Spatial Co-Location for Device Association : The Connected Object Way

Ming Ki Chong
Lancaster University, UK
mingki@acm.org

Fahim Kawsar
Bell Labs, Belgium
fahim.kawsar@alcatel-lucent.com

Hans Gellersen
Lancaster University, UK
hwg@comp.lancs.ac.uk

ABSTRACT

Device association is one of the most common interaction primitives used in today's mobile device space. Yet, the existing approaches rely on traditional (e.g., sharing a pass key across devices, etc.) or tangible methods (e.g., bumping devices together, shaking with a similar pattern, etc.) and are mostly limited to pairing two devices. In this paper, we present "GroupTap", an intuitive and secure device association mechanism for multiple devices. It leverages spatial co-location features while ensuring tangibility and user centric control for better user experience. The fact that mobile devices need to be physically co-located for forming an association, the spatial co-location could be used as a trigger for initiating the association. For conforming co-location, we exploit connected physical objects and simple tapping interaction, i.e., devices need to tap one or multiple connected physical objects (placed in a particular location) to convey association request. We envisage that this approach is simple, rapid, efficient and would yield a superior user experience. We discuss the design rationales and present technical details of GroupTap in this paper.

Author Keywords

Device Association, Connected Objects, Spatial Co-Location.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation (e.g., HCI): User Interfaces—*Interaction styles*; K.6.5 Management of Computing and Information Systems: Security and Protection—*Authentication*.

General Terms

Design, Human Factors, Security.

INTRODUCTION

Mobile devices (such as cellular phones, tablet computers, media players, etc.) now widely support wireless ad hoc networking. Wireless networking enables users to establish device connections and share resources in a spontaneous and serendipitous manner. Yet, before devices can transfer any data wirelessly, they are required to establish a virtual connection amongst one another, a process known as *device association*[1]. Ideally, devices should connect automatically when requested. However, to avoid accidental, unwanted or untrusted communication, user mediation is essential, where users initiate and control the process by explicitly identifying the target devices they wish to connect.

Bluetooth pairing is a common and widely deployed association mechanism nowadays. A device first scans its surroundings and displays the available devices to its user. The user then selects the name of the target device. If required, the user also enters a passkey into the connecting devices. This scheme is a tedious process. For instance, the scanning operation requires a waiting period, which elongates the overall procedure. Also, the target device's name needs to be known beforehand and the user needs to correlate the name with the actual device – multiple devices with the same name can cause ambiguity. To overcome such issues, alternative approaches have been demonstrated in research. Some focused on simplicity of user interaction (e.g., bumping devices [4], shaking devices [6], synchronous button pushes [10]), others were motivated by technology (e.g., touch screens [5], intra-body communication [9], short-range communication [11, 12]) as well as addressing security issues (e.g., interactive authentication protocols [1, 8, 14]).

Research has shown many methods for associating devices, but the work often focused on pairing two devices. To associate multiple devices, a straightforward way is adopting those techniques to have each device to pair with a master device individually. The master device then bridges the connections amongst other devices. This consequently requires $N - 1$ associations (where N is the number of devices), and hence, as the device cardinality increases, the number of associations with the master device increases accordingly. Several research work reduced the number of associations by using a single shared passkey [2, 15]. However, those techniques remains cumbersome, as the device discovery process still exists. In situations where users only need a transient connection (e.g., file transfer, where the connection is no longer needed after the file is sent), it is unnecessary to burden the users with a long setup procedure. Instead, a fast

NoME-IoT'11, September 18, 2011, Beijing, China.

[1] Other literature has adopted alternative terminologies, such as *pairing*, *binding*, or *coupling* of devices. They essentially refer to the same underlying concept – establishing an ad hoc network amongst multiple devices.

association approach is required.

In this paper, we present *GroupTap*, a new interaction technique that leverages spatial proximity to associate multiple devices. GroupTag simplifies association by reducing the number of individual associations and user interactions amongst the devices, eliminating the device discovery process as the users do not need to explicitly select the target devices. GroupTap promotes an intuitive way of touching one or multiple common physical objects spatially co-located with the devices as an explicit trigger for an association. Consequently, the notion of smart connected objects (uniquely identifiable, instrumented with awareness technology and connected to Internet for sociality) fits naturally for GroupTap as a co-location facilitator. The association process starts with devices touching one or multiple connected objects sequentially and thereby explicitly placing the control at users hand while ensuring tangibility, ease of interaction and spontaneity. The resulting process is simple, rapid and secure as we will demonstrate in this paper. In what follows, we discuss the design space for GroupTap, followed by its technical details. We end the paper by discussing the limitations of GroupTap and reflecting its implications on other connected object services.

DESIGN SPACE

In this section, first we present a scenario to explain our basic proposition – exploiting spatial co-location of devices for forming an association with connected objects as co-location facilitator.

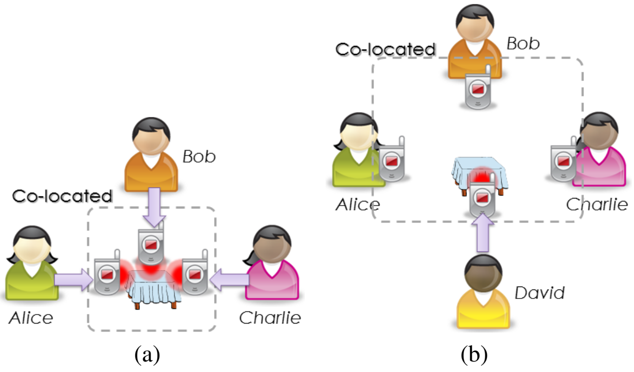


Figure 1. Instantiation of co-location. (a) Alice, Bob and Charlie tap their devices on an augmented smart table to prove that their devices are co-located around the table. (b) A new party, David, wants his device to join the co-location, so he taps his device on the same table.

Our approach here is to use the identity and the location reference of augmented objects to represent co-location of devices. Using proximity-based technology (e.g., Near Field Communication), a mobile device can capture the identity of an object if they are in close proximity. Pursuing this concept, when multiple devices capture the same identity, the action implies that they are all co-located. For example, Alice, Bob and Charlie, each of them has a mobile device (see fig. 1(a)). To prove that their devices are co-located, they first choose a nearby object (e.g., a table) as a location reference. They then tap their devices on the object to capture its unique identity. By doing so, only the involved devices can present themselves as co-located at that specific reference point.

The process of proving co-location – since only the involved devices could be present – can be capitalised as an out-of-band channel[2] for forming secure device association. Continuing the example earlier, if a fourth party, David, also wants his device to be co-located with the devices of Alice, Bob and Charlie, David needs to physically bring his device to capture the same object's identity (see fig. 1(b)). His action would be obvious to the Alice, Bob and Charlie, as they are all present at the reference location. Hence, anyone who is involved in the procedure of proving co-location would immediately be recognised by other users. This is only achievable if the process of capturing an object's identity requires the presentation of the device in front of other users; hence, enabled by the short distance of NFC technology.

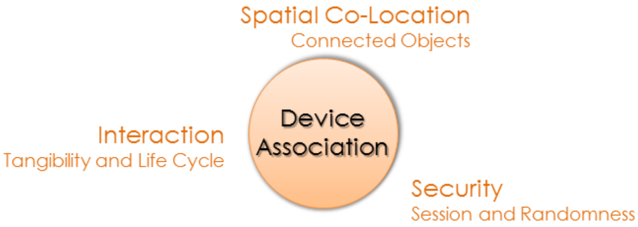


Figure 2. Design space for group device association

Considering the above scenarios, we see three major design aspects for GroupTap, as shown in fig. 2. In the following we discuss these aspects and their corresponding implications in GroupTap.

Spatial Co-Location - Connected Objects

GroupTap exploits the fact that devices required to be physically close to one another for forming an association. Such proximity can be reasoned in several ways, e.g., Bluetooth uses device discovery mechanism to search nearby devices. However, GroupTap takes an intuitive approach leveraging the availability of physical objects. Amongst the physical objects around us many of them have a designated location. The static nature of these objects provides location reference points [7]. GroupTap utilises these objects for reasoning the spatial co-location of devices. Considering the increasing instrumentations of everyday objects and their connectivity to the Internet (so called the Internet of Things), we assume that physical objects can be easily extended to support such association services. For example, in the above scenarios, the table acts as the location reference point and facilitates the association formation. For supporting association through connected objects, there are two further design considerations:

[2] In device association, wireless messages are transmitted over an insecure channel (the in-band channel), authentication data for establishing a connection can be transmitted over an external channel, known as the out-of-band (OOB) channel. Ideally, an OOB channel is impervious by adversaries while only the legitimate users have full control over the channel.

- **Connected Object Cardinality**: Association could be formed by utilising one or multiple connected objects. The simplest approach is using a single object to prove co-location; hence, the interaction centralises on one object. With only one object, it allows fast interaction, as each user only needs to tap their device on the object once. On the contrary, although the use of multiple objects requires greater effort from users, it provides higher randomness and it has a direct implication in securing the association phase as we will discuss later.

- **Connected Object Mobility**: Physical objects could be static or mobile and both could be used for association.

  - **Static objects** are fixed in location. They do not need to be situated or anchored to a physical building. A static object has a physical designated spot where it permanently resides. For example, a refrigerator in a kitchen usually remains in same spot once it is installed. Static objects provide location references, as they are stationary.
  - **Mobile objects** are not fixed in a permanent spot. They are often moved. For example, a watch, a coffee mug, etc. They cannot be attached to any static object, otherwise they become part of it. Since mobile objects do not have a permanent location, they cannot be used as a location reference. Nonetheless, as long as an object is physically visible to users, they can still use a mobile object to prove co-location.

  Mobility provides unpredictability, as it is difficult to predict when an object will be present at a specific location and time. Thus, mobile objects provide higher randomness than static objects. The use of mobile objects enhances security. Since it is harder for an assailant to predict users' selection of mobile objects, it is also harder for the assailant to join their co-location.

Interaction - Tangibility and Association Life Cycle

The next design aspect is the interaction modality. We argue tangibility brings simplicity, spontaneity and better user centric control as users can explicitly convey their association request. At the same time this is easily achievable using NFC technology. Accordingly, we have decided to use tapping as the prime interaction technique for forming the association, i.e., users carrying devices tap NFC tag augmented objects to initiate an association by identifying the target object. However, just initiating the association is inadequate. For instance, after a group of users have proven their devices are co-located, they may move to another location. After they have moved, other people can utilise the same object. Obviously, the new group should not be identified as co-located with the previous group. For this, we need a mechanism to denote the end of a co-location identification. We proposed two methods to trigger the denotation.

- **Timeout Threshold**: A naïve approach is to let the event time out. As soon as the first device captures the identity of a selected object, the timer starts. Every time another device captures the object's identity the timer resets. Once all of the involved devices have captured the object's identity, the co-location procedure expires after an idle period of receiving no new request for the object's identity.

- **Explicit Action**: An alternative method is having a user performing an explicit action, which denotes the end of the procedure. A user can explicitly send a command to the system, which informs the system that all involved devices have been identified, and thus, concluding the process. For example, when all of the devices are done, the first device that captured the object's identity can perform the same action again. By recording the object's identity twice, the system can acknowledge it as a termination command.

The timeout method allows the system to automatically conclude the procedure. However, the duration of the timeout interval can affect usability. A long duration can cause unnecessary waiting, whilst a short duration can conclude the process prematurely. On the other hand, the explicit action method also has its pro and con. Whilst the method gives users the control of ending the process immediately, the users need to constantly remember that a member needs to input the last action. Fortunately, the two methods are not exclusive of each other. A system can implement both triggering conditions. So, if users forget to input the last action, the timeout method will be a fail-safe function that prevents the process running infinitely.

Security

The final design aspect is the assurance of security of the association procedure so that the consequent communication channel is secure. Security has a direct relationship with the number of connected objects used in the association as well as the temporal span of the association phase. Accordingly, we observe two aspects of security:

- **Object Cardinality**: Using single object radically simplifies the interaction, but bringing multiple objects into the interaction for association provides higher randomness, which results in better security. For example, if a room has N objects, the chances of users selecting an object is $\frac{1}{N}$. However, if M objects (where $M \geq 2$) were selected, the chances of guessing the correct objects differ greatly. If the order of identifying the objects matters, we have $\frac{1}{{}_N P_M}$ (where the function ${}_n P_r$ denotes permutation without repetition), otherwise, we have $\frac{1}{{}_N C_M}$ (where ${}_n C_r$ denotes combination). The use of combination should be avoided, as it decreases the system's randomness drastically. The more objects available in the environment, the harder an assailant would be able to guess the select objects correctly. Nonetheless, this is only a theoretical estimation. In reality, social engineering may provide a better estimation of users' choices.

- **Session-based Information**: A smart object's identity (ID) is acquired during the instance of users establishing a co-location of devices. If the ID is constant and reused, the ID is known to other devices that have previously used the same object. Consequently, an unwanted party can use the known ID to join the co-location remotely, without presenting oneself. To avoid this, the system needs to utilise session based information (similar to one-time passwords). New information is generated for every co-location session. This forces the involved users to be physically present by the vicinity of the object to acquire this information. Hence, having only previously stored information is insufficient.

In the next section we present the actual interaction mechanism for the proposed association procedure along with some technical details and discuss how these design considerations are addressed in our approach.

ASSOCIATION PROCEDURE

User Interaction

The types of interaction are categorised according to the number of objects used to association devices. If a single object is used, users only interact with the chosen object. The process begins with a group of people wanting to associate their mobile devices in an environment with smart objects. The users first agree upon and select a nearby smart object for the association. One of the users (i.e., the group leader) elects to initiate a co-location by tapping his/her device on the selected object. The rest of the users then follow by performing the same action. Once every device has tapped on the object, the leader taps his/her device on the object again to denote the end of the co-location session (i.e., an explicit action).

For association that involves multiple objects, the group of users first elect a leader. The leader then randomly picks several objects within his/her vicinity, and taps his/her device on the objects in a sequence. The rest of the users follow by tapping on the same objects, in the same sequential order. Once every user is done, the leader taps his/her device on the first object again to denote the end of the co-location session. The device association is established only if all of the devices tapped on the correct objects in the right order.

Establishing a Secure Virtual Connection

Valkonen et al. presented two protocol approaches, *Numeric Comparison* and *Passkey Based Protocol*, for establishing secure group associations [15]. Each of the approaches requires different user action. Numeric comparison requires the users to compare an output value from the devices. The association is only successful if every device outputs the same value; otherwise, a fault/threat exists. With passkey based protocol, the users agree upon a passkey (e.g., a password) and they enter the passkey into their devices. The protocol uses the passkey as a shared secret for authentication.

The former approach is unsuitable for the interaction of GroupTap, as it requires an extra step of comparing numbers. Our design adopts the passkey based approach. However, instead of users manually selecting a passkey, we simplify the process by having a device to generate the passkey automatically (i.e., a session-based information). The device then implicitly shares the passkey amongst other devices during the user interaction.

Single Object Group Association

We begin with illustrating association using a single object. To start, the users mutually select a smart object (A_1) and one device, say D_1, to lead the GroupTap association[3].

1. D_1 taps on A_1.

2. D_1 generates a session-based random number K_{Rand}.

3. D_1 sends K_{Rand} and its network address $Addr_{D_1}$ (i.e., a Wi-Fi MAC address) to the object A_1 via the NFC channel. The object stores the information temporarily for the duration of the session.

   $\mathsf{D}_1 \rightarrow \mathsf{A}_1 : K_{Rand}$ and $Addr_{D_1}$

4. The remaining devices $\mathsf{D}_{2,...,n}$ ($n =$ the number of devices) tap on the same object and receive the random number as well as D_1's address via the NFC channel. Also, the devices send their network addresses $Addr_{D_{2,...,n}}$ to the object.

   $\mathsf{D}_i \leftarrow \mathsf{A}_1 : K_{Rand}$ and $Addr_{D_1}$, where $i = 2, ..., n$

   $\mathsf{D}_i \rightarrow \mathsf{A}_1 : Addr_{D_i}$, where $i = 2, ..., n$

5. After the devices have received K_{Rand} from the object, D_1 taps on the object again and reads the network addresses of $\mathsf{D}_{2,...,n}$. In addition, D_1 sends a *terminate* message to the object to erase K_{Rand}. This prevents an unwanted device from associating with the group.

   $\mathsf{D}_1 \leftarrow \mathsf{A}_1 : Addr_{D_i}$, where $i = 2, ..., n$

   $\mathsf{D}_1 \rightarrow \mathsf{A}_1 : Message_{terminate}$

   A_1 : erase K_{Rand}

6. D_1 now has the addresses of $\mathsf{D}_{2,...,n}$, and vice versa, without going through a device discovery process. The group of devices execute Valkonen et al.'s passkey based protocol [15] and form a group association using K_{Rand} as a shared passkey.

Multiple Objects Group Association

We adjust the above protocol to fit associations that use multiple objects. To start, the users select several smart objects ($\mathsf{A}_{1,...,m}$, where $m =$ the number of objects) and one device, say D_1, to lead the GroupTap association.

1. D_1 taps on $\mathsf{A}_{1,...,m}$ in a sequence.

   For each object that D_1 *has tapped on*:

2. D_1 generates a new random number, K_{Rand_j}, for the selected object.

3. D_1 sends K_{Rand_j} to the object A_j via the NFC channel, but only sending its network address $Addr_{D_1}$ to the first object A_1. The object stores the information temporarily for the duration of the session.

   $\mathsf{D}_1 \rightarrow \mathsf{A}_j : K_{Rand_j}$, where $j = 1, ..., m$

   $\mathsf{D}_1 \rightarrow \mathsf{A}_1 : Addr_{D_1}$; *for A_1 only

[3] Variables in sans-serif font indicate physical objects or devices.

Repeat step 2 and 3 until every object has received a random number.

4. The remaining devices $\mathsf{D}_{2,...,n}$ tap on the same objects in the same sequence. The devices receive the random numbers as well as D_1's address via the NFC channel. Also, the devices send their network addresses $Addr_{D_{2,...,n}}$ only to the first object, A_1.

   $\mathsf{D}_i \leftarrow \mathsf{A}_1 : K_{Rand_1}$ and $Addr_{D_1}$, where $i = 2, ..., n$

   $\mathsf{D}_i \leftarrow \mathsf{A}_j : K_{Rand_j}$, where $i = 2, ..., n$; $j = 2, ..., m$

   $\mathsf{D}_i \rightarrow \mathsf{A}_1 : Addr_{D_i}$, where $i = 2, ..., n$; *for A_1 only

5. After the devices have received K_{Rand_j} from the objects, D_1 taps on the first object to read other devices' network addresses. In addition, D_1 sends a $terminate$ message. The object then forwards the message to the remaining objects via the Internet. Upon receiving the message, they erase $K_{Rand_{1,...,m}}$.

   $\mathsf{D}_1 \leftarrow \mathsf{A}_1 : Addr_{D_i}$, where $i = 2, ..., n$

   $\mathsf{D}_1 \rightarrow \mathsf{A}_1 : Message_{terminate}$

   $\mathsf{A}_1 \rightarrow \mathsf{A}_j : Message_{terminate}$, where $j = 2, ..., m$ [4]

   A_j : erase K_{Rand_j}, where $j = 1, ..., m$

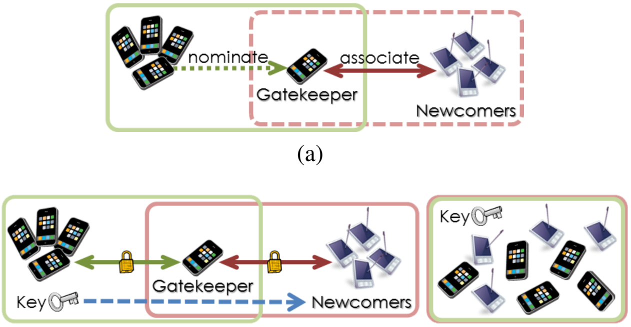
6. D_1 now has the addresses of $\mathsf{D}_{2,...,n}$, and vice versa. The group of devices then execute Valkonen et al.'s passkey based protocol [15] and form a group association using $h(K_{Rand_1} \mid K_{Rand_2} \mid ... \mid K_{Rand_m})$ as a shared passkey[5].

In step 3, D_1 sends a random number and its address to the first artefact (A_1), while the remaining of artefacts only receive a random number. By doing this, it eliminates the redundant transfer of $Addr_{D_1}$ in step 4. Each of the peer devices only reads the address once.

The hash function in step 6 requires its inputs $K_{Rand_{1...m}}$ to be concatenated in the right sequence across all devices; hence, permutation. If the inputs are not in the correct sequence, the hash function produces a difference output value; thus, the association is unsuccessful.

Joining New Members

The above protocols show methods of forming a new group association. In reality, an associated group may want new devices to join their existing association. To include new devices, the original group can nominate one or more devices as a gatekeeper. The gatekeeper associates with the newcomers, using the single- or multi-object approach. After the gatekeeper has associated with the new members, the gatekeeper provides the necessary information (e.g., the encryption key from the original group) for the new members to join the original group [15]. Alternatively, all of the devices can negotiate a new group key. After the newcomers have joined the original group, the sub-association between the gatekeeper and the newcomers is discarded. See fig. 3 for an illustration.



(a)

(b)

(c)

Figure 3. An illustration of new devices joining a pre-associated group. The rounded corner boxes indicate group associations and the arrows indicate actions. (a) The pre-associated group nominates a gatekeeper. The newcomers establish an association with the gatekeeper. (b) The gatekeeper is securely associated with its original group as well as the new group. The original group sends its key to the new group via the gatekeeper. (c) The devices are associated and share the group key. The association between the gatekeeper and the newcomers is discarded.

Similarly, for situations where two or more groups want to associate into one, each group nominates a gatekeeper. The gatekeepers negotiates a new association and then bridges the association with their original groups.

Up to now, we have explained the basic operation of using co-location for associating devices. Nonetheless, there are several limitations as well as opportunities which GroupTap can offer. In the next section, we discuss the limitations and the advantages, which designers can contemplate when adopting GroupTap for device association.

DISCUSSION

Limitations

GroupTap requires the surrounding objects within the vicinity of the users to be NFC enabled. This requirement strictly limits the context where device association can be performed. For example, in an outdoor natural environment, the surroundings cannot support the specification that GroupTap requires. Designers cannot restrict users to associate devices only in places where smart objects are available. A simple solution to this is, during the association, the users can elect a mobile device to act as a smart artefact. The rest of the devices can then perform a single artefact GroupTap association using the elected device. However, to achieve this, the technical details of GroupTap need to be changed to include the elected device in the association, which we will address in our future work.

People's unfamiliarity with NFC technology also limits GroupTap's potential. Despite the simplicity of NFC touch-based interaction, many people are still unfamiliar with this kind of physical interaction [3]. Although some commercial applications have applied NFC in their systems (such as micropayment, token-based identification, access control, etc.), the use of touch-based technology for associating devices is uncommon.

Other than people's unfamiliarity with the NFC interaction,

[4] The $terminate$ message between the artefacts is sent via the artefacts' back-end Internet connection.

[5] $h(...)$ denotes a hash function and $|$ denotes a concatenation.

affordance also plays an important role. For example, the NFC tag location on smart objects may not be visually obvious. For small objects, users can assume the entire object is NFC enabled; however, when the objects have large surfaces, like a refrigerator, users cannot randomly pick a spot on the object and assume it is NFC enabled. Instead, the users need visual cues to give them a perception of on where they should tap their devices.

For multiple objects group association, the system assumes that artefacts are connected in the back-end, such as to the Internet. However, many objects may not have such facility; hence, in step 5, the artefacts cannot automatically forward the $terminate$ message to one another. In situations where between artefacts communication is missing, we can adjust the user interaction. So, instead of the leader tapping only on the first artefact, the leader must tap on all of the involved artefacts to send them the $terminate$ message.

Advantages

To understand the opportunities that GroupTap brings, we use the following use-case scenario to convey our idea.

During a meeting, attendees may want to associate their mobile devices (e.g., tablet computers) to share files. To enabled an experience of transient file sharing, the attendees can use a projector in the meeting room as a smart artefact, so they can tap their devices on the projector to quickly form a group association and share the files. Before the shared files are accessed, the company may have a strict policy that confidential information can only be accessed within the company's premises. Hence, the users need to authentication their location before they can open the files. Since the users already tapped on the projector, the projector can provide its location context information; with this, the users' devices can inform the system that they are within the premises. An idea similar to Seifert et al.'s TreasurePhone concept [13].

Beside the advantage of forming an association, using co-location also provides services based on the location context of the associating devices. From the example above, a system implicitly uses the context information to offer enhance services such as security, without the users explicitly inputting the context information.

CONCLUSION

We have presented GroupTap, an intuitive and secure device association scheme for a group of devices. Our work exploits the use of spatial co-location of mobile devices with connected objects to achieve secure device association. We provided the design space of using location references for device association, the technical details of the association procedure, as well as a discussion on the limitations and advantages of GroupTap.

REFERENCES

1. D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. NDSS 2002*, 2002.
2. M. K. Chong, G. Marsden, and H. Gellersen. GesturePIN: using discrete gestures for associating mobile devices. In *Proc. MobileHCI 2010*, pages 261–264. ACM Press, 2010.
3. A. Hang, G. Broll, and A. Wiethoff. Visual design of physical user interfaces for nfc-based mobile interaction. In *Proc. DIS2010*, pages 292–301. ACM Press, 2010.
4. K. Hinckley. Synchronous gestures for multiple persons and computers. In *Proc. UIST 2003*, pages 149–158. ACM Press, 2003.
5. K. Hinckley, G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith. Stitching: pen gestures that span multiple displays. In *Proc. AVI 2004*, pages 23–31. ACM Press, 2004.
6. L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. Smart-Its Friends: A technique for users to easily establish connections between smart artefacts. In *Proc. UbiComp 2001*, pages 116–122. Springer-Verlag, 2001.
7. F. Kawsar, K. Fujinami, and T. Nakajima. A lightweight indoor location model for sentient artefacts using sentient artefacts. In *Proc. SAC 2007*, pages 1624–1631. ACM Press, 2007.
8. T. Kindberg and K. Zhang. Validating and securing spontaneous associations between wireless devices. In *Proc. ISC 2003*, pages 44–53, 2003.
9. D. G. Park, J. K. Kim, J. B. Sung, J. H. Hwang, C. H. Hyung, and S. W. Kang. TAP: Touch-And-Play. In *Proc. CHI 2006*, pages 677–680. ACM Press, 2006.
10. J. Rekimoto. SyncTap: synchronous user operation for spontaneous network connection. *Pers. Ubiquit. Comput.*, 8(2):126–134, 2004.
11. J. Rekimoto, Y. Ayatsuka, M. Kohno, and H. Oba. Proximal interactions: A direct manipulation technique for wireless networking. In *Proc. INTERACT 2003*, pages 511–518. IOS Press, 2003.
12. K. Seewoonauth, E. Rukzio, R. Hardy, and P. Holleis. Touch & connect and touch & select: interacting with a computer by touching it with a mobile phone. In *Proc. MobileHCI 2009*, pages 36:1–36:9. ACM, 2009.
13. J. Seifert, A. De Luca, B. Conradi, and H. Hussmann. Treasurephone: Context-sensitive user data protection on mobile phones. In *Proc. Pervasive 2010*, pages 130–137. Springer-Verlag, 2010.
14. F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proc. 7th International Workshop on Security Protocols*, pages 172–194. Springer-Verlag, 1999.
15. J. Valkonen, N. Asokan, and K. Nyberg. Ad hoc security associations for groups. In *Proc. ESAS 2006*, pages 150–164. Springer-Verlag, 2006.