2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing

TGIS: Booting Trust for Secure Information Sharing in Mobile Group Collaborations

Katharine Chang[1], Xinwen Zhang[2], Guoqiang Wang[2], and Kang G. Shin[1]
[1] The University of Michigan, Ann Arbor, MI, {katchang, kgshin}@eecs.umich.edu
[2] Huawei Research Center, Santa Clara, CA, {xinwen.zhang, gq.wang}@huawei.com

***Abstract*—Mobile computing has drawn considerable attention because of the various types of mobile devices and services that are becoming available. This paper explores dynamic group collaboration and information sharing with mobile devices, such as smartphones and tablets. In particular, we propose trusted group-based information sharing (TGIS), a protocol for mobile devices to establish a trust relationship in order to form group-based information sharing. We exploit existing (group or organizational) identity hierarchies of mobile users to establish trust between group members with hierarchical identity-based encryption (HIBE). In order to control information sharing within a group and between groups, we further leverage attribute-based encryption (ABE) for secure access control, where attribute secret keys are distributed with the trust relationship with HIBE. We have implemented and evaluated TGIS on Android phones, demonstrating its viability.**

I. Introduction

Over the past decade, mobile computing has drawn much attention because of the various types of mobile devices (e.g., smartphones) and communication protocols (3G/4G and WiFi) that are becoming available. Even though mobile devices have been increasingly used for entertainment and social applications over the last few years, there has been an important oversight: more mobile applications will appear at public safety, healthcare, and even military facilities/sites. For example, military has started to use smartphones in the battlefield for communication and collaboration purposes. Of these trends, here we explore the use of mobile devices for dynamic group communications among them. For example, soldiers from different units form a group for a particular task. Or, agents of local police offices, the Department of Homeland Security (DHS), and FBI dynamically react to an accident in a local area.

Several security requirements need to be met for dynamic group communication and information sharing with mobile devices. For example, devices must be able to securely communicate and collaborate with one another within a group. That is, the information shared in one group may only be accessible to its group members, e.g., authorized by a group leader. Furthermore, even within a group, information may not be freely shared by all group members, e.g., due to different security levels, expertise, and job duties. In addition, inter-group communication is necessary in many scenarios. For example, one soldier in a military unit may want to request assistance from another unit on identifying some (enemy) weapons, while not wanting to share the information with every soldier in that unit. We find it to be a common requirement in many dynamic group communications, such as healthcare and first responders. All of the above-mentioned security requirements require trust management between group members.

Aiming to bootstrap trust for dynamic group-based information sharing and access control, we propose and implement trusted group-based information sharing (TGIS). TGIS is a distributed security protocol built upon existing trust infrastructures in individual organizations to enable trust management for group collaborations. We assume that each device belongs to one organization, which has implemented mechanisms to deploy credentials for trust management of users within its organization. We then leverage the user identity hierarchies to establish trust between group members by exploiting hierarchical identity-based encryption (HIBE) [1]. Specifically, a group leader can use a user's hierarchical identity as a public key to distribute group keys. For controlling information sharing within a group, we use attribute-based encryption (ABE) [2] for secure access control, where the group leader defines group-wide attributes, generates attribute secret keys, and distributes them to individual group members. By exposing public information of a group in an authentic manner, users in other groups can also send information to users in the group with controlled sharing.

The remainder of this paper is organized as follows. We first present the motivation and applications of TGIS in Section II. In Section III, we review the cryptographic primitives in TGIS. We then give an overview of the TGIS design in Section IV and describe the details of TGIS in Section V. Section VI describes our implementation and performance evaluation. Finally, we discuss related work in Section VII and conclude the paper in Section VIII.

II. Motivation

The main goal of TGIS is to let users use their mobile devices to establish a trust relationship to collaborate and communicate with their collaborators, and to have access control over their shared information among the collaborators.

First responders are trained rescuers that would go to emergency scenes and perform search-and-rescue. Taking the example of an earthquake, first responders perform search-and-rescue in the disaster. The emergency medical services, fire departments, police departments, and DHS agents will all collaborate to assist with the recovery efforts. TGIS allows the first responders from different organizations to collaborate

978-0-7695-4578-3/11 $26.00 © 2011 IEEE
DOI

in a secure manner and establish a trust relationship with one another. The first responders will be able to authenticate themselves and join a dynamic collaboration rescue team for disaster relief. On the other hand, news reporters and others may not be able to successfully authenticate themselves to join the rescue team and access the shared information. In the rescue team, there are classified information that only people with the appropriate level of clearance can view. Such information might only be accessible to polices or DHS agents, and TGIS helps with data access control in such a scenario.

Another application of TGIS is for the collaboration of military soldiers in the battlefield. For example, in a military task where a scout unit is sent out to detect if there are weapons or mines in the battlefield, or to eavesdrop or intercept on the enemy's conversation, the scout unit would consist of weapon specialists and soldiers with different ranks. TGIS can be used for the scout unit members to form a dynamic collaboration coalition using their mobile devices. It would save device processing time and power if the devices in the scout unit can collaborate or offload heavy computation programs to other devices without their information been eavesdropped by the enemies. With TGIS, unit members can share information to other members with the appropriate level of clearance and help execute programs for other members, like helping to execute the translation program on the eavesdropped enemy conversation.

As for inter-group collaboration, group members may want to share information between users in different groups. First responders share information about different scenes of accidents, and only members with the appropriate clearance can read certain classified information. Soldiers exchange information in the battlefield regarding specific tasks, and only soldiers with the appropriate clearance for the task can read the information. In our scout unit example, weapon specialists may exchange related intelligence between different scout units and only weapon specialists are authorized to view any confidential information. Our proposed TGIS protocol is designed to be used in all of the above scenarios.

III. Cryptographic Primitives

In this section, we review the cryptographic primitives we use for the construction and design of TGIS.

A. Hierarchical Identity-Based Encryption

Identity-based cryptography (IBC) was first introduced by Adi Shamir in 1984, who implemented identity-based signature (IBS) and proposed identity-based encryption (IBE) in [3]. However, IBE was not realized until 2001 by Boneh and Franklin [4], and then by Cocks [5]. IBC is a type of public-key cryptography. In IBC, a public identity is used as a public key string to simplify certificate management in public key infrastructure (PKI). The public identity could be an email address, phone number, or a hierarchical identity within an organization. IBC is different from traditional PKI, where an entity (e.g., a user or a host) generates its public/private key pair and obtains public key certificate from a certificate authority (CA). In IBC, the private key is generated by a trusted third party called the private key generator (PKG) with its corresponding identity and system parameters.

More specifically, in an identity-based system, a PKG generates a master secret key (MSK) and public system parameters (SP). The MSK is kept as a secret and used by the PKG only to generate corresponding private keys for individual users, and the SP is published publicly. Any user can use the published SP and the publicly known user identity to generate public keys for other users.

Based on IBE, hierarchical identity-based encryption (HIBE) [1], [6] was introduced to create hierarchies of PKGs and allow higher-level PKG to control the keys given to its subordinate lower-level PKGs. HIBE allows root PKG to distribute private key computation workload to lower-level PKGs and ease the private key distribution problem and improve scalability. It also removes a single-point of failure and the disclosure of a lower-level PKG's secret will not compromise higher-level PKGs' secrets or other parts of the hierarchy.

B. Attribute-Based Encryption

Attribute-based encryption (ABE) enables complete access control on encrypted data by specifying the expressive access policies/rules in private keys and ciphertexts [2], [7]. There are two categories of ABE, the ciphertext-policy ABE (CP-ABE) [2] and key-policy ABE (KP-ABE) [7]. In CP-ABE [2], the private keys are associated with a set of attributes, and messages are encrypted to access policies which specifies what private keys with the desired attributes will be able to decrypt the ciphertexts. Whereas in KP-ABE [7], it's the ciphertexts that are associated with sets of attributes and the private keys are associated with the access policies.

We use CP-ABE in TGIS for ensuring access control in data sharing. In CP-ABE, a user will specify an access tree structure of access policy for the message to be encrypted. Only if another user with a private key that is associated with the desired attributes will be able to decrypt the ciphertext.

IV. Overview of TGIS

Trust relationship between entities indicates that an entity has certain assurance that it can share data with another entity without releasing information to any other entity. This is typically achieved by identity authentication, shared keys, and data encryptions. To establish trust relationship among users, we first bootstrap trust in dynamic groups for secure collaboration and communication, and then enforce access control for data sharing. To bootstrap trust among group members, we leverage the existing organization identity hierarchies to establish trust between group members and let a group creator/leader generates the private keys for group members and securely distributes the keys using HIBE. Since group members have different privileges, we use CP-ABE for secure access control within a group and also among different groups.

A. System Architecture and Assumptions

Our system consists of users carrying mobile devices for communication and collaboration with other users in the

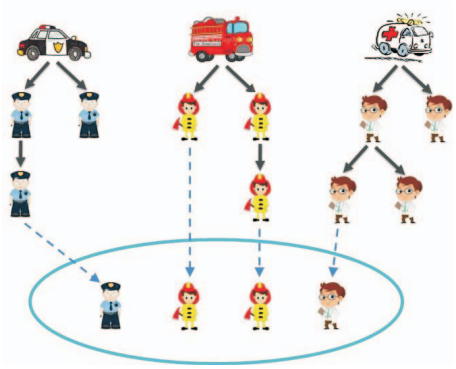

Fig. 1. System architecture of users in different organizations forming a dynamic group.

TABLE I
NOTATIONS USED.

Term	Notation
$a, b, \ldots$	entities such as users/devices
ID_a	identity of user a
$A, B, \ldots$	domains/organizations or dynamic groups
HSP_A	HIBE system parameters for domain A
HMS_A	HIBE master secret for domain A
$A.HPK_a$	HIBE public key for user a in domain A
$A.HPrK_a$	HIBE private key for user a in domain A
$AMSK_A$	ABE master secret key for group A
APK_A	ABE public key for group A
$A.ASK_a$	ABE secret key for user a in group A
S_A	attributes set of group A
$A.attr_a$	attributes set assigned to user a in group A, and $A.attr_a \subset S_A$
T	access tree built with logical expression over attributes

network. Each user belongs to an existing hierarchical domain organization. The identity of a user/device is a hierarchical domain structure and is unique. For each user, the user's hierarchical identity is the concatenation of the identities from the root to the user. For example, Alice in the Surveillance Unit in the Police Department will have "Police/Surveillance/Alice" as her ID.

Here we use HIBE as described in Section III-A for the security basis. We assume each domain/organization has a hierarchical architecture and each intermediate user is a private key generator (PKG) that is responsible for assigning private keys for its subordinate users. The intermediate users are different levels of managers or authorizers in an organization. The top level is the root PKG that is responsible for generating the public known system parameters (SP). Each user gets his private key from their immediate upper level PKG. There is no private/public key pair for each user and instead, user identity is used as the public key in HIBE. A nice property of HIBE is only the domain SP and user ID is needed in order for others to generate the user's public key. It is very flexible and one does not need to know the user's intermediate PKG's public key to generate the user public key.

We assume users create dynamic groups for different events and purposes. In a dynamically formed group, the group creator (or leader) controls access to data and user privileges in the group and generates corresponding group private keys for each group member. Users can share information with other members in the same group or even with users in other groups. Group members can be from different organizations to form a dynamic group. Therefore, the group leader acts as the PKG of the group so that it generates and distributes group private keys for other members.

We assume that when users communicate in a group, they can either use the existing base stations or setup mobile routers when they are in the wilds and no base stations are available. For example, femtocell [1], which is a small cellular base station designed for use in a home or small business, can also be setup for group communication. Figure 1 presents the system architecture of users in different organizations forming a dynamic group using a mobile router.

[1]http://www.femtoforum.org/femto/

B. Attack Model

In our attack model, attackers can eavesdrop on the communication channel between users and can also replay, spoof, or insert false data into the network. Also, attackers can masquerade as legitimate users to join group collaboration. Furthermore, an attacker can be a group member such that it tries to access and propagate data without proper privileges. The attacker can also launch Sybil attack and fake to be multiple identities in the network.

V. TGIS PROTOCOL

The TGIS protocol consists of five phases: offline domain setup, group setup, user enrollment, intra-group communication, and inter-group communication. This section describes the details of each. Table I lists the notations used in the rest of this paper.

A. Domain Setup

Before deployment, the mobile devices are in the offline domain setup phase, and are all assumed to be secured. In this phase, each user registers his device with an identity in his own hierarchical domain and receives the domain HIBE private key. Each domain root PKG generates the domain HSP and HMS. HSP is made public and is used for generating HIBE public keys (HPK) together with user identities. HMS is kept secret by the domain root PKG and is used for generating HIBE private keys ($HPrK$) for users. Each user has a $HPrK$ that is generated and assigned by his parent PKG. The detailed protocol for domain setup with HIBE private and public key generation and distribution is listed as follows.

DomainSetup(Root PKG $r \in$ Domain D)

r: RootSetup(Domain D) $\rightarrow HSP_D, HMS_D$

u_{t-1}: ExtractHIBEKey($D.HPrK_{u_{t-1}}, D.S_{u_{t-1}}, ID_{u_t}$) $\rightarrow D.HPrK_{u_t}$ for user $u_t \in Level_t$ that is u_{t-1}'s child

$u_{t-1} \rightarrow u_t$: $D.HPrK_{u_t}$

u_1: CreateHIBEPubKey(HSP_D, ID_{u_2}) $\rightarrow D.HPK_{u_2}$

Example: The Michigan State Police Department $PoliceStateMI$ is the root PKG for the police departments in Michigan. $PoliceStateMI$ generates HSP_P and HMS_P for the entire Michigan State Police Department and its subordinate bureaus. The city police departments in Michigan are the level-1 PKGs and receive private keys from $PoliceStateMI$. Alice is a police officer in the Ann Arbor City Policy Department $PoliceCityAA$. Therefore, Alice's ID is $PoliceStateMI/PoliceCityAA/Alice$. $PoliceStateMI$ generates $P.HPrK_{PoliceCityAA}$ for the Ann Arbor City Police Department using its private key $P.HPrK_{PoliceStateMI}$, its secret HMS_P, and the Ann Arbor City Police Department ID $PoliceStateMI/PoliceCityAA$. $PoliceCityAA$ generates $P.HPrK_{Alice}$ for Alice using its private key $P.HPrK_{PoliceCityAA}$, its secret $P.S_{PoliceCityAA}$, and Alice's ID $PoliceStateMI/PoliceCityAA/Alice$. One needs to know HSP_P and Alice's ID in order to generate Alice's public key $P.HPK_{Alice}$ and encrypt a message for Alice.

B. Group Setup

After the offline domain setup phase, users carry their devices with installed domain private keys. In an event that requires user collaboration and dynamic group setup, the users enter the group setup phase. In this phase, a group leader creates a group and generates group parameters and keys.

When a group of users want to form dynamic trust collaboration, a group leader l generates a group G and group keys APK_G and $AMSK_G$ and a set of group attributes S_G. APK_G and S_G are made public in clear text, but l signs the message using his HIBE private key $D.HPrK_l$ for authentication purpose. Group members can use l's HIBE public key $D.HPK_l$ to verify the message. The detailed protocols for group setup is listed as follows.

GroupSetup(Group Leader $l \in$ Domain D)	
l:	CreatGroup(Group G) $\rightarrow APK_G, AMSK_G$, Attributes set S_G
$l \rightarrow u$:	DistributeGroupAPK($D.HPrK_l, APK_G, S_G$) $\rightarrow K = \{APK_G, S_G, sign(APK_G, S_G)_{D.HPrK_l}\}$
u:	CreateHIBEPubKey(HSP_D, ID_l) $\rightarrow D.HPK_l$
u:	RetriveGroupAPK($D.HPK_l, K$) $\rightarrow APK_G, S_G$ if $sign(APK_G, S_G)_{D.HPrK_l}$ is verified by $D.HPK_l$

Example: When police officers are in a rescue mission and need to create a rescue team with other first responders, Alice in the police department becomes the group leader and creates the rescue team A-team. Alice generates $AMSK_{A-team}$, APK_{A-team}, and attributes set $S_{A-team} = \{security\_level, profession\}$ to represent levels of information clearance. And attribute values are $security\_level$ ={top_secret, secret, public} and $profession$ ={general, medical, detective}. Alice signs APK_{A-team} and S_{A-team} with $P.HPrK_{Alice}$ and publishes the A-team public parameters to the A-team.

C. User Enrollment

After the group setup phase, a group is created by the group leader and users enter user enrollment phase to join a group. Upon accepting a user to join the group, the group leader decides what kind of privileges that the user can have, and assigns the user the attributes $attr$ that corresponds to his privileges. As shown below, for each group member u in group G, the group leader l assigns it the corresponding attributes $G.attr_u$. Then l generates the group private key $G.ASK_u$ for u which binds $G.attr_u$ with $AMSK_G$. Then l distributes $G.ASK_u$ to u by encrypting it with u's HIBE public key $E.HPK_u$ generated from u's ID and sending it to u. Only u can decrypt the ciphertext with his HIBE private key $E.HPrK_u$ and receive $G.ASK_u$ sent by l. Note that l and u don't need to belong to the same domain. The detailed protocol for user enrollment is listed as follows.

UserEnrollment(Group Leader $l \in$ Domain D)	
$u \rightarrow l$:	RequestJoinGroup(G) where $u \in$ Domain E
l:	AssignAttrToMember(u) $\rightarrow G.attr_u$ where $G.attr_u \subset S_G$
l:	CreateMemberKey($u, AMSK_G, G.attr_u$) $\rightarrow G.ASK_u$
l:	CreateHIBEPubKey(HSP_E, ID_u) $\rightarrow E.HPK_u$
$l \rightarrow u$:	DistributeMemberKey($E.HPK_u, G.ASK_u$) $\rightarrow K = \{G.ASK_u\}_{E.HPK_u}$
u:	RetrieveMemberKey($E.HPrK_u, K$) $\rightarrow G.ASK_u$

Example: When a fire fighter Bob wants to join the rescue team A-team created by Alice, he sends $RequestJoinGroup(A-team)$ to Alice. Alice then grants Bob membership and decides he has a low-level clearance so she grants him $A-team.attr_{Bob}$ ={public, general} and generates $A-team.ASK_{Bob}$ with Bob's clearance level. Alice distributes $A-team.ASK_{Bob}$ to Bob by encrypting the key with Bob's HIBE public key $F.HPK_{Bob}$, which is generated by Bob's ID and the fire department HSP_F. Bob can retrieve the member key by decrypting the message with his HIBE private key $F.HPrK_{Bob}$.

D. Intra-group Communication

After the user enrollment phase, group members can have secure group communication and collaboration using their ASK and APK. Group members can encrypt data to be shared with flexible and expressive policies defined by access tree structures. Only users with the required attributes/privileges can decrypt the ciphertext and access the shared data.

In the group communication phase, ABE construction ensures that only group members with the corresponding attributes are able to decrypt data. ABE keys guard access to user data and group member u that encrypts message M to ciphertext C controls which attributes can decrypt C. u uses the group APK and an access trees T to encrypt M for members with matching attributes. Only members with ASK that satisfies T can decrypt C and read M. The detailed protocol for intra-group communication is listed as follows.

IntraGroupComm(user $u_1 \in$ Group G, $u_2 \in G$, Message M)	
$u_1 \rightarrow u_2$:	ABEEncrypt(APK_G, M, Access Tree T) $\rightarrow$ Ciphertext C
u_2:	ABEDecrypt($C, G.ASK_{u_2}$) $\rightarrow M$ only if $G.Attr_{u_2}$ satisfies T

Example: When Alice wants to share her location with members of A-team, she encrypts it with APK_{A-team} and the access tree T ={public} since her location is a low-level clearance information. Bob can decrypt Alice's shared message and read her location with his member key $A - team.ASK_{Bob}$ since $A - team.attr_{Bob}$ ={public, general} satisfies T.

E. Inter-group Collaboration

For groups that want to collaborate and share information, secure communication between groups can be done in a similar way as intra-group communication. As shown below, for group collaboration, members in Group B need to know Group A's APK_A and attribute set S_A. Members in B use APK_A and an access tree T created from S_A to encrypt data for A's group members. Members in A are able to decrypt the shared data using their ASKs with matching privileges. The detailed protocol for inter-group collaboration is listed as follows.

InterGroupComm(Group Leader $l_a \in$ Group A and Domain D, User $u_b \in$ Group B)	
$u_b \rightarrow l_a$:	GetGroupAPK(ID_{u_b})
$l_a \rightarrow u_b$:	DistributeGroupAPK($D.HPrK_{l_a}, APK_A, S_A$) $\rightarrow K = \{APK_A, S_A, sign(APK_A, S_A)_{D.HPrK_{l_a}}\}$
u_b:	CreateHIBEPubKey(HSP_D, ID_{l_a}) $\rightarrow D.HPK_{l_a}$
u_b:	RetriveGroupAPK($D.HPK_{l_a}, K$) $\rightarrow APK_A, S_A$ if $sign(APK_A, S_A)_{D.HPrK_{l_a}}$ is verified by $D.HPK_{l_a}$
$u_b \rightarrow u_a$:	ABEEnrypt(APK_A, M, T) $\rightarrow C$ for $u_a \in A$
u_a:	ABEDecrypt($C, A.ASK_{u_a}$) $\rightarrow M$ only if $A.Attr_{u_a}$ satisfies T

Example: When the rescue team $B - team$ members would like to share information with $A - team$ members, they need to receive $A - team$'s APK_{A-team} and S_{A-team} from their group leader Alice. Then the $B - team$ members can encrypt the missing people list with APK_{A-team} and access tree T ={top_secret} and send the message to $A - team$. Bob in $A - team$ can not decrypt the message since $A - team.attr_{Bob}$ ={public, general} does not satisfy T. But members in $A - team$ with high-level clearance can decrypt the message and read the missing people list.

F. Message Authentication

We would like to note that to achieve message authentication and integrity for messages exchanged in intra-group and inter-group communications, we propose to use identity-based signatures (IBS) [8]. IBS lets the users sign the messages with their own identities to achieve message authenticity. IBS combined with HIBE can be viewed as a complete package to provide authenticity, integrity, and confidentiality.

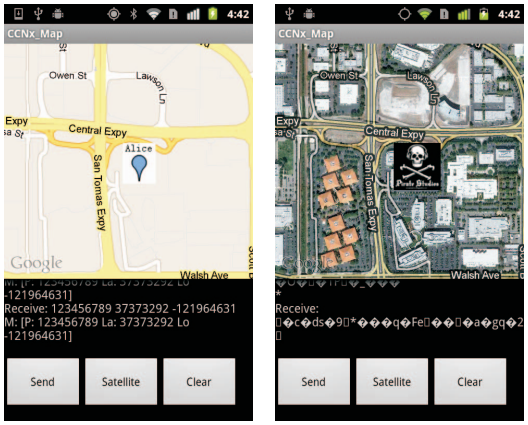

Fig. 2. Snapshots of the location-based application over TGIS on Android.

VI. Implementation and Evaluation

A. Prototype Implementation

We have implemented a prototype of TGIS on Android over a local WiFi access network. Our implementation includes a set of Nexus S devices running Android 2.3. We further run an OpenFire XMPP server [2] in the same network for message broadcast. XMPP provides flexible one-to-one and one-to-many communication and push services between online entities with XML format over HTTP.

We use the open source pairing-based cryptography (PBC) library [3] and implement HIBE [1] and CP-ABE [2] algorithms. All system parameters, master secret keys, and attributes keys are generated and stored as individual files in the mobile device SD card, which can be shared with data sharing applications.

We implemented a data sharing application over TGIS. The application is a Google Latitude-like [4] location-based application on Android to share a user's location data to others. Upon selecting to share his location, a user selects some predefined polices to specify who can access his location data (concatenation of GPS coordinates).The data is then encrypted with the policy and broadcasted to all online group members via the XMPP server. Figure 2 shows the snapshots of the location-based application over TGIS running on Android. The left snapshot shows when the application successfully decrypts a location message, and the right snapshot shows when the application failed to decrypt a location message.

B. Performance

The OpenFire XMPP server acts as both the root PKG and level-1 PKG in our implementation. For the first time the user logins, the client application receives the private key for the user, which is a one-time operation. Similarly, the group creation and group attribute key distribution are also one-time operations for a single group.

[2]http://www.igniterealtime.org/projects/openfire
[3]http://crypto.stanford.edu/pbc
[4]https://www.google.com/latitude

We implemented HIBE with Java Pairing Based Cryptography Library (jPBC) [5], which is a Java porting of the PBC library written in C, and run the evaluation on Nexus S devices running Android 2.3. We measure the processing time for HIBE operations taking on Android. For HIBE encryption, it takes around 1.714 seconds to encrypt a message. This value is the average time to encrypt 30 messages for the size from 50 bytes to 5120 bytes. HIBE decryption averages 0.650 seconds, with IBS signature generation taking an average of 2.034 seconds and IBS signature verification 2.072 seconds. We observed similar performance with CP-ABE. Although the performance seems worse than traditional PKI approach such as RSA encryption and decryption, we believe that mechanisms such as key encapsulation can improve the performance.

The current implementation is developed with the jPBC library, we plan to port the C-based PBC library into Android device with Android Native Development Kit (NDK) [6] for better performance.

VII. Related Work

In this section, we review some related work that provides trust management for bootstrapping security in mobile ad hoc networks (MANETs). We also discuss the related work that exploits identity-based encryption (IBE) and attributed-based encryption (ABE) for access and privacy control.

A. Trust Management

Trust management and security bootstrapping in MANETs is typically difficult to achieve because of the lack of an online trusted entity. There are several papers discussing about trust management and bootstrapping security for MANETs in a distributed manner and without the need of a trusted entity. In [9], the authors introduced two identity-based authentication and key exchange (IDAKE) schemes for MANETS. IDAKE schemes allow two nodes in MANETS to compute a pre-shared secret key for secure communication using their private keys. In [10], the authors also proposed solutions for session key establishment between two nodes exploiting pairing-based cryptography. However, all the schemes proposed are geared more toward one-to-one communication between nodes and TGIS allows nodes in the network to have secure group communication.

B. Access Control

Hengartner and Steenkiste proposed a proof-based access-control architecture that exploits HIBE in pervasive computing [11]. In their scheme, multiple hierarchies are established as policies for access control, and multiple HIBE private keys are used for different policies. Baden et al. proposed Persona, a protocol providing access control for user data over online social networks [12]. Persona uses ABE to allow users to apply access control policies over their data, and let them control who can view their data. ABE are also widely used in cloud computing in providing access control to the data stored in the cloud [13], [14], [15]. In TGIS, we use CP-ABE for information sharing access control in group communication.

[5]http://gas.dia.unisa.it/projects/jpbc/index.html

[6]http://developer.android.com/sdk/ndk/index.html

VIII. Conclusion

To bootstrap trust for dynamic group based information sharing and access control, we propose TGIS for dynamic group collaboration and information sharing with mobile devices. TGIS is a distributed security protocol built upon existing trust infrastructures in individual organizations to enable trust management for group collaborations. Specifically, we have shown how HIBE and CP-ABE can be combined to provide trust management and flexible access control in dynamic group collaborations on mobile devices. We have implemented and evaluated TGIS on Android phones and show it can be used in different applications. The average time for TGIS to perform a HIBE encryption is 1.714 seconds and decryption is 0.650 seconds on an Android phone. The performance is acceptable since the HIBE operations are invoked infrequently during group setup.

Acknowledgement

The work reported in this paper was supported in part by the US AFOSR under Grant No. FA9550-10-1-0393.

References

[1] C. Gentry and A. Silverberg, "Hierarchical id-based cryptography," in *Proceedings of ASIACRYPT*, 2002.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of IEEE Symposium on S&P*, 2007.

[3] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of Advances in Cryptology (CRYPTO'85)*, 1985.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of Advances in Cryptology (CRYPTO'01)*, 2001.

[5] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Proc of IMA Int'l Conf on Cryptography and Coding*, 2001.

[6] D. Boneh, E.-J. Goh, and X. Boyen, "Hierarchical identity based encryption with constant size ciphertext," in *Proc. EUROCRYPT*, 2005.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for ffine-grained access control of encrypted data," in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2006.

[8] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proceedings of ACM SAC*, 2002.

[9] K. Hoeper and G. Gong, "Bootstrapping security in mobile ad hoc networks using identity-based schemes with key revocation," Centre for Applied Cryptographic Research, Tech Report CACR 2006-04, 2006.

[10] W. Shin, C. A. Gunter, S. Kiyomoto, K. Fukushima, and T. Tanaka, "How to bootstrap security for ad-hoc network: Revisited," in *Proc. of IFIP International Information Security Conference (SEC)*, 2009.

[11] U. Hengartner and P. Steenkiste, "Exploiting hierarchical identity-based encryption for access control to pervasive computing information," in *Proceedings of IEEE SecureComm*, 2005.

[12] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," in *Proc. of ACM SIGCOMM Conference on Data Communication*, 2009.

[13] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2010.

[14] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. of the Conference on Information Communications (InfoCom)*, 2010.

[15] Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in *Cryptology ePrint Archive*, 2011.