Electronic Preprint for Journal of Information Processing Vol.23 No.3

Regular Paper

# Accumulative Secure Group Association in an Ad-hoc and Ubiquitous Network (Version 3.0)

Oyuntungalag Chagnaadorj[1,a) Jiro Tanaka[1]

Received: May 14, 2014, Accepted: January 7, 2015

***Abstract:*** The increased popularity of mobile devices, such as laptops, mobile phones, tablets, accessory devices, and more, brings new challenges in the area of network security in the ubiquitous environment, where ad-hoc networks may be formed using mobile devices. There has been little discussion of setting up secure connections among a group of devices. A number of group security protocols has been proposed; however, these have tended to focus on a scenario whereby all devices must be located in one place at one time. In this paper, we describe a new secure group association method that associates mobile devices in an accumulative manner, so it does not require all group members to be physically present. It utilizes digital certification and extends the concept of out-of-band channels, through which the authentication data can be transferred using human user involvement. We have implemented a prototype system and conducted a comparative user experiment to demonstrate viability of the accumulative association.

***Keywords:*** secure mobile group, secure group association, out-of-band channel, certification

## 1. Introduction

Over the last few decades, there has been tremendous growth in the area of ubiquitous computing. Numerous portable devices that perform many complex operations and can be used in a wide variety of applications have been developed. Following this mobile computing trend, more and more collaborative tasks are shifting into mobile devices. Various technologies, including Wi-Fi and Bluetooth, exist to enable wireless communication between them. However, compared with their wired counterparts, wireless networks are more vulnerable to security threats, particularly to eavesdropping and alteration, also termed as man-in-the-middle (MitM) attack [1]. It is generally assumed that major security issues, including MitM attack, can be addressed if one's cryptography public key can be authenticated. Authentication methods of wired networks, such as using a pre-installed key or trusted third-party authentication, cannot be adopted, because wireless networks are usually set up on an *ad-hoc* basis, typically involving unfamiliar devices. Therefore, a new authentication mechanism is needed for ad-hoc and ubiquitous networks. Large numbers of researches have been focusing on the user-aided authentication to bootstraps the problem.

Throughout this paper, the term secure group association (SGA) is used to mean that establishment of secure connections among a group of mobile devices. Similarly, the term secure device pairing (SDP) is used to refer to establishment of a secure connection between two mobile devices. Both SDP and SGA have been employing the user-aided authentication to enable security.

The main principle of the user-aided authentication is to use human user's involvement in the authentication process. In this paradigm, an additional auxiliary channel that is perceivable by the user, called the out-of-band (OOB) channel, exists between two mobile devices as well as the ordinary wireless channel. The simplest user-aided authentication process is straightforward as shown in **Fig. 1**: the verifier (the device that does the authenticating) receives both the cryptographic public key through a wireless channel and a hash of the same key (the authentication data) via the OOB channel from the requester (the device that is to be authenticated). The verifier then generates another hash of the requester's public key and checks it against the received hash data.

As it can be seen from Fig. 1, the user-aided authentication consists of two independent yet inseparable parts: an authentication protocol and an OOB channel. Various SDP protocols have been proposed so far [2], [3], [4]. Vaudenay et al. introduced a SAS (Short Authenticated String) protocol [5], which is well suited for many low bandwidth OOB channels as it reduces the authentication data to 15 bits while providing a reasonable level of security. To date, a significant number of OOB channels [6], [7], [8], [9], [10], [11], [12] have been proposed as well. To elucidate how the authentication data is transferred via an OOB channel, we will look at two of them briefly. McCune et al. proposed a barcode-based OOB channel [6]. In this system, the requester encodes the authentication data into a two-dimensional barcode and shows it on its screen. The verifier reads the barcode

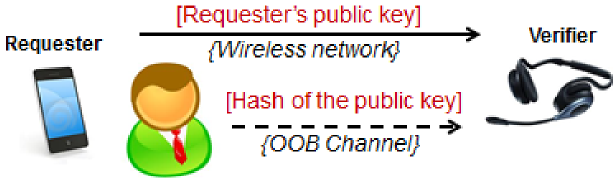

**Fig. 1** Simple user-aided authentication.

[1] University of Tsukuba, Tsukuba, Ibaraki 305–8572, Japan
a) chtungalag@gmail.com

using a camera. Chagnaadorj et al. developed a gesture-based OOB channel [12]. The requester converts the authentication data into a sequence of gestures and displays them. The user performs the gestures one by one holding the verifier device that has a built-in accelerometer.

There is considerable body of literature in the field of SDP; however, little attention has been paid to SGA. Few group security protocols have been proposed, but they have tended to focus on a scenario whereby all group members must be physically located in the same place to perform the association. We have designed a new SGA method that does not require all devices to be in one place at one time. It occurs in an accumulative manner, allowing each device to join the group independently from the other group members and their associations.

Our accumulative SGA method assumes that mobile devices are capable of pair using one or more OOB channels. In practice, it is not difficult for a mobile device to find a suitable channel among the existing OOB channels. To join a group, a device pairs with one of the existing group members using the user-aided authentication. On completion of successful authentication, the verifier issues a digital certificate to the requester. Once the devices have exchanged certificates, pairing with other group members becomes automatic owing to a certification path.

We have reported comparison on the accumulative SGA method and the existing protocols, and it clearly demonstrates the proposed method has a number of attractive features, such as stronger scalability, greater flexibility, unlimited group size, better device diversity and etc. Also, we have implemented a prototype system and conducted a comparative user experiment, which proved that the accumulative group association is fast enough.

The inspiration for the proposed method comes from a common drawback of OOB channels. Even though there are a large number of OOB channels available, none are generally accepted as a standard. Furthermore, a single OOB channel is unlikely to become an accepted standard in the near future because not all mobile devices are capable of using a given OOB channel. Moreover, recent studies have found that, in real life, people tend to select different pairing methods [13], and in addition, people are likely to choose different OOB channels depending on the given situation [14]. Our accumulative SGA method allows mobile devices to use their desired OOB channels whereas the existing SGA protocols require all group members to possess the same or similar OOB channels.

The remainder of this paper is organized as follows: Section 2 briefly describes related work; Section 3 compares two types of associations; Section 4 details the main elements of our proposed method, Section 5 reports a comparative study, and Section 6 concludes the paper.

## 2. Related Work

There have been relatively few studies on secure mobile group; however, a number of group security protocols including Multi-Party [15], SAS-GMA [16], and SPATE [17] have been proposed. 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 some significant 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. introduced a slightly different group association protocol, called GAnGS [18]. 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 requires all devices to have both a camera and a display. Moreover, subgrouping increases the burden on users.

When talking about the user-aided authentication, two important aspects must be considered: what kind of OOB channel the method utilizes and how much the user involvement the protocol requires. These two aspects of the existing SGA protocols are presented in **Table 1**. If a single member broadcasts its authentication data and the remaining members compare it with their own, the number of OOB channel transfers can be (n-1), where n is the size of the group, for SAS-GMA and Multi-Party. Furthermore, in order to detect intruders, group security protocols basically require the users to verify the size of the group by counting the potential group members, and then either by inputting it to the system before the association or by comparing it with the number of group members after the association. Moreover, GAnGS uses a barcode-based OOB channel to collect group members' information in one place.

Recently, few attempts [19], [20] were made to apply SAS-GMA protocol [16] to the existing OOB channels and to investigate the practical usability of secure group association. However, these studies used three and four OOB channels respectively, which require rich user interfaces, as most OOB channels are not suitable for the protocol. In addition, one of the significant findings from these studies is that many failures are caused by miscommunication between users, even in a small group (of fewer than six).

A few group security protocols for body area network (BAN) in where a group consists of many low-capability sensor devices and one controller device that has richer user interfaces and higher computational power, and work together toward monitoring patient's health in healthcare systems, has been proposed [21], [22]. BAN may be considered as a specific type of mobile network that needs high level of privacy and security in its interactions. However, it lacks ad-hoc nature of mobile groups in general, and so it was reflected in designed methods. Keoh at al. [22] proposed as system, for example, that assumes every device and employees are certified by the hospital before deployment.

**Table 1** Summary of existing group association methods.

| SGA methods | OOB Channel Type | User Involvement | |
|---|---|---|---|
| | | OOB Channel Transfer | Other |
| Multi-Party | Screen | ⩾ n-1 | Size verification |
| SAS-GMA | Does not specify | ⩾ n-1 | - |
| SPATE | Screen | n-1 | Size verification |
| GAnGS | Screen + Camera | n-1 | Size verification, data collection |

## 3. Secure Group Association Types

In general, SGA can be classified into all-at-once and one-by-one association. **Table 2** compares some important characteristics of the two categories and each characteristic is expanded on as follows:

*Literature reports:* There has been little discussion of SGA, and all published studies have described all-at-once group association [15], [16], [17], [18]. The comparison in Table 2 is between these association protocols and our proposed 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 accumulative 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 as shown in Table 1. Even though a specific output was not mentioned in SAS-GMA protocol [16], it needs all devices to be equipped with similar interfaces so that users could compare authentication data. However, the authentication of each device in our accumulative method involves 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-once 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 our method addresses this problem. In the proposed method, only the new device pairs with one existing group member, and the remaining process is automatic.

*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 [19], [20]. In contrast, the size of a group in one-by-one association can grow as big as possible 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 addition to transferring the authentication data through OOB channel, all-at-once group association also involves the users to verify the size of the group (Table 1). Moreover, there will be noticeable difference in terms of user's effort over a long period of time. The total effort in all-at-once associations is increased if group reforms many times while our association remains constant.

**Table 2** Characteristics of all-at-once and one-by-one associations.

| Characteristics | All-at-once | One-by-one |
|---|---|---|
| Literature reports | A 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 |
| Robustness | Weak | Strong |
| Group Key Establishment | Some | No |

*Robustness*: OOB channel authentication has a relatively high possibility of encountering errors due to user involvement [23], [24]. If an abnormal occurs during an all-at-once association, entire process aborts even it is relevant to one member. In other hand, authentication in the proposed method involves two devices only and its failure does not affect other group members and their associations.

*Group Key Establishment*: In general, the process of establishing a secure group consists of two stages: secure group association and group key establishment. Some of the existing protocols solved both stages [15], [16]. However, our proposed method considers SGA only, and thus, to interact securely within the group, determining a group key is necessary. Once the group members are authenticated successfully in our accumulative SGA method, some or all members can use their authenticated public keys to establish the group key using one of the existing group key establishment protocols, such as Ref. [25].

## 4. Accumulative Secure Group Association

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 Base Concept

Suppose that Bob has several mobile devices as shown in **Fig. 2**. He securely paired his smart phone and laptop as well as his smart phone and wireless headset using user-aided authentication. He now wants secure connection between the laptop and the wireless headset, but he does not like using user-aided authentication that consumes considerable amount of time and effort [23], [24] for already authenticated devices. We believe that our method is the most appropriate to facilitate this kind of scenario.

Our accumulative SGA method has two principle aspects, which are described below:

- *Certification:* Two devices are paired using user-aided authentication. Once the public key is verified successfully, the verifier issues a digital certificate to the requester, as depicted in Fig. 2. The certificate is saved in a protected repos-

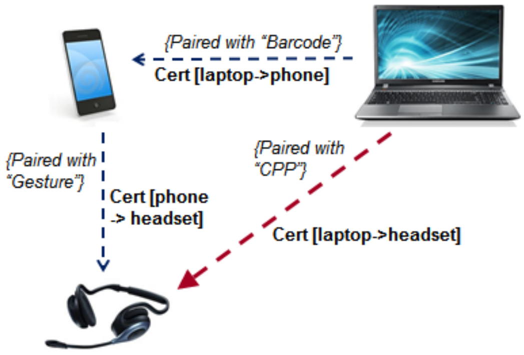

**Fig. 2** The base concept of the proposed method.

itory, 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. For instance, suppose Bob's smart phone has already received a certificate from the laptop and 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*), as shown in Fig. 2. Given this, the laptop can have confidence in the public key of the headset, and so issues a certificate without the user-aided authentication.

Although CPP allows pairing of two mobile devices, its result shows the essential features of SGA. When CPP is completed successfully, more than two mobile devices possess the authenticated public keys of other devices. In this situation, they can easily build secure connections at any time.

## 4.2 Centralized Model

The proposed accumulative SGA method assumes that every mobile device possesses at least one OOB channel. However, even if this requirement is satisfied, establishing a fully authenticated group may be impossible in some cases. For example, suppose Bob has seven devices and he wants to associate all of them. He performed all possible secure device pairings as shown in **Fig. 3**, but association cannot be accomplished due to the following cases:

- *No matching OOB channel:* For example, Bob's camera can pair with other devices only using an LED-based OOB channel [7]; however, no other devices can communicate using this channel.
- *No direct path*: Even if there is a certification path, for example, from the iPod to the headset, CPP may not occur between them directly because the certification path is supposed to include two certificates only: one is the requester received from the middle device and another is the verifier issued to the middle device in the proposed method. Therefore, in order to execute CPP between iPod and the headset that has the certification path of three certificates, 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 executed between some pairs of devices due to the lack of the certification path between them. If two mobile devices share the same OOB channel, they are able to be paired using the user-aided authentication. In Bob's case, the printer is paired with the iPad only and the laptop is paired with the phone and iPod. However, there is no certification path found between the printer and the laptop.

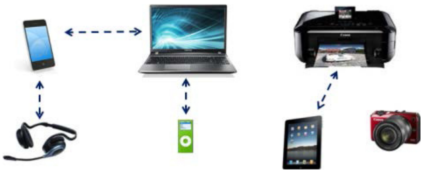

**Fig. 3** Limitations of the accumulative SGA method.

The centralized model of the method can address these limitations. To build the centralized model, Bob must designate one of his mobile devices as the groupHub. 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 groupHub, whereas only one OOB channel is required for the other mobile devices. In centralized model, the accumulative SGA proceeds as follows. First, each device is paired with the groupHub using their desired OOB channel, and certificates are issued. Following this, every pair of devices can automatically be paired using CPP, as shown in **Fig. 4**. A device can join the group at any time by following these steps.

However, relying on one mobile device, the groupHub, in ubiquitous environment is not practical and association should occur at any cases. Therefore, if there is no sign of the groupHub when adding a new device, a different device in the group can be designated as the groupHub. In that case, the new device is able to join the group as following the same steps as usual because the new groupHub has already exchanged certificates with other group members. That means that the group can have more than one groupHub.

## 4.3 Certification Path Pairing (CPP) Protocol

After a new member pairs with the groupHub successfully, it is associated with other group members using the certification path. Therefore, we proposed Certification Path Pairing (CPP) protocol. CPP protocol is designed in such a way that the new member can mutually associate one or more existing group members simultaneously, as depicted in **Fig. 5**.

In CPP protocol, three rounds are performed between the new device and every group member to certify mutually. The new device initiates the protocol by broadcasting a request (Request_ND). When the group member receives the request, it establishes a regular unicast connection with the new device, and

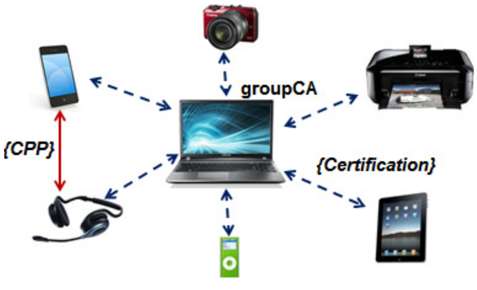

**Fig. 4** Centralized model.

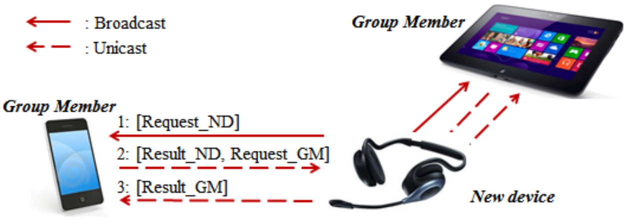

**Fig. 5** CPP protocol in general.

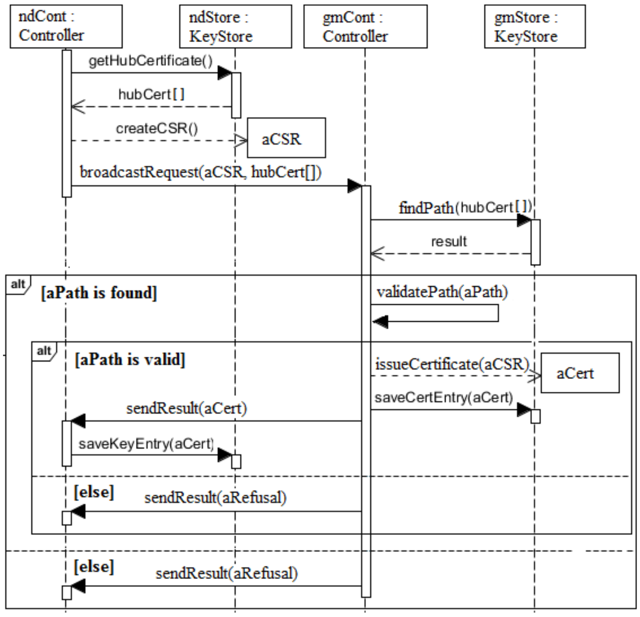

**Fig. 6** CPP protocol sequence.

sends the result of the request (Result_ND) along with its own request (Request_GM). In the end, the new device sends the result of each received request (Result_GM).

**Figure 6** depicts the sequence of the first half (Request_ND and Result_GM) of the CPP protocol as another half (Request_GM and Result_GM) is the exactly same. ndCont and ndStore represent objects on the new device, and gmCont and gmStore are on the group member side. The rounds of the protocol are expanded on as follows.

Round 1: The new group member broadcasts a request, which includes the certificate sender request (aCSR) and the certificates that were issued by the groupHubs (hubCert[]). Mobile devices may join many different groups, thus there may be more than one groupHub certificates in their KeyStore. It does not matter which groupHub certificate is used as long as there is certification path between two devices.

Round 2: After receiving the request, group member searches a certification path to the new device (findPath). The certification path is supposed to consist of two certificates: one is groupHub's issued certificate to the new device and another is the group member's issued certificate to the groupHub. The new device sends groupHub's issued certificate, and therefore, to find the path, the group member tries to find the certificate, of which the subject is the equal to the issuer of the received certificate, from its KeyStore. If the path exists, it needs to be validated (validatePath). To do that, expiration dates of both certificates are checked first. Then, the digital signature of the received groupHub certificate is verified by using its public key on certificate of the group member. If the path is valid, group member issues a certificate (aCert) to the new device and saves it in its KeyStore as a certificate entry (saveCertEntry). Otherwise, it refuses to certify. Therefore, the result includes either refusal or newly issued certificate. The new device also saves the received certificate in its own KeyStore as a key entry (saveKeyEntry).

### 4.4 Role of the GroupHub

The groupHub is not a trusted third party and does not take on the same level of responsibility of a certificate authority. Instead, it serves only as a bridge between other devices to enable authentication. Moreover, relying on one device completely in ubiquitous environment should be avoided. In other words, group association should be as flexible as possible and have the ability to move on in any cases. In our method, if a new member joins the group and there is no sign of the groupHub, different device in the group can be chosen as the groupHub.

A mobile device can join more than one group, and remain there for a long period of time, due to the ad-hoc nature of mobile networks. For example, Bob can create a secure group for his personal devices at home. Furthermore, his laptop may also be a member of another group in his office. This means the device can possess several groupHub certificates. Selecting a specific group is not considered in CPP protocol. However, the proximity of mobile devices may assist to distinguish the groups. For example, when Bob starts CPP protocol with his laptop in the office, Alice's device is most likely to find a certification path with the groupHub of the office.

### 4.5 Group Management

To add a new member is easy in the proposed accumulative SGA method. To do that, the new member pairs with groupHub first, and then it associates with other group members using CPP protocol. However, how to remove the member from the group is not a simple task and must be considered carefully. A successful execution of the proposed method exchanges digital certificates among group members; therefore the system needs to revoke certificates to remove the undesired member. In an ad-hoc and ubiquitous network, distributing a certificate revocation list is not practical. The simplest solution to this problem is to adjust certificate lifetimes. In addition, the group can exclude the undesired member from the group key establishment protocol.

## 5. Comparative Study

In this section, we conducted a comparative study of the one-by-one and all-at-once group association methods in terms of the completion time. One-by-one association has a number of advantages over all-at-once association, as reported in Section 3. However, one-by-one association is intuitively considered that it needs much more time to complete. Therefore, the goal of this experiment is to clarify this particular assumption. The hypothesis of the study is that one-by-one association is not slower than all-at-once association.

### 5.1 Experiment Setup

For a comparative experiment, we implemented two prototype systems: our proposed method and an all-at-once association. We chose SAS-GMA [17] protocol as a representative of the all-at-once association to compare with our accumulative SGA method for a couple of reasons. First, as presented in Table 1, it requires the minimum user involvement. Without verifying the size of the group, all-at-once association cannot prevent intruders from joining. Thus, it is a serious drawback of SAS_GMA protocol. If

**Table 3** Technical specification of handsets and their computational cost of the cryptographic operations.

| | Samsung Galaxy Note | | Samsung Nexus S | | Samsung Galaxy 2S | | Pantech Mirach | | Sharp IS01 | |
|---|---|---|---|---|---|---|---|---|---|---|
| CPU | Cortex-A9 dual-core 1.4 GHz | | Cortex-A8 dual-core 1.5GHz | | Qualcomm Scorpion dual-core 1.5GHz | | Qualcomm Snapdragon dual-core 1GHz | | Qualcomm QSD8650 1GHz | |
| RAM | 1GB | | 1 GB | | 1GB | | 512 MB | | 256 MB | |
| Android Version | 4.0 Ice Cream Sandwich | | 4.0 Ice Cream Sandwich | | 2.3 Gingerbread | | 2.3 Gingerbread | | 1.6 Donut | |
| Operation (msec) | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| RSA Key Generation | 620.2 | 285.18 | 799.3 | 472.83 | 625.78 | 328.03 | 910.2 | 614.82 | 804.4 | 410.24 |
| CSR Generation | 57.92 | 18.06 | 35.4 | 13.96 | 30.16 | 36.86 | 36.84 | 6.63 | 48.3 | 35.75 |
| Certificate Verification | 69.18 | 15.1 | 34.28 | 5.67 | 25.5 | 1.31 | 30.84 | 1.26 | 50.66 | 34.91 |
| Certificate Generation | 37.56 | 6.82 | 94.4 | 36.49 | 93 | 43.19 | 104.5 | 35.89 | 297.28 | 209.21 |

size verification is added to the protocol, its completion time will notably increase.

Second, unlike other SGA protocols, SAS-GMA does not specify a type of OOB channel that it needs. However, it has to be the same or similar for all group members, so that SAS authentication data can be verified simultaneously. OOB channels may be divided into three main categories based on the user involvement; device-to-device (d2d), such as barcode-based [7], two device-to-human (2 x d2h), such as LED light-based [7], and device-to-human + human-to-device (d2h+h2d), such as gesture-based [12]. In d2d type of OOB channel, the users have relatively little influence on the success of the transmission. Those channels transfer the authentication data directly from the requester to the verifier and the users only assist in making the transmission possible. Therefore, we selected the barcode-based OOB channel for SAS-GMA implementation to avoid the human related errors because the completion time of the successful association is needed for our experiment. Furthermore, smart phones that possess both a camera and a display are used in our experiment, and therefore, the barcode-based transfer was the handiest in our situation. To equalize with SAS-GMA, the accumulative SGA method also utilizes the barcode-based OOB channel for its pairings.

### 5.2 Implementation

Both prototype systems were developed on smart phones running the Android operating system. The implementation used Java language and cryptography part was written using Bouncy Castle library [26], which contains a lightweight cryptography API suitable for memory-constrained mobile devices. The barcode-based OOB channel was implemented using ZXing open-source library [27]. The communication between devices used WLAN.

**Table 3** presents the technical specification of the smart phones handsets that used in our experiments as well as the computational overhead of the handsets for several cryptographic operations. RSA operations were performed using 1,024-bit keys taking into account low-power mobile devices. As can be seen from the table, the key generation consumed much more time; however, the key pair is only created when the system runs for the first time.

Our proposed accumulative SGA method consists of two parts. To join a group, a new device pairs with groupHub first, and then it associates other group members using CPP protocol, explained in Section 4.3. For the first part, we adopted modified SAS protocol [9] as it can authenticate the public keys of both devices with one OOB channel transfer. As a result, the number of barcode-based transfer becomes n-1, where n is the size of the group. The sequence of the modified SAS protocol in our implementation is depicted in **Fig. 7**. It consists of six rounds, including the OOB channel transfer.

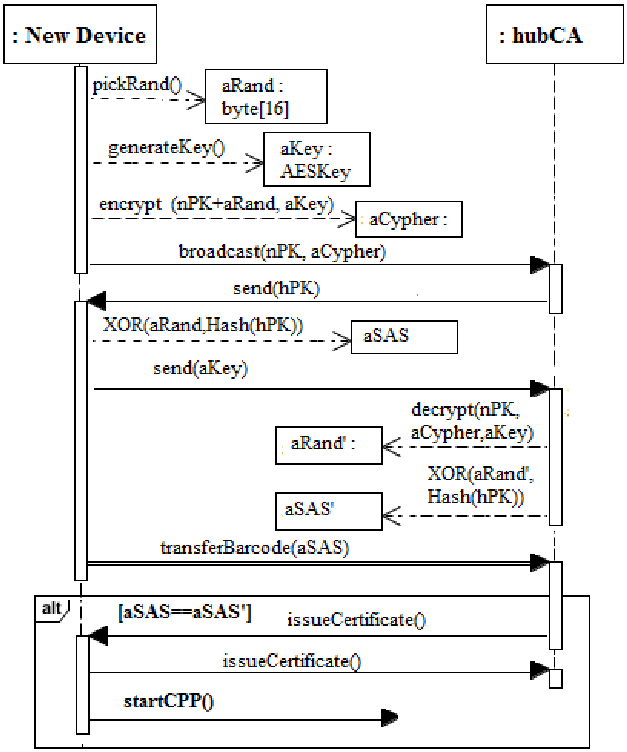

**Fig. 7** Sequence of SAS protocol implementation.

Round 1: Primary purpose of the SAS protocol [13] is to reduce the length of the authentication data, so that the low bandwidth OOB channels can be used. However, our selected barcode-base OOB channel is capable of transferring data that is bigger than the size of the public key. Thus, the prototype uses 16 byte SAS data instead of 16 bits (pickRand). Moreover, a symmetric cryptography function (generateKey) is used for the implementation of the commitment scheme. The encrypt() function replaces the COMMIT phase and the decrypt() function replaces the OPEN phase of the commitment scheme. Thus, the new device encrypts its public key (nPK) as well as the random (aRand) using the generated key (aKey), and then it broadcasts its public key (nPK) along with the encrypted data (aCypher).

Round 2: When the groupHub receives the request, it establishes a regular unicast connection with the new device, and sends its public key (hPK). Figure shows the public keys in round 1 and round 2, for the sake of the simplicity. However, in actual prototype, a certificate sender request (CSR), which includes not only the public key but also all necessary information about the device for the certification, is sent.

Round 3: Once the new device receives the public key of the groupHub, it sends back the symmetric key (aKey). The groupHub then decrypts the received data in round 1 using the key to get the committed value of the new device. Finally, both devices compute the SAS authentication data (aSAS) independently by performing exclusive OR operation between the committed value (aRand) and hash of the public key of the groupHub.

Round 4: BARCODE transfer takes place with the assistance of the user. The new device converts own SAS data into a QR barcode and displays it. The groupHub scans the code using its camera.

Round 5: The groupHub converts the scanned QR code into the SAS data. If the received SAS data through the barcode transfer is the exactly same as its own computed SAS data, the groupHub issues a certificate to the new device. Otherwise, the groupHub sends the refusal.

Round 6: If the new device receives the certificate from the groupHub, it also issues a certificate and sends it to the groupHub.

On completion of the successful pairing with the groupHub, the new device launches the CPP protocol by broadcasting a request in order to associate other group members.

SAS-GMA protocol is very similar to the SAS pairing protocol. Main difference is that there is no unicast connection between devices. Instead, all group members simultaneously communicate through broadcasting, as shown in **Fig. 8**. It has five rounds including the barcode-based OOB channel transfers.

Round 1: Every device broadcasts a request that includes their public key and encrypted data simultaneously. The procedure of committing the random value is the same as the one, described in round 1 of the accumulative SGA method implementation.

Round 2: Every device receives the request of other group members. Round 1 and round 2 occurs synchronously. SAS-GMA protocol is designed for devices to wait for a while before moving to the next round. Our initial version of the prototype was implemented as it is. However, the main goal of the experiment is to compare the mean completion time of two types of associations. If one system stops for seconds during the association, another system will get benefit from it. Therefore we changed the protocol slightly and in our final version of the prototype, the participants input the expected group size in the system. As a result, instead of waiting, the system proceeds to the next round when it receives the same number of the request as the group size.

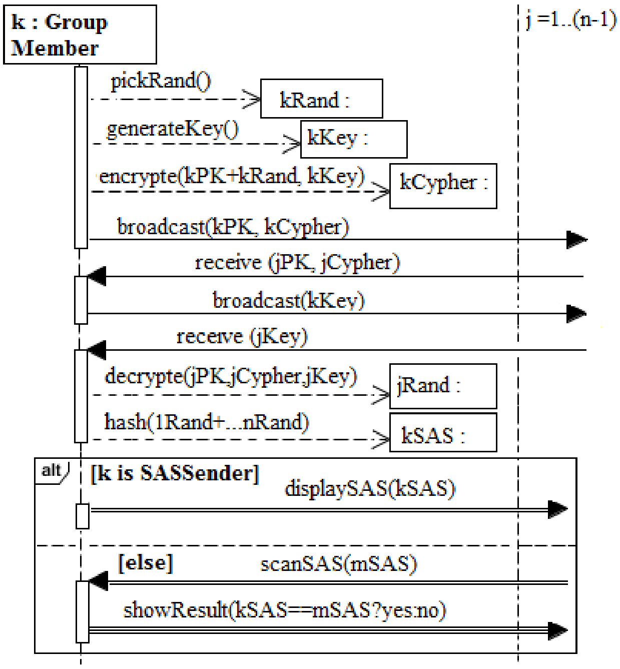

**Fig. 8** Sequence of SAS-GMA protocol implementation.

Round 3: Every device broadcast their encryption key to open their commitments.

Round 4: Every device receives the encryption keys of the other group members. Once the device receives the keys from all devices that have sent the request, it decrypts all received data in round 1. Every device then independently computes the SAS authentication data. To do that, all committed values must be put into an order by their sender's identities. Our prototype sorts them based on IP address of the device. Then, we concatenate the sorted values because the barcode-based OOB channel is not low bandwidth, and so the length of the SAS data is not a problem in the prototype. Finally, SAS is ready by hashing the concatenated value (hash (1Rand + $\cdots$ + nRand)).

Round 5: BARCODE transfer takes place with the assistance of the users. Instead of transferring SAS data between every pair of devices, the prototype chooses one device (SASSender) to display the SAS data the form of QR code (displaySAS) and other members scan it by their camera (scanSAS). It reduces the number of OOB channel transfers from n*(n-1)/2 to n-1, where n is the size of the group. After scanning the code, the device checks the received SAS data against its own created one, and then informs the result (showResult). The association accomplishes, if all devices verify the SAS data.

### 5.3 Participants and Tasks

Fifteen volunteers (eight females and seven males) participated in our experiment; all of them were either undergraduate or graduate students. Their mean age was 29.7 years (SD = 5.79 years, range 20–38). Before beginning the experiment, a brief introduction was given on how to carry out both associations.

In line with other SGA studies [17], [19], [20], we considered a common setting with small groups (sizes 4–6). Various com-

**Table 4** The completion time of associations - multiple users (sec).

| Size | Proposed method | | | SAS-GMA | |
|---|---|---|---|---|---|
| | Mean | SD | Paired tTest (df=9) | Mean | SD |
| 4 | 27.05 | 3.01 | t=0.92, $p$=0.383 | 25.29 | 4.55 |
| 5 | 34.32 | 2.68 | t=1.38, $p$=0.200 | 32.18 | 3.35 |
| 6 | 42.11 | 3.22 | t=-0.57, $p$=0.581 | 43.94 | 8.08 |

**Table 5** The completion time of associations - single user (sec).

| Size | Proposed method | | | SAS-GMA | |
|---|---|---|---|---|---|
| | Mean | SD | Paired tTest (df=14) | Mean | SD |
| 4 | 26.70 | 4.72 | t=0.95, $p$=0.359 | 28.90 | 8.27 |
| 5 | 33.60 | 5.31 | t=-1.09, $p$=0.295 | 35.83 | 6.01 |
| 6 | 42.87 | 8.27 | t=0.16, $p$=0.872 | 42.39 | 5.11 |

binations of different handsets, of which technical specifications are presented in Table 3, were tested. Participants were randomly selected into 4, 5, and 6 persons groups. Each group carried out both one-by-one and all-at-once associations. We have conducted total of ten associations for each method in all three group sizes. Therefore, a single participant has involved in more than one association, but no two groups were the exactly same.

In the proposed association, each participant holds the groupHub in turns to pair it with their respective devices using SAS protocol with the barcode-based OOB channel. As sson as the pairing is completed, the newly joined device starts pairing again with other group members simultaneously using CPP protocol. In SAS-GMA protocol, all participants start the association with their respective devices at the same time. On completion of the association, each participant holds the SASSender in turn to verify the computed SAS data of their devices using the barcode-based transfer.

The completion time of the proposed method is measured from starting the first member pairing with the groupHub to completing CPP protocol of the last member. In SAS-GMA protocol, the completion time is measured from launching association on all devices to ending the last device displaying the result of the verification. Before each association, one device is designated as the groupHub in our method and the SASSender in SAS-GMA protocol. In addition, participants input into their respective devices how many devices would be associated by selecting the group size on the settings in SAS-GMA association. However, time for these actions is not counted in the total association time.

### 5.4 Result

**Table 4** summarizes the completion time of two methods in three group sizes. The collected data was analyzed using paired tTest between two methods in each group size to determine if there are statistically significant differences.

The results are presented on the middle column of the table, and they clearly demonstrate that no significant differences between the completion times of one-by-one and all-at-once associations ($p > 0.05$) were found for group size of 4, 5, and 6 devices.

In addition to the multiple user experiment, we have conducted the same experiment in a single user scenario. Thus, each of fifteen participants was asked to carry out both group associations in three different group sizes alone. However, the results were very similar to those of the multiple user cases, as shown in **Table 5**.

In the end, we analyzed the same data again to see whether there are statistically significant difference between the completion time of single user and multiple user cases. Paired tTest was also used for each association method in all three group sizes. The result shows that no significant differences ($p > 0.05$) in all six cases were found.

As a result, it can be concluded that both one-by-one and all-at-once group associations spend the similar amount of time in the small group of less than 6 devices for both single and multiple user cases. Moreover, the number of users does not affect the completion time of the group associations.

### 5.5 Discussion

In our user experiment, we asked the participants to associate all devices immediately because the purpose of the study was to measure the association time. However, in general, the proposed accumulative SGA method is supposed to associate devices without any limitation on time and place. For instance, assume that 5 employers associated their personal devices during a meeting using our method. On the next scheduled meeting, an employer could not participate while two new comers joined. New employers could easily add their devices to the group even though one member was missing. The device of employer who was absent is able to associate with new group members in any place at any time using CPP protocol that is automatic.

The completion time of the user-aided authentication relies profoundly on the user involvement in the association process because the speed of OOB channel transfer is much slower than wireless communications no matter how slow the network is and how many rounds the protocol performs. Moreover, it varies depending on OOB channel type. We used a barcode-based OOB channel in our experiment for both of all-at-once and one-by-one associations. All-at-once association basically requires the same OOB channels for all members. In contrast, our method is supposed to associate devices using their desired OOB channels. Therefore, the experiment we have conducted is not realistic to some extent. However, it has proved that a tendency to believe that the one-by-one association spends more time is wrong.

As can be seen from Table 3, operation time increases noticeably as computational power of mobile devices becomes lower. Therefore, the proposed method need to adopt other security technologies, such as TinyECC [28], which are more suitable for the resource constraint mobile devices, and also can substitute expensive public key cryptography operations while offering the reasonable degree of security and functionality.

## 6. Conclusion

We designed a new SGA method that associates mobile devices in an accumulative manner. It utilizes digital certification as well as the user-aided authentication to achieve the goal. Although the proposed method uses established techniques, it has a number of important advantages, such as stronger scalability, greater flexibility, unlimited group size, better device diversity and etc. over the existing SGA protocols.

We implemented a prototype system and conducted a comparative user experiment in order to ensure viability of the proposed method. The result proved that a tendency to believe that the accumulative association spends more time is wrong.

In future work, we would like to implement a complete library that is based on our accumulative SGA method. The library include all kinds of secure device pairing protocols as well as all types of OOB channels, so that any mobile device is able to pair or associate in the ubiquitous environment regardless of their physical capabilities.

**Acknowledgments** 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] Stajano, F. and Anderson, R.: The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks, *Proc. SPW '99*, Springer Press (1999).

[2] Balfanc, D., Smetters, D.K., Stewart, P. and Wong, W.C.: Talking to Strangers: Authentication in Ad-hoc Wireless Networks, *Proc. NDSS '02*, Internet Society Press (2002).

[3] Hoepman, J.H.: Ephemeral Pairing on Anonymous Networks, *Proc. SPC '05*, Springer Press (2004).

[4] Laur, S. and Nyberg, K.: Efficient Mutual Data Authentication Using Mutually Authenticated Strings, *Proc. CANS '06*, Springer Press (2006).

[5] Vaudenay, S.: Secure Communications over Insecure Channels Based on Short Authenticated Strings, *Proc. Crypto '05*, Springer Press (2005).

[6] McCune, J.M. and Perrig, A.: Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication, *International Journal of Security and Networks*, Vol.4, No.1/2, pp.43–55 (2009).

[7] Saxena, N. and Uddin, M.B.: Automated Device Pairing for Asymmetric Pairing Scenarios, *Proc. ICICS '08*, Springer Press (2008).

[8] Goodrich, M.T., Sirivianos, M., Solis, J., Tsudik, G. and Uzun, E.: Loud and Clear: Human-Verifiable Authentication Based on Audio, *Proc. ICDCS '06*, IEEE Press (2006).

[9] Saxena, N., Ekberg, J., Kostainen, K. and Asokan, N.: Secure Device Pairing Based on a Visual Channel, *Proc. Security and Privacy Symposium*, IEEE Press (2006).

[10] Soriente, C., Tsudik, G. and Uzun, E.: BEDA: Button-Enabled Device Association, *Proc. IWSSI '07*, ACM Press (2007).

[11] Soriente, C., Tsudik, G. and Uzun, E.: HAPADEP: Human-Assisted Pure Audio Device Pairing, *Proc. ISC '08*, Springer Press (2008).

[12] Chagnaadorj, O. and Tanaka, J.: Gesture Input as an Out-of-band Channel, *Journal of Information Processing Systems*, Vol.10, No.1, pp.92–102 (2014).

[13] Ion, I., Langheinrich, M., Kumaraguru, P. and Capkun, S.: Influence of User Perception, Security Needs, and Social Factors on Device Pairing Method Choices, *Proc. SOUPS '10*, ACM Press (2010).

[14] Chong, M.K. and Gellersen, H.: How Users Associate Wireless Devices, *Proc. CHI '11*, ACM Press (2011).

[15] Creese, S., Goldsmith, M. and Roscoe, B.: Bootstrapping Multi-Party Ad-Hoc Security, *Proc. SAC '06*, ACM Press (2006).

[16] Laur, S. and Pasini, S.: SAS-based Group Authentication and Key Agreement Protocols, *Proc. PKC '08*, ACM Press (2008).

[17] Lin, Y., Studer, A., Hsiao, H., McCune, J.M., Wang, K., Krohn, M., Lin, P., Perrig, A., Sun, H. and Yang, B.: SPATE: Small-group PKI-less Authenticated Trust Establishment, *Proc. MobiSys '09*, ACM Press (2009).

[18] Chen, C.O., Chen, C., Kuo, C., Lai, Y., McCune, J.M., Studer, A., Perrig, A., Yang, B. and Wu, T.: GAnGS: Gather, Authenticate 'n Group Securely, *Proc. MobiCom '08*, ACM Press (2008).

[19] Kainda, R., Flechais, I. and Roscoe, A.W.: Two Heads Are Better Than One: Security and Usability of Device Associations in Group Snearious, *Proc. SOUPS '10*, ACM Press (2010).

[20] Nithyanand, R., Saxena, N., Tsudik, G. and Uzun, E.: Groupthink: Usability of Secure Group Association for Wireless Devices, *UbiComp '10*, ACM Press (2010).

[21] Ming, L., Shucheng, Y., Wenjing, L. and Kui, R.: Group Device Pairing based Secure Sensor Association and Key Management for Body Area Network, *Proc. IEEE INFOCOM '10* (2010).

[22] Keoh, S.L., Lupu, E. and Sloman, M.: *Securing Body Sensor Networks: Sensor Association and Key Management*, IEEE Press (2009).

[23] Kumar, A., Saxena, N., Tsudik, G. and Uzun, E.: A comparative study of secure device pairing methods, *Pervasive and Mobile Computing*, Vol.5, No.6, pp.734–749 (2009)

[24] Kainda, R., Flechais, I. and Roscoe, A.W.: Usability and Security of Out-of-band Channels in Secure Device Pairing Protocols, *Proc. SOUPS '09*, ACM Press (2009).

[25] Steiner, M., Tsudik, G. and Waidner, M.: Key Agreement in Dynamic Peer Groups, *IEEE Trans. Parallel and Distributed Systems*, Vol.11, No.8, pp.769–780 (2000)

[26] The Legion of the Bouncy Castle,
available from ⟨http://www.bouncycastle.org/⟩.

[27] ZXing Library, available from ⟨https://github.com/zxing/zxing/⟩.

[28] Liu, A. and Ning, P.: TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks, *Proc. ISPN '08*, IEEE Press (2008).

**Oyuntungalag Chagnaadorj** is a Ph.D. candidate in computer science at University of Tsukuba, Japan. Her research interests include secure pairing of mobile devices and ubiquitous computing. She received a B.S. at National University of Mongolia, Mongolia and a M.S. at Ritsumeikan University, Japan in 2004.

**Jiro Tanaka** is a Professor of Department of Computer Science, University of Tsukuba. His research interests include ubiquitous computing, interactive programming, and computer-human interaction. He received a B.Sc. and a M.Sc. from the University of Tokyo in 1975 and 1977. He received a Ph.D. in computer science from University of Utah in 1984. He is a member of ACM, IEEE and IPSJ.