**Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage**

Yong Yu, Man Ho Au, Member, IEEE, Giuseppe Ateniese, Xinyi Huang, Willy Susilo, Yuanshun Dai, and Geyong Min

**Abstract**

Remote data integrity checking (RDIC) enables a data storage server, say a cloud server, to prove to a verifier that it is actually storing a data owner's data honestly. To date, a number of RDIC protocols have been proposed in the literature, but most of the constructions suffer from the issue of a complex key management, that is, they rely on the expensive public key infrastructure (PKI), which might hinder the deployment of RDIC in practice. In this paper, we propose a new construction of identity-based (ID-based) RDIC protocol by making use of key-homomorphic cryptographic primitive to reduce the system complexity and the cost for establishing and managing the public key authentication framework in PKI-based RDIC schemes. We formalize ID-based RDIC and its security model, including security against a malicious cloud server and zero knowledge privacy against a third party verifier. The proposed ID-based RDIC protocol leaks no information of the stored data to the verifier during the RDIC process. The new construction is proven secure against the malicious server in the generic group model and achieves zero knowledge privacy against a verifier. Extensive security analysis and implementation resuits demonstrate that real-world applications.

Index Terms-Cloud storage, data integrity, privacy preserving, identity-based cryptography.

Manuscript received January 18, 2016; revised April 6, 2016; accepted April 6, 2016. Date of publication October 7, 2016; date of current version January 25,2017 . This work was supported in part by the National Natural Science Foundation of China under Grant 61501333, Grant 61300213, Grant 61272436 , and Grant 61472083 , in part by the Fok Ying Tung Education Foundation under Grant 141065, in part by the Program for New Century Excellent Talents in Fujian University under Grant JA14067, and in part by the Fundamental Research Funds for the Central Universities under manuscript and approving it for publication was Prof. Chip-Hong Chang. (Corresponding author: Man Ho Au.)

Y. Yu is with the School of Computer Science, Shaanxi Normal University, Xi'an 710062 , China, and also with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731 , China.

M. H. Au is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: [csallen@comp.polyu.edu.hk](mailto:csallen@comp.polyu.edu.hk)).

G. Ateniese is with the Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ 07030, USA.

X. Huang is with the Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350007, China.

W. Susilo is with the Center for Computer and Information Security Research, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia.

Y. Dai is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.

G. Min is with the College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, U.K.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIFS.2016.2615853

**1. INTRODUCTION**

model represents a new vision of providing computing services can enjoy low management overhead and immediate access to data wherever they have a network, rather than having to stay nearby their computers. referred the data source from international data corporation cloud users' fears. One of the major security concerns of they no longer physically possess their data and thus lose reliability, the cloud security alliance (CSA) published an because it might be too late to recover the corrupted data. if their outsourced data are stored properly.

IDC Enterprise Panel, 2010

[www.businessinsider.com/amazon-lost-data-2011-4](http://www.businessinsider.com/amazon-lost-data-2011-4). LOUD computing [1], which has received considerable attention from research communities in academia as well as industry, is a distributed computation model over a large pool of shared-virtualized computing resources, such as storage, processing power, applications and services. Cloud users are provisioned and release recourses as they want in cloud computing environment. This kind of new computation as public utilities like water and electricity. Cloud computing brings a number of benefits for cloud users. For example, (1) Users can reduce capital expenditure on hardware, software and services because they pay only for what they use; (2) Users a wide range of applications; and (3) Users can access their

However, there is a vast variety of barriers before cloud computing can be widely deployed. A recent survey by Oracle enterprise panel, showing that security represents of cloud users is the integrity of their outsourced files since the control over their data. Moreover, the cloud server is not fully trusted and it is not mandatory for the cloud server to report data loss incidents. Indeed, to ascertain cloud computing analysis of cloud vulnerability incidents. The investigation [2] revealed that the incident of data Loss & Leakage accounted for of all incidents, ranked second only to "Insecure Interfaces & APIs". Take Amazon's cloud crash disaster as an example. In 2011 , Amazon's huge EC2 cloud services crash permanently destroyed some data of cloud users. The data loss was apparently small relative to the total data stored, but anyone who runs a website can immediately understand how terrifying a prospect any data loss is. Sometimes it is insufficient to detect data corruption when accessing the data As a result, it is necessary for cloud users to frequently check

— 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See <http://www.ieee.org/publications_standards/publications/rights/index.html> for more information. The size of the cloud data is huge, downloading the entire file to check the integrity might be prohibitive in terms of bandwidth cost, and hence, very impractical. Moreover, traditional cryptographic primitives for data integrity checking such as hash functions, authorisation code (MAC) cannot apply here directly due to being short of a copy of the original file in verification. In conclusion, remote data integrity checking for secure cloud storage is a highly desirable as well as a challenging research topic.

Blum proposed an auditing issue for the first time that enables data owners to verify the integrity of remote data without explicit knowledge of the entire data [3]. Recently, remote data integrity checking becomes more and more significant due to the development of distributed storage systems and online storage systems. Provable data possession (PDP) [4], [5] at untrusted stores, introduced by Ateniese et al., is a novel technique for "blockless validating" data integrity over remote servers. In PDP, the data owner generates some metadata for a file, and then sends his data file together with the metadata to a remote server and deletes the file from its local storage. To generate a proof that the server stores the original file correctly, the server computes a response to a challenge from the verifier. The verifier can verify if the file keeps unchanged via checking the correctness of the response. PDP is a practical approach to checking the integrity of cloud data since it adopts a spot-checking technique. Specifically, a file is divided into blocks and a verifier only challenges a small set of randomly chosen clocks for integrity checking. According to the example given by Ateniese et al. [4], for a file with 10,000 blocks, if the server has deleted of the blocks, then a verifier can detect server's misbehavior with probability greater than by asking proof of possession for only 460 randomly selected blocks. Ateniese et al. proposed two concrete PDP constructions by making use of RSA-based homomorphic linear authenticators. Due to its necessity and practicability, remote data integrity checking has attracted extensive research interest [7]-[10], [12]-[22], [26] in recent years. Shacham and Waters [7] proposed the notion of compact proofs of retrievability by making use of publicly verifiable homomorphic authenticators from BLS signature [37]. This scheme also relies on homomorphic properties to aggregate a proof into a small authenticator value and as a result, the public retrievability can be achieved.

Ateniese et al. [23] considered dynamic PDP scheme for the first time based on hash functions and symmetric key encryptions, which means the data owner can dynamically update their file after they store their data on the cloud server. The dynamics operation involves data insertion, modification, deletion and appending. This scheme [23] is efficient but has only limited number of queries and block insertion cannot explicitly be supported. Erway et al. [25] extended the PDP model to dynamic PDP model by utilizing rankbased authenticated skip lists. Wang et al. [15] improved the previous PDP models by manipulating the Merkle Hash Tree (MHT) for block tag authentication. A recent work due to Liu et al. [26] showed that MHT itself is not enough to verify the block indices, which may lead to replace attack. They gave top-down levelled multi-replica MHT based data auditing scheme for dynamic big data storage on the cloud.

In data integrity checking with public verifiability, an external auditor (or anyone) is able to verify the integrity of the cloud data. In this scenario, data privacy against the third party verifier is highly essential since the cloud users may store confidential or sensitive files say business contracts or medical records to the cloud. However, this issue has not been fully investigated. The "privacy" definition in the previous privacy-preserving public auditing scheme [13] requires that the verifier cannot recover the whole blocks from the responses generated by the cloud server. However, this definition is not strong enough, for example, it is vulnerable to dictionary attack. Wang et al. [16] proposed the notion of "zero-knowledge public auditing" to resist off-line guessing attack. However, a formal security model is not provided in this work. Yu et al. [27] recently enhanced the privacy of remote data integrity checking protocols for secure cloud storage, but their model works only in public key infrastructure (PKI) based scenario instead of the identity-based framework. Encrypting the file before outsourcing can partially address the data privacy issue but leads to losing the flexibility of the protocols, since privacy preserving RDIC protocols can be used as a building block for other primitives. For example, Ateniese et al. [28] proposed a framework for building leakage-resilient identification protocols in the bounded retrieval model from publicly verifiable proofs of storage that are computationally zero-knowledge, but in the identification schemes, even the encrypted files must stay private.

Currently, a majority of the existing RDIC constructions rely on PKI where a digital certificate is used to guarantee the genuine of a user's public key. These constructions incur complex key management procedures since certificate generation, certificate storage, certificate update and certificate revocation are time-consuming and expensive. There is a variety of standards, say the Internet X.509 PKI certificate policy and certification practices framework (RFC 2527), that cover aspects of PKI. However, it lacks predominant governing body to enforce these standards. Despite a certificate authority (CA) is often regarded as trusted, drawbacks in the security procedures of various CAs have jeopardized trust in the entire PKI on which the Internet depends on. For instance, after discovering more than 500 fake certificates, web browser vendors were forced to blacklist all certificates issued by DigiNotar, a Dutch CA, in 2011. An alternative approach to using a certificate to authenticate public key is identity-based cryptography [29], in which the public key of a user is simply his identity, say, his name, email or IP address. A trusted key distribution center generates a secret key for each user corresponding to his identity. When all users have their secret keys issued by the same , individual public keys become obsolete, thus removing the need for explicit certification and all associated costs. These features make the identitybased paradigm particularly appealing for use in conjunction with organization-oriented PDP. For example, a university purchases the cloud storage service for the staff and students, who have a valid E-mail address issued by the IT department of the university. All the members of the university can have a secret key provided by the , say the IT department. The members of the university can store their data together with the meta-data of the file to the cloud. To ensure the data are stored properly, an auditor, a staff of IT department can check the integrity for any member with his E-mail address only, which can relieve the complex key management caused by PKI. The first ID-based PDP was proposed in [30] which converted the ID-based aggregate signature due to Gentry and Ramzan [31] to an ID-based PDP protocol. Wang [32] proposed another identity-based provable data possession in multi-cloud storage. However, their security model called unforgeability for identity-based PDP is not strong enough for capturing the property of soundness in the sense that, the challenged data blocks are not allowed for TagGen queries in this model, which indicates that the adversary cannot access the tags of those blocks. This is clearly not consistent with the real cloud storage where the cloud server is, in fact, storing the tags of all data blocks. Moreover, the concrete identitybased PDP protocol in [32] fails to achieve soundness, a basic security requirement of PDP schemes. The reason is that, the hashed value of each block is used for generating a tag of the block, as a consequence, a malicious cloud server can keep only the hash value of the blocks for generating a valid response to a challenge.

Our Contributions: The contributions of this paper are summarized as follows.

* In an ID-based signature scheme, anyone with access to the signer's identity can verify a signature of the signer. Similarly, in ID-based RDIC protocols, anyone knowing a cloud user's identity, say a third party auditor (TPA), is able to check the data integrity on behalf of the cloud user. Thus, public verifiability is more desirable than private verification in ID-based RDIC, especially for the resource constrained cloud users. In this case, the property of zero-knowledge privacy is highly essential for data confidentiality in ID-based RDIC protocols. Our first contribution is to formalize the security model of zeroknowledge privacy against the TPA in ID-based RDIC protocols for the first time.
* We fill the gap that there is no a secure and novel ID-based RDIC scheme to date. Specifically, we propose a concrete ID-based RDIC protocol, which is a novel construction that is different from the previous ones, by making use of the idea of a new primitive called asymmetric group key agreement [33], [34]. To be more specific, our challenge-response protocol is a two party key agreement between the TPA and the cloud server, the challenged blocks must be used when generating a shared key, which is a response to a challenge from the TPA, by the cloud server.
* We provide detailed security proofs of the new protocol, including the soundness and zero-knowledge privacy of the stored data. Our security proofs are carried out in the generic group model [35]. This is the first correct security proof of ID-based RDIC protocol. Thus, the new security proof method itself may be of independent interest.
* We show the practicality of the proposal by developing a prototype implementation of the protocol. Organization: The rest of the paper are organized as follows. In Section II, we review some preliminaries used in ID-based RDIC construction. In Section III, we formalize the system model and security model of ID-based RDIC protocols. We describe our concrete construction of ID-based RDIC protocol in section IV. We formally prove the correctness, soundness and zero knowledge privacy of our ID-based RDIC protocol in section . We report the performance and implementation results in section VI. Section VII concludes our paper.

**2. PRELIMINARIES**

In this section, we review some preliminary knowledge used in this paper, including bilinear pairings and zero-knowledge proof.

**3. A. Bilinear Pairing**

A bilinear pairing [29] maps a pair of group elements to another group element. Specifically, let be two cyclic groups of order and denote generators of and respectively. A function is called a bilinear pairing if it has the following properties:

Bilinearity: For all and holds.

Non-Degeneracy: where is the identity element of .

Efficient Computation: can be computed efficiently (in polynomial time) for all .

**离散对数等式**

令 是一个有限循环群且有 ，是素数， 是的生成元。下列协议能使一个证明者 向一个验证者 证明两个元素 分别以 和具有相同的离散对数。

承诺： 随机选择 ，计算 并发送 给 。

挑战： 随机选择一个挑战 并把 发送回 。

响应： 计算 并返回给 。

验证： 接受证据当且仅当 成立。

这个协议可以被转换成一个更有效率的非交互版本，表示为 ，通过用承诺的哈希来代替挑战，即 , 其中 是一个安全哈希函数。

**5. ID-Based Signature**

An identity-based signature (IDS) scheme [39], [40] consists of four polynomial-time, probabilistic algorithms described below.

: This algorithm takes as input the security parameter and outputs the master secret key msk and the master public key mpk.

Extract(msk, ID): This algorithm takes as input a user's identity ID, the master secret key and generates a secret key usk for the user.

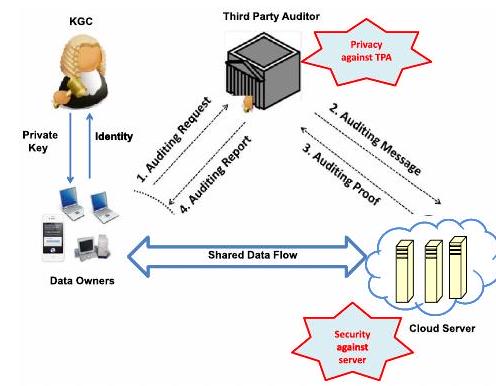


Fig. 1. The system model of identity-based RDIC.

: This algorithm takes as input a user's identity , a message and the user's secret key usk and generates a signature of the message .

: This algorithm takes as input a signature , a message , an identity and the master public key , and outputs if the signature is valid or not.

**3 系统模型和安全模型**

In this section, we describe the system model and security model of identity-based RDIC protocols.

Usually, data owners themselves can check the integrity of their cloud data by running a two-party RDIC protocol. However, the auditing result from either the data owner or the cloud server might be regarded as biased in a two-party scenario. The RDIC protocols with public verifiability enable anyone to audit the integrity of the outsourced data. To make the description of the publicly verifiable RDIC protocols clearly, we assume there exits a third party auditor (TPA) who has expertise and capabilities to do the verification work. With this in mind, the ID-based RDIC architecture is illustrated in Fig 1. Four different entities namely the , the cloud user, the cloud server and the TPA are involved in the system. The KGC generates secret keys for all the users according to their identities. The cloud user has large amount of files to be stored on cloud without keeping a local copy, and the cloud server has significant storage space and computation resources and provides data storage services for cloud users. TPA has expertise and capabilities that cloud users do not have and is trusted to check the integrity of the cloud data on behalf of the cloud user upon request. Each entity has their own obligations and benefits respectively. The cloud server could be self-interested, and for his own benefits, such as to maintain a good reputation, the cloud server might even decide to hide data corruption incidents to cloud users. However, we assume that the cloud server has no incentives to reveal the hosted data to TPA because of regulations and financial incentives. The TPA's job is to perform the data integrity checking on behalf the cloud user, but the TPA is also curious in the sense that he is willing to learn some information of the users' data during the data integrity checking procedure.

**B．系统组成和它的安全性**

一个基于身份的RDIC包括六个算法，Setup, Extract, TagGen, Challenge, ProofGen 和ProofCheck

* Setup 是一个由KGC运行的概率算法。它输入一个安全参数k，输出系统参数param和主密钥 。
* Extract(param, , ) 是一个由KGC运行的概率算法。它输入系统参数param，主密钥 ，和一个用户的身份 ， 输出身份对应的私钥 。
* TagGen(param, 是一个由身份为的数据拥有者运行的概率算法。输入系统参数param，用户的私钥和一个要存储的文件 ，输出每个文件块的标签集合 ，它会和文件一起保存在云上。
* Challenge (param, ,) 是一个由TPA运行的随机算法。输入系统参数param，数据拥有者的身份 ，和一个唯一的 ，输出代表身份对文件名的挑战。
* ProofGen(param, ,, 是一个由云服务器运行的概率算法。输入系统参数 param，挑战信息，数据拥有者的身份，标签 ，文件 和文件名 ，输出一个被挑战块的数据拥有证据。
* ProofCheck ( param, , , ) 是一个由 TPA运行的确定性算法。输入系统参数param，挑战信息 ，数据拥有者的身份，文件名 和一个声称的数据拥有证据 ，输出1或0来表明文件是否完整。

在基于身份的远程数据完整性检查协议中，我们考虑了三个安全属性，即完整性、针对恶意服务器的安全性（可靠性）和针对TPA的隐私（完美的数据隐私）。根据Shacham 和Waters 提出的安全概念，基于身份的RDIC方案对服务器称为安全方案，如果不存在多项式时间算法，可以以不可忽略的概率欺骗TPA并且存在一个多项式时间提取器，它可以通过多次运行挑战响应协议来恢复文件。完整性表明，当与有效的云服务器交互时，验证检查算法将接受证明。可靠性说明，一个可以说服TPA它正在存储数据文件的作弊证明者，实际上是在存储那个文件。我们现在在下面形式化了基于身份的远程数据完整性检查的可靠性安全模型，其中涉及到扮演不受信任服务器角色的敌手和代表数据所有者的挑战者。

对于服务器的安全性：这个安全游戏捕获到，如果一个敌手无法与给定挑战信息对应的用户的所有块，就无法成功生成有效的完整性证明，除非它猜测所有被挑战的块。游戏由一下阶段组成[36]：

* Setup阶段: 挑战者运行Setup算法获得系统参数param和主密钥, 并把param传给敌手，而自己保密。
* Queries阶段: 敌手向挑战者做一系列询问，包括自适应的提取询问和标签询问。

1. 提取询问：敌手可以询问任意身份 的私钥。挑战者通过Extract算法计算私钥 并发送给敌手。
2. 标签询问： 敌手可以在身份下询问任意文件 的标签。挑战者运行Extract算法获得私钥，并运行TagGen算法生成 的标签。最后，挑战者返回标签的集合给敌手。

* ProofGen阶段: 对于一个已经被做过标签询问的文件 ，敌手可以通过指定数据所有者的身份ID 和文件名来执行ProofGen算法。证明过程中挑战者扮演TPA 的角色而敌手 扮演证明者。最后，当协议执行完成时，敌手可以从挑战者那里获得 的输出。
* Output阶段:：最后，敌手选择一个文件名 和一个用户身份 必须没有被做过私钥询问，并且存在一个使用 作为输入的标签询问。敌手输出对一个 可接受的证明者 的描述，其是如下定义的。

我们说一个仿冒的证明者是 可接受的，如果他令人信服地回答了完整性挑战的部分。即， 。这里的概率超过验证者和证明者两者的硬币。敌手赢得游戏，如果如果他能成功输出一个 可接受的证明者。

一个ID-RDIC方案被称为可靠的，如果存在一个提取算法Extr，对于每一个敌手，每当进行可靠性安全游戏，都会基于身份和文件名输出一个 可接受的证明者，Extr算法从中恢复出，例如 ，除非可能性可以忽略不计。

针对TPA的完美的数据隐私：“完美数据隐私”是指TPA没有获得外包数据的任何信息，也就是说，无论TPA学习到什么，TPA都可以自己学习，而无需与云服务器进行任何交互。我们使用模拟器将此模型形式化，如下所示

定义1：一个基于身份的RDIC协议实现完美的数据隐私，如果每一个假冒的验证者，存在一个多项式时间非交互式模拟器 对，对于每一个有效的公共输入, chal, Tag和私人输入，以下两个随机变量计算不可区分：

1. ，其中 表示该协议使用的随机硬币。
2. 。

也就是说，模拟器只获取公共输入的信息，并且与服务器没有交互，但仍然设法输出一个与在交互过程中学习到的任何无法区分的响应。

**4 OUR CONSTRUCTION**

在本节中，我们提供了一个基于身份的安全远程数据完整性检查协议的具体构建，支持完美的数据隐私保护。我们的方案的工作原理如下。在密钥提取中，我们采用Boneh等人[37]提出的短签名算法对用户的身份进行签名，获得用户的密钥。在TagGen中，我们发明了一种新的算法来生成文件块的标记，当计算对挑战的响应时，这些标记可以聚合为单个元素。在挑战阶段，TPA通过选择块的一些索引和随机值来挑战云。在证明生成过程中，云服务器使用被挑战的块计算响应，获得相应的明文并将其转发给TPA。协议的细节如下：

Setup：输入一个安全参数，KGC选择两个素数阶的乘法循环群和，是的生成元。存在一个双线性映射。KGC随机选择作为主密钥，并设置。最后，KGC选择三个哈希函数 : 和。系统参数为，

Extract：输入主密钥 和一个用户身份 ，这个算法输出用户的私钥 。

TagGen：给定一个文件 ，文件名为fname，数据拥有者首先将它分成 个文件块，其中 并选择一个随机数 ，计算 。对于每一个块，数据拥有者计算

文件块 的标签就是 。数据拥有者把文件M和 name 一起存储到云中，其中IDS name 是一个由数据拥有者对 name做的基于身份的签名。

Challenge：为了检查M的完整性，验证者随机选择一个集合的c个元素的子集，对于每个 ，选择一个随机元素。令 为的集合。为了生成挑战信息，验证者选择一个随机的 ，计算 并如下操作：

1. 计算。
2. 生成一个证据：

验证者发送挑战信息 给服务器。

GenProof：一旦收到挑战信息，服务器首先计算出。然后，它验证完整性证据。如果它是无效的，审计终止。否则，服务器计算 and 和 并把返回给验证者。

CheckProof：一旦收到服务器发来的 ，验证者检查是否IDS name 是否是数据拥有这对消息 的有效签名。如果不是，完整性证据是无效的。否则，验证者检查是否

如果等式成立，验证者接受证据；否则证据是无效的。Challenge, CheckProof 和GenProof 的过程总结在 Fig. 中。

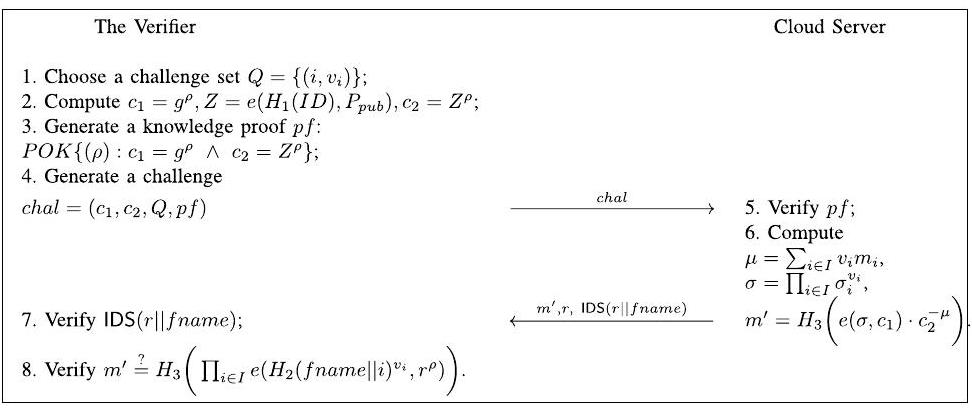


Fig. 2. Identity-based remote data integrity checking protocol.

**5 安全性分析**

在本节中，我们证明了该方案实现了完整性、可靠性和完美的数据隐私保护的特性。完整性保证了协议的正确性，而可靠性表明该协议对不受信任的服务器是安全的。完美的数据隐私表明，该协议不会向验证者泄露所存储文件的信息。**A. 完整性**

如果数据所有者和云服务器都是诚实的，那么对于每个有效的标签和一个随机的挑战，云服务器总是可以通过验证。该协议的完整性可以详细阐述如下。

**B. 可靠性**

先前协议的响应通常包括项，这有助于构建一个提取器来提取被挑战的块。例如，在Ateniese等人[4]的协议中，他们利用指数假设的知识构建了这样的提取器。然而，在我们的新协议中没有这部分。在这一部分中，我们证明了任何一般算法在打破我们的协议的可靠性方面的概率都是可以忽略不计的。一般算法是一种不利用底层有限循环群的代数结构的算法。一般群模型旨在捕获任何一般算法的行为。我们简要回顾一下这个想法。设是一个素数，而对于是两个独立的随机编码函数。一般群表示为。由于一般算法不利用群的结构，我们说给定一个元素，除了相等之外，不能推断出任何关于该元素的东西。两个喻言机， 用于模拟群组的动作：对于任何元素和，输入查询，返回元素（用于乘法）或（用于除法）。另一个喻言机，，用于表示的配对操作。具体来说，输入和，返回。

证明：在讨论了设置之后，我们将证明我们的构造对于随机模型中的任何通用算法都是安全的。假设是一个通用对手，分别对做次查询，我们展示如何构造一个模拟器来模拟一般群模型中的视图，并从中提取消息的集合。

* 初始化。在一般群模型和随机喻言模型中，公开参数可以表示为

其中，前三个喻言机表示双线性配对运算和群运算，后三个喻言机表示哈希运算，分别表示元素和的编码。它们是由选择的随机比特串。将编码与常数多项式1关联，将与多元多项式关联。我们可以看到，在任何阶段，任意群元素都将与一个多元多项式相关联。具体来说，呈现给的元素都将与变量中的一个多元多项式相关联。我们将讨论多项式关联的规则。

* 哈希询问：对于任意输入对的询问，返回一个随机比特串并将其与一个多元多项式关联。对于任意输入对的询问，返回一个随机比特串并将其与一个多元多项式 关联。对于任意对的询问，返回一个随机比特串（不是一个群元素）。
* 群的运算（喻言机）。提交给两个元素和两个整数。搜索它的历史记录来定位和相关的多项式（假设它们分别是 和）。如果这些元素不存在，则意味着不是有效的群元素，拒绝该运算。计算，得到的多项式记为。搜索它的历史记录来检查是否中是否存在与关联的元素。如果是，则返回。否则，选择一个随机比特串并将它与关联。返回给。注意元素表示群运算的次方乘的次方的结果。
* 群的运算（喻言机）。其处理方式与O1查询的处理方式相同。
* 双线性配对运算（喻言机）。提交给两个元素。搜索它的历史记录来定位和相关的多项式（假设它们分别是 和）。如果这些元素不存在，则意味着不是有效的群元素，拒绝该运算。计算 ，得到的多项式记为。搜索它的历史记录来检查是否中是否存在与关联的元素。如果是，则返回。否则，选择一个随机比特串并将它与关联。返回给。注意元素表示元素和的双线性配对的结果。
* 提取询问。提交给一个身份。定位一个具有相同输入的询问。如果没有，它将隐式地进行一次询问。它得到表示的元素，记为并把它和多项式关联。如果中存在一个元素，其已经和多项式关联，返回。否则，随机选择一个随机比特串将它与多项式关联并将它返回给。
* 标签询问。提交给一个身份和文件，文件名为 。用定位一个提取询问。如果没有，它将隐式地用作为输入进行一次提取询问。假设元素与多项式关联。还会获得一个元素的集合 。（如果不存在，进行询问 ）。假设元素跟多项式相关联，选择一个随机比特串并将它和多项式关联。对于，计算多项式。如果一个元素已经和多项式关联，返回作为文件名的文件块的标签。否则，选择一个随机比特串并将它和关联，并返回作为文件块的标签。比特串在这个阶段也交给敌手。注意，表示对于一个文件，文件名为。一个对 基于身份的签名IDS 也被交给敌手。

在某个阶段，IDA和某个文件决定在一个M=(m1，...，mn)上被挑战，但A从未对I进行过提取查询 D. 我们可以安全地假设A已经对I进行了哈希查询 D. 我们假设元素H1(ID)与多项式I相关联。s使用ID和M作为输入，使用a进行TagGen查询。假设返回到A的元素为(ξηj，ξj，1，...，ξj，n)。为了清晰起见，下面我们将去掉下标j。现在，S扮演着挑战者的角色，而a在协议中扮演着云服务器的角色。在某个阶段，决定被一个确定的身份和一个确定的名为的文件挑战，限制为没有对做过提取询问。我们可以安全地假设对做过哈希询问。我们假设元素t 与多项式关联。实施一次标签查询以使用和作为输入。假设返回给的元素是。为了清晰起见，下面我们将去掉下标。在，扮演着挑战者的角色，而在协议中扮演着云服务器的角色。

* （挑战）。选择一个子集，且对于每一个 ，选择。进一步选择一个随机比特串并将它与多项式关联。它还会选择另一个随机比特串并将它与多项式关联。和分别表示群元素和群元素。还会选择一个随机比特串并将它与多项式关联。和一个模拟证据一起交给。
* （多项式计算）。回想一下，是与关联的多项式。令是与元素关联的多项式。另外，令是与元素关联的多项式。因此，的标签会与多项式相关联。在这个阶段，随机选择除以下变量之外的所有变量的值：

换句话说，所有给的多项式都是变量中的多元多项式。实例化不会影响前一个模拟的视图，只要它不会创建任何“不正确”的关系。

* （生成证据）。最后返回一个值，和一个群元素以及它的基于身份的签名IDS 。

在随机喻言模型中，成功返回的唯一方式就是做一个中元素的查询。由于基于身份的签名的不可伪造性，我们可以安全地假设（代表验证过程中使用的元素）是在标签询问中返回给的值。

为了使以不可忽略的概率获胜，必须与一个多项式关联起来：

仍然需要证明，为了生成一个关联多项式满足上述等价关系的元素，必须进行一组群运算查询，从而允许提取常数集。此外，这些询问是在挑战阶段之后进行的。首先，请注意，对于每个询问，最多获得一个新的群元素，因此由获得的元素总数是有限的。在挑战阶段之前，没有这样的元素，其多项式与变量或相关联。在挑战阶段之前，每一个中的元素都与下列形式的多项式相关联：

其中是对已知的系数。同样，群中的每个元素都与以下形式的多项式相关联：

其中和是与群中的元素相关联的多项式，且是一个常数。很容易看出，在挑战前获得的多项式将不能满足所需的关系，因为缺少变量。

在挑战阶段之后，可以访问两个额外的元素，它们的多项式分别是中的和中的。由此可见，由得到的所有的元素都与以下形式的多项式相关联：

其中所有的系数对和都是已知的。将这些系数等于，除了 和 所有的系数都是0。特别地，对于所有，且。

换句话说，可以从中提取项，也就是。使用交互作用，可以计算的所有值。此外，观察到导致这个系数的运算必须在挑战后发布，因为它们与多项式有关。这就完成了证明。

**C．完美的数据隐私保护**

为了证明该方案保持了数据隐私，我们展示了如何构建一个能够访问验证器的模拟器，可以在不知道数据文件块也不知道相应的的情况下模拟远程数据完整性检查协议。

我们假设被发送给。换句话说，我们的协议既不保存文件名也不保存参数的隐私。因为，其中是一个由数据拥有者选择的随机值，我们可以合理地说不包含文件库的信息。下面我们展示怎么回答由提交的挑战。

一旦收到来自的挑战信息，将解析为。接下来，从V中提取值。由于的可靠性，得到了，和。将解析为，并计算。输出作为对这一挑战的响应。

对于每个挑战，都有一个有效的唯一值。因此，上述模拟是完美的。

TABLE I

SUMMERISE OF THE TIME COST FOR A 1 MB FILE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Setup | Extract | TagGen: off-line | TagGen: on-line | Challenge | GenProof | CheckProof |
|  |  | second | second | 351 ns per challenge | per challenge | per challenge |

say does not contain the information of the file blocks. Below we show how answers the challenge given by a verifier .

Upon receiving the challenge chal from parses chal as . Next, extracts from the value . Due to the soundness of obtains such that and . parses as and computes outputs as the response to this challenge.

For each challenge , there is a unique value that is valid. Thus, the above simulation is perfect.

**14. PERFORMANCE AND IMPLEMENTATION**

In this section, we first numerically analyze the cost of the new protocol, and then report its experimental results.

**15. A. Numerical Analysis**

We provide a numerical analysis of costs regarding computation, communication and storage of the proposed protocol in this part.

Computation Cost: We present the computation cost from the viewpoint of the , the data owner, the cloud server and the verifier (TPA). For simplicity, we denote by the exponentiations in and the multiplication in and the pairing computation and the mapto-point hash function respectively. We ignore the ordinary hash functions in numerical analysis since the cost of these functions is negligible compared with other operations.

The primary computation of the is generating system parameters and private key for each user. Thus, the main computational cost of is . The dominated computation of data owner is generating tags for file blocks as , which is the most expensive operation in the protocol but fortunately it can be done offline. For a file with blocks, the cost of the data owner is . The main cost of the verifier is generating a challenge and checking the validity of a proof. The verifier needs to perform 1 pairing operation, and 6 exponentiations in to generate a challenge when using the proof of equality of discrete logarithm given in [38]. When checking a proof, the cost of the verifier is . The main computation cost of generating a proof by the cloud server is calculating the aggregation of , that is , and the total cost is .

Communication Cost: In the challenge phase, the verifier sends to the server, where denotes the challenge set. In practice, we can employ a pseudorandom function and a pseudo-random permutation as the public parameters. The verifier only needs to include a key of and a key of in the challenge, instead of the challenge set , which can reduce the communication cost significantly. In this case, the communication cost is of binary length . In the response phase, the server returns as the response to the verifier. An identity-based signature usually contains two points ( 320 bits) of elliptic curves, thus the communication cost of the response is of binary length

Storage Cost: Regarding the storage cost of the cloud server, the data owner and the verifier, since identity-based data auditing schemes are publicly verifiable, both the data and the tags are stored on the cloud server. An identity-based digital signature algorithm is used to protect the commitment from being tampered by external and internal adversaries. In this case, what stored on the cloud are as follows.



The data owner needs to store nothing in the context of public verifiability. With the knowledge of the identity information of the data owner, a verifier is able to check the integrity of the data on behalf the data owner. Thus, what the verifier needs to store is the challenge set and the value randomly selected by himself. The storage cost of a verifier can be reduced to bits. The storage cost of the block tags is upper bounded by bits when employing proper elliptic curves [29]. The cloud server hosts the data, the tags, and its signature. As a consequence, the storage cost of the cloud server is upper bounded by .

**16. B. Implementation Results**

The implementation was conducted with pbc-0.5.13 [41] with pbc wrapper-0.8.0 [42] on Intel i7-4700MQ CPU @ . The memory is always sufficient since the scheme only requires a polynomial space. In our implementation, we made use of parameter a.param, one of the standard parameter settings of pbc library. Parameter a.param provides a symmetric pairing with the fastest speed among all default parameters. The implementation time overheads of the protocol are displayed in the following two parts.

In the first part, we setup a file of a constant size of , and observe the impact of the number of challenged blocks in terms of the time cost. In our setting, the size of a data block is bounded by the group order , i.e., 160 bits. Hence, we have 50,000 blocks in total. This implies that the timing results for Setup, Extract and TagGen steps are constant for this part. See Table I for more details. We can see that both the Setup algorithm and the Extract algorithm are extremely fast. The Setup algorithm picks some random values and compute a modular exponentiation in , which costs , and the Extract algorithm needs to perform one modular exponentiation in for generating the private key of a cloud user, which cost . The TagGen algorithm is

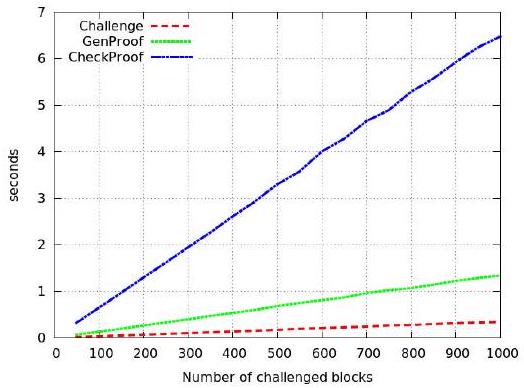


Fig. 3. Increasing number of challenges for fixed size of file.

expensive and we show that the TagGen timing result consists of two phases, an off-line phase, where the data owner can preprocess without knowing the actual data; and an on-line phase, where the data owner needs to compute for each data block. (Note that since all those exponentiations are carried out with a same base, we can use a lookup table to accelerate those single-base multiple-exponent operations.) The time cost of off-line computation of generating tags for 1 MB file is seconds while the on-online time cost is seconds. This is an acceptable result compared with the previous preprocessing result in [5]. Note that we can employ the technique of verifiable outsourcing computation [43]-[46] which enables a data owner to offload the computation of some function, to other perhaps untrusted servers such as a cloud, while maintaining verifiable results, to improve the efficiency of tag generation if necessary.

We then increase the number of challenged blocks from 50 to 1000 with an increment of 50 for each test to see the time cost of Challenge, GenProof and CheckProof steps. As one shall see from Figure 3 , the timing cost of those three parts increases with the increase of the number of challenges. This is consistent with the empirical analysis because when the number of challenged block rises, more random values need to be selected in the challenge algorithm, and the cloud server has an increasing computations of while the TPA has more exponentiations, multiplication operations and pairing computations in . But it is fair to say that the proposed identity-based RDIC protocol is efficient for both the cloud server and the verifier. For example, let us investigate a point on the curve. Ateniese et al. [4] demonstrated that, if the cloud server has polluted of the blocks, then the verifier can challenge 460 blocks in order to achieve the probability of server misbehavior detection of at least . We can see that it costs the verifier only about seconds to verify a response and the server seconds to generate a response when challenging 460 blocks.

In the second part, we test the most expensive algorithm TagGen of the protocol by increasing the file size from to , that is, from 10,000 blocks to 100,000 blocks

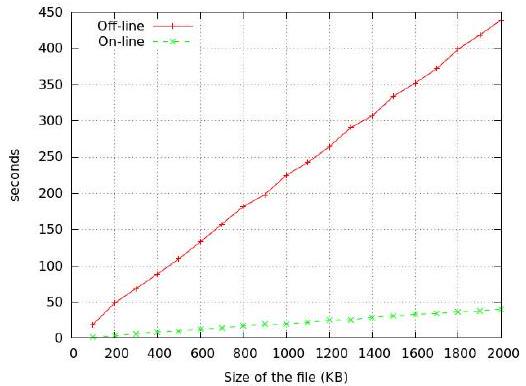


Fig. 4. Tag generation time for increased size of files.

accordingly, and record the time for TagGen. As expected, both on-line and off-line time to generate tags for a given file increases almost linearly with the increase of the file size. Figure 4 gives more details. The implementation shows that generating tags is more expensive than other parts but fortunately, computing tags for a file is a one time task, as compared to challenging the outsourced data, which will be done repeatedly. Since the cloud users can do the off-line work completely parallelizable in advance, we pay only attention to the on-line cost. We can see that the efficiency of TagGen of our protocol is comparable to that of the existing well-known schemes, say [5]. To generate tags for a file, it costs almost 42 seconds. As such, one shall be able to anticipate the time cost of generating tags for any size of files.

**17. CONCLUSION**

In this paper, we investigated a new primitive called identity-based remote data integrity checking for secure cloud storage. We formalized the security model of two important properties of this primitive, namely, soundness and perfect data privacy. We provided a new construction of of this primitive and showed that it achieves soundness and perfect data privacy. Both the numerical analysis and the implementation demonstrated that the proposed protocol is efficient and practical.

**18. REFERENCES**

[1] P. Mell and T. Grance. (Jun. 3, 2009). Draft NIST Working Definition of Cloud Computing. [Online]. Available: <http://csrc.nist.gov/groups/SNC/cloud-computing/index.html>

[2] Cloud Security Alliance. (2010). Top Threats to Cloud Computing. [Online]. Available: <http://www.cloudsecurityalliance.org>

[3] M. Blum, W. Evans, P. Gemmell, S. Kannan, and M. Naor, "Checking the correctness of memories," Algorithmica, vol. 12, no. 2 , pp. 225-244, Sep.

[4] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 598-609.

[5] G. Ateniese et al., "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Security, vol. 14, no. 1, May 2011, art. no.

[6] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, Pp.

[7] H. Shacham and B. Waters, "Compact proofs of retrievability"' in . Adv. Cryptol. -ASIACRYPT, 2008, pp. 90-107. [8] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. ASIACRYPT, 2009, pp. 319-333.

[9] A. F. Barsoum and M. A. Hasan, "Provable multicopy dynamic data possession in cloud computing systems," IEEE Trans. Inf. Forensics Security, vol. 10, no. 3, pp. 485-497, Mar. 2015 .

[10] 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.

[11] J. Liu, K. Huang, H. Rong, H. Wang, and M. Xian, "Privacypreserving public auditing for regenerating-code-based cloud storage," IEEE Trans. Inf. Forensics Security, vol. 10, no. 7, pp. 1513-1528, Jul. 2015

[12] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing,' in Proc. Comput. Security-ESORICS, vol. 5789. 2009, pp. 355-370.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1-9.

[14] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," IEEE Netw., vol. 24, no. 4, pp. 19-24, Jul./Aug. 2010

[15] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 22 , no. 5 , pp. , May

[16] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage" IEEE Trans. Comput., vol. 62, no. 2, pp. 362-375, Feb. 2013 .

[17] . 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. , Sep. 2013 .

[18] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds," IEEE Trans. Services Comput., vol. 6, no. 2, pp. 227-238, Apr./Jun.

[19] Y. Zhu, H. Hu, G.-J. Ahn, and S. S. Yau, "Efficient audit service outsourcing for data integrity in clouds," J. Syst. Softw., vol. 85, no. 5 , pp.

[20] H. Wang and Y. Zhang, "On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. , Jan.

[21] J. Wang, X. Chen, X. Huang, I. You, and Y. Xiang, "Verifiable auditing for outsourced database in cloud computing," IEEE Trans. Comput., vol. 64, no. 11 , pp. , Nov. 2015

[22] Y. Yu, Y. Li, J. Ni, G. Yang, Y. Mu, and W. Susilo, "Comments on 'Public integrity auditing for dynamic data sharing with multiuser modification'," IEEE Trans. Inf. Forensics Security, vol. 11, no. 3 , pp. 658-659, Mar.

[23] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. SecureComm, 2008, art. no.

[24] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage secruty in cloud computing," in Proc. , vol. 2009. 2009, pp. 1-9.

[25] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," ACM Trans. Inf. Syst. Security, vol. 17, no. 4, 2015 , art. no.

[26] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "MuR-DPA: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud," IEEE Trans. Comput., vol. 64, no. 9, pp. , Sep. 2015 .

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

[28] G. Ateniese, A. Faonio, and S. Kamara, "Leakage-resilient identification schemes from zero-knowledge proofs of storage", in Proc. IMA Int. Conf., 2015 , pp.

[29] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Proc. CRYPTO, vol. 2139. 2001, pp. 213-229.

[30] J. Zhao, C. Xu, F. Li, and W. Zhang, "Identity-based public verification with privacy-preserving for data storage security in cloud computing," IEICE Trans. Fundam. Electron., Commun. Comput. Sci., vol. E96-A, pp. , Dec.

[31] C. Gentry and Z. Ramzan, "Identity-based aggregate signatures," in Proc. Public Key Cryptography, vol. 3958. 2006, pp. 257-273.

[32] H. Wang, "Identity-based distributed provable data possession in multicloud storage," IEEE Trans. Service Comput., vol. 8, no. 2, pp. , Mar./Apr. 2015 [33] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement," in Proc. EUROCRYPT, vol. 5479. 2009, pp.

[34] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and Z. Dong, "Roundefficient and sender-unrestricted dynamic group key agreement protocol for secure group communications," IEEE Trans. Inf. Forensics Security, vol. 10 , no. 11 , pp. , Nov. 2015

[35] V. Shoup, "Lower bounds for discrete logarithms and related problems," in Proc. EUROCRYPT, 1997, pp. 256-266.

[36] Y. Yu, Y. Zhang, Y. Mu, W. Susilo, and H. Liu, "Provably secure identity based provable data possession," in Proc. Provable Security, vol. 2015 , pp.

[37] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing", . Cryptol., vol. 17, no. 4, pp. 297-319, 2004.

[38] D. Chaum and T.P. Pedersen, "Wallet databases with observers"' in Proc. CRYPTO, 2001, pp. 89-105

[39] J. C. Cha and J. H. Cheon, "An identity-based signature from gap DiffieHellman groups", in Proc. PKC, vol. 2567. 2003, pp. 18-30.

[40] F. Hess, "Efficient identity based signature schemes based on pairings," in Proc. Sel. Areas Cryptography, vol. 2595. 2003 , pp. .

[41] B. Lynn. (2013). The Pairing-Based Cryptography Library (0.5.13). [Online]. Available: <http://crypto.stanford.edu/pbc/>

[42] A. Kate. The Pairing-Based Cryptography Library- per Classes , accessed on Sep. 8, 2015. [Online]. Available: <http://crysp.uwaterloo.ca/software/PBCWrapper/>

[43] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in Proc. CRYPTO, vol. 6223. 2010, pp.

[44] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," IEEE Trans. Depend. Sec. Comput. vol. 12 , no. 5, pp. 546-556, Sep./Oct. 2015

[45] J. Lai, R. H. Deng, H. Pang, and J. Weng, "Verifiable computation on outsourced encrypted data," in Proc. ESORICS, 2014, pp. 273-291.

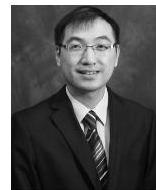
[46] X. Liu, R. Choo, R. Deng, R. Lu, and J. Weng, "Efficient and privacypreserving outsourced calculation of rational numbers," IEEE Trans. Depend. Sec. Comput., to be published.

Yong Yu received the Ph.D. degree in



cryptography from Xidian University in 2008 . He is currently a Professor of Shaanxi Normal University. He holds a one hundred talent Professorship of Shaanxi Province as well. He has authored of Shaanxi Province as well. He has authored His research interests are cryptography and its applications, especially public encryption, digital signature, and secure cloud computing.

Man Ho Au (M'12) received the Ph.D. degree



from the University of Wollongong, Australia, in 2009 . He is currently an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He has authored over 100 papers in refereed international conferences and journals. His research interests include information security and privacy, applied cryptography, accountable anonymity and cloud computing. He has served as a Program Committee Member in many international conferences. He is also an Associate Editor of the Journal of Information Security and Applications.



Giuseppe Ateniese was with Sapienza-University of Rome, Italy, and Assistant/Associate Professor with Johns Hopkins University, USA, and one of the founders of the JHU Information Security Institute. He was a Researcher with IBM Zurich Research Laboratory, Switzerland, and a Scientist with the Information Sciences Institute, University of Southern California, USA. He also briefly worked as a Visiting Professor with Microsoft, Redmond, USA. He is currently the David and Farber Endowed Chair in computer science and the Department Director with Stevens Institute of Technology. He received the NSF CAREER Award for his research in privacy and security, and the Google Faculty Research Award and the IBM Faculty Award for his research on cloud security. He has contributed to areas such as proxy re-cryptography, anony mous communication, two-party computation, secure storage, and provable data possession. He has served in the program committees of international security conferences including the ACM CCS, IEEE Oakland, and NDSS and as Panelist of the US National Science Foundation. He is currently working on cloud security and machine learning applied to security and intelligence issues for which he received an IBM SUR Award. He is also investigating new security applications for decentralized computing based on the blockchain/bitcoin technology.



Xinyi Huang received the Ph.D. degree from the School of Computer Science and Software Engineering, University of Wollongong, Australia. He is currently a Professor with the School of Mathematics and Computer Science, Fujian Normal University, China, and the Co-Director of Fujian Provincial Key Laboratory of Network Security and Cryptology. He has authored over 100 research papers in refereed international conferences and journals. His research interests include applied cryptography and network security. His work has been cited more than 2900 times at Google Scholar (H-Index: 27). He is also an Associate Editor of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. Willy Susilo received the Ph.D. degree in



computer science from the University of Wollongong, Wollongong, NSW, Australia. He is currently a Professor, the Head of the School of Computing and Information Technology, and the Director of Center for Computer and Information Security Research, University of Wollongong. His current research interests include cloud security, cryptography, and information security. He is also an Associate Editor of many international journals, such as the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. He has served as a Program Committee Member in many international conferences. He is also the Editor-in-Chief of Information Journal. He has been awarded with a prestigious Future Fellowship by the Australian Research Council.

Yuanshun Dai is currently a Professor with the



University of Electronic Science and Technology of China (UESTC) and the Dean of School of Computer Science and Engineering. UESTC He has authored over 100 referred journal and conference papers. His research interest is cloud computing.



Geyong Min received the [B.Sc](http://B.Sc). degree in computer science from the Huazhong University of Science and Technology, China, in 1995, and the Ph.D degree in computing science from the University of Glasgow, U.K., in 2003 . He is currently a Professor of high performance computing and networking of high performance computing and networking Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, U.K. His Physical Sciences, University of Exeter, U.K. His networks, wireless communications, multimedia networks, wireless communications, multimedia systems, information security, high performance computing, ubiquitous computing, modeling, and performance engineering.