^{\pi_j}, v_f) \quad (8)$$

Note that $\tilde{\pi}_j = \pi_j$ cannot hold for all $1 \le j \le c$ since otherwise, $\tilde{\sigma} = \sigma$. If denote $\Delta\pi_j = \tilde{\pi}_j - \pi_j$, then at least one element of $\{\Delta\pi_j : 1 \le j \le c\}$ should be nonzero. Dividing Equality (7) by Equality (8) on respective sides and assuming $v_f = (g^\alpha)^{\bar{a}_f}$, we get

$$\hat{e}(\tilde{\sigma}/\sigma, g) = \hat{e}(\prod_{j=1}^c u_j^{\Delta\pi_j}, v_f) = \hat{e}(\prod_{j=1}^c (g_2^{\theta_j} g^{\vartheta_j})^{\Delta\pi_j}, (g^\alpha)^{\bar{a}_f})$$

It further implies

$$\hat{e}(\tilde{\sigma}\sigma^{-1}(g^\alpha)^{-\bar{a}_f \sum_{j=1}^c \vartheta_j \Delta\pi_j}, g) = \hat{e}(h, g^\alpha)^{\bar{a}_f \sum_{j=1}^c \theta_j \Delta\pi_j}$$

Thus, we get the solution of the given CDH instance as follows

$$h^\alpha = (\tilde{\sigma}\sigma^{-1}(g^\alpha)^{-\bar{a}_f \sum_{j=1}^c \vartheta_j \Delta\pi_j})^{\frac{1}{\bar{a}_f \sum_{j=1}^c \theta_j \Delta\pi_j}}$$

as long as $\bar{a}_f \sum_{j=1}^c \theta_j \Delta\pi_j \neq 0 \bmod q$. Since some of $\{\Delta\pi_j : 1 \le j \le c\}$ are nonzero and $\bar{a}_f, \theta_j$ for $1 \le j \le c$ are random values, we know $\Pr[\bar{a}_f \sum_{j=1}^c \theta_j \Delta\pi_j = 0 \bmod q] = 1/q$. Thus, if the difference between the adversary's probabilities of success in Game 0 and Game 1 is non-negligible, then we can construct an algorithm $\mathcal{C}$ as discussed above to solve the CDH problem.

Game 2. This game is identical to Game 1, with the following difference. As in Game 1, the challenger maintains all the processed files that have been provided to the adversary in Setup phase. In the final round of integrity auditing protocol, the adversary outputs a proof that satisfies verification equation (6) for the challenge $\widetilde{C}$, warrant $\widetilde{W}$ and processed file $\widetilde{M}^*$, while at least one of the aggregate challenged file sectors is not equal to that generated by the challenger according to its maintained information.

Analysis. Suppose the adversary $\mathcal{A}$ succeeds in Game 2 with non-negligible probability. Then, we can construct an algorithm $\mathcal{C}$ to solve the discrete logarithm (DL) problem. Algorithm $\mathcal{C}$ is given a group $G = \langle g \rangle$ with prime order q

TABLE II

COMPARISON ON COMPUTATION COSTS OF EACH ALGORITHM IN IBDO SCHEME AND SW SCHEME

Algorithm	Costs (at server side)	Costs (at client side)
Our IBDO scheme		
Setup	$2\mathbb{E}$	—
Regst	$(l+1)\mathbb{M}+2\mathbb{E}+1\mathbb{H}$	$l\mathbb{M}+2\mathbb{P}+1\mathbb{M}_T+1\mathbb{H}$
Dlgtn (generation)	—	$(\ell+1)\mathbb{M}+2\mathbb{E}+1\mathbb{H}$
Dlgtn (verification)	—	$(l+\ell)\mathbb{M}+3\mathbb{P}+2\mathbb{M}_T+2\mathbb{H}$
IBDOsc	—	$(rc+r+1)\mathbb{E}+(rc+r)\mathbb{M}+r\mathbb{H}$
Audit	$c\|I\|\mathbb{M}_q+c(\|I\|-1)\mathbb{A}_q+(\|I\|-1)\mathbb{M}+\|I\|\mathbb{E}$	$(\|I\|-1)\mathbb{A}_q+6\mathbb{P}+(2l+\ell+c+\|I\|-1)\mathbb{M}$ $+(\|I\|+c)\mathbb{E}+2\mathbb{E}_T+4\mathbb{M}_T+(\|I\|+3)\mathbb{H}$
SW scheme in bilinear groups		
Processing file	—	$(rc+r)\mathbb{E}+rc\mathbb{M}+r\mathbb{H}$
Audit	$c\|I\|\mathbb{M}_q+c(\|I\|-1)\mathbb{A}_q+(\|I\|-1)\mathbb{M}+\|I\|\mathbb{E}$	$2\mathbb{P}+(c+\|I\|-1)\mathbb{M}+(\|I\|+c)\mathbb{E}+\|I\|\mathbb{H}$

and a DL instance (g,h), whose goal is to output α so that $h=g^{\alpha}$ by interacting with $\mathcal{A}$.

Algorithm $\mathcal{C}$ behaves like a challenger of Game 1, but with the following differences:

- For processing a file M under W, it first computes $\vec{u}_p \leftarrow H_1(ID_p)$ and extracts a private key sk_p for ID_p as in the proposed scheme. Also, it processing file M by following our scheme but using parameters $u_j = g_2^{\theta_j} g^{\vartheta_j}$ for each $1 \le j \le c$, where $\theta_j, \vartheta_j \overset{R}{\leftarrow} Z_q^*$. It is easy to see that the metadata generated in this way is computationally indistinguishable from the real metadata in the adversary's view.
- Algorithm $\mathcal{C}$ interacts with $\mathcal{A}$ to perform the integrity auditing protocol. As specified in Game 2, the game aborts if the outputted aggregate file sectors $\{\tilde{\pi}_j : 1 \le j \le c\}$ are different from the expected ones $\{\pi_j : 1 \le j \le c\}$ in a round of auditing.

According to Game 1, we know that $\tilde{\sigma}=\sigma$. By checking Equality (7) and Equality (8), it follows that

$$\hat{e}(\prod_{i\in I} H_3(\widetilde{W}\|\widetilde{fid}\|i)^{s_i}\cdot\prod_{j=1}^{c} u_j^{\tilde{\pi}_j}, v_f) = \hat{e}(\prod_{i\in I} H_3(\widetilde{W}\|\widetilde{fid}\|i)^{s_i}\cdot\prod_{j=1}^{c} u_j^{\pi_j}, v_f)$$

It further implies that $\prod_{j=1}^{c} u_j^{\tilde{\pi}_j} = \prod_{j=1}^{c} u_j^{\pi_j}$. Let $\Delta\pi_j = \tilde{\pi}_j - \pi_j$ for each $1 \le j \le c$. Thus, at least one element of $\{\Delta\pi_j : 1 \le j \le c\}$ should be nonzero. We get

$$\prod_{j=1}^{c} u_j^{\Delta\pi_j} = \prod_{j=1}^{c}\left(g_2^{\theta_j} g^{\vartheta_j}\right)^{\Delta\pi_j} = h^{\sum_{j=1}^{c}\theta_j\Delta\pi_j} g^{\sum_{j=1}^{c}\vartheta_j\Delta\pi_j} = 1$$

Thus, we get the solution of the given DL instance as follows

$$\alpha = -(\sum_{j=1}^{c}\vartheta_j\Delta\pi_j)(\sum_{j=1}^{c}\theta_j\Delta\pi_j)^{-1} \mod q$$

as long as $\sum_{j=1}^{c}\theta_j\Delta\pi_j \neq 0 \bmod q$. Since some of $\{\Delta\pi_j : 1 \le j \le c\}$ are nonzero and θ_j for $1 \le j \le c$ are random values, we know $\Pr[\sum_{j=1}^{c}\theta_j\Delta\pi_j = 0 \bmod q] = 1/q$. Thus, if the difference between the adversary's probabilities of success in Game 1 and Game 2 is non-negligible, then we can construct an algorithm $\mathcal{C}$ as discussed above to solve the DL problem.

Since the hardness of CDH problem implies that of DL problem, we complete the proof by combing Game 0, Game 1 and Game 2. □

VI. EVALUATION

A. Modification Checkability Analysis

The detection probability of the storage server's misbehavior in PoS related schemes has been discussed in [4] and [10] etc. In our scheme, when auditing an outsourced file, both the delegation σ_w and the aggregate metadata σ should be validated by the auditor. If the delegation has been tampered with, then it can always be detected by the auditor in an execution of auditing protocol according to Equality (5) and Theorem 2. Furthermore, similarly to the results in [4], if t blocks have been tampered with or deleted by the storage server, then the auditor can detect this misbehavior with probability $p = 1-\prod_{i=0}^{|I|-1}\frac{r-t-i}{r-i}$. Particularly, if t percent of file blocks have been tampered with, the detection probability p on the corrupted file is only determined by the number of challenged blocks $|I|$, that is, $p \approx 1-(1-t)^{|I|}$. In this case, the auditor can challenge $|I| \approx \ln(1-p)/\ln(1-t)$ blocks in a round of auditing to achieve detection probability p. For example, if the delegation is intact while one percent of blocks are corrupted, then it can be detected with probability of 90% and 99% by random choosing 230 and 460 blocks for auditing in a challenge, respectively.

B. Theoretical Analysis

We summarize the computation costs of each algorithm and protocol in Table II, which shows a comparison between our scheme and Shacham and Waters' publicly verifiable PoR scheme over bilinear groups [9]. The costs of file tag generation and verification are not counted as they are determined by the specific signature scheme $\mathcal{S}$ chosen at the processing file phase. In the table, $\mathbb{M}$ and $\mathbb{E}$ denote one multiplication and one exponentiation in G, respectively; similarly, $\mathbb{M}_T$ and $\mathbb{E}_T$ respective denote one multiplication and one exponentiation in G_T; $\mathbb{M}_q$ and $\mathbb{A}_q$ represent one multiplication and one addition in Z_q, respectively; $\mathbb{P}$ denotes one bilinear pairing evaluation $\hat{e}: G\times G\to G_T$. We do not differentiate hash evaluations of H_1, H_2 or H_3, and denote them commonly as $\mathbb{H}$. Since both g_1 and g_2 are public parameters in our IBDO scheme, $\hat{e}(g_1,g_2)$ can be pre-computed and looks also as a public value. Thus, it

TABLE III
COMPARISON WITH SW SCHEME IN BILINEAR GROUPS

Schemes	Storage costs at cloud side	Communication costs in auditing	Setting	Delegation enabled	Multi-user
SW scheme	$\|M\| + rES_G$	$1ES_G + (2\|I\| + c)ES_q$	Public key	×	×
Ours	$\|M\| + (r+1)ES_G$	$2ES_G + (2\|I\| + c)ES_q$	Identity based	√	√

is not included in Table II. As known that exponentiations and pairings are more time-consuming compared to the other ones, they would essentially determine the efficiency of these two schemes.

In both schemes, all verification cases at user side take only constant pairings, that is, when verifying a private key issued by registry server, validating a delegation generated by a file owner, or checking a proof in a round of (comprehensive) auditing. All other phases do not involve pairing evaluations. For processing a file M with sectors $\{m_{i,j}\}_{r\times c}$, SW scheme takes $(rc + r)$ exponentiations, rc multiplications and r hash evaluations. Our scheme only incurs one more time-consuming exponentiation, plus r efficient multiplications. Thus, both SW scheme and ours enjoy the same efficiency level for processing the same file. When performing a round of auditing protocol on an outsourced file, both schemes require the same number (i.e., $|I|$) of exponentiations in group G at cloud storage server side. While at auditor side, SW scheme takes $(|I| + c)$ exponentiations in group G and two pairings to check a proof, and our scheme would take another four pairings in order to auditing the corresponding delegation.

Table III further compares our IBDO scheme with SW scheme in terms of storage costs at the cloud side, communication overheads in a round of auditing as well as designing settings. In the table, ES_G and ES_q respectively denote the byte size of group elements in G and Z_p. The block numbers $i \in I$ in a challenge are taken as elements in Z_q. For realizing origin auditing on outsourced file, our scheme should let cloud storage server to keep one more element in G, that is, the first element in the delegation for this file, when compared to SW scheme. Also, this additional element would be sent to the auditor when performing comprehensive auditing in our scheme.

VII. EXPERIMENTAL ANALYSIS

We conducted experiments on our IBDO scheme and the SW schem using Pairing Based Cryptography (PBC) library (http://crypto.stanford.edu/pbc/). All algorithms and protocol were coded using C programming language and conducted on a system with Intel(R) Core(TM) i5-5200U CPU at 2.20GHz and 2.20 GHz and 4.00GB RAM in Windows 8. The elliptic curve is of type $y^2 = x^3 + x$ with $|q| = 160$ bits and $ES_G = 256$ bits. We set $l = 160$ and $\ell = 160$. The file sectors are of 160 bits size.

The performance of generating and verifying a private key for some user in Regst, and that of generating and verifying a delegation in Dlgtn are shown in Figure 2. The time consumed by each of these steps is roughly 10ms, which is negligible for deploying in real-world applications. As discussed in Section VI-B, processing a file using both our scheme and the SW scheme would run a number of exponentiations in group

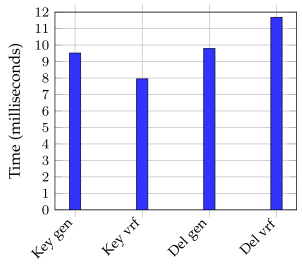


Fig. 2. Performance of Regst and Dlgtn

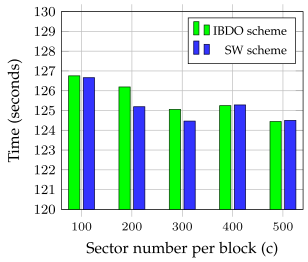


Fig. 3. Performance of processing a 1MB file with different sector numbers

G. In detail, when processing a file of S bytes by splitting into sectors of size s_{sec}, it will create $r = \lceil S/(c \cdot s_{sec}) \rceil$ file blocks. Thus, the number of required exponentiations under both schemes for generating metadata would respectively be

$$N_{IBDO} = \lceil \frac{S}{c \cdot s_{sec}} \rceil (c+1) + 1, \quad \text{and}$$
$$N_{SW} = \lceil \frac{S}{c \cdot s_{sec}} \rceil (c+1).$$

We compare the efficiency of both schemes by letting them processing a 1MB file, and consider several cases with different splitting manners, that is, we set $c = 100, \cdots, 500$, respectively. In our experiment setting, $s_{sec} = 20$ bytes. The experimental results shown in Figure 3 indicate that both schemes enjoy the same efficiency level in all processing cases. This is consistent with above theoretical analysis.

Figure 4 shows the time to audit an outsourced file with 1% corruption. We do not take into account the time costs of preparing a challenge C since it can be run offline for sampling a series of random elements. In the experiments, each file block consists of 100 sectors, which means that it has around 4KB of size. We consider several cases of achieving different detection probability of corruption, i.e., $0.5, \cdots, 0.99$. The simulation results of Figure 4 demonstrate that our IBDO scheme has comparable efficiency as SW scheme at both sides of the auditor and cloud storage server in carrying out the auditing protocol. For example, for realizing 0.9 detection probability, the auditor in both schemes can finish in less than

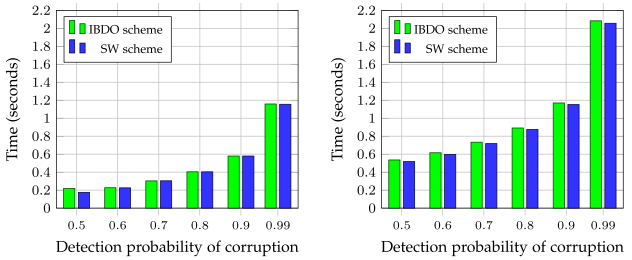


Fig. 4. Performance in a round of (comprehensive) auditing protocol with different detection probability on a 1% corrupted file. (a) Cloud side. (b) Auditor side.

1.2 seconds. Also, in both schemes, the time cost at the auditor side is larger than that at the cloud side, which is consistent with the theoretical analysis shown in Table II. Note that the former can be shared by several auditors in a multi-auditor setting. That is, several auditors can cooperatively audit the same file to achieve a high detection probability of corruption, where each auditor needs only to audit a different part of the outsourced file with a low detection probability. Lower detection probability requires smaller amount of computations so that it would be more efficient for each involved auditor.

VIII. Conclusion

In this paper, we investigated proofs of storage in cloud in a multi-user setting. We introduced the notion of identity-based data outsourcing and proposed a secure IBDO scheme. It allows the file-owner to delegate her outsourcing capability to proxies. Only the authorized proxy can process and outsource the file on behalf of the file-owner. Both the file origin and file integrity can be verified by a public auditor. The identity-based feature and the comprehensive auditing feature make our scheme advantageous over existing PDP/PoR schemes. Security analyses and experimental results show that the proposed scheme is secure and has comparable performance as the SW scheme.

References

[1] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud data protection for the masses," *Computer*, vol. 45, no. 1, pp. 39–45, Jan. 2012.

[2] C.-K. Chu, W.-T. Zhu, J. Han, J. K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 50–57, Oct. 2013.

[3] K. Yang and X. Jia, "Data storage auditing service in cloud computing: Challenges, methods and opportunities," *World Wide Web*, vol. 15, no. 4, pp. 409–428, 2012.

[4] G. Ateniese *et al.*, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2007, pp. 598–609.

[5] J. Sun and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 6, pp. 754–764, Jun. 2010.

[6] J. Sun, X. Zhu, C. Zhang, and Y. Fang, "HCPP: Cryptography based secure EHR system for patient privacy and emergency healthcare," in *Proc. IEEE 31st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2011, pp. 373–382.

[7] L. Guo, C. Zhang, J. Sun, and Y. Fang, "PAAS: A privacy-preserving attribute-based authentication system for eHealth networks," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2012, pp. 224–233.

[8] A. Juels and B. S. Kaliski, Jr., "PORs: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2007, pp. 584–597.

[9] H. Shacham and B. Waters, "Compact proofs of retrievability," *J. Cryptol.*, vol. 26, no. 3, pp. 442–483, 2013.

[10] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 275–362, Feb. 2013.

[11] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," *IEEE Trans. Serv. Comput.*, vol. 8, no. 1, pp. 92–106, Feb. 2015.

[12] F. Chen, T. Xiang, Y. Yang, and S. S. M. Chow, "Secure cloud storage meets with secure network coding," *IEEE Trans. Comput.*, vol. 65, no. 6, pp. 1936–1948, Jun. 2016.

[13] H. Wang, "Proxy provable data possession in public clouds," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 551–559, Oct./Dec. 2013.

[14] S.-T. Shen and W.-G. Tzeng, "Delegable provable data possession for remote data in the clouds," in *Information and Communications Security* (Lecture Notes in Computer Science), vol. 7043, S. Qing, W. Susilo, G. Wang, and D. Liu, Eds. Berlin, Germany: Springer, 2011, pp. 93–111.

[15] F. Armknecht, J.-M. Bohli, G. O. Karame, Z. Liu, and C. A. Reuter, "Outsourced proofs of retrievability," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Nov. 2014, pp. 831–843.

[16] H. Wang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," *Inf. Security, IET*, vol. 8, no. 2, pp. 114–121, Mar. 2014.

[17] G. Ateniese, R. di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw.*, New York, NY, USA, 2008, Art. no. 9.

[18] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

[19] B. Wang, S. S. M. Chow, M. Li, and H. Li, "Storing shared data on the cloud via security-mediator," in *Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2013, pp. 124–133.

[20] Y. Wang *et al.*, "Securely outsourcing exponentiations with single untrusted program for cloud storage," in *Computer Security—ESORICS*, vol. 8712. Switzerland: Springer, 2014, pp. 326–343.

[21] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2231–2244, Dec. 2012.

[22] H. Wang, "Identity-based distributed provable data possession in multicloud storage," *IEEE Trans. Services Computing*, vol. 8, no. 2, pp. 328–340, Mar. 2015.

[23] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1167–1179, Jun. 2015.

[24] Y. Wang, Q. Wu, B. Qin, X. Chen, X. Chen, X. Huang, and J. Lou, "Ownership-hidden group-oriented proofs of storage from pre-homomorphic signatures," *Peer-to-Peer Netw. Appl.*, pp. 1–17, 2016. [Online]. Available: http://link.springer.com/article/10.1007/s12083-016-0530-8?no-access=true

[25] T. Jiang, X. Chen, and J. Ma, "Public integrity auditing for shared dynamic cloud data with group user revocation," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2363–2373, Aug. 2016.

[26] X. Fan, G. Yang, Y. Mu, and Y. Yu, "On indistinguishability in remote data integrity checking," *Comput. J.*, vol. 58, no. 4, pp. 823–830, 2015.

[27] Y. Yu *et al.*, "Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage," *Int. J. Inf. Secur.*, vol. 14, no. 4, pp. 307–318, Aug. 2015.

[28] Y. Yu *et al.*, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Trans. Inf. Forensics Security*, to be published.

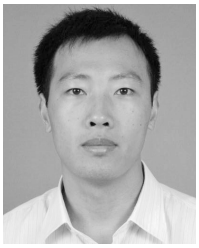
[29] J. Zhang and Q. Dong, "Efficient ID-based public auditing for the outsourced data in cloud storage," *Inf. Sci.*, vols. 343–344, pp. 1–14, May 2016.

[30] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang, "SCLPV: Secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors," *IEEE Trans. Comput. Social Syst.*, vol. 2, no. 4, pp. 159–170, Dec. 2015.

[31] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2009, pp. 187–198.

[32] K. G. Paterson and J. C. N. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Information Security and Privacy* (Lecture Notes in Computer Science), vol. 4058, L. Batten and R. Safavi-Naini, Eds. Heidelberg, Germany: Springer, 2006, pp. 207–222.

[33] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. ASIACRYPT*, vol. 2248. 2001, pp. 514–532.
[34] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology–EUROCRYPT* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Heidelberg, Germany: Springer, 2005, pp. 114–127.



Yujue Wang received the Ph.D. degrees from Wuhan University, Wuhan, China, and City University of Hong Kong, Hong Kong, in 2015, under the joint Ph.D. program. He is currently a Research Fellow with the School of Information Systems, Singapore Management University. His research interests include applied cryptography, database security, and cloud computing security.

Qianhong Wu (M'15) received the Ph.D. degree in cryptography from Xidian University in 2004. Since 2004, he has been with Wollongong University, Australia, as an Associate Research Fellow, with Wuhan University, China, as an Associate Professor, and with Universitat Rovira i Virgili, Spain, as a Research Director. He is currently a Professor in Beihang University, China. He has been a holder or co-holder of eight China/Australia/Spain funded projects. He has authored over 20 patents and over 120 publications in leading journals and conferences. His research interests include cryptography, information security and privacy, VANET security, and cloud computing security. He is a member of IACR. He has served on the program committee of several international conferences on information security and privacy.

Bo Qin received the Ph.D. degree in cryptography from Xidian University, China, in 2008. Since 2008, she has been with the Xi'an University of Technology, China, as a Lecturer and with Universitat Rovira i Virgili, Catalonia, as a Post-Doctoral Researcher. She is currently a Lecturer with Renmin University, China. She has been a holder/co-holder of five China/Spain funded projects. She has authored over 80 publications in well-recognized journals and conferences. Her research interests include pairing-based cryptography, data security and privacy, and VANET security. She has served on the program committee of a number of international conferences in information security.

Wenchang Shi received the bachelor's degree from Peking University, and the master's and Ph.D. degrees from the Chinese Academy of Science. He is currently a Full Professor with the Renmin University of China and also an Adjunct Professor with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include information system security and operation system security. He has authored over 100 publications in these areas and holds over 10 patents.

Robert H. Deng (F'16) was a Principal Scientist and the Manager of the Infocomm Security Department, Institute for Infocomm Research, Singapore. He has been a Professor with the School of Information Systems, Singapore Management University, since 2004. His research interests include data security and privacy, multimedia security, and network and system security. He received the University Outstanding Researcher Award from the National University of Singapore in 1999 and the Lee Kuan Yew Fellow for Research Excellence from the Singapore Management University in 2006. He was named Community Service Star and Showcased Senior Information Security Professional by $(ISC)^2$ under its Asia-Pacific Information Security Leadership Achievements Program in 2010. He has served on the editorial boards of many international journals, including IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the *Journal of Computer Science and Technology* (the Chinese Academy of Sciences), and the *International Journal of Information Security*. He is currently the Chair of the Steering Committee of the ACM Asia Conference on Computer and Communications Security.

Jiankun Hu received the Ph.D. degree in control engineering from the Harbin Institute of Technology, China, in 1993, and the master's degree in computer science and software engineering from Monash University, Australia, in 2000. He was a Research Fellow with the Delft University of Technology, The Netherlands, from 1997 to 1998, and Melbourne University, Australia, from 1998 to 1999. He has been with Ruhr University, Bochum, Germany. He is currently a Full Professor and the Research Director of the Cyber Security Laboratory, School of Engineering and Information Technology, University of New South Wales, Australia. His main research interest is in the field of cyber security, including biometrics security, where he has authored many papers in high-quality conferences and journals, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has received seven Australian Research Council (ARC) Grants, and serves at the prestigious Panel of Mathematics, Information and Computing Sciences and the ARC Excellence in Research for Australia Evaluation Committee. He received the prestigious German Alexander von Humboldt Fellowship from Ruhr University from 1995 to 1996. He has served on the Editorial Board of up to seven international journals, including IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and served as the Security Symposium Chair of IEEE ICC and IEEE Globecom.