Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 21 (2013) 242 – 249

Procedia Computer Science

The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013)

MobileCA: Accumulative Secure Group Association with a Certification Path

Oyuntungalag Chagnaadorj[a], Jiro Tanaka[b], b*

[a]*University of Tsukuba, Tennodai 2-1,Tsukuba city, Ibaraki 305-0006, Japan*
[b]*University of Tsukuba, Tennodai 1-1-1, Tsukuba city, Ibaraki 305-8573, Japan*

Abstract

In recent years, there has been growing interest in secure pairing, which refers to the establishment of a secure connection between two mobile devices. Many published studies have described the various types of out-of-band (OOB) channels through which authentication data can be transferred with user control and involvement. However, there has been little discussion of setting up secure connections between groups of mobile devices. Some security protocols have been proposed, but they have tended to focus on a scenario whereby all devices must be located in one place to perform the association.

In this paper, we describe a new group association method, called MobileCA. Our method is designed for a broader range of scenarios and does not require all devices to be in one place at one time. MobileCA extends the OOB channel concept and utilizes digital certification. We have implemented a prototype system using smart phone handsets.

Selection and peer-review under responsibility of Elhadi M. Shakshuki

Keywords: Secure Group Association; Authentication; Out-of-band Channel; Security Protocol

1. Introduction

As personal computers become more ubiquitous, a wider variety of mobile devices has come into our daily lives. Various technologies, including Wi-Fi, Bluetooth, and ZigBee, exist to enable wireless communication between mobile devices. However, compared to their wired counterparts, wireless networks are more vulnerable to security threats, especially to eavesdropping and alteration: so-called

* Corresponding author. Tel.: +81-29-853-5343; fax: +81-29-853-5809.
E-mail address: jiro@cs.tsukuba.ac.jp

1877-0509 © 2013 The Authors. Published by Elsevier B.V.
Selection and peer-review under responsibility of Elhadi M. Shakshuki
doi:10.1016/j.procs.2013.09.032

man-in-the-middle (MitM) attack [16]. It is generally assumed that the major security issues, including MitM attack, can be addressed if the cryptographic public key can be authenticated. In a wired network, authentication is essentially a solved problem, using digital certification and a trusted third party, usually called the certificate authority (CA). However, establishing a trusted third party among mobile devices is not practical, because wireless networks are usually set up on a completely ad-hoc basis, typically involving unfamiliar devices.

User involvement and control in the authentication process bootstraps the problem. In this paradigm, an additional auxiliary channel, termed out-of-band (OOB), exists between two mobile devices, in addition to the ordinary wireless channel. Throughout this paper, 'the requester' refers to the device that is about to be authenticated and 'the verifier' refers to the device that does the authenticating. To authenticate the requester, the verifier receives both the cryptographic public key through a wireless channel and the hash of the same key (authentication data) via the OOB channel. Then the verifier generates another hash of the requester's public key and checks it against the received hash data. If cross-authentication is required, the same action is repeated in the reverse direction.

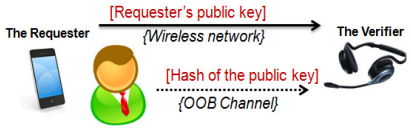

Fig. 1. OOB channel authentication

To date, more than 20 different OOB channels have been proposed [9]. Many low-bandwidth OOB channels have utilized the short-authentication-string (SAS) protocol [17], which reduces the authentication data to 15 bits while providing a reasonable level of security. The SAS protocol can also be used to authenticate the public keys of both devices with a one-directional OOB channel. To elucidate how data are transferred via an OOB channel, we will look at two of them briefly. McCune et al. proposed a barcode-based OOB channel [13]. In this system, the requester encodes the authentication data into a two-dimensional barcode that is shown on the screen of the requester device. The verifier reads the barcode using a camera. Chagnaadorj et al. developed a gesture-based OOB channel [2]. The requester converts authentication data into gestures and displays them to the user. The user performs the gestures one by one with the verifier using a built-in accelerometer.

Although there has been a lot of research in the field of secure device pairing, little attention has been paid to secure association of groups of mobile devices. The term 'secure group association' (SGA) is generally understood to mean that every device authenticates the public keys of other group members. If all necessary authentications are performed successfully, group members can determine a shared secret key for their collaborative interactions. SGAs can be divided into two categories: all-at-once and one-by-one. All previous studies in the field of SGA have focused on all-at-once association. In this paper, we introduce a new group association method, MobileCA, which belongs to the one-by-one category. We assume that all mobile devices are capable of pairing using at least one OOB channel. In practice, it is not difficult for a mobile device to find a suitable channel among existing OOB channels. To join a group, a device must perform OOB channel pairing with one of the group members. On completion of this authentication, the verifier issues a certificate to the requester. Once the devices have exchanged certificates, pairing with other group members becomes automatic owing to the certification path.

The inspiration for the MobileCA association method comes from a common drawback of OOB channels. Even though there are plenty of OOB channels, none of them is generally accepted, and none will be in the near future because they currently cannot be adapted to various mobile devices. Moreover,

recent studies have found that, in real life, people tend to select different pairing methods [4], and furthermore, people are likely to choose different OOB channels depending on the given situation [7].

2. Related Work

To date, there have been relatively few studies on SGA; however, several group security protocols have been proposed [5, 10, 11]. In simple terms, each device sends its public key to every other device via a wireless network, and thus each device generates authentication data independently. Authentication is successful if all of the generated data are the same. However, this approach has well-known limitations. First, all mobile devices must have the proper output for data to be compared. Second, checking the data from all devices is tedious, and becomes even more so as the number of devices increases.

Chen et al. proposed a slightly different group association protocol [3]. Because previous methods were difficult to implement with large groups, they divided groups into subgroups for verification. However, the protocol used a barcode-based OOB channel, which required all devices to have both a camera and display. Moreover, subgrouping increased the burden on users.

Recently, attempts [8, 14] have been made to apply a group protocol [10] to existing OOB channels and to investigate the practical usability of group association methods. However, these studies used three or four OOB channels, which require rich user interfaces, because most OOB channels are not suitable for the protocol. In addition, one of the significant findings from these studies was that many failures were caused by miscommunication between group members, even in a small group (of fewer than six).

Lucero et al. developed an alternative group-formation method [12]. To form a group, each device must touch the next device to the right of it. This method is easy, fast, and also can prevent an adversary from joining. However, the issue of how to protect communication once a group has been formed was not addressed.

3. Group Association Types

In general, SGA can be classified into all-at-once and one-by-one association. Table 1 lists some of the characteristics of the two categories. The characteristics of SGA types are expanded on as follows:

Table 1. Characteristics of all-at once and one-by-one associations

Characteristics	All-at-once	One-by-one
Previous Work	Few	None
Proximity	Required	Not required
Synchronicity	Required	Not required
Device Diversity	Limited	Partly limited
Scalability	Weak	Strong
Group Size	Limited	Unlimited
User Involvement	Full	Semi-automatic
Group Formation	Dependent	Partially dependent

- *Previous Work:* There has been little discussion of SGA, and all published studies have described all-at-once group association [3, 5, 10, 11]. The comparison in Table 1 is between these association protocols and our MobileCA method.
- *Proximity and Synchronicity:* All group members have to be physically located in the same place and must participate actively and simultaneously in all-at-once group association. In contrast, as its name

implies, one-by-one association occurs in an incremental manner, allowing each device to join the group independently of the other group members and their associations.

- *Device Diversity:* In the existing protocols, all group members are expected to possess the same physical method for verifying authentication data. For example, screens may be required [5, 11], or both a camera and a display may be needed [3]. In [10], even though a specific output was not mentioned, the protocol needed all devices to be equipped with similar interfaces so that users could compare authentication data. However, the verification phase of each device in MobileCA involves only two devices: the new device and an existing group member. Therefore, only one device from the group is required to be compatible with the new device.
- *Scalability:* Scalability is a major disadvantage of all-at-one association. To add a new device to the group, all group members that are already associated must be assembled and perform the association together. Group association should allow for the addition of a new device without all of the existing devices being physically present, and MobileCA addresses this problem.
- *Group Size:* As a group grows in number, all-at-once association becomes difficult to carry out and user effort increases considerably. Recent usability studies have shown that many failures can be caused by miscommunication of group members, even in small groups of fewer than six users [8, 14]. In contrast, the size of a group in MobileCA can grow without most of the group members even being aware of the growth.
- *User Involvement:* Even though it is impossible to eliminate user involvement completely, group association should be as automatic as possible. In existing protocols, users must actively participate in the entire association process, whereas MobileCA requires user involvement in only n–1 pairings; the remaining association is automatic. Moreover, all-at-once group association is inappropriate when one user manages multiple personal mobile devices, because the user has to compare authentication data with every device. In contrast, MobileCA can be applied to both single and multiple users' cases.
- *Group Formation:* In general, the process of establishing a secure group consists of two parts: group association (authenticating the public keys of each member) and group formation (readying collaborative interactions, for example, by deciding on a shared key). These two actions should occur as independently as possible. Once group association has been successfully completed, all or some of the associated members can safely form the group at any time. However, a problem arises if unassociated devices are included in the group. This situation tends to happen frequently due to the ad-hoc nature of mobile networks, and so the most efficient handling is required. All-at-once association must be performed again, even if all devices except one are previously associated. However, in MobileCA, only the new device pairs with one existing group member, and the remaining process is automatic.

4. MobileCA

In this section, we describe the MobileCA SGA method in detail. We assume that all mobile devices are equipped with a wireless network protocol, the installation of at least one OOB channel, and computational hardware that is sufficient for the basic cryptographic operations.

4.1. Design

Assume that Bob has several mobile devices. To protect wireless communication between his smart phone and laptop, he uses a barcode-based OOB channel. In addition, he establishes a secure connection between his smart phone and his wireless headset using a gesture-based OOB channel. He now wants to secure the communication channel between the laptop and the wireless headset. The following question

arises: must he perform OOB channel pairing, which consumes a considerable amount of time and effort [9] again? We believe that our method is the most appropriate to facilitate this kind of scenario.

MobileCA has two principle aspects, which are described below:

- *Certification:* Once the public key is verified using the selected OOB channel, the verifier issues a certificate to the requester. If the OOB channel uses the SAS protocol, both devices certify each other. The certificate is saved in a protected repository, called the KeyStore, on both devices. KeyStore holds two types of certificates: certificate entries and key entries. Certificate entries are certificates that the device itself has issued to other devices, whereas key entries are used to ensure the public key of the device and are signed by other devices.
- *Certification Path Pairing (CPP):* Saved certificates are used for subsequent pairing, as shown in Figure 2. For instance, suppose Bob's smart phone has already received a certificate from the laptop and that the headset has also received a certificate from the phone. This means that there is a certification path between the laptop and the headset, i.e., Cert (Laptop -> Phone) + Cert (Phone -> Headset). Given this, the laptop can have confidence in the public key of the headset, and so issues a certificate to it without OOB channel-based pairing. If cross certification is required, CPP is repeated in the reverse direction.

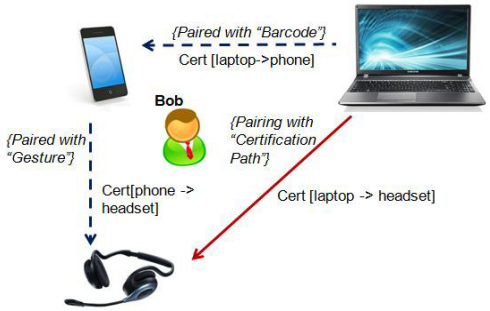

Fig. 2. The base concept of MobileCA

Compared to a system in which every device in the group is compared with every other device, the effort required is reduced from O(N**2) to O(N). Saving the user time and effort is not the only merit of CPP, however, as it also enables the pairing of two mobile devices without a common OOB channel. For example, a wireless headset cannot be paired using the barcode-based OOB channel, which requires a camera and a display. Assume that the laptop also cannot communicate using the gesture-based OOB channel. Consequently, Bob can use the smart phone, which can pair using both barcode- and gesture-based OOB channels, as a bridge between the laptop and the headset, as shown in Figure 2.

Although CPP allows pairing of two mobile devices, this result shows the essential features of SGA. Using MobileCA, more than two mobile devices can possess the authenticated public keys of other devices. In this situation, they can easily build a secure group at any time.

4.2. Centralized MobileCA Model

MobileCA assumes that every mobile device is able to pair using at least one OOB channel. However, even if this requirement is satisfied, establishing a fully formed secure group may be impossible in some cases. For example, assume Bob has seven devices, as shown in Figure 3, and he wants to associate all of them. All possible OOB channel pairings are performed and the certificates are issued. Consider the following cases:

- *No matching OOB channel:* For example, Bob's camera can pair with other devices only using an LED-based OOB channel [15]; however, no other devices can communicate using this channel.
- *No path found:* Even if there is a certification path, for example, from the iPod to the headset, CPP may not occur between them. This problem can be solved if CPP is carried out first either between the headset and the laptop or between the iPod and the phone.
- *No path at all:* CPP may never be accomplished between some pairs of devices, for example, the printer and the laptop.

Fig. 3. Limitations of MobileCA

Fig. 4. Centralized MobileCA model

The centralized MobileCA model can address all of these limitations. To build the centralized model, Bob must designate one of his mobile devices as a hubCA. This should be the device with the greatest computing power and the most user interfaces. All OOB channels that are to be used must be installed in the hubCA, whereas only one OOB channel is sufficient for the other mobile devices.

Secure device association with centralized MobileCA will follow two general steps. First, each device is paired with the hubCA using the desired OOB channel, and certificates are issued. Following this, every pair of devices can automatically be paired using CPP, as shown in Figure 4. A device can join the group at any time by following these steps.

Mobile devices may join many different groups, thus they could have more than one hubCA certificate in their KeyStore. It does not matter which hubCA certificate is used as long as there are certification paths between the two devices. Figure 5 illustrates the sequence of the CPP protocol. Two rounds are performed between two mobile devices: reqCont and reqStore represent objects on the requester device, and verCont and verStore are on the verifier side.

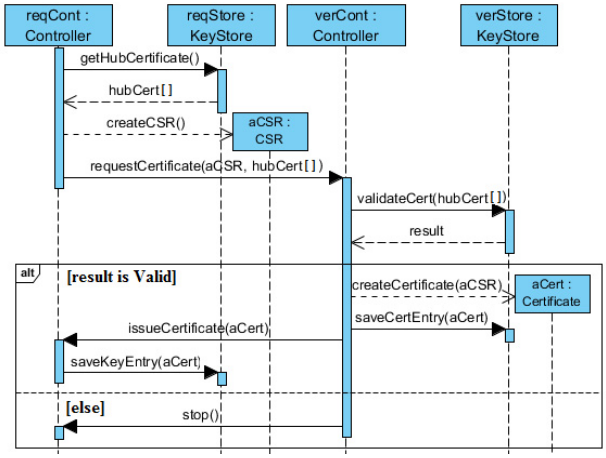

Fig. 5. The sequence diagram of centralized CPP

Round 1. The requester simultaneously sends a certificate sender request (aCSR) and the certificates that were issued by the hubCAs (hubCert []).

Round 2. The verifier validates the received certificates (validateCert) using its own certificates to the hubCAs. If all received certificates are invalid or the verifier did not certify any hubCAs, the verifier stops CPP (stop). Otherwise, the verifier issues a certificate (aCert) to the requester, and then saves it in its KeyStore (saveCertEntry ()) as a certificate entry. The requester also saves the received certificate in its own KeyStore as a key entry (saveKeyEntry ()).

4.3. Implementation and Microbenchmark

We implemented MobileCA on smart phones running the Android operating system. The prototype system was implemented in Java and cryptography was implemented using Bouncy Castle [1], which contains a lightweight cryptography API suitable for memory-constrained devices. The communication between devices used WLAN.

Table 2 shows the computational overhead of the two smart phones: a Samsung Galaxy 2S (dual-core 1.5-GHz Qualcomm Scorpion CPU with 1 GB of RAM) and a Pantech Mirach (dual-core 1-GHz Qualcomm Snapdragon CPU with 512 MB of RAM) for several cryptographic operations used in MobileCA. RSA operations were performed using 1024-bit keys. As can be seen from the table, the key generation consumed the most time; however, the key pair is only created when MobileCA runs for the first time.

Table 2. Computational cost of the cryptographic operations

Operations	Galaxy (sec)	Mirach (sec)
RSA Key Generation	1.38	1.8
CSR Generation	0.03	0.044
Certificate Verification	0.019	0.023
Certificate Generation	0.068	0.13

5. Discussion

Here we focus on several minor yet important concerns that have not yet been covered in this paper. First, MobileCA can certainly fit into an all-at-once group association. If a group of people gather and decide to associate their mobile devices for some collaborative interactions, they should designate one of their devices as the hubCA, and then complete the required steps of MobileCA.

Second, MobileCA is the association part of our project of establishing a secure group. In the group formation part, we propose another protocol for a set of mobile devices to determine a shared key. Briefly, one device is selected as a moderator, and the other devices inform the moderator of their participation; then the moderator sends a generated shared key to all members safely using certificates.

Finally, the concept that every node on a network issues certificates to other nodes without a trusted third party is not a new idea. It is termed a user-centric trust model, and some wired network systems that also have an ad-hoc nature, such as peer-to-peer or multi-agent systems, use this model. Therefore, it can easily be adapted to wireless networks, and the authentication essentially relies on the trust calculation. However, quantifying authentication is a controversial topic in itself [6]. MobileCA exploits the user's involvement and control in the authentication process; thus, a validated certificate provides better assurance that the public key is authentic.

6. Conclusion

We have described an accumulative method for associating a secure group of mobile devices, called MobileCA. It utilizes OOB channels and digital certificates to achieve the goal. Although MobileCA is straightforward and uses established techniques, it has two major advantages over existing association methods. First, a device can join the group at anytime, and not all group members are required to be physically located in the same place. Second, user involvement can be minimal since the group association is simplified.

In future work, we will test MobileCA using practical group interactions such as file sharing. In addition, we plan to develop the method further to study alternative wireless technologies such as Bluetooth and WiFi Direct.

Acknowledgements

We would like to thank the members of the Interactive Programming laboratory of the University of Tsukuba for their many valuable comments and feedback.

References

[1] The Legion of the Bouncy Castle. http://www.bouncycastle.org/
[2] Chagnaadorj O, Tanaka J. MimicGesture: Secure Device Pairing with Accelerometer-based Gesture Input. In *Proc. CUTE'12*. Springer Press (2012).
[3] Chen CO, Chen C, Kuo C, Lai Y, McCune JM, Studer A, Perrig A, Yang B, Wu T. GAnGS: Gather, Authenticate 'n Group Securely. In *Proc. MobiCom'08*. ACM Press (2008).
[4] Chong MK, Gellersen H. How Users Associate Wireless Devices. In *Proc. CHI'11*. ACM Press (2011).
[5] Creese S, Goldsmith M, Roscoe B. Bootstrapping Multi-Party Ad-Hoc Security. In *Proc. SAC'06*. ACM Press (2006).
[6] Epp CE. Relationship Management: Secure Collaboration in a Ubiquitous Environment. *PERVASIVE computing, IEEE*. 2, 2 (Apr 2003) , 62-71
[7] Ion I, Langheinrich M, Kumaraguru P, Capkun S. Influence of User Perception, Security Needs, and Social Factors on Device Pairing Method Choices. In *Proc. SOUPS'10*. ACM Press (2010).
[8] Kainda R, Flechais I, Roscoe AW. Two Heads are Better Than One: Security and Usability of Device Associations in Group Scenarios. In *Proc. SOUPS'10*. ACM Press (2010)
[9] Kumar A, Saxena N, Tsudic G, Uzun E. A comparative study of secure device pairing methods. In *Proc. PerCom'09*. IEEE Press (2009)
[10] Laur S, Pasini S. SAS-based Group Authentication and Key Agreement Protocols. In *Proc. PKC'08*, ACM Press (2008).
[11] Lin Y, Studer A, Hsiao H, McCune JM, Wang K, Krohn M, Lin P, Perrig A, Sun H, Yang B. SPATE: Small-group PKI-less Authenticated Trust Establishment. In *Proc. MobiSys'09*. ACM Press (2009).
[12] Lucero A, Jokela T, Palin A, Aaltonen V, Nikar J. EasyGroups: Binding Mobile Devices for Collaborative Interactions. In *Proc. CHI'12*. ACM Press (2012).
[13] McCune JM, Perrig A. Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication. *International Journal of Security and Networks*. 4, 1/2 (Feb. 2009), 43-55.
[14] Nithyanand R, Saxena N, Tsudik G, Uzun E. Groupthink: Usability of Secure Group Association for Wireless Devices. In *Proc. UbiComp'10*. ACM Press (2010).
[15] Saxena N, Uddin MB. Automated Device Pairing for Asymmetric Pairing Scenarios. In *Proc. ICICS'08*. Springer Press (2008).
[16] Stajano F, Anderson R. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Proc. SPW'99*. Springer Press (1999).
[17] Vaudenay S. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In *Proc. Crypto'05,* Springer Press (2005).