Secure Data Processing Framework for Mobile Cloud Computing

Dijiang Huang, Zhibin Zhou, Le Xu, Tianyi Xing, Yunji Zhong
Arizona State University

Abstract— **In mobile cloud computing, mobile devices can rely on cloud computing and information storage resource to perform computationally intensive operations such as searching, data mining, and multimedia processing. In addition to providing traditional computation services, mobile cloud also enhances the operation of traditional ad hoc network by treating mobile devices as service nodes, e.g., sensing services. The sensed information, such as location coordinates, health related information, should be processed and stored in a secure fashion to protect user's privacy in the cloud. To this end, we present a new mobile cloud data processing framework through trust management and private data isolation. Finally, an implementation pilot for improving teenagers' driving safety, which is called FocusDrive, is presented to demonstrate the solution.**

Keywords: Security, Privacy, Mobile Cloud, Data Security

I. Introduction

The use of mobile devices to establish ad-hoc communication systems is a viable solution that provides global connectivity to support a broad range of applications. With the development of wireless access technologies such as 3/4G, LTE, and WiMax, mobile devices can gain access to the network core over longer distance and higher bandwidth. This allows for very effective communication between mobile devices and the cloud infrastructure. A new secure mobile cloud service architecture is necessary to address the requirements of users in their unique operational environment. In general, mobile users can be benefited greatly from cloud services for computationally intensive information processing and collection such as information search, data processing, data mining, network status monitoring, field sensing, etc.

In [1], we present a secure mobile cloud computing framework, called MobiCloud, which transforms traditional MANETs into a new service-oriented communication architecture, in which each mobile device is treated as a Service Node (SN), and it is mirrored to one or more Extended Semi-Shadow Images (ESSIs) in the cloud in order to address the communication and computation deficiencies of a mobile device. In MobiCloud, a mobile device can outsource its computing and storage services to its corresponding ESSI and Secure Storage (SS). Moreover, the device will send its sensed information such as moving trajectory to the cloud. As a return, the cloud can provide better location-based services according to the mobility information provided by the mobile device. In MobiCloud [1], mobile users must trust the cloud service provider to protect the data received from mobile devices. However, it is a big concern for mobile users for storing their privacy sensitive information in a public cloud. This paper targets to address this privacy issue.

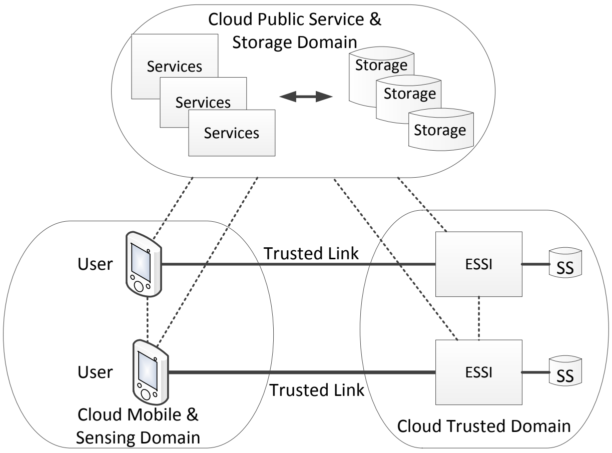


Fig. 1. Reference Service Model of mobile cloud.

The proposed new secure data processing mobile cloud infrastructure is highlighted in Figure 1, The mobile cloud is composed by three main domains: (i) the cloud mobile and sensing domain, (ii) the cloud trusted domain, and (iii) the cloud public service and storage domain. In this framework, each mobile device is virtualized as an ESSI in the cloud trusted domain and each ESSI can be represented as an SN in a particular application (a.k.a., a service domain). The introduced ESSIs can be used to address communication and computation deficiencies of a mobile device, and provide enhanced security and privacy protections. A mobile device and its corresponding ESSI can also act like a service provider or a service broker according to its capability, e.g., available computation and communication capabilities to support a particular communication or sensing service. This approach takes maximum advantage of each mobile node in the system by utilizing cloud computing technologies. In this way, the cloud's boundary is extended to the customer device domain. Note that an ESSI can be an exact clone, a partial clone, or an image containing extended functions of the physical device. The networking between a user and its ESSI is through a secure connection, e.g., SSL, IPSec, etc.

We must note that the presented research work is unique. Although the presentation is based on the MobiCloud framework presented in [1], to the best of our knowledge, we cannot identify existing comparative research work in the same field. In the following sections, we first present the mobile cloud secure data processing model in Section II. Particularly, the presentation focuses on trust management, multi-tenant secure data management, and ESSI data processing model. In Section III, we present a proof-of-concept pilot – FocusDrive, and describe the future work.

II. MOBILE CLOUD SECURE DATA PROCESSING MODEL

An ESSI is a virtual machine that is designed for an end user having full control of the information stored in its virtual hard drive. However, the networking functions and running processes are customized through the mobile cloud service provider. Note that the cloud trusted domain and cloud public service and storage domain are physically isolated to provide strong security protection to user's data. They can belong to two different cloud service providers.

Within the cloud trusted domain, strict security policies are enforced through a distributed Firewall system (i.e., each ESSI runs its own Firewall). Data flows in/out the trusted domain must be scanned through the distributed Firewall system to make sure no malicious traffic is sent/received. The mobile cloud data processing model includes three main components: trust management, multi-tenant secure data management, and ESSI data processing model, which are described in details in the following subsections.

A. Mobile Cloud Trust Management

The trust management model of mobile cloud includes identity management, key management, and security policy enforcement. An ESSI owner has the full control over the data possessed in the ESSI, and thus a user-centric identity management framework is a natural choice. The user-centric identity management (also frequently referred to as identity 2.0 [2]) allows an individual has full control of his/her identities, in which third party authenticates them. It also implies that a user has control over the data his/her sharing over the Internet, and can transfer and delete the data when required. In this paper, we introduce an integrated solution involving identity-based cryptography [3] and attribute-based data access control [4] as the building blocks to construct the trust management system for mobile cloud. Particularly, the presented mobile cloud communication framework usually involves the establishment of a virtual private communication group.

1) Mobile Cloud Identity and Trust Management: A Trusted Authority (TA) is assumed to manage security keys and certificates for mobile users. In the following presentation, without special notice, we always assume that there is a TA available, which is responsible for key and certificate distribution. Based on this assumption, the TA is responsible to deploy an Attribute-Based Identity Management (ABIDM) for mobile cloud's identity and trust management. The basic identity representation of ABIDM is shown in Figure 2. Using ABIDM, we first need to define the "point of network presence (PoNP)". A mobile node's relationship can be thought of as lines radiating from the PoNP to the various counterparties. Each line is distinct and tagged with the attribute used by a particular counterparty. In particular, we define a default PoNP (i.e., native PoNP) for everyone. The default PoNP has to be linked by a unique native ID. The uniqueness of the native ID is not difficult to achieve. Indeed, any user can have a unique native ID by simply hashing any one of his/her unique identifiers, such as a driver license ID, email address, social security number, etc. Each PoNP has two properties: type and value. The type value provides the information such as (i) the identity issuer, (ii) the private key issuer, and (iii) the validation period. The identity and private key issuer can be either self-generated or derived from a Trusted Authority (TA). The value of the PoNP can be used as a part of the user's identity with its type for a particular scenario. Identity-based cryptography can be used, and a private key is assigned to the PoNP identity. A message receiver can use the sender's identity to verify the received signature for authentication purpose.

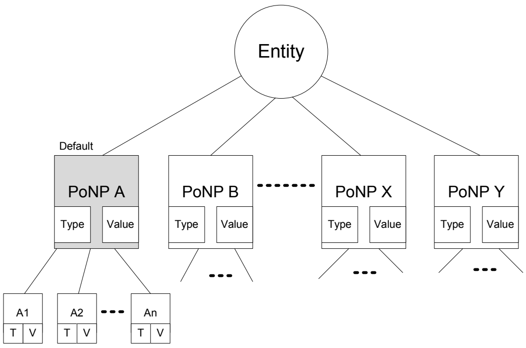


Fig. 2. Identity representation scheme.

Each PoNP is associates with one or multiple attributes (i.e., A1...An), and each attribute has type and value properties. Attributes can be assigned as predefined attributes that do not change frequently, which are called as static attributes. To differentiate PoNPs, the number of attributes can be reduced for each PoNP for later secure communication.

The major benefit of using this identity representation is the "standardization" of identity management. In practice, the numbers of PoNPs for every mobile node can be restricted to a certain number of known scenarios. This can be done by consulting the TA, which provides the public key certificate services and provides the ontology of identities and attributes for mobile users. Self-managed identity and private keys can be used to form a self-managed and trusted ecosystem. This feature will be useful for managing trust based on social network applications.

2) An Example of mobile cloud Trust Management: The basic identity representation of ABIDM is shown in Figure 2. An ABIDM example is presented as follows:

Identity Representation:

PoNP:{Native},Type: {ID, TA, ExpDate}, Value:{011};

A_0:{attribute},Type: {name}, Value:{identity− >David_Kurt};

A_1:{attribute},Type: $\{\bar{B}_0\}$, Value:{0};

A_2:{attribute},Type: $\{B_1\}$, Value:{1};

A_3:{attribute},Type: $\{B_2\}$, Value:{1};

...

PoNP:{identity},Type: {name, Self-Gen, ExpDate},
Value:{David_Kurt};
A_j:{attribute},Type: {organization},
Value:{ASU-Fulton-CIDSE-CSE};
A_{j+1}:{attribute},Type: {device}, Value:
{communication− >Mac_address− >01:23:45:67:89:ab};
A_{j+2}:{attribute},Type: {email}, Value: {David.Kurt@asu.edu};
...
PoNP:{communication},Type: {Mac_address, TA, ExpDate},
Value:{01:23:45:67:89:ab};
A_k:{attribute},Type: {organization},
Value:{ASU-Fulton-CIDSE-CSE};
A_{k+1}:{attribute},Type: {owner}, Value:{David_Kurt};
A_{k+2}:{attribute},Type: {device_model}, Value:{iPhone_3G};
...

The first PoNP is "Native", in which its value is unique for each entity. Attribute A_0 usually points to other PoNPs. The number of bits for the ID value should be long enough to guarantee that every entity will have a unique value, where B_x represents the bit at position x from the leftmost side. For demonstration purposes, three bits are used for the native ID value. The second PoNP describes the identity "David Kurt" and his associated attributes; the attribute "organization" describes where David works; the device attribute points to another PoNP "communication"; email is another attribute for David. The third PoNP is "communication", the entity is represented by a MAC address and the attribute "owner" describes who owns this device. In this example, attributes for different PoNP can be overlapped; on the other hand, an attribute in one PoNP may not exist in another PoNP. It also shows that ABIDM can easily integrate both the organization-centric IDM and user-centric IDM by considering an organization as an attribute for a user. In [5], the authors proposed to use data objects to represent attributes and use name objects to construct name graphs, where a name graph is rooted by the user's name and directional links are pointed from the user's attributes. ABIDM introduces a user graph approach, where directional links link PoNP to its attributes. An attribute can also point to another PoNP (e.g., $\{communication- > Mac\_address- > 01:23:45:67:89:ab\}$). From the identity management perspective, "⋆.David_Kurt" can be used as an entity. However, it might not be meaningful in a mobile cloud since there may be many individuals called David Kurt. Thus, practically, more attributes values can be involved, such as:

$$\star.David\_Kurt|ASU-Fulton-CIDSE-CSE$$

or

$$\star.David\_Kurt.01:23:45:67:89:ab,$$

where "." represents a pointer to next PoNP attribute value and "|" separates attributes within the same PoNP. $\star$ represents the bit-assignment values in the native PoNP, e.g., $\star = h(David\_Kurt|ASU-Fulton-CIDSE-CSE) = 011$ and $h()$ is publicly known hash function. In this example, David Kurt may have multiple native values (e.g., another is derived from $\star\star = h(David\_Kurt.01:23:45:67:89:ab) = 101$). The user needs to derive the private key for its native ID from the TA for later secure communication.

Attribute-based group formation and private key generation:

In the mobile cloud communication environment, a secure communication session can be either one-to-one or one-to-many [1] (e.g., an ESSI wants to share a picture with several ESSIs, which form an ad hoc group). In the terminology of secure group communication, these communication patterns can be represented as group (or subgroup) communication. Thus, a shared key needs to be established among group members. In literature, a secure group communication includes 3 phases [6]: (i) secrets pre-distribution, (ii) group key update, and (iii) secure group communication. Phase (i) can be done offline before sending the encrypted data. Based on current hardware/software solutions, phase (iii) can be processed very quickly. Thus, the main bottleneck is in phase (ii). To address this bottleneck, the design goal is to reduce the group-based key management overhead and support efficient key distribution in a dynamic communication environment, where the communication peers may keep on changing.

Attribute Based Encryption (ABE) [4], [7] had been proposed for data encryption and decryption. ABE is an extension of IBE scheme in that multiple public known attributes as the public key. Using threshold secret sharing scheme [8], the encryptor can construct an data access policy by forming an encryption policy tree, where leaf nodes are attributes and the internal nodes are logical gates such as "AND" and "OR". To integrate ABE with the presented identity management scheme, we present an novel approach that each bit in a native ID can be assigned to a unique attribute, and the attribute does not need to be meaningful (i.e., can be any randomly generated string. Due to page limit, interested readers can refer to [7] for more details). Based on the previously presented example, assume that David Kurt needs to communicate with a group of routers with addresses $\{001, 011, 100, 101, 110, 111\}$, in which users $\{000, 010\}$ are not included. A naïve approach is to construct an access tree by using an "OR" (or $+$) logic at the root. To reduce the number of involved attributes, a membership function (M) using Boolean Function Minimization (BFM) approach is used:

$$\begin{aligned} & M_{\{001,011,100,101,110,111\}} \\ &= \bar{B_1}\bar{B_2}B_3+\bar{B_1}B_2B_3+B_1\bar{B_2}\bar{B_3}+B_1\bar{B_2}B_3 \\ &\quad +B_1B_2\bar{B_3}+B_1B_2B_3 \\ &= B_0+B_2, \end{aligned}$$

where B_i and $\bar{B_i}$ represent the bit values "1 and 0", respectively, at position i. The final group access tree only involves two attributes assigned to B_0 or B_2 to secure the encrypted data, in which both group members 000 and 010 are not able to access.

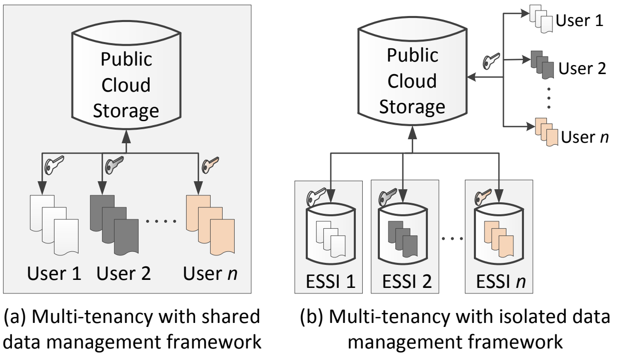
Bootstrap of Secure Communication Group:

ABIDM uses identity-based signature schemes for authentication and attribute-based encryption scheme for data access control. A mobile user uses ABIDM to establish a virtual private communication service among ESSIs. The goal of ABIDM is to establish a common sharing group key among a selected group of users (or ESSIs). Based on the presented example, the bootstrap of the secure group communication using ABIDM is presented as follows:

- Sender 011 would like to establish a virtual private communication group {001,011,100,101,110,111}.
- Sender 011 uses BFM to derive the group-based attribute composition as $B_0 + B_2$, where "+" is a logical OR gate.
- Sender 011 uses attribute-based encryption scheme [4] to encrypt a group key k_s.
- Sender 011 generates the signature for the encrypted message based on identity-based signature scheme (e.g., [9]) using the private key of identity "011".
- Sender 011 sends the encrypted group key and its signature to receivers.
- Each receiver verifies the signature and decrypts the group key k_s. Note that the verification is based on the sender's ID (i.e., 011) and see if it satisfies the logic $B_0 + B_2$.
- Then, the group-based secure communication session begins.

B. Multi-tenant Secure Data Management

As shown in Figure 1, the dashed lines represent the ad hoc connection between entities, and the solid lines represent dedicated secure connections. The cloud public service and storage domain provides services for all mobile devices and ESSIs. A mobile device can request services directly from the public service and storage domain, or it can request services through its ESSI. An ESSI is the security policy enforcer for its associated mobile device(s). The user can specify what data should be protected and stored in its ESSI. Users' private information is maintained in their corresponding Secure Storage (SS).



(a) Multi-tenancy with shared data management framework

(b) Multi-tenancy with isolated data management framework

Fig. 3. Multi-tenant Secure Data Management in MobiCooud.

Multi-tenancy is one of the key features of cloud services. To secure each user's data, traditional approaches are shown in Figure 3(a), where users' data is stored in one big database and a unique encryption key is used to secure data for each user. This approach has several drawbacks. First, it is not scalable when the database is huge. Data storage operations can incur heavy data operations that require extensive computing resources. Second, data encrypting keys for users are maintained in a centralized location, which is vulnerable to the single-point failure problem. Moreover, users usually have concerns that the cryptographic key is maintained by the cloud provider. To address these drawbacks, the presented solution utilizes a decentralized approach, which is presented in Figure 3(b). The proposed multi-tenant data management system partition the data into two security levels: (i) critical data and (ii) normal data. The critical data must be secured by the data encrypting key generated by the user, and the normal data is secured by the data encrypting key generated by the cloud storage service provider. The presented multi-tenant secure data management system can address the drawbacks of traditional approaches. First, the data operations such as indexing, data retrieval, data addition, etc., are distributed to ESSIs. In addition, the security functions, such as encryption/decryption/integrity, are also distributed to ESSIs. As a result, the computation overhead is distributed to multiple processors in the cloud system. Second, ESSIs enhance the users' security by adding one additional layer of security, in which the critical data are stored in each ESSI. As a result, compromising one ESSI will not impact other ESSIs.

C. ESSI Data Processing Model

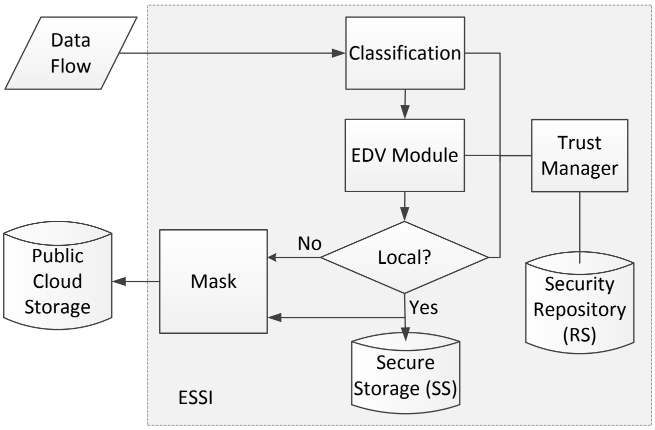


Fig. 4. Data processing in ESSIs.

The ESSI's data processing model is built on the security capability model enabled for Linux Kernel 2.2 and above [10]. Based on the security capability model, we can build a Tri-rooted ESSI that has a cloud root, a user root, and an auditing root. The privilege of the user root includes maintaining user's data in its SS and encryption/decryption/verification related processes. The cloud root is to perform the maintenance functions of ESSI, and it does not have the access to SS and related security functions. The auditing root is used to log the activities of both cloud root and user root. The log data can be only accessible for investigation purpose when regulation violations are identified. Usually, the log data is maintained by a third trusted party. In this way, the cloud provider cannot easily breach the privacy of users.

The ESSI's data processing model is presented in Figure 4. SS is installed in ESSI's virtual hard drive. A user's private information and security credentials are stored in the Security Repository (RS) managed by the ESSI mapped to the user's mobile device. The critical data is stored in the SS. Data flow arriving at the ESSI is processed as follows: (i) Data flow is inspected by the classification model that classifies the data as critical data or normal data. (ii) If the data is classified as normal, the normal data will be sent to the public cloud storage through a masking procedure. (iii) The Encryption/Decryption/Verification (EDV) module is then used on the critical data and stores the processed data in SS. The masking procedure is used to remove private information associated with the user and anonymize the data content. The masking procedure can be configured differently according to the level of the criticality of the data. It is up to the user's preference, and it is operated through the trust manager. For example, ESSI can generate a masked index value for the public cloud storage for indexing purpose. This index value includes the ESSI's identifier (can be a pseudonym) and corresponding indexing category. Once the public cloud storage service receives the index value, it then uses it to identify which ESSI is responsible for the requested searching data.

III. Discussion and Future Work

We have developed a pilot mobile cloud system [11] to implement the cloud trusted domain as presented in Figure 1. To demonstrate the presented security and privacy protection features, we have developed a pilot application "FocusDrive" project [12], which is presented in the following subsection.

A. FocusDrive Project

The FocusDrive project is conducted by the Secure Networking And Computing (SNAC) research group at ASU. Studies show teenagers are especially prone to text-and-drive, which can be as dangerous as DUI (Drive Under Influence). The goal of FocusDrive is to improve the driving safety of teenage drivers in collaboration with their parents. Particularly, it focuses on restricting improper usage of cell phone texting while teenagers are driving. FocusDrive develops an application running in mobile phones as a background application to dynamically monitor the speed of the phone. Running this application, a cellphone will automatically enable and disable the texting function according to the driving speed and road conditions.

FocusDrive involves the Microsoft Azure cloud computing platform as the public cloud that provides realtime traffic information from BingMap API and performs location tracking on the geographic map. It also includes our MobiCloud platform as the cloud trust domain to protect users' privacy, i.e., the location traces of teenagers. On a cell phone, once the speedchecker detects the moving speed of the cellphone above a certain threshold set by the teenager' parents, the FocusDrive application will communicate with the ESSI in the MobiCloud and periodically update its GPS location in the SS. At the cloud side, the parents can check their children's location (get the map service from Bing Map) and control their children's cell phone texting function accordingly. The end users (both the teenager and parents) can select to upload their moving trajectory into the Azure's storage for realtime traffic monitoring applications that running in the public cloud domain. Before sending the location information to the public cloud, the ESSI can select an anonymized identity with a certificate issued by the TA to authenticate the trace data. In this way, user's privacy is protected and security is also provided to prevent fake information. We must note that the identity certificate can be used for revocation in case misuses are detected. The FocusDrive has proven the security and privacy functionalities presented in this paper.

B. Future Work

In this paper, we present a prototype of the secure data processing model for mobile cloud computing. In the future, we will focus on the follow research: (i) investigate more application scenarios that require data sharing between cloud private domain and public domain; (ii) investigate the robustness of the Tri-rooted ESSI solution; and (iii) investigate the security monitoring, auditing, and misuse detection in the mobile cloud system.

References

[1] D. Huang, X. Zhang, M. Kang, and J. Luo, "Mobicloud: A secure mobile cloud framework for pervasive mobile computing and communication," in *Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering*, 2010.

[2] "Identity 2.0," http://en.wikipedia.org/wiki/Identity_2.0.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. of Computing*, no. 3, pp. 586–615, 2003.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," *Proceedings of the 28th IEEE Symposium on Security and Privacy (Oakland)*, 2007.

[5] J. Su, J. Scott, P. Hui, E. Upton, M. Lim, C. Diot, J. Crowcroft, A. Goel, and E. de Lara, "Haggle: Clean-slate networking for mobile devices," *Technical Report, UCAM-CL-TR-680, University of Cambridge*, 2006.

[6] D. Huang and D. Medhi, "A Key-chain Based Keying Scheme For Many-to-Many Secure Group Communication," *ACM Transactions on Information and System Security*, vol. 7, no. 4, pp. 523 – 552, 2004.

[7] Z. Zhou and D. Huang, "An optimal key distribution scheme for multicast group communication," in *IEEE Conference on Computer Communications (Infocom)*, 2010.

[8] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[9] P. Barreto, B. Libert, N. McCullagh, and J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," *Advances in Cryptology-ASIACRYPT 2005*, pp. 515–532, 2005.

[10] D. Ducamp and H. Schauer, "Linux's Security Capabilities," available at http://www.hsc.fr/ressources/presentations/linux2000/linux2000.htm.en, 2000.

[11] Secure Networking And Computing Research Group (SNAC), "MobiCloud," available at http://mobicloud.asu.edu, 2010.

[12] ——, "Focusdrive project," available at http://www.snacwiki.asu.edu/focusdrive.