Groupthink: Usability of Secure Group Association for Wireless Devices

Rishab Nithyanand
University of California, Irvine
rishabn@ics.uci.edu

Nitesh Saxena
Polytechnic Institute of NYU
nsaxena@poly.edu

Gene Tsudik
University of California, Irvine
gts@ics.uci.edu

Ersin Uzun
University of California, Irvine
euzun@ics.uci.edu

ABSTRACT

A fairly common modern setting entails users, each in possession of a personal wireless device, wanting to communicate securely, via their devices. If these users (and their devices) have no prior association, a new security context must be established. In order to prevent potential attacks, the initial context (association) establishment process must involve only the intended devices and their users.

A number of methods for initial secure association of *two* devices have been proposed; their usability factors have been explored and compared extensively. However, a more challenging problem of initial secure association of a *group* of devices (and users) has not received much attention. Although a few secure group association methods have been proposed, their usability aspects have not been studied, especially, in a comparative manner. This paper discusses desirable features and evaluation criteria for secure group association, identifies suitable methods and presents a comparative usability study. Results show that some simple methods (e.g., peer- or leader-based number comparisons) are quite attractive for small groups, being fast, reasonably secure and well-received by users.

Author Keywords

Device Pairing, Group Association, Usability

ACM Classification Keywords

D.4.6 Security and Protection: Authentication; C.2.0 Computer-Communication Networks: General; H5.m Information interfaces and presentation (e.g. HCI): Miscellaneous

General Terms

Human Factors, Security.

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.

INTRODUCTION

Short-range wireless communication (based on technologies such as Bluetooth, WiFi, Zigbee and WUSB) has become very common in many spheres of everyday life. Increasing proliferation of personal wireless gadgets (PDAs, cellphones, headsets, cameras and media players) continuously offers new services and possibilities for ordinary users. There are many use cases where two wireless devices need to "collaborate", e.g., a Bluetooth headset and a cellphone, a wireless access point and a laptop, or two Bluetooth smartphones. Scenarios involving more than two devices and users are also emerging. Consider, for example, an impromptu meeting involving a small group of people with no association history. They need to establish a group communication channel in order to exchange messages or documents.

The surge in popularity of wireless devices brings about certain security risks. The wireless channel is easy to eavesdrop upon or simply jam. It is equally easy to inject traffic. These inherent vulnerabilities of the wireless channel prompt a number of realistic threats, such as *Man-in-the-Middle* (MitM) or *Evil Twin* attacks. In order to secure the wireless channel, secure communication must be bootstrapped, i.e., a set of wireless devices must be securely associated.

One of the main challenges in secure device association is that, due to the wide variety of devices and lack of standards, no global security infrastructure exists today and none is likely to materialize in the near future. To address this problem, one fruitful research direction has been the use of auxiliary – often called "out-of-band" (OOB) – channels, that are both perceivable and manageable by the human users who own and operate the devices. An OOB channel takes advantage of human sensory capabilities to authenticate human-imperceptible (and hence subject to MitM attacks) information exchanged over the wireless channel. OOB channels can be realized using sensory input, such as audio, visual and tactile. Unlike the main (usually wireless) channel, the adversary can not remain undetected if it actively interferes with the OOB channel; although, the adversary can still eavesdrop.[1]

[1]It is important to note that this approach only requires the OOB channel to be authenticated, but not secret, in contrast to traditional Bluetooth pairing or key exchange protocols based on "user-selected" secret PINs.

Since some degree of human involvement is unavoidable, usability of device pairing methods is a crucial issue. Moreover, since a typical OOB channel is relatively slow (low-bandwidth), there is an incentive to minimize the amount of transmitted information, for the sake of both usability and efficiency. Recently proposed pairing methods (overviewed in the background section) typically require transmitting few bits over an OOB channel to achieve reasonable security. Most are based on comparing OOB strings: matching strings on both devices imply a successful pairing, and distinct strings imply an attack.

Secure Group Association

There has been a considerable amount of work on the usability of two-device pairing [18, 16, 14, 19]. However, the application domain for secure pairing methods is not limited to two devices or even two users. A group of users might want to establish a common channel for exchanging or sharing information, such as email, instant messages, documents and multi-media content. The main advantage of using a wireless channel in this scenario is that no infrastructure is needed. Thus, *ad hoc* communication can take place with no extra cost to the users. *Group association* of users' devices is necessary to secure the common channel, i.e., to prevent eavesdropping on, modification of, and injection of, information.

Generally, we view group association as having several flavors:

1. One user (owner) bootstrapping secure group communication context for a set of his/her own wireless devices. For example, a user has a few wireless devices and needs to set up (from scratch) a new secure group communication context among them.
2. One user incrementally introducing one device (at a time) to an existing secure group communication context. For example, a user – who already has several wireless devices in a home network – buys a new gadget and wants enroll it into the group.
3. Multiple (> 2) users, each with a personal wireless device, want to bootstrap a new (perhaps only short-term) secure group communication context.

Of course, other variants are possible, i.e., the above list is not exhaustive. However, in this paper, we concentrate on the last category — a group of users, each with a personal device. Hereafter, we use the term "group device association" to refer to that category only. More specifically, this paper focuses on the *usability of group device association*. This topic has not received much attention thus far. Although a few group association protocol and methods [39, 23, 7, 25] (for category 3 above) have been proposed, their usability aspects have not been studied, especially, in a comparative fashion.

Usability Challenges

One of the main challenges in developing two-device pairing methods is the need to accommodate a wide-variety of devices. Some devices have rich user interfaces (e.g., PDAs and cellphones), while others have very limited features (e.g., wireless headsets, WiFi access points and sensors). In contrast, the group setting considered in this paper involves high-end wireless devices with adequate means of user-perceivable input and output. This is because devices with constrained interfaces are not appealing for establishing a multi-user communication context. A typical group of devices might include different models of laptops, PDAs and smartphones, produced by various manufacturers. This "feature" of group association simplifies the problem to some extent. However, other issues complicate matters and present new challenges that influence usability of group association:

- **SCALABILITY**: In two-device pairing, scalability is not an issue. In other words, one or two users can be asked to perform any reasonable (not too burdensome) task. However, since group pairing requires participation of multiple users and devices, certain tasks become unsustainable. For instance, a method requiring two devices to be shaken simultaneously [27] is clearly unsuitable for group association. Also, methods that dictate very close proximity between devices – such as camera-based [29, 30] or image comparison-based [32] methods – are not very amenable to group association. This prompts us to ask: *what existing methods, or their variants, can scale up to support groups?*
- **ROBUSTNESS and SPEED:** As in the two-device case, security and usability of group association depends on robustness and complexity of the underlying method. Participation of multiple users/devices makes verification tasks (such as string or number comparisons) more complicated, slower and potentially more error-prone. For example, a group association method based on number comparison [37] needs multiple interactions among group members to match 5-8 digit numbers displayed on their respective devices (see *Figure 1*). Naturally, each user-involved interaction increases the probability of errors and the consequent need to re-run the entire protocol.

  Even if the number of users/devices is reasonably small (e.g., 4-8), the sheer duration of the group association process is also likely to influence the usability of each method. A natural question, therefore, is: *can verification tasks be performed robustly and efficiently?*
- **GROUP SIZE:** Unlike the two-device case, group association requires participants to perform an additional task of correctly determining group size (participant count). In case of a counting error (particularly if someone overcounts) an adversary might insert malicious nodes into the group. Therefore, another important issue is: *can the distributed counting task be performed robustly for groups of reasonable sizes?*

Intended Contributions

Questions posed above form the primary motivation for the study presented in this paper. More broadly, our work seeks to answer the question: *what pairing methods can be deployed in real-world secure group association scenarios?*

In line with recent work on group association [25], we consider a common setting with small groups (sizes 4-6). We

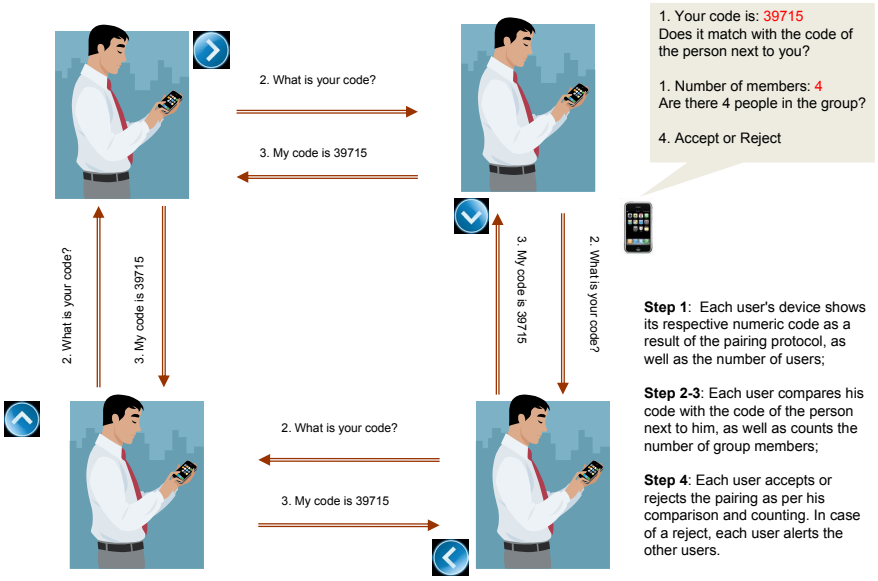

Figure 1. Group Pairing: peer-based numeric comparison with circular topology

first overview prominent two-device pairing methods, and identify those (or variants thereof) amenable for the group setting. We implemented five selected methods atop a common software platform and conducted a comprehensive and comparative study, focusing on both usability and security. Clearly, without a thorough study, it is hard to answer the above question, based only on intuition or prior test results for two-device pairing scenarios [18, 16, 14, 19].

Results of our study show that some simple methods (e.g., peer- or leader-based number comparisons) are quite attractive for small groups, being fast, reasonably secure and highly rated by users. We believe that this is an important and timely first step in exploring real-world usability and security of pairing methods for emerging group scenarios.

BACKGROUND

We now describe notable cryptographic protocols and pairing methods and then identify methods suitable for group association. The term *cryptographic protocol* denotes the entire interaction involved, and information exchanged, in the course of device pairing. The term *pairing method* refers to the pairing process as viewed by the users, i.e., user interactions. As discussed later, a single cryptographic protocol can be coupled with many pairing methods.

Cryptographic Protocols

One simple protocol was suggested in [2]: devices A and B exchange their respective public keys pk_A, pk_B over the insecure channel and the corresponding hashes $H(pk_A)$ and $H(pk_B)$ – over the OOB channel. Although non-interactive, the protocol requires $H()$ to be a collision-resistant hash function and thus needs at least 80 bits of OOB data in each direction. MANA protocols [11] reduce the size of OOB messages to k bits while limiting adversary's success probability to 2^{-k}. However, these protocols require a stronger assumption on the OOB channel: the adversary is assumed to be incapable of delaying or replaying any OOB messages.

In [6][40], the authors presented the protocol based on Short Authenticated Strings (SAS), which limits attack probability to 2^{-k} for k-bit OOB channels, even when the adversary can delay/replay OOB messages. This protocol utilizes commitment schemes (which can be based upon hash functions such as SHA-1, MD5) and requires 4-round of communication over the wireless channel. Subsequent work ([22] and [31]) developed 3-round SAS protocols.

SAS protocols have been extended to the group setting [39, 23]. They follow the same security model as the two-party protocols, but the communication among group members takes place over a broadcast wireless communication channel. The success/failure of group pairing is also based on whether or not the SAS strings computed by all devices match. The group association methods that we evaluate in this paper are all based on these SAS protocols. Recall that, in addition to comparing SAS values, group association also involves the user(s) to correctly count the number of members taking part in the pairing protocol.

Similar to two-party password-based authenticated key exchange protocols (PAKA) (e.g., [4]), group association can be based on a shared secret password, as shown in [1]. However, security of this protocol relies on password secrecy, which might be hard to guarantee in practice. In this paper, we only consider group association methods based on OOB channels that do not require secrecy.

Pairing Methods and Their Applicability to Group Setting

Based on cryptographic protocols described above, a number of pairing methods have been proposed. They operate over different OOB channels and offer varying degrees of usability.

"Resurrecting Duckling" [36] is the initial attempt to address the device pairing problem in the presence of MiTM attacks. It involves standardized physical interfaces and cables. Unfortunately, requiring physical equipment (i.e., a cable) defeats the purpose and convenience of using wireless connections and this method clearly does not scale beyond two users.

Another early method is "Talking to Strangers" [2], that uses infrared (IR) communication as the OOB channel and requires almost no user involvement, except for initial setup. However, this method is deceptively simple since IR demands line-of-sight alignment and its set-up requires the user to find IR ports on both devices – not a trivial task for many users – and align them. Also, IR is not completely immune to MiTM attacks and it has been gradually displaced by other wireless technologies (e.g., Bluetooth and WiFi).

An alternative approach involves image comparison: OOB data is encoded as images and the user is asked to compare two images displayed on two devices. Prominent examples include "Snowflake" [12], "Random Arts Visual Hash" [32] and "Colorful Flag" [9]. The group association method proposed in [25] is based on a variant of the colorful flag representation. All these methods require either close proximity among devices and their users, or necessitate exchange of devices[2]. Such "features" are not suitable for group association.

Another research direction yielded the "Seeing-is-Believing" (SiB) pairing method [29]. In its simplest instantiation, SiB requires a unidirectional visual OOB channels: one device encodes and displays OOB data as a two-dimensional barcode. The other device then "reads it" using a photo camera, operated by the user. A related approach, called "Blinking Lights" was investigated in [30], where OOB data is encoded as a blinking sequence of an LED and a video camera on the other device records it. However, without additional centralized equipment (such as a projector, as in [7]), both of these methods require close enough proximity among devices. Moreover, they are only applicable to devices with cameras, which would exclude (at least some) laptops expected to be among devices in a group association setting. Also, laptop cameras are meant to be used as webcams used primarily for capturing images and/or video of the laptop user. It might be cumbersome to capture other objects and devices belonging to other users using a lid-mounted laptop camera.

Another recent method is called "Loud-and-Clear" (L&C) [26]. It uses MadLib sentences that represent the digest of information exchanged over the main wireless channel. There are two L&C variants: "Display-Speaker" and "Speaker-Speaker". In the latter, the user compares two vocalized sentences and in the former – displayed sentence with its vocalized counterpart. Minimal device requirements include a speaker (or audio-out port) on one device and a speaker or a display on the other. The user needs to compare two respective (vocalized and/or displayed) MadLib sentences and either accept or abort the pairing based on the outcome of the comparison. Device-vocalized sentences are hard to use in a group association. However, a variant involving only displayed sentences could be used. However, this would represent overkill, since numeric comparison (as discussed below) is a better and simpler choice.

An experimental investigation [37] reported on a comparative usability study of simple pairing methods for devices with displays capable of showing a few (4-8) decimal digits. In the "Compare-and-Confirm" approach, the user simply compares two 4-, 6- or 8-digit numbers displayed by devices. In the "Select-and-Confirm" approach, one device displays to the user a set of numbers, the user selects the one that matches the number displayed by the other device. In the "Copy-and-Confirm" approach, the user copies a number from one device to the other. The last variant is "Choose-and-Enter" which asks the user to pick a "random" 4-to-8-digit number and enter it into both devices. Compare-and-Conform and Copy-and-Conform methods can be extended to groups of reasonable sizes (as we will discuss in the next section). Select-and-Confirm, on the other hand, was not found to be usable from prior tests [37]. As shown in [1], Choose-and-Enter can be extended to groups. However, its security crucially relies upon the secrecy of the PIN, and securely distributing a shared secret PIN in a group setting might be difficult.

Some follow-on work (HAPADEP [35, 13]) considered pairing devices that – at least at pairing time – have no common wireless channel. HAPADEP uses pure audio to transmit cryptographic protocol messages and requires the user to merely monitor device interaction for any extraneous interference. It requires both devices to have speakers and microphones, interfaces that are relatively common on modern devices. To appeal to more basic settings, a variant of HAPADEP, that uses the wireless channel for cryptographic protocol messages and the audio as the OOB channel, can be employed. In this variant, only one device needs a speaker and the other – a microphone. Also, the user is asked to perform any comparisons. As discussed in the following section, HAPADEP can be extended to groups of reasonable sizes.

There are also pairing methods geared for devices with severely limited user interfaces (such as access points or headsets). For example, [33] and [38] proposed a pairing method based on comparison of audio-visual patterns, such as blinking or beeping. They only require devices to have 1-2 LEDs and a beeper. BEDA [34] is a method that requires devices to have only a button and an LED (or vibration capability). As discussed earlier, group association involves personal devices (such as laptops or smartphones) that offer a relatively rich set of user interfaces. Thus, methods aimed at pairing interface-constrained devices are out of place in a group setting.

[2]As discussed in [19], users might be unwilling to hand over their devices to others during the pairing process, due to security and privacy concerns.

Finally, there are methods involving more exotic components or features. For example, [15] suggests using ultrasound as the OOB channel. A similar approach uses laser as the OOB channel and requires laser transceivers on devices [28]. In "Smart-Its-Friends" [21], a common movement pattern is communicated as a shared secret to both devices by shaking them together. A related technique is recommended in "Shake Well Before Use" [27]. Both methods require each device to be equipped with an accelerometer and involve user-conducted physical shaking/twirling of devices. However, large devices, such as laptops, can not (and should not) be shaken. Also, shaking multiple (e.g., 4 or 6) devices simultaneously is a physically challenging task. Furthermore, it would necessitate handing over one's personal device to others, which most users would prefer to avoid, for security and privacy reasons [19].

STUDY DESIGN AND PRELIMINARIES

We now describe the parameters of our usability study, including selection criteria for tested methods and devices, as well as the architecture of the software platform.

Selection of Methods

As follows from the overview above, there is a large body of prior research literature on secure device pairing and many methods have been proposed in the context of two-device pairing. In total, we identified over twenty methods (including variations). However, as discussed earlier, many are unsuitable for the group setting and are thus eliminated from our study. They include:

- Resurrecting-Duckling: obsolete due to physical interface and cable requirements; also, not scalable to groups.
- Talking-to-Strangers: obsolete, since IR ports have become uncommon;
- Image or phrase comparison: the former requires close proximity and is unsuitable for groups. Numbers are found to be superior to phrases in prior studies.
- Audio-visual comparison and button-enabled transfer: represent overkill for group setting; also, not scalable.
- Camera-based methods: require close proximity and are not well-suited for groups
- Smart-its-Friends, Shake-Well-Before-Use as well as Ultrasound- and Laser-based methods: require interfaces uncommon on current personal wireless devices (e.g., Laptops) and necessitate handing over one's device to others; also, do not scale to groups.

Remaining methods have been included in our study. These include Numeric-Comparison, Copy-and-Confirm and HAPADEP variant. Their inclusion was primarily based upon applicability to a typical devices involved in group association (i.e., phones, PDAs and laptops) as well as *potential* scalability and simplicity.

However, we had to modify selected methods to make them amenable to the group setting. Some methods involve a centralized group member, called a *leader*, while others are *peer-based*. For the latter, we assumed a circular topology.

Recall that, in addition to comparing SAS values, group association requires everyone to correctly count the number of participants. Our methods also depend on how the group size is validated (by input into one's own device or by comparing with the value shown by the device) and how SAS values are validated (by comparing, by copying or by transferring over audio). For each tested method, we provide a brief description of user interactions below.

1. **Leader-VerifySize-VerifySAS (L-VS-VS):** The leader counts group members and inputs the number into its device. If this value does not match the value computed by the device, device indicates failure and the leader warns others to abort the process. Otherwise, the leader's device displays the SAS value as a 5-digit decimal number and the leader announces it to the group. Other members' devices display group size and respective SAS values. Each member verifies group size by counting and comparing their respective SAS values with that announced by the leader. If group size is incorrect or SAS values do not match, a member aborts the process on its device and asks everyone else to do the same. If no one identifies an error, each member accepts group association on its device.

2. **Leader-VerifySize-CopySAS (L-VS-CS):** The leader counts group members and inputs the total into its device. If entered group size does not match that of the device, failure is indicated and the leader warns others to abort the process. Otherwise, the leader's device displays the SAS value encoded as a 5-digit number and the leader announces it to the group. Other members input the announced SAS value into their devices. They also verify group size by counting. If the group size is incorrect or devices indicate failure (SAS value mismatch) they abort the process and warn others to do the same. Otherwise, everyone accepts.

3. **Leader-VerifySize-AudioSAS (L-VS-AS):** Leader's device encodes a 5-digit SAS value as an audio stream, which is then recorded and decoded by other devices. All members verify – by counting – group size displayed on their devices. If the group size is incorrect or devices indicate failure, they abort the process and warn others to do the same. Otherwise, everyone accepts.

4. **Peer-VerifySize-VerifySAS (P-VS-VS):** Each member's device displays group size and a 5-digit SAS value. Each member compares its SAS value with that of the person on their immediate right and verifies group size by counting. In case of failure (SAS value or group size mismatch) a member aborts and tells others to do the same. Otherwise, everyone accepts. (This is the method in *Figure 1*.)

5. **Peer-InputSize-VerifySAS (P-IS-VS):** Each member counts participants and inputs group size into its device. In case of failure (group size mismatch), member aborts and warns others to do the same. Otherwise, each device displays a 5-digit SAS value and each member compares its SAS value with that of their neighbor on the right. In case of a mismatch, a member aborts the process and instructs others to do the same. Otherwise, everyone accepts.

Group Sizes

Group size is an important parameter in our study. It has been claimed that typical group association scenarios are expected to involve less than 10 participants [25]. To this end, our study was originally planned for group sizes up to 10. However, to achieve ecologically valid and statistically evident results, a sufficient number of test instances must be run for each selected group size. Since that would require a large pool of test participants to be simultaneously present to perform the tests, it was infeasible to work with group sizes larger than 6. Therefore, we elected to confine our study to groups of sizes 4 and 6. Armed with the results of – and lessons learned from – our current study as a first step, our future work plans include studies with larger groups.

Selection of Devices

In the entire study, we used Nokia cell-phone model N95[3], as the uniform test device. This model has been released in 2007 and hence does not represent the *cutting edge*. This was done on purpose, to avoid devices with uncommon or expensive features. Nokia N95 has the following features:

- User-input: keypad, microphone
- User-output: speaker, color screen
- Wireless: Wi-Fi, Bluetooth and IR

In all tests, Wi-Fi was used as the wireless (human-imperceptible) channel. We consider this to be a natural choice, since Wi-Fi is widely available and inexpensive. It also allows broadcasting and positioning flexibility within reasonable physical space.

Implementation Details

In comparative usability studies, meaningful and fair results can only be achieved if all methods are tested under similar conditions. In our case, the fair comparison basis is formed by: (1) keeping the same test devices, (2) employing consistent GUI design practices (e.g., safe defaults), and (3) unifying targeted (theoretical) security level for all methods. Our goal is to isolate – to the extent possible – user interaction in different methods as the only independent variable throughout all tests.

To achieve a unified software platform, our implementation used the open-source comparative usability testing framework developed by Kostiainen, et al. [17]. This framework is implemented using JAVA-MIDP and provides basic communication primitives between devices as well as automated logging and timing functionality. However, we had to amend it to accommodate unlimited number of devices and change the inter-device communication channel from Bluetooth to Wi-Fi. We also incorporated support for a control node (i.e., a laptop), used by the test administrator to determine when the next test starts, so as to give users enough time to evaluate the methods. Control node behaves as a wireless network hub among devices simulating a broadcast channel. It also allows protocol messages to be easily deleted, modified or injected to simulate various attack scenarios. We also implemented separate user interfaces and simulated functionality on Nokia devices for all tested methods. Furthermore, we created several test-cases to simulate "no-attack" and "under-attack" scenarios. For all methods, we kept a constant SAS length of 17 bits. Note that, in practice, this length provides a reasonable level of security [40]. We also tried to keep all graphical user interfaces similar with clear instructions and simple language.

We believe that, in our implementation, both the user experience and the interaction model are very realistic. The only difference between our variants and real methods is the omission of initial rounds of the underlying cryptographic protocol that use the wireless channel; they are completely user-transparent and do not influence timings. Instead, our implementation supplies devices with synthetic SAS strings and the group view (i.e., number of devices interacting over the Wi-Fi channel) to easily simulate normal and attack scenarios.

USABILITY TESTING DETAILS

Having implemented all selected group association methods on a common platform, our goal is to evaluate and compare methods with respect to the following measures:

1. Speed: how long each method takes to complete.
2. Robustness: how often each method leads to false positives (rejection of successful association) and false negatives (acceptance of failed association). Following the terminology introduced in [37], we refer to the former as *safe errors* and the latter as *fatal errors*. Recall that errors can occur either during the verification/transfer of SAS or validation of group size.
3. User Feedback: how each method fares in terms of user acceptability.

Study Participants

We recruited 64 participants for our user study which lasted over one month. They were chosen on a first-come first-serve basis from the pool of respondents to recruiting email messages and posters. Participants were randomly split into 4-person or 6-person groups. In total, the study involved seven 4-person and six 6-person groups.

None of the participants reported any physical impairments that could interfere with their ability to complete given tasks. The gender split was 58% male and 42% female. Most participants were university students resulting in a fairly young (80% aged 18-29), well-educated and technology-savvy group. Our study, therefore, represents only the initial step towards identifying methods suitable for the broad cross-section of users.

Test Cases

Three test-cases were considered for each method, simulating normal and abnormal (attack) scenarios. In the former, all information presented to the user was correct (i.e., SAS

[3]For N95 specs, see: `http://www.nokiausa.com/find-products/phones/nokia-n95/specifications`

values matched on all devices and correct group size was shown, whenever applicable). Whereas, in two abnormal cases, two attack types were simulated, as follows:

1. *Insertion Attack:* a realistic and powerful attack, whereby the adversary inserts itself into the protocol as a group member. It results in the same SAS value computed by all devices, while the perceived group size is exceeds by one the actual number of members. To simulate this attack, all participant devices displayed the same SAS value and (incorrect by one) group size.

   In this test-case, we aimed to determine how likely users are to over-count group size. This represents the worst-case scenario: if users can correctly detect insertion of one adversarial node (for a given group size), they can also do so in case of multiple adversarial nodes.

2. *Evil Twin Attack:* a more sophisticated attack, whereby the adversary isolates one member (victim) from the group by tricking it into pairing with virtual (adversarial) nodes. Meanwhile, the adversary inserts itself into the group in place of the victim. This is sometimes called a *group-in-the-middle* attack [20, 7]. If successful, it results in correct group sizes displayed to all users. Moreover, SAS values also match for all devices, except for the victim.

   In simulation, the SAS string fed to one randomly chosen victim device (excluding the leader in leader-based methods) was arbitrarily different from the SAS string fed to all other devices. However, all devices were given the same *correct* group size. Similar to the insertion attack, this test-case represents the worst-case scenario, since this attack is the hardest to detect and requires a fairly sophisticated adversary.

Testing Process

The study was conducted in a conference room at a university campus. After being greeted and guided to the meeting room, participants were asked to take their preferred place around an 8-seat oval-shaped table. They were first given a brief overview of the study and its goals. Then, they were asked to complete a background questionnaire used for collecting demographic information. The questionnaire included a question about the participant's potential visual or hearing impairments, as well as any other conditions that might interfere with their ability to steadily hold objects or type on small keypads (none had these problems).

After completing background questionnaires, participants were given a brief introduction to the cell-phone used in study. Before the actual tests, subjects were asked to imagine themselves at an impromptu meeting where there is a need to securely share documents with others. We also described the methods being tested and different ways of boostrapping a secure communication channel.

Next, each participant was given a device and was asked to follow on-screen instructions to complete each task. The study was conducted as repeated measures; five methods were presented in random order to reduce the learning effect. Various test cases were also randomized within the testing of each method, for the same reason. For leader-based methods, the leader choice was random and was automatically assigned by the test devices. All user interactions and timings were automatically logged by the testing framework.

After finishing three test-cases for each method, subjects completed the System Usability Scale (SUS) questionnaire [5], a widely used and highly reliable 10-item 5-point Likert scale. It polls satisfaction with computer systems [3], in order to assess usability of the method that was just tested. We used the original questions from [5], except that "system" was replaced with "method". Via an additional question, subjects rated perceived security of each method on the same scale.

On average, it took about 40 minutes for each group to finish the entire process. Subjects were allowed to participate in this study only once and each subject was rewarded with two movie tickets.

Test Results

We collected data in two ways: (1) by timing and logging user all interaction, and (2) via questionnaires.

Method Name	Group Size	Successful Completion Rate	Avg. Completion Time in secs	Avg. SUS Score	Perception of Security
L-VS-AS	4	85.7%	36.71 (6.46)	62.86 (3.86)	2.96 (0.10)
	6	100.0%	42.33 (9.71)	67.85 (5.20)	3.25 (0.21)
L-VS-CS	4	85.7%	51.57 (9.34)	64.91 (4.07)	3.07 (0.32)
	6	83.3%	49.67 (6.24)	65.97 (4.38)	3.31 (0.23)
L-VS-VS	4	100.0%	41.29 (7.77)	68.75 (4.01)	3.36 (0.33)
	6	100.0%	31.33 (4.64)	65.76 (2.39)	3.25 (0.23)
P-IS-VS	4	85.7%	27.57 (3.66)	68.57 (3.87)	3.57 (0.09)
	6	100.0%	38.83 (3.00)	77.36 (5.16)	4.14 (0.24)
P-VS-VS	4	100.0%	40.43 (8.64)	63.57 (3.31)	3.14 (0.27)
	6	100.0%	43.00 (8.27)	74.03 (5.96)	3.75 (0.35)

Values in paranthesis show the standard error of the mean.

Figure 2. Summary of Usability Measures for Each Method

Figure 2 summarizes usability measures for each tested method and group size. These include: successful completion rate[4], completion time for normal (no attack) test-cases, SUS scores, and ratings of perceived security. *Figure 3* shows results corresponding to simulated attacks.[5]

ANALYSIS AND INTERPRETATION

We now analyze and (attempt to) interpret the study results. We first consider various mechanical data, i.e., time to completion and error rates. Then, we analyze user feedback (i.e., perceived usability and security), and, finally, evaluate overall usability and security by considering all data collectively.

Interpreting Time and Error Results

Our results, reflected in *Figure 2* and *Figure 3*, prompt a number of observations. One way to interpret them is by looking at completion times under normal (no attack) circumstances.[6] Based on this metric, as reflected in *Figure 2*,

[4]Safe error rate can be calculated as a complement of success rate (i.e., safe error rate = 1− successful completion rate).

[5]Fatal error rate can be computed as a complement of secure completion rate (i.e., fatal error rate = 1− secure completion rate).

[6]Note that, in most real-world settings, attacks would not occur.

Test Case	Method Name	Group Size	Avg. Completion Time in secs	Secure Completion Rate
Insertion Attack	L-VS-AS	4	23.57 (4.67)	100.0%
		6	31.83 (6.72)	100.0%
	L-VS-CS	4	47.14 (14.1)	100.0%
		6	48.50 (5.93)	100.0%
	L-VS-VS	4	52.14 (18.4)	100.0%
		6	36.67 (11.8)	100.0%
	P-IS-VS	4	44.57 (16.4)	100.0%
		6	57.50 (13.00)	100.0%
	P-VS-VS	4	37.86 (15.1)	100.0%
		6	33.00 (6.87)	**83.3%**
Evil Twin Attack	L-VS-AS	4	33.86 (9.67)	100.0%
		6	26.33 (5.94)	**66.7%**
	L-VS-CS	4	42.00 (7.03)	**57.1%**
		6	36.33 (4.39)	**50.0%**
	L-VS-VS	4	29.00 (2.54)	100.0%
		6	36.00 (5.90)	**50.0%**
	P-IS-VS	4	28.00 (6.20)	100.0%
		6	36.17 (10.5)	**66.7%**
	P-VS-VS	4	32.43 (6.23)	100.0%
		6	29.50 (2.08)	**83.3%**

Values in paranthesis show the standard error of the mean.

Figure 3. Secure Completions Under Attack Simulations

all methods are fairly fast, taking less than a minute to complete.

However, pairwise t-tests (paired) revealed that L-VS-CS is significantly ($p < 0.05$) slower than L-VS-VS and P-IS-VS for both group sizes. This result is intuitive, since copying numbers (in L-VS-CS) is more time-consuming than comparing them (in L-VS-VS and P-IS-VS). In terms of average completion times, the fastest method is P-IS-VS (27.57 sec.) for smaller groups and L-VS-VS (31.33 secs.) for larger groups.

Looking at safe error rates, most methods fare relatively well in normal test-cases. However, L-VS-VS and P-VS-VS stand out with 100% successful completion rates. On the lowest end, L-VS-CS yields the most safe errors for both group sizes.

In terms of simulated attacks, we see that all methods, except P-VS-VS, report no fatal errors under insertion attacks. They all yield 100% secure completion rate with no group member accepting an erroneous pairing. This is an encouraging result showing the near impossibility of insertion attacks for groups of size 4 and 6. We anticipate this to also hold for groups of sizes up to 10.

On the other hand, evil twin attacks are quite effective, yielding much higher fatal error rates. In particular, L-VS-CS appears vulnerable to this attack, since at least one group member accepted the pairing in almost half of attack cases, irrespective of group size. In contrast, P-VS-VS seems to be the most resilient, as only one of thirteen groups had a member who erroneously accepted a tainted pairing. Error rates also suggest that larger groups are more susceptible to evil twin attacks. Recall, however, that this is the strongest attack on a group association protocol, which is also difficult for an adversary to launch. In practice, multiple devices would wind up with mismatched SAS values and this attack is more likely to be detected.

Taking both speed and error rates into account, P-VS-VS, P-IS-VS and L-VS-AS performed well overall. Whereas, L-VS-CS turns out to be the slowest and the most error-prone.

Interpreting User Feedback

System usability scale (SUS) is a popular set of 10 questions for assessing user satisfaction with a computer system. Considering that industry average for SUS scores tends to hover in the 60–70 range [24], all tested methods fare relatively well in achieving mean SUS scores over 65. However, P-IS-VS is clearly perceived as the most usable method (considering overall scores for groups of both sizes). Pairwise t-tests (paired) revealed that SUS scores for P-IS-VS were significantly higher ($p < 0.01$) than those for all other leader-based methods. This seems to indicate that, for small groups, peer-based methods are generally more acceptable than leader-based ones. This might be because users found it more appealing and less burdensome to interact with nearest seated peers than to coordinate with the leader.

There is also evidence ($p < 0.054$) suggesting that P-IS-VS is perceived as more usable than P-VS-VS. Since the only difference between them is the way of validating group size, the implication is that users prefer entering group size over verifying it.

Looking at perceived security ratings, P-IS-VS is again the clear winner, showing observably higher scores than other methods. P-VS-VS scores as the second highest. These results indicate that peer-based methods are also perceived as more secure than their leader-based counterparts.

Usability Measures Combined

An ideal association method should perform well with respect to all usability measures discussed so far. To better understand correlations among our measures, we performed linear cross-correlations among all four: completion time, successful completion rate, SUS score, and perceived security. *Table 1* shows the correlation coefficients and their respective statistical significance.

	Successful Completion Rate	Completion Time	SUS Score
Completion Time	-0.200	-	
SUS Score	0.371**	-0.304*	-
Perceived Security	0.045	0.053	0.604**

Table 1. Cross-Correlation of Usability Measures ("*" denotes $p < 0.05$ and "" denotes $p < 0.01$)**

In the domain of Social Sciences, correlation coefficients in ranges $[-0.3, -0.1]$ and $[0.1, 0.3]$ are generally regarded as small, while $[-0.5, -0.3]$ and $[0.3, 0.5]$ – as medium [8]. The only high correlation we observed is between SUS score and perceived security, which seems to indicate that group association methods rated as usable were also perceived to

be secure. However, in line with the findings of [10], we cannot consider any usability measure to be sufficiently correlated with all others that it could be justifiably omitted. On the other hand, since measures are indeed correlated to varying degrees, we are motivated to also consider them as a whole. To this end, we next present cluster analysis of our usability measures, based on principal components. This was done to identify methods closely related in terms of all usability measures.

	PC1	PC2	PC3	PC4
Eigenvalue	1.817	1.134	0.789	0.260
Proportion of Variance	0.454	0.283	0.197	0.065
Cumulative Proportion	0.454	0.738	0.935	1.000

Table 2. Principle Components of Usability Measures

Table 2 shows four principal components that clarify 100% of data variance. The first component, PC1, accounts for about 45.4% of the variance, and the second component, PC2, adds 28.3%.

Since PC1 and PC2, together, account for over 70% of the variance, we can disregard PC3 and PC4 for all practical purposes, since they contribute so little. *Table 3* shows factor loadings of PC1 and PC2. We see that completion time loads negatively, while all other measures show positive loading with respect to PC1. Consequently, higher PC1 scores can be interpreted as indicative of better overall usability. However, we cannot interpret PC2 in a similar (simple) manner.

	PC1	PC2
Successful Completion Rate	0.416	-0.439
Completion Time	-0.323	0.642
SUS Score	0.683	0.113
Perceived Security	0.506	0.618

Table 3. Factor Loadings of PC1

Figure 4 shows mean values of PC1 and PC2 scores for all five methods. Since higher PC1 values are representative of better overall usability, methods towards the right can be regarded as more usable in general. Three observed clusters (using the Euclidian distance and average linkage method) based on principal components are also superimposed on

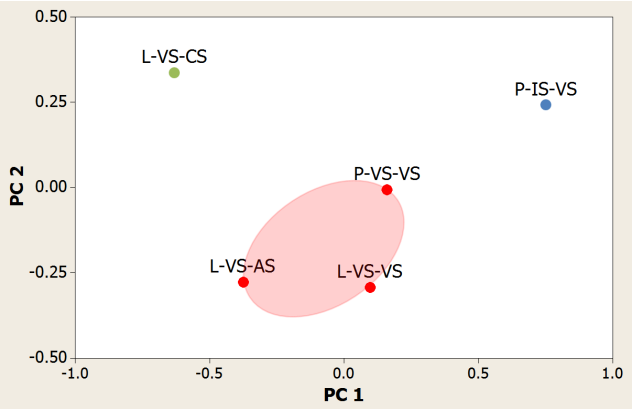

Figure 4. Method scorings and clusters based on PC1 and PC2

Figure 4. P-IS-VS forms a cluster by itself as the most usable method. L-VS-CS forms another cluster as the least usable method, while the remainder (L-VS-AS, L-VS-VS, P-VS-VS) fall into the third cluster with *reasonable* overall usability.

CONCLUSIONS AND SUMMARY OF RESULTS

This paper presented the first experimental evaluation of prominent group device association methods. Our observations are summarized as follows:

– P-IS-VS clearly exhibits the best overall usability. Fortunately, it also shows high secure completion rates (i.e., low fatal error rates) under simulated attacks. Thus, we view it as one of best methods for group association.

– For methods with reasonable overall usability, P-VS-VS can always be replaced by P-IS-VS, since the latter is more usable. Thus, L-VS-VS and L-VS-AS are natural choices if circular topology with reasonable pair-wise user proximity is hard to achieve. L-VS-VS is probably the better of the two, since it yielded 100% successful completion rate and is more suitable for a noisy environment.

– All methods, except P-VS-VS, are resistant to node insertion attacks for 4- and 6-user groups. We anticipate the same to hold for slightly larger groups (up to 10).

– Entering group size was perceived as more usable and more secure than verifying group size (as displayed by one's device).

– In small groups (sizes 4 and 6), peer-based methods seem to be generally more acceptable. They also provide a stronger sense of security than their leader-based counterparts.

We believe that our work represents an important and timely first step in exploring real-world usability of group association methods. Our results show that certain simple methods (P-IS-VS and L-VS-VS) are quite attractive overall: fast, reasonably secure and acceptable by users. In terms of future work, we plan to conduct usability studies with more diverse user samples as well as with larger groups (e.g., $7-10$ users). Another direction is the inclusion of multiple types of devices, e.g., smartphones and laptops of different makes, models and software platforms.

Acknowledgements

We sincerely thank Ubicomp'10 anonymous reviewers for their helpful feedback. This work was supported, in part, by NSF Cybertrust awards CNS-0831397, CNS-0831526, IIS-0953071 and Google research award "Secure and Usable Group Association of Personal Wireless Devices".

REFERENCES

1. N. Asokan and P. Ginzboorg. Key agreement in ad hoc networks. *Computer Communications*, 23(17):1627–1637, 2000.
2. D. Balfanz et al. Talking to strangers: Authentication in ad-hoc wireless networks. In *Network and Distributed System Security Symposium (NDSS)*, 2002.
3. A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008.

4. V. Boyko, P. MacKenzie, and S. Patel. Provably Secure Password-Authenticated Key Exchange Using Diffie-Heilman. In *International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt)*, 2000.
5. J. Brooke. SUS: a "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, editors, *Usability Evaluation in Industry*. Taylor and Francis, London, 1996.
6. M. Cagalj, S. Capkun, and J. Hubaux. Key agreement in peer-to-peer wireless networks. In *Proceedings of the IEEE (Special Issue on Cryptography and Security)*, 2006.
7. C.-H. O. Chen et al. Gangs: gather, authenticate 'n group securely. In *MobiCom: ACM international conference on Mobile computing and networking*, 2008.
8. J. Cohen, P. Cohen, S. G. West, and L. S. Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.
9. C. M. Ellison and S. Dohrmann. Public-key support for group collaboration. *ACM Transactions on Information and System Security (TISSEC)*, 6(4):547–565, 2003.
10. E. Frkjr, M. Hertzum, and K. Hornbk. Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *CHI: Conference on Human Factors in Computing Systems*, 2000.
11. C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *RSA CryptoBytes*, 7(1):29 – 37, 2004.
12. I. Goldberg. Visual Key Fingerprint Code. `http://www.cs.berkeley.edu/iang/visprint.c`, 1996.
13. M. Goodrich et al. Audio-based secure device pairing. In *International Journal of Security and Networks (IJSN)*, volume 4, 2009.
14. R. Kainda, I. Flechais, and A. W. Roscoe. Usability and security of out-of-band channels in secure device pairing protocols. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.
15. T. Kindberg and K. Zhang. Validating and securing spontaneous associations between wireless devices. In *Information Security Conference (ISC)*, pages 44–53, 2003.
16. A. Kobsa, R. Sonawalla, G. Tsudik, E. Uzun, and Y. Wang. Serial hook-ups: A comparative usability study of secure device pairing methods. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.
17. K. Kostiainen and E. Uzun. Framework for comparative usability testing of distributed applications. In *Security User Studies: Methodologies and Best Practices Workshop*, 2007.
18. A. Kumar et al. Caveat Emptor: A Comparative Study of Secure Device Pairing Methods. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2009.
19. A. Kumar, N. Saxena, and E. Uzun. Alice meets bob: A comparative usability study of wireless device pairing methods for a "two-user" setting. *CoRR*, abs/0907.4743, 2009.
20. C. Kuo, A. Studer, and A. Perrig. Mind your manners: socially appropriate wireless key establishment for groups. In *WiSec: ACM conference on Wireless network security*, 2008.
21. L. Holmquist et al. Smart-its friends: A technique for users to easily establish connections between smart artifacts. In *ACM International Conference on Ubiquitous Computing (Ubicomp)*, 2001.
22. S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. *International Conference on Cryptology and Network Security (CANS)*, 4301:90–107, 2006.
23. S. Laur and S. Pasini. Sas-based group authentication and key agreement protocols. In *Public Key Cryptography*, 2008.
24. J. Lewis and J. Sauro. The factor structure of the system usability scale. In *Human Computer Interaction International Conference (HCII)*, 2009.
25. Y.-H. Lin et al. Spate: small-group pki-less authenticated trust establishment. In *MobiSys: Conference on Mobile systems, applications, and services*, 2009.
26. M. Goodrich et al. Loud and Clear: Human-Verifiable Authentication Based on Audio. In *International Conference on Distributed Computing Systems (ICDCS)*, 2006.
27. R. Mayrhofer and H. Gellersen. Shake well before use: Authentication based on accelerometer data. *International Conference on Pervasive Computing (Pervasive)*, 2007.
28. R. Mayrhofer and M. Welch. A human-verifiable authentication protocol using visible laser light. In *IEEE International Conference on Availability, Reliability and Security*, 2007.
29. J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *IEEE Symposium on Security and Privacy*, 2005.
30. N. Saxena et al. Extended abstract: Secure device pairing based on a visual channel. In *IEEE Symposium on Security and Privacy*, 2006.
31. S. Pasini and S. Vaudenay. SAS-Based Authenticated Key Agreement. In *International Conference on Theory and Practice of Public-Key Cryptography (PKC)*, 2006.
32. A. Perrig and D. Song. Hash visualization: a new technique to improve real-world security. In *International Workshop on Cryptographic Techniques and E-Commerce*, 1999.
33. R. Prasad and N. Saxena. Efficient device pairing using "human-comparable" synchronized audiovisual patterns. In *Applied Cryptography and Network Security (ACNS)*, 2008.
34. C. Soriente, G. Tsudik, and E. Uzun. BEDA: Button-Enabled Device Association. In *International Workshop on Security and Privacy in Spontaneous Interaction (IWSSI)*, 2007.
35. C. Soriente, G. Tsudik, and E. Uzun. HAPADEP: human-assisted pure audio device pairing. In *Information Security Conference (ISC)*, pages 385–400, 2008.
36. F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols Workshop*, 1999.
37. E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Financial Cryptography and Data Security*, pages 307–324, 2007.
38. V. Roth et al. Simple and effective defense against evil twin access points. In *ACM Conference on Wireless Network Security (WiSec)*, pages 220–235, 2008.
39. J. Valkonen, N. Asokan, and K. Nyberg. Ad hoc security associations for groups. In *Security and Privacy in Ad-Hoc and Sensor Networks (ESAS)*, 2006.
40. S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *International Cryptology Conference (CRYPTO)*, 2005.