A Cloud based Dual-Root Trust Model for Secure Mobile Online Transactions

Li Li[1], Dijiang Huang[2], Zhidong Shen[1], Samia Bouzefrane[3],
[1] Wuhan University, P. R. China, Email: {lli, shenzd}@whu.edu.cn
[2] Arizona State University, U.S.A., Email: dijiang@asu.edu
[3] Conservatoire National des Arts et Métiers, France, Email: samia.bouzefrane@cnam.fr

***Abstract*—With rapid growth of mobile devices and the emergency of mobile cloud services, it is a trend to use mobile devices for mobile-centric applications, and expand the mobile capabilities and provide needed security by mobile cloud services. However, due to the mobility of the device and the semi-trust of the mobile cloud, how to build trust in the mobile applications is a big concern. In this paper, we propose a dual-root trust online transaction model that provides a dual-root trust model including both the user's mobile device and a delegation mobile cloud. We design a dual-root trust protocol by leveraging a modified CP-ABE cryptography and the trust execution environment embedded in a mobile device to provide device-specific transaction confirmations for online transactions initiated by the mobile user. The performance evaluation of the protocol demonstrates that it is a lightweight scheme for mobile devices since most cryptographic functions are delegated from users to the mobile cloud.**

I. Introduction

As the world becomes more interconnected, integrated and intelligent, mobile devices are playing ever-increasing roles in changing the ways that people live, work and communicate. Then mobile devices become an ideal platform for carrying identity credentials and using them for logical access and online transactions [1]. Unfortunately, mobile devices are prone to loss and theft due to their small size and high-portability [2]. Attackers may compromise data stored on a mobile device or launch transactions using credentials stored in mobile devices.

With the emergency of cloud-based service models, e.g., [3], [4], building the credential platform for mobile devices in the cloud has become a natural choice for mobile applications, e.g., Google Wallet [5] that considers the cloud is fully trusted. However, it is necessary to address critical security issues of mobile applications that rely on both mobile devices and clouds, such as how to make sure the cloud will not launch transactions on behalf of users without users' acknowledgements and how to assure that it is the right user using the right device to launch transactions when online transactions fully trust mobile devices or clouds.

To address the above-described issues, we present a Dual-Root Trust (DRT) scheme to secure mobile online transactions by considering both mobile devices and clouds as semi-trusted, and they need to work together as the verification parties of mobile online transactions. DRT requires secure actions from both mobile devices and the cloud, i.e., none of parties can individually process and confirm an online transaction. DRT is not a simply two-factor authentication solution and it is different from traditional two-factor authentication approaches, e.g., using SMS as a second authentication approach [6] and security RSA ID tokens [7], in that DRT natively incorporates policy-enforcement by using attribute-based cryptography [8], in which transaction contents and users' roles can be verified during the online transaction verification procedure. Thus, DRT provides a more flexible policy-based dual-root verification model. Moreover, DRT may rely on recent developed Trusted Executive Environment (TEE) approaches, such as [14], [9], [10], to develop the trust root functions on mobile devices to deal with compromised mobile OS. This requires the mobile trust root functions must be efficient due to limited hardware support of TEE components such as a smartcard. In summary, the presented DRT scheme achieves the following contributions: 1) a new dual-root trust model considering both mobile devices and clouds as trust roots; 2) a policy based online transaction verification procedure; and 3) an ABE-based computing delegation model that allocates minimal computing overhead on mobile devices.

The presented performance evaluation study demonstrates the security strength against untrusted participants impersonating system participating entities (i.e., mobile devices, clouds, and service providers), and the presented computation overhead analysis shows the practicality of the DRT scheme considering the execution on resource constrained mobile devices.

The rest of the paper is organized as follows. Section II describes the related work. Section III outlines the system and assumptions for the proposed scheme. Section IV presents detailed Dual-Root Trust Scheme. Section V analyzes the proposed DRT scheme. We conclude the paper in Section VI.

II. Related Work

Filyanov A. et al. [11] designed a secure transaction confirmation architecture called UTP that provides assurance to a remote server that the user of a client system has indeed confirmed a proposed action. The "just one device" transaction confirmation is similar to our device-specific confirmation, but UTP provides more assurance to service providers than to client users since the feedback available to client users remains susceptible to manipulated by malwares. E-EMV [12] is a software-based credit card application to secure transaction confirmation. TruWalletM [13] uses M-Shield [14] on a mobile

phone to protect login credentials of a user and invoked only during login. Nokia's onBoard Credentials project explores an open credential platform and using ARM TrustZone [9] as a platform for the prototype [15], but there is no transactions considered based on the platform. M. Nauman et. [16] proposed extensions to the OAuth protocol to provide "device-specific" authorization for native applications on TPM based smartphone platforms, and also did not consider transaction confirmations. There is no mobile cloud involved in these schemes that limits the computation ability of the resource constrained device.

Tassanaviboon and Gong [17] proposed an authorization scheme named AAuth using a modified CP-ABE to provide end-to-end encryption and tokens to enable authorization by both authorities and owners. Compared to our scheme, AAuth can only provide authorization for tokens and the computations is much heavier if the client is a resource constrained mobile device.

Krautheim [18] proposed the Private Virtual Infrastructures that represents a new cloud management model, and shares the responsibility of security in cloud computing between the service provider and the client. This dual root infrastructure assumed changes to the IaaS, while we focus on building trusts between parties involved in an online transaction in the application level.

III. DRT System and Models for Mobile Online Transactions

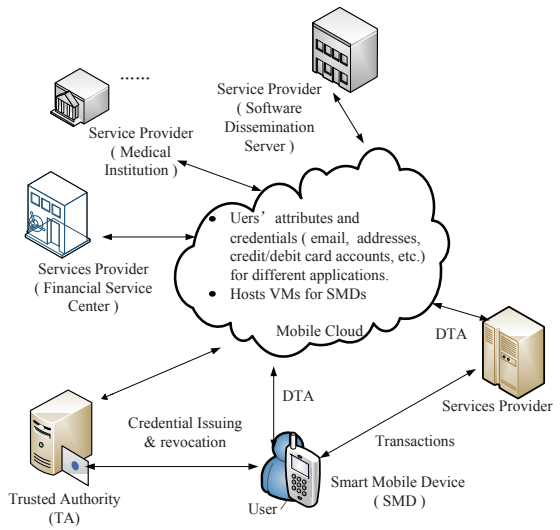


Fig. 1. System models for DRT.

A. DRT System Design Requirements and Assumptions

In the presented DRT protocol running environment, a smart device may be used for different device-specific online transactions. For example, users request to download software from a software dissemination server, remotely log into servers to access business files, or launch an online payment. In the private-business mixed transactions environment, mobile users may have many credentials such as online IDs, billing and delivery addresses, credit card information, secret keys, etc. Keeping these private information on mobile devices is insecure and executing cryptography operations on credentials in the mobile device is power consuming.

To solve the problem, we propose a mobile transaction system model as shown in Fig. 1. Mobile users delegate data and credentials to the mobile cloud but make device-specific confirmations to all mobile online transactions launched in the device. This requirement needs mobile users to be involved as a human-in-the-loop solution to prevent adversaries from deploying mobile online fraud transactions. As a result, the DRT requires two trusted roots: (a) a security element on each mobile device, and (b) a cloud-based supporting system.

To achieve the described DRT design requirements, we have the following assumptions: 1) Each mobile device used in the scheme is equipped with a hardware-based trusted execution environment (TEE) that provides: (a) isolated and integrity protected execution of trusted codes, and (b) secure storage for secrets. We assume that attackers cannot bypass the security mechanism of the TEE to get the user's credential to perform the user-in-the-loop security operations described in the DRT protocol; 2) There is a dedicated virtual machine providing storage, networking, and computing resources for mobile devices in the mobile cloud; 3) Service providers and the Mobile Cloud set up SSL/TLS security channels to protect inter-communications. All secrets preinstalled in the system from the TA are protected through a secure channel.

B. DRT System Components

There are five following major system components in the presented DRT system model as shown in Figure 1:

- *Trusted Authority (TA)*: A TA is a trusted party to generate secrets key for DRT participants.
- *Samrt Mobile Device (SMD)*: Each SMD used in DRT is equipped with a hardware-based TEE. Secret keys should be preinstalled in the TEE and only if with correct password, can the TEE authorization procedure be initiated and corresponding secret keys be used. There is a unique device identity (e.g., International Mobile Equipment Identity – IMEI) that is stored in the ROM of each mobile device.
- *Mobile User*: A user is the owner of a SMD. The user authenticates himself to a SMD with traditional username/password and launches transactions through applications in the SMD. There is a binding between the user and her SMD. If the user lost her SMD, the binding can be broken.
- *Mobile Cloud (MC)*: The MC provides secure storage and computational services to users, and provides management of credentials stored in the cloud. Data in the MC should be managed under an access control policy. Due to page limits, we do not discuss the details in this paper on how to implement secure storage and access control scheme in the MC.
- *Service Provider (SP)*: An SP provides services to mobile users. SPs can be authenticated through traditional certificates, e.g., using SSL certificates by the MC.

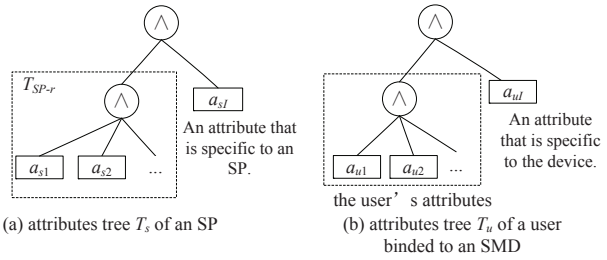
C. Adversary Model

Adversaries attacking the DRT system can be both external and internal attackers. Attackers have complete control over the network among DRT participants. No entities in DRT trusts the others but trusts the protocol. Internal attackers who are working for services providers or the MC can get access to mobile users' private data (although data in ciphertext or protected). They are interested in launching fraudulent transactions illegitimately on behalf of users. Mobile device may be lost. Users may launch transactions using others' device.

D. Attribute-Based DRT Cryptography System

The DRT's crypto system is designed based on the Ciphertext Policy Attributed-Based Encryption (CP-ABE) [8] scheme. CP-ABE uses the bilinear pairing that is a bilinear map function $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$, where $\mathbb{G}_0$ is an addition group and $\mathbb{G}_1$ is a multiplicative cyclic group. The Discrete Logarithm Problem (DLP) on both $\mathbb{G}_0$ and $\mathbb{G}_1$ are hard. Pairing has the *Bilinearity* property:

$$e(P^a, Q^b) = e(P, Q)^{ab}, \quad \forall P, Q \in \mathbb{G}_0, \forall a, b \in \mathbb{Z}_p^*.$$



(a) attributes tree T_s of an SP

(b) attributes tree T_u of a user binded to an SMD

Fig. 2. Attributes tree in the DRT.

.

An attribute tree in our DRT model is composed by leaf nodes and internal nodes. Each leaf nodes represents an attribute, and each internal node is a logical gate, such as "AND", "OR", "n-of-m", as shown in Fig. 2.

The following functions and terms that will be used in next section are defined as follows:

- $att(x)$ returns the attribute associated with the leaf node x.
- $\mathbf{parent}(x)$: return the parent node of node x;
- num_x is the number of children of a node x. A child y of node x is uniquely identified by an index integer $\mathbf{index}(y)$ from 1 to num_x.
- The threshold value $k_x = num_x - 1$ when x is an "AND", and $k_x = 0$ when x is an "OR" gate or a leaf node. k_x is used as the polynomial degree for node x using the threshold secret sharing scheme.

IV. Description of DRT Transaction Scheme

A. System Parameters Setup

The TA runs **DRT_Setup**(k) procedure to generate a set of public parameters that are shared by all participants as follows:

TABLE I
CRYPTOGRAPHIC NOTATIONS USED IN THE DRT.

Notation	Description
a_{uI}	An attribute specific to the device, e.g., IMEI.
A_u	$A_u = \{a_{uI}\} \cap \{a_{ui} \mid i = 1, ..., j\}$, where a_{ui} are attributes to describe the user that stored in the MC.
A_s	$A_s = \{a_{s_i} \mid i = 1, ..., k\}$ includes attributes to describe the SP, and there is one attribute denoted as $a_{sI} \in A_s$ that will be discussed later.
r_u	$r_u \in \mathbb{Z}_p$, a random number generated by the TA for each user.
r_s	$r_s \in \mathbb{Z}_p$, a random number generated by the TA for each SP.
SK_u	$SK_u = \langle D_u = g^{(\alpha + r_u)/\beta}$; $\forall a_j \in A_u : D_j = g^{r_u} \times H(a_j)^{r_j}, D'_j = g^{r_j} \rangle$, where D_u is stored in the SMD, and D_j and D'_j are stored in the MC.
SK_s	$SK_s = \langle D_s = g^{(\alpha + r_s)/\beta}$; $\forall a_k \in A_s : D_k = g^{r_s} \times H(a_k)^{r_k}, D'_k = g^{r_k} \rangle$, and SK_s is stored in the SP itself .
a	$a \in \mathbb{Z}_p$ is a random number generated by the TA for each device, and stored in the SMD.

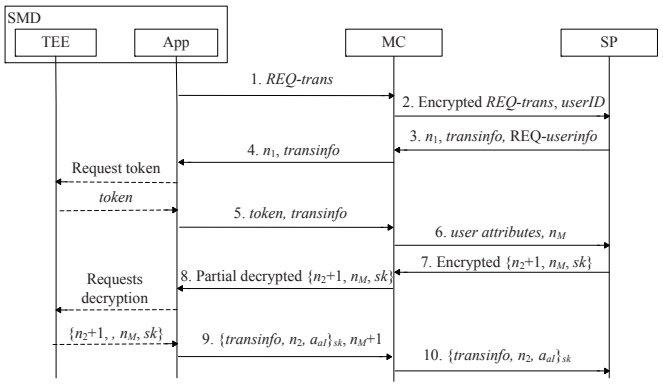


Fig. 3. DRT protocol.

DRT_Setup(k): Chooses a security parameter k and outputs a bilinear group map between two cyclic groups $\mathbb{G}_0$ and $\mathbb{G}_1$. The $\mathbb{G}_0$ is a bilinear group of prime order p with generator g. $H : \{0,1\}^* \rightarrow \mathbb{G}_0$ is a hash function. The TA chooses two random values α, $\beta \in \mathbb{Z}_p$. The TA keeps the master key $MK = (\beta, g^\alpha)$ secret, and published the following public parameters:

$$PK = \langle \mathbb{G}_0, g, h = g^\beta, e(g, g)^\alpha \rangle. \tag{1}$$

B. DRT Key Generation and Registration

The key generation algorithm takes attributes in A_u of the user and attributes in A_s of the SP as shown in Table I as input and outputs secret keys for a user and an SP separately. The detailed DRT key generation function **DRT_Genkey**(MK, A_u) for users' attributes is presented as follows:

1) Chooses random numbers $r_u, a \in \mathbb{Z}_p$, and calculates g^{r_u}, g^a, where a acts as a private key of the device.
2) Chooses random numbers $r_j \in \mathbb{Z}_p$ for each attribute $a_j \in A_u$.
3) Computes the secret keys as follows:

$$SK_u = \langle D_u = g^{(\alpha + r_u)/\beta}; \quad \forall a_j \in A_u : D_j = g^{r_u} \times H(a_j)^{r_j}, D'_j = g^{r_j} \rangle.$$

4) Registers the user in the MC by sending g^a, A_u and $\langle \forall a_j \in A_u : D_j = g^{r_u} \times H(a_j)^{r_j}, D'_j = g^{r_j} \rangle$ to the

MC through a secure channel and binding the user with the device attribute a_{uI} in A_u.
5) Stores D_u and a in the TEE of the SMD.

The TA performs **DRT_Genkey(MK,**A_s**)** to generate secret keys for a SP as follows:

1) Chooses a random number $r_s \in \mathbb{Z}_p$, and calculates g^{r_s}.
2) Chooses random numbers $r_k \in \mathbb{Z}_p$ for each attribute $a_k \in A_s$.

$$SK_s = \langle D_s = g^{(\alpha+r_s)/\beta}; \forall a_k \in A_s : D_k = g^{r_s} \times H(a_k)^{r_k}, D'_k = g^{r_k} \rangle.$$

3) Delivers SK_s to the SP through a secure channel.

C. Duel-Root Trust Protocol

We assume that the user opens an online application in the SMD. To use DRT, the user should first connect to MC using his device through a secure channel, e.g., a VPN. Then the following protocol takes steps as shown in Figure 3.

1) First, the user initiates a request of transaction through the application (hereafter, it is referred as "App") in his SMD to request an online transaction from the SP. To do this, the App uses attributes for the expected SP, for example, the DNS name, IP address etc., to generate an attributes tree $\mathcal{T}_{SP}$ as shown in Fig 2(a), which has only one attribute in the right subtree. To be simple and minimum the communication, we assumed attributes trees discussed later are all AND gates trees. The attribute in the right subtree is denoted as a_{sI} which is one of the SP's attribute will be handled in SMD. Then the App generates randomly an 1-degree polynomial $q_R(x)$ [8] where R is the root node of $\mathcal{T}_{SP}$, and sets $s = q_R(0)$, $s_1 = q_R(1)$, and $s_2 = q_R(2)$, and then generates two random nonces $n_1, n_2 \in \mathbb{Z}_p$ and performs **SemiEncrypt(**PK**,** M_u**,** a_{sI}**)** as follows, where $M_u = \{n_1, n_2, a_{uI}\}$:
   a) Calculates $C_0 = \langle g^{s_2}, H(a_{sI})^{s_2} \rangle$
   b) Computes $C_u = M_u e(g,g)^{\alpha s}$, and $C_u = h^s$.
   c) Sends $REQ-trans$ to the MC, where $REQ-trans = \{\mathcal{T}_{SP}, C_0, \widetilde{C}_u, C_u, s_1\}$:

$$SMD \xrightarrow{REQ-trans} MC.$$

2) The MC retrieves the required attributes according $\mathcal{T}_{SP-r}$ and performs **Encrypt**$(\mathcal{T}_{\mathcal{SP}}, s_1)$ as follows:
   a) $\forall x \in \mathcal{T}_{SP-r}$, randomly chooses a polynomial q_x with degree $d_x = k_x - 1$, where k_x is the secret sharing threshold value[8]: For the root node R_{SP-r} of $\mathcal{T}_{SP-r}$, chooses a $d_{R_{SP-r}}$-degree polynomial with $q_{R_{SP-r}}(0) = s_1$. $\forall x \in \mathcal{T}_{SP-r} \setminus R_{SP-r}$ sets d_x-degree polynomial with $q_x(0) = q_{\mathbf{parent}(x)}(\mathbf{index}(x))$.
   b) Calculates the following ciphertext:

$$\{\forall y \in Y_{SP-r} : C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)}\},$$

   where Y_{SP-r} is the set of leaf nodes in $\mathcal{T}_{SP-r}$.
   c) Gets all the following ciphertexts:

$$C_{\mathcal{T}_{SP}} = \langle \mathcal{T}_{SP}; \widetilde{C}_u = M_u e(g,g)^{\alpha s}; C_u = h^s; \forall y \in Y_{SP} : C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)} \rangle.$$

   d) Sends $C_{\mathcal{T}_{SP}}$ to the SP.

$$MC \xrightarrow{\{C_{\mathcal{T}_{SP}}, userID\}} SP.$$

3) The SP retrieves secret keys $\widetilde{SK}_s \subseteq SK_s$ that are related to attributes in $\mathcal{T}_{\mathcal{SP}}$, and invokes **Decrypt**$(\widetilde{SK}_s, C_{\mathcal{T}_{SP}})$[8] as follows :
For each leaf node y that contains an attribute in $\mathcal{T}_{SP}$, the SP runs a function **DecryptNode**$(C_{\mathcal{T}_{SP}}, \widetilde{SK}_s, y)$[8] as follows:

$$\begin{aligned} &\mathrm{DecryptNode}(C_{\mathcal{T}_{SP}}, \widetilde{SK}_s, y) \\ &= \frac{e(D_i, C_y)}{e(D'_i, C'_y)} = \frac{e(g^{r_s} \cdot H(i)^{r_i}, g^{q_y(0)})}{e(g^{r_i}, H(i)^{q_y(0)})} \\ &= e(g,g)^{r_s q_y(0)} = F_y. \end{aligned} \tag{2}$$

For each non-leaf node x in in $\mathcal{T}_{SP}$, the recursion is processed as follows: $\forall y$ is the child of x, it calls **DecryptNode**$(C_{\mathcal{T}_{SP}}, \widetilde{SK}_s, y)$ and stores the output as F_y. Let S_x be a k_x-sized set of children nodes of x, the SP computes:

$$\begin{aligned} F_x &= \prod_{y \in S_x} F_y^{\Delta_{i,S'_x(0)}} = \prod_{y \in S_x} (e(g,g)^{r_s \cdot q_y(0)})^{\Delta_{i,S'_x}(0)} \\ &= \prod_{y \in S_x} (e(g,g)^{r_s \cdot q_{\mathbf{parent}(y)(\mathbf{index}(y))}})^{\Delta_{i,S'_x}(0)} \\ &= \prod_{y \in S_x} (e(g,g)^{r_s \cdot qx(i) \cdot \Delta_{i,S'_x}(0)} \\ &= e(g,g)^{r_s q_x(0)}, \end{aligned} \tag{3}$$

where $i = \mathbf{index}(z)$ and $S'_x = \{\mathbf{index}(z) : z \in S_x\}$, $\Delta_{i,S'_x}(0)$ is the Lagrange coefficient. If the tree satisfied by attributes, the recursive function **DecryptNode**$(C_{\mathcal{T}_{SP}}, \widetilde{SK}_s, root)$, where $root$ is the root of $\mathcal{T}_{SP}$, returns $A = e(g,g)^{r_s s}$. Then computes:
$\widetilde{C}_u/(e(C_u, D_s)/A)$
$= M_u e(g,g)^{\alpha s}/(e(h^s, g^{(\alpha+r_s)/\beta})/e(g,g)^{r_s s}) = M_u$
to get the plaintext $M_u = \{n_1, n_2, a_{uI}\}$ generated by the SMD. The SP checks the freshness of nonces. If the request is valid, the SP replies to the MC with $transinfo$, where $transinfo$ includes $sessionID$, $userID$ and SP_ID that uniquely denotes the transaction between the user and the SP, to request attributes of the user for the following transactions. The SP sends the request to the MC.

$$SP \xrightarrow{\{n_1, transinfo, REQ-userinfo\}} MC.$$

4) The MC transfers the request to the App in the SMD.

$$MC \xrightarrow{n_1+1, transinfo} SMD.$$

5) The App checks $n_1 + 1$ and $transinfo$ so that it can confirm the response to setp 1 is correct, and then sends the request to the TEE embedded in the device. The TEE first calculates $D_0 = (H(a_{uI} \oplus n_1))^a$, and then generates a random number $t \in \mathbb{Z}_p$, calculates $\widetilde{D} = {D_u}^t$. After

that, the TEE sends the token $\{D_0, \widetilde{D}\}$ to the App. The App sends the token to the MC:

$$SMD \xrightarrow{\{D_0, \widetilde{D}, transinfo\}} MC.$$

This token is to provide evidence to the MC that the SMD confirms the first challenge-response with the SP and agrees to provide attributes.

6) The MC retrieves the user's attributes and gets a_uI and g^a, and checks whether $e(H(a_uI \oplus n_1), g^a)$ equals $e((H(a_uI \oplus n_1))^a, g)$. If they are equal, the MC authenticates the user's attempt for the device-specific transaction and retrieves required attributes and builds the attribute tree $\mathcal{T}_U$. As shown in Figure 2 (b), $\mathcal{T}_U$ is an all AND-gate tree and the root has at least two subtrees. Attributes on the left subtree include descriptive attributes of the user, while attributes on the right subtree is an attribute specific to the SMD, e.g. a_uI. The MC generates a random nonce $n_M \in \mathbb{Z}_p$ and sends to the SP.

$$MC \xrightarrow{\{\mathcal{T}_u, n_M\}} SP.$$

7) The SP checks whether attributes in $\mathcal{T}_u$ meets the $REQ - userinfo$, and then performs **Encrypt**$(PK, M_s, \mathcal{T}_u)$ as follows, where $M_s = \{n_2 + 1, n_M, sk\}$ and $sk \in \mathbb{Z}_p$ is a session key for later use:
   a) $\forall x \in \mathcal{T}_u$, randomly chooses a polynomial q_x with degree $d_x = k_x - 1$, where k_x is the secret sharing threshold value[8]. Chooses a random $s' \in \mathbb{Z}_p$ and sets $s' = q_{R_u}(0)$, where R_u is the root node of $\mathcal{T}_u$;
   b) Generates a ciphertext from top to down:
   $$\{\forall y \in Y_u : C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)}\},$$
   where Y_u is the set of leaf nodes in $\mathcal{T}_u$.
   c) Computes $C_{SP} = M_s e(g,g)^{\alpha s'}$ and $C_{SP} = h^{s'}$.
   d) Sends
   $$C_{\mathcal{T}_u} = \{\widetilde{C}_{SP}, C_{SP}, \forall y \in Y_u : C_y, C'_y\},$$
   to the cloud.

$$SP \xrightarrow{\{C_{\mathcal{T}_U}\}} MC.$$

8) The MC retrieves:

$$\widetilde{SK}_u = \langle \forall j \in Y_u : D_j = g^{r_u} \times H(j)^{r_j}; D'_j = g^{r_j} \rangle,$$

and runs the algorithm **Decrypt**$(\widetilde{SK}_u, C_{\mathcal{T}_u})$ and gets $A = e(g,g)^{r_u s'}$. The MC retrieves $\widetilde{D}$ received in the step (5), and computes

$$B = e(C_{SP}, \widetilde{D}) = e(h^{s'}, g^{t(\alpha + r_u)/\beta}) = e(g,g)^{t\alpha s'} \cdot e(g,g)^{t r_u s'}.$$

The MC sends $\{A, B, \widetilde{C}_{SP}\}$ to the SMD:

$$MC \xrightarrow{\{A, B, \widetilde{C}_{SP}\}} SMD.$$

9) The SMD sends $\{A, B, \widetilde{C}_{SP}\}$ to the TEE. The TEE finalized the last step for the decryption to get $M_s = \{transinfo, n_2, sk\}$ as follows:

$$M_s = \frac{\widetilde{C}_{SP}}{((B^{t^{-1}})/A)} = \frac{M_s e(g,g)^{\alpha s'}}{(e(g,g)^{\alpha s'} \cdot e(g,g)^{r_u s'})/e(g,g)^{r_u s'}}.$$

Till now, the SMD gets the second response to the challenge n_2 sent in the step 1. The TEE returns M_s to the App. After validating the n_2, the App encrypts $transinfo, n_2, a_uI$ using sk and sends them to the MC as follows :

$$SMD \xrightarrow{\{\{transinfo, n_2, a_uI\}_{sk}, n_M + 1\}} MC.$$

10) The MC checks n_M, and then sends to the SP.

$$MC \xrightarrow{\{transinfo, n_2, a_uI\}_{sk}} SP.$$

The SP decrypts the ciphertext using the session key it generated in step 7 and checks the plaintext. The SP confirms that the transaction is valid and launched by the user from the expected device after validating the message.

V. Performance Evaluation

In this section, we first provide security assessment of DRT, and then we evaluate the cost of DRT in terms of computation cost.

A. Security Assessment

In DRT, all entities in the system are considered semi-trusted. Entities only trust knowledges that they get from the protocol to set up trusts. In the follows we analyze knowledges entities can get and their potential misbehaviors to show the strength of DRT protocol. Due to the limits of space, the security analysis on the modified CP-ABE used in DRT can be refer to [19].

The user can confirm the transaction between her and the expected SP through two rounds of response-challenges. The user generates two nonce n_1, n_2 encrypted by the SP's attributes in step 1, and then get one response to n_1 in step 4 and the other response to n_2 in step 8. There is no way for others to fake responses except using secret keys related to the SP's attributes which we assume they are hosted in the SP securely.

Although a user is unlikely to cheat or counterfeit the tokens in the transaction she initiated, she may use other people's mobile device to launch transactions or pretends to be someone else using her own devices. In these cases, first, the MC can easily detect the misbehaviors since each user has registered and been bounded with a mobile device in the MC, second, we assume that the access to devices' TEE cannot bypass without a password.

The potential misbehaviors for the MC is that it may impersonate the user to launch transactions with the SP without users' consent. The MC may launch step 2 and 6, then gets messages from step 3 to 7 even without users' involvement, but it cannot fake the token in step 5 since it has no private key a and the device-specific secret key D. Even if the MC has the old blinded secret key $\widetilde{D} = g^{t(\alpha + r)/\beta}$, it cannot deduce D, since the blinded factor t is the exponent of the generator g, deriving it can be reduced to a Discrete Logarithm Problem that is considered to be hard. Without the token, the MC cannot decrypt the ciphertext in step 7.

The SP only communicates with the MC who is on behalf of the user, but it can confirm that it is the expected user on the specific device after the SP decrypts the ciphertext in step 10. The session key sk acts as a challenge to the user, only the user who has the associated secret keys can decrypt the ciphertext in step 7 to get sk. $\{transinfo, n_2, a_{uI}\}_{sk}$ is the user's confirmation to the online transaction.

To eavesdropping attacks, keys or credentials are never exposed in plaintext and nonces are used randomly only for one session in order to prevent attackers replaying the captured messages. The MC is authenticated by the TA through common authentication procedures without described in the DRT protocol, which is our assumption in that the communication between participating entities can be protected by SSL/TLS channels that can resist the man-in-middle attack.

B. Computational Performance Evaluation

In the setup procedure, the TA needs to generate secret keys that causes $2|A_u| + 3$ exponentiations on $\mathbb{G}_0$ for each SMD, and $2|A_s|+2$ exponentiations on $\mathbb{G}_0$ for each SP. These computations are only calculated for once in a period time and not a burden to the TA since we assume that the TA usually has powerful computational capability.

TABLE II
ONLINE CRYPTOGRAPHIC OPERATION COSTS.

	Exp $\mathbb{G}_0/\mathbb{G}_1$	Pairing	Mul $\mathbb{G}_1$	Hash
SMD	5 / 2	0	3	2
MC	$2a_1/a_2$	$2a_2+3$	$2a_2-1$	a_1
SP	$2a_2+1/a_1+2$	$2a_1+3$	$2a_1+3$	a_2

a_1 is the number of attributes in $\mathcal{T}_s$, a_2 is the number of attributes in $\mathcal{T}_u$.

Expensive cryptographic operations over $\mathbb{G}_0$ and $\mathbb{G}_0$ performed by entities in the DRT is shown in Table II. From Table II, we can see that the computation overhead is linear for the MC and the SP, who are considered have sufficient resources to process these operations. Pairing and exponentiation operations are considered the most computationally intensive among all computations. using DRT only a constant number of cryptographic operations are required, which is efficient and feasible for resource-constrained devices. The most expensive operations such as pairings are outsourced into the MC. Since the order of the computation overhead on smartcards is constant, the presented DRT scheme shows significant performance gain when the attribute tree is large, i.e., all the attribute-related operations are outsourced to the cloud side.

VI. CONCLUSION

It becomes a global trend to use mobile devices as a platform for secure transactions in different domains. The dual root trust model proposed in this paper provides confirmation for both users and SPs in each transaction between users and SPs, and it provides trust from both mobile devices and the mobile cloud. In the future, we will work on extending our work to provide a detailed performance evaluation using the TrustZone embedded mobile devices and enhance dual-root trust level relying on the trusted execution environment and the mobile cloud.

VII. ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (Grant No. 61003185). The research of Dijiang Huang is sponsored by ONR YIP award.

REFERENCES

[1] S. C. Alliance, "Mobile devices and identity applications," September 2012.

[2] A. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, 2011, pp. 3–14.

[3] D. Huang, "Mobile cloud computing," in *Proccedings of the sixth conference on computer systems*. ACM, 2011, pp. 301–314.

[4] P. M. M. N. B. G. Chun, S. Ihm and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *IEEE COMSOC Multimedia Communicatoins Technical Committee (MMTC) E-letter*, vol. 6(10), Octorber 2011, pp. 27–31.

[5] Google Inc., "Google Wallet," in *available at http://www.google.com/wallet/*.

[6] M. Wu, G. S., and R. Miller, "Secure web authentication with mobile phones," in *DIMACS workshop on usable privacy and security software*, 2004, pp. 9–10.

[7] EMC, "Rsa secureid," in *available at http://www.emc.com/security/rsa-securid.htm*.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, ser. SP '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 321–334.

[9] ARM, "Trustzone technology overview," 2009.

[10] D. Davenport, "Nexus S Enables the NFC Secure Element," in *available at http://thinkd2c.wordpress.com/2011/07/07/nexus-s-enables-the-nfc-secure-element/*, 2011.

[11] A. Filyanov, J. M. McCuney, A. Sadeghiz, and M. Winandy, "Uni-directional trusted path: Transaction confirmation on just one device," in *Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems&Networks*, ser. DSN '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–12.

[12] S. Balfe and K. G. Paterson, "e-emv: emulating emv for internet payments with trusted computing technologies," in *Proceedings of the 3rd ACM workshop on Scalable trusted computing*, ser. STC '08. New York, NY, USA: ACM, 2008, pp. 81–92.

[13] S. Bugiel, A. Dmitrienko, K. Kostiainen, A. R. Sadeghi, and M. Winandy, "Truwalletm: Secure web authentication on mobile platforms," in *Proceedings of 2nd International Conference on Trusted Systems (INTRUST)*, 2011, pp. 219–236.

[14] J. Azema and G. Fayad, "M-shield mobile security technology: making wireless secure," 2008.

[15] K. Kostiainen, J. Ekberg, N. Asokan, and A. Rantala, "On-board credentials with open provisioning," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ser. ASIACCS '09. New York, NY, USA: ACM, 2009, pp. 104–115.

[16] M. Nauman, S. Khan, A. T. Othman, S. U. Rehman, and N. U. Rehman, "Poauth: privacy-aware open authorization for native apps on smartphone platforms," in *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*, ser. ICUIMC '12. New York, NY, USA: ACM, 2012, pp. 60:1–60:8.

[17] A. Tassanaviboon and G. Gong, "Oauth and abe based authorization in semi-trusted cloud computing: aauth," in *Proceedings of the second international workshop on Data intensive computing in the clouds*, ser. DataCloud-SC '11. New York, NY, USA: ACM, 2011, pp. 41–50.

[18] F. J. Krautheim, "Building trust into utility cloud computing," Ph.D. dissertation, University of Maryland, 2010.

[19] Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in *Proceedings of 8th International Conference on Network and Service Management*, 2012.