# Security by Spatial Reference: Using Relative Positioning to Authenticate Devices for Spontaneous Interaction

Rene Mayrhofer, Hans Gellersen, and Mike Hazas

Lancaster University, Computing Department, South Drive, Lancaster LA1 4WA, UK
{rene,hwg,hazas}@comp.lancs.ac.uk

**Abstract**

Spontaneous interaction is a desirable characteristic associated with mobile and ubiquitous computing. The aim is to enable users to connect their personal devices with devices encountered in their environment in order to take advantage of interaction opportunities in accordance with their situation. However, it is difficult to secure spontaneous interaction as this requires authentication of the encountered device, in the absence of any prior knowledge of the device. In this paper we present a method for establishing and securing spontaneous interactions on the basis of *spatial references* that capture the spatial relationship of the involved devices. Spatial references are obtained by accurate sensing of relative device positions, presented to the user for initiation of interactions, and used in a peer authentication protocol that exploits a novel mechanism for message transfer over ultrasound to ensures spatial authenticity of the sender.

## 1 Introduction

Spontaneous networking is of potentially great value to mobile users as it can enable them to associate their personal devices with devices encountered in their environment, and thereby to take advantage of serendipitous interaction opportunities. Spontaneous interaction in ubiquitous computing has for example been studied for applications such as social interaction and game-playing in mobile user communities. However, the potential of such interactions extends into areas that may involve more sensitive data and transactions, such as use of a vending machine over a wireless link, or direct payment transactions between two mobile devices. For such applications to be acceptable in a spontaneous network setting, a user must be able to authenticate the interaction of their personal device with the intended target device. They must be able to ascertain that the network entity their device connects to is identical with the physical device 'in front of them'. Furthermore, given the inherent vulnerability of a wireless communication channel, they must be able to rule out the presence of a third party established as 'man-in-the-middle' between their device and the target.

In a managed network environment, device-to-device authentication would be based on prior knowledge of each other or access to a trusted third party, but

in spontaneous networks neither can be assumed to be available. Instead it is necessary to provide an out-of-band mechanism alongside the wireless channel, for secure key exchange or verification of keys that have been 'speculatively' exchanged over the wireless channel. A wide range of mechanisms have been discussed in the literature, from user entry of PIN codes [1] and direct electrical contact [2] to use of communication channels with inherent physical limitations, such as infrared, audio and ultrasound [3,4].

In this paper we present a novel approach for device-to-device authentication in spontaneous networks. The main contribution is a method that uses *spatial references* to establish and authenticate interaction between a pair of devices. Spatial references capture the spatial relationship with a target device in terms of bearing and distance, and are used in an authentication protocol that couples key verification with verification of the relative position of the sender. The method and protocol are a general contribution in the sense that they can be implemented with any peer-to-peer sensing approach capable of providing accurate relative bearing and distance. However, we also contribute a concrete implementation, using a combination of radio frequency (RF) and ultrasonic (US) communication for measurement of spatial relationships.

As ultrasonic ranging is susceptible to certain attack scenarios (as we will explain in the course of the paper), we further contribute a novel coding technique for spatially-dependent message transfer over an ultrasonic channel. This technique allows a sender to transmit a message such that it can only be successfully decoded if it is received at a particular range. The technique is a key component in the protocol implementation we present, but can have wider application in ultrasonic systems independent of the particular problem we consider here.

In the subsequent section we will position our research with respect to related work, and then proceed to a description of the overall design of our method, the underlying sensing approach and the proposed user interface. This will be followed by a threat analysis, the description of a peer device authentication protocol as our core contribution, and an analysis of security and performance.

## 2 Related Work

Peer device authentication was first highlighted as a distinct security challenge emerging in ubiquitous computing by Stajano and Anderson, who proposed the 'Resurrecting Duckling' model for secure transient device association, bootstrapped from direct electrical contact [2]. Others have proposed channels for authentication that do not require direct contact but are 'location-limited' [3] or 'physically constrained' [4], including infrared beams [3], laser beams [5], and ultrasound [6]. Our method of spatial references effectively expands on the idea of location-limitation, using spatial measurements in addition to channel limitations, in order to further limit the position from which a device can successfully authenticate.

A variety of methods rely more on the user for device authentication, for instance for manual key entry [1], scanning of visual tags on the target and com-

parison with wirelessly received material [7] and verification of spoken messages generated by devices [8]. Our approach also has the user in the loop, however does not involve any user interaction *solely* targeted at security. Instead, we provide the user with a spatial technique for initiating interaction with another device; the spatial relationship is captured in this process and is then used for securing the interaction without need for further intervention of the user.

Our concrete implementation is based on the use of US as out-of-band channel. Kindberg et al. have before us proposed the use of US alongside an RF wireless channel in a protocol for validation and securing of spontaneous interaction [6]. The idea of the protocol is for devices to first exchange keys, and then to verify that the intended device is in possession of the correct key, by having the device send a nonce in plaintext over ultrasound and over RF. However, the protocol design does not consider potential attacks on the ultrasonic channel. A specific problem is the reliance on ultrasonic time-of-flight measurements for verification of device authenticity, as these involve synchronisation over the RF channel and are open to attack scenarios in which an attacker may appear nearer or further than they are [9]. As the protocol has not been implemented it is also not clear how precisely the nonce would be transmitted and what the security implications of this would be. In its general design, our protocol is similar to that of Kindberg et al., but we attend specifically to the issue of trustworthiness of ultrasonic ranging, and provide a complete implementation with security and performance analysis.

Other related work includes spatial interaction techniques. Hazas et al. [10], while not considering security, have presented an approach that uses ultrasonic peer-to-peer sensing for spatial discovery of other devices within interaction range, and work expanding on this has considered visualisation of the devices' positions in the user interface in order to ease interaction across devices (e.g. enabling transfer of a document to another device by a simple drag-and-drop operation) [11,12]. We employ the same principle in our method to let users initiate spontaneous interactions by means of spatial discovery and selection of the target device, but extend the approach by adding security in a seamless manner.

## 3 Security by Spatial Reference

Central to our method is the concept of *Spatial References*. A spatial reference captures the spatial relationship of a client device with a target device. A key aspect of spatial references is that they can be obtained independently by a user (seeing devices in front of them) and by their device (using sensors), and that a user can match what their device senses with what they see. Spatial references thus serve to establish shared context between a user and their device: a device can report a discovered network entity in a manner that the user can match with encountered devices, and a user can identify a target device in a way that their device can match with network entities.

### 3.1 Design of the method

In our method for establishing and securing spontaneous interactions, spatial references are used for discovery of devices, for selection of a target devices, and for verification that interaction is secured between the 'right' devices. This involves the following steps:

1. The user's device uses a combination of network discovery and spatial sensing for *spatially-bounded discovery* of devices.
2. The spatially discovered devices perform spatial measurements to compute their relative positions.
3. Users are provided with a visualisation of available devices, integrated in the user interface of their personal device and laid out in correspondence with computed positions relative to the user's device.
4. Users initiate interaction and communication with a device by selection of the corresponding visual object, using direct manipulation techniques available in their user interface.
5. Selection of a device for spontaneous interaction triggers a protocol for key exchange with the target device and verification that no other devices can be present as 'man-in-the-middle' between the user's device and the target.
6. Once it has been asserted that exchanged keys are authentic, they are used for securing the communication channel between user device and target, and the initiated interaction can take place.

### 3.2 Spatial discovery and sensing

For a concrete implementation of spatial discovery and sensing we base our method on the *Relate* system for relative positioning introduced by Hazas et al. [10]. The Relate system provides wireless sensors implemented as USB dongles that can be readily used to extend host devices (such as laptops or PDAs) with spatial sensing. The Relate sensors contain three ultrasonic transducers (to cover space in front, left and right of the device) and they operate their own ad hoc network over combined radio frequency (RF) and ultrasound (US) channels (note this sensor network is separate from the wireless network that connects their host devices). Protocol functions implemented over the sensor network include network discovery and management, collaborative ultrasonic sensing, collection of measurements, and exchange of host information. The Relate sensors specifically support spatial discovery of their host devices by exchanging the hosts' network addresses over the sensor network.

The Relate sensors use RF messages to co-ordinate ultrasonic sensing. Sensing is performed by one node emitting ultrasound on its transducers, while all other nodes listen for a pulse on their transducers. The receiving sensors measure the peak signal values and the times-of-flight of the ultrasonic pulse with their three transducers. The smallest time-of-flight is used to calculate a distance estimate, and an angle-of-arrival estimate is derived from the relative spread of peak signal values measured across the transducers. The Relate sensors use RF

**Fig. 1.** Integration of spatial references to near-by devices in the mobile user interface; left: extension of Guinard et al.'s Gateways [12]; right: Kortuem et al.'s map view [11].

to share and collect sensor data, and each sensor provides the collected data to its host device. This then enables the host devices to compute their relative positions very accurately. Hazas et al. report a 90% precision around 8 cm in position and 25° in orientation [10]: these figures and our practical experience suggest sufficient accuracy for reliable disambiguation of devices. By collaboratively sharing US measurements over RF, partial obstruction can be dealt with in principle. However, for spatial authentication we rely on direct line of sight between the authenticating devices.

### 3.3 User interface design

Spatial discovery and sensing happen automatically and unobtrusively. Users are then provided with a visualisation of the computed relative positions of devices in the interface on their own personal device. The visualisation has to be such that a user can associate a visual screen object with a device in their environment. Figure 1 shows two possible implementation. The one on the left is based on Guinard et al.'s *Gateways* [12]: these are screen objects arranged around the edge of the user interface, representing devices in the indicated direction relative to the user's device, and here extended to also show distance information. The one on the right is adapted from [11] and shows a map view with icons spatially arranged in correspondence with the actual layout of devices discovered around the user's device. Key to our concept is that the visualisation reflects the 'real' spatial layout, so that users can make a connection between what they see and what their device sees (and visualises). This allows users to invoke interactions by spatial reference, for example simply by dragging an object onto a Gateway or icon representing a remote device. A device thus selected as targeted is associated with a particular bearing and distance as measured with on-board sensors.

## 4 Threat Analysis

The key idea underlying our method is to use spatial references for verification of device authenticity. In this section we consider threats in the context of the ultrasonic sensing approach we introduced above, as well as threat scenarios that arise on application level.

### 4.1 Attacker capabilities

There are three channels of concern: the communication network between devices, e.g. wireless LAN with a TCP/IP stack, the radio frequency channel used for communication between spatial sensing devices ($RF$), and the ultrasound channel used for sending and receiving ultrasonic pulses ($US$). We assume an attacker ('Eve') to be capable of gaining complete control over the wireless communication channels. This allows Eve to perform a 'man-in-the-middle' (MITM) attack on the wireless channels. Assuming to devices A and B, the attacker E can pretend to A that it is B, and to B that it is A, and thus agree to a cryptographic key with A and separately with B. A and B will be unaware of this and believe to communicate securely with each other when in fact they are communicating via E (who might be partially or completely relaying their messages).

The aim of our method is to prevent that a man-in-the-middle can succeed. To this end, spatial references are used during the authentication process, and are therefore subject to potential attack. We can distinguish between three different attacker capabilities with regards to tampering with spatial references, in order of increasing complexity:

1. *RF-only*: Attacks on any of the wireless channels (RF) are the most dangerous, because they can be carried out inconspicuously (see e.g. [13]). With directed antennas, the possible range of an attacker can significantly exceed the normal range of the RF channel, as has been demonstrated by an attack on mobile phones via Bluetooth over a distance of over 1.7 km.
2. *US in room*: Control over the US channel, on the other hand, is assumed to be limited. First, for attacks on this channel, an attacker needs to be physically present in the same room (US is effectively blocked by solid materials such as walls, doors, and windows). Second, although eavesdropping is easily possible, injecting US pulses is more difficult. We assume an attacker to be capable of injecting US pulses at any time with arbitrary strength. Injection in this sense means to insert completely new messages into the US channel, while modifying, replacing, or removing other messages is not possible without detection.
3. *US in line*: An attacker in the same room can inject US pulses, but receiving devices will be able to detect the different angle of arrival. The reason is that – in contrast to distance measurements – angle of arrival is inferred from relative measurements, i.e. differences in time of arrival or signal strength. We assume it impossible to fake the angle of arrival of a US pulse, bar the capability of sound forming for US (which has not yet been shown to be

possible). However, an attacker could be placed in line with A and B, and thus not be required to fake the angle.

### 4.2 Sensing-level threats

Attacking the RF channels creates three threats specific to our spatial sensing system:
(a) by removing all RF messages sent from or to a single device, an attacker Eve can prevent the device from entering the sensor network, and thus *make a specific device disappear* for all other devices – however, this can be detected by the device in question.
(b) by changing RF messages, Eve can *tamper with shared measurements*, i.e. those that are taken by remote sensors and exchanged between Relate sensors. Additionally, US ranging depends on trigger packets sent via RF.
(c) by controlling these trigger packets, Eve can *manipulate distance measurements*.

If Eve is spatially aligned line with A and B, she could also send US messages delayed or ahead of schedule to the effect that her position, from Alice's point of view, appears to be where Bob is. This creates a fourth specific threat, namely (d) to *fake the perceived distance*.

Note that, in contrast to ranging measurements, angle of arrival measurements are trustworthy in our sensing system, as they are derived from signal peak values measured on with sensors oriented in different directions, and not from time-of-flight as proposed in [6].

### 4.3 Application-level threats

The possibility to tamper with spatial references leads to three specific attack scenarios on the application level.

1. *Replacement*: The first possibility for attack is to virtually replace another device. This requires two steps: First, the original communication partner, in this case B, needs to be 'silenced' so that it will no longer be visible in terms of wireless communication and measurements. Second, Eve needs to fake her position to appear at the same place where the user ('Alice') expects B to be. In this attack, interaction happens only between Alice and Eve, and no interaction happens with B. Scenarios for this threat are thus limited to asymmetric settings where B is an infrastructure device not monitored by users.
2. *Asynchronous MITM*: When the scenario includes application-level feedback from B to Alice, there is the possibility for an asynchronous MITM attack. An example for such expected feedback is printing: Alice, when sending a document to B, expects her document to print shortly afterwards. In this case, Eve first replaces B as in the first threat, but only temporarily. After finishing authentication with Alice, she authenticates with B and forwards the intercepted messages that were originally intended for it. Eve could therefore

try to avoid detection by forwarding to B and thus completing the high-level interaction. This scenario requires that B does not verify the origin of the messages, i.e. that only Alice authenticates B, but not the other way around.

3. *Synchronous MITM*: For live interaction, like a chat or voice communication between two users (Alice and Bob) over the secure channel, even the slight delay of an asynchronous MITM attack would be noticeable. The most dangerous threat, because it is hard to detect when the attack is being performed, is that of a synchronous MITM. For a synchronous MITM, Eve first attacks the wireless channel as in the previous threat scenarios. But then she remains passive during spatial discovery and mutual positioning of Alice and Bob. Only during spatial authentication she tampers with the spatial measurements. Thus, she remains virtually undetectable for both Alice and Bob, while still having full access to their communication. This requires Eve to be physically between Alice and Bob, because both verify angle of arrival of spatial relationships.

## 5 Key Agreement and Peer Authentication

We secure spontaneous interaction between two devices A and B in two phases, *key agreement* and *peer authentication*, as shown in Fig. 2. In the first phase, we let A and B establish a shared key using a standard, unauthenticated key agreement protocol, such as Diffie-Hellman (DH) [14]. If this is successful, then A and B can use the agreed key to protect their communication against eavesdropping and tampering, with E being unable to gain sufficient knowledge of that shared key. To protect A and B against MITM attacks, we use a second phase for peer authentication (A establishing that it is really talking to B, and vice versa), and for verification that A and B are in possession of the same key (which would rule out the presence of a MITM due to the unique-key property of a protocol such as DH).

### 5.1 Peer authentication

The peer authentication process is designed to be symmetric, which means that the two devices A and B authenticate each other. Even though the interaction is initiated by A in response to Alice's selection of B as target, it will often be appropriate that B can also verify the sending device and its relative position, for example to provide its user Bob with a verified visual indication in his user interface of *where* a received document has been sent from (and thus prevent replacement or asynchronous MITM attacks). As a starting point for authentication, A has a spatial reference to B as derived from the user's selection of B as her target, and B can base authentication on a corresponding spatial reference to A.

Devices A and B use the RF and US channels of their sensor nodes for peer authentication in order to tightly couple this process with spatial sensing. The devices engage in a protocol designed to establish that (i) they have agreed

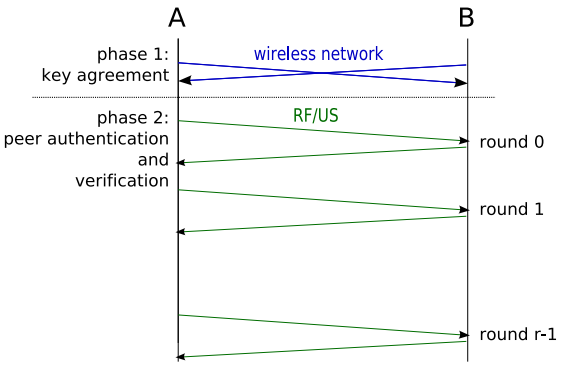


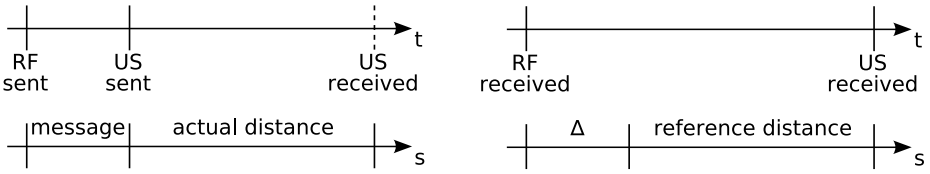

**Fig. 2.** Devices A and B secure their interaction by key agreement over a wireless network channel, followed by peer authentication over the RF and US channels of their spatial sensors.

to the same key, and (ii) they are A and B as mutually verifiable by spatial reference. The devices approach this by generating a nonce (a random number used only once) and by transmitting the nonce encrypted over the RF channel. They also transmit the plaintext nonce over the US channel in a series of smaller parts that are coded within the actual distance measurements. When the devices receive these transmissions, they decrypt the RF message, verify that the content matches the nonce received via US, and thus establish whether their keys match. For this approach to be secure, the encoding and the transmission of these nonces need to be coordinated. In the following, we discuss these two issues and how they interact with each other.

### 5.2 A spatial coding technique for trustworthy ultrasonic ranging

When a device receives an ultrasonic pulse, it computes a distance measurement based on the time-of-flight. As explained above in section 4.2, these distances can be tampered with. We therefore introduce a method to embed information in ultrasound pulses, which (i) allows to use US as an out-of-band channel for message exchange, and (ii) makes the distances trustworthy.

During authentication, the sender delays the sending of pulses to the effect of adding a certain perceived distance to the measurement, where the added distance represents information (in our protocol, a substring of the nonce). When for instance A receives a pulse and computes a distance, this distance is the actual distance from the sender plus a distance representing the message. A proceeds with subtracting the reference distance it has of B (note the reference distance is captured when the user selects a device for interaction). This will let A retrieve the information (represented as added distance) correctly only if the received pulse has been sent from a range that corresponds with the relative position of B. That is, a correct reconstruction of the message implies that the distance is




(a) The sender delays an US pulse after the RF trigger to code a message, which corresponds to a distance in the spatial domain.

(b) The receiver measures time from trigger to pulse arrival, and subtracts a stored reference from the corresponding distance to retrieve the message.

**Fig. 3.** Message transmission embedded with ultrasonic ranging: The receiver will only be able to retrieve the message if the sender's distance matches the stored reference.

equal to the reference measurement, and therefore constitutes and implicit check of spatial integrity. Figure 3 illustrates this mechanism for message transmission over ultrasound with implicit verification of sending range. In addition to this implicit distance check, A can verify that the pulse was received from a direction corresponding with the reference held for B, thus effectively eliminating the possibility that the US transmission originates from another device but B.

### 5.3 Preventing MITM relaying

A and B can thus verify that ultrasound pulses are received from the intended partner device but it is still possible that E is present as MITM on the RF channel. E would be able to infer the nonces exchanged between A and B by taking its own US measurements (note that this only requires eavesdropping on US pulses, which is simple to do as long as E is in the same room), and it could then use its keys (maliciously agreed with A and B in the key agreement phase) to re-encrypt the nonces in order to pass the key verification checks of A and B. To rule this possibility out we use an interlock protocol, which in essence commits the sender of a message to the message content before it has been transferred completely [15]. For this purpose, A and B split the encrypted nonces into multiple parts and take turns in transmitting their parts. The nonces are encrypted with a block cipher, which means that all message parts need to be reassembled before the message can be decrypted to retrieve the nonce. If E now receives a message part from A intended for B, it can not retrieve any part of the nonce. E will also not receive more message parts from A unless it passes the current one on to B, as A and B strictly adhere to turn-taking. E's only choices are then to guess the content for all message parts that will 'pass through' (before they are even transmitted by A and B, let alone decrypted by them) in order to re-encrypt these successfully (this is practically impossible), or to relay message parts unchanged in which case A and B will discover that their keys do not match (thereby detecting the presence of a MITM and aborting authentication). The interlock protocol thus rules out that a MITM attack on the RF channel can succeed during peer authentication.

### 5.4 Protocol specification

An overview of the protocol phases is shown in Fig. 2. Key agreement takes place over a wireless network channel, and subsequent key verification and peer authentication over the RF/US channels of their spatial sensors. The second phase involves turn-taking of the parties in an interlock protocol over a number of rounds $r$. This number will be agreed between devices, in consideration of the security level, protocol duration, and US channel capacity. The US channel capacity $b_u$ is the number of bits that can be reliably transmitted as distance offset in each round, and will depend on the characteristics of the sensors used and sensing protocol details. Assuming a nonce of 128 bits, we would need $\lceil 128/b_u \rceil$ rounds for transmission of the nonce over US. However, a smaller number of rounds may be agreed to complete the protocol faster, compromising on how many bits of the nonce are eventually compared for key verification. With $r$ agreed, we then set the number of bits that will transmitted over the RF channel in each round to $b_m := \lceil 128/r \rceil$, splitting the encrypted nonce into equal message parts.

We will now describe our protocol more formally using the following notation: $c := E(K, m)$ describes the encryption of plaintext $m$ under key $K$ with a symmetric block cipher, $m := D(K, c)$ the corresponding decryption, $H(m)$ describes the hashing of the message $m$ with a secure hash algorithm, and $m||n$ describes the concatenation of strings $m$ and $n$. Additionally, the notation $M[a : b]$ is used to describe the substring of a message $M$ starting at bit $a$ and ending at bit $b$. Messages that are transmitted to the other party are printed in bold.

1. *Key agreement*, using the Diffie-Hellman key establishment protocol:
   (a) A chooses a random number $a \in \{1, ..., q-1\}$ and transmits $\mathbf{X} := g^a$,
   B chooses a random number $b \in \{1, ..., q-1\}$ and transmits $\mathbf{Y} := g^b$
   (b) A computes $K_a^{Sess} := H(\mathbf{Y}^a)$ and $K_a^{Auth} := H(\mathbf{Y}^a||PAD)$ with some secure hash algorithm,
   B generates $K_b^{Sess}$ and $K_b^{Auth}$ correspondingly from $\mathbf{X}^b$
   The numbers $g$, $q$ and the string $PAD$ are assumed to be publicly known. Although we envisage the use of ephemeral keys, i.e. new values for $a$ and $b$ for each protocol run, it might be advantageous to use long-term values for performance reasons. We use $K^{Auth}$ $(= K_a^{Auth} = K_b^{Auth})$ for key verification in the peer authentication phase, and $K^{Sess}$ $(= K_a^{Sess} = K_b^{Sess})$ for subsequent channel security if the verification succeeds. The additional hashing to compute two different shared keys provides forward secrecy in the case of leaked authentication key material (cf. [16, section 15.8.4]), for example by a known plaintext attack on $E(K_x^{Auth}, N_x)$ after the respective $N_x$ is revealed in the following steps.
2. *Peer authentication*:
   (a) A chooses a nonce $N_a \in \{1, ..., 2^{128} - 1\}$ and computes $M_a := E(K_a^{Auth}, N_a)$,
   B chooses $N_b$ and computes $M_b$ correspondingly with $K_b^{Auth}$
   (b) *For each round* $i := 0 \ldots r-1$:
      – A transmits a RF packet $\mathbf{M_a^i} := M_a[i \cdot b_m : (i+1) \cdot b_m - 1]$ and an US pulse $\mathbf{USP_a^i}$ delayed by $N_a[i \cdot b_u : (i+1) \cdot b_u - 1]$ units,

- B receives message part $\mathbf{M_a^i}$ and US pulse $\mathbf{USP_a^i}$, derives a distance measurement $d_{b,a}^i$, and uses the stored reference measurement $d_{b,a}$ to reconstruct the distance-coded message $\Delta_a^i := d_{b,a}^i - d_{b,a}$. B also verifies the angle of arrival $\alpha_{b,a}^i$ and compares it with the stored reference measurement $\alpha_{b,a}$. If the difference exceeds the typical measurement error, B aborts the authentication protocol with an error message.
- B transmits $\mathbf{M_b^i} := M_b[i \cdot b_m : (i+1) \cdot b_m - 1]$ and $\mathbf{USP_b^i}$ delayed by $N_b[i \cdot b_u : (i+1) \cdot b_u - 1]$ units, and acknowledges receipt of A's RF and US messages for round $i$,
- A receives $\mathbf{M_b^i}$ and $\mathbf{USP_b^i}$, verifies angle of arrival, computes $d_{a,b}^i$, uses the reference measurement $d_{a,b}$ to reconstruct $\Delta_b^i := d_{a,b}^i - d_{a,b}$, and acknowledges B's messages for round $i$

(c) A reassembles all received RF packets $M'_b := \mathbf{M_b^0} || \ldots || \mathbf{M_b^{r-1}}$, decrypts the message $N'_b := D(K_a^{Auth}, M'_b)$, reassembles the nonce from the distance offsets $N''_b := \Delta_b^0 || \ldots || \Delta_b^{r-1}$, verifies that $N''_b = N'_b[0 : r \cdot b_u - 1]$, and sets $K := K_a^{Sess}$ on match or $K := null$ otherwise,
B reassembles $M'_a := M_a^0 || \ldots || M_a^{r-1}$, decrypts $N'_a := D(K_b^{Auth}, M'_a)$, reassembles $N''_a := \Delta_a^0 || \ldots || \Delta_a^{r-1}$, verifies that $N''_a = N'_a[0 : r \cdot b_u - 1]$, and sets $K := K_b^{Sess}$ on match or $K := null$ otherwise

Note, if $b_u < b_m$ (i.e. if fewer bits are transmitted via US than via RF) then step 2c) only compares $r \cdot b_u$ bits of the nonce.

If key agreement and peer authentication are completed successfully, then A and B can use the session key $K$ to establish a secure channel. The key can be used as a shared secret for one of the standard protocols such as IPSec with PSK authentication, or one of the recently specified TLS-PSK cipher suites [17]. Other options are WPA2-PSK or EAP-FAST. $K$ can be used directly as key material, rendering additional asymmetric cryptographical operations in the secure channel implementation unnecessary and thus speeding up channel establishment.

### 5.5 Implementation

We have implemented the key agreement phase of our protocol over TCP/IP. As a secure hash we use $\text{SHA}_{\text{DBL}}$-256, which is a double execution of the standard SHA-256 message digest to safeguard against length extension and partial-message collision attacks [16]: $\text{SHA}_{\text{DBL}}\text{-256} = \text{SHA-256}\left(\left(\text{SHA-256}\left(m\right)\right)|m\right)$.

The peer authentication phase of the protocol has been implemented over the RF/US channel of the Relate sensors, using AES (Rijndael) with a key size of 256 bits as secure block cipher for the interlock protocol. The protocol is tightly integrated with the Relate spatial sensing protocol. RF packets transmitted for authentication serve simultaneously as trigger packets for ultrasonic time-of-flight measurement. Pulses emitted on the US channel serve simultaneously for ranging and for transmission of nonce message parts.

Derived from the characteristics of the Relate sensors, we have set the number of bits transmitted in each round over US to $b_u := 3$. In each round, the 3 bit number is coded as multiples of 25.6 cm which the sender adds as offset to the

receiver-perceived distance by delaying the US pulse. At the receiver end, this allows for +/-12.8 cm of measurement inaccuracy to retrieve the 3 bits correctly (note the reported precision of Relate sensors for this level of accuracy is over 95%). The duration of a round is about 200 ms (longer if other devices present are allowed to 'interrupt' the authenticating peers for spatial sensing and exchange of measurements). Transmission of the complete nonce would require 43 rounds but the number of rounds has been kept variable in our implementation to allow users to define their required level of security.

## 6 Security Analysis

### 6.1 Message channels

In our case, information is transmitted both via RF and via US. To safeguard against *eavesdropping* all RF packets are encrypted with an authentication key, but over US the nonce will become gradually revealed as the protocol proceeds. The interlock protocol ensures that this will be of no use to an attacker, as the protocol forces commitment of encrypted nonce message parts over RF before the entire nonce can be intercepted on the US channel. The nonce is also strictly used only once which rules out *replay* attacks. Complete or selective *denial-of-service* attacks can not be protected against under our assumption of completely insecure RF channels.

As described above, the main motivation for using the interlock protocol is to protect against man-in-the-middle attacks *during* authentication. An RF-only MITM attack would be noticed, and we therefore need to analyse the possibilities for a concurrent attack on the US channel.

### 6.2 Ultrasonic sensing and message transmission

Our approach to coding random nonces (section 5.2) and transmitting them via interlock (section 5.3) prevents all the threats outlined in section 4.2: Threat (a) constitutes a selective denial-of-service attack that can be detected by time-outs (when the selected device does not respond at all) or authentication failures (when the attacking devices responds from a different spatial position). Threat (b) does not apply to our protocol, because shared measurements are not used during authentication. Threats (c) and (d) are prevented by the random delays. As E can not know in advance when a US pulse will be sent by A or B (the delays are derived from the random nonce part that is kept secret until sending the pulse), it can not construct the encrypted RF packets to match these delays. If E injected own US pulses, A and B would also receive the original ones and thus detect that an attack is happening. E's only chance would be to cancel US pulses in-transit by generating appropriate anti-US pulses, but this is considered prohibitively difficult. Furthermore, E would need to be positioned precisely in the line-of-sight between authenticating devices in order to attempt interception and manipulation of US pulses but this presence literally in the middle between

devices would be obvious to the user. Note that this MITM device can not be arbitrarily small due to a physical limits on the minimum size of ultrasound transducers.

One remaining risk is that E is positioned in line with A and B, but farther away instead of in between. If E performs a selective denial-of-service attack on B and forges distance measurements before authentication is started, it will be able to fake its perceived and subsequently visualised position as seen by A. Although for security purposes one does usually not trust other devices's measurements (they might be collaborating for an attack), we note that these measurements, shared by benign devices over the Relate RF network, may serve to reveal ongoing attacks such as this one. The shared measurements are not used for increasing trust in an authentication protocol run or providing proof of authentication, but they may still be used for decreasing trust in a protocol run, when shared measurements do not match local ones. Attacking networks of multiple Relate devices should therefore be considered significantly harder than attacking just two devices.

We should also note that attacks on the sensing level become harder in scenarios involving mobility of devices. Positioning an attacker unsuspiciously and directly in line between A and B is not trivial even in static settings. When at least one of the interacting devices is mobile, an attacker would need to be constantly re-positioned (or virtualized by sound forming, which is considered infeasible with the current state of the art in ultrasonic systems).

### 6.3 Applications

The application-level threats described in section 4.3 are specific to our method. With the protections of the sensing level described above, the remaining threat is the misrepresentation of E at the position of B as seen by A. *Replacement* of infrastructure devices is hard to detect, and therefore difficult to protect against. One possibility is to create an explicit application-level feedback from B that can be verified by Alice, for example to lighting an LED for a few seconds whenever authentication has succeeded. If Eve replaces B, then B will not light its LED and Alice can subsequently abort the interaction. The same protection can be used against *asynchronous MITM*, which effectively transforms these two scenarios into a *synchronous MITM* setting. However, this adds an additional step in the interaction process that may not be desirable for many applications. A more pragmatic protection against these remaining replacement and asynchronous MITM threats is to protect against E being in line with A and B by physical means, e.g. simply placing B directly in front of a wall and thus making it impossible for E to be hiding 'behind' it.

*Synchronous MITM* seems prohibitively difficult to perform under the above analysis of the sensing level protection, because of the necessary in-transit attacks on US pulses.

### 6.4 User interaction

The overall security of our method depends on the correct selection of the target device, and the correct association of the target with a spatial reference. We need to consider two possible sources of error or incorrect association. One is that the network communication in the initial steps of our method is not secure. A user can trust the relative position information it has of other devices as this is measured with on-board sensors but any additional information exchanged may be interfered with by an attacker. For example, the Gateway interface shown in Fig. 1 is based on locally measured spatial information but in addition visualises type of discovered device based on information received over the wireless network. An attacker might tamper with this to the effect that a different device type is indicated, which might mislead the user.

The second risk at the level of user interaction is that the user selects the 'wrong' device in their user interface, in the worst case an attacker positioned near the actual target. i.e. E instead of B, in their user interface. The visual design of the UI and the accuracy of the spatial layout in correspondence with the 'real world' arrangement of devices will be key factors in reducing the risk of faulty selection, which of course will also be dependent on number and arrangement of devices discovered and visualised.

## 7 Performance Evaluation

The authentication protocol involves evaluation of sensor data with inherent limitations in accuracy and precision. It is therefore critical to assess impact of sensor limitations on practical performance.

### 7.1 Robustness against 'false negatives'

Sensors are inherently imprecise. Our authentication protocol is designed to account for the resulting variance in sensor readings, but only within limits that are consistent with secure authentication of devices by spatial reference (i.e. there must be no possibility that devices become confused due to allowances made for sensor error). As a consequence, the protocol can fail to authenticate legitimate peers when sensor errors occur that exceed built-in tolerance.

Figure 4 shows the success rate of authenticating legitimate peers dependent on the number of rounds of the interlock protocol and the distance between the devices. For this experiment, two devices were positioned facing each other in direct line of sight at distances of 50cm and 100cm. For each number of rounds, 250 protocol runs were performed. As shown in Fig. 4, success rates are at least 85% and typically above 95%.

Authentication only succeeds if every single US measurement taken during the protocol rounds is sufficiently accurate. In our experiment, the success rate did not decrease significantly with the number of rounds. However under less controlled conditions (e.g. slight movement of devices during the protocol run)

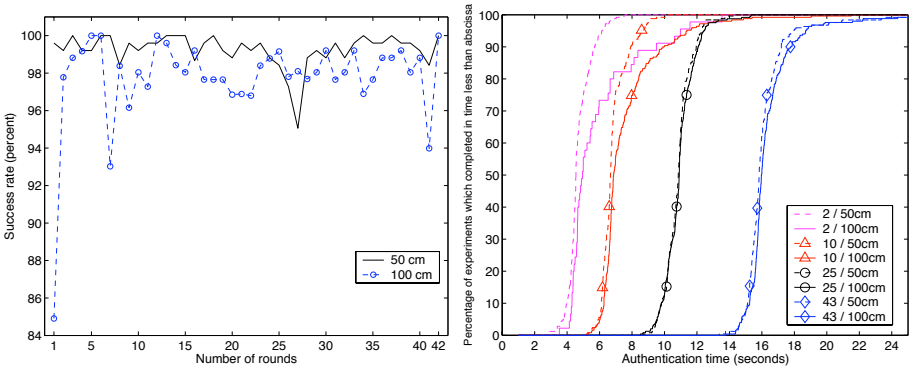



**Fig. 4.** Authentication success rate depending on the security level and distance (left); Authentication speed depending on the security level and distance (right).

a more notable decrease might be expected, as the probability of an erroneous measurement increases with the number of rounds. Note that the impact of distance on success rate is not very pronounced and appears to be within error of measurement (success rates for the larger distance are on average lower, but not consistently).

### 7.2 Speed versus security

There is an inherent trade-off in our protocol between speed and security. The resistance against attacks increases with the number of rounds used for the interlock protocol, because each round transmits 3 bits of entropy for verifying the nonce. Therefore, an attacker's chance of guessing a nonce equals $1/2^{3r}$. To put this into perspective, after only 5 rounds a nonce would already be harder to guess than a randomly generated 4-digit PIN number. Also note that our protocol is symmetric, which means that an attacker would need to guess two nonces correctly in order to deceive the authenticating devices as MITM.

Figure 4 (right) shows this trade-off with measurements taken for 2, 10, 25, and 43 rounds, obtained with the same experimental setup of devices as described above. The variations in the time necessary for authentication over a certain number of rounds stem from the specifics of the underlying RF/US sensing protocol which can require message retransmits. The dependency on the distance between the devices is again marginal. As can be seen, a complete authentication takes around 12 s for 25 rounds.

It is important to understand that a compromise on the number of rounds in our protocol only impacts on an attacker's one-off chance to guess the correct nonce to stage an undetected MITM attack. It does not impact on the security level of 128 bits that will be provided after successful authentication. This difference is even more pronounced than in the usual online vs. offline attack discussion, because of the tight coupling with interaction at the user level. An

attacker can not repeatedly attack the authentication protocol with an online attack, because it is only triggered by an explicit user action. Therefore, there is only a one-off chance for an attack, and any computational attacks are therefore matched with a security level of 128 bits. Nonetheless, our protocol allows the user or application to choose the best compromise between speed and security and scales up to a 128 bit level even for the single attack possibility.

## 8 Conclusion

We have contributed and discussed a method for establishing secure spontaneous interaction on the basis of spatial references. Spatial references are a type of context that allows users to match what they see with what their device sees. At the core of our method is a peer authentication protocol that uses relative bearing and distance between devices. We have presented an implementation using ultrasound for spatial measurements; however, the method can also be realised with other sensors. For example, one could consider use of cameras (which are becoming ubiquitous in mobile phones and handhelds) and vision techniques to obtain spatial references between devices.

The concrete implementation we have presented uses ultrasound, for peer-to-peer spatial sensing, and for out-of-band message transfer as part of a key verification protocol. We have provided a comprehensive threat analysis for ultrasonic ranging and contributed a novel coding technique that allows a sender to guarantee that a message was sent from a particular range. This technique can thus be used to to construct a spatially-authentic channel from sender to receiver.

Our protocol implementation is embedded in a spatial sensing scheme that more generally provides devices with accurate relative positions of peers discovered within interaction range. The method further involves a user interaction model based on visualisation of relative device positions, integrated in the user interface for direct manipulation. The method as presented relies on spatial sensors, however, the sensors are not specific to the purpose of providing security but have broader use for support of spatial interaction and services. Cameras and various other sensors are already ubiquitous in mobile devices, and given the general utility of ultrasonic transducers for ranging tasks it is easily perceivable that these will become commonplace as well.

As a final note it has to be stressed that the presented approach fundamentally differs from proximity-based methods such as near-field communication (NFC). Any proximity-based method that relies on a *quantitative* property of the out-of-band channel such as radio signal strength is open to attack from further afield — for example to attack NFC by increasing communication range with more powerful senders and/or more sensitive receivers. In contrast, our method exploits the *qualitative* out-of-band properties of ultrasound: that it is blocked by solid materials and that angle of arrival can not be faked.

The complete source code is available under an open source license at `http://ubicomp.lancs.ac.uk/relate/`.

## Acknowledgements

We acknowledge support for the presented research by the Commission of the European Union under contracts 013790 "RELATE" and the FP6 Marie Curie Intra-European Fellowship program contract MEIF-CT-2006-042194 “CAPER”, and by the Engineering and Physical Sciences Research Council in the UK under grant GR/S77097/01.

## References

1. Gehrmann, C., Mitchell, C.J., Nyberg, K.: Manual authentication for wireless devices. RSA Cryptobytes **7**(1) (2004) 29–37
2. Stajano, F., Anderson, R.: The resurrecting duckling: Security issues for ad-hoc wireless networks. In: Proc. 7th Int. Workshop on Security Protocols, Springer-Verlag (April 1999) 172–194
3. Balfanz, D., Smetters, D.K., Stewart, P., Wong, H.C.: Talking to strangers: Authentication in ad-hoc wireless networks. In: Proc. NDSS'02, The Internet Society (February 2002)
4. Kindberg, T., Zhang, K., Shankar, N.: Context authentication using constrained channels. In: Proc. WMCSA 2002, IEEE CS Press (June 2002) 14–21
5. Kindberg, T., Zhang, K.: Secure spontaneous devices association. In: Proc. UbiComp 2003, Springer-Verlag (October 2003) 124–131
6. Kindberg, T., Zhang, K.: Validating and securing spontaneous associations between wireless devices. In: Proc. ISC'03, Springer-Verlag (October 2003) 44–53
7. McCune, J.M., Perrig, A., Reiter, M.K.: Seeing-is-believing: Using camera phones for human-verifiable authentication. In: Proc. IEEE Symp. on Security and Privacy, IEEE CS Press (May 2005) 110–124
8. Goodrich, M.T., Sirivianos, M., Solis, J., Tsudik, G., Uzun, E.: Loud and clear: Human verifiable authentication based on audio. In: Proc. ICDCS 2006, IEEE CS Press (July 2006) 10
9. Clulow, J., Hancke, G.P., Kuhn, M.G., Moore, T.: So near and yet so far: Distance-bounding attacks in wireless networks. In: Proc. ESAS 2006, Springer-Verlag (2006) 83–97
10. Hazas, M., Kray, C., Gellersen, H., Agbota, H., Kortuem, G., Krohn, A.: A relative positioning system for co-located mobile devices. In: Proc. MobiSys 2005, ACM Press (June 2005) 177–190
11. Kortuem, G., Kray, C., Gellersen, H.: Sensing and visualizing spatial relations of mobile devices. In: Proc. UIST 2005, ACM Press (October 2005) 93–102
12. Guinard, D., Streng, S., Gellersen, H.: Relategateways: A user interface for spontaneous mobile interaction with pervasive services. In: CHI 2007 Workshop on Mobile Spatial Interaction. (2007)
13. Shaked, Y., Wool, A.: Cracking the Bluetooth PIN. In: Proc. MobiSys 2005, ACM Press (June 2005) 39–50
14. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. on Information Theory **IT-22**(6) (1976) 644–654
15. Rivest, R.L., Shamir, A.: How to expose an eavesdropper. Commununications of ACM **27**(4) (1984) 393–394
16. Ferguson, N., Schneier, B.: Practical Cryptography. Wiley Publishing (2003)
17. Eronen, P., Tschofenig, H.: RFC4279: Pre-shared key ciphersuites for transport layer security (TLS) (December 2005)