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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

Service Outsourcing in F2C Architecture with Attribute-based Anonymous Access Control and Bounded Service Number

Jianan Hong, Kaiping Xue, *Senior Member, IEEE,* Na Gai, David S.L. Wei, Peilin Hong

Abstract—F2C (fog-to-cloud) enables service providers to rent the low-cost cloud/fog resources to publish their services, and the fog nodes, which are deployed at the edge, can provide short-latency service to users. However, new security threats come along with this new computing paradigm, where the access control and trusted payment are concerned in this work. We propose a privacy-preserving authentication scheme. By integrating k-times anonymous authentication (k-TAA) and attribute-based access control, in our proposed scheme, service providers can autonomously determine a fine-grained access policy and the maximal access times for authorized users. Thus, users who satisfy the access policy can receive benefits of this service for certain number of times without leaking any private information. Our authentication phase has a low-latency because it is offloaded to the fog as what the service does. This paper presents a lightweight and trusted billing mechanism using Merkle Hash Tree (MHT), which can detect the cloud's service forgery with high probability, without costing too much of service provider's bandwidth and computation. Rigorous security analysis proves that the proposed scheme is secure against malicious users, fogs, and cloud, and the experimental results show the significant performance advantage on both the delay reduction and service providers' cost saving.

Index Terms—Fog-to-Cloud architecture, attribute-based access control, privacy preserving authentication, merkle-hash tree.

1 Introduction

With fast advances in cloud computing technologies [1], it is forseeable that there will be an explosive demand of various service outsourcing paradigms in the near future. The concept of service outsourcing is popular due to the technical demand of service functionality virtualization [2] and the relief of clients' burden from facility maintenance with a low-cost payment according to pay-as-you-use style [3]. In this service paradigm, service providers rent the cloud server to outsource their services to the public users, and thus no need to worry about the complicated infrastructure allocation and management [4]. Among the services to be outsourced, latency-sensitive ones and location-based ones (e.g., vehicular service, advertisement delivery, online games) have drawn vast attention from a large number of researchers and practitioners. However, a centralized cloud architecture cannot undertake these outsourced services due to its unbearable transmission delay. Fog computing [5] (or named edge computing [6]) has been proposed as a supplement to cloud computing by offering a mobility-support, location-aware and low-latency platform. The Fog-to-Cloud (F2C) architecture is further studied to integrate their advantages [7], whose access control issue has attracted many attentions. Yi *et al.* [8] have raised the challenges on how to enforce access control with these fog nodes in their survey, but have not proposed relevant mechanisms. It becomes a problem on how to make full use of the real-time processing brought in by the fogs, and preserve the security features as well [9].

This paper studies a service outsourcing system based on F2C architecture, where the resource-limited service providers (SP) outsource their services to the cloud, and the cloud selects some fog nodes to undertake the actual services. The cloud, as a powerful platform, masters the detailed distribution of each fog node, thus, a more feasible fog selection strategy can be made by the cloud, rather than SPs themselves, so as to obtain better economic utility according to contract theory [10]. Therefore, F2C shows its significant advantage, as a service outsourcing paradigm, over exisiting paradigms. Through this architecture, traditional cloud providers will be empowered to extend their reach closer to the edge, and at the same time fog devices will be able to offer more capacities than currently envisioned [7].

However, this compelling paradigm comes with its inherent security concerns, including the realization of access control and trusted payment. Obviously, as a commercial service platform, the cloud and fog providers cannot be fully trusted (semi-trust), meaning that, they may pretend to obey the contract for their reputation, but try to enlarge their benefits if the cheating behavior will not be discovered. Since the services are practically served by the semi-trust platforms or enterprises, traditional access control mechanism cannot work. In a pay-as-you-use billing model, an SP should not pay for the services to unauthorized users. Therefore, a big challenge comes to us: how to make sure that every account paid by SPs is really a service for authorized users?

For the above issues, authentications based on Kerberos protocol [11] or PKI (public key infrastructure) [12] can be

J. Hong, K. Xue, N. Gai and P. Hong are with Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China (Email: kpxue@ustc.edu.cn).

D. Wei is with the Computer and Information Science Department, Fordham University, New York, NY 10458, USA.

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

treated as the simple solutions, whereas, the former brings in complex interactions between SPs and users, and the latter cannot protect users' privacy. Since the fogs, the clouds, and most service providers are commercial enterprises, it becomes a serious concern that they will collect and reveal users' private information for their benefits. Thus, the protection of user's private information, including identity and attributes, should also be taken into account.

The work presented in this paper aims at proposing a fine-grained and anonymous access control scheme (based on *Attribute-based Encryption* to be discussed in Section 2) for the service outsourcing system, with a trusted billing mechanism. However, how to bound the number of access times for each authorized user is also an important issue needed to be addressed in outsourced services. In most practical systems, the number of service access times should be bounded for some good reasons: 1) most applications in our consideration are with bounded times, e.g., coupons; 2) Services with unlimited number of times do not suit pay-as-you-use model, since SPs undertake the payments, and it will inevitably lead to collusion attacks between fogs/cloud and users.

By integrating k-times anonymous authentication (k-TAA), proposed by Teranishi *et al.* [13] and attribute-based access control mechanisms, in this paper, we propose a secure service outsourcing scheme. In our scheme, SPs can autonomously determine access policy and the maximal access times for those authorized users. The service and verifications of authorized users are both allocated to fog nodes such that real-time property is ensured. On the one hand, unauthorized access or unbounded authentications can be detected or traced by the fogs and trusted third parties. On the other hand, the verifier, without the system's secret parameters, can only judge whether an access is authorized, but additional private information, e.g., the user's detailed attribute set and identity, will be protected. Moreover, considering the constraints of SPs' communication and computing resource, a lightweight and trust billing mechanism is proposed to resist cloud's forgeries: With the utilization of merkle hash tree, SP only needs to receive and verify a very small amount of services, but still can prevent cloud from forging the service amount due to the utility concern of the cloud in the detection phase. The main contributions of this work are summarized as follows:

1) To the best of our knowledge, this work is the first one that proposes fine-grained access control for outsourced service. With the combination of attributed-based signature (ABS) and k-TAA, the service providers can autonomously determine which users to be served and the maximal number of times of service. The unsatisfied service request should be denied. Otherwise, the fog or cloud will not receive any payment from this service.
2) Using non-interactive authentication, we propose an effective and trusted payment mechanism. This mechanism efficiently outsources the verification procedure and the services.
3) Through the implementation of merkle hash tree, the payment interaction can proceed without consuming much of SP's communication and computation cost, but still prevents the cloud's cheating behavior effectively. A quantified analysis is presented in this paper to discuss this tradeoff between performance and security.

The rest of this paper is organized as follows: Section 2 briefly discusses the related work. Then we introduce necessary preliminaries in Section 3. In Section 4, we present the system and security model. Detailed construction is presented in Section 5. Then, the security analysis and the experimental results are shown in Section 6 and 7, respectively. Finally, in Section 8, we conclude this paper.

2 Related Works

In the area of computing outsourcing, security issues have been extensively studied by lots of researchers. Among these researches, the notion of secure/verifiable computing, e.g., [14–16], makes the cloud or fog return a desired computed result, but leaking as least privacy to these entities as possible. From the privacy perspective, clients' access pattern is also concerned in this area [17, 18], which helps a client to get required data, but the storage provider has no knowledge on which data block has been requested. Access control is another critical issue in this area, since the environment of semi-trust providers makes it more difficult to resolve. Faced with storage applications, attribute-based encryption (ABE) [19] provides a fine-grained access mechanism, such as [20–24]. Since the access control is realized based on user's own decryption capability, only the user whose attribute set satisfies a designed policy can decrypt the data. However, this method is not suitable for the case when service objects are not data (e.g., navigation, recommendation system, commodity coupons).

To address this issue, the technique of attribute-based signature (ABS) and anonymous credential (AC) provides relevant solutions. ABS [25, 26] provides fine-grained access control for such authentication system. A user in ABS signs a message to prove his/her attributes satisfy an access policy, but without revealing the identity. Our work follows the idea of Maji *et al.*'s work [25] rather than Li *et al.*'s [26], as the algorithm in the latter one is an encryption-based one. Anonymous Credential [27] also provides similar techniques. With such credential, a prover can convince the verifier that he/she is permitted to access a service, but without leaking any other secrets in the credential, especially the user's identity. Compared with ABS, AC lacks enough expression of the attributes for practical usage; but it has its own advantage, and the most attractive one is that a private range assertion is realized based on zero knowledge proof. Some attribute-based encryption schemes, such as [22], also take it into account.

However users' service accesses cannot be bounded and accounted with ABS algorithm alone. Thus it does not suit the *pay-as-you-use* billing model of service outsourcing paradigms. Teranishi *et al.* [13] proposed k-times anonymous authentication (k-TAA) that addresses unbounded-accessing problem. In k-TAA, the provider can arbitrarily determine the maximum number of times an individual user can authenticate anonymously. An authentication with exceeded number of attempts will be detected, or even be traced. Nguyen *et al.* [28] propose a k-TAA scheme supporting user dynamic granting and revocation. Au *et al.* further

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

improved the unlinkability protection in [29], whereas the interactive messages are enlarged.

Yuen *et al.* [4] firstly combined the ideas of attribute-based with k-TAA, and provided a scheme, where the authorized user can anonymously prove that his/her attributes satisfy an access policy within bounded times. However, some important issues inherited from [13] still have not yet been addressed for the following reasons: Firstly, recent k-TAA schemes require an interactive zero-knowledge proof between a verifier and a prover. As it is not a non-repudiation authentication, the verifier can simulate a valid *prove-verify* interaction. This indicates that accountability and trusted billing cannot be realized, and verification outsourcing cannot be achieved. In fact, the threat from cloud's forgery for service logs are critical, and has been considered in some researches, such as [3]. Secondly, the whole system can only undertake one service/application in these schemes, which is not preferred in practical service outsourcing systems. Thirdly, the bounded access attempts is realized due to a centralized verifier to cache an authentication log, but how to apply it to a distributed system (e.g., fog computing) is still a big challenge.

3 Preliminaries

3.1 Attribute-based Access Structure

In this paper, every user is associated with a set of attributes, specifying his/her characters in this system. For a certain service, to judge whether a user can access it is according to an access structure. For instance, a structure "$Att_1 \wedge (Att_2 \vee Att_3)$" can be accessed by users owning attributes Att_1 and Att_2, or users with Att_1 and Att_3.

This paper follows [30] to express the access structure as an LSSS (*linear secret sharing scheme*) format (M, ρ). In this format, M is an $l \times n$ matrix, and $\rho(i), i \in [1, l]$ is the attribute that labels the i^{th} row of M. Let $I \subset [1, l]$, if and only if $\{\rho(i) : i \in I\}$ is an attribute set that satisfies the access structure (M, ρ), there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that:

$$\sum_{i \in I} \omega_i M_{ij} = \begin{cases} 1 & \text{if } j = 1, \\ 0 & \textit{otherwise}. \end{cases} \quad (1)$$

Using the property of Eq. (1), we will provide a privacy-preserving solution, and such solution enables a verifier to verify that a user's attribute set satisfies a required access policy without leaking any more information on the user's attribute set.

3.2 Bilinear Pairings

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of the same prime order p. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map holding the following properties [31]:

1) *Bilinearity.* For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
2) *Non-degeneracy.* $e(g, g) \neq 1_{\mathbb{G}_T}$, where $g \in \mathbb{G}$.
3) *Computability.* There is an efficient algorithm to compute $e(u, v)$ for all $u, v \in \mathbb{G}$.

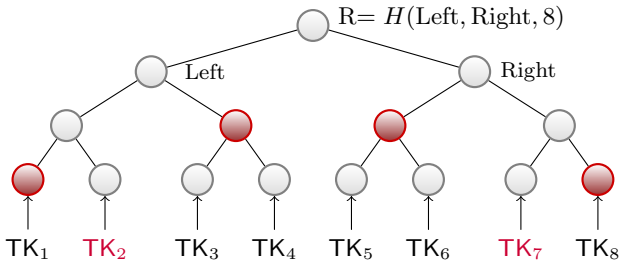


Fig. 1. Merkle Hash Tree Authentication of 8 TKs. (The checked elements are with the format TK_i to echo later contexts)

3.3 Merkle Hash Tree

Merkle Hash Tree (MHT) [32] is an effective authentication mechanism, which was firstly introduced to efficiently and securely prove that a set of elements are undamaged and unaltered [33]. As shown in Fig. 1, the construction is a binary tree, where the leaves are the hashes of authentic element values, and for a non-leaf node a, its value is as:

$$v_a = H(v_L, v_R, \mathsf{NUM}_a),$$

where v_L and v_R are the values of its left and right child respectively, and NUM_a is the number of its dominated leaf nodes. We can see that $\mathsf{NUM}_a = \mathsf{NUM}_L + \mathsf{NUM}_R$.

A prover in Fig. 1 submits R as a commitment and claims that the element number is 8. A verifier checks the validation of $\{\mathsf{TK}_2, \mathsf{TK}_7\}$. Besides these two elements, other nodes are required as the auxiliary information $\Omega(2, 7)$. We define $\Omega(\cdot)$ as follows:

Definition 1. *Authentication Information* $\Omega(\{i | i \in \mathcal{S}_{ind}\})$. It is a set of nodes of MHT generated as follows: 1) Firstly, for each node on the path of R to $i \in \mathcal{S}_{ind}$, we collect its sibling nodes in Ω; 2) If an element in Ω is through the path of R to any $i \in \mathcal{S}_{ind}$, then remove it.

To revisit Fig. 1 helps explain it more clearly. The red nodes constitute the set $\Omega(2, 7)$. Also, the relevant NUMs are also counted as Ω's components.

Upon the reception of these responses, the verifier is not only able to check the validation of TK_2 and TK_7, but also to detect whether they are in the 2^{nd} and 7^{th} positions of the commitment. In our detailed introduction later, we will explain the mechanism more thoroughly.

3.4 Zero-Knowledge Proof

A zero-knowledge proof (ZKP) [34] is a cryptographic method, by which, one party (as the prover) can prove to another party (as the verifier) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true. Formally, a ZKP should satisfy three properties:

- Completeness: if the statement is true, the honest verifier will be convinced of this fact by an honest prover.
- Soundness: if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

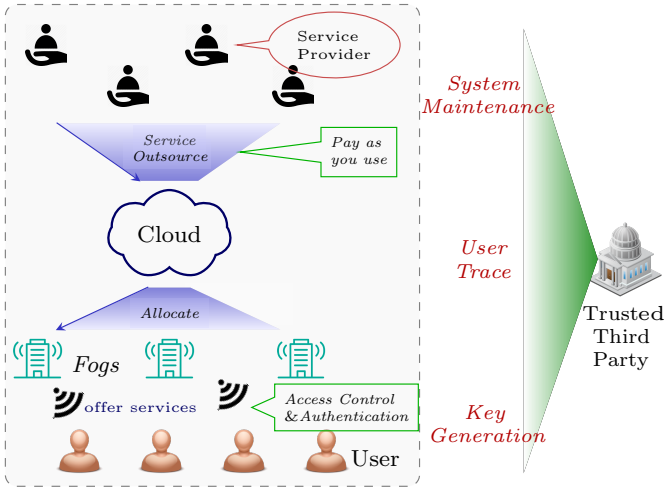


Fig. 2. Service Outsourcing System based on F2C Architecture

- Zero-knowledge: no verifier learns anything other than the fact that the statement is true. This is formalized by showing that every verifier has some simulator that, given only the statement to be proved (and no access to the prover), can produce a transcript that "looks like" a valid interaction.

This paper uses Eq. (2) to define a ZKP transcript.

$$ZKP\{\{x_1,\ldots,x_n\}:\{x_1,\ldots,x_n\}\in\mathcal{L}\}, \tag{2}$$

where the variables $\{x_1,\ldots,x_n\}$ are the secret witnesses that cannot be leaked, while $\mathcal{L}$ denotes their relationship in a certain statement. For instance, in a ZKP for *Discrete logarithm* ($ZKP\{x:x\in\mathcal{L}\}$), $\mathcal{L}$ means that x is a logarithm of a commited public parameter.

4 System Model

As shown in Fig. 2, our system consists of several services providers (SP), a *cloud* server, *fog* nodes and users, and the trusted parties to manage the security. Especially, two parities play the role of the system manager: a trusted third party (TTP) and an attribute authority (AA). Their roles and security assumptions are described in what follows.

4.1 Architecture

In this architecture, TTP initializes the whole system, publishes relevant parameters and deals with user registrations; AA classifies the users by their attributes, and generates secret keys for each user according to his/her attribute set. In addition, TTP executes malicious user tracing if there is a need. We allow TTP and AA to be a unified entity, but our design also works for more practical situations.

The *cloud* server provides a strong service platform for other entities. As the core component of this service paradigm, the cloud server is responsible for the outsourced services management, advertising, and allocation. *Cloud* is assumed to get detailed information about the distribution of *fog* nodes and their service characters so as to be easier to make a feasible decision. *Fogs* are located at the edge of networks, and offer real-time or location-based services for mobile users.

The SPs publish services with bounded access times for intended users according to users' attribute sets. For each outsourced service, the SP designates an access policy, expressing who can receive benefits of the service and how many times an authorized user can access this service.

The payment occurs between SPs and *Cloud*, and between *Cloud* and *Fogs*, both of which follow the pay-as-you-use style. It is worth noting that the user does not need to pay the bill if his/her service access times have not exceeded SP's pre-defined value. Naturally, users can also receive benefits of the services and personally pay for the exceeded access times, which is beyond the scope of this paper.

4.2 Trust Model and Security Requirements

In our scheme, TTP and AA, as the core management entities, can be totally trusted. The TTP has the capability to trace users, but it will not leak its secret parameters to others to disclose honest users' privacy. As a key generation entity, AA can be trusted to issue user's secret key according to his/her attribute set strictly.

Cloud and fogs are semi-trusted, who concerns about their reputations and financial benefits critically. On the one hand, for their own reputation, cloud and fogs will provide clients with their best service to attract more potential customers. On the other hand, as rational parties, to enlarge the utility, they will make some forgeries if it can only be detected with negligible probabilities. In addition, their curiosity about users' privacy also brings in challenging security threat.

Unauthorized users are considered as malicious ones who try to access the services with any feasible means. Hence, the following security aspects are considered in this paper:

- *Access Control.* Only authorized users can successfully request services for limited times. Unauthorized service requests can be detected or traced.
- *Privacy Preserving.* An honest authentication should leak no private information, including user's identity and attributes.
- *Resist Fog's Forgery.* Fog's malicious forgery cannot succeed with non-negligible probability.
- *Resist Cloud's Forgery.* Cloud's malicious forgery can be detected with a sufficiently high probability, such that, the threat can be prevented due to cloud's rationality [14].

5 The Proposed Scheme

This scheme generally consists of the following six phases: *initialization*, *user registration*, *service publishing*, *authentication*, *billing*, and *tracing*. At the beginning of Construction Description, we firstly present an overall idea of our system, and then the detailed scheme will be introduced.

5.1 Overview

The two aspects of security requirements for SPs are: 1) only their intended users can access the services, and 2) each users' access times should be constrained. In our scheme, SP itself can determine the service's access policy, which not

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

only determines who can receive benefits of the service, but also limits how many times they can receive the service. These two aspects are realized by the following ideas.

- To tell whether a user can access is by an *attribute-based* access structure. A variant of attribute-based signature [25, 35] is used to distinguish whether the user's attribute set satisfies the structure without revealing other information on this set.
- Limited access is controlled with the idea of k-time anonymous authentication (k-TAA) [13]. In k-TAA, k_{max} is defined as the maximal number of times an individual user can authenticate.

To access a service, user's submitted request message carries the authentication information, which is a combination of ABS and k-TAA, but with a non-interactive manner. The fogs immediately verify the user (without knowing user's private information) and provide the relevant service. Fogs can gain benefits for this service from the cloud according to the service amount it provided, which is calculated by its selected user's authentication messages.

A user tracing mechanism is needed since distributed fogs play the role of verifiers. However, the verifiers do not control the entire verification log. Thus, a tracing mechanism helps punish users who use duplicated k on different fogs.

Besides, to resist cloud's forgery for its benefits, a sampling method based on *Merkle hash tree* is used. With this method, the SPs only undertake quite little communication and computation cost to verify quite a large amount of the outsourced service. The cloud will take unbearable risks or get negligible benefits when faking its service amount, and thus our scheme achieves security and efficiency simultaneously.

5.2 Initialization

To jointly manage and monitor the system, the TTP and AA respectively conduct initial operations to setup some important parameters. It is a two-step phase as follows:

5.2.1 TTP Setup

Let N be the size of system's universe attribute set. Let $\mathbb{G}, \mathbb{G}_T$ be two multiplicative groups with the same prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be the bilinear map. Let $H_1 : (0,1)^* \to \mathbb{Z}_p^*$ and $H_2 : (0,1)^* \to \mathbb{G}^*$ be two hash functions.

The TTP randomly selects five generators $g, \hat{g}, h, \tilde{g}, \tilde{h} \in \mathbb{G}$, and $N+1$ different secrets $\delta_i \in \mathbb{Z}_p^*, i \in [0, N]$. Especially, δ_0 is the secret key for integrity protection. The public parameter of TTP is as:

$$\{g, \hat{g}, h, \tilde{g}, \tilde{h}, h_0 = h^{\delta_0}; \forall i \in [1, N] : h_i = h^{\delta_i}, \hat{h_i} = h^{1/\delta_i}\}.$$

5.2.2 Authority Setup

After obtaining TTP's public parameters, the AA sets its secret key $(\alpha, a) \in \mathbb{Z}_p$. To organize its managed attribute set with at most n dimensions, AA publishes a table to associate $[1, N]$ with the attribute set as Table 1.

Despite Table 1, AA sets its public parameters as follows:

$$\{g^{\alpha}; \forall i \in [1, N] : PK_i = \hat{h_i}^{a}\}.$$

TABLE 1
Public Attribute Table

Attribute Index	Attribute Name
1	Att_1
2	Att_2
$\vdots$	$\vdots$
m	Att_m
$> m$	Reserved

5.3 User Registration and Secret Key Generation

For a user to be registered (denoted as U_j), TTP firstly chooses user's key pair $(y_j \in \mathbb{Z}_p, Y_j = \hat{g}^{y_j})$, and signs Y together with user's identity ID_j as his/her certification. The format is as:

$$cert_j = \{ID_j, Y_j, \sigma = H_2(ID_j || Y_j)^{\delta_0}\}.$$

Then, TTP securely sends y_j and $cert_j$ to the user, and stores y_j, ID_j to its user list.

With $cert_j$, U_j turns to AA to request the secret key for his/her attribute set (say $\mathcal{S}_j \subset [1, n]$). After verifying $cert_j$ by checking $e(\sigma, h) \stackrel{?}{=} e(H_2(ID_j || Y_j), h_0)$, AA selects a random $t \in \mathbb{Z}_p^*$ and generates his/her attribute associated key as:

$$SK_j = \{K = (Y_j g)^{\frac{1}{\alpha + at}}, L_1 = g^{at}, L_2 = h^{at}, \\ \forall i \in \mathcal{S}_j : K_i = h_i^t\}.$$

Additionally, y_j is also set as part of the secret key.

5.4 Service Publishing with Policy Determination

To publish a service to the system, a service provider SP firstly requests a unique identity ID_{ser} from the cloud. An access policy is designed with the format as $< (M, \rho), k_{\max} >$. In this format, (M, ρ) expresses who are authorized to receive benefits of the service, as has been described in Section 3.1, and $k_{\max}$ represents the service limitation for each authorized user.

For a service with ID_{ser}, the task of its resource allocation, management, and advertising is outsourced to the cloud. According to service's specified property (e.g., latency, coverage, location preference, required facilities), the cloud will choose a group of fog nodes to undertake the actual service provision. All the selected fogs have the knowledge of service's access policy $< (M, \rho), k_{\max} >$ and its identity.

5.5 User Authentication and Service Providing

Based on the access policy $< (M, \rho), k_{\max} >$ of a required service, a user U_j should generate a credential to convince the fog (with identity ID_{fog}) that he/she has the privilege to receive benefits of the service. The procedure is as follows:

Firstly, an integer k is randomly selected, holding two constraints: 1) $1 \leq k \leq k_{\max}$; 2) it has not been used for ID_{ser} by U_j. An authentication token is firstly generated as follows:

$$\mathsf{TK} = (k, C = e(g, g)^{\frac{1}{y + H(ID_{ser} || k)}}).$$

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

From $\mathbb{Z}_p^*$ randomly selected:

$\mu_K, \nu_K, \mu_{L_1}, \mu_{L_2}, \forall i \in [1, L] : \mu_i;$

$\gamma_y, \gamma_{\beta_1}, \gamma_{\beta_2}, \gamma_{\mu_K}, \gamma_{\nu_K}, \gamma_{\mu_{L_1}}, \gamma_{\mu_{L_2}}, \forall i \in [1, L] : \gamma_{\mu_i}$

Compute:

$A_K = K\tilde{g}^{\mu_K}$ $\quad T_1 = \tilde{g}^{\gamma_{\nu_K}} \tilde{h}^{\gamma_{\mu_K}}$ $\quad R_1 = e(\tilde{g}, h)^{-\gamma_{\mu_{L_2}}} \prod_{i=1}^{l} e(\tilde{g}, \hat{h}_{\rho(i)}^{aM_{i1}})^{\gamma_{\mu_i}}$

$B_K = \tilde{g}^{\nu_K} \tilde{h}^{\nu_K}$ $\quad T_2 = B_K^{-\gamma_{\mu_{L_1}}} \tilde{g}^{\gamma_{\beta_2}} \tilde{h}^{\gamma_{\beta_1}}$ $\quad \forall j \in [2, n] :$

$A_{L_1} = L_1 \tilde{g}^{\mu_{L_1}}$ $\quad T_3 = e(\tilde{g}, g^{\alpha} A_{L_1})^{\gamma_{\mu_K}} e(A_K, \tilde{g})^{\gamma_{\mu_{L_1}}}$ $\quad R_j = \prod_{i=1}^{l} e(\tilde{g}, \hat{h}_{\rho(i)}^{aM_{ij}})^{\gamma_{\mu_i}}$

$A_{L_2} = L_2 \tilde{g}^{\mu_{L_2}}$ $\quad \cdot e(\tilde{g}, \tilde{g})^{-\gamma_{\beta_1}} e(\tilde{g}, g)^{\gamma_y}$

$\forall i \in [1, l] :$ $\quad T_4 = e(h, \tilde{g})^{\gamma_{\mu_{L_1}}} e(\tilde{g}, g)^{-\gamma_{\mu_{L_2}}}$

$A_i = K_{\rho(i)}^{\omega_i} \tilde{g}^{\mu_i}$ $\quad T_5 = C^{\gamma_y}$ $\quad \hat{c} = H_1(ID_{fog}; T_1, \ldots, T_5; R_1, \ldots, R_n)$

$\Rightarrow$

$r_y = \gamma_y - \hat{c}y$

$r_{\mu_K} = \gamma_{\mu_K} - \hat{c}\mu_K$

$r_{\nu_K} = \gamma_{\nu_K} - \hat{c}\nu_K$

$r_{\mu_{L_1}} = \gamma_{\mu_{L_1}} - \hat{c}\mu_{L_1}$

$r_{\mu_{L_2}} = \gamma_{\mu_{L_2}} - \hat{c}\mu_{L_2}$

$r_{\beta_1} = \gamma_{\beta_1} - \hat{c}\mu_K\mu_{L_1}$

$r_{\beta_2} = \gamma_{\beta_2} - \hat{c}\nu_K\nu_{L_1}$

$\forall i \in [1, l] :$

$r_{\mu_i} = \gamma_{\mu_i} - \hat{c}\mu_i$

Output:

$\sigma = (ID_{fog}, A_K, B_K, A_{L_1}, A_{L_2}, A_1, \ldots A_l, T_1, \ldots, T_5, R_1, \ldots R_n, r_y, r_{\mu_K}, r_{\nu_K}, r_{\mu_{L_1}}, r_{\mu_{L_2}}, r_{\beta_1}, r_{\beta_2}, r_{\mu_1}, \ldots, r_{\mu_l})$

Fig. 3. User Authentication for Service at A Fog Node with ID_{fog}

Note that y is to replace y_j in this phase for clarification.

If user's attribute set satisfies (M, ρ), then constants ω_i, for some i's, exist, and Eq. (1) holds. Here, $i \in [1, l]$ are integers, whose $\rho(i) \in \mathcal{S}_j$. As an extension to further preserve user's privacy, for those $\rho(i) \notin \mathcal{S}_j$, we can set $\omega_i = 0$, and the following equation still holds:

$$\sum_{i=1}^{l} \omega_i M_{ij} = \begin{cases} 1 & \text{if } j = 1, \\ 0 & \textit{otherwise}. \end{cases} \tag{3}$$

It is worth noting that $K_{\rho(i)}^{\omega_i} = 1$ when ω_i is set to zero. Thus a user can get $K_{\rho(i)}^{\omega_i}$ without the knowledge of $K_{\rho(i)}$.

With this extension, user authentication (denoted as σ) is generated as shown in Fig. 3. Intuitively, σ can be regarded as a non-interactive zero-knowledge proof as follows:

$$\begin{aligned} ZKP\{(y, K, L_1, L_2, \forall i \in [1, l] : K_{\rho(i)}^{\omega_i}) : \\ e(K, g^{\alpha} L_1) = e(\hat{g}^y g, g) \\ \wedge e(h, L_1) = e(L_2, g) \\ \wedge C = e(g, g)^{\frac{1}{y + H(ID_{ser}||k)}}) \\ \wedge \prod_{i=1}^{l} e(K_{\rho(i)}^{\omega_i}, \hat{h}_{\rho(i)}^{aM_{ij}}) = \begin{cases} e(L_2, h) & \text{if } j = 1, \\ 1 & \textit{otherwise}. \end{cases} \} \end{aligned} \tag{4}$$

Especially, the last line follows the feature in Eq. (3). The user submits (TK, σ) to fog node as his/her service request. In the remaining, the pair (TK, σ) is denoted as AUTH.

Before providing service, the fog node firstly checks (TK, σ) to verify whether this user has the privilege. At the beginning, it checks whether $k \in [1, k_{max}]$ and looks up its caching data to check whether there are no same TK: if not, the fog aborts the request; otherwise, it computes $\hat{c} = H_1(ID_{fog}; T_1, \ldots, T_5; R_1, \ldots, R_n)$ and then validates the following equations:

$$\begin{aligned} T_1 &\stackrel{?}{=} B_K^{\hat{c}} \tilde{g}^{r_{\nu_K}} \tilde{h}^{r_{\mu_k}}, \\ T_2 &\stackrel{?}{=} B_K^{-r_{\mu_{L_1}}} \tilde{g}^{r_{\beta_2}} \tilde{h}^{r^{\beta_1}}, \\ T_3 &\stackrel{?}{=} e(A_K, g^{\alpha} A_{L_1})^{\hat{c}} e(\tilde{g}, g^{\alpha} A_{L_1})^{r_{\mu_K}} e(A_K, \tilde{g})^{r_{\mu_{L_1}}} \\ &\quad \cdot e(\tilde{g}, \tilde{g})^{-r_{\beta_1}} e(\hat{g}, g)^{r_y}, \\ T_4 &\stackrel{?}{=} e(h, A_{L_1})^{\hat{c}} e(A_{L_2}, g)^{-\hat{c}} e(h, \tilde{g})^{r_{\mu_{L_1}}} e(\tilde{g}, g)^{-r_{\mu_{L_2}}}, \\ T_5 &\stackrel{?}{=} C^{-\hat{c} \cdot H(ID_{ser}||k) + r_y} e(g, g)^{\hat{c}}, \\ R_1 &\stackrel{?}{=} e(A_{L_2}, h)^{-\hat{c}} e(\tilde{g}, h)^{-r_{\mu_{L_2}}} \prod_{i=1}^{l} e(A_i^{\hat{c}} \tilde{g}^{r_{\mu_i}}, \hat{h}_{\rho(i)}^{aM_{i1}}), \\ R_j &\stackrel{?}{=} \prod_{i=1}^{l} e(A_i^{\hat{c}} \tilde{g}^{r_{\mu_i}}, \hat{h}_{\rho(i)}^{aM_{ij}}), \quad \forall j \in [2, n]. \end{aligned} \tag{5}$$

If all the above equations hold, the authentication is successfully completed and the fog provides the service. Otherwise, it refuses the request. After the entire phase, the fog caches (TK, σ) for payment request and further malicious user detection.

5.6 Billing based on Verified Service Amount

Periodically, fogs will request payment from the cloud according to their service amount, and then cloud demands the payment from the SP. This paper does not take into account the payment/amount contracts between them. We mainly focus on developing mechanisms to resist the cheating behavior of those semi-trust entities. In what follows, the interaction between fog and cloud and the interaction between cloud and SP will be introduced.

5.6.1 Between Fog and Cloud

The fog uploads its cached AUTH to the cloud. Besides the verification procedures that have been executed by the fogs (as shown in Section 5.5), the cloud will make further checking as follows:

1) Check all received TKs to see if there exist no duplicated tuples (k, C).
2) Check σ to see whether ID_{fog} is the identity of the certain fog node.

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

If either is not satisfied, the service cannot be counted. Especially, if Case 1 does not hold, the cloud submits the duplicated AUTH to TTP for user tracing, as to be described in Section 5.7.

5.6.2 Between Cloud and SP

The biggest obstacle to realize a trusted payment between cloud and SP is the constrained communication and computation capability of SPs. A trivial sampling method is not nearly enough. *Through our study, a constant-size commitment should exist, with which the cloud can claim the service number and existence of each service. Then, when SP wants to verify a service of a certain index, the cloud cannot use a substitute to slip through.* In our scheme, *merkle hash tree* works as this function.

For all collected AUTHs with one service, the cloud extracts the TKs and organizes them into an MHT structure. Assume that the root of MHT is R, with $\mathsf{NUM}_R = n_c$. For an honest cloud, n_c equals to the number of AUTHs (denoted as n_t). The cloud submits (R, n_c) to SP. Two sets $cha_2 \subset cha_1 \subset [1, n_c]$ is selected by SP and sent to the cloud.

The cloud responds the challenge by sending $\mathsf{AUHT}_i, \forall i \in cha_2$; $\mathsf{TK}_i, \forall i \in cha_1$ and auxiliary information $\Omega(cha_1)$.

The verification of SP is as follows:

1) $\forall i \in cha_2$, check the validation of AUTH_i using the procedure of Section 5.6.1;
2) Check whether there exist no duplicated TKs in cha_1;
3) $\forall i \in cha_1$, check whether TK_i is in the i^{th} position by calling Algorithm 1.

Algorithm 1: Position Checking for i

Input: The index i; cha_1; auxiliary information $\Omega(cha_1)$
Output: TRUE or FALSE

```
Init. x_1 ← 0; x_2 ← 0;
foreach j ∈ cha_1 do
    if j < i then
        x_1 + +;
    end
end
foreach j ∈ Ω(cha_1) do
    if j lies at the left of PATH (R ↔ i) then
        x_2 ← x_2 + NUM_j;
    end
end
return i ?= x_1 + x_2 + 1
```

If any of the above does not hold, the verification fails, and SP can request a punishment on cloud for its falsifying. Otherwise, SP pays the bill according to n_c.

Note that cha_1 and cha_2 are arbitrarily determined, whose sizes should be well determined with the tradeoff between SP's overhead and SP's detection ratio. A quantified discussion is to be presented later in Section 7.3, and we will further discuss in that section on, why the challenging sets are defined as $cha_2 \subset cha_1$.

5.7 Malicious User Tracing

We will not consider to trace users who repeatedly use the same k for one service on the same fog, as these accesses will be detected and rejected by the fog. However, when duplicated k is used on different fogs, the users can get services beyond the limitation k_{max}, and we should start this phase to trace the user for punishment.

Upon receiving a tracing request from the cloud, TTP checks these AUTHs for the following conditions: 1) whether TKs are duplicated; 2) ID_{fog} in these AUTHs are different and σ can be verified. If either does not hold, TTP aborts this phase: Especially, the event that former condition does not hold indicates that, the repeated authentications happens in one fog node. It should be, and is easily be detected and filtered by the fog, so as the latter condition. Otherwise, when both conditions hold, the tracing phase goes on.

TTP traverses its user list and extracts y_j of each user, and computes $C_j^{\dagger} = e(g,g)^{\frac{1}{y_j + H(ID_{ser}||k)}}$. If $C_j^{\dagger} = C$ in TK, this phase is completed and j is outputted as the exposed user identity.

6 Security Analysis

6.1 Unauthorized User Filtering and Semi-Trust Party Forgery Resist

We can make the semi-trust fogs filter out unauthorized user mainly because of the soundness of zero knowledge proof in Eq. (4) and replay attack resist against not only malicious users but also the fogs. We will first analyze the soundness of Fig. 3 and then prove the security against forgery attack.

Soundness of Fig. 3: Corresponding to this authentication phase, we can construct a simulator $\mathcal{B}$, which can extract the witnesses $\{y, K, L_1, L_2, \{K_{\rho(i)}^{\omega_i}\}_{1 \le i \le l}\}$. Therefore, for any user, his/her capability to complete the authentication task is equivalent to mastering the knowledge of witnesses. A more detailed proof can be referred to papers [4, 13].

Theorem 1. The proposed scheme is authenticated against unauthorized users based on its soundness.

PROOF 1. Due to the soundness of this authentication, an unauthorized user (either has unsatisfied attribute or has exceeded the maximal attempts) cannot complete a legal authentication message. This conclusion also works for outsider adversaries due to his/her empty attribute set.

The adversary $\mathcal{A}$ who has eavesdropped other users' authentication messages will be blocked if it tries to execute a forgery: Assume that $\mathcal{A}$ obtained several authentications, each of which contains the pair $(k, C = e(g,g)^{\frac{1}{y + H(ID_{ser}||k)}})$. A new σ with the same pair is not valid according to the verification, and to change k without modifying C is difficult due to collision resistance of the hash function and Discrete Logarithm assumption in $\mathbb{G}_T$. On the other hand, to change C means a new T_5 (see Fig. 3) and different $\hat{c}$. The soundness shows if this task can be completed with a non-negligible advantage, then the witnesses can be extracted.

Thus, forgery by replay attack will not be allowed in our scheme. Furthermore, our further analysis also works for the fog servers, which means semi-trust servers will not fake their service amount by this method.

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

It is worth noting that, the resist against semi-trust party forgery is the base of trusted payment. Since the non-reputation authentication let the fog and cloud hardly forge a valid AUTH, every asserted service can be trusted.

6.2 Privacy Preservation

This paper's privacy issues take into account user's identity and attributes. Thus, this section shows that the fogs and the cloud cannot learn any of these knowledges from the authentication messages.

Referring to Fig. 3, we firstly see that except $T_1, \ldots, T_5$ and $R_j, \forall j \in [1, n]$, other elements in σ leak no privacy. The reason is intuitive, as every A_* and B_* is blinded by individual noises, $\hat{c}$ is output by a one-way function, and each r_* is blinded by individual γ_*. Thus, when obtaining these elements, together with TK, the verifier cannot get any knowledge on the protected elements in Eq. (4).

In addition, the verification procedure Eq. (5) tells us that: with the above elements, any user, without the private information can simulate the authentication, and generates $T_1, \ldots, T_5$ and $R_j, \forall j \in [1, n]$. The entire σ can be simulated, which means that the authentication message is zero knowledge, and will not leak the private information.

The same feature also holds when a verifier has obtained multiple valid σ, which preserves unlinkability.

7 Experimentation

7.1 Computational Complexity

We implement our schemes in C program with PBC library 0.5.14. All experiment results below were measured on a standard 64-bit Fedora release 21 operation system with a 3.4 GHz Intel Core i3 processor. The measurements are performed based on Type A curve and Type D159 (MNT159) curve, and the average time on various operations are shown in Table 2.

TABLE 2
Average time taken by various operations for the two curves. All operations are measured in milliseconds

Operations	Type A		MNT159	
	$\mathbb{G}$	$\mathbb{G}_T$	$\mathbb{G}$	$\mathbb{G}_T$
Multiplication	.007	<0.001	.024	.004
Exponentiation	1.381	.115	3.594	.810
Pairing	0.779		2.784	
$H(\cdot)$	$< 10^{-4}$ (based on SHA256)			

One can see from Table 2 that MNT159 based algorithm costs about 3 times of Type A curve for the more expensive operations (e.g., exponentiation on $\mathbb{G}$ and pairing). Our further measurement is based on Type A, and a coarse analysis of the scheme feasibility will also be given when implementing stronger curves.

In the remaining of this section, we will firstly revisit the authentication and verification phases to analyze a theoretical performance, and then give an intuitive measurement and comparison on these phases. It should be noted that these two measured phases affect the service experience most in this system, as they bring response latency to users. For other phases, the payment interaction between fogs and cloud is beyond our scope, since they are similar to the verification phase; interaction between cloud and service providers are discussed in Section 7.3; and efficiency of user tracing will be briefly discussed later in this section.

7.1.1 Revisit The Scheme for Performance Optimization

Referring to [36], *multi-exponentiation* is defined as the format $\prod_{1 \le j \le k} g_j^{e_j}$. There exist efficient multi-exponentiation algorithms to accelerate the computations (e.g., [37] takes about 1.25 exponentiation time to compute a *multi-exponentiation*).

Authentication. A careful study on Fig. 3 can help find that $A_K, B_K, A_{L_1}, A_{L_2}$ and all T_i's can be generated beforehand without knowing any knowledge of the access policy. Then, the left online task is only $\{A_i\}_{i \in [1,l]}, \{R_j\}_{j \in [1,n]}$, and all r_*'s based on $\hat{c}$.

According to Table 2, the cost of hash $H(\cdot)$ can be ignored, so as the addiction and multiplication operations in $\mathbb{Z}_q$. Thus, we focus on whether there is optimization room for generation all A_i's and R_j's.

- $A_i = K_{\rho(i)}^{\omega_i} \tilde{g}^{\mu_i}$. Without the knowledge of policy, the user can still prepare $\tilde{g}^{\mu_i}$. Then it consumes one multiplication and exponentiation in $\mathbb{G}$ for each A_i if $\omega_i \neq 0$.
- R_j. It seems to take complex operations for $O(l \times n)$ times. However, when we rearrange the equations as $R_j = \prod_{i=1}^{l} e(\tilde{g}, \hat{h}_{\rho(i)}^{a\gamma_{\mu_i}})^{M_{ij}}$, the pairing operations are the same for every j. Hence, pairing operations are linear to row number l. Also, $e(\tilde{g}, h)^{-\gamma_{\mu_{L_2}}}$ can be prepared.

This phase can be further optimized, because every pairing step for R_j can be computed without knowing the access policy. This method can significantly save time, but expands user's storage if system's universal attribute set size is large. Thus, we measure both cases in our work. "Simple prepare" is denoted as the method without further preparing R_j, and "Deep prepare" represents the other approach.

Verification. In Eq. (5), each item is a multi-exponentiation in either $\mathbb{G}$ or $\mathbb{G}_T$. The verification of R_j will have the same property as in *Authentication* phase, since complex tasks do not need to repeat at all.

As it is executed by fog/cloud, we can rely on more powerful techniques to accelerate this phase, such as parallel computing [16] and hardware accelerators (e.g., Intel's QuickAssist Technology). In this paper, we only discuss the former method. Thus, the time consuming of verification will become the longest time spent by individual equation in Eq. (5) when equipped with sufficient cores. However, in the follows, we will analyze the performance in two cases: 1) without using parallel computing method (1 core), and 2) using parallel computing to accelerate the execution (3 cores are deployed in our implementation).

7.1.2 Implementation

The access policy in our measurement is a 7×5 matrix. The minimal size of attribute set is 2 to satisfy this policy; the maximal is 4. Fig. 4 shows the running time of authentication and verification phases.

For the same policy, authentication time is linear to the required attribute number (about 1.4 ms for one more attribute). Indeed, the complexity of the policy also affects the

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

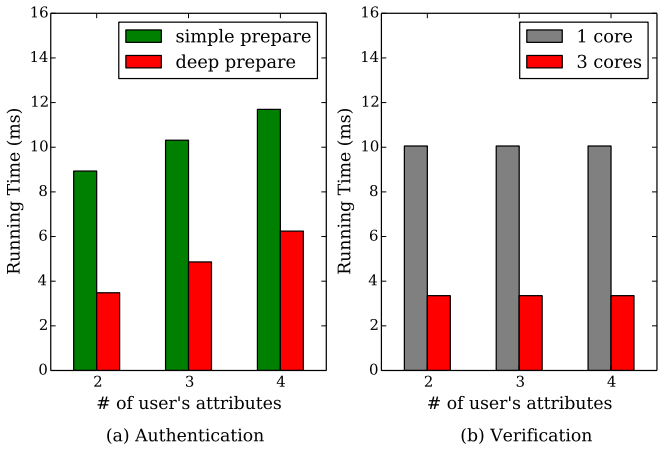


Fig. 4. Time Consumed in Authentication and Verification phases. (In Authentication, a *deep prepare* executes every $e(\tilde{g}, \hat{h}_{\rho(i)})^{\alpha\gamma_{\mu_i}}$ beforehand, without knowing the access result; In Verification, multiple cores calculate in parallel for different equations)

performance: 1) Algorithm using simple prepare is affected by both of the row number and column number of the policy; and 2) time with deep prepare will only be affected by the column number. Fig. 4(a) shows the execution time in authentication versus diverse attributes, and Table 3 gives a more detailed theoretical analysis.

The verification time is not affected by user's actually used attributes. From Fig. 4(b), a 10-ms response time seems good enough if equipped with only one core, and by increasing acceptable number of paralleled cores makes it more suitable for latency-sensitive service authentication. The verification cost is mainly affected by the complexity of access policy. However, as the complexity is difficult to measure in our implementation, we only list the theoretical analysis in Table 3.

From Fig. 4, we can find that, with multi-core (3-core) verification technique, the computation cost totally occupies the delay of around 8 ms (5 ms for authentication and 3 ms for verification) with deeper preparing, or about 14 ms with basic preparing (due to the longer authentication time). This property helps create the performance advantage for the proposed scheme in Section 7.2.

TABLE 3
Theoretical Efficiency Analysis

	pairing	exp		multi-exp	
		$\mathbb{G}$	$\mathbb{G}_T$	$\mathbb{G}$	$\mathbb{G}_T$
Authentication (simple)	l	i	0	0	n
Authentication (deep)	0	i	0	0	n
Verification	l	0	0	2	$n+3$

Note: l and n are row and column numbers in access policy, respectively;
i represents user's used attribute numbers.

7.1.3 Computation Complexness for User Tracing

In our scheme, the taken time is increasing along with the registered user number. Whereas, since it is an offline task to penalize privilege abuse, it can tolerate a certain amount of time to finish this job.

The test for each user in Section 5.7 takes only 0.115 ms. For the worst case, to trace a malicious user in 1-million-user system spends less than 2 minutes. It is a totally acceptable overhead.

7.2 Latency Analysis

In this measurement, we analyze and compare the delay of different schemes. The experiment setting is as follows. The network is a real topology in our laboratory, with about 50 devices accessing the LAN simultaneously, as the background flow. The link between fog and user is a two-hop one, one of which is Wi-Fi. In this setting, a round-trip time between fog and user is 3.54 ms on average, with the jitter of 8.16 ms, and round-trip time between a cloud server and user is 36.12 ms, with the jitter of 8.89 ms.

We compare our work with the work of [4] (denoted as *Yuen2015*) and a trivial service system with service providers' participation (denoted as *Basic*). Taking into account both the computation and transmission, the overall delay is as shown in Table 4. Especially, the *Basic* approach judges user's privilege based on some information (e.g., user list) in SP's cache in plaintext-form, but the messages between SP and users are protected against the cloud.

TABLE 4
Performance Comparison in User Authentication

Scheme	Response Delay (ms)		Online SP	Privacy
	Type A	MNT159		
Proposed	11.754	28.2	Not Needed	✓
Yuen2015	80.354	96.8	Not Needed	✓
Basic	36.3	36.3	Required	×

Delay with *MNT159* is estimated based on the one with *Type A* according to Table 2.

It is clear that in Table 4 the realization of authentication at the edge and non-interactive proof significantly reduce the access delay compared with *Yuen2015*'s Scheme [4]. In detail, the centralized access control in [4] requires two round-trip transmissions between cloud server and the user terminal, which contributes the major latency. What is more, authentication based on interaction cannot support trusted payment, since the cloud can simulate valid authentication messages itself without secret keys.

Also, compared with the *Basic* approach, our scheme shows slightly better performance, even in terms of the stronger curve (MNT159). This result shows that, even in terms of other stronger curves (e.g., MNT224), the performance disadvantage compared with the *Basic* approach can be acceptable. It is worth noting that, besides the problem of SP's participation, it is another critical issue that this *Basic* approach leaks user's privacy to the SP. It may be easy to repair the privacy problem, but its access delay will be even worse than [4], due to longer transmission path and weaker computation capability of SP.

7.3 Analysis on Payment Accounting for SP

In this section, we analyze SP's burden to realize the trusted payment between cloud and SP according to the mechanism of Section 5.6.2.

Due to the property of MKT structure, it is indeed true that the cloud cannot falsify when SP's random challenge "unluckily" hits any invalid leaf node of the asserted

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

structure. However, due to the gap between SP's limited resources and large amount of the services, the sample size of the challenge cannot be too large, whereas, inadequate challenge enforces the cloud to forge the service amount for undeserved benefits. Let us review Fig. 1 for instance.

Assume that only 7 valid service AUTHs (TK, σ) are collected in reality. To forge an 8-service amount, the cloud constructs the MHT as Fig. 1, where one of the leaf nodes (let it be the i-th one) can be a copy of any of the rest (denoted as j-th). Then, when SP randomly samples two as a challenge, cloud's falsifying will not be detected unless the sampled AUTHs are, by coincidence, the i-th and j-th. The detection rate seems too low, and SP has a really high probability to pay for the nonexistent service. This section analyzes how many should be sampled at least, to ensure SP's rights.

7.3.1 Definition on Economic Model

Before the quantitative analysis on SP' sampling amount, we firstly define the economic model, including the penalty contracts for cloud's forgery, and what method the cloud will follow to forge service amount.

Let n_r and n_c be the real and asserted service amount, respectively. If the *Billing* step finds no falsifying behavior, SP pays for n_c services. Otherwise, a punishment is executed on the cloud. A penalty contract includes that the cloud will gain no payment from this interaction. Furthermore, reputation loss also prevents the cloud from overladen forgery. However, the reputation factor is difficult to measure. For analysis clarity, this paper removes this part. Such analysis is conservative, meaning that if the analysis outputs a feasible configuration, the real system is secure, since the cloud takes on more economic risks due to the unconsidered reputation loss.

Then we describe cloud's tactics for forging an nonexistent service. The ideal way is presented in the third paragraph of Section 7.3. When either of the i-th and j-th is sampled, the cloud submits the j-th AUTH.

We make $n_r > n_c/2$. Otherwise, SP sacrifices too much unnecessary payment. Also, our following analysis shows that the cloud will not assert a too outrageous service amount.

It should be noted that, the cloud indeed has many forgery behaviors: e.g., directly generating an invalid TK/AUTH, or copying a valid TK which has already been copied more than once. However, under the sampling tactics to be proposed, the cloud can win the largest probability to escape from detection and largest utility, with our given behavior. Thus, in the following analysis, we fix cloud's forgery tactic.

7.3.2 Quantitative Analysis

Due to our setting, there exist $d\ (= n_c - n_r)$ AUTHs whose TKs are duplicated used in MKT. Let $\mathcal{X}_i$ be the event that the i-th of them is not detected. Then a successful detection is denoted as $\mathcal{V}$, whose probability holds:

$$\begin{aligned} &\Pr(\mathcal{V}) = 1 - \Pr(\bar{\mathcal{V}}), \\ &\Pr(\bar{\mathcal{V}}) = \Pr(\mathcal{X}_1) \cdot \Pr(\mathcal{X}_2|\mathcal{X}_1) \ldots \Pr(\mathcal{X}_d|\mathcal{X}_{1,2,\ldots,d-1}) \\ &\qquad < \prod_{i=1}^{d} \Pr(\mathcal{X}_i) = \Pr(\mathcal{X}_1)^d. \end{aligned} \tag{6}$$

The inequality $\Pr(\mathcal{X}_i|\bar{\mathcal{X}}_{1,\ldots,i-1}) < \Pr(\mathcal{X}_i)$ holds, because when the previous $i-1$ duplications are not detected, there remains more samples, which decrease the probability of $\mathcal{X}_i$.

In a payment interaction with sample amount m, the probability of event $\mathcal{X}_1$ is as:

$$\begin{aligned} \Pr(\mathcal{X}_1) &= 1 - (1 - \frac{n_c - m}{n_c})(1 - \frac{n_c - m - 1}{n_c - 1}) \\ &= 1 - \frac{m^2}{n_c^2 - n_c}, \end{aligned} \tag{7}$$

which means that both of the copies are not selected in the challenge. Referring to Eq. (6), we have

$$\Pr(\bar{\mathcal{V}}) < (1 - \frac{m^2}{n_c^2 - n_c})^d. \tag{8}$$

Under our economic model, the cloud will not make a falsifying tactics under which cloud's utility is even lower than an honest payment interaction. Thus, $n_c \cdot \Pr(\bar{\mathcal{V}}) > n_r$, we can further get:

$$\begin{aligned} &(1 - \frac{m^2}{n_c^2 - n_c})^d > \frac{n_r}{n_c} \quad (m \ll n_c) \\ &\Rightarrow (1 - \frac{m^2}{n_c^2 - n_c})^{-\frac{n_c^2 - n_c}{m^2} d} < (\frac{n_r}{n_c})^{-\frac{n_c^2 - n_c}{m^2}} \\ &\Rightarrow m > \sqrt{-\frac{n_c^2 - n_c}{n_c - n_r} \ln \frac{n_r}{n_c}}. \end{aligned} \tag{9}$$

Fig. 5 depicts the required sample amount against diverse real/asserted service amounts. The result looks ideal since SP only needs a little cost to achieve detection with sufficient accuracy. Fig. 6 analyzes it on another aspect: when there are 20,000 real services, and the sample amount is fixed, cloud's utility is shown in the figure versus increasing forgeries. The result shows cloud's most rational tactics is to honestly report its service amount when the challenge contains 200 samples.

Note that, if there are fewer samples, the utility curves increase alongside the forgery amount. This phenomena occurs due to two deviations in our analysis: 1) Cloud's reputation loss is beyond our scope; 2) The inequality in Eq. (6) makes our calculated utility expectation higher, especially when the forgery amount increases. Calibration for these two issues will prevent the cloud from submitting a outrageous forgery when executing adequate challenges.

Fig. 7 shows the required auxiliary information in MHT structure with asserted service n_c = 30,000. Each result is measured in real sampling experiments for 10,000 times. An auxiliary node is a hash value and is far smaller than AUTH (e.g., 32 Bytes with SHA256). Thus a 250-sample payment interaction needs only about 50-MB bandwidth for the auxiliary information.

It is worth noting the case when selfish cloud just forges AUTH, whose TK is distinct to existent ones. In that case, cloud has to generate invalid σ for each AUTH, and the probability formulation versus sample amount is as

$$\Pr(\bar{\mathcal{V}}) = \frac{C_{n_r}^{m'}}{C_{n_c}^{m'}}, \tag{10}$$

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/TDSC.2018.2845381, IEEE Transactions on Dependable and Secure Computing

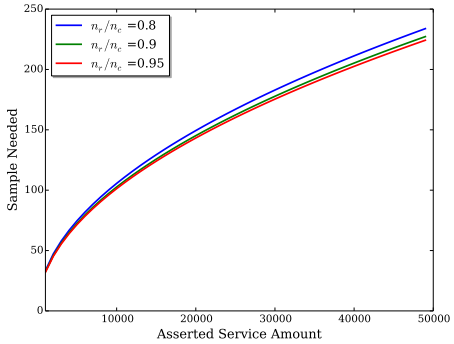


Fig. 5. Required Samples

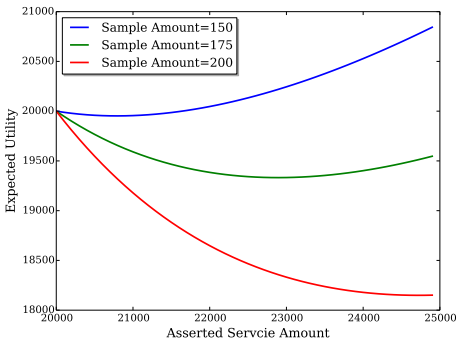


Fig. 6. Cloud's Utility ($n_r = 20,000$)

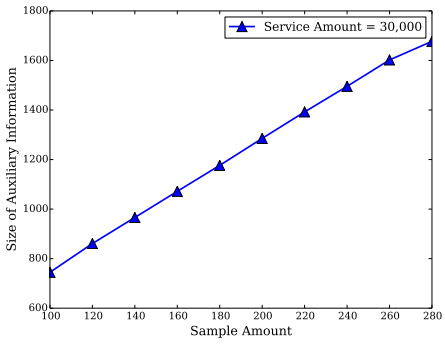


Fig. 7. Required Auxiliary Information in MHT

which only needs far smaller samples than Eq. (8) to achieve the same detection ratio. This is the reason why the chosen challenging set cha_2 can be smaller than cha_1 in Section 5.6.2. Let the size of cha_2 be $m' = 80$ in the former measurement, and the communication overhead in the interaction is about 100 MB (Using Type A curve), half of which is for the auxiliary information, which is an acceptable cost for SP.

8 Conclusion

In this paper, we proposed a new privacy-preserving authentication scheme for outsourced service in fog-to-cloud architecture. We allow every service provider to autonomously determine the access policy for its published service, including that who can receive the benefits of the service and how many times he/she is permitted to receive benefits from the service. The authentication messages can be further used as the credential of the trusted payment in pay-as-you-use billing model. With the implementation of merkle hash tree, service providers only take little communication and computation burden to prevent the cloud/fog from forging service amount for additional profits. The security proof and experimental measurements indicate that our scheme can realize the security and performance requirements as we expected, and the advantages are significant compared with related works.

Thus, authentication outsourcing brings great benefits to the system, and is worthwhile to be researched further. Regarding future work, we plan to design functionalities faced with diverse scenarios for outsourced latency-sensitive services.

Acknowledgment

The authors sincerely thank the anonymous referees for their invaluable suggestions that have led to the present improved version of the original manuscript. This work is partly supported by the National Key Research and Development Plan of China under Grant No. 2017YFB0801702, the National Natural Science Foundation of China under Grant No. 61671420, Youth Innovation Promotion Association CAS under Grant No. 2016394 and the Fundamental Research Funds for the Central Universities.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, A. Katz, Randy, G. Lee, D. Patterson, A. Rabkin, I. Storia, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] S. Ayoubi, Y. Zhang, and C. Assi, "A reliable embedding framework for elastic virtualized services in the cloud," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 489–503, 2016.

[3] S. Zawoad, A. K. Dutta, and R. Hasan, "Towards building forensics enabled cloud through secure logging-as-a-service," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 148–162, 2016.

[4] T. H. Yuen, J. K. Liu, M. H. Au, X. Huang, W. Susilo, and J. Zhou, "k-Times attribute-based anonymous access control for cloud computing," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2595–2608, 2015.

[5] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*, pp. 169–186, Springer, 2014.

[6] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing-a key technology towards 5G," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[7] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 120–128, 2016.

[8] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data (MOBIDATA)*, pp. 37–42, ACM, 2015.

[9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of 2012 Workshop on Mobile Cloud Computing (MCC)*, pp. 13–16, ACM, 2012.

[10] Y. Zhong, K. Xu, X.-Y. Li, H. Su, and Q. Xiao, "ESTRA: Incentivizing storage trading for edge caching in mobile content delivery," in *Proceeding of 2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2015.

[11] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The kerberos network authentication service (V5)," *RFC4120*, 2005.

[12] R. Housley, W. Polk, W. Ford, and D. Solo, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," *RFC3280*, 2002.

[13] I. Teranishi, J. Furukawa, and K. Sako, "K-times anonymous authentication," in *Proceedings of 2004 International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt)*, vol. 3329, pp. 308–322, Springer, 2004.

[14] A. Kupcu, "Incentivized outsourced computation resistant to malicious contractors," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 6, pp. 633–649, 2017.

[15] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proceedings of the 30th Annual Cryptology Conference (CRYPTO)*, pp. 465–482, Springer, 2010.

[16] K. Xue, S. Li, J. Hong, Y. Xue, N. Yu, and P. Hong, "Two-cloud secure database for numeric-related sql range queries with privacy preserving," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1596–1608, 2017.

[17] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4075–4088, 2017.

[18] P. A. A. Shah, M. Habib, T. Sajjad, M. Umar, and M. Babar, "Applications and challenges faced by internet of things-a survey," in *International Conference on Future Intelligent Vehicular Technologies*, pp. 182–188, Springer, 2016.

[19] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 28th IEEE Symposium on Security and Privacy (S&P)*, pp. 321–334, IEEE, 2007.

[20] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in *Proceeding of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 91–98, IEEE, 2011.

[21] H. Ma, R. Zhang, Z. Wan, Y. Lu, and S. Lin, "Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 6, pp. 679–692, 2017.

[22] K. Xue, J. Hong, Y. Xue, D. S. Wei, N. Yu, and P. Hong, "CABE: A new comparable attribute-based encryption construction with 0-encoding and 1-encoding," *IEEE Transactions on Computers*, 2017.

[23] K. Xue, Y. Xue, J. Hong, W. Li, H. Yue, D. S. Wei, and P. Hong, "RAAC: Robust and auditable access control with multiple attribute authorities for public cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 953–967, 2017.

[24] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1484–1496, 2016.

[25] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Proceedings of 2011 Cryptographers' Track at the RSA Conference (CT-RSA)*, vol. 6558, pp. 376–392, Springer, 2011.

[26] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren, "Attribute-based signature and its applications," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pp. 60–69, ACM, 2010.

[27] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," *Proceedings of 2001 Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pp. 93–118, 2001.

[28] L. Nguyen and R. Safavi-Naini, "Dynamic k-times anonymous authentication," in *Proceedings of 2005 International Conference on Applied Cryptography and Network Security (ACNS)*, vol. 3531, pp. 318–333, Springer, 2005.

[29] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," in *Proceedings of 2006 International Conference on Security and Cryptography for Networks (SCN)*, pp. 111–125, Springer, 2006.

[30] R. Cramer, I. Damgård, and U. Maurer, "General secure multi-party computation from any linear secret-sharing scheme," in *Proceedings of 2000 International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt)*, pp. 316–334, Springer, 2000.

[31] P. S. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Proceedings of 2005 International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, pp. 515–532, Springer, 2005.

[32] R. C. Merkle, "Protocols for public key cryptosystems," in *processing of 1980 IEEE Symposium on Security and Privacy (S&P)*, pp. 122–122, IEEE, 1980.

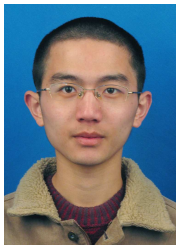
[33] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," *Proceedings of 2009 European Symposium on Research in Computer Security (ESORICS)*, pp. 355–370, 2009.

[34] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, 1994.

[35] X. Zhu, S. Shi, J. Sun, and S. Jiang, "Privacy-preserving attribute-based ring signcryption for health social network," in *Proceedings of 2014 IEEE Global Communications Conference (GLOBECOM)*, pp. 3032–3036, IEEE, 2014.

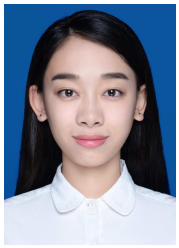
[36] B. Möller, "Improved techniques for fast exponentiation," in *Proceedings of 2002 International Conference on Information Security and Cryptology (ICISC)*, vol. 2587, pp. 298–312, Springer, 2002.

[37] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.



Jianan Hong received the B.S. degree from the department of Information Security, University of Science and Technology of China (USTC), in 2012. He is currently working toward the Ph.D. degree in Information Security from the Department of Electronic Engineering and Information Science (EEIS), USTC. His research interests include secure cloud computing and mobile network security.

Kaiping Xue (M'09-SM'15) received her B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003 and received his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. Currently, he is an Associate Professor in the Department of Information Security and Department of EEIS, USTC. His research interests include next-generation Internet, distributed networks and network security. He is the corresponding author of this paper.



Na Gai received the B.S. degree from the department of Information Security, University of Science and Technology of China (USTC), in 2018. She is currently a graduated student in Information Security from the Department of Electronic Engineering and Information Science (EEIS), USTC. Her research interests include network security protocol design and analysis.

David S.L. Wei (SM'07) received his Ph.D. degree in Computer and Information Science from the University of Pennsylvania in 1991. He is currently a Professor of Computer and Information Science Department at Fordham University. From May 1993 to August 1997 he was on the Faculty of Computer Science and Engineering at the University of Aizu, Japan (as an Associate Professor and then a Professor). He has authored and co-authored more than 100 technical papers in various archival journals and conference proceedings. He served on the program committee and was a session chair for several reputed international conferences. He was an associate editor of IEEE Transactions of Cloud Computing, a lead guest editor of IEEE Journal on Selected Areas in Communications for the special issue on Mobile Computing and Networking, a lead guest editor of IEEE Journal on Selected Areas in Communications for the special issue on Networking Challenges in Cloud Computing Systems and Applications, a guest editor of IEEE Journal on Selected Areas in Communications for the special issue on Peer-to-Peer Communications and Applications, and a lead guest editor of IEEE Transactions on Cloud Computing for the special issue on Cloud Security. He is currently an Associate Editor of Journal of Circuits, Systems and Computers, and a guest editor of IEEE Transactions on Big Data for the special issue on Trustworthiness in Big Data and Cloud Computing Systems. Currently, His research interests include cloud computing, big data, IoT, and cognitive radio networks.

Peilin Hong received her B.S. and M.S. degrees from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1983 and 1986. Currently, she is a Professor and Advisor for Ph.D. candidates in the Department of EEIS, USTC. Her research interests include next-generation Internet, policy control, IP QoS, and information security. She has published 2 books and over 150 academic papers in several journals and conference proceedings.