# Secure and Efficient Privacy-Preserving Identity-Based Batch Public Auditing with Proxy Processing

**Jining Zhao[1], Chunxiang Xu[1] and Kefei Chen[2]**
[1] Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China
Chengdu, 610051, China
[e-mail: kevinchao86@gmail.com;chxxu@uestc.edu.cn]
[2] Hangzhou Key Laboratory of Cryptography and Network Security, Hangzhou Normal University
Hangzhou, 311121, China
[e-mail: kfchen@hznu.edu.cn]
*Corresponding author: Jining Zhao

*Received June 28, 2018; revised October 11, 2018; accepted November 21, 2018; published February 28, 2019*

## Abstract

With delegating proxy to process data before outsourcing, data owners in restricted access could enjoy flexible and powerful cloud storage service for productivity, but still confront with data integrity breach. Identity-based data auditing as a critical technology, could address this security concern efficiently and eliminate complicated owners' public key certificates management issue. Recently, Yu et al. proposed an Identity-Based Public Auditing for Dynamic Outsourced Data with Proxy Processing (https://doi.org/10.3837/tiis.2017.10.019). It aims to offer identity-based, privacy-preserving and batch auditing for multiple owners' data on different clouds, while allowing proxy processing. In this article, we first demonstrate this scheme is insecure in the sense that malicious cloud could pass integrity auditing without original data. Additionally, clouds and owners are able to recover proxy's private key and thus impersonate it to forge tags for any data. Secondly, we propose an improved scheme with provable security in the random oracle model, to achieve desirable secure identity based privacy-preserving batch public auditing with proxy processing. Thirdly, based on theoretical analysis and performance simulation, our scheme shows better efficiency over existing identity-based auditing scheme with proxy processing on single owner and single cloud effort, which will benefit secure big data storage if extrapolating in real application.

***Keywords:*** Cloud computing, identity-based cryptography, public auditing, proxy, security analysis

## 1. Introduction

By offering flexible access and powerful data management services, cloud storage plays an indispensable role in creating revenue for enterprises' business and benefiting individuals' life. From reports of IDG and Garter, 127 billion USD is spent globally on public cloud in 2017 [1], and approximately 28% of the total market revenue of cloud service, will be produced from infrastructure, application and business processing service in 2021 [2]. In the age of "Big Data" with critical "Data that is big", cloud data storage size will swell to trillion gigabytes in 2025[1]. By managing huge data on different storage clouds, a great number of data owners enjoy customized applications for their business or utilities. When data owners' mobile devices are of limited computation capacity or belong to an organization, their connections to remote cloud are always controllable. In such restricted outsourcing setting, a proxy with authorizations, could help data owners to perform data processing tasks before outsourcing them to remote clouds [3]. However, security risk of outsourcing data integrity still remains as major problem for cloud storage services even if equipped with encryption technologies [4], since outsourced data may be tampered by cloud system failure and external attack. To keep good reputations, cloud storage providers might also hide from the data owners that service quality is deteriorated by deleting data to save cost.

Fortunately, with remote integrity checking technology [5], data owners could turn to a third party auditor (TPA) for public auditing service, but great tasks from huge data of multiple owners will degrade TPA's performance and it is also undesirable when individual owner's data content is visible for TPA during the auditing process. Therefore, in the delegated proxy processing setting, it is imperative to enable secure and efficient remote integrity auditing for multiple owners. Meanwhile, data should be privacy-preserving if integrity auditing is conducted publicly by a third party auditor.

Provable Data Possession (PDP) [5] proposed by Ateniese, as a critical probabilistic remote integrity checking technology, could allow efficient data integrity auditing without having to download the entire data copy. With error correcting code, Shacham designed proof of retrievability [6] to check possibility of polynomial time data recovering. In 2010, based on Public key Infrastructure (PKI), Wang et al. supported cloud data integrity public auditing for the first time in [7], by employing a third party auditor to perform PDP in a privacy preserving manner. For auditing scalable storage data, distributed cloud data integrity for single owner was studied by Zhu et al.'s cooperative PDP [8], and Yang et al. made further effort of enabling the multiple clouds data integrity auditing for multiple data owners [9]. There are also data auditing schemes with special features, such as multiple data storage replica [10] and group user data share [11] and revocation. In [12][13], PDP scheme is investigated to support auditing for data with dynamic update. For recent years, continuous progress were made on cloud data auditing in [14-16] and key word search on encrypted data for fog computing and crowdsourcing [17-20]. However, these famous works were all built on PKI, where each owner's public key certificate is required to be transferred to check public key indeed belonging to the owner.

To eliminate the complicated management issue of public key certificates, Zhao et al. proposed the first identity-based public auditing scheme [21] with PDP, to enable public auditing with identity based cryptography [22] and privacy-privacy auditing with TPA. In 2015, Wang et al. designed the identity based distributed PDP [23] to support multi-cloud storage for single owner. By combining PKI based PDP and Identity based signature [24], in

2016, Liu et al. considered generic construction of identity-based PDP [25]. Later, Yu et al. enabled zero knowledge privacy integrity checking for identity based PDP in [26]. For data auditing in the cloud access restricted setting, Wang et al. for the first time proposed an identity based PDP scheme with authorized proxy to process data in [27]. This work could support single owner's data on single storage cloud, but not considering privacy-preserving issue for public auditing. There is also design on the lattices cryptography [28]. Spontaneously, security flaws were found in some classic design but luckily were repaired in [25][29][30]. So the challenging problem still remains to be unsolved, i.e., how to efficiently perform multiple cloud data auditing with all the following desirable features: 1) by identity-based cryptography, 2) with privacy-preserving, 3) for multiple data owners, and 4) with proxy data processing.

In 2017, Yu et al. designed an identity based batch public auditing scheme [32], trying to facilitate secure data integrity auditing to address the challenges mentioned above. However, after careful analysis upon potential malicious behaviors, this work is not able to achieve better efficiency and security simultaneously.

**Contributions:** Firstly, for the sake of data security, we demonstrate that Yu et al.'s work [32] is vulnerable to data loss and proxy private key recovering attacks. On one hand, malicious clouds are able to use masked data in place of original data to pass integrity auditing. For the other, arbitrary two pairs of data and tags are sufficient to recover private key of owners' authorized proxy. In this way, the exclusive right of generating proxy tag will be undermined by clouds and data owners. Secondly, we propose our improved scheme for this proxy processing setting, which could perform identity-based privacy-preserving batch public auditing and resist these above security flaws. Thirdly, we prove security of our scheme in random oracle under CDH, BDH and DL assumptions. In the end, with the extensive overhead analysis and simulation, our improved scheme illustrates better auditing efficiency over an identity-based proxy-oriented data uploading and remote data integrity checking in public cloud (ID-PUIC) [27] with single owner effort on single storage cloud, such that it could contribute to secure big data storage if extrapolated to real application.

**Paper Organization:** The rest of the paper starts with notations in Section 2 and reviews of system model of identity-based batch public auditing with proxy processing scheme (ID-BPAPP) along with its system components and security model in Section 3. After revisiting of Yu et al.'s construction of an ID-BPAPP scheme in Section 4, two security shortcomings are demonstrated in Section 5. We present our improved scheme Sec-ID-BPAPP in Section 6, and formally prove its security in Subsection 6.1 under random oracle model. In Section 7, we compare our improved scheme with Wang et al.'s ID-PUIC, in the context of overheads theoretical analysis and simulation, to study the trend of efficiency for computation and communication. Section 8 concludes our paper.

## 2. Preliminary

### 2.1 Notations and computational assumption

- $G_1$ and $G_2$ are two cyclic groups of same large prime order $q$, additive and multiplicative groups respectively. $e$ is a bilinear pairing mapping, where $e: G_1 \times G_1 \rightarrow G_2$.
- $(mpk, msk)$ are the Private Key Generator (PKG)'s master public and private key pair. $sk_i$ is $i$-th data owner's corresponding identity-based private key.
- There are $n_O$ number of data owners, outsourcing total $N$ number of blocks, on $n_J$ number of clouds. $\tilde{F}_{ijk}$ is the $i$-th data owner's $k$-th block outsourced on $j$-th cloud $CS_j$. $\sigma_{ijk}$ is the tag of block $\tilde{F}_{ijk}$.

- $f$ is a pseudo random function (PRF) $f: Z_q \times \{1, \cdots, N\} \to Z_q$ for generating challenging co-efficient to combine challenged blocks.
- $\pi$ is a pseudo random permutation $\pi: Z_q \times \{1, \cdots, N\} \to \{1, \cdots, N\}$ for generating index of challenged block.
- $chal$ is the challenge token generated by auditor, and $chal_j$ is the specific challenge token for $CS_j$. $c_{ij}$ is number of challenged blocks for $i$-th owner on $CS_j$, where $c_{ij} < N$.
- $a_{ij} \in [1, c_{ij}]$ indicates the $a_{ij}$-th selected block of total $c_{ij}$ challenged blocks, which should further specify index of $i$-th data owner's $k$-th block outsourced on $j$-th cloud, i.e., $k = \pi_{v_{ij,1}}(a_{ij})$.
- $C$ is the index set of challenged clouds picked by auditor. $O$ is the index set of data owners on challenged blocks, and $J$ is the index set of challenged clouds, where $|O| = n_1$, $|J| = n_2$. $P_j$ is the proof of storage generated by challenged cloud $CS_j$.

**CDH problem** on $G_1$: Given $g, g^a, g^b \in G_1$, to compute $g^{ab}$ with a probabilistic polynomial time (PPT) algorithm, without knowing random $a, b \in Z_q$.

**BDH problem** on $G_2$: Given $g, g^a, g^b, g^w \in G_1$, to compute $e(g,g)^{abw} \in G_2$ with a PPT algorithm, without knowing random $a, b, w \in Z_q$.

**DL problem** on $G_2$: For $g' \in G_2$, given $g'^a$, to compute $a$ with a PPT algorithm.

## 3. System model and Security model of ID-BBPAP system

In this section, we will first present system model of Identity-Based Batch Pubic Auditing scheme with Proxy Processing (ID-BPAPP) from the original paper [32]. The system components are described with general structures of seven algorithms. We also give the security model of the ID-BPAPP system.

### 3.1 System Model

As it depicts in **Fig. 1**, there are five kinds of entities in an ID-BPAPP scheme, i.e., the *PKG*, data *Owners*, *Proxy*, multiple *Clouds* ($\{Cloud_j\}$), and a *TPA*. *PKG* initializes the system parameters and extracts private keys for data owners of their own identities. Data *Owners* delegate *Proxy* to process their massive data before storing them in multiple clouds. *Proxy* of abundant computation and bandwidth resource, helps data owners to generate proxy data tags and upload them to clouds, with data owners' special warrants. Multiple *Clouds* maintain powerful storage and computation resources to provide storage service for data owners. The *TPA* is a trusted third party auditor to offer the batch data integrity verification on multiple clouds for the data owners.

### 3.2 System components of an ID-BPAPP scheme

- Setup $(1^k) \to (params, mpk, msk)$ is initialized by PKG with security parameter $k$ as input. It outputs public parameters $params$, master key pairs $(mpk, msk)$.
- Extract $(params, msk, ID_i) \to sk_i$ is executed by PKG with as input parameters $params$, master private key $msk$ and data owner's identity $ID_i$, and outputs the private key $sk_i$ for this owner. It also extracts private key $sk_p$ for proxy of $ID_p$.
- ProxyKeyGen $(params, ID_i, sk_i, ID_p, sk_p) \to u_{pi}$ is run by proxy with interaction of data owner. With input of parameter $params$ and its private key $sk_i$, data owner of $ID_i$ generates warrant and corresponding signature to send to proxy. Then the proxy of $ID_p$

outputs the proxy secret key $u_{pi}$ with its private key $sk_p$.

- TagGen $(params, ID_i, sk_p, u_{pi}, mpk, \{\tilde{F}_{ijk}\}) \rightarrow \{\sigma_{ijk}\}$ is run by proxy. It takes as input public parameters $params$, owner's identity $ID_i$, its individual private key $sk_p$, corresponding proxy secret key $u_{pi}$, master public key $mpk$ and owner's data blocks $\{\tilde{F}_{ijk}\}$ to be outsourced on the corresponding clouds. Then the proxy tags $\{\sigma_{ijk}\}$ of above blocks could be generated.

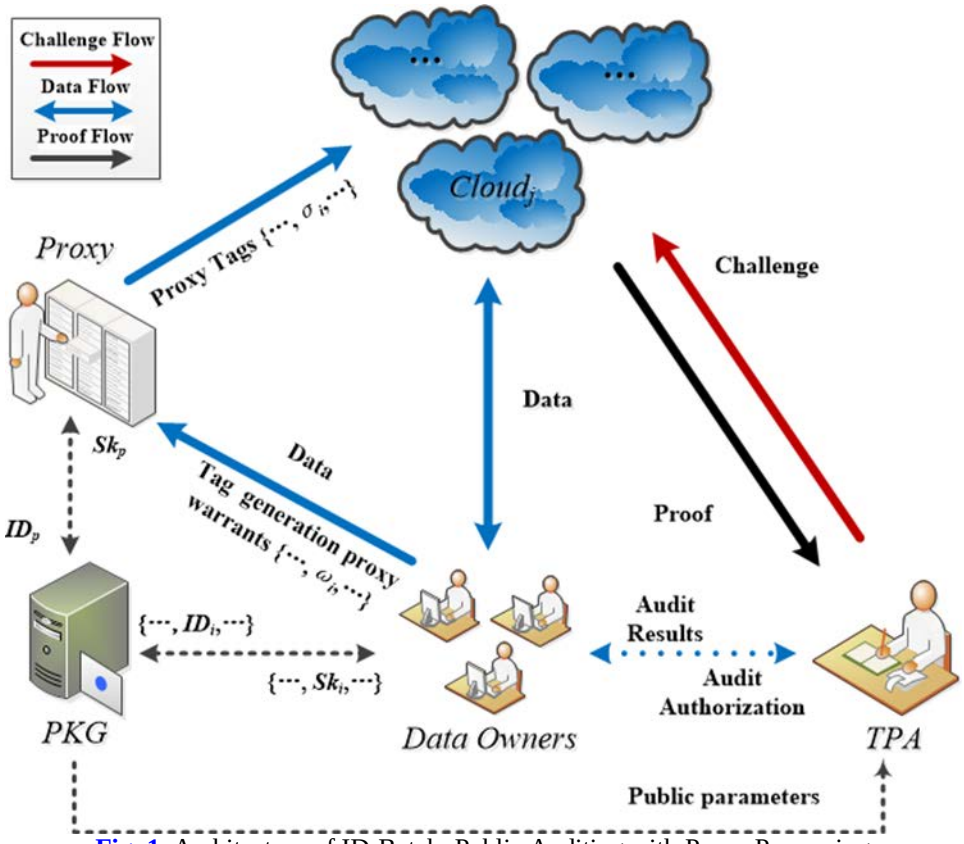


**Fig. 1.** Architecture of ID-Batch Public Auditing with Proxy Processing

- Challenge $(\{(i,j,k)\}) \rightarrow (chal, \{chal_j\})$ is executed by third party auditor (TPA). It takes as input data index set $\{(i,j,k)\}$ and randomly selects some indexes as the challenge token $chal$ for one instance. According to the specified indexes $\{j\}$, challenge token $chal$ is further divided into a set of tokens $\{chal_j\}$ and only forward $chal_j$ to the corresponding $j$-th cloud $CS_j$.
- ProofGen $(params, chal_j, \{ID_i\}, \{\sigma_{ijk}\}, \{\tilde{F}_{ijk}\}) \rightarrow P_j$ is run by cloud $CS_j$. It takes as input parameters $params$, challenge token received $chal_j$, the specified set of data owners' identities $\{ID_i\}$, the set of tags $\{\sigma_{ijk}\}$, and the blocks $\{\tilde{F}_{ijk}\}$. Then the proof $P_j$ is generated for challenge token $chal_j$, and is sent back to TPA.
- Verify $(params, chal, \{ID_i\}, \{P_j\}, mpk) \rightarrow \{0,1\}$ is executed by TPA. It takes as input public parameters $params$, challenge token $chal$, specified set of data owners' identities $\{ID_i\}$, set of proofs $\{P_j\}$ from all challenged clouds, and the master public key $mpk$. 1 will be output if the proofs are valid, otherwise 0 is output.

## 3.3 Security Model

In an ID-BPAPP scheme, we assume PKG is trusted to execute the scheme, and proxy honestly generates tags but may have management fault of data before tag generation. Meanwhile, original data owners might generate data tag themselves without the delegated proxy. Clouds could also hide data accident for the sake of reputation and saving cost, and TPA is trusted but curious about the data content. A secure ID-BPAPP scheme should satisfy three properties:

1) Proxy-protection: Data owners themselves are not able to masquerade as proxy to generate tags. Only proxy with authorization warrant could generate proxy tags.

2) Unforgeability: It is infeasible to fabricate valid data storage proofs to pass the auditing of TPA if any cloud data is modified or deleted.

3) Privacy-preserving: Real data content will not be revealed during the process of auditing.

According to the security requirements, we review the three formal definitions as follows:

- **Definition 1** (**Proxy-Protection**): The scheme is proxy-protected, if any probabilistic polynomial data owner wins the proxy Tag-Forge game below in probabilistic polynomial time (PPT), with negligible probability.
  - *Setup*: The challenger $\boldsymbol{C}_1$ playing in the role of PKG and TPA, first generates master public/private key pairs and system parameters. It runs Extract to generate private key $sk_p$ for proxy of identity $ID_p$ and keeps it secret. Those public and not secret parameters could be sent to the adversary $\boldsymbol{A}_1$, who acts as data owner.
  - *Queries*: Besides all hash functions, $\boldsymbol{A}_1$ could adaptively query Extract for private key $sk_i$ of identity $ID_i$ except $ID_p$ of proxy. Denote index set of identities as $S_1$, $(p \notin S_1)$. It could query proxy tag secret keys $u_{p'i}$ for the pair $(ID_{p'}, ID_i)$ except for pairs having proxy $ID_p$. Denote index set of pairs as $S_1'$ $((p,i) \notin S_1')$. Upon data block $\tilde{F}_{ijk}$, $\boldsymbol{A}_1$ could also adaptively query proxy tag $\sigma_{p'ijk}$ for this identity pair except having $ID_p$ as proxy. Let us denote tuples set of corresponding indexes and data as $S_1''$, $\left(p,i,j,k,\tilde{F}_{ijk}\right) \notin S_1''$.
  - *Output*: $\boldsymbol{A}_1$ wins the game if it creates a valid proxy tag $\sigma_{i^*j^*k^*}$ for data block $\tilde{F}_{i^*j^*k^*}$ by itself, for which it has neither extracted private key nor proxy tag secret key for proxy $ID_p$, i.e., where $p \notin S_1$, $(p,i^*) \notin S_1'$, $\left(p,i^*,j^*,k^*,\tilde{F}_{i^*j^*k^*}\right) \notin S_1''$.
- **Definition 2** (**Unforgeability**): The scheme is unforgeable if any PPT clouds win the Proofs-Forge game below, with negligible probability.
  - *Setup*: The challenger $\boldsymbol{C}_2$ playing in the role of PKG and TPA, first generates master public/private keys pair and system parameters. It runs Extract to generate private key $sk_p$ for proxy of identity $ID_p$ and keeps it secret. Those public and not secret parameters could be sent to the adversary $\boldsymbol{A}_2$, who acts as clouds.
  - *First phase queries*: Besides all hash functions, $\boldsymbol{A}_2$ could adaptively query Extract for private key $sk_i$ of identity $ID_i$ except $ID_p$ of proxy. Denote index set of identities as $S_2$, $(p \notin S_2)$. It could query proxy tag secret keys $u_{p'i}$ for the pair $(ID_{p'}, ID_i)$ except for pairs having proxy $ID_p$. Denote index set of pairs as $S_2'$ $((p,i) \notin S_2')$. Upon data block $\tilde{F}_{ijk}$, $\boldsymbol{A}_2$ could also adaptively query proxy tag $\sigma_{p'ijk}$ for this identity pair except having $ID_p$ as proxy. Let us denote tuples set of corresponding indexes and data as $S_2''$, $\left(p,i,j,k,\tilde{F}_{ijk}\right) \notin S_2''$.
  - *Challenge*: $\boldsymbol{C}_2$ generates challenge set $chal$ with ordered number collection $\{c_{i^*j^*}\}$ to specify every block $\tilde{F}_{i^*j^*k^*}$on the $j^*$th cloud for owner of $ID_{i^*}$, where $\{(p,i^*,j^*,k_n^*)|1 \leq$

$n \le c_{i^*j^*}\}$, $i^* \ne p$, $(p, i^*) \notin S_2'$, $\left(p, i^*, j^*, k_n^*, \tilde{F}_{i^*j^*k_n^*}\right) \notin S_2''$. $chal$ will be sent to $\boldsymbol{A}_2$.

- *Second phase queries* : similar to *First phase queries*, denote index set of identities for Extract private key queries as $S_3$, index set of identity pairs for proxy tag secret key queries as $S_3'$, tuple set of index and data for proxy tags queries as $S_3''$. We require that $p \notin S_2 \cup S_3$, $(p, i) \notin S_2' \cup S_3'$ and $\left(p, i, j, k, \tilde{F}_{ijk}\right) \notin S_2'' \cup S_3''$
- *Output*: $\boldsymbol{A}_2$ wins the game if it fabricates valid proofs $\{P_j\}$ for the same challenge set $chal$ on the specified set of data blocks.

- **Definition 3** (**Privacy-Preserving**): The ID-BPAPP scheme is privacy-preserving against TPA, if any PPT time TPA could extract any original block of data owners in the "challenge-proof-verify" integrity auditing interactions with clouds, with negligible probability. In this definition, we require that curious TPA is not allowed to recover data blocks even if it is able to fulfill task of auditing integrity of cloud data.

## 4. Revisiting Yu et al.'s construction of an of ID-BPAPP scheme

In this section, we will revisit the Yu et al.'s construction of an ID-BPAPP scheme with concrete designs of seven algorithms in [32].

- Setup: PKG uses this algorithm to generate a bilinear map $e: G_1 \times G_1 \to G_2$ with two groups $G_1$ and $G_2$ of the same order $q > 2^k$, where $g$ is the generator of $G_1$ and $k$ is security parameter. It also selects three cryptographic hash functions, $H_1: \{0,1\}^* \to G_1$, $H_2: \{0,1\}^* \to Z_q$, $H_3: Z_q \times \{0,1\}^* \to Z_q$, a pseudo random permutation $\pi: Z_q \times \{1, \cdots, N\} \to \{1, \cdots, N\}$ and a pseudo random function $f: Z_q \times \{1, \cdots, N\} \to Z_q$. It picks random $x \in Z_q$ as master private key $msk$ and computes $g^x$ as master public key $mpk$.The global parameters are $(e, G_1, G_2, g, mpk, H_1, H_2, H_3, \pi, f)$.
- Extract: Given identity $ID_i$, PKG extracts the identity-based private key as $sk_i = H_1(ID_i)^x$ and returns to the data owner. For the proxy of identity $ID_p$, the private key is extracted as $sk_p = H_1\left(ID_p\right)^x$.
- ProxyKeyGen: Data owner of $ID_i$ picks up random $r_i \in Z_q$ and creates its proxy warrant $\omega_i$ with signature $U_i = sk_i^{r_i H_2(\omega_i || R_i)}$, $\xi_i = g^{r_i}$, where $R_i = H_1(ID_i)^{r_i}$. $(\omega_i, U_i, R_i, \xi_i)$ are sent to proxy, clouds and TPA. Upon the warrant $\omega_i$, TPA and proxy could verify it with signature as $e(R_i, g) = e(H_1(ID_i), \xi_i)$, $e(U_i, g) = e(R_i^{H_2(\omega_i || R_i)}, mpk)$, and notify the data owner if any of equations does not hold. Proxy generates the proxy secret key as $u_{pi} = U_i \cdot sk_p^{r_{pi}} = H_1(ID_i)^{x \cdot r_i H_2(\omega_i || R_i)} \cdot H_1\left(ID_p\right)^{x \cdot r_{pi}}$ by selecting up random $r_{pi} \in Z_q$. It also computes $R_{pi} = H_1\left(ID_p\right)^{r_{pi}}$, which is not secret and sent to the TPA for future verification.
- TagGen: Data owner of $ID_i$ first divides original data $\tilde{F}_i$ into blocks $\{\tilde{F}_{ijk}\}$, and computes each $F_{ijk} = \tilde{F}_{ijk} + H_2(\tilde{F}_{ijk})$. Data blocks $\{\tilde{F}_{ijk}\}$, are outsourced to corresponding clouds while masked $\{F_{ijk}\}$ are sent to the proxy. Then the proxy generates proxy tag for each data block as

$$\sigma_{ijk} = sk_p^{H_3(i||j||k, name_{ijk}||time_{ijk})} \cdot u_{pi}^{F_{ijk}} \tag{1}$$

where $name_{ijk}$ is the name of block $\tilde{F}_{ijk}$, and $time_{ijk}$ is the time stamp when proxy generates the tag. All the tags $\{\sigma_{ijk}\}$ and the not secret $R_{pi}$ will be transferred to corresponding clouds, which will not accept them and inform the owner unless the warrant

$\omega_i$ and the proxy tag $\sigma_{ijk}$ could be verified by having the following equations hold as

$$e(R_i, g) = e(H_1(ID_i), \xi_i), e(U_i, g) = e\left(R_i^{H_2(\omega_i||R_i)}, mpk\right)$$
$$e(\sigma_{ijk}, g) = e\left(H_1(ID_p)^{H_3(i||j||k, name_{ijk}||time_{ijk})} \cdot \left(R_i{}^{H_2(\omega_i||R_i)} \cdot R_{pi}\right)^{F_{ijk}}, mpk\right) \quad (2)$$

- Challenge: For data owner of $ID_i$ on $j$-th cloud's data, TPA picks up number of challenged blocks $c_{ij} < N$ and random $v_{ij,1}$ and $v_{ij,2} \in Z_q$. Denote $O_j$ as index set of identities for owners having data on $j$-th cloud. It generates the challenge token $chal_j = \{(c_{ij}, v_{ij,1}, v_{ij,2})\}_{i \in O_j}$, and sends it to $j$-th cloud.
- ProofGen: According to the challenge token $chal_j = \{(c_{ij}, v_{ij,1}, v_{ij,2})\}_{i \in O_j}$, the $j$-th challenged cloud first generates index set $\delta_{ij}$ of challenged blocks for the data owner of $ID_i$ where each index $k = \pi_{v_{ij,1}}(a_{ij})(1 \le a_{ij} \le c_{ij})$ according to the individual challenged number $c_{ij}$ (e.g., assuming $c_{ij} = 4$, $a_{ij} \in [1,4]$ could be permutated into 4 challenged blocks indexes $k \in \{234, 8, 364, 25\}$ with $\pi_{v_{ij,1}}(\cdot)$) and then the corresponding co-efficient $h_{ijk} = f_{v_{ij,2}}(i, j, k) \in Z_q$.The proof of storage $P_j$ includes aggregate tag $T_j'$ and masked data proof $\{F_{ij}'\}$ for its data owners of identities with index set $O_j$:

$$T_j' = \prod_{i \in O_j} \prod_{k \in \delta_{ij}} \sigma_{ijk}^{h_{ijk}}, F_{ij}' = \sum_{k \in \delta_{ij}} h_{ijk} \cdot F_{ijk}$$

Where $F_{ijk} = \tilde{F}_{ijk} + H_2(\tilde{F}_{ijk})$. $P_j = (T_j', \{F_{ij}'\}_{i \in O_j})$ will be sent to the TPA.

- Verify: After receiving all the proofs $\{P_j\}$ from challenged clouds, the TPA denotes $O = \bigcup_{j \in J} O_j$ as identity index set of all the challenged owners according to challenge tokens $\{chal_j\} = \left\{\{(c_{ij}, v_{ij,1}, v_{ij,2})\}_{i \in O_j}\right\}_{j \in J}$, and computes index set of all challenged blocks by $\{k\} = \{\pi_{v_{ij,1}}(a_{ij}) | 1 \le a_{ij} \le c_{ij}\}$ and co-efficient set $\{h_{ijk}\} = \{f_{v_{ij,2}}(i, j, k)\}$, as in ProofGen. With all valid set of warrants $\{\omega_i\}$ and corresponding signatures $\{(U_i, R_i, \xi_i)\}$ from data owners, together with blocks' names and time stamps $\{(name_{ijk}, time_{ijk})\}$, TPA is able to audit data integrity as :

$$e\left(\prod_{j \in J} T_j', g\right) = e\left(\prod_{i \in O} \left(R_i{}^{H_2(\omega_i||R_i)} \cdot R_{pi}\right)^{\sum_{j \in J} F_{ij}'} \cdot H_1(ID_p)^{\sum_{i \in O} \sum_{j \in J} \sum_{k \in \delta_{ij}} h_{ijk} \cdot H_3(i||j||k, name_{ijk}||time_{ijk})}, mpk\right) \quad (3)$$

It will outputs 1 (valid) if the above equation holds and 0 (valid) otherwise.

## 5. On the security of Yu et al.'s construction of an ID-BPAPP scheme

With security analysis, Yu et al.'s construction of an ID-BPAPP scheme in [32] should satisfy security properties for data proof unforgeability and tag generation proxy-protection. However, this scheme may suffer from two security issues, as the analysis in the following.

### 5.1 First issue: generating valid proof without original data

In Yu et al.'s ID-BPAPP scheme, the TPA utilizes masked data proof to evaluate the original

data integrity on the cloud. This design indeed makes original data content invisible to TPA to allow privacy-preserving auditing, but also leaves the room for malicious cloud to launch data attack as follows.

In the Proof, for data part $\{F'_{ij}\}_{i\in O_j}$ of proof $P_j$, honest cloud takes original data $\tilde{F}_{ijk}$ as input to get masked data $F_{ijk} = \tilde{F}_{ijk} + H_2(\tilde{F}_{ijk})$, and do the combination with the fresh challenge co-efficient $\{h_{ijk}\}$, as $F'_{ij} = \sum_{k\in\delta_{ij}} h_{ijk} \cdot F_{ijk}$. Obviously, the fresh challenge co-efficient is combined with masked data, rather than directly with the original data. Therefore, after generating tag part $T'_j$ from correct tags, malicious cloud is able to generate valid integrity proof $P_j = (T'_j, \{F'_{ij}\}_{i\in O_j})$ without having to store original data $\tilde{F}_{ijk}$, just combing pre-computed masked data $F_{ijk}$ and challenge co-efficient. In this way, malicious cloud could successfully pass TPA's integrity auditing, when original data $\tilde{F}_{ijk}$ is modified as $\tilde{F}^*_{ijk}$ or even gets deleted.

## 5.2 Second issue: recovering private key of proxy and proxy tag secret key

With proxy-protection property, only proxy with authorization could generate the data tags for integrity auditing. As analysis below, we could find that it is feasible to recover proxy's private key and thus impersonate proxy to generate data tag, for those who could access the data and tags.

In TagGen, for data $\tilde{F}_{ijk}$, tag $\sigma_{ijk} = sk_p^{H_3(i||j||k, name_{ijk}||time_{ijk})} \cdot u_{pi}^{F_{ijk}}$ is generated by proxy, with its individual private key $sk_p$ and proxy tag secret key $u_{pi}$, and then uploads tag on the cloud. Afterwards, malicious cloud or curious data owner of $ID_i$, retrieve two arbitrary data blocks $(\tilde{F}_{ijk_1}, \tilde{F}_{ijk_2})$ with corresponding tags $(\sigma_{ijk_1}, \sigma_{ijk_2})$, and do the computation:

$$sk_p = \left(\sigma_{ijk_1}^{\frac{1}{F_{ijk_1}}} \Big/ \sigma_{ijk_2}^{\frac{1}{F_{ijk_2}}}\right)^{\frac{F_{ijk_1}F_{ijk_2}}{H_3(i||j||k_1, name_{ijk_1}||time_{ijk_1})F_{ijk_2} - H_3(i||j||k_2, name_{ijk_2}||time_{ijk_2})F_{ijk_1}}}$$

$$u_{pi} = \left(\left(\sigma_{ijk_1}^{\frac{1}{H_3(i||j||k_1, name_{ijk_1}||time_{ijk_1})}}\right) \Big/ \left(\sigma_{ijk_2}^{\frac{1}{H_3(i||j||k_2, name_{ijk_2}||time_{ijk_2})}}\right)\right)^{EX}$$

$$EX = \frac{H_3(i||j||k_1, name_{ijk_1}||time_{ijk_1})H_3(i||j||k_2, name_{ijk_2}||time_{ijk_2})}{F_{ijk_1}H_3(i||j||k_2, name_{ijk_2}||time_{ijk_2}) - F_{ijk_2}H_3(i||j||k_1, name_{ijk_1}||time_{ijk_1})}$$

Where masked data $(F_{ijk_1}, F_{ijk_2}) = (\tilde{F}_{ijk_1} + H_2(\tilde{F}_{ijk_1}), \tilde{F}_{ijk_2} + H_2(\tilde{F}_{ijk_2}))$.

With the recovered proxy private key $sk_p$ and proxy tag secret key $u_{pi}$, three kinds of security problems could happen. First, for new block $\tilde{F}_{ijk_3}$, data owner could generate the proxy tag as $\sigma_{ijk_3} = sk_p^{H_3(i||j||k_3, name_{ijk_3}||time_{ijk_3})} \cdot u_{pi}^{F_{ijk_3}}$ without proxy's processing, which will keep equations (2) (3) hold and finally help data to pass the TPA auditing. Thus proxy-protection security property cannot be guaranteed. Second, if the original block is modified to $\tilde{F}^*_{ijk_3}$, the malicious cloud could generate valid tag as $\sigma^*_{ijk_3} = sk_p^{H_3(i||j||k_3, name_{ijk_3}||time_{ijk_3})} \cdot u_{pi}^{F^*_{ijk_3}}$, where $F^*_{ijk_3} = \tilde{F}^*_{ijk_3} + H_2(\tilde{F}^*_{ijk_3})$, without awareness of data owner and proxy. Certainly the two tags will also keep equations (2) (3) hold and help to generate valid integrity proof, but unforgeability property cannot be

guaranteed for falling to check data modification. This will leads to the serious data loss situation: cloud could keep only one block and delete rest of data to pretend that all the blocks are equal in the value, simply computing valid proxy tags with all their indexes and information. Third, the digital property belonging to proxy, will be in the great risk of illegal access, due to the recovered proxy individual private key by other entities.

## 6. Our improved construction of an ID-BPAPP scheme

- Setup: PKG uses this algorithm to generate a bilinear map $e: G_1 \times G_1 \to G_2$ with two groups $G_1$ and $G_2$ of the same order $q > 2^k$, where $g$ is the generator of $G_1$ and $k$ is security parameter. It also selects four cryptographic hash functions, $H_1: \{0,1\}^* \to G_1$, $H_2: \{0,1\}^* \to Z_q$, $H_3: Z_q \times \{0,1\}^* \to Z_q$, $H_4: Z_q \times \{0,1\}^* \to G_1$, a pseudo random permutation $\pi: Z_q \times \{1,\cdots,N\} \to \{1,\cdots,N\}$ and a pseudo random function $f: Z_q \times \{1,\cdots,N\} \to Z_q$. It picks random $x \in Z_q$ as master private key $msk$ and computes $g^x$ as master public key $mpk$.The global parameters are $(e, G_1, G_2, g, mpk, H_1, H_2, H_3, H_4, \pi, f)$.
- Extract: Given identity $ID_i$, PKG extracts the identity-based private key as $sk_i = H_1(ID_i)^x$ and returns to the data owner. For the proxy of identity $ID_p$, the private key is extracted as $sk_p = H_1(ID_p)^x$.
- ProxyKeyGen: For data owner of $ID_i$, it picks up random $r_i \in Z_q$ and creates its proxy warrant $\omega_i$ with its signature $U_i = sk_i^{r_i H_2(\omega_i||R_i)}$, $\xi_i = g^{r_i}$, where $R_i = H_1(ID_i)^{r_i}$. $(\omega_i, U_i, R_i, \xi_i)$ are sent to proxy, clouds and TPA. Upon the warrant $\omega_i$, TPA and proxy could verify it with signature as $e(R_i, g) = e(H_1(ID_i), \xi_i)$, $e(U_i, g) = e(R_i^{H_2(\omega_i||R_i)}, mpk)$, and notify the data owner if any of equations does not hold. Proxy generates the proxy secret key as $u_{pi} = U_i \cdot sk_p^{r_{pi}} = H_1(ID_i)^{x \cdot r_i H_2(\omega_i||R_i)} \cdot H_1(ID_p)^{x \cdot r_{pi}}$ by picking up random $r_{pi} \in Z_q$. It also computes $R_{pi} = H_1(ID_p)^{r_{pi}}$, which is not secret and sent to the TPA for future verification.
- TagGen: Data owner of $ID_i$ first divides original data $\tilde{F}_i$ into blocks $\{F_{ijk}\}$, where $F_{ijk} \in Z_q$. They are outsourced to corresponding clouds and sent to the proxy. For each data block, proxy generates tag $\sigma_{ijk} = (T_{ijk}, S)$ as

$$\begin{aligned} T_{ijk} &= (sk_p \cdot u_{pi})^{H_3(i||j||k, name_i||time_{ijk}) + F_{ijk}} \cdot H_4(i||j||k, name_i||time_{ijk}||S)^{\eta} \\ S &= g^{\eta} \end{aligned} \tag{4}$$

where $name_i$ is the name of file $\tilde{F}_i$, and $time_{ijk}$ is the time stamp when proxy generates the tag, $\eta \in Z_q$. All the tags $\{\sigma_{ijk}\}$ and the not secret $R_{pi}$ will be transferred to corresponding clouds, which will not accept them and inform the owner unless the warrant $\omega_i$ and the proxy tag $\sigma_{ijk}$ could be verified by having the following equations hold as

$$\begin{aligned} &e(R_i, g) = e(H_1(ID_i), \xi_i),\ e(U_i, g) = e(R_i^{H_2(\omega_i||R_i)}, mpk) \\ &e(T_{ijk}, g) = e\left(\left(H_1(ID_p) \cdot \left(R_i{}^{H_2(\omega_i||R_i)} \cdot R_{pi}\right)\right)^{H_3(i||j||k, name_i||time_{ijk}) + F_{ijk}}, mpk\right) \\ &\qquad \cdot e(H_4(i||j||k, name_i||time_{ijk}||S), S) \end{aligned} \tag{5}$$

- Challenge: For data owner of $ID_i$ on $j$-th cloud's data, TPA picks up number of challenged blocks $c_{ij} < N$, random $v_{ij,1}, v_{ij,2} \in Z_q$ and masking element $M = mpk^w$ for $w \in Z_q$. Denote $O_j$ as the index set of identities for owners having data on cloud $CS_j$. It generates the challenge token $chal_j = \left(\{(c_{ij}, v_{ij,1}, v_{ij,2})\}_{i \in O_j}, M\right)$, and sends it to the cloud.

- ProofGen: According to the challenge token $chal_j = \left(\{(c_{ij}, v_{ij,1}, v_{ij,2})\}_{i\in O_j}, M\right)$, the $j$-th challenged cloud first generates index set $\delta_{ij}$ of challenged blocks for the data owner of $ID_i$ where each index $k = \pi_{v_{ij,1}}(a_{ij})(1 \le a_{ij} \le c_{ij})$ according to the individual challenged number $c_{ij}$ (e.g., assuming $c_{ij} = 4$, $a_{ij} \in [1,4]$ could be permutated into 4 challenged blocks indexes $k \in \{234, 8, 364, 25\}$ with $\pi_{v_{ij,1}}(\cdot)$ ) and then the corresponding co-efficient $h_{ijk} = f_{v_{ij,2}}(i,j,k) \in Z_q$. The proof of storage $P_j$ includes aggregate tag $T_j', S'$ and masked data proof $M_j'$ for its data owners of identities with indexes in $O_j$:

$$T_j' = \prod_{i\in O_j}\prod_{k\in\delta_{ij}} T_{ijk}^{h_{ijk}}, S' = S, M_j' = e\left(\prod_{i\in O_j}\left(H_1(ID_p)\cdot\left(R_i{}^{H_2(\omega_i||R_i)}\cdot R_{pi}\right)\right)^{F_{ij}'}, M\right)$$

Where $F_{ij}' = \sum_{k\in\delta_{ij}} h_{ijk}\cdot F_{ijk}$. Proof $P_j = (T_j', S', M_j')$ will be sent to the TPA. (Cloud could send the proof to TPA in the secure channel or prevent modification with identity-based signature technology).

- Verify: After receiving all the proofs $\{P_j\}$ from challenged clouds, the TPA denotes $O = \bigcup_{j\in J} O_j$ as identity index set of all the challenged owners according to challenge tokens $\{chal_j\} = \left\{\left(\{(c_{ij}, v_{ij,1}, v_{ij,2})\}_{i\in O_j}, M\right)\right\}_{j\in J}$ where $M = mpk^w$, and computes index set of all challenged blocks by $\{k\} = \{\pi_{v_{ij,1}}(a_{ij}) | 1 \le a_{ij} \le c_{ij}\}$ and co-efficient set $\{h_{ijk}\} = \{f_{v_{ij,2}}(i,j,k)\}$, as in ProofGen. With all valid set of warrants $\{\omega_i\}$ and corresponding signatures $\{(U_i, R_i, \xi_i)\}$ from data owners, together with files' names and blocks' time stamps $\{(name_i, time_{ijk})\}$, TPA is able to audit data integrity as :

$$e\left(\prod_{j\in J} T_j', g^w\right) = e\left(\prod_{i\in O}\left(H_1(ID_p)\cdot\left(R_i{}^{H_2(\omega_i||R_i)}\cdot R_{pi}\right)\right)^{L_i}, M\right)$$
$$\cdot e\left(\prod_{i\in O}\prod_{j\in J}\prod_{k\in\delta_{ij}}\left(H_4(i||j||k, name_i||time_{ijk}||S')\right)^{h_{ijk}}, S'^w\right)\cdot\prod_{j\in J} M_j' \quad (6)$$

where $L_i = \sum_{j\in J}\sum_{k\in\delta_{ij}} h_{ijk}\cdot H_3(i||j||k, name_i||time_{ijk})$.

It will outputs 1 (valid) if the above equation holds and 0 (valid) otherwise.

**Correctness:**

$$LHS = e\left(\prod_{j\in J}\prod_{i\in O_j}\prod_{k\in\delta_{ij}}\left(H_1(ID_p)\right.\right.$$
$$\left.\cdot\left(R_i{}^{H_2(\omega_i||R_i)}\cdot R_{pi}\right)\right)^{(H_3(i||j||k, name_i||time_{ijk})+F_{ijk})h_{ijk}}, (g^a)^w\Bigg)$$
$$\cdot e\left(\prod_{j\in J}\prod_{i\in O_j}\prod_{k\in\delta_{ij}} H_4(i||j||k, name_i||time_{ijk}||S)^{h_{ijk}}, (g^\eta)^w\right)$$

$$= e\left(\prod_{i\in O}\left(H_1(ID_p)\cdot\left(R_i{}^{H_2(\omega_i||R_i)}\cdot R_{pi}\right)\right)^{\sum_{j\in J}\sum_{k\in\delta_{ij}} h_{ijk}\cdot H_3(i||j||k,name_i||time_{ijk})}, M\right)$$
$$\cdot e\left(\prod_{i\in O}\prod_{j\in J}\prod_{k\in\delta_{ij}}\left(H_4(i||j||k,name_i||time_{ijk}||S')\right)^{h_{ijk}}, S'^{w}\right)$$
$$\cdot\prod_{j\in J} e\left(\left(H_1(ID_p)\cdot\left(R_i{}^{H_2(\omega_i||R_i)}\cdot R_{pi}\right)\right)^{\sum_{i\in O_j}\sum_{k\in\delta_{ij}} h_{ijk}\cdot F_{ijk}}, M\right) = RHS$$

## 6.1Security analysis of improved scheme

Based on the system model of an ID-BPAPP scheme (Subsection 3.1) and corresponding system components (Subsection 3.2) and security model (Subsection 3.3), in this section, we prove security of our improved scheme. Compared with [27]'s security analysis, we also utilize Coron [31]'s random oracle model to define the interactions between adversary of our scheme and challenger, but with refined oracles for hash and tag queries. To prevent security flaws in [32], corresponding security reduction methods are also re-designed.

There are $|O|$ number of owners, $c_{i^*j^*}$ number of challenged blocks on corresponding cloud for specified owner, and $c^* = \left(\sum_{i^*\in O, j^*\in J} c_{i^*j^*}\right)^{-1}$, $\widehat{N}$ number of selected identities. For oracle, $q_H$ hash, $q_E$ Extract, $q_P$ ProxyKeyGen, and $q_T$ TagGen queries are made. We assume both inversion and exponentiation operations on $G_1$ require $t_{G_1}$, so it is with $t_{G_2}$, and pairing takes $t_e$. $\hat{e}$ is the natural logarithm.

Our security analysis below shows that CDH problem will be solved if breaking our scheme through forging valid proxy tag, BDH problem will be solved if fabricating storage proof without rejection, and DL problem will be solved if breaking our scheme through retrieving data value during auditing, with non-negligible probability under polynomial time.

**Theorem 1** (Proxy-Protection) If there exists Probabilistic Polynomial Time (PPT) $(t_1, \epsilon_1)$-adversary $\boldsymbol{A}_1$ who could generate valid proxy tag without proxy individual private key in our Sec-ID-BPAPP, then our scheme is proxy-protective when challenger $\boldsymbol{C}_1$ could solve CDH problem with non-negligibility $\epsilon_1\left(\widehat{N}-1\right)^{q_E}/(\hat{e}\,\widehat{N}^{q_E+q_P}(q_E+q_T+1)\;)$ within PPT time $t_1 + t_{G_1}\cdot(q_H+q_E+q_P+4q_T+5)$.

**Proof**: There are $\widehat{N}$ number of selected identities $\{ID_i\}_{i\in O}$ having the proxy $ID_p$. The original file $\{\tilde{F}_i\}_{i\in O}$ will be split into blocks $\{F_{ijk}\}_{i\in O, j\in J, k\in\delta_{ij}}$ before being outsourced on clouds $\{CS_j\}_{j\in J}$.

· Setup: $\boldsymbol{C}_1$ plays in the role of PKG to choose random $a\in Z_q$, then the master private /public keys pair $(msk, mpk) = (a, g^a)$, upon generator $g\in G_1$. It also picks random $b\in Z_q$. CDH instance is $g^a, g^b\in G_1$ to compute $g^{ab}$. Although $\boldsymbol{A}_1$ is not allowed to query the target proxy tag secret keys $u_{pi}$, the $R_{pi}$ could be accessed as $H_1(ID_p)^{r_{pi}}$ by $\boldsymbol{C}_1$ picking up $r_{pi}\in Z_q$.

$\boldsymbol{C}_1$ answers query by maintaining input and output list for every oracle. Especially, output is retrieved from existing record of same input, otherwise is generated as follows and $\boldsymbol{C}_1$ builds new record in the corresponding list.

· Hash function Oracle: $H_2$ and $H_3$ work as normal hash functions.

$H_1$-oracle: $\boldsymbol{C}_1$ answers with $g^{y_i}$ for $y_i\in Z_q$ if $i\neq p$, and $y_i = b$ for $i = p$.

$H_4$-oracle: $\boldsymbol{C}_1$ answers with $g^{z_{ijk}}$ with $z_{ijk}\in Z_q$.

- Extract-oracle: $\boldsymbol{C}_1$ answers $sk_i = (g^a)^{y_i}$ from $H_1$, if $i \neq p$; else aborts. Denote indexes set of identities extracting private key as $S_1 (p \notin S_1)$ .
- ProxyKeyGen-oracle: $\boldsymbol{C}_1$ answers $u_{p'i} = U_i \cdot (g^a)^{y_{p'} \cdot r_{p'i}}$ from $H_1$ and $r_{p'i} \in Z_q$, if $i \neq p$; else aborts. Denote index pair set of identities as $S_1'((p,i) \notin S_1')$.
- Tag-oracle: $\boldsymbol{C}_1$ answers $T_{ijk} = \left((g^a)^{y_{p'}} \cdot u_{p'i}\right)^{H_3, ijk + F_{ijk}} \cdot S_{ijk}^{z_{ijk}}$ with $S_{ijk} \in G_1$ from $H_1, H_4$, and ProxyKeyGen, if $p' \neq p$. Certainly this tag is valid to pass equation (5) and computational indistinguishable from real one for $\boldsymbol{A}_1$'s view; else aborts. Denote query input as set $S_1''((p,i,j,k,F_{ijk}) \notin S_1'')$.

**Forgery Output**: Finally, $\boldsymbol{A}_1$ itself outputs a valid tag $\sigma_{i^*j^*k^*} = (T_{i^*j^*k^*}, S')$ for data block $F_{i^*j^*k^*}$generated by proxy $ID_p$ with warrant $\omega_{i^*}$ and its signature $(U_{i^*}, R_{i^*}, \xi_{i^*})$. $\boldsymbol{C}_1$ looks up lists of all oracles. It will not abort and terminate only when none of corresponding records exists, i.e., requiring $ID_{i^*} \neq ID_p$, $(p, i^*) \notin S_1'$, $(p, i^*, j^*, k^*, F_{i^*j^*k^*}) \notin S_1''$. If game could proceed, $\boldsymbol{C}_1$ keeps on checking all hash function oracles and makes queries itself if there is no relative record in their lists. $R_{pi^*} = H_1(ID_p)^{r_{pi^*}}$ in Setup and $U_{i^*} = (g^a)^{y_{i^*} r_{i^*} H_2(\omega_{i^*} || R_{i^*})}$ for validity of warrant $\omega_{i^*}$.

Since $\sigma_{i^*j^*k^*} = (T_{i^*j^*k^*}, S')$ satisfies equation (5) as valid tag, with corresponding records of oracles and properties of bilinear mapping:

$$
\begin{aligned}
e(T_{i^*j^*k^*}, g) = e\Bigg(&\Big(H_1(ID_p) \\
&\cdot \left(R_{i^*}^{H_2(\omega_{i^*} || R_{i^*})} \cdot R_{pi^*}\right)\Big)^{H_3(i^* || j^* || k^*, name_{i^*} || time_{i^*j^*k^*}) + F_{i^*j^*k^*}}, g^a\Bigg) \\
&\cdot e(g^{z_{i^*j^*k^*}}, S') \\
= e\Big(&\left(g^{ab} \cdot \left(U_{i^*} \cdot g^{abr_{pi^*}}\right)\right)^{H_3(i^* || j^* || k^*, name_{i^*} || time_{i^*j^*k^*}) + F_{i^*j^*k^*}} \cdot S'^{z_{i^*j^*k^*}}, g\Big)
\end{aligned}
$$

we will have a solution of CDH problem after simplification

$$
g^{ab} = \left(T_{i^*j^*k^*} \cdot S'^{-z_{i^*j^*k^*}} \cdot U_{i^*}^{-H_3(i^* || j^* || k^*, name_{i^*} || time_{i^*j^*k^*}) - F_{i^*j^*k^*}}\right)^{\frac{1}{W}}
$$

Where $W = (1 + r_{pi^*})(H_3(i^* || j^* || k^*, name_{i^*} || time_{i^*j^*k^*}) + F_{i^*j^*k^*})$.

**Probability and Time Analysis**

- We analyze $\boldsymbol{C}_1$'s probability and time of solving CDH problem with the $\boldsymbol{A}_1$'s ability to forge tag of our improved scheme. For the following four events:
  - $\mathcal{E}_1$: $\boldsymbol{C}_1$ does not abort for any $\boldsymbol{A}_1$'s Extract queries.
  - $\mathcal{E}_2$: $\boldsymbol{C}_1$ does not abort for any $\boldsymbol{A}_1$'s ProxyKeyGen queries.
  - $\mathcal{E}_3$: $\boldsymbol{C}_1$ does not abort for any $\boldsymbol{A}_1$'s Tag queries.
  - $\mathcal{E}_4$: $\boldsymbol{A}_1$ generates a valid tag $\sigma_{i^*j^*k^*}$ for block$F_{i^*j^*k^*}$for proxy $ID_p$ with warrant $\omega_{i^*}$, where $i^* \neq p, (p, i^*) \notin S_1', (p, i^*, j^*, k^*) \notin S_1''$

  If $\boldsymbol{A}_1$ succeeds in all the above events and $H_1$ answers $g^b$with probability $(1 - \delta)$, then $\boldsymbol{C}_1$'s probability for CDH solution is: $\Pr[\mathcal{E}_1 \wedge \mathcal{E}_2 \wedge \mathcal{E}_3 \wedge \mathcal{E}_4] = \Pr[\mathcal{E}_1] \Pr[\mathcal{E}_2 | \mathcal{E}_1] \Pr[\mathcal{E}_3 | \mathcal{E}_2 \wedge \mathcal{E}_1] \Pr[\mathcal{E}_4 | \mathcal{E}_3 \wedge \mathcal{E}_2 \wedge \mathcal{E}_1] = \left(\delta(\widehat{N} - 1)/\widehat{N}\right)^{q_E} \left(1/\widehat{N}\right)^{q_P} \delta^{q_T} \epsilon_1 (1 - \delta)$ . With $\delta = (q_E + q_T)/(q_E + q_T + 1)$, the probability is at least $\epsilon_1 \left(\widehat{N} - 1\right)^{q_E} / (\hat{e}\, \widehat{N}^{q_E + q_P} (q_E + q_T + 1))$, where $\hat{e}$ is the natural logarithm, $\widehat{N}$ is number of selected identities.

The total running time of $\boldsymbol{C}_1$ comprises of $\boldsymbol{A}_1$'s running time $t_1$ and additional time, where $\boldsymbol{C}_1$ responds with $(q_H + q_T)$ hash, $q_E$ Extract, $q_P$ ProxyKeyGen, $q_T$ TagGen queries and final

CDH problem transforming time. Hash response, Extract and ProxyKeyGen require at most once exponentiation on group $G_1$ for each query, while it takes triple exponentiation for Tag oracle query. $S'^{-z_{i^*j^*k^*}}$ could be computed by one exponentiation on $S'$ and one inversion, and so it is with computation on $U_{i^*}$. Final $(\cdot)^{1/W}$ requires exponentiation with $1/W$. So twice inversion and triple exponentiation on $G_1$ are required for final output of CDH solution. Therefore, the total running time is at most $t_1 + t_{G_1} \cdot (q_H + q_E + q_P + 4q_T + 5)$.

**Theorem 2** (Unforgeability) If there exists PPT time $(t_2, \epsilon_2)$-adversary $\boldsymbol{A}_2$ who could fabricate valid proof of our Sec-ID-BPAPP, then our scheme is unforgeable when challenger $\boldsymbol{C}_2$ could solve BDH problem with non-negligibility $\epsilon_2\left(\widehat{N}-1\right)^{q_E}/(\hat{e}^{c^*}\ \widehat{N}^{q_E+q_P}(q_E+q_T+1)$ ) with PPT time $t_2 + t_{G_1} \cdot (q_H + q_E + q_P + 4q_T + 2|O| + 4) + 2t_{G_2} + t_e$.

**Proof**: There are $\widehat{N}$ number of selected identities $\{ID_i\}_{i\in O}$ having the proxy $ID_p$. The original data file $\{\tilde{F}_i\}_{i\in O}$ will be divided into blocks $\{F_{ijk}\}_{i\in O, j\in J, k\in\delta_{ij}}$ before being outsourced on clouds $\{CS_j\}_{j\in J}$.

- Setup: Like Theorem 1, $\boldsymbol{C}_2$ in the role of PKG, generates master key pairs $(msk, mpk) = (a, g^a)$from generator $g$ with $a, b, w \in Z_q$, and creates BDH instance as $g, g^a, g^b, g^w \in G_1$ to compute $e(g,g)^{abw} \in G_2$. It also allows $\boldsymbol{A}_2$ to access $R_{pi}$ as $H_1\left(ID_p\right)^{r_{pi}}$ where $r_{pi} \in Z_q$.
- $H_1$ -oracle, $H_2$ -oracle, $H_3$ -oracle, $H_4$ -oracle, Extract-oracle, ProxyKeyGen-oracle, Tag-oracle, remain the same as Theorem 1.
- First phase queries: $\boldsymbol{A}_2$ could access all the oracles. Let us denote index set $ID_i$ of private key extracting as $S_2, (p \notin S_2)$, the index pair set $(ID_{p'}, ID_i)$ of proxy tag secret key query as $S_2'$ $((p,i) \notin S_2')$, the tuple set of index and data for proxy tag query as $S_2''((p,i,j,k,F_{ijk}) \notin S_2'')$.
- Challenge phase: $\boldsymbol{C}_2$ generates challenge set $chal$ with ordered number collection $\{c_{i^*j^*}\}$ to specify every block $F_{i^*j^*k_n^*}$ on the $j^*$th cloud for owner of $ID_{i^*}$( $\{(p, i^*, j^*, k_n^*)|1 \le n \le c_{i^*j^*}\}$, and $i^* \neq p$, $(p, i^*) \notin S_2'$ , $(p, i^*, j^*, k_n^*, F_{i^*j^*k_n^*}) \notin S_2''$), and masking $M = mpk^w$ for privacy-preserving auditing. $chal$ will be sent to $\boldsymbol{A}_2$.
- Second phase queries: $\boldsymbol{A}_2$ makes queries similar to *First phase*. Denote index set of identities for Extract private key queries as $S_3$, index set of identity pairs for proxy tag secret key queries as $S_3'$, tuple set of index and data for proxy tags queries as $S_3''$. We require that $p \notin S_2 \cup S_3$, $(p,i) \notin S_2' \cup S_3'$ and $(p,i,j,k,F_{ijk}) \notin S_2'' \cup S_3''$.

**Forgery Output**: Finally, $\boldsymbol{A}_2$ itself outputs valid proof $\{P_{j^*}\}_{j^*\in J}$ for $\{F_{i^*j^*k_n^*}\}_{1\le n\le c_{i^*j^*}}$ and tags generated by proxy $ID_p$ with warrants $\{\omega_{i^*}\}_{i^*\in O}$ and signatures $\{(U_{i^*}, R_{i^*}, \xi_{i^*})\}_{i^*\in O}$. $\boldsymbol{C}_2$ looks up lists of Extract-oracle, ProxyKeyGen-oracle and Tag-oracle. It will abort and terminate unless none of corresponding records exists. If game could proceed, $\boldsymbol{C}_2$ keeps on checking all hash function oracles and makes queries itself if there is no relative record in their lists. $R_{pi^*} = H_1\left(ID_p\right)^{r_{pi^*}}$, $g^w$ in Setup and $U_{i^*} = (g^a)^{y_{i^*}r_{i^*}H_2(\omega_{i^*}||R_{i^*})}$ for validity of warrant $\omega_{i^*}$.

Since valid proof $\{P_{j^*}\}_{j^*\in J} = \{(T_{j^*}', S', M_{j^*}')\}_{j^*\in J}$ satisfies (6), with corresponding records of oracles and properties of bilinear mapping as:

$$e(\prod_{j^*\in J} T'_{j^*}, g^w) = e\left(\prod_{i^*\in O}\left(H_1(ID_p)\cdot R_{i^*}^{H_2(\omega_{i^*}||R_{i^*})} \cdot R_{pi^*}\right)^{\sum_{j^*\in J}\sum_{n\in[1,c_{i^*j^*}]} h_{i^*j^*k_n^*}\cdot H_3(i^*||j^*||k_n^*,name_{i^*}||time_{i^*j^*k_n^*})}, (g^a)^w\right)$$
$$\cdot e\left(\prod_{i^*\in O}\prod_{j^*\in J}\prod_{n\in[1,c_{i^*j^*}]}\left(g^{z_{i^*j^*k_n^*}}\right)^{h_{i^*j^*k_n^*}}, S'^w\right)\cdot\prod_{j^*\in J} M'_{j^*}$$
$$= e\left(g^{ab\sum_{i^*\in O}\sum_{j^*\in J}\sum_{n\in[1,c_{i^*j^*}]}(1+r_{pi^*})h_{i^*j^*k_n^*}\cdot H_3(i^*||j^*||k_n^*,name_{i^*}||time_{i^*j^*k_n^*})}\right.$$
$$\cdot\prod_{i^*\in O} U_{i^*}^{\sum_{j^*\in J}\sum_{n\in[1,c_{i^*j^*}]} h_{i^*j^*k_n^*}\cdot H_3(i^*||j^*||k_n^*,name_{i^*}||time_{i^*j^*k_n^*})}$$
$$\left.\cdot S'^{\sum_{i^*\in O}\sum_{j^*\in J}\sum_{n\in[1,c_{i^*j^*}]} z_{i^*j^*k_n^*}\cdot h_{i^*j^*k_n^*}}, \ g^w\right)\cdot\prod_{j^*\in J} M'_{j^*}$$

The BDH problem solution is obtained after simplifications:

$$e(g,g)^{abw} = \left(e(\textstyle\prod_{j^*\in J} T'_{j^*}\cdot W'^{-1}, g^w)\cdot M'^{-1}\right)^{\frac{1}{E}},$$

Where $M' = \sum_{j^*\in J} M'_{j^*}$, $h_{i^*j^*k_n^*} = f_{v_{i^*j^*,2}}(i^*, j^*, k_n^*)$,

$$W' = \prod_{i^*\in O} U_{i^*}^{-\sum_{j^*\in J}\sum_{n\in[1,c_{i^*j^*}]} h_{i^*j^*k_n^*}\cdot H_3(i^*||j^*||k_n^*,name_{i^*}||time_{i^*j^*k_n^*})} \cdot S'^{-\sum_{i^*\in O}\sum_{j^*\in J}\sum_{n\in[1,c_{i^*j^*}]} z_{i^*j^*k_n^*}\cdot h_{i^*j^*k_n^*}}$$

$$E = \sum_{i^*\in O}\sum_{j^*\in J}\sum_{n\in[1,c_{i^*j^*}]}(1+r_{pi^*})h_{i^*j^*k_n^*}\cdot H_3(i^*||j^*||k_n^*,name_{i^*}||time_{i^*j^*k_n^*})$$

**Probability and Time Analysis**

· We analyze $\boldsymbol{C}_2$'s probability and time of solving BDH problem with the $\boldsymbol{A}_2$'s ability to forge proof of our improved scheme. For the following four events:

- $\mathcal{E}_1$: $\boldsymbol{C}_2$ does not abort for any $\boldsymbol{A}_2$'s Extract queries.
- $\mathcal{E}_2$: $\boldsymbol{C}_2$ does not abort for any $\boldsymbol{A}_2$'s ProxyKeyGen queries.
- $\mathcal{E}_3$: $\boldsymbol{C}_2$ does not abort for any $\boldsymbol{A}_2$'s TagGen queries.
- $\mathcal{E}_4$: $\boldsymbol{A}_2$ generates a valid proof $\{P_{j^*}\}_{j^*\in J}$ for challenged blocks $\{F_{i^*j^*k_n^*}\}_{n\in[1,c_{i^*j^*}]}$ by proxy $ID_p$ with warrants $\{\omega_{i^*}\}_{i^*\in O}$, where $i^*\neq p$,$(p,i^*)\notin S'_2\cup S'_3$, $(p,i^*,j^*,k_n^*)\notin S''_2\cup S''_3$.

If $\boldsymbol{A}_2$ succeeds in all the above events and $H_1$ answers $g^b$ with $(1-\delta)$, then $\boldsymbol{C}_2$'s probability for BDH solution is: $\Pr[\mathcal{E}_1\wedge\mathcal{E}_2\wedge\mathcal{E}_3\wedge\mathcal{E}_4] = \Pr[\mathcal{E}_1]\Pr[\mathcal{E}_2|\mathcal{E}_1]\Pr[\mathcal{E}_3|\mathcal{E}_2\wedge\mathcal{E}_1]\Pr[\mathcal{E}_4|\mathcal{E}_3\wedge\mathcal{E}_2\wedge\mathcal{E}_1] = \left(\delta(\widehat{N}-1)/\widehat{N}\right)^{q_E}\left(1/\widehat{N}\right)^{q_P}\delta^{q_T}\epsilon_2(1-\delta^{c^{*-1}})$. With $\delta = ((q_E+q_T)/(q_E+q_T+1))^{c^*}$, the probability is at least $\epsilon_2\left(\widehat{N}-1\right)^{q_E}/(\hat{e}^{c^*}\widehat{N}^{q_E+q_P}(q_E+q_T+1))$, where $\hat{e}$ is the natural logarithm, $\widehat{N}$ is number of selected identities, $c_{i^*j^*}$is the number of challenged blocks on corresponding cloud for specified owner, and $c^* = \left(\sum_{i^*\in O, j^*\in J} c_{i^*j^*}\right)^{-1}$.

The total running time of $\boldsymbol{C}_2$ comprises of $\boldsymbol{A}_2$'s running time $t_2$ and additional time, where there are $|O|$ number of owners, $\boldsymbol{C}_2$ responds with $(q_H+q_T)$ hash, $q_E$ Extract, $q_P$ ProxyKeyGen, $q_T$ TagGen queries and final BDH problem transforming time. Hash response, Extract and ProxyKeyGen require at most once exponentiation on group $G_1$ for each query, while it takes triple exponentiation for Tag oracle query. One pairing, $(|O|+2)$ inversion and $(|O|+2)$ exponentiation on $G_1$, one inversion and one exponentiation on $G_2$ are spent for

final output of BDH solution. Therefore, the total running time is at most $t_2 + t_{G_1} \cdot (q_H + q_E + q_P + 4q_T + 2|O| + 4) + 2t_{G_2} + t_e$. We complete the proof.

**Theorem 3** (Privacy-preserving) If there exists PPT time TPA which could recover original data in our Sec-ID-BPAPP, then our scheme is privacy-preserving when challenger could solve DL problem with non-negligibility with PPT time.

**Proof**: After TPA receives masked data proof as $M'_j = e\left(\prod_{i\in O_j}\left(H_1(ID_p)\cdot\left(R_i{}^{H_2(\omega_i||R_i)}\cdot R_{pi}\right)\right)^{F'_{ij}}, M\right)$. Denote $g' = e\left(\prod_{i\in O_j} H_1(ID_p)\cdot\left(R_i{}^{H_2(\omega_i||R_i)}\cdot R_{pi}\right), M\right)$ and thus $M'_j = (g')^{F'_{ij}}$. If TPA retrieves original data combination $F'_{ij} = \sum_{k\in\delta_{ij}} h_{ijk}\cdot F_{ijk}$ for further recovering data blocks $\{F_{ijk}\}$, then challenger could solve DL problem as given $g' \in G_2$, $(g')^{F'_{ij}} \in G_2$, obtaining $F'_{ij} \in Z_q$. We complete the proof.

## 7. Efficiency Analysis

In this section, we compare overheads of computation and communication of our improved scheme Sec-ID-BPAPP, with Wang et al.'s ID-PUIC [27], summarized in **Table 1** and **Table 2**, respectively. In addition, the performance comparison on computation is depicted in **Fig. 2**, based on results from simulation of the two schemes on a laptop, to evaluate efficiency trend when number of data owners, clouds and data amount increases.

**Table 1.** Computation Cost Comparison for Multiple Owners and Multiple Clouds

| Schemes | TagGen | ProofGen | Verify | Privacy |
|---|---|---|---|---|
| ID-PUIC [27] | $2NC_{exp}$ | $cC_{exp}$ | $(2n_1\,n_2)C_e + (c + n_1\,n_2)C_{exp}$ | not |
| Sec-ID-BPAPP | $(2N + n_O)C_{exp}$ | $(c + n_1 n_2)C_{exp} + n_2 C_e$ | $3C_e + (c + n_1 + 2)C_{exp}$ | Yes |

**Table 2.** Communication Cost Comparison for Multiple Owners and Multiple Clouds

| Schemes | Challenge | ProofGen | Privacy |
|---|---|---|---|
| ID-PUIC [27] | $n_1 n_2\ log_2 N + 2n_1 n_2\ log_2 q$ | $n_1 n_2\ \mathcal{G}_1 + n_1 n_2\ log_2 q$ | Not |
| Sec-ID-BPAPP | $n_1 n_2\ log_2 N + 2n_1 n_2\ log_2 q + n_2\ \mathcal{G}_1$ | $2n_2\ \mathcal{G}_1 + n_2\ \mathcal{G}_2$ | Yes |

- Assume there are $n_O$ data owners storing total $N$ blocks $\{F_{ijk}\}$ on $n_J$ clouds, by only *one-off* TagGen and upload. To prove data integrity, *periodical* Challenge and Verify will be executed between clouds and TPA, upon randomly selected $c$ data blocks of $n_1$ data owners on $n_2$ clouds with their tags, element size of group $G_1$ is $\mathcal{G}_1$, $\mathcal{G}_2$ is for $G_2$. Consequently, the dominant cost of this scheme is mostly contributed by ProofGen and Verify.
- Among all the operations, bilinear pairings $C_e$, exponentiation $C_{exp}$ on group $G_1$, and hash $C_h$ on blocks are most expensive, compared with multiplication on $G_1$and $G_2$, operation on $Z_q$, and other hash operations, which are efficient or can be done for only once. Additionally, since ID-PUIC only offers single owner's data auditing on one cloud, we consider repeating $n_1 n_2$ loops of ID-PUIC instances, with $N/(n_1 n_2)$ outsourced blocks and only challenged $c/(n_1 n_2)$ blocks per loop.

**Analysis for computation**: In order to fully protect tags $\{\sigma_{ijk} = (S_{ijk}, T_{ijk})\}$ from being utilized to recover its private keys by adversaries, proxy requires $(2N + n_O)C_{exp}$ operation for $n_O$ data owners in TagGen. Luckily, these could be performed off line for proxy as one-off task, although a little bit expensive. After one exponentiation $C_{exp}$ for masking element $M$ in Challenge, our Sec-ID-BPAPP spends $(c + n_1 n_2)C_{exp} + n_2 C_e$ for all $\{P_j\}$ in ProofGen, where $n_2$ clouds additionally perform $n_1 n_2 C_{exp} + n_2 C_e$ for generating masked data proof, in order to realize privacy-preserving auditing on TPA's side and reduce its computation load. And thus in Verify, TPA needs only 3 bilinear pairing to allow batch auditing at one time, which achieves enhanced security of proxy private key protection and still outperforms $2n_1 n_2$ pairings in Wang et al.'s ID-PUIC [27], if applied to the multiple clouds and multiple owners scenario in **Table 1**.

**Analysis for communication**: To enable privacy-preserving auditing, we first require special $n_2\, \mathcal{G}_1$ size of element from Challenge to mask data in ProofGen, which later successfully outputs masked data in the size of $n_2\, \mathcal{G}_2$ for final auditing. But for total proof, which includes both aggregate tag and masked data, our Sec-ID-BPAPP of $n_2\ (2\mathcal{G}_1 + \mathcal{G}_2)$ is still less than ID-PUIC's $n_1 n_2\ (\mathcal{G}_1 + log_2 q)$, which is linear to both $n_1$ and $n_2$. If taking Challenge and ProofGen together, our proposed scheme introduces less bandwidth than ID-PUIC, since $n_2 \ll n_1$ in the multiple clouds and multiple owners' setting in **Table 2**.

**Simulation**: In order to compare the performance about Wang et al.'s ID-PUIC [27] versus our Sec-ID-BPAPP, we simulate data owners, proxy, storage clouds, and TPA on a laptop of Intel core i5 480 M at 2.67 GHz and 4G RAM running Linux operation system (Ubuntu 18.04 64bit with kernel 4.15.0-23-generic), in C programming language. Both of schemes are based on Pairing-Based Cryptography Library (PBC 0.5.14) [33], GNU Multiple Precision Arithmetic Library (GMP 6.1.2) [34] and OpenSSL Library (OpenSSL-1.1.0) [35].

To achieve 80-bit AES level of security, the elliptic curve we are using is of 160 bit group order with 512 bit length finite field element for $G_1$ and $G_2$, from Type-A pairing in PBC library. Therefore, the size of element is $\mathcal{G}_1 = \mathcal{G}_2 = 64$ Bytes, and $q$ is 20 Bytes length prime. For generating challenging co-efficient $\{h_{ijk}\}$, we consider HMAC-SHA256 as pseudo random function $f$ in OpenSSL library. We set each data block $F_{ijk}$ as 20 B. The simulation has run 10 trials and collected their mean values as results.

For TagGen computation of proxy tags for total 1000000 blocks of 50 data owners, ID-PUIC requires 6825.433 seconds and Sec-ID-BPAPP is 6275.664 seconds. In order to prove total 1000000 blocks outsourced on 10 clouds for 50 data owners, running time of ProofGen is 2745.109 seconds of Sec-ID-BPAPP versus 53.984 seconds in ID-PUIC. Our Sec-ID-BPAPP indeed takes more time to generate masked data proof on the clouds. But this enables privacy-preserving public auditing advantage over ID-PUIC, and reduces TPA's computation in the batch owners and clouds integrity auditing task on in Verify as follows.

On the left half of **Fig. 2**, the computation time on TPA's side is depicted for Wang et al.'s ID-PUIC [27] (marked in blue bar) and our Sec-ID-BPAPP (in yellow bar), when challenged data owners increases from 50 to 250. For the fairness of evaluation, we repeat Wang et al.'s scheme to achieve the same number of data owners and clouds. Assume there are 10 clouds, each of which stores 2000 blocks for every data owner, and the total number of challenged data blocks will range from $1.0 \times 10^6$ to $5.0 \times 10^6$ (marked on the top X-axis), based on 100% probability to detect 1% rate of modification. It is illustrated that our improved scheme has less computation overheads on TPA's side versus Wang et al.'s scheme.

Followed up with the right half, in **Fig. 2**, we present the computation time of on TPA's side as the number of challenged clouds increases from 10 to 50, for ID-PUIC [27] (marked in blue

bar) and Sec-ID-BPAPP (in yellow bar), based on 100% probability to detect 1% rate of modification. Imagine there are 50 data owners, each of which outsources 2000 blocks on every cloud, and thus the total number of challenged data blocks will range from $1.0 \times 10^6$ to $5.0 \times 10^6$ (shown on the top X-axis). We also repeat ID-PUIC for the fairness of evaluation. It is shown that Sec-ID-BPAPP introduces less computation overheads on TPA's side.

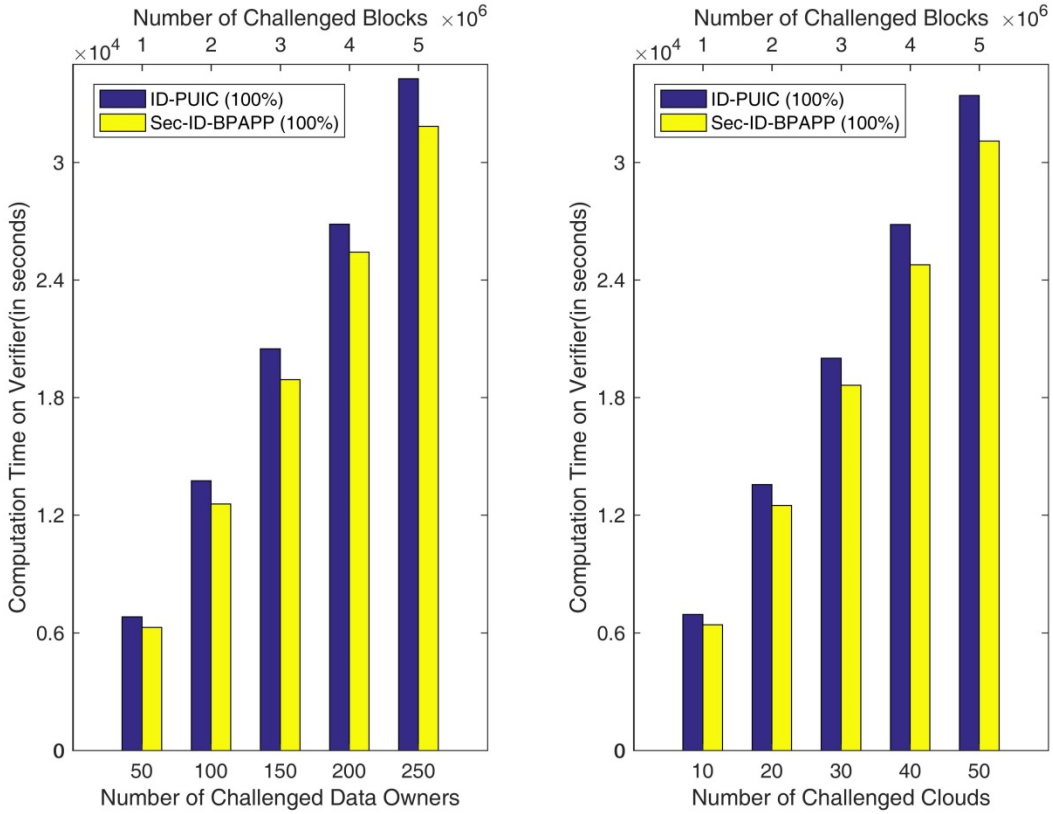


**Fig. 2.** Comparison of computation on TPA:
1) as number of Owners increases: Total 10 Clouds of each stores 2000 blocks per owner;
2) as number of Clouds increases: Total 50 Owners of each outsources 2000 blocks per cloud

The difference illustrated in the **Fig. 2** is able to predict their trend of performance upon extrapolation to real multiple clouds storage system, which are equipped with powerful CPUs and huge memories, even if the performance of two schemes are temporarily limited by our simulated laptop. Therefore, our Sec-ID-BPAPP is more efficient than Wang et al.'s ID-PUIC for the secure big data storage, which might have billion number of data owners, large number of storage clouds and large volume of data, in terms of storage integrity.

For total communication overheads of Challenge and ProofGen, our Sec-ID-BPAPP of $(n_1n_2/8\ log_2N + 40n_1n_2 + 256n_2)$ outperforms ID-PUIC's $(n_1n_2/8\ log_2N + 124n_1n_2)$, since the number of challenged clouds $n_2$ is usually much smaller than the number of challenged owners $n_1$. Especially, Sec-ID-BPAPP requires $(n_1n_2/8\ log_2N + 40n_1n_2 + 64n_2)$ B and $192n_2$ B while ID-PUIC costs $(n_1n_2/8\ log_2N + 40n_1n_2)$ B and $84n_1n_2$ B, for Challenge and ProofGen respectively, upon 64 B element size of group $G_1$ and $G_2$, 20B per block. In real cloud storage, TPA could employ sampling technology in [5] for economic auditing, e.g. randomly challenging 460 blocks is sufficient to detect 1% data error with 99%

probability among entire multiple clouds storage system.

## 8. Conclusions and Open Problem

In this paper, we revisited an identity-based batch public auditing with proxy processing (ID-BPAPP) scheme [32] designed by Yu et al. in KSII transactions on Internet and Information Systems 2017 October, and demonstrated that any cloud in their scheme could deceive TPA without original data. In particular, it is also feasible to recover proxy's private key to generate tags by malicious clouds or data owners themselves. This will inevitably incur potential impersonation, and even might be leveraged to threaten digital properties of proxy. Therefore, we propose our solution to repair the security flaws and thus enhance the security, at the expense of reasonable overheads while still enjoy better auditing efficiency over ID-PUIC [27].

Despite these security flaws above, it is still of great value for Yu et al. to tackle the batch public data auditing problem with proxy processing, under identity based cryptography infrastructure. As a future work, we will keep on seeking to improve the efficiency of our proposed scheme, to enable practical and secure data integrity auditing on distributed clouds system for multiple owners of restricted access.

## Acknowledgement

This work was supported by State Scholarship Fund Program of China Scholarship Council under Grant 201506070077, the National Key R&D Program of China under Grant 2017YFB0802000 and the National Natural Science Foundation of China under Grant 61370203 and 61872060.

## Reference

[1] Gartner.com, "Gartner Forecasts Worldwide Public Cloud Services Revenue to Reach $260 Billion in 2017," October 12, 2017. Article (CrossRef Link)

[2] IDC.com, "Worldwide Public Cloud Services Spending Forecast to Reach $122.5 Billion in 2017, According to IDC," February 20, 2017. Article (CrossRef Link)

[3] Y. Wang, Q. Wu, B. Qin, W. Shi, R. H. Deng, J. Hu, "Identity-Based Data Outsourcing with Comprehensive Auditing in Clouds, " *IEEE Transactions on Information Forensics and Security*,12(4), 940-952, 2017. Article (CrossRef Link)

[4] Z. Fu, X. Wu, C. Guan, X. Sun, K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, 11(12), 2706-2716, 2016. Article (CrossRef Link)

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, "Provable Data Possession at Untrusted Stores," in *Proc. of ACM CCS 2007*, pp. 598-609, 2007. Article (CrossRef Link)

[6] H. Shacham, B. Waters, "Compact proofs of retrievability," In *Proceedings of ASIACRYPT 2008*, pp. 90-107, 2008. Article (CrossRef Link)

[7] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," *IEEE Transactions on Computers*, 62(2), 362-375, February, 2013. Article (CrossRef Link)

[8] Y. Zhu, H. Hu, G. J. Ahn, M.Yu, "Cooperative Provable Data Possession for Integrity Verification in MultiCloud Storage," *IEEE Transactions Parallel and Distributed Systems*, 23(12), 2231-2244, December, 2012. Article (CrossRef Link)

[9] K. Yang, X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing,"*IEEE Transactions on Parallel and Distributed Systems*, 24(9), 1717-1726, 2013. Article (CrossRef Link)
[10] R. Curtmola, O. Khan, R. Burns, G. Ateniese, "MR-PDP: Multiple-replica provable data possession," In *Proceedings of ICDCS 2008*, pp. 411-420 (2008). Article (CrossRef Link)
[11] B. Wang, B. Li, H. Li, "Panda: public auditing for shared data with efficient user revocation in the cloud," *IEEE Transactions on Services Computing*, 8(1), 92-106, 2015. Article (CrossRef Link)
[12] Q. Wang, C. Wang, K. Ren, W. Lou, J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," *IEEE Transactions on Parallel and Distributed Systems*, 22(5), 847-859, 2011. Article (CrossRef Link)
[13] C. Erway, A. Küpçü, C. Papamanthou, R. Tamassia, "Dynamic Provable Data Possession," *ACM Transactions on Information and System Security*, 17(4), 2015. Article (CrossRef Link)
[14] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, J. Chen, "MuRDPA: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Transactions. on Computers*, 64(9), 2609-2622, 2015. Article (CrossRef Link)
[15] A. F. Barsoum, M. A. Hasan, "Provable multicopy dynamic data possession in cloud computing systems," *IEEE Transactions on Information Forensics and Security*, 10(3), pp. 485-497, 2015. Article (CrossRef Link)
[16] J. Wang, X. Chen, X. Huang, I. You, and Y. Xiang, "Verifiable auditing for outsourced database in cloud computing," *IEEE Transactions on Computers*, 64(11), 3293-3303, 2015. Article (CrossRef Link)
[17] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, "Attribute-based keyword search over hierarchical data in cloud computing," *IEEE Transactions on Services Computing*, 2018. Article (CrossRef Link)
[18] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, 5 (4), 3008-3018, 2018. Article (CrossRef Link)
[19] Y. Miao, J. Ma, X. Liu, J. Weng, and H. Li, H Li, "Lightweight fine-grained search over encrypted data in fog computing," *IEEE Transactions on Services Computing*, 2018. Article (CrossRef Link)
[20] Y. Miao, J. Weng, X. Liu, KKR Choo, Z. Liu, and H. Li, "Enabling verifiable multiple keywords search over encrypted cloud data," *Information Sciences*, 465, 21-37, 2018. Article (CrossRef Link)
[21] J. Zhao, C. Xu, F. Li, W. Zhang, "Identity-based public verification with privacy preserving for data storage security in cloud computing," *IEICE Transactions Fundamentals Electronics, Communications and Computer Sciences*, 96(12), 2709-2716, 2013. Article (CrossRef Link)
[22] D. Boneh, M. Franklin, "Identity-based encryption from the weil pairing, " in *Proc. of CRYPTO 2001*, LNCS 2139, pp. 213-229, 2001. Article (CrossRef Link)
[23] H. Wang, "Identity-based distributed provable data possession in multicloud storage," *IEEE Transactions on Services Computing*, 8(2), 328-340, 2015. Article (CrossRef Link)
[24] Y. Yu, Y. Zhang, Y. Mu, W. Susilo, "Provably Secure Identity based Provable Data Possession," in *Proc. of ProvSec 2015*, LNCS 9451, pp. 1-16, Springer, Heidelberg, 2015. Article (CrossRef Link)
[25] H. Liu, Y. Mu, J. Zhao, C. Xu, H. Wang, L. Chen, et al., "Identity-based provable data possession revisited: security analysis and generic construction," *Computer Standards & Interfaces*, 54(1), 10-19, 2017. Article (CrossRef Link)
[26] Y. Yu, M. H. A. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, G. Min , "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Transactions on Information Forensics and Security*, 12(4), 767-778, April, 2017. Article (CrossRef Link)
[27] H. Wang, D. He, S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Transactions on Information Forensics and Security*, 11(6), 1165-1176, 2016. Article (CrossRef Link)
[28] X. Zhang, H. Wang and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," *Information Science*, 472, 223-234, 2019. Article (CrossRef Link)

[29] S. Peng, F. Zhou, J. Xu, Z. Xu, “Comments on "Identity-Based Distributed Provable Data Possession in Multicloud Storage, ” *IEEE Transactions on Services Computing*, 9(6), 996-998, Nov.-Dec, 2016. Article (CrossRef Link)
[30] J. Zhao, C. Xu and K. Chen, “A Security-Enhanced Identity-Based Batch Provable Data Possession Scheme for Big Data Storage,” *KSII Transactions on Internet and Information Systems*, 12(9), 4576-4598, 2018. Article (CrossRef Link)
[31] J. Coron, “On the exact security of full domain hash,” In Bellare, M. (ed.) *CRYPTO 2000*. LNCS, vol. 1880, pp. 220-235. Springer, Heidelberg (2000). Article (CrossRef Link)
[32] H. Yu, Y. Cai, S. Kong, et al, “Efficient and Secure Identity-Based Public Auditing for Dynamic Outsourced Data with Proxy,” *KSII transactions on Internet and Information Systems*, 11(10), Oct. 5039-5061, 2017. Article (CrossRef Link)
[33] The Pairing-Based Cryptography Library (PBC). Article (CrossRef Link)
[34] The GNU Multiple Precision Arithmetic Library (GMP). Article (CrossRef Link)
[35] OpenSSL: cryptography and SSL/TLS Toolkit. Article (CrossRef Link)

**Jining Zhao** received his M.Sc. degree in University of Electronic Science and Technology of China (UESTC) in 2013, P.R. China. He is a Ph.D. degree candidate in information security at University of Electronic Science and Technology of China (UESTC). From 2016/03 to 2018/03, he worked as visiting research scholar in Department of Mathematics and Computer Science, Emory University, USA. He is presently engaged in cloud computing security, network security and cryptography.

**Chunxiang Xu** received her Ph.D. degrees at Xidian University, in 2004 , P.R. China. Dr. Xu is presently engaged in information security, cloud computing security and cryptography as a professor at University of Electronic Science and Technology of China (UESTC).

**Kefei Chen** received his Ph.D. degree from Justus Liebig University Giessen, Germany, in 1994. From 1996 to 2013, he was a professor in Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Currently, he is a Distinguished Professor in School of Science, Hangzhou Normal University, China. His research interests include cryptography and network security.