This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

Cooperative Query Answer Authentication Scheme over Anonymous Sensing Data

Liangmin Wang, *Member, IEEE* Qingqing Xie, and Hong Zhong

Abstract—In cloud service over crowd-sensing data, the data owner (DO) publishes the sensing data through the cloud server, so that the user can obtain the information of interest on demand. But the cloud service providers (CSP) are often untrustworthy. The privacy and security concerns emerge over the authenticity of the query answer and the leakage of the DO identity. To solve these issues, many researchers study the query answer authentication scheme for cloud service system. The traditional technique is providing DO's signature for the published data. But the signature would always reveal DO's identity. To deal with this disadvantage, this paper proposes a cooperative query answer authentication scheme, based on the ring signature, the Merkle hash tree (MHT) and the non-repudiable service protocol. Through the cooperation among the entities in cloud service system, the proposed scheme could not only verify the query answer but also protect the DO's identity. First, it picks up the internal nodes of MHT to sign, as well as the root node. Thus, the verification computation complexity could be significantly reduced from $O(log_2 N)$ to $O(log_2 N^{0.5})$ in the best case. Then it improves an existing ring signature to sign the selected nodes. Furthermore, the proposed scheme employs the non-repudiation protocol during the transmission of query answer and verification object (VO) to protect trading behavior between the CSP and users. The security and performance analysis prove the security and feasibility of the proposed scheme. Extensive experimental results demonstrate its superiority of verification efficiency and communication overhead.

Index Terms—Cooperation, query answer authentication, identity privacy, non-repudiation protocol, crowd-sensing data.

1 Introduction

With the advances of wireless sensor networks and Internet of things, crowd-sensing big data is collected by scattering sensors over a vast field. As time goes by, the fast-growing data volumes make it hard for the sensors to store due to their weak storage and computing resources. It becomes a problem that how to store these crowd-sensing data economically, as well as perform queries on it efficiently. Considering the flexible, on-demand and low-cost usage of cloud storage resources [1], [2], [3], the enterprises and individuals, i.e., data owners (DO), outsource their data to the cloud server. Thus, the users can get the information of interest by asking the cloud service provider (CSP) for searching the outsourced data [3], [4], [5], [6]. Such a cloud service system based on crowd-sensing data comes into being, as shown in Fig. 1.

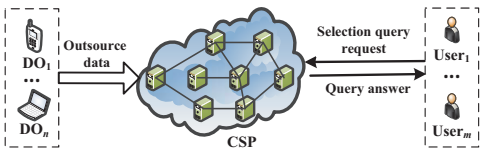

Fig. 1. The cloud service environment over crowd-sensing data.

Observing the service model in Fig. 1, there are three entities in the system: DO, user, and CSP. The crowd-sensing data is provided by many data owners. More users and CSP would join in the system for utilizing these data. Due to the collaborative operation among DO, user and CSP, multiple security and privacy problems have to be taken into consideration. The security and privacy requirements include:

- In demand for privacy preservation, the DO tends to outsource the data anonymously. Thus in some specific application scenarios, the DO is also called as the anonymous data provider.
- The CSP provides the paid service for users. Hence in pursuit of commercial profits, the CSP requires that users can not deny having been served by the CSP if the CSP has sent the proper query answers to the users.
- Since the CSP is often untrustworthy, the users desire urgently for an efficient query answer authentication scheme.

In brief, there are three aspects of requirements: the anonymity of DO identity, the efficient verification for the users' query answers and the non-repudiation of query transaction for the CSP.

At present, there have been some researches related to the query answer authentication over outsourced data [7], [8], [9], [10], [11], [12], [13], [14], [15]. Nevertheless, the existing research works can not satisfy the aforementioned requirements simultaneously. Moreover, the existing research scenarios are far away from the above big-data environment based on crowd-sensing: multiple DOs with anonymity requirement, a very complicated user base who may be dishonest, and an untrustworthy CSP. Furthermore, the anonymity requirement of DO conflicts with the trusti-

- *Liangmin Wang is with the Center of Information Support & Assurance Technology, Anhui University, Hefei, Anhui, 230601 PRC. E-mail: wanglm@ujs.edu.cn*
- *Qingqing Xie and Hong Zhong are with the Department of Computer Science & technology, Anhui University, Hefei, Anhui, 230601. E-mail: xieqn@ahu.edu.cn, zhongh@ahu.edu.cn*
- *Corresponding author's E-mail: xieqn@ahu.edu.cn*

Manuscript received Jan. 20, 2017; revised August 26, 2015.

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

ness authentication for data sources. Hence, the challenge is how to satisfy the aforementioned security and privacy requirements simultaneously, in such a complex cloud service environment.

To overcome the challenges above, we propose a novel cooperative query answer authentication scheme. The DO, CSP and users collaboratively store, search and verify the data. In the proposed scheme, we make the following contributions:

1) We select some internal nodes and root node of MHT as key nodes (KN). The KNs account for $1/2^{\lfloor \log_2 \sqrt{n}-0.5 \rfloor}+1/n$ of the total data records, which greatly reduces the computational and storage overhead, thereby improves the verification efficiency.
2) We adapt the linkable ring signature scheme to sign the KNs. It's worth noting that this step paves the perfect way for anonymous authentication.
3) We construct a non-repudiation protocol based on verification object (VO) to protect the secure interaction between the CSP and the user.

Here assume users issue selection queries of the form as
SELECT * FROM *stream* WHERE $l_i < A_i < u_i$
where (l_i , u_i) are the selection ranges over attribute A_i.

The rest of this paper is structured as follows. In Section 2, we discuss the related works. Section 3 presents some preliminaries. Section 4 describes our system model and the scheme framework. Section 5 presents our proposed cooperative query answer authentication scheme in detail. In Section 6, we analyze the security and performance of the proposed scheme in theory. In Section 7, we evaluate the performance of our scheme by experiment. Finally, Section 8 concludes this paper.

2 Related Works

This paper presents a suite of cooperative query answer authentication scheme over anonymous sensing data. The related works include anonymous data publishing, query answer authentication, and non-repudiation service.

2.1 Anonymous Data Publishing

The anonymous data publishing techniques are aiming to protect individual identification. The research on anonymous data publishing originated from the K-anonymity model proposed by Samarati P. and Sweeney L. in 1998 [16], later amended and supplemented in 2002 [17], [18]. Subsequently, some new models appeared, such as l-diversity model [19], t-closeness model [20], uncertain data models with anonymity [21], [22], etc. The existing anonymous methods include generalization [16], [18], suppression [16], [18], clustering [23], microaggregation [24], anatomy [25], permutation [26], and so on.

In the above data publishing techniques, there is a common request that the data publisher must be trusted. As we all know, it cannot be guaranteed in cloud environment, since the CSP plays the role of data publisher. So we introduce ring signature scheme. It is good at cutting off the correlation between the data and the data signer (i.e., DO). Moreover, the trustiness of data source is well guaranteed. The seminal construction of ring signature scheme was proposed by Rivest, Shamir and Tauman in 2001 [27]. Subsequently, there exist many constructions and variants [28], [29], [30], [31], [32]. In ring signature scheme, DO could sign the message anonymously, and user can check the signature trustiness without knowing the signer [33], [34], [35], [36]. However, if applying the ring signature scheme directly, DO needs to sign all the data records one by one. It is prohibitively impractical when facing the crowd-sensing data in cloud environment.

2.2 Query Answer Authentication

The query answer authentication schemes are used to serve the query user to verify the downloaded data. The straightforward solution for verifying a set of N values is to generate N digital signatures. An improvement on this solution is based on the MHT [10]. Its basic idea is exactly to replace signatures with the much cheaper hashes. The MHT is a binary tree where each leaf contains the hash of a data value, and each internal node contains the hash of the concatenation of its two children. Verification of data values is based on the fact that the hash value of the tree root is authentically published using a digital signature s. To prove the authenticity of any value, DO provides the user with the data value itself and the hash values of the siblings of the nodes that lie in the path that connects the root of the tree with this value. The user, by iteratively computing and concatenating the appropriate hashes, can recompute the hash of the root and verify its correctness using s. Correctness is guaranteed by the security of the public-key digital signature for the hash value of the root node, as well as the collision resistance of the hash functions. By hashing a given node, it becomes computationally infeasible for an adversary to modify the node in a way that ultimately preserves the hash value of the root. MHT is mainly used in query authentication over outsourced data [37], [38], [39]. F. Li, K. Yi, M. Hadjieleftheriou and G. Kollios proposed an authentication of sliding window queries on streams on the basis of MHT later on [8], [12]. D. Wu, B. Choi, J. Xu and C. S. Jensen proposed an authentication of moving top-k spatial keyword queries by extending the MHT [39].

Nevertheless, when facing big-data environment, there exist several disadvantages in MHT. First, the constructed MHT is always extremely high. When executing verification, the user can not get the hash value of the root until completing computing many hash values of internal nodes from the bottom up. Moreover, if the corresponding signature of the root is tampered, these hashing calculating will be meaningless. Thus it has low authentication efficiency and weak security. Second, multiple DOs are not well supported.

2.3 Non-Repudiation Service

Non-repudiation services are aiming to collect, maintain, make available, and validate irrefutable evidence regarding a transaction. They protect the parties involved in a transaction against the other party denying that a particular event or action took place, and collect irrefutable evidence to support the resolution of any such disagreement. The basic non-repudiation services have two properties, i.e.,

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

non-repudiation of origin and non-repudiation of receipt. They are composed of four distinct phases: evidence generation, evidence transfer and storage, evidence verification, and dispute resolution. The fair non-repudiation protocol, proposed by Zhou J. and Gollmann D. in 1996, was the most widely studied [40], [41]. Subsequently, a variety of formal verification and multiple protocol variants appear [42], [43], [44], [45], [46], [47], [48], [49]. So far, the non-repudiation protocols have used various forms of encryption technique, such as encryption technique based on hash function, block encryption technique based on the Chinese Remainder Theorem, multiple blocks encryption technique, encryption technique and secret sharing technique, etc.

There exist several models of non-repudiation protocols. According to whether a trusted authority (TA) is included, they can be divided into two types: non-repudiation protocol with TA and non-repudiation protocol with non-TA. Non-repudiation protocol with non-TA exchanges information step by step [50], [51]. It can achieve just probabilistic security. Its high requirement for computing power of the communicating parties, however, makes it a bottleneck to apply into cloud service model successfully. Non-repudiation protocol with TA can protect the security of non-repudiation services well. Considering the frequent transactions between CSP and user in cloud, we apply a non-repudiation protocol with offline TA [49] to ensure the interests of both sides, where TA does not intervene in the protocol while both the CSP and the user have no incorrect behavior, and no network error occurs.

3 Preliminaries

In this section, we provide the preliminary background of the collision-resistant hash functions and the public-key digital signature schemes.

Collision-resistant hash functions. A hash function H is an efficiently computable function that takes a variable-length input x to a fixed-length output $y = H(x)$. Collision resistance states that it is computationally infeasible to find two different inputs, $x_1 \neq x_2$, such that $H(x_1) = H(x_2)$. Collision-resistant hash functions can be built provably based on various cryptographic assumptions, such as hardness of discrete logarithms. Reference [11] has tested the time cost for hashing. One hashing operation takes approximately 2 to 3 us, which is around 50 times faster than modular multiplication.

Public-key digital signature schemes. A public-key digital signature scheme is used to authenticate the integrity and ownership of the signed message. Firstly, the signer generates a pair of secret and public keys, denoted as (SK, PK), where the secret key SK is kept secret, and the public key PK associated with his identity is published. Subsequently, for any message m that he sends, he computes a signature s_m by: $s_m = S(SK, m)$. The recipient of s_m and m can verify the validity of s_m via $V(PK, m, s_m)$. If $s_m = V(PK, m, s_m)$, s_m is a valid signature and the message m has not been changed.

4 System Model and Framework

4.1 System Model

In our system, there are four entities: CSP, DO, user, and TA, as shown in Fig. 2. The TA is introduced to achieve non-repudiation service between the CSP and user.

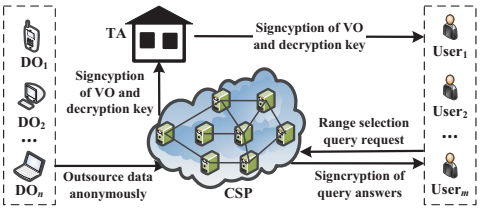

Fig. 2. The cloud system model over crowd-sensing data.

We assume that the CSP is untrustworthy. The DO loses the direct control over the outsourced data. The outsourced data may be tampered, lost, and forged by the CSP, or the attackers. Hence, users suspect whether the query answers provided by the CSP are authentic, complete and trusted.

There are multiple DOs in practical cloud service systems. DO, as the only prover, can provide convincing evidence, such as signature, to verify the stored data in cloud servers. But, the signature is always associated with DO's identity. DO worries that the signature will expose identity privacy to a significant risk. Assume there is a ring signature group consisting of such n DOs.

User is an important and complex role in cloud service systems. He should pay to CSP for the service. But there exist some users who deny having been served by CSP with the intention of avoiding payment. Here, we do not take the data privacy and query privacy into consideration, which is orthogonal to our paper, and can be guaranteed by the existing searchable encryption [52], [53] and order-preserving encryption schemes [54], [55].

TA is assumed to be trustworthy, who are supervised by the government offices. It can ensure the fair transaction between CSP and user, by providing non-repudiable evidence for possible happening denial behavior.

We propose a cooperative query answer authentication scheme, to satisfy the aforementioned three aspects of requirements in Section 1, including the identity privacy preservation for the DO, the efficient verification of query answers for the users, the non-repudiation service between the CSP and users. We will present the scheme framework next. To make our discussion clear, we first explain some notations, as shown in Table 1.

4.2 Framework

Our scheme is a tuple of five algorithms:

• $Setup(1^\lambda) \rightarrow PA$: This algorithm takes as input the security parameter 1^λ, and outputs the public system parameters PA.

• $KeyGenerate(PA) \rightarrow (sk_i, pk_i)$: This algorithm outputs a private/public key pair (sk_i, pk_i) for DO_i.

• $Sign(A_{data}, sk_i, n, Y) \rightarrow \{s\}$: This algorithm takes as input the outsourced data set A_{data}, a certain private key sk_i, the number n of DO members in the signature group, and the corresponding set Y of the n DOs' public keys,

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

TABLE 1
Notations for system construction.

Notation	Description
$A_{data} = \{a_j\}_{j=1,2,\ldots,N}$	a data set, where N is the number of data
$Y = \{pk_1, pk_2, \ldots, pk_n\}$	a set of n DOs' public keys
rs	a range selection query
R	a query answer
VO	a verification object
$H_1 : \{0,1\}^* \rightarrow G$	a hash function, where G is a group of prime order p
$H_2 : \{0,1\}^* \rightarrow Z_p$	a collision-resistant hash function
$X \rightarrow W$	transmission from entity X to entity W
$E_K()$	a symmetric-key encryption function under key K
$D_K()$	a symmetric-key decryption function under key K
$E_X()$	a public-key encryption function under entity X's public key
$D_X()$	a public-key decryption function under entity X's private key
$Sign_X()$	the signature function of entity X
f	a flag indicating the purpose of a message.
$L = H_2(R, K, VO)$	a label that in conjunction with $(CSP, User)$ uniquely identifies a protocol run
$EOO = Sign_{CSP}(f_{EOO}, User, TA, L, H_2(E_K(R)))$	the evidence of origin for the ciphered answer R
$EOR = Sign_{User}(f_{EOR}, CSP, TA, L, H_2(E_K(R)))$	the evidence of receipt for the ciphered answer R
$Sub = Sign_{CSP}(f_{Sub}, User, L, E_{TA}(K\|\|VO))$	the submission evidence for K and VO
$EOO_{K\|\|VO} = Sign_{CSP}(f_{EOO_{K\|\|VO}}, User, L, K, VO)$	the evidence of origin for K and VO
$EOR_{K\|\|VO} = Sign_{User}(f_{EOR_{K\|\|VO}}, CSP, L, K, VO)$	the evidence of receipt for K and VO
$Rec_X = Sign_X(f_{Rec_X}, Y, L)$	the recovery request, where if X is CSP then Y is $User$, else Y is CSP
$Con_{K\|\|VO} = Sign_{TA}(f_{Con_{K\|\|VO}}, CSP, User, L, K, VO)$	the confirmation evidence for K and VO
$Abort = Sign_{CSP}(f_{Abort}, User, L)$	the abort request
$Con_A = Sign_{TA}(f_{Con_A}, CSP, User, L)$	the abort confirmation evidence

then outputs a signature set $\{s\}$, where sk_i's public key pk_i belongs to Y, i.e. $pk_i \in Y$.

• $ServiceTransact(rs) \rightarrow (R, VO, E)$: According to a query request rs, this protocol computes and transacts the corresponding query answer R and the verification object VO. During the transaction, the evidence E of non-repudiation service are generated for solving some transaction disputes.

• $ServiceVerify(n, Y, R, VO) \rightarrow accept/reject$: This algorithm takes as input a public keys set Y, a query answer R, and the corresponding verification object VO, then verifies the query answer R.

5 Our Cooperative Query Answer Authentication Scheme

In this section, we will present the design details of our scheme. Assume the data owner DO_i will store a set $A_{data} = \{a_j\}_{j=1,2,\ldots,N}$ of data to the cloud server. We will take it as an example to present our scheme design.

5.1 $Setup(1^\lambda) \rightarrow PA$

Let G be a group of prime order p where the underlying discrete logarithm problem is intractable. Let $H_1 : \{0,1\}^* \rightarrow G$ and $H_2 : \{0,1\}^* \rightarrow Z_p$ be two hash functions. Let $g = H_1(\text{“}GENERATOR - g\text{”})$ and $h = H_1(\text{“}GENERATOR - h\text{”})$. The public system parameters are $PA = (G, g, h, p, H_1, H_2, \text{“}GENERATOR - g\text{”}, \text{“}GENERATOR - h\text{”})$.

5.2 $KeyGenerate(PA) \rightarrow (sk_i, pk_i)$

DO_i randomly chooses $x_i, y_i \in_R Z_p$ as his secret key $sk_i = (x_i, y_i)$, then computes his public key $pk_i = g^{x_i} h^{y_i}$.

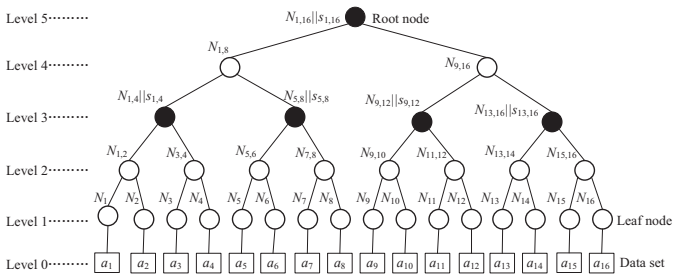

Fig. 3. An example of MHT construction and KN selection.

5.3 $Sign(A_{data}, sk_i, n, Y) \rightarrow \{s\}$

DO_i executes this algorithm to sign the data set A_{data} before uploading it. The procedure includes two steps:

Step 1: Construct a MHT, and choose some key nodes (KN).

Definition 5.1. A key node is defined as the node whose hash value will be signed.

Assume that A_{data} has been sorted on query attribute. Let us construct a MHT whose each leaf contains the hash of a data value, and each internal node contains the hash of the concatenation of its two children [10]. Assume N=16, as shown in Fig. 3, computing $N_1 = H(a_1)$, $N_2 = H(a_2)$, $N_3 = H(a_3)$, $N_4 = H(a_4)$, …, $N_{1,2} = H(N_1||N_2)$, $N_{3,4} = H(N_3||N_4)$, …, in a similar way, $N_{1,4} = H(N_{1,2}||N_{3,4})$, $N_{1,8} = H(N_{1,4}||N_{5,8})$. At last the hash of root node is $N_{1,16} = H(N_{1,8}||N_{9,16})$.

Then we will pick the root node and the internal nodes on the middlemost level $l = \left\lfloor \log_2 \sqrt{N} + 0.5 \right\rfloor$ (From down to up, the levels are defined as level 0, level 1, …, hence the root-node level as level $log_2 N + 1$) as KNs, marked by solid black nodes in Fig. 3. Then the DO will sign the hash values of these KNs, as described in Step 2.

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

Theorem 5.1. When selecting each node on the middlemost level $\left\lfloor \log_2 \sqrt{N} + 0.5 \right\rfloor$ and the root node as KNs, the total number of signed nodes is $\frac{N}{2^{\lfloor \log_2 \sqrt{N} - 0.5 \rfloor}} + 1$.

Proof. Let's prove by mathematical induction. Assume there are $\frac{N}{2^{i-1}}$ nodes on level i ($1 \leq i \leq \log_2 N + 1$). on level 1, i.e., the leaf level, there are $\frac{N}{2^0}$ nodes. The assumption holds for level 1. Suppose the assumption also holds for level i. Since the number of nodes on level i is the double of that on level i+1, there are $\frac{N}{2^i}$ nodes on level $i+1$. The above induction shows that the assumption holds. In consequence, there are $\frac{N}{2^{\lfloor \log_2 \sqrt{N} - 0.5 \rfloor}}$ nodes on level $\left\lfloor \log_2 \sqrt{N} + 0.5 \right\rfloor$. We can claim that total number of signed nodes is $\frac{N}{2^{\lfloor \log_2 \sqrt{N} - 0.5 \rfloor}} + 1$, together with the root node.

Step 2: Sign the hash values of the KNs one by one.

We adapt the linkable ring signature scheme to sign the hash values of these KNs one by one. Algorithm 1 shows the signing procedure, where there are n DOs to participate in signing hash value collaboratively, instead of just DO_i, hence hiding the real data provider DO_i. Moreover, the user still can verify the signature without knowing the true data provider DO_i. The detailed verification procedure will be shown in Section 5.5.

Algorithm 1 AnonySign

Input:
$hashvalue$: the hash value $hashvalue$ of one KN;
sk_i: (x_i, y_i), the private key of the DO_i, $1 \leq i \leq n$;
Y: $\{pk_1, pk_2, \ldots, pk_n\}$, the public key list Y of n DOs in the ring;
Output:
s: the signature of $hashvalue$;
1: choose $r_x, r_y, c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n \in_R Z_p$ randomly;
2: compute $K = g^{r_x} h^{r_y} \prod_{j=1, j \neq i}^{n} z_j^{c_j}$;
3: find c_i satisfying $c_1 + \ldots + c_n \bmod p = H_2(Y \| hashvalue \| K)$, where $H_2 : \{0,1\}^* \rightarrow Z_p$;
4: compute $\tilde{x} = r_x - c_i x_i \bmod p, \tilde{y} = r_y - c_i y_i \bmod p$;
return s= $(\tilde{x}, \tilde{y}, c_1, \ldots, c_n)$.

After signing all the hash values of KNs, DO_i uploads the data set A_{data} and all the signatures to the cloud server. Next, we discuss how to conduct service transaction between the CSP and the user.

5.4 $ServiceTransact(rs) \rightarrow (R, VO, E)$

In this procedure, we will introduce the TA to achieve fair service transaction between the CSP and the user. When the user submits a range selection query rs to the CSP, the CSP first computes query answer R and verification object VO, then sends them to the user following a predetermined protocol. This interaction procedure includes the following two main steps.

Step 1: The CSP computes the R and VO.

When receiving a query request rs from a user, the CSP reconstructs the MHT, and traverses it to find out the query answer R, and simultaneously generates VO by Algorithm 2, where the VO is used to verify R.

Algorithm 2 VOgenerator

Input:
$R = \{a_k, a_{k+1}, \ldots, a_j\}$: the query answer;
$\{s\}$: the signature set generated in Section 5.3;
Output:
VO: the verification object for R;
1: **if** $k > 1$ **then**
2: $VO = \{a_{k-1}\}$, $k' = k - 1$; // the left boundary data
3: **else**
4: $VO = \{null\}$, $k' = k$;
5: **end if**
6: **if** $j < n$ **then**
7: $VO = VO \cup \{a_{j+1}\}$, $j' = j+1$; // the right boundary data
8: **else**
9: $VO = VO \cup \{Null\}$, $j' = j$;
10: **end if**
11: put all the KNs into a KN set, denoted as S_{KN}, where the root node is denoted as $root$;
12: find the KN from down to up in the reconstructed MHT, denoted as $kn \in S_{KN}$, which is the first parent to cover all the hash values of $\{a_{k'}, a_{k'+1}, \ldots, a_{j'}\}$;
13: $VO = VO \cup \{h_{kn} \| s(h_{kn})\}$, where $s(h_{kn})$ is the signature of hash value of kn.
14: let $A(a'_k, kn)$ represents the set of nodes in the path from the leaf node containing a'_k to the KN kn;
15: **for** each node $\alpha \in A(a'_k, kn)$ **do**
16: **if** there exists a left sibling of the node α, denoted as α_left **then**
17: $VO = VO \cup \{the\ hash\ value\ of \alpha\_left\}$;
18: **end if**
19: **end for**
20: let $B(a'_j, kn)$ represents the set of nodes in the path from the leaf node containing a'_j to the KN kn;
21: **for** each node $\beta \in B(a'_j, kn)$ **do**
22: **if** there exists a right sibling of the node β, denoted as β_right **then**
23: $VO = VO \cup \{the\ hash\ value\ of \beta\_right\}$;
24: **end if**
25: **end for**
return VO;//the first two elements are the left and right boundary data of R, the next one is a signature together with the signed hash value, the rest are the necessary hash values.

Let us take Fig. 4 for instance to explain Algorithm 2. The nodes marked by dashed circles make up VO. Assume the query answer $R = \{a_2, a_3\}$, whose left and right boundary data values are $\{a_1, a_4\}$. From down to up, the first KN covering the union section of query answer and two boundary data values, i.e., $\{a_1$,a_2,a_3,$a_4\}$, is the node $N_{1,4}$. There is no left (right) sibling nodes in the path from a_1 (a_4) to $N_{1,4}$. Hence $VO = \{a_1, a_4, N_{1,4} \| s_{N_{1,4}}\}$, where $s_{N_{1,4}}$ is the signature of $N_{1,4}$. While for the query answer $\{a_8$,$a_9\}$, the KN included in VO is the root node.

Step 2: The CSP transmits the R and VO to the user.

Definition 5.2 (Resilient channel). A resilient channel is a channel that delivers data correctly after a finite, but unknown amount of time.

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

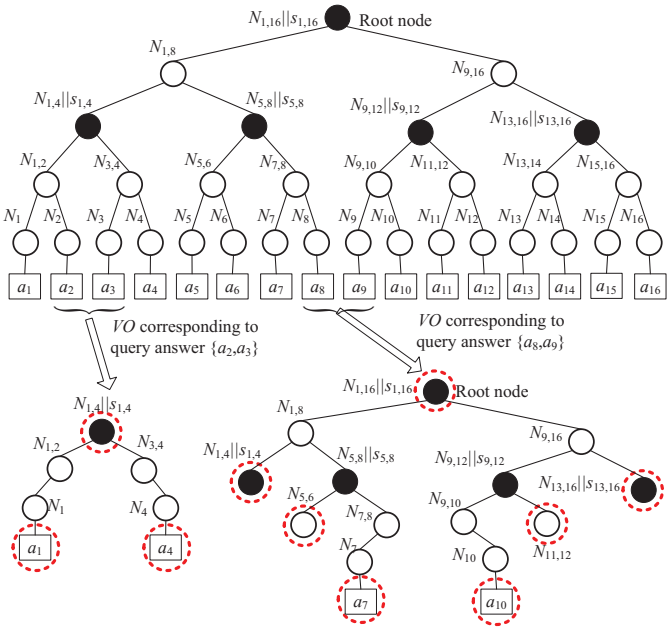

Fig. 4. Examples of VO.

Definition 5.3 (Unreliable channel). An unreliable channel is a channel that may loss data.

Here assume that the communication channels between the TA and both the CSP and user are resilient, while the communication channels between the CSP and user may be unreliable. Inspired by [49], we will apply a fair non-repudiation protocol with off-line TA into transmitting R and VO from the CSP to the user. Here the TA is off-line, because it does not intervene in the protocol while both the CSP and user have no incorrect behavior, and no network error occurs.

Main protocol:

1) $CSP \rightarrow User$: f_{EOO}, f_{Sub}, $User$, TA, L, $E_K(R)$, $E_{TA}(K||VO)$, EOO, Sub
2) $User \rightarrow CSP$: f_{EOR}, CSP, TA, L, EOR
   if CSP times out then **Abort**
3) $CSP \rightarrow User$: $f_{EOO_{K||VO}}$, $User$, L, K, VO, $EOO_{K||VO}$
   if $User$ times out then **Recovery**
4) $User \rightarrow CSP$: $f_{EOR_{K||VO}}$, CSP, L, $EOR_{K||VO}$
   if CSP times out then **Recovery**

Abort Protocol:

1) $CSP \rightarrow User$: f_{Abort}, $User$, $Abort$
   if $aborted$ or $recovered$ then stop
   else $aborted = true$
2) $TA \rightarrow CSP$: f_{Con_A}, CSP, $User$, L, Con_A
3) $TA \rightarrow User$: f_{Con_A}, CSP, $User$, L, Con_A

Recovery Protocol:

1) $X \rightarrow TA$: f_{Rec_X}, f_{Sub}, Y, L, $h(E_K(R))$, $E_{TA}(K||VO)$, Rec_X, Sub, EOR, EOO
   if $aborted$ or $recovered$ then stop
   else $recovered = true$
2) $TA \rightarrow CSP$: $f_{Con_{K||VO}}$, CSP, $User$, L, K, VO, $Con_{K||VO}$, EOR
3) $TA \rightarrow User$: $f_{Con_{K||VO}}$, CSP, $User$, L, K, VO, $Con_{K||VO}$

Our protocol is divided into three sub-protocols, a **Main Protocol**, an **Abort Protocol**, and a **Recovery Protocol**, as shown in above three protocols, respectively. The TA does not intervene in the main protocol. The main protocol consists of two parts. The first part is the exchange of the ciphertext of R under the CSP's key K, and the evidence of origin for the cipher against an evidence of receipt for this cipher. The second part consists of the exchange of the key K, VO and the corresponding evidence of origin against the evidence of receipt for the key K and VO. If the second message in Main protocol does not arrive to the CSP, the CSP executes an Abort protocol. If the third or fourth messages in Main protocol does not arrive, the user and CSP, respectively, can launch a Recovery protocol. The Recovery protocol aims to provide the CSP with the possibly missing evidence of receipt for the cipher (EOR), as well as a substitution ($Con_{K||VO}$) for the evidence of receipt of K, VO, and user with a substitution ($Con_{K||VO}$) of the missing evidence of origin for K, VO, as well as K, VO themselves.

Table 2 shows the evidences owned by the CSP and user, respectively, after executing the above protocols. In practice, most of the time no problem will occur, only the Main Protocol is launched. In this case, the evidence of non-repudiation receipt is $\{EOR, EOR_{K||VO}\}$, saved by CSP. The evidence of non-repudiation origin is $\{EOO, Sub, EOO_{K||VO}\}$, saved by user.

According to [49], it is known that our interaction protocol can guarantee that after transmitting the R and VO, both the CSP and user can receive all the expected non-repudiation evidences related to query behavior, fairly and timely. In Section 6, we will prove the interaction traceability between the CSP and user by utilizing these non-repudiation evidences.

5.5 $ServiceVerify(n, Y, R, VO) \rightarrow accept/reject$

With the public key of CSP, the user first verifies the evidence $EOO_{k||VO}$ to obtain K and VO, next verifies the evidence EOO to obtain the $E_K(R)$. He finally decrypts the $E_K(R)$ with the key K. Now, the user gets the query answer R and verification object VO, and could verify the R according to Algorithm 3.

It is worth noting that what differs from existing query answer authentication schemes is that, the signature provided for user is always companied with the corresponding signed hash value. The purpose is to check the signature's trustiness before verifying the authenticity and completeness of the query answer. Because if the signature is tampered or forged, the subsequent query answer authentication processing (from the 5-th line to the last line in Algorithm 3) is meaningless. The first four lines of code in Algorithm 3 can guarantee that query authentication work is done on the basic of the trusted signature. To some extent, it reduces meaningless verification computation cost. In addition, nobody can identify whom the query data and signature truly come from. Thus our scheme realizes anonymous authentication.

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

TABLE 2
Evidence Notations for our non-repudiation protocol with offline TA.

	No problem occurs	CSP launches **Abort protocol**	CSP or User launches **Recovery Protocol**
Evidences owned by CSP	EOR, $EOR_{K\|\|VO}$	Con_A	EOR, $Con_{K\|\|VO}$
Evidences owned by User	EOO, Sub, $EOO_{K\|\|VO}$	EOO, Sub, Con_A	EOO, Sub, $Con_{K\|\|VO}$

Algorithm 3 Verification

Input:
R: the query answer;
$Y = \{pk_1, pk_2, \ldots, pk_n\}$: a set of public keys;
VO: the verification object;

Output:
$accept$ or $reject$;

1: obtain the KN's signature and hash value from the VO, denoted as s=$(\tilde{x}, \tilde{y}, c_1, \ldots, c_n)$ and $hashvalue$, respectively;
2: compute $c_0 = H_2\left(Y \,\|hashvalue\,\|\, g^{\tilde{x}} h^{\tilde{y}} \prod_{j=1}^{n} pk_j^{c_j}\right)$;
3: **if** $\sum_{j=1}^{n} c_j \bmod p \neq c_0$ **then return** $reject$;
4: **end if**
5: reconstruct the KN's hash value, denoted as $hashvalue'$, according to R and VO;
6: **if** $hashvalue' = hashvalue$ **then return** $accept$;
7: **else return** $reject$;
8: **end if**

6 Security and Performance Analysis

In this section, we first prove the security of our scheme in terms of: the unforgeability and anonymity of the signature, the completeness, authenticity, and trustiness of the query answer, the interaction traceability between CSP and user. Then we present the performance analysis of our scheme.

6.1 Security Analysis

The security of our proposed scheme is mainly based on the discrete logarithm assumption(DLA) is hard.

Definition 6.1 (DLA). For any probabilistic polynomial time (PPT) algorithm A, the probability that $Pr[A(g, g^a) = a]$ is negligible, where g, $g^a \in_R G$.

This Computational Assumption is reasonable, since DLP in large number field is widely considered to be intractable [56], [57], [58]. Therefore a is not deducible from g^a even if g is publicly known. In this paper, the field G is large enough to ensure the security of our scheme.

Definition 6.2 (Unforgeability). A signature scheme is unforgeable if for all PPT adversary $\mathcal{A}$, the probability that he can construct a forged signature s_M which satisfies $V(Y, M, s_M)$=$accept$, denoted as $\mathbf{Adv}_{\mathcal{A}}^{unf} = \Pr[\mathcal{A}$ forges a valid signature$]$ is negligible, where M is a message, and s_M is the signature on M.

Definition 6.3 (Anonymity). A signature scheme is anonymous if for any PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ can guess the signer, denoted as $\mathbf{Adv}_{\mathcal{A}}^{anon} = \Pr[\mathcal{A}$ infers the signer's public key$] - \frac{1}{n}$, is negligible.

Next we will analyze the security of our scheme from seven aspects, as shown in the following seven theorems.

Theorem 6.1 (Unforgeability). Our signature scheme is unforgeable.

Proof: Let's prove by contradiction. Assume a hash value hv, whose signature is denoted as $s = (\tilde{x}, \tilde{y}, c_1, \ldots, c_n)$, provided by DO_i. From Algorithm 3, we can get

$$c_0 = H_2\left(Y \,\|hv\,\|\, g^{\tilde{x}} h^{\tilde{y}} \prod_{j=1}^{n} pk_j^{c_j}\right)$$

where $\sum_{j=1}^{n} c_j \bmod p = c_0$. Now, an PPT adversary forges a signature, denoted as $s' = (\tilde{x}', \tilde{y}', c_1', \ldots, c_n')$, where there exists at least one element is not equal to the counterpart in s, i.e. $\tilde{x}' \neq \tilde{x}$, or $\tilde{y}' \neq \tilde{y}$, or $c_j' \neq c_j, j = 1, \ldots, n$. s' is assumed to satisfy the following equation:

$$c_0' = H_2\left(Y \,\|hv\,\|\, g^{\tilde{x}'} h^{\tilde{y}'} \prod_{j=1}^{n} pk_j^{c_j'}\right)$$

where $\sum_{j=1}^{n} c_j' \bmod p = c_0'$.

1) $c_0' = c_0$. Since the collision-resistance of hash function H_2, we can deduce

$$g^{\tilde{x}'} h^{\tilde{y}'} \prod_{j=1}^{n} pk_j^{c_j'} = g^{\tilde{x}} h^{\tilde{y}} \prod_{j=1}^{n} pk_j^{c_j}$$
$$\Rightarrow g^{\tilde{x}' + \sum_{j=1}^{n} c_j' x_j} h^{\tilde{y}' + \sum_{j=1}^{n} c_j' y_j} = g^{\tilde{x} + \sum_{j=1}^{n} c_j x_j} h^{\tilde{y} + \sum_{j=1}^{n} c_j y_j}$$
$$\Rightarrow \begin{cases} \tilde{x}' + \sum_{j=1}^{n} c_j' x_j = \tilde{x} + \sum_{j=1}^{n} c_j x_j \\ \tilde{y}' + \sum_{j=1}^{n} c_j' y_j = \tilde{y} + \sum_{j=1}^{n} c_j y_j \end{cases}$$
$$\Rightarrow \begin{cases} (\tilde{x}' - \tilde{x}) + \sum_{j=1}^{n} \left(c_j' - c_j\right) x_j = 0 \\ (\tilde{y}' - \tilde{y}) + \sum_{j=1}^{n} \left(c_j' - c_j\right) y_j = 0 \end{cases}$$

   a) One obvious solution to the above equation is $\tilde{x}' = \tilde{x}, \tilde{y}' = \tilde{y}, c_j' = c_j, j = 1, \ldots, n$, which contradicts the assumption that $\tilde{x}' \neq \tilde{x}$, or $\tilde{y}' \neq \tilde{y}$, or $c_j' \neq c_j, j = 1, \ldots, n$.
   b) Other solutions are hard to calculate, since $x_j, y_j, j = 1, \ldots, n$ are kept secret.

2) $c' \neq c_0$. It is obviously hard to compute $\tilde{x}', \tilde{y}', c_1', \ldots, c_n'$ such that $\sum_{j=1}^{n} c_j' = H_2\left(Y \,\|hv\,\|\, g^{\tilde{x}'} h^{\tilde{y}'} \prod_{j=1}^{n} pk_j^{c_j'}\right) \bmod p$, since the one-way hash function and the hard DLP.

Hence, our signature algorithm proves unforgeable.

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

Theorem 6.2 (Anonymity). Our scheme can guarantee DO's identity anonymity.

Proof. Here we assume only public key pk_i can reveal the identity of DO_i. All the information the adversary can get is the signature $s = \{\tilde{x}, \tilde{y}, c_1, \ldots, c_n\}$, public key list Y, and public parameter PA. These information holds the following equation:

$$\sum_{j=1}^{n} c_j = H_2\left(Y \,\|hv\,\|\, g^{\tilde{x}} h^{\tilde{y}} \prod_{j=1}^{n} pk_j^{c_j}\right) \mod p$$

where the prefix $Y||hv$ can tell nothing about the identity of DO_i. We just consider the suffix, i.e., $g^{\tilde{x}} h^{\tilde{y}} \prod_{j=1}^{n} pk_j^{c_j}$. We denote $B = \prod_{j=1}^{n} pk_j^{c_j}$, and $A = g^{\tilde{x}} h^{\tilde{y}} B$. From Algorithm 1, we can obtain

$$\begin{aligned} \tilde{x} &= r_x - c_i x_i \mod p \\ \tilde{y} &= r_y - c_i y_i \mod p \end{aligned}.$$

Hence the following equation holds:

$$\begin{aligned} &A = g^{r_x - c_i x_i} h^{r_y - c_i y_i} B = \frac{g^{r_x} h^{r_y} B}{pk_i^{c_i}} \\ &\Rightarrow pk_i^{c_i} = \frac{B}{A} g^{r_x} h^{r_y} \end{aligned}$$

As described in Algorithm 1, r_x, r_y are random values and kept secret. Hence we can not compute the public key pk_i of DO_i.

Hence, our scheme can guarantee DO's anonymity.

Compared with the scheme in [9], the sign computation cost and verification computation cost are reduced from $E + 2M$ and $2M$ to $E + M$ and M respectively, where E represents an exponentiation, M represents a multi-bases exponentiation which is equal to the cost of approximate 1.3 exponentiation. We achieve this by discarding its linkability.

Theorem 6.3 (Completeness). If hash function is collision-resistant, the query answer and VO are all authentic, then our scheme can verify the completeness of query answer.

Proof. Assume the query answer is $\{a_k, a_{k+1}, \ldots, a_j\}$. In general, the first two elements of VO, as the output of Algorithm 2, are the left and right boundary data. When they are not null, we denote them as $a_{k'}$ and $a_{j'}$, respectively. $a_{k'}$ and $a_{j'}$ surely dissatisfy the query claims. If we can prove that $a_{k'}(a_{j'})$ is left (right) adjacent tightly to $a_k(a_j)$, then our scheme can verify the completeness of $\{a_k, a_{k+1}, \ldots, a_j\}$. Let's prove by contradiction. Assume that $a_{k'}(a_{j'})$ is not left (right) adjacent to $a_k(a_j)$. That is to say, there are other data records between $a_{k'}(a_{j'})$ and $a_k(a_j)$. When executing Algorithm 3, the verification result is $hashvalue' \neq hashvalue$ and failing, since hash function is collision-resistant. In consequence, only when $a_{k'}(a_{j'})$ is left (right) adjacent tightly to $a_k(a_j)$, will the authentication succeeds. So far the proof is completed. Our scheme can verify the completeness of query answer.

Theorem 6.4 (Authenticity). If hash function is collision-resistant, and the query answer is complete, our scheme can verify the authenticity of query answer.

Proof. Let's prove by contradiction. If the query answer is tamped or forged, hash value of the KN, reconstructed using unauthentic query answer and hash values of VO, will differ with the hash value computed from the signature of the same KN. Algorithm 3 will output *reject*. In consequence, only when $\{a_k, a_{k+1}, \ldots, a_j\}$ is authentic, will the verification succeed. The proof is completed. Our scheme can verify the authenticity of query answer.

Theorem 6.5 (Trustiness). If hash function is collision-resistant, our scheme can verify the trustiness of the query answer.

Proof. Theorems 6.3 and 6.4 have proven our scheme can verify the completeness and authenticity of query answer. As for the trustiness, it relies completely on the trustiness of the signature in VO. Assume the signature is denoted as s= $(\tilde{x}, \tilde{y}, c_1, \ldots, c_n)$. From the signing procedure in Algorithm 1, we can obtain

$$\begin{aligned} &c_1 + \ldots + c_n \mod p = H_2\left(Y \,\|hashvalue\,\|K\right) \\ &= H_2\left(Y \,\|hashvalue\,\|\, g^{\tilde{x}+c_i x_i} h^{\tilde{y}+c_i y_i} \prod_{j=1, j\neq i}^{n} pk_j^{c_j}\right) \\ &= H_2\left(Y \,\|hashvalue\,\|\, g^{\tilde{x}} h^{\tilde{y}} (g^{x_i} h^{y_i})^{c_i} \prod_{j=1, j\neq i}^{n} pk_j^{c_j}\right) \\ &= H_2\left(Y \,\|hashvalue\,\|\, g^{\tilde{x}} h^{\tilde{y}} \prod_{j=1}^{n} pk_j^{c_j}\right). \end{aligned} \tag{1}$$

Algorithm 3 tells us that

$$c_0 = H_2\left(Y \,\|hashvalue\,\|\, g^{\tilde{x}} h^{\tilde{y}} \prod_{j=1}^{n} pk_j^{c_j}\right). \tag{2}$$

So $\sum_{j=1}^{n} c_j \mod p$=$c_0$ holds. Till now, the trustiness of the signature is proved. Hence, our scheme can verify the trustiness of the query answer.

Theorem 6.6 (Traceability). Our scheme can guarantee interaction traceability between CSP and user.

Proof. When the transaction between CSP and user is successful, either only the Main Protocol, or both the Main Protocol and Recovery Protocol, are launched. In each case, we can prove the interaction traceability of our scheme.

1. Only the Main Protocol is launched.

CSP has the evidence of non-repudiation receipt $\{EOR, EOR_{K||VO}\}$, and user has the evidence of non-repudiation origin $\{EOO, Sub, EOO_{K||VO}\}$.

After verifying R is correct through Algorithm 3, if user denies receipt of R, CSP can prove his receipt by presenting R, $E_K(R)$, K, VO, L and $\{EOR, EOR_{K||VO}\}$ to TA. TA executes three checks: 1) $EOR_{K||VO}$ is user's signature on $(f_{EOR_{K||VO}}, CSP, L, K, VO)$; 2) EOR is user's signature on $(f_{EOR}, CSP, TA, L, H_2(E_K(R)))$; 3) $R = D_K(E_K(R))$. If the above three checks are positive, TA will conclude user received R.

When CSP denies the origin of incorrect R, user has to present R, $E_K(R)$, K, VO, L and $\{EOO, Sub, EOO_{K||VO}\}$ to TA. Similarly, TA executes four checks: 1) $EOO_{K||VO}$ is CSP's signature on $(f_{EOO}, Cient, L, K, VO)$; 2) EOO is CSP's signature on $(f_{EOO}, User, TA, L, H_2(E_K(R)))$; 3) Sub is CSP's signature on $(f_{Sub}, User, L, E_{TA}(K||VO))$; 4)

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

$R = D_K(E_K(R))$. If the above four checks are all positive, TA claims that CSP is at the origin of incorrect R.

2. Both the Main and Recovery Protocol are launched.

CSP has the evidence of non-repudiation receipt $\{EOR, Con_{K||VO}\}$, and user has the evidence of non-repudiation origin $\{EOO, Sub, Con_{K||VO}\}$. As described in Section 4.2.4, $Con_{K||VO}$, the signature of TA on K, VO, is the substitution for $EOR_{K||VO}$ and $EOO_{K||VO}$, which has equal functions to $EOR_{K||VO}$ and $EOO_{K||VO}$. Hence we can prove the interaction traceability of our scheme in a similar way to the first case.

In all, we get the conclusion that our scheme can guarantee interaction traceability between CSP and user.

6.2 Performance Analysis

Theorem 6.7 (High Efficiency). The high efficiency of our scheme is mainly manifested in $ServiceVerify$ from two aspects: on the one hand, it reconstructs the subtree rooted in the KN of VO; on the other hand, it first checks the trustiness of signature in VO.

Proof. Even though one hashing operation is around 50 times faster than modular multiplication [11], the hashing computation cost is still high when facing the big data. To reduce the hashing computation overhead at user, our scheme improves the traditional MHT-based verification scheme from the following two aspects.

On the one hand, our scheme signs the hash values of the root node, as well as the nodes on level $\left\lfloor \log_2 \sqrt{N} + 0.5 \right\rfloor$. So if there is one KN on internal level covering the query answer and the two boundary values, our scheme just needs to reconstruct the subtree rooted in such a KN, while the traditional schemes always reconstruct the whole MHT.

On the other hand, when verifying query answer, previous schemes firstly reconstruct the hash value of the root node using a lot of hash computation, then compare it with the hash value computed from the signature. If the signature is not trusted, not only much hash computation is wasted, but also the query answer will prove to be false all the same even though it is correct. However, such problem can be avoided in our scheme by concatenating the hash values with their signatures, then checking trustiness before verifying the completeness and authenticity.

Next we will compare our scheme performance with those in [9] and [7]. Our scheme is improved from both [9] and [7]. The scheme proposed in [9] signs the values one by one. Chapter 3 in [7] describes an authentication scheme for selection queries on the basis of MHT. Assume the query answer R contains l data values. The comparison and analysis results are shown in Table 3. In Table 3, our scheme has two cases about verification computation cost and VO size, respectively. It is caused by the KN in VO. If the KN is the root node, both verification computation cost and VO size are the same as those in [7]. If the KN is an internal node, both verification computation cost and VO size are much smaller. In essence, our scheme sacrifices slightly higher sign computation cost to improve the verification efficiency and reduce the communication cost. Hence the sign computation cost is a little higher than that in [7], but far lower than that in [9], which will be proved by experiments in Section 7.

7 Experiment

All algorithms are implemented using Visual C++ 6.0 on a Windows 8.1 system with Intel CORE i7-4500U CPU @ 1.80GHz and 8.00G RAM. We implement the proposed scheme and evaluate the performance over multiple groups of random data. The selection queries are of the form as SELECT * FROM *stream* WHERE $l_i < A_i < u_i$.
In order to compare with other schemes fairly, there are different numbers of random data in each group. Moreover, 1000 random queries are processed. We compute averages of sign computation time, VO generation time, verification computation time, VO size and tamper detection time, respectively, as experimental results shown in Figs. 5–9.

Figs. 5–8 show the cost comparison results of sign computation, VO generation, verification computation and VO storage, respectively. Firstly, it is evident that all these cost, except the VO generation cost, as shown in the lines marked by circles, increase quickly. The scheme proposed in [9] signs all the data values one by one. As a consequence, the CSP need do nothing for VO generation, and just takes the signatures as the VO. In brief, the signature cost is proportional to the whole database size, and the VO generation cost is trivial. In our scenario, the volume of the data is very large. Hence the scheme [9] isn't applicable. By the way, the verification computation cost and VO size in [9] are experimentally proportional to the number of data in a query answer. But for simplicity, we present all these performance in one figure.

Next, let's see the comparison results between our scheme and the scheme proposed in [7], shown in lines marked by triangles and rectangles, respectively. We can see that our scheme has slightly higher sign computation cost, but the lower verification cost and the smaller VO size. Our scheme sacrifices slightly higher sign computation cost to improve the verification efficiency and reduce the communication cost, as analyzed in Theorem 6.7.

Finally, we compare the efficiency in detecting whether the signature is valid or not. Fig. 9 shows our scheme achieves the highest detection efficiency. The detection time cost of the schemes in [7], [9] is almost the same as the verification computation cost in Fig. 7. It is because that the signature tampering can not be detected until the whole verify process is completed. In our scheme, the signature is always companied with the corresponding signed hash value, which can help us to detect the signature tampering efficiently by Algorithm 3.

All the experimental results meet the theoretical analysis in Table 3.

8 Conclusion

Since there are multiple data providers and a wide range of users in cloud service systems, it is hard to take full advantage of cloud data to serve people well on the premise of not infringing upon the interests of others. In this paper, it is the first time to propose a cooperative query answer authentication scheme which applies to cloud. This scheme can not only verify the trustiness, completeness, authenticity of the query answers efficiently, but also satisfy DO's requirement for anonymity and guarantee non-repudiation service between CSP and user. Firstly, the proposed scheme

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

TABLE 3
Comparison and analysis

Scheme	Authenticity and completeness	High efficiency	Multi-DO supporting and DO's anonymity	Query non-repudiablility	Sign computation cost	Verification computation cost	VO size
[9]	×	×	√	×	$O(N)$	$O(l)$	$O(l)$
[7]	√	×	×	×	$O(log_2 N)$	$O(log_2 N)$	$O(log_2 N) + 2log_2 N$
Ours	√	√	√	√	$O\left(\log_2 N + \sqrt{N}\right)$	$O\left(\log_2 \sqrt{N}\right)$ or $O(\log_2 N)$	$O\left(\log_2 \sqrt{N}\right) + 2\log_2 N$ or $O(\log_2 N) + 2\log_2 N$

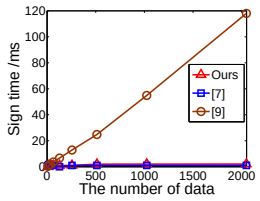

Fig. 5. Sign computation cost.

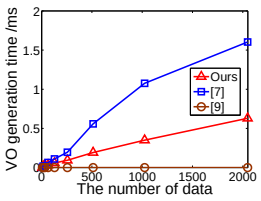

Fig. 6. VO generation cost.

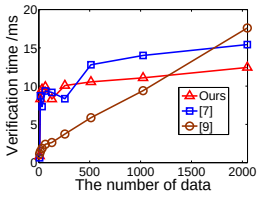

Fig. 7. Verification computation cost.

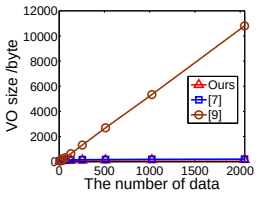

Fig. 8. VO size.

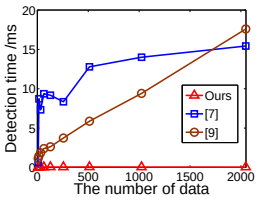

Fig. 9. Detection time cost for signature tampering.

chooses and signs the KN in the MHT based on the ring signature scheme, which can both verify the correct of query result when keeping DO anonymous, and supports multiple DOs. Secondly, we introduce a non-repudiation protocol based on VO to solve the repudiable behaviors of CSP and user. Finally, the experimental results show our proposed scheme is of higher efficiency and lower communication cost than others.

ACKNOWLEDGMENTS

The work was supported by the National Natural Science Foundation of China (61472001, 61472283, U1405255), and the scientific research construction fee of Anhui University.

REFERENCES

[1] D. Kwak, R. Liu, D. Kim, B. Nath, and L. Iftode, "Seeing is believing: Sharing real-time visual traffic information via vehicular clouds," *IEEE Access*, vol. 4, pp. 3617–3631, 2016.

[2] Q. Yang, B. Zhu, and S. Wu, "An architecture of cloud-assisted information dissemination in vehicular networks," *IEEE Access*, vol. 4, pp. 2764–2770, 2016.

[3] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on parallel and distributed systems*, vol. 25, no. 1, pp. 222–233, 2014.

[4] W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1566–1577, 2016.

[5] S. Tian, Y. Cai, and Z. Hu, "A parity-based data outsourcing model for query authentication and correction," in *IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2016, pp. 395–404.

[6] J. Li, A. Squicciarini, D. Lin, S. Sundareswaran, and C. Jia, "Mmbcloud-tree: Authenticated index for verifiable cloud service selection," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–14, 2015.

[7] H. Pang and K.-L. Tan, "Query answer authentication," *Synthesis Lectures on Data Management*, vol. 4, no. 2, pp. 1–103, 2012.

[8] F. Li, K. Yi, M. Hadjieleftheriou, and G. Kollios, "Proof-infused streams: Enabling authentication of sliding window queries on streams," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 147–158.

[9] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 157–165, 2014.

[10] R. C. Merkle, "A certified digital signature," in *Conference on the Theory and Application of Cryptology*. Springer, 1989, pp. 218–238.

[11] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic authenticated index structures for outsourced databases," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 121–132.

[12] ——, "Authenticated index structures for aggregation queries," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 4, p. 32, 2010.

[13] F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera, "Range query integrity in cloud data streams with efficient insertion," in *International Conference on Cryptology and Network Security*. Springer, 2016, pp. 719–724.

[14] Q. Chen, H. Hu, and J. Xu, "Authenticated online data integration services," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 167–181.

[15] R. Vyas, A. Singh, J. Singh, G. Soni, and B. Purushothama, "Design of an efficient verification scheme for correctness of outsourced computations in cloud computing," in *International Symposium on Security in Computing and Communication*. Springer, 2015, pp. 66–77.

[16] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," Technical report, SRI International, Tech. Rep., 1998.

[17] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[18] ——, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[19] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[20] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 106–115.

[21] C. C. Aggarwal, "On unifying privacy and uncertain data models," in *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008, pp. 386–395.

[22] X. M. Ren, J. Yang, J. P. Zhang, and Z. F. Jia, "Uncertain data privacy protection based on k-anonymity via anatomy," in *Advanced Engineering Forum*, vol. 6. Trans Tech Publ, 2012, pp. 64–69.

[23] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving anonymity via clustering,"

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

in *Proceedings of the 25 ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2006, pp. 153–162.

[24] J.-L. Lin, T.-H. Wen, J.-C. Hsieh, and P.-C. Chang, "Density-based microaggregation for statistical disclosure control," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3256–3263, 2010.

[25] X. Xiao and Y. Tao, "Anatomy: Simple and effective privacy preservation," in *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 139–150.

[26] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate query answering on anonymized tables," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 116–125.

[27] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2001, pp. 552–565.

[28] S. S. Chow, S.-M. Yiu, and L. C. Hui, "Efficient identity based ring signature," in *International Conference on Applied Cryptography and Network Security*. Springer, 2005, pp. 499–512.

[29] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup, "Anonymous identification in ad hoc groups," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2004, pp. 609–626.

[30] F. Zhang and K. Kim, "Id-based blind signature and ring signature from pairings," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2002, pp. 533–547.

[31] X. Huang, J. K. Liu, S. Tang, Y. Xiang, K. Liang, L. Xu, and J. Zhou, "Cost-effective authentic and anonymous data sharing with forward security," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 971–983, 2015.

[32] X. Yang, W. Wu, J. K. Liu, and X. Chen, "Lightweight anonymous authentication for ad hoc group: A ring signature approach," in *International Conference on Provable Security*. Springer, 2015, pp. 215–226.

[33] P. P. Tsang, M. H. Au, J. K. Liu, W. Susilo, and D. S. Wong, "A suite of non-pairing id-based threshold ring signature schemes with different levels of anonymity," in *International Conference on Provable Security*. Springer, 2010, pp. 166–183.

[34] E. Bresson, J. Stern, and M. Szydlo, "Threshold ring signatures and applications to ad-hoc groups," in *Annual International Cryptology Conference*. Springer, 2002, pp. 465–480.

[35] C. A. Melchor, P.-L. Cayrel, P. Gaborit, and F. Laguillaumie, "A new efficient threshold ring signature scheme based on coding theory," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4833–4842, 2011.

[36] T. H. Yuen, J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Efficient linkable and/or threshold ring signature without random oracles," *The Computer Journal*, vol. 56, no. 4, pp. 407–421, 2013.

[37] H. Gan and L. Chen, "An efficient data integrity verification and fault-tolerant scheme," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*. IEEE, 2014, pp. 1157–1160.

[38] 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 Transactions on Computers*, vol. 64, no. 9, pp. 2609–2622, 2015.

[39] D. Wu, B. Choi, J. Xu, and C. S. Jensen, "Authentication of moving top-k spatial keyword queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 922–935, 2015.

[40] J. Zhou and D. Gollmann, "An efficient non-repudiation protocol," in *Computer Security Foundations Workshop, 1997. Proceedings., 10th*. IEEE, 1997, pp. 126–132.

[41] ——, "A fair non-repudiation protocol," in *IEEE symposium on security and privacy*. Citeseer, 1996, pp. 55–61.

[42] H. U. Yildiz, K. Bicakci, B. Tavli, H. Gultekin, and D. Incebacak, "Maximizing wireless sensor network lifetime by communication/computation energy optimization of non-repudiation security service: Node level versus network level strategies," *Ad Hoc Networks*, vol. 37, pp. 301–323, 2016.

[43] J. Li, H. Lu, and M. Guizani, "Acpn: a novel authentication framework with conditional privacy-preservation and non-repudiation for vanets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938–948, 2015.

[44] C.-Y. Wu, Y. Xiong, W.-C. Huang, Q.-W. Lu, and X.-D. Gong, "A trusted fair non-repudiation protocol based on dynamic third party in mobile ad hoc networks," *Acta Electronica Sinica*, vol. 2, p. 004, 2013.

[45] J. Feng, Y. Chen, D. Summerville, W.-S. Ku, and Z. Su, "Enhancing cloud storage security against roll-back attacks with a new fair multi-party non-repudiation protocol," in *2011 IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2011, pp. 521–522.

[46] J. Feng, Y. Chen, W.-S. Ku, and P. Liu, "Analysis of integrity vulnerabilities and a non-repudiation protocol for cloud data storage platforms," in *2010 39th International Conference on Parallel Processing Workshops*. IEEE, 2010, pp. 251–258.

[47] J.-Z. Luo, Z.-G. Han, and L.-m. Wang, "Trustworthy and controllable network architecture and protocol framework," *Chinese Journal of Computers*, vol. 32, no. 3, pp. 391–404, 2009.

[48] Z.-g. HAN and J.-z. LUO, "Analysis and improvement of timeliness of a multi-party non-repudiation protocol [j]," *Acta Electronica Sinica*, vol. 2, p. 025, 2009.

[49] S. Kremer, O. Markowitch, and J. Zhou, "An intensive survey of fair non-repudiation protocols," *Computer communications*, vol. 25, no. 17, pp. 1606–1621, 2002.

[50] T. Tedrick, "How to exchange half a bit," in *Advances in Cryptology*. Springer, 1984, pp. 147–151.

[51] ——, "Fair exchange of secrets," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1984, pp. 434–438.

[52] Q. Wang, M. He, M. Du, S. S. Chow, R. W. Lai, and Q. Zou, "Searchable encryption over feature-rich data," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[53] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 18, 2015.

[54] K. Li, W. Zhang, C. Yang, and N. Yu, "Security analysis on one-to-many order preserving encryption-based cloud data search," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1918–1926, 2015.

[55] Z. Liu, X. Chen, J. Yang, C. Jia, and I. You, "New order preserving encryption model for outsourced databases in cloud environments," *Journal of Network and Computer Applications*, vol. 59, pp. 198–207, 2016.

[56] X.-Y. Li and T. Jung, "Search me if you can: privacy-preserving location query service," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2760–2768.

[57] T. Jung, X.-Y. Li, and M. Wan, "Collusion-tolerable privacy-preserving sum and product calculation without secure channel," *IEEE Transactions on Dependable and secure computing*, vol. 12, no. 1, pp. 45–57, 2015.

[58] T. Jung, X.-Y. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 190–199, 2015.

Liangmin Wang received his B. S. degree in Computational Mathematics in Jilin University, Changchun, China, in 1999, and the Ph.D degree in Cryptology from Xidian University, Xi'an, China, in 2007. He is a full professor in the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. He has been honored as a "Wan-Jiang Scholar" of Anhui Province since Nov. 2013. Now his research interests include security protocols for wireless networks and Privacy & Security of Big Data.

He has published over 60 technical papers at premium international journals and conferences, like IEEE Transactions on Information Forensics and Security, IEEE Transactions on Vehicular Technology, IEEE GlobeCOM, IEEE WCNC. Dr WANG has been severed as the TPC of many IEEE conferences, such as IEEE ICC, IEEE HPCC, IEEE TrustCOM. Now he is an associate editor of Security and Communication Networks, a member of IEEE, ACM, and a senior member of Chinese Computer Federation.

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2676008, IEEE Access

Qingqing Xie received the B. Eng. degree in computer science and technology from Anhui University, PRC in 2012. She is currently a Ph.D. candidate in Department of Computer Science and Technology at Anhui University. She is also a research visitor in Department of Computer Science at Boise State University.

Her research interest includes data security, cloud computing, and applied cryptography.

Hong Zhong received her B. S. degree in applied mathematics in Anhui University, China, in 1986, and the Ph.D degree in computer science and technology from University of Science and Technology of China (USTC), China, in 2005. Now she is a professor and Phd Advisor of Anhui University.

Her research interests include security protocols and wireless sensor networks.

2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.