Point&Connect: Intention-based Device Pairing for Mobile Phone Users

Chunyi Peng[1], Guobin Shen[1], Yongguang Zhang[1], Songwu Lu[2]
[1]Microsoft Research Asia, Beijing, 100190, China
[2]Univ. of California, Los Angeles, CA 90095, USA
{chunyip,jackysh,ygz}@microsoft.com, slu@cs.ucla.edu

Abstract

Point&Connect (P&C) offers an intuitive and resilient device pairing solution on standard mobile phones. Its operation follows the simple sequence of *point*-and-*connect*: when a user plans to pair her mobile phone with another device nearby, she makes a simple hand gesture that points her phone towards the intended target. The system will capture the user's gesture, understand the target selection intention, and complete the device pairing. P&C is intention-based, intuitive, and reduces user efforts in device pairing. The main technical challenge is to come up with a simple system technique to effectively capture and understand the user intention, and pick the right device among many others nearby. It should further work on any mobile phones or small devices without relying on infrastructure or special hardware. P&C meets this challenge with a novel collaborative scheme to measure maximum distance change based on acoustic signals. Using only a speaker and a microphone, P&C can be implemented solely in user-level software and work on COTS phones. P&C adds additional mechanisms to improve resiliency against imperfect user actions, acoustic disturbance, and even certain malicious attacks. We have implemented P&C in Windows Mobile phones and conducted extensive experimental evaluation, and showed that it is a cool and effective way to do device pairing.

1 Introduction

In recent years, mobile phones are becoming increasingly popular. This has led to many new applications such as file swaping, music sharing, and collaborative gaming, where nearby users engage in spontaneous wireless data communications through their mobile phone Bluetooth or WiFi interfaces. A prerequisite for such in-situ device-to-device connectivity is *device pairing* – the first-time introduction and association between two devices, often without prior context. Such a connection must be set up before the two phones can engage in an interaction like the above applications.

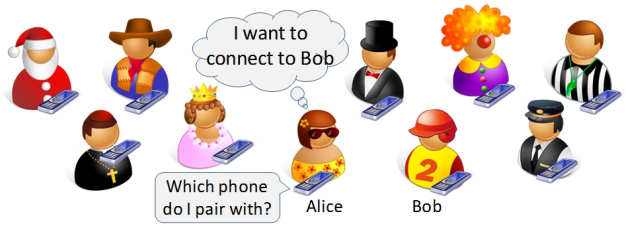

Figure 1. Motivating scenario

In this research, we focus on device pairing in a multi-party setting – when there are many mobile devices within the communication range, a user who initiates device pairing must first identify the intended target and convey her selection to the initiating phone in her hand. Figure 1 illustrates a motivating example. Alice has made a new friend Bob and wants to send some pictures from her phone to his. To pair her phone with his, Alice needs a mechanism to let her phone know which nearby phone is the intended target. This involves bridging a "perception gap" in device pairing, where a user knows clearly in her mind what is the intended target (e.g., the mobile phone in Bob's hand), but she must translate it into a piece of identification information understood at the device level. Currently there are many ways to achieve this. For example, Bluetooth adopts a "scan-and-select" model, where the initiating device scans the wireless channel and lists a set of nearby devices for the user to pick her selection. Other research proposals rely on both parties sharing some private information [10, 17, 25] or taking some synchronized actions together [11–13, 15]. All of them require various degrees of user involvements and system efforts.

In this paper, we seek to further minimize this perception gap with a new intention-based device pairing paradigm. It is based on the ability for a user to express, and the system to capture, an intention of device selection via a simple pointing action. The proposed solution, *Point & Connect* (or P&C for short), works as follows. When a user wants to connect her phone to another device, she can express her intention by simply *pointing* her mobile phone towards the intended target device (see the example in Figure 2). The mobile phone software can capture this intention and select the right device to complete the pairing operation.

The fundamental challenge here is how to effectively capture and understand the intention of such user pointing action. That is, among the many nearby devices, the system must be able to identify the one along the moving direction

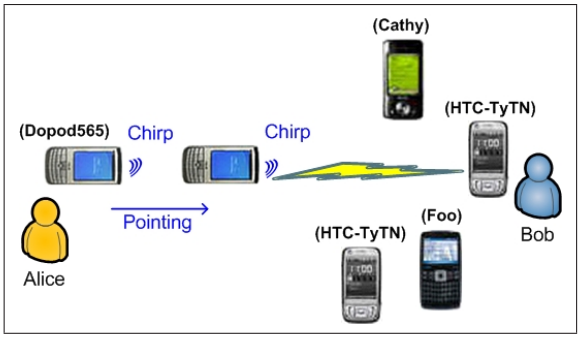

Figure 2. P&C example: Alice pairs her mobile phone with Bob's

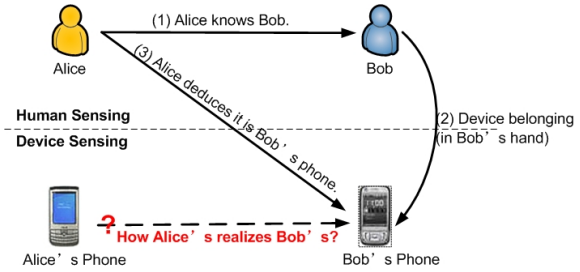

Figure 3. Perception gap between user and device space in device pairing

of the user's mobile phone. It is a non-trivial problem to establish the positioning relationship among devices in the 3-D physical world. Approaches such as visual recognition [18] or applying motion tracking [14] either have too much overhead or require an infrastructure which is not available for in-situ use.

We have met this challenge with a novel acoustic-based distance change detection technique. When a user moves her phone towards the target device, the system requires each "candidate" nearby to observe and report the relative distance change. To assist the measurement, the selecting phone will emit two sequential "chirp" sound signals during the pointing action and each candidate will independently compute the elapsed time of arrival (ETOA) of the two chirps heard. P&C subsequently exploits the fact that the candidate along the point direction should report the largest relative distance change. In addition, P&C devises several techniques to enhance its robustness against various dimensions of factors, including user operational uncertainty and ambient noise, and to defend against a few common malicious attacks.

Equally important, P&C is easy to implement on mobile phones. The above novel mechanism operates on a minimum set of hardware capability – the built-in speaker and microphone. This commodity-based solution will obviously have wider applications and cost less, because speaker-and-mic is indeed quite often a common denominator of many mobile devices, including mobile phones, PDAs, media players, etc. In addition, P&C can be implemented solely in software and in user-space, making it easy to adopt and deploy.

The rest of this paper is organized as follows. Section 2 identifies the challenges and reviews the solution space. Section 3 provides the P&C system overview. Sections 4 and 5 present the intention-based design and sensing techniques, respectively. Section 6 describes techniques to enhance resilience against uncertainty and malicious attacks. Section 7 presents the prototype implementation and Section 8 evaluates the performance. Sections 9 and 10 discuss additional issues and compare with the related work, respectively. Section 11 concludes the paper.

2 Device Pairing Models

2.1 Design Challenges

Device pairing typically involves two steps, target identification and secure communication channel setup. The second step is well studied and there are many successful approaches, but most device pairing solutions ignore the target identification issue and leave it for users to handle manually.

The main challenge in target identification is how to bridge the perception gap between how human identifies a nearby device in mind and how devices must identify themselves among one another, as illustrated in Figure 3. On one side, it is natural for a person to identify a surrounding device by its relative position. Using the same motivating example, Alice knows which device she wants to talk to and where it is – the mobile phone in Bob's hand. On the other side, device pairing requires device-level identification information, such as device name, MAC address, or some other alias, which may easily confuse the user since many of them do not use an intuitive, easy-to-identify naming convention. To bridge this gap under current device pairing models, one must translate the entity in her mind to the device-level identification. In the Bluetooth example, Alice would have to ask Bob for his phone's device name and look it up in the list returned by device discovery.

Ideally, it is desirable to have a device pairing paradigm without such a perception gap. That is, a user needs only to convey her selection target in mind, perhaps with some form of natural action, and the system can capture the user intention and automatically identify the right pairing target. The technical challenge for this vision is to find an intuitive and yet effective way to express an intention of selecting a device, and at the same time to ensure an efficient system can be built to capture and understand such an intention. The engineering challenge is to build such a system that works under current hardware software constraints, preferable on COTS mobile phones and with only user-level software, and yet resilient against imperfect user actions, ambient disturbances, or even some malicious attacks.

2.2 Other Approaches

Many existing device pairing solutions adopt a model similar to Bluetooth's "scan-and-select". To bridge the perception gap, a user will need to manually map the intended target into some form of IDs, device names, or network addresses.

If two mobile phones share some private knowledge, this can act as the basis for device selection. Many prior work such as Seeing-Is-Believing [17], blinking-LED [25], Loud-and-Clear [10], and others [28], etc., can be used here to assist device selection.

Two users interesting in device pairing can use explicit synchronized actions as the basis for device selection and subsequent pairing [11–13, 15, 22]. For example, two users can take synchronous actions of button presses and releases

on both phones [22], or place their phones together and move in same trajectory [12]. These approaches usually require both users agree upon a common action sequent, and some further require mobile phones to be equipped with special sensors such as motion sensors use in [12].

P&C belongs to yet another category in the device pairing solution space. The selection intention is expressed through hand gestures which are simple, intuitive, and very natural.

3 P&C Design Overview

P&C offers a pure software-based solution to device pairing, and works with a standard phone hardware setting of a microphone, a speaker, and a wireless interface such as Bluetooth and/or WiFi. It does not rely on any infrastructure support or any operating system (OS) modifications. The wireless connection, thus established between the device pair, works in an ad-hoc, peer-to-peer fashion.

3.1 Design Guidelines

P&C should explore an intention-based design paradigm taking the action sequence of point-and-connect. A user will express her intention via actions, and user action and device software will work together to overcome the hardware limitation of a COTS phone. Specifically, the function of target device identification will use a simple human gesture "pointing".

P&C should further provide a verification function that exploits human perception to validate the selection result. For example, human can easily see a blinking screen or hear a chosen sound such as "horn" on the selected device. These natural human inputs can greatly simplify the design complexity on the device.

P&C should also seek to balance operation simplicity and system resilience. It is resilient under a variety of imperfect operating environments. It functions well in the presence of environmental noise, imperfect human gesture, and multipath fading over the acoustic frequency band. It can also defend against a few malicious attacks. However, we seek to reduce the design complexity whenever possible. To this end, we should only use a few fairly simple techniques to enhance resilience, including leveraging human perception feedback, updating blacklists for attackers, and adaptive backoff. For example, in the scenario of Figure 2, if Alice erroneously points to a wrong target, say, Cathy's phone, then Cathy's phone will blink its screen or emit a chosen, alarm sound once selected. Alice can simply redo the pointing. Nevertheless, we do not make it a goal to offer maximum degree of security against the largest set of possible attacks, because doing so will increase design complexity and reduce its usability and simplicity.

3.2 Application Scenarios

The application scenario for P&C is fairly generic. It is applicable to simple file sharing, picture swapping, instant messaging. It is also applicable to social networking applications via phones. The typical setting for a P&C working scenarios is that the pair of devices, which intend to interconnect, are located within a short distance. Therefore, line-of-sight communication over the acoustic channel is possible. The devices do not need multi-hop wireless communication for data exchange. Besides the scenario illustrated in Figure 2, P&C also works when Alice selects a device out of many devices that are *statically* placed (which may not be actively carried by users). The setting is a showcase of scenarios at home, conference room, classroom, etc., where the user intends to pair the device at her hand with another devices already permanent in the location.

P&C operates when the relative positioning of devices does not change much. It still works when the devices are moving together, such as all of them are in a moving bus or train. It does not focus much on the device mobility case, where a few devices are highly mobile relative to other devices. P&C targets a single pair of devices that initiate interconnection at a given time. Its base design does not support multi-pair, simultaneous pairing in a given application scenario. In such a case, other pairs of devices can simply wait until the current pair finishes their connection process. Anyway, the process typically lasts for a few seconds.

4 Intention-based Device Pairing

Our intention-based device-pairing paradigm employs two main mechanisms. First, pointing gesture is used as a way to express user intention in target selection. This nameless pairing bridges the gap between user perception and device naming. Second, the detection of the intended target is achieved through a collaborative distance change measurement method. In this section, we discuss both mechanisms in details.

4.1 Intentional Selection via Pointing Gesture

The first issue is how to let a user express, and make the system capture, her intention to select a given target device. The solution has to be simple and intuitive, in order to work well with an ordinary user.

Our solution is to let the user take a simple gesture of "pointing" the device to the target phone. The user holds the cell phone in hand, and performs a "pointing" gesture that moves her arm, as well as her phone in hand, straightly towards the target device. The rationale for this design comes from natural human behavior in daily life – it is very natural for users to identify and select objects via pointing-style hand motions. This philosophy is also supported by a recent user-behavior study on mobile phone interaction in the literature [23]. The study shows that, the most common user-object interactions are the actions of *touching, pointing, and scanning* when performing user-mediated object selection and indirect remote controls. When the user and the object have a line of sight but still are separated by a distance, pointing is the preferred choice.

The above solution addresses the issue of the perception gap between the user and the device. The users do not need to know any device naming. The target device is implicitly identified with its current physical location, and the selection is expressed through the pointing action. In a sense, P&C offers a nameless device-pairing scheme since the semantic name of the device, e.g., MAC address, IP address, or alias, is not needed in the selection process.

We believe our solution further reduces the user efforts needed in device pairing, compared with various other previous approaches described earlier. For example, we do not require precise synchronized actions on both users. Further,

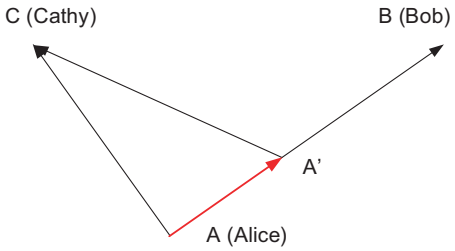

Figure 4. Principle of the P&C detection mechanism. As Alice points her phone towards Bob's, Bob sees a larger distance reduction than Cathy, as ensured by the triangular inequality: $d_{AB} - d_{A'B} = d_{AA'} > d_{AC} - d_{A'C}$.

our solution does not require motion sensors or other special hardware.

4.2 Detection via Maximum Distance Change

To capture the user intention of a pointing action in a simple and effective manner, P&C devises a simple detection technique based on relative positioning. The main idea is based on a geometric, triangle inequality regarding the relative distance change when the pointing action is taken. The example of Figure 4 best illustrates how the detection works. In the example, Alice selects Bob's mobile phone at position B, and Cathy's phone is also located at a neighborhood position C. To this end, Alice moves her phone straightly towards Bob's from position A to position A'. After this pointing gesture is completed, Bob's and Cathy's phones see distance changes equal to $d_{AB} - d_{A'B} = d_{AA'}$ and $d_{AC} - d_{A'C}$, respectively. Regarding these two distance changes, the triangle inequality states that, $d_{AA'} > d_{AC} - d_{A'C}$. That is, Bob's phone (the intended target device being pointed to) will see the maximum distance reduction before and after the pointing action.

In summary, P&C correctly selects the pairing target by measuring and comparing the distance change at each candidate device before and after the pointing action. Among all the candidates, the device being pointed to will observe the maximum distance reduction before and after the gesture, as ensured by the well-known triangular inequality. Therefore, P&C infers the device orientation associated with the pointing action through detecting the relative distance change at each candidate device.

The P&C detection mechanism also exhibits benefits compared with other design alternatives. The popular approach to detecting the pointing action is to use visual markers such as QR code [2], Semacode [3], or image processing [27] with the assistance of camera. This approach needs cameras and requires the detected objects be tagged with visual markers in advance. It is computationally intensive, thus not suitable for mobile phones and other COTS embedded devices. Moreover, the practical constraints in placing cameras and markers also limit the operating scope. Techniques using laser or infrared pointers can detect a direct pointing action with lesser degree of difficulty. However, such techniques require that objects be placed within a short range and need special hardware such as a light sensor or an IrDA, which is not available on most mobile phones now.

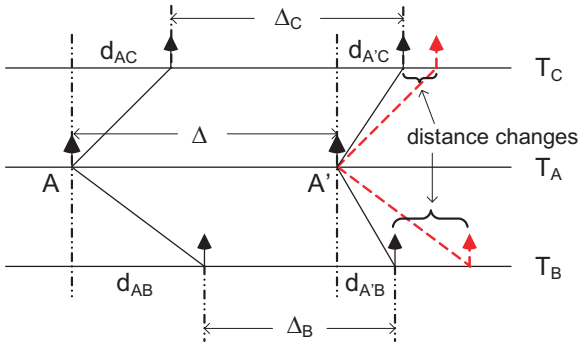

Figure 5. Illustration of the translation from maximum distance reduction to minimum interval between the two sound signals.

5 Acoustic Sensing in P&C

To make the intention-based scheme work, we need to devise a sensing solution that accurately detects the distance change at the target, which is surrounded by many devices. The solution has to work with the standard hardware setting of a COTS mobile phone. To this end, we need to address three technical issues in P&C sensing: (1) How to achieve high accuracy relative distance change measurement without any pre-deployed localization/ranging infrastructure? (2) How to perform scalable measurement such that the measurement overhead remains constant as the number of candidate devices increases? (3) How to find/notify the intended target, out of many candidate devices, with as little communication overhead as possible, without any *a priori* one-one communication established with the target?

5.1 Detecting Distance Change via Acoustic Sensing

To detect the relative distance changes, several solution alternatives are available but all have downsides. A simple way is to detect the relative difference of receive signal strength, say, RSSI values, at different receiving devices. However, two factors severely degrade the accuracy of this RSSI-based scheme. First, RSSI values fluctuate a lot (e.g., $3-8$ dB according to previous measurements) even at a fixed receiver location. Second, the distance difference from the sender to each receiving device also makes the distance base for comparisons different. Another way to obtain such distance changes is to derive them from the sequential, pairwise ranging measurements taken before and after the pointing action. However, since pairwise ranging is performed between the phone and each candidate target, it does not scale: To select one out of N devices, 2N measurement should be performed and the overall process would be extremely long in time and thus more vulnerable to ranging errors. The accuracy is impaired as the error bound is doubled because of the distance substraction operation.

We propose a novel solution that eliminates the scaling issue incurred by the sequential scheme. We let the selecting device emit two sound signals before and after the pointing action, and ask all the candidates to record and detect the two signals and report the interval in between, then we can translate maximum distance reduction to the minimum measured interval and select the winner accordingly.

Figure 5 illustrates such a translation. Because all the devices are located in a close proximity, the propagation speed of sound signal to all candidates will be the same. The distance is therefore strictly proportional to the time duration the sound signal propagates, as depicted by the slanted lines in the figure. If the selecting device does not move, then the second sound signal will take the same time as the first signal to reach the candidate devices as indicated by the two dotted slanted lines. In this case, the measured intervals at candidate devices, Δ_C and Δ_B, will be equal to Δ – the interval between the two sound signals at the selecting device. As the selecting device moves closer to the candidates after the pointing action, it takes less time for the second sound signal to reach them. As seen in Figure 5, the distance reduction at B can be expressed by

$$\Delta_{d_B} = d_{AB} - d_{A'B} = v \cdot (\Delta - \Delta_B), \quad (1)$$

where v is the sound speed and is also roughly the same in close proximity. The distance change at C, smaller than that at B, can be measured similarly. Therefore, the maximum distance reduction can be translated to the minimum detected interval.

Therefore, the problem (1) of high accurate relative distance measurement is converted to obtaining the accurate, elapsed time of arrivals (ETOA) of two sound signals in the presence of inaccurate system clock. The traditional way of taking a timestamp of their respective local clock at the moment the signal is received to calculate the duration may meet with intrinsic receiving uncertainties due to the lack of real-time control, software delay, interrupt handling delay, system loads in a real system. Therefore, we use the sampling counting scheme proposed in [19] that addresses the problem. The solution leverages the fact the independent A/D convertor works at a fixed and stable sampling frequency, thus generating high-accuracy time information without unknown delay factors from the phone's operating system. In P&C, the target device can be determined through comparing the sample numbers between two sound arrivals among multiple devices. the device with the minimal number of samples between two received signals is the one along (or closest to) the movement direction.

Effectively, our solution simultaneously estimates the pointing phone's distance changes to all candidates. No matter how many candidate targets are around, only two signals are required. These two chirps will be used by multiple candidates to estimate the distance changes in parallel. The overall signaling overhead is independent of the number of candidate devices, and our sensing scheme yields an accurate device identification, scalable to the number of candidates.

5.2 Exchanging ETOA Measurements Between Devices

The above sensing scheme enables a scalable solution to detecting the relative distance change at each candidate device when the pointing action is taken. The next issue is to find and notify the device that detects the maximum distance change, i.e., the intended target. The goal is to exchange ETOA measurements between the selecting device and the target phone, as well as other nearby devices. Our solution exploits the radio interface on COTS phones, such as Bluetooth or even Wi-Fi.

We use a backoff-based mechanism to detect and notify the target device, which measures a minimum ETOA interval. Each device sets a backoff timer, in proportion to its ETOA interval measurement, when exchanging such ETOA information with each other. The backoff timer is implemented at the link buffer, thus different from the low-level MAC backoff mechanism. The timer granularity is also coarser, say, tens or hundreds of milliseconds for each backoff unit in our implementation. The backoff scheme effectively provides a fully distributed mechanism to prioritize the reporting process based on the increasing order of ETOA measurements. Using the backoff timer, the one which has minimum ETOA will send its message first, and others will defer when hearing such an earliest message. To this end, a Bluetooth or Wi-Fi channel is used as the broadcast channel to exchange the ETOA measurement. The target device, having the minimum ETOA interval, gets its backoff timer expired first, and sends its ETOA interval, together with its MAC address, over the broadcast channel. This way, the selecting device can quickly find the target phone by receiving the minimum ETOA message. Upon receiving the expected ETOA, it subsequently sends an "acknowledgment" message to notify the device. Upon receiving this acknowledgment message, other devices cancel their backoff timers and stop sending their ETOA values. The selected device then blinks its screen when receiving the acknowledgment, so that the user can also verify its selection via visual observation. A one-to-one data communication channel using Bluetooth or Wi-Fi can then be set up between the pairing devices.

Note that our above solution also scales to the number of candidate devices. The selection process does not incur additional ETOA message overhead, in that the number of transmitted ETOA messages does not grow with the device population.

6 Enhancing Resilience

In this section, we first identify a few factors that may affect the resilience of P&C, and then discuss the proposed solution techniques. The goal is to ensure that P&C is both robust against various uncertainty factors and secure against a few common, malicious attacks.

There are various dimensions of uncertainty in the pointing action, device, ambient disturbances, as well as malicious attacks. These factors can be roughly classified into the following three categories:

- Imperfect user action: various gesture uncertainties or operation errors may exist.
- Malicious attacks: practical attacks may occur, such as denial-of-service (DoS) attacks, replay attacks, and forged messages over wireless broadcast.
- Acoustic sensing disturbances: various error sources exist, including multipath propagation and fading, ambient noise, etc.

P&C uses four techniques to handle human errors and defend against attacks. The first one is to rely on user feedback. The second is to maintain a black-list for identified attackers. The third is an adaptive backoff and acknowledgment mechanism, which refines the mechanism of Section 5.2 and offers the selecting device a method to control how

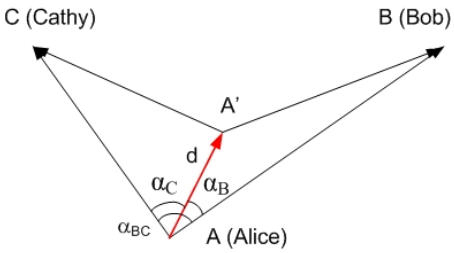

Figure 6. Robustness against pointing orientation error

many ETOAs it wants to receive by controlling when to send back the acknowledgment. It is useful to defend against DoS attacks. The last technique is the hash chain mechanism, which helps to ensure message authentication over wireless broadcast. We now describe how P&C handles the above factors in details.

6.1 Handling Imperfect User Action

The first type of errors is related to the human gesture uncertainty when using P&C. There are three common errors/mistakes a user makes during the pointing action: (1) pointing to the wrong target: Instead of pointing to the target device, the user mistakenly points to other objects or devices located in the neighborhood. (2) non-direct pointing action: The pointing trajectory by the user is not a straight line towards the target. Rather, it is a curve or other arbitrary shape. This reduces the effective distance change to the target and may increase the distance change detected at other nearby devices; (3) short pointing: The pointing gesture follows a straight line to the target, but travels too short a distance. This can also compromise the detection accuracy in P&C.

Our solution to the above human gesture uncertainty still takes the user-centric approach. We rely on user perception feedback to resolve the issues. Once selected, the device blinks its screen for a short period of time or alarms an alert sound. By observing which device's screen blinks or alarms the alert right after the pointing action, the user can verify whether she has selected the correct target. If not, the user simply re-takes the pointing action to her intended target until she sees the right device blinking. Therefore, the user has an independent source to reliably verify whether the selection is correct or not and correct her mistakes.

Robustness against pointing orientation error We now show that P&C itself is fairly robust against the pointing direction errors. Therefore, the above technique is not invoked in normal cases.

We use the example scenario of Figure 6 to show that P&C is still able to identify the correct target B even when the pointing error α_B is nonzero but not too large, say, less than 45^o. By applying Cosine Theorem, the distance reduction at B is:

$$\Delta_{d_B} = d_{AB} - d_{A'B} = b - \sqrt{b^2 + d^2 - 2bd \cdot \cos(\alpha_B)}, \quad (2)$$

where b and d stand for d_{AB} and $d_{AA'}$. The above function is monotonically decreasing with respect to $\alpha_B \in [0, \pi]$, and reaches the maximum peak at $\alpha_B = 0$, independent of b. In particular, when $b \gg d$ and α_B is sufficiently small (the typical case in reality), the reduced moving distance Δ_{d_B} can be

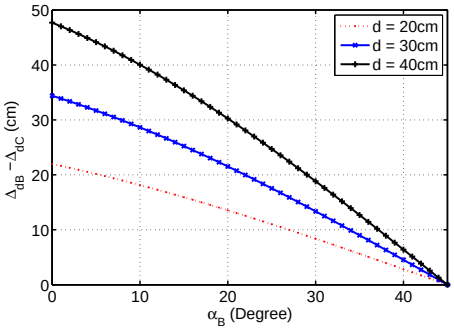

Figure 7. Differential distance change in response to a_B

approximated by

$$\Delta_{d_B} \approx d \cdot \cos(\alpha_B). \quad (3)$$

Hence, the distance reduction approximates the projection of the "pointing" along the original direction to the listener phone.

We now assess the impact of an imperfect pointing angle α_B (i.e., $\alpha_B \neq 0$) upon the distance reduction in the example scenario, where B and C are fixed at $d_{AB} = d_{AC} = 100cm$ and $\alpha_{BC} = 90^o$[1]. Figure 7 plots the difference of distance reduction $\Delta_{d_B} - \Delta_{d_C}$ under different pointing movement 20cm, 30cm and 40cm. The figure shows that, in realistic cases where the pointing error (α_B is small but nonzero, say, less than half of α_{BC}) exists, the one with smallest α (i.e., the one closest to *"pointing"* direction) can still be correctly identified by comparing the distance reduction.

6.2 Defending Against Malicious Attacks

DoS attacks include forging a small ETOA value, generating a very long chirp signal to collide the second chirp sound emitted from the selecting device. The attacker can also replay obsolete messages from the past, forge messages on behalf on another well-behaving device, and perform a Man-in-the-Middle(MiTM) attack over the wireless channels.

Fake ETOA attack This is one type of denial of service (DoS) attacks against P&C. In the attack, the malicious attacker fabricates a smaller ETOA value than it should be, in order to cheat the user to never select the right target but choose the attacker's device. There are two specific attack patterns. Our solution is illustrated in Figure 8.

In the first pattern, the attacker launches his attack by announcing his fake ETOA message without changing his authentic ID, say, its MAC address. Then the solution has three components. First, P&C filters out any invalid ETOA value that is beyond the typical operation range. If the attacker without hearing two chirp signals sets an overly small value, it cannot pass the ETOA filter. This provides the first line of defense against forged ETOA attack. The operation range of ETOA can be obtained through conservative estimates on historical operations by users. Second, the selected target is asked to blink his screen or alarm an alert once selected. Then the selection failure of choosing a wrong device (the attacker's phone in this case) is quickly detected by the

[1] More cases of different d and α_{BC}, with different combinations of d_{AB} and d_{AC}, yield the same conclusion.

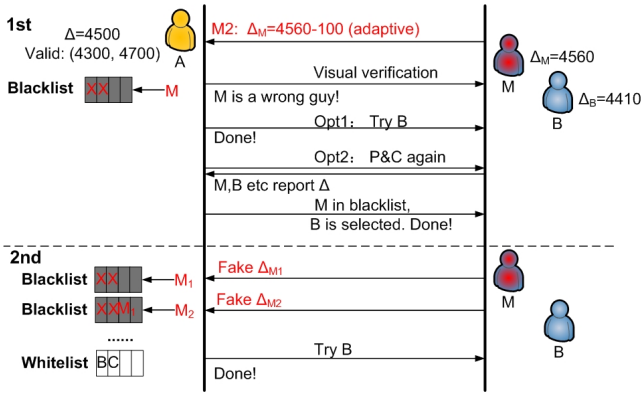

Figure 8. Defend against fake ETOA attack

user via her visual observation. Once detected, the user will re-take the P&C operations by pointing to the target device again. Third, if the attacker still fabricates a small ETOA message, P&C records the attacker device's ID and places it onto the *blacklist*. A device on the blacklist will be forbidden to be selected for a period of time, say, 5 minutes. When the user takes the P&C operation again at the second time, every candidate device will also use the backoff mechanism to report his ETOA message. This time, the selecting device will not send an acknowledgment upon receiving the first ETOA message, which may again come from the attacker. Instead, she hears two ETOA reports before replying with the acknowledgment message. Therefore, the backoff process will not terminate upon hearing a single report, but one more new message beyond the ETOA message heard during the first round that is possibly sent by the attacker. Since the attacker does not fabricate his ID, the selecting device easily ignores the wrong one and selects the right one that reports the second smallest ETOA value.

The above mechanism can also be extended to handle multiple attackers by using the solution in multiple rounds. In each round, the above mechanism will eliminate one attacker and place it onto the blacklist. The operations last for $M+1$ rounds by identifying the right target and recording the M attackers.

In the second attack pattern, the attacker can also forge his ID, such as an MAC address. Then the blacklist based approach, which records the attacker ID and consequently eliminates the attacker from selection, does not work. By using a newly forged ID in each round, the attacker pretends to be a new device and invalidates the blacklist history.

Our solution is to actively record the ETOA value of several devices in each round. The solution repeats the iterative process for multiple rounds. Ideally, $M+1$ values, also through $M+1$ rounds, need to be recorded at the selecting device by assuming M attackers. The rationale is that the right target device reports its true ETOA measurement, keeps on showing up on the $M+1$ reports and is not on the black-list. Then P&C uses its showup frequency to select it as the intended target.

Forged messages over wireless broadcast In this case, the attacker fabricates messages on behalf of other well-behaving nodes over wireless broadcast. In wireless broadcast, the attacker can pretend to be the authentic sender, and the receiver has no way to differentiate.

Our solution is based on the hash-chain message that ensures message authentication over the broadcast medium [8]. Each ACK message from the selecting device, which includes the ID of the selecting device, is embedded with an 8-bit or 16-bit message authentication code (MAC). This MAC is verified at each receiver regarding whether the sender is the authentic one or not. The attacker can thus not pretend to be the selecting device and fool the target device to initiate a blink-screen action. The operation of the hash-chain based message authentication is as follows. The authentic sender, i.e., the selecting device, generates a chain of encoded, one-way hash functions, say, $H_1, H_2, \ldots, H7, \ldots$. The generating rule is that $H_i = h_e(H_{i+1})$, where h_e is the encoded one-way hash function. Then the i-th ACK message, sent at time t_i, is embedded with a hashed value based on hash function H_i. The hash function H_i will not be released until t_{i+k} by the authentic sender, where k is a constant, say, $k=3$, which reflects the delayed factor for the hash release. The attacker thus cannot forge an ACK by pretending to be the authentic sender at time t_i. The receiving party verifies the MAC value at time t_{i+k}, so that the target device can differentiate the authentic selecting device from an attacker.

Forged Chirp attack This is another type of DoS attack. The attacker plays a long chirp sound once he hears the first chirp signal emitted from the selecting device. Using a forged chirp signal that lasts for an extended period of time, the attacker effectively disrupts the reception of the second chirp signal from the selecting device. Our solution relies on the user to partially defend against such an attack. The user can move closer to the target device when taking the pointing action. This can effectively increase the signal-to-interference ratio at the receiving side. There is no perfect solution to this type of physical-layer DoS attack.

Replay attack The attacker can also launch replay attacks by recording historical, authentic messages. The solution is a standard one based on timestamps and nounces. Each message also includes a random nounce to prevent the attacker from reusing obsolete messages.

MiTM attack The attacker can impersonate each endpoint to the other, making the victims believe that they are talking directly to each other when in fact they are talking with the attacker independently. A common solution to prevent MITM attacks is endpoint authentication. P&C can leverage existing device authentication solutions [10, 28]: it uses the acoustic channel to exchange both data and verification information among devices and involve users to verify the same audio source from the right device pair.

6.3 Handling Acoustic Sensing Disturbances

The ambient noise and multipath effect may distort or attenuate the acoustic chirp signals, thus significantly reducing ETOA detection accuracy if not handled well. P&C adopts two similar techniques in [19] to handle both background noise and multipath effects, and proposes an efficient joint detection.

To suppress the effect of ambient noise, the overall scheme takes a correlation-based detection approach. Each device records the received signal, which is correlated with

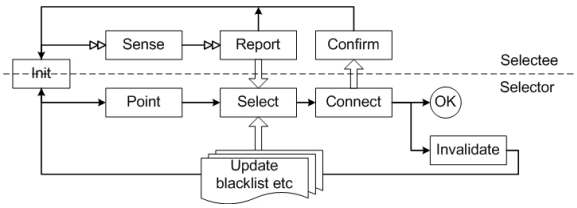

Figure 9. Block Diagram of P&C

the reference chirp signal stored at each device. The maximum peak is located if the cross-correlation of the signal and the reference is beyond a threshold ratio compared with the cross-correlation of the background noise and the reference. To this end, we compute the L2-norm of the cross-correlation value within a small window of samples around the peak; this reflects the energy level of the received signal. We also calculate the L2-norm of correlation values over a time window right before the peak, which reflects the noise level. When the energy level of the signal is twice or larger than the noise, we infer that the received chirp signal is located and ETOA is then calculated.

To handle multipath effect caused by reflection from a secondary path, we locate the first correlation peak rather than the maximal one. This is because the maximum peak may appear along the secondary path, which lags behind the signal on the primary path. In P&C, we locate the earliest, sharp peak in the time window of interest. The sharpness captures the level of the peak with respect to its surrounding side-lobes, and is computed by the ratio of the peak correlation and the average cross-correlation values in its vicinity.

The third technique is a joint ETOA detection scheme that can further improve the detection accuracy and also reduce the computation workload. It is based on the observation that P&C uses a pair of chirp signals in its sensing and these signal arrivals follow certain time pattern. We can then detect both arrivals in a joint fashion rather than by two independent detection. The elapsed time spans $[\Delta - t_{max}, \Delta + t_{max}]$, where t_{max} denotes the spread time for the maximal distance change. For example, the maximal distance change is about 1 meter, which corresponds to 130 samples in the typical setting of 44.1KHz sampling frequency. Consider several seconds of time interval between two chirp signals (about 44100 samples per second), joint TOA detection infers that the second signal should come in the window $[\Delta - t_{max}, \Delta + t_{max}]$ after the first chirp signal. It is thus unnecessary to perform correlation calculations for all recorded signals before that time. The saved computation can reach $1 - 2 * 130/44100 \approx 99\%$.

7 Implementation

P&C is implemented as a user-level software solution. It has eight function modules that work in concert in the current prototype. Figure 9 shows a block diagram of how they work during the P&C operations. The three modules *sense*, *report* and *confirm* above the slashed line are major functionalities for the selectee. The other four modules *point*, *select*, *connect* and *invalidate* are for the selecting device. The *Init* module is used by both.

The *init()* module pre-sets up the system when starting P&C. The *point()* module is executed when the user takes a "pointing" gesture. It is triggered by a simple activation button on the phone. When the button is pressed, the selecting device emits a chirp signal. The user holds the button until it travels certain distance toward the target device. When the pointing gesture completes, the button is released. Then the second chirp signal is emitted. The *sense()* module is executed at all other nearby devices, including the target one during a "pointing" action. It records the sound signals over the acoustic channel, and thus captures both chirp signals. Once each receiving device detects the two chirp signals, it stops recording. Then it calculates the elapsed time (measured in the sample counts) between the two chirp signals. The *report()* module enables each device to report its measured ETOA value. It then uses the backoff-based mechanism to report back over the radio broadcast channel. The *select()* module allows the selecting device to identify the target device by receiving the ETOA measurement from the intended target device. It also utilizes historical information like blacklist to make the final selection decision. It then sends an acknowledgment message over the broadcast channel. The *connect()* module sets up a private radio communication channel between the selecting device and the identified target. This procedure occurs after hearing the authenticated ACK message. The initiator connects to the target device using MAC address (Bluetooth) or joins the WiFi network with the target's SSID (WiFi). The *confirm()* module provides another independent source of feedback to the user, for example, it flashes the screen of the target device or vibrates itself to confirm the pairing. The *invalidate()* module is used when a wrong device is selected. It may occur when a wrong gesture is taken or an attack exists. When it is invoked, the module tears down the radio communication channel with the previous device that was set up during the last connect() operation and put it into the blacklist for the instant, here, at least for 5 min.

The above modules have been implemented on smartphones and PocketPC phones running Windows Mobile 5.0. We use multimedia services (`waveXxx` series APIs) embedded in Windows Mobile, to control microphones and speakers on the phones; and use the NDIS User Mode I/O interface (NDISUIO) and the Windows Sockets functions (`WSALookupServiceXxx`) and structures for wireless communication. The software runs as a user-mode dynamic linkable library that other applications can load and use as a service. The software is downloadable from our web site.

8 Evaluation

In this section, we use both experiments and numerical analysis to assess the performance of P&C.

We have evaluated our implementation on several smartphone platforms including Dopod 838 (i.e., HTC Wizard), HTC S620, HTC P3300, HTC Touch, HTC Tornado, HP iPAQ rw6828 and MWG Atom Life etc. In the experiments, we mainly use four phone models, Dopod 838, HTC Tornado, HP iPAQ rw6828 and MWG Atom Life. Other phone models produce similar results. In the rest of this section, we assess P&C correctness against different parameters in a laboratory setting that emulates indoor/outdoor environmental scenarios, examine resilience on imperfect pointing, report P&C energy consumption, and describe field trial experiences.

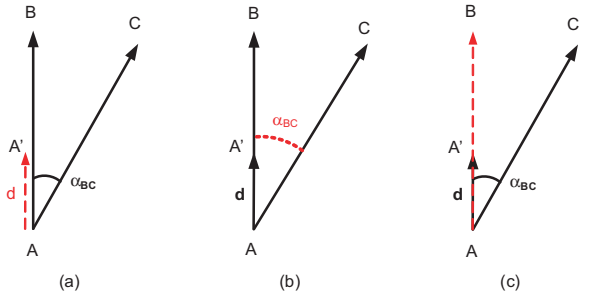

Figure 10. Experiment settings. We vary one factor at a time and keep the rest factors unchanged in each experiment. Broken and thick solid lines indicate the varying and fixed factors in each test case.

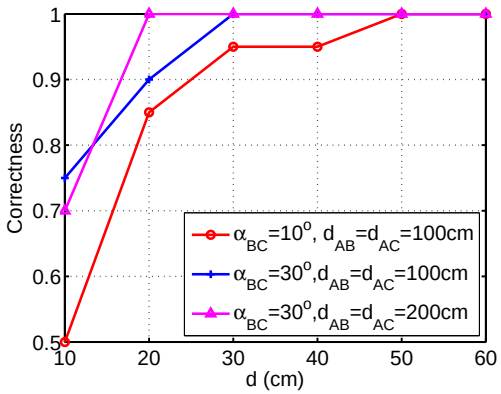

Figure 11. P&C correctness vs the displacement of the selecting phone in the pointing action

8.1 P&C Correctness

We first evaluate the correctness of P&C against factors, which include: displacement of the selecting phone (d) in the pointing action, target separation angle (α_{BC}), and distances between phones. We evaluate one factor at a time while fixing all the other factors. Test cases are depicted in Figure 10. Each experiment is repeated for 20 runs. Experiments are conducted under different environmental settings, such as quite indoor (conference room) and noisy outdoor (near subway station entrance) environments. The user always points directly to the target when using P&C.

Phone displacement d This set of experiments examine the impact of displacements d of the selecting phone on the correctness of P&C. We tested the performance under different combinations of α and the distance between phones, as shown in Figure 10-(a). Due to practical constraints (e.g., a person's arm length), we only evaluate the displacement distance up to 60 cm.

Figure 11 plots the test results. We can see that larger displacement leads to more reliable selection. The results suggest that in practice, we need to move the phone by at least 20cm to obtain accurate selection. Furthermore, when the candidate devices are separated by only a small angle (e.g., 10 degrees), a larger displacement is needed.

For the sake of practical interest, we set the displacement of selecting phone to 20cm and 30cm in the remaining experiments.

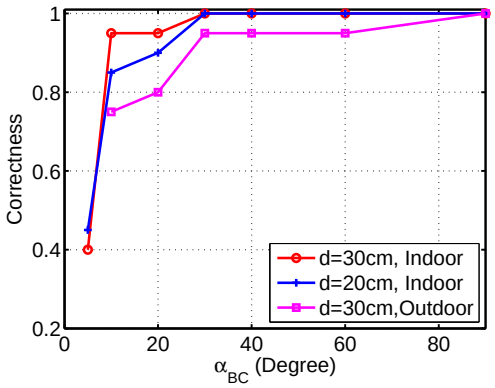

Figure 12. P&C correctness vs the target separation angle. In all cases, we fixed $d_{AB} = d_{AC} = 100cm$.

Target separation angle α_{BC} We evaluate the relation between the target separation angle α_{BC} and the selection correctness. In all cases, we have fixed the distance $d_{AB} = d_{AC} = 100cm$. The results are shown in Figure 12. As expected, the larger the separation between the target device and its closest neighbor, the more accurate the selection is. In most cases, 20cm displacement works fairly well, when the devices are separated by 30 degrees or more.

Note that, however, when α_{BC} is very small, the selecting result suffers and looks more random. This is easy to understand since when α_{BC} is small, it is indeed very challenging even for a person who takes a large-distance pointing gesture. In our setting, $\alpha_{BC} = 5^o$ implies the distance between the target and its nearest neighbor is only 9cm. The figure shows that, there is also a sudden increase in correctness when α_{BC} increases. This reflects the effect of the ETOA detection accuracy ε_{res}. When α_{BC} is very small, the distance changes at B and C are within ε_{res} and thus cannot be differentiated.

Distance d_{AB} This set of experiments evaluates the impact of the distance d_{AB} between the selecting phone and the target on the selection correctness. In all experiments, we fixed $\alpha_{BC} = 30^o$ and $d = 30cm$. In the first case, we fixed the distance $d_{AC} = 100cm$. In other cases, we let the distance vary while keeping $d_{AC} = d_{AB}$. The results are plotted in Figure 13.

We see that the selection is very accurate when the distance is less than three meters. However, when the distance goes even larger (e.g., four meters in indoor scenario and eight meters in outdoor settings), the selection correctness degrades. This is not consistent with the theoretical analysis to be presented in Section 8.5. In theory, d_{AB} should have little impact on selection correctness when $d_{AB} \gg d$. We examined the experimental traces, and found that this result is due to the drops in the ETOA detection accuracy, as caused by multipath effect and signal energy reduction for indoor and outdoor cases, respectively. Similar findings have also been reported in [19].

8.2 P&C Resilience

Tolerance on imperfect pointing angle α_B This set of experiments examines the effect of imperfect pointing angle

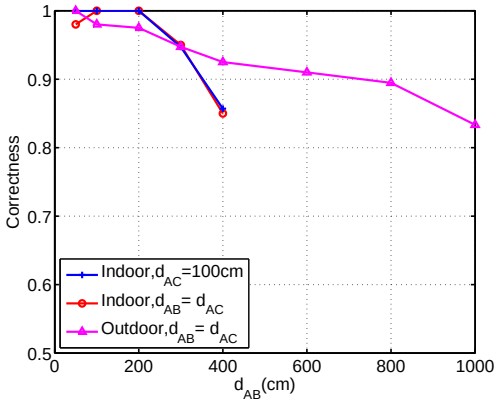

Figure 13. P&C correctness vs d_{AB}. In all cases, we fixed α_{BC} to 30 degrees, and $d = 30cm$.

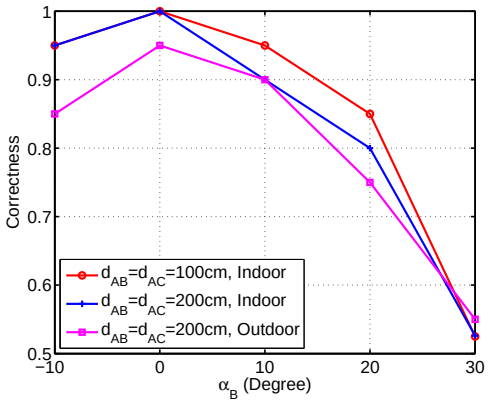

Figure 14. Resilience on pointing error. We fixed the parameters as $d_{AB} = d_{AC} = 100cm$, $\alpha_{BC} = 60^o$, $d = 30cm$.

α_B on P&C. In all experiments, we fixed the parameters as $\alpha_{BC} = 60^o$, and $d = 30cm$. The results are shown in Figure 14. Indeed, we see that in most cases, we still correctly select the target as long as the pointing direction is more leaned towards the right target. The result also confirms that the direct pointing action ($\alpha_B = 0$) yields best results. However, there is a sharp drop when the pointing angle increases from 20^o to 30^o, which is about half of α_{BC}. This confirms that the pointing error cannot be too big.

Response time against ETOA attack We also measure the time it takes P&C to respond to ETOA attack. In our experiments, we let one phone emulate the attacker who keeps on sending very small, forged ETOA value, while other phones are sending larger but valid ETOA measurements. The attacker does not forge its MAC address in the experiment. P&C indeed selects the wrong target during the first round, but correctly chooses the right one the second round. The time it takes to start the pointing gesture until selecting the right device is about 13.6 seconds in our experiment. This includes that the user retakes the pointing gesture, places the attack phone onto blacklist after seeing the wrong device blinking, and selects the right target finally.

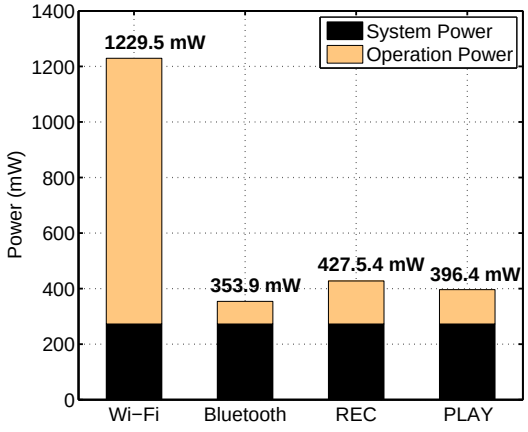

Figure 15. Power consumption of different tasks

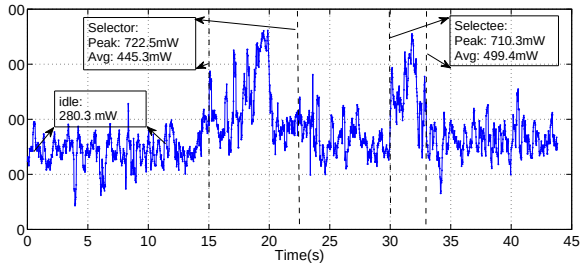

Figure 16. Power consumption of one P&C procedure

8.3 P&C Energy Consumption

We measure the power consumption of different functional tasks on HTC Tornado. Figure 15 plots the power consumption of different functional operations. The basic system, including backlight, CPU, and memory, consumes about 275mW. The power consumption for different tasks of Wi-Fi, Bluetooth, recording, and chirp playback is about 957mW, 81mW, 155mW, and 124mW, respectively. We can see that the acoustic sensing operation in P&C, which includes playback of two chirp signals at the selecting device side and recording at the receiving side, need use extra 150/120+mW, comparable to the phone system module. Therefore, a conservative estimate for P&C power consumption is about 430mW on the HTC Tornado phone. On the other hand, the Wi-Fi channel is power hungry, whereas the Bluetooth is much more efficient.

Figure 16 shows the power consumption of an entire round of P&C procedures over time. We did not include the power consumption of Wi-Fi or Bluetooth here, to highlight the energy used by main P&C. The initial $[0, 30]$ seconds are for operations at the selecting device, and $[30, 44]$ seconds are when the device is being selected. The procedure at the selecting device includes initial idle state (when P&C is invoked but no other operation is performed) during $[0, 15]$, and the pointing, selecting, idle operation afterwards. The target device starts its recording state for 1-2 seconds, followed by signal detection, ETOA measurements and report etc in $[30, 33]$. We can see that, compared with the basic system power consumption, a P&C in the idle mode consumes extra 5mW or so, a P&C as the selector consumes

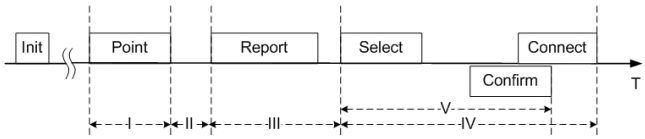

Figure 17. Illustration of time consumption for each step

	Mean	Std	Min	Max
I	1351.6	958	645	2430
II	366.3	16.1	315.5	393.4
III	2324.7	16.5	2298	2346
IV	2517.6	644	2048.4	3021.1
V	2026	1659.8	1207.4	4523.1
II+III+IV	5208.5	641.7	4679.4	6102.3

Table 1. Statistics of P&C time consumption by users

extra 170mW for about 6-8 seconds, leading to 1.2J used energy; it consumes additional 225mW for about 3 seconds at each candidate device at the target's proximity. Therefore, the power consumption by P&C modules is fairly modest.

8.4 Field Trial Experience

We also conducted some field trials using our prototype. We invited about 20 college students that have no device pairing experience before. They were divided into 10 pairs and asked to do Bluetooth pairing and P&C pairing. Before experiments, we briefed them for 2 minutes, and provided a printout of the detailed steps. We asked their feedback on our prototype, and recorded the timestamps for each major operation at the selecting device as illustrated in Figure 17. Each user ran 20 times using Dopod838, HTC Touch, HP iPAQ rw6828 and MWG Atom Life models. Since some components (e.g., the pointing gesture) change over different usage patterns, the results also vary.

The time spent on each major operation, averaged over all users, is shown in Table 1. We can see that, a user typically takes 1.4 seconds to complete the pointing gesture, 0.4 seconds to receive the ETOA report over the wireless channel, and 2.3 seconds to make the select decision after using the backoff mechanism. Once selected, the phone pair uses about 2.5 seconds to set up the one-to-one communication channel after seeing the screen-blinking confirmation. Each user takes an average 2 seconds to see the screen blinking and react by pressing a confirmation button in our experiment. Overall, it takes about 6.5 seconds to complete device pairing using P&C, while it takes about 57.2 seconds on average to use the Bluetooth pairing solution in our field trials. All the users indicated that our solution is very easy to use and intuitive, with almost no learning curve.

8.5 Numerical Assessment on Resolution

In this section, we provide a numerical assessment for the P&C resolution and learn practical ways to improve resolution, which refers to the minimum spatial requirement on neighboring devices so that they can be differentiated from the target device. Note that the P&C base scheme can be summarized as follows:

Target Device	
= the one with minimal α_B,	$\Leftarrow$ User intention
= the one with maximal Δ_{d_B},	$\Rightarrow$ Device iden.
= the one with minimal ETOA,	$\Rightarrow$ Sensing

In the example of Figure 6, the resolution defines the condition so that B can be differentiated from C. Therefore, the resolution requirement translates to:

$$\Delta_{d_B} - \Delta_{d_C} > d_{res}, \tag{4}$$

where d_{res} denotes the distance reduction measurement resolution. Since we adopted the sampling counting method to measure distance change, d_{res} is thus determined by the ETOA detection accuracy ε_{res}, i.e.,

$$d_{res} = \varepsilon_{res}/f_s * v \tag{5}$$

where f_s is the sampling frequency and v is the sound speed. The minimal d_{res} is achieved at only one sample resolution, and it is roughly $d_{res} = 1/44100 * 34600 \approx 0.7cm$ in a typical setting of $f_s = 44.1KHz$ and $v = 346m/s$. Moreover, BeepBeep [19] has shown that the accuracy d_{res} within 2cm is achievable.

In principle, node C can be differentiated from A when $\Delta_{d_B} - \Delta_{d_C} \geq d_{res}$. While the closed-form derivation can be quite involved, we start with a few realistic cases:

Case 1: Perfect pointing action if $\alpha_B = 0$. In this case, Equation (4) can be simplified as

$$d - \left(c - \sqrt{d^2 + c^2 - 2dc \cdot \cos(\alpha_C)} \right) \geq d_{res}$$

If $c \gg d$, we have

$$\cos(\alpha_C) \leq 1 - \frac{d_{res}}{d} \tag{6}$$

Case 2: The device is far away when $b \gg d, c \gg d$. The reduced distances can then be approximated by $\Delta_{d_B} \approx d \cdot \cos(\alpha_B)$, and $\Delta_{d_C} \approx d \cdot \cos(\alpha_C)$. Hence, we have

$$\cos(\alpha_C) \leq \cos(\alpha_B) - \frac{d_{res}}{d} \tag{7}$$

In the setting when perfect pointing to B with $d = 30cm$ and $d_{res} = 0.7cm$ (i.e., 1 sample), only node C with $\alpha_C \subseteq (-10.7^o, 10.7^o)$ can not be differentiated, which corresponds to about 40cm between B and C that are 2m away from A. In this case, B and C are very close, not a common scenario in practice.

We further plot Figure 18, to show the angle resolution within which close-by nodes cannot be differentiated from the target for four cases[2]. The four settings are given in Table 2, where we use Case 1 as the main reference for comparisons. From Figure 18, we make several observations on how these factors affect the identification accuracy.

First, the non-distinguishable angle scope increases as $|\alpha_B|$ increases, shown in all cases. It implies that a perfect pointing angle $\alpha_B = 0$ may bring the maximal distance reduction and it is best to distinguish the target device from neighboring nodes.

Second, the absolute distance, from the target node or its neighbor to the selecting device, has little impact on identification. The results at 100cm/200cm are almost the same

[2]More cases of different combinations of d_{AB}, d_{AC}, d and ε_{res} have been tested and yield the same conclusion.

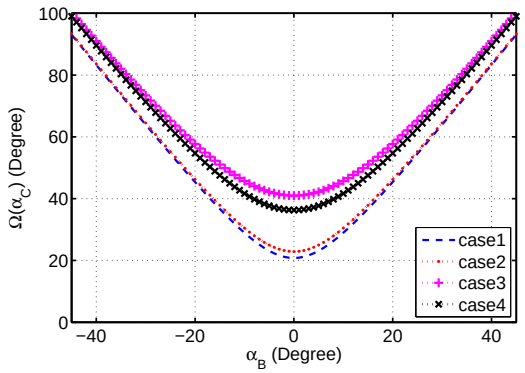

Figure 18. P&C resolution vs pointing angle. Define $\Omega(\alpha_C)$ as $\Omega(\alpha_C) = \max(\alpha_C) - \min(\alpha_C)$, where $\max(\alpha_C)$ and $\min(\alpha_C)$ are the maximal and minimal non-distinguishable angles.

	d_{AB}	d_{AC}	d	d_{res}
Case 1	$100cm$	$100cm$	$30cm$	$0.7cm$
Case 2	$200cm$	$200cm$	$30cm$	$0.7cm$
Case 3	$100cm$	$100cm$	$10cm$	$0.7cm$
Case 4	$100cm$	$100cm$	$30cm$	$2.1cm$

Table 2. Four example cases for angle resolution

for Cases (1) and (2). This can also be seen from Equation (7), which intuitively implies that it is the angle, not distance that matters.

Third, the pointing magnitude does matter much when comparing Cases (1) and (3). The larger the movement, the easier to differentiate from neighboring nodes. We can see that $\Omega(\alpha_C)$ decreases as d increases. From the derivation of (7), we have $(\cos(\alpha_B) - \cos(\alpha_C)) = 2\sin(\frac{\alpha_C+\alpha_B}{2})\sin(\frac{\alpha_C-\alpha_B}{2}) \leq \frac{d_{res}}{d}$. When $\alpha_C - \alpha_B$ is small, it can be approximated by $k \cdot (\alpha_C - \alpha_B) \leq \frac{1}{d}$. Hence, the resolution scope is inverse proportion to the moving distance. Heuristically, larger d should also increase identification accuracy.

Finally, the sample counting resolution is in proportion to the identification angle resolution, as shown in Cases (1) and (4). The lower accuracy the ETOA detection, the larger the neighbor scope that can not be identified from the target one.

In summary, to improve the identification resolution that separates the target device from other neighbors, we have three options: (1) we can point directly to the target node (i.e., $\alpha_B \to 0$); (2) we can extend the pointing action (i.e., larger travel distance d when pointing), and (3) we can reduce the sensing granularity (i.e., small ε_{res}). These findings are consistent with the experimental results.

9 Discussion

Supporting multiple device pairing We do not believe that the case, where multiple pairs of devices are doing simultaneous selection in a spatial proximity, is very common. Note that this case is different from the scenario where multiple devices are doing device pairing at different locations, but only one pair at each location. Our solution works well in the latter case. Nevertheless, we have a couple of techniques to handle the simultaneous pairing at a single location.

The first technique is to simply defer its pairing operation when a device detects an ongoing pairing procedure. Anyway, each pairing operation only lasts for a few seconds, and this short delay will not affect most users' tasks. Each device needs to listen to the channel for a period of time, say, a few seconds, to ensure that no chirp signal is detected. The second technique is to add a header field in the chirp signal that differentiates the chirp signals emitted by different devices. Then, we can permit multiple pairs of chirp signals to be emitted during a single interval. We use the sequence such as ChirpA1.....ChirpB1....ChirpA2....ChirpB2, to enable two simultaneous pairing groups in the interval. Note that the span between two chirp signals in one device pairing is usually 1-2 seconds and one chirp is 50ms. Therefore, we can readily support two to three pairs.

Energy efficiency P&C also seeks to improve energy savings. The energy cost comes from two factors: the energy consumed over the acoustic sensing, and the energy used in the radio channel. The main issue is how to reduce energy consumption during the idle period when P&C is not working in its pairing operation mode. To make device pairing function, both acoustic sensing and radio channel may have to be turned on. The energy cost in acoustic sensing is due to the recording at the receiving device and the transmission of two chirp signals. The waste over Bluetooth/Wi-Fi occurs during the regular scanning operation.

P&C uses the press-button at the selecting device to invoke the working mode while placing the acoustic channels into sleep mode before that. It also activates the typically dozing wireless interface. Therefore, energy-saving at the sending device side is relatively easier. To reduce energy consumption at each receiving device, the target included, we have two options. One is to let the acoustic sensing component sleep and use the wireless channel to wake it up. The device periodically listens to the radio channel from its doze mode. To this end, the selecting side of P&C needs to be modified slightly. When pairing starts, the sender broadcasts a message over the radio channel to notify others, each of which periodically listens to the channel, to activate acoustic sensing. The other alternative is periodic wakeup of the acoustic recording and signal detection, while the device is in the sleep mode during other times. The Wi-Fi or Bluetooth interfaces are only turned on after the button is pressed at the sender or a valid chirp signal is detected at the receiving device. A downside of the solution is the possibly slower response due to periodic wakeup of the acoustic sensing operation. Therefore, P&C trades off energy savings and response time.

10 Related Work

We now compare P&C and the related work. Many solutions to device pairing have appeared in the literature [11–13,15,16,22] . In general, they can be classified into two categories: those requiring extra hardware or infrastructure support, and those working on the standard mobile phone.

In the first category, extra sensors/hardware (e.g., accelerator, visual markers, Ired or NFC) or infrastructure support will be needed to identify which pair of devices are to be connected. Smart-Its-Friends scheme [12] uses motion

sensors to capture the movement patterns when the pairing devices are placed and moved together. The recorded sensing data will be used to identify the target later on. Other solutions such as synchronous gestures [11], Shake Well Before Use [15], Martini Synch [13] use sensors to record synchronous actions by users (e.g., device shaking [15]) or events on both devices (e.g., one bump into the other [11]), so that such sync information can be used to detect the right pairing device. Other approach rely on localization infrastructure to infer spatial reference, in order to differentiate the target from other devices, for example, [16] uses both radio frequency and ultrasonic communications to measure the relative position of the pairing device. Other work also adopts the pointing action to select pairing devices, including those using laser pointer or the camera with 3D visual processing [18, 27], or visual mark like QR code [2] and Semacode [3]. [24] compares their performance and usability to identify devices. P&C is different from these solutions in that it does not use any extra hardware or infrastructure, and still preserves the excellent usability.

The second category of solution works on standard COTS phones. One prominent example is the "scan-browse-select" scheme used by Bluetooth pairing. Each candidate device is identified with a name or address, and the selecting device scans all the nearby devices and retrieves the name/address list. The user then browses the list and selects the one she believes as the right one. Another proposal SyncTap [22] requires both users to simultaneously press and release the buttons on both devices. It still uses the the synchronous action to achieve device pairing but does not require extra sensors. Amigo [30] uses the common radio profile specific to a given location and time to differentiate the pair of devices. It works well when the two pairing devices are in close physical proximity but far away from others; However, the physical proximity identified using radio information is of coarse granularity, and it is hard to identify the target device if it is surrounded by others. P&C is different from these solutions in that the operation is simpler, and does not require explicit user synchronization from both people.

There is also significant research effort focusing on the security aspect of device pairing. They establish shared secret (usually a temporary key or PIN) to differentiate the device pair from others, through three popular techniques: physical contact, PIN input, and out-of-band (OOB) channel. In the physical contact based approach, several industry proposals exist, including Wireless USB Association Models [6] using USB cables, Wi-Fi Protected Setup [4] with its practice Windows Connect Now [5] using Ethernet or USB cable and Bluetooth Simple Pairing [1] using Near Field Communication (NFC). They all use at least one type of auxiliary channel to connect both pairing parties. In the PIN based approach, users may need to input keys on the pairing device. The common practice is bluetooth pairing scheme and UIA [9] uses more user-friendly key introduction, consisting of three words randomly chosen from a dictionary. Users need to tell each other their keys and select accordingly from screen prompts. [29] compares usability of simple PIN-based pairing schemes. In the OOB based approach, infrared, visual and audio channels are used to establish authentication and secrecy. These OOB channels, formed through the sensor and actuator pairs available on the devices, are used to verify whether the keys computed at both devices are identical. Talking-to-strangers [7] illustrates a solution based on bidirectional infrared channels or other location-limited channel. Seeing-Is-Believing [17] and ViC [25] utilize visual channels consisting barcodes or blinking LED for pairing; Loud-and-Clear [10] and HAPADEP [28] devise schemes using audible channels with visual assistance. [20] and [26] use auxiliary visual and audio channels to compare short and simple synchronized audiovisual patterns (i.e., "beeping" and "blinking"). P&C does not focus on the authentication protocol in device pairing, but addresses a few practical attacks. Therefore, P&C complements these authentication schemes and may use one of them, e.g., the OOB-based approach, to further ensure device-level authentication.

11 Conclusion

In this research, we have designed, implemented, and evaluated Point&Connect, an intention-based device pairing solution for mobile phone users. To pair one's phone with another device, a user simply points her phone towards the intended target, in a setting where many other phones are present in the proximity. Each device captures the pointing gesture by measuring the distance reduction via acoustic sensing techniques. The user then selects the device that observes the largest distance change during the gesture. Our experiments have confirmed the effectiveness of the solution.

In a broader context, P&C explores a new design paradigm for solutions over COTS phones. It seeks to exploit simple user action and perception, combined with the standard mobile device capabilities, to achieve features that are previously available only to high-end phones with additional sensors or with infrastructure support. P&C relies on human perception capabilities, which serve as *sensors* and *actuators* for the device, to offset the limitations of COTS phones. Through our experience of P&C, we show that it is indeed a cool and effective way to achieve intuitive device pairing.

12 References

[1] Bluetooth simple pairing. http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf .

[2] QR code. http://www.nttdocomo.co.jp/english/service/imode/make/content/barcode/tool/ .

[3] Semacode. http://semacode.com/ .

[4] WiFi Protected Setup. http://www.wi-fi.org/wifi-protected-setup .

[5] Windows Connect Now Technology. http://www.microsoft.com/windowsxp/using/networking/getstarted/windowsconnectnow.mspx .

[6] Wireless USB Association Models. http://www.usb.org/developers/wusb/ .

[7] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of Network and Distributed System Security Symposium 2002 (NDSS'02)*, San Diego, CA, February 2002.

[8] A. P. et al. Spins: security protocols for sensor networks. In *ACM Mobicom'02*, pages 521–534, 2002.

[9] B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea, F. Kaashoek, and R. Morris. Persistent personal names for globally connected mobile devices. In *OSDI '06: Proceedings of the 7th symposium on Oper-*

ating systems design and implementation, pages 233–248, Berkeley, CA, USA, 2006. USENIX Association.

[10] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human-verifiable authentication based on audio. In *ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, page 10, Washington, DC, USA, 2006. IEEE Computer Society.

[11] K. Hinckley. Synchronous gestures for multiple persons and computers. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 149–158, New York, NY, USA, 2003. ACM.

[12] 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 *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, pages 116–122, London, UK, 2001. Springer-Verlag.

[13] D. Kirovski, M. Sinclair, and D. Wilson. The martini synch: Device pairing via joint quantization. In *ISIT '07: Proceedings of IEEE International Symposium on Information Theory, 2007*, July 2007.

[14] B. Kusy, A. Ledeczi, and X. Koutsoukos. Tracking mobile nodes using rf doppler shifts. In *SenSys '07: Proc. of the 5th Intl. conference on Embedded networked sensor systems*, pages 29–42, New York, NY, USA, 2007. ACM.

[15] R. Mayrhofer and H. Gellersen. Shake well before use: Authentication based on accelerometer data. In A. LaMarca, M. Langheinrich, and K. N. Truong, editors, *Pervasive*, volume 4480 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2007.

[16] R. Mayrhofer, H. Gellersen, and M. Hazas. Security by spatial reference: Using relative positioning to authenticate devices for spontaneous interaction. In *Ubicomp*, pages 199–216, 2007.

[17] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 110–124, Washington, DC, USA, 2005. IEEE Computer Society.

[18] K. Nickel and R. Stiefelhagen. Visual recognition of pointing gestures for human-robot interaction. *Image Vision Comput.*, 25(12):1875–1884, 2007.

[19] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14, New York, NY, USA, 2007. ACM.

[20] R. Prasad and N. Saxena. Efficient device pairing using "human-comparable" synchronized audiovisual patterns. In S. M. Bellovin, R. Gennaro, A. D. Keromytis, and M. Yung, editors, *ACNS*, volume 5037 of *Lecture Notes in Computer Science*, pages 328–345, 2008.

[21] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Sixth*, Boston, MA, USA, Aug. 2000.

[22] J. Rekimoto, Y. Ayatsuka, and M. Kohno. Synctap: An interaction technique for mobile networking. In *Mobile HCI*, pages 104–115, 2003.

[23] E. Rukzio. *Physical Mobile Interactions: Mobile Devices as Pervasive Mediators for Interactions with the Real World*. PhD thesis, University of Munich, 2006.

[24] E. Rukzio, K. Leichtenstern, V. Callaghan, A. Schmidt, P. Holleis, and J. Chin. An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In *Ubicomp'06: Eighth International Conference on Ubiquitous Computing*, September 2006.

[25] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 306–313, Washington, DC, USA, 2006. IEEE Computer Society.

[26] N. Saxena, M. B. Uddin, and J. Voris. Universal device pairing using an auxiliary device. In *SOUPS '08: Proceedings of the 4th symposium on Usable privacy and security*, pages 56–67, New York, NY, USA, 2008. ACM.

[27] G. Schmidt, Y. Baillot, D. G. Brown, E. B. Tomlin, and J. E. I. Swan. Toward disambiguating multiple selections for frustum-based pointing. In *3DUI '06: Proceedings of the 3D User Interfaces*, pages 87–94, Washington, DC, USA, 2006. IEEE Computer Society.

[28] C. Soriente, G. Tsudik, and E. Uzun. Hapadep: Human-assisted pure audio device pairing. In *ISC*, pages 385–400, 2008. `http://eprint.iacr.org/2007/093.pdf` .

[29] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. `http://research.nokia.com/files/NRC-TR-2007-002.pdf` , 2007.

[30] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara. Amigo: Proximity-based authentication of mobile devices. In *Ubicomp*, pages 253–270, 2007.