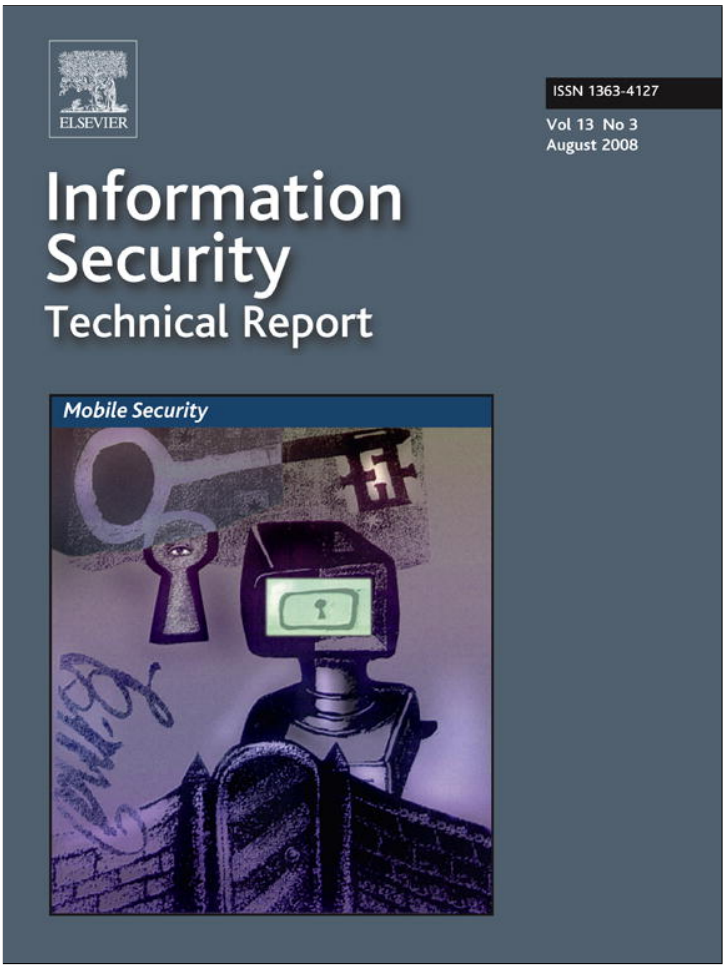


INFORMATION SECURITY TECHNICAL REPORT 13 (2008) 136–150

available at www.sciencedirect.com

ScienceDirect

www.compseconline.com/publications/prodinf.htm

Information Security Technical Report

Spontaneous mobile device authentication based on sensor data☆

Rene Mayrhofer[a,*], *Hans Gellersen*[b]

[a]*Faculty of Computer Science, University of Vienna, Dr.-Karl-Lueger-Ring 1, A-1010 Vienna, Austria*
[b]*Computer Science Department, Lancaster University, Infolab21, South Drive, Lancaster LA1 4WA, UK*

Keywords:
Spontaneous interaction
Authentication
Device pairing
Ubiquitous computing
Mobile computing

ABSTRACT

Small, mobile devices or infrastructure devices without user interfaces, such as Bluetooth headsets, wireless LAN access points, or printers, often need to communicate securely over wireless networks. Active attacks can only be prevented by authenticating wireless communication, which is problematic when devices do not have any a priori information about each other. In this article, we describe three different authentication methods for device-to-device authentication based on sensor data from various physical out-of-band channels: shaking devices together, authentication based on spatial reference, and transmission via visible laser.

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. That is, the user must be in the loop, for example to enter a shared secret such as a PIN code into both devices.

A challenge is to find mechanisms for users to pair devices that are not only secure but also scale well for use in ubiquitous computing. Specific challenges are that devices will, in many cases, be too small to reasonably include key pads and displays, or that interaction should happen at a distance, as well as that required user attention must be minimal to be acceptable for spontaneous and short-lived interactions. Securing wireless communication during the interaction must be unobtrusive and implicit; additional steps required "just for security" will most likely be unacceptable.

Pairing of a mobile phone with a headset for interaction over a wireless channel is a familiar example: we would like to not only achieve such interaction in a spontaneous manner (i.e. not requiring pre-configuration of phone and headset for each other) but also ensure that it is secure. Another example

☆ This article presents a summary and extension of four previous conference papers (Mayrhofer and Gellersen, May 2007; Mayrhofer et al., 2007; Mayrhofer and Welch, 2007; Mayrhofer, 2006).

* *Corresponding author.*
E-mail address: rene.mayrhofer@univie.ac.at (R. Mayrhofer).

1363-4127/$ – see front matter
doi:10.1016/j.istr.2008.10.005

is granting (temporary or permanent) access to a wireless LAN, a printer, or a projector to new mobile devices.

The wireless communication channel between these devices is susceptible to attacks ranging from eavesdropping to man-in-the-middle (MITM). If an attacker were successful in establishing themselves between, for example, phone and headset, during the pairing process, then they would obtain complete control over all phone calls. To safeguard against such attacks, a so-called *out-of-band channel* is used during pairing in order to authenticate communication over the primary channel. The out-of-band channel must be limited such that it is user-controllable that only the intended devices can communicate over it for the purposes of authentication. Note that authentication and the subsequent pairing can be anonymous or ''ephemeral'' (Hoepman, 2004), i.e. based on information only shared over the out-of-band-channel rather than actual device identities.

In this article we describe and compare three methods for device-to-device authentication:

The first method, *Shake Well Before Use*, is based on shared movement patterns which a user can simply generate by shaking devices together (Mayrhofer and Gellersen, May 2007). Using embedded accelerometers, devices can recognise correlation of their movement and use movement patterns for authentication. From a user perspective, jointly shaking is a simple technique for associating devices (Holmquist et al., 2001). In this method, it simultaneously serves as out-of-band mechanism.

The second method uses *Spatial References* to establish and authenticate interaction between a pair of devices (Mayrhofer et al., September 2007). 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. A concrete implementation uses a combination of radio frequency (RF) and ultrasonic (US) communication for measurement of spatial relationships. As ultrasonic ranging is susceptible to certain attack scenarios, we further present a novel coding technique for spatially dependent message transfer over an ultrasonic channel.

The third method uses a visual laser for selecting a remote, typically stationary and larger device and for transmitting cryptographic material over the same channel (Mayrhofer and Welch, 2007).

All methods combine cryptographic primitives with sensor data analysis to establish secure wireless channels by creating authenticated secret keys. If device authentication is completed successfully, then A and B can use the created 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 (Eronen and Tschofenig, 2005). 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.

In addition to these specific device authentication methods, we summarize the previously introduced concept of a *context authentication proxy* to allow authentication between devices that are physically separated. A context authentication proxy is pre-authenticated to one of the devices and can use any context- or sensor-based device authentication method with the second device, for example one of the methods presented in more detail.

For the formal descriptions of our protocols, we use the following notation: $c = E(K, m)$ describes the encryption of plain text m under key K with a symmetric cipher, $m = D(K, c)$ the corresponding decryption, $H(m)$ describes the hashing of message m with some secure hash, while HMAC_K refers to an HMAC (Krawczyk et al., 1997) with key K. $m|n$ means the concatenation of strings m and n, and notation $M[a{:}b]$ is used to describe the substring of a message M starting at bit a and ending at bit b. The symbol $\oplus$ describes bit-wise XOR and $|S|$ the number of elements in a set S. If a message $\mathbf{M}$ is transmitted over an insecure channel, we denote the received message $\tilde{\mathbf{M}}$ to point out that it may have been modified in transit, by noise or attack. Bold font signifies that the respective message is actually being transmitted over some channel and that it may therefore potentially be eavesdropped on. C refers to some publicly known constant. We use AES as a block cipher for E and D and $\mathrm{SHA}_{\mathrm{DBL}}$-256 as a secure hash for H, which is a double execution of the standard SHA-256 message digest to safeguard against length extension and partial-message collision attacks (Ferguson and Schneier, 2003) and is defined as $\mathrm{SHA}_{\mathrm{DBL}}\text{-}256 = \mathrm{SHA\text{-}256}((\mathrm{SHA\text{-}256}(m))|m)$.

2. Related work

First concepts on secure device pairing suggested direct electrical contact (Stajano and Anderson, 1999), while other suggestions to implement an out-of-band channel include a ''physical interlock'' and the ''Harmony'' protocol (Kindberg et al., 2005), ultrasound (Kindberg and Zhang, 2003b), visual markers and cameras (McCune et al., 2005), audio messages (Goodrich et al., 2006), the GSM short message service (SMS) (Nicholson et al., 2006), key comparison, distance bounding and integrity codes (Čagalj et al., 2006), or manual input (Gehrmann et al., 2004; Hoepman, 2004). The DH–DB protocol proposed by Čagalj et al. (2006) might also be applicable to an interactive challenge-response scheme based on sensor data such as accelerometer data. These approaches, with the exception of using camera phones, have in common that they scale poorly from a user point of view. That is, they tend to be obtrusive and require the user's attention. In all methods presented in this article, selection of remote devices to interact with is combined with implicit authentication and thus suggested to scale better in terms of user attention.

The idea of shaking two (or multiple) devices together to pair them has first been described as ''Smart-Its Friends'' (Holmquist et al., 2001). Shaking of devices for the purpose of using shared movement for authentication was originally introduced by us in a conference paper on which this article expands (Mayrhofer and Gellersen, May 2007) but has since been explored also by others. Kirovski et al. (2007) present a method in which devices that are moved together to perform joint fuzzy hashing, similar in general design to the second of

our two protocols, but with differences in how key material is extracted from acceleration time series. Bichler et al. (2007) likewise present an approach in which acceleration data is used for key generation and in this sense also more closely related to our second protocol, however, using acceleration features in the time domain (whereas both our protocols are based on features in the frequency domain). Beyond these recent efforts, we contribute an implementation of two alternatives to understand trade-offs in authentication based on accelerometer data.

Our concrete implementation of authentication using spatial references is based on the use of ultrasound as out-of-band channel. Kindberg and Zhang (2003b) have before us proposed the use of US alongside an RF wireless channel in a protocol for validation and securing of spontaneous interaction. However, the protocol design does not consider potential attacks on the ultrasonic channel. 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.

For authentication using visible lasers, Ringwald was among the first to present a working prototype for device-to-device interaction using lasers (Ringwald, 2002), followed by Patel and Abowd (2003). Both used relatively simple ways of modulating a laser diode and reconstructing the signal at the receiver end, whilst using the laser as an out-of-band method for initiating wireless communication by transmitting the device network address, although without considering security of the interaction. Kindberg and Zhang (2003a) previously suggested the transmission of secret keys via modulated laser light, under the assumption that the laser emits no light except onto the receiving sensor. In contrast, we do not assume the laser transmission to be confidential.

3. Threat analysis

We generally assume the users' personal devices and any remote devices selected by users for spontaneous interaction to be secure and trustworthy – if only for the specific spontaneous interaction the user is interested in. Cases where information sent to it by the user is forwarded after successful authentication are out of the scope of this article. We are mainly concerned with direct attacks on the interaction between these legitimate devices.

3.1. Wireless channels

The main threats to address for spontaneous authentication between devices are the so-called "man-in-the-middle" (MITM) attacks on wireless networks, which we will refer to as the RF channel. Under the realistic assumption that wireless networks are open to any manipulation, including eavesdropping, injection, modification, delaying, and replay of packets, such attacks are simple to perform and nearly impossible to detect on the wireless channel itself. Attacks on any of the wireless channels are the most dangerous, because they can be carried out inconspicuously (see, e.g. Shaked and Wool, 2005). 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. We explicitly assume an attacker ("Eve") to be capable of gaining complete control over all wireless communication channels and therefore perform active MITM attacks. 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).

To prevent this threat, out-of-band channels are required and consequently employed by all three methods presented in this article. Attacks on wireless channels can also influence such out-of-band channels when co-ordination over RF is required, for example for synchronising ultrasonic pulse transmission. Denial-of-service attacks are generally simpler to perform – both on wireless and out-of-band channels – and out of scope of this article.

3.2. Attacks on out-of-band channels

3.2.1. Channel I: shared movement

Because acceleration is a local physical phenomenon measurable by embedded accelerometers, it seems nearly impossible to manipulate remotely. A potential attacker therefore has two options: (I.a) to generate accelerations by shaking their own device sufficiently similar to the target device (which seems practically infeasible); or (I.b) to estimate the accelerations experienced locally by the target device with sufficient accuracy, e.g. using video analysis methods (the success probability depends on the uncertainty of the attacker about the specific movement).

3.2.2. Channel II: ultrasonic sensing

Control over the ultrasonic (US) channel is also assumed to be limited. First, for attacks on this channel, an attacker needs to be physically present in the same room (ultrasound is effectively blocked by solid materials such as walls, doors, and windows). Second, although eavesdropping is easily possible, injecting ultrasonic 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.

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 ultrasonic 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. A detailed analysis and techniques to

prevent specific attacks has been presented separately (Mayrhofer and Gellersen, March 2007). Summarising, angle of arrival can be trusted, distance measurements cannot by themselves, and therefore an attacker can: (II.a) make specific devices disappear; or (II.b) fake distance measurements.

3.2.3. *Channel III: laser*

Previous work assumed a modulated laser beam to be confidential from attackers (Kindberg and Zhang, 2003a). However, this assumption does not seem valid, considering that a typical laser beam is observable both at the sender (light emitted by the laser diode can be seen from almost any angle within its front hemisphere even if the majority is emitted along the primary axis) and the target (laser light is reflected as scattered light from most surfaces, including photovoltaic elements suitable for use as receivers). With high-speed cameras, it seems possible to capture the modulated signals with reasonable accuracy. We therefore do not assume a modulated laser channel to be confidential.

It is also questionable whether this channel can be assumed to be authentic, because most photovoltaic elements suitable for receivers cannot distinguish angle of arrival and thus not between different senders. It is possible for an attacker Eve to point their laser beam on the receiver and therefore inject their own messages into the out-of-band channel. However, any such message injection is likely to modify the original messages sent by the user's personal device. Therefore, we can only assume that E cannot easily block or completely change the information transmitted via a modulated laser beam without previous knowledge of the message contents.

3.3. *Application-level threats*

We also need to consider attacks not aiming at the lower-level channels but directly at the ''application level''. The common threat on this level 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. 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 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 in the case of spatial reference, or e.g. by making the ''target'' for laser authentication obvious and preventing the ''overlay'' of attacking photosensors again by physical means. For authentication by shaking, application-level threats do not seem to be relevant.

4. Shake-well-before-use: authentication based on accelerometer data

Our first method for device authentication uses shared movement in terms of shaking small, mobile devices such as mobile phones and Bluetooth headsets together in one hand. Local device accelerations are measured with embedded accelerometers. This method has been presented in more detail in a previous conference paper (Mayrhofer and Gellersen, May 2007).

Fig. 1 shows our architecture for authenticating device pairings with shaking patterns. Two cryptographic protocols make use of the same three pre-processing tasks 1–3. They are executed locally on each device and result in ''active'' time series segments of equidistant samples. Our two protocols differ in tasks 4 and 5, which can both be interactive, i.e. communicate with the remote device to which the pairing is in process.

For protocol 1, tasks 4.1 and 5.1 are actually executed in parallel: after generating a secret key with standard Diffie–Hellman (DH) key agreement (which is the first phase of task 5.1), the devices exchange their time series segments via an interlock protocol. Then they compare their locally generated segment with the one received from the remote device to check if they are similar enough. If they pass this check, the second phase of task 5.1 derives the secret session key that will be used for consecutive secure communication. This design is conservative from a security point of view and, due to the non-interactive feature extraction and comparison, allows the devices to use different means of verification. The disadvantage of splitting task 5.1 into two phases is potentially a larger delay for authentication, and the disadvantage of using DH is higher computational load.

Protocol 2 executes its tasks 4.2 and 5.2 in order: discrete (in contrast to the real-valued samples) feature vectors are extracted in task 4.2, which act as input to the interactive key agreement in task 5.2. This is an iterative process. In each time step, feature vectors generated by 4.2 are checked for matches in task 5.2. After sufficient iterations, a secret shared key can be generated out of the collected matching feature vectors in task 5.2. This design has the advantages of more dynamic key agreement, with devices being able to ''tune into'' other device's key streams, and of being less computationally expensive. On the other hand, it does not provide forward secrecy and protection against offline attacks as protocol 1 does, and is more unconventional and thus less well studied from a security point of view.

4.1. *Pre-processing of accelerometer data*

The three pre-processing tasks, executed as consecutive steps, are used to sample and segment the sensor data so that feature extraction can build on normalized time series.

4.1.1. *Task 1: sensor data acquisition*

This first task is conceptually straight forward, but requires careful implementation. Sensor data is assumed to be available in the form of time series of acceleration values in all three dimensions, sampled at equidistant time steps. These must be taken locally and not be communicated wirelessly – for security purposes, it is critical not to leak any of this raw data, which can be difficult considering the possibility of powerful side-channel attacks (see e.g. Batina et al., 2005). Our practical experience shows a sample rate between 100 and 600 Hz to be appropriate.

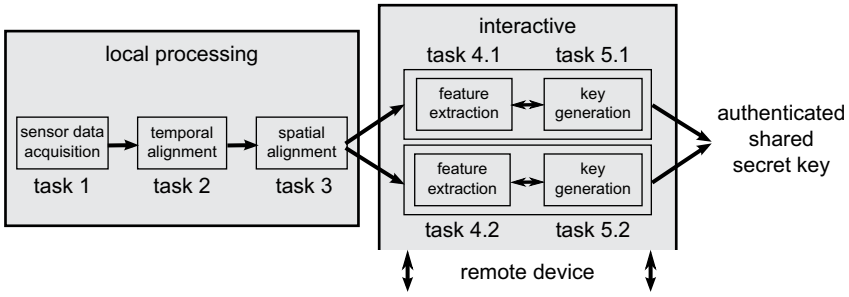


Fig. 1 – Architecture for both authentication protocols based on shared movement.

4.1.2. *Task 2: temporal alignment*

As the two devices sample accelerometer time series independently in task 1, we require temporal synchronization for comparison. Temporal alignment requires *triggering* the authentication procedure and *synchronising* the starting points for time series comparison. For both triggering and synchronization, we detect motion and align those parts of the time series where shaking is detected, which we call *active segments*, by their start times. Segments are considered active when the variance of a sliding window exceeds a threshold. Practical experiments show good results at a sample rate between $f = [128;512]$ Hz with a sliding window of $v = f/2$ samples, i.e. 1/2 s, and a normalised variance threshold around $T_\sigma = 0.045$. The end of an active segment can be determined when the motion subsides (this approach is used in the second protocol) or be defined by a fixed segment length (we use 3 s in the first protocol).

4.1.3. *Task 3: spatial alignment*

Shaking is inherently a three-dimensional movement. In addition to the need to capture all three dimensions, the alignment between the two devices is unknown. This means that the three dimensions recorded by the two devices will not be aligned, which is a hard problem in itself. We reduce the three dimensions to a single: by taking only the magnitude over all normalized dimensions, i.e. the length of the vector, we solve the alignment problem.

The result of these steps is that, when shaken together, both devices will extract *active segments* of one-dimensional acceleration magnitude vectors. Even without synchronised clocks, the start times of these independent time series are typically synchronised within a few samples.

4.2. *Feature extraction*

Two devices that are shaken together will experience similar, but not exactly the same movement patterns. Even assuming noise-free sampling of accelerations, the two accelerometers must have physically separate centers. Whenever rotation is part of the movement, these separate centers will necessarily experience different accelerations, thus causing different sensor time series even if the devices remain fixed in relation to each other. The problem of verifying that two devices are shaken, or more generally, moved together therefore becomes a classification problem.

In deciding if time series are similar enough for authentication, the aim of the feature extraction task is twofold: (a) to extract feature values that are robust to small variations in the shaking patterns and to sampling noise and (b) to extract a sufficiently large feature vector for use in the authentication protocol. In our approach, the feature vector will be used to authenticate a key or to directly generate a key, and thus it needs to be of high entropy from an attacker's point of view.

4.2.1. *Coherence*

Coherence (cf. Lester et al., 2004) is approximated by the magnitude squared coherence (MSC) as

$$C_{xy}(f) = \frac{P_{xy}(f)}{P_{xx}(f) \cdot P_{yy}(f)}$$

with (cross-) power spectra

$$P_{xy}(f) = \frac{1}{n} \sum_{k=0}^{n-1} x_k(f) \cdot \bar{y}_k(f)$$

computed over FFT coefficients $x_k(f) = \mathrm{FFT}(a_k(t) \cdot h(t))$ and $y_k(f) = \mathrm{FFT}(b_k(t) \cdot h(t))$ using the standard von-Hann window $h(t) = (1 - \cos(2\pi t/w))/2$. That is, it is computed as the power spectrum correlation between two signals split into n (optionally overlapping) averaged slices a_k and b_k of the signals a and b, respectively, normalized by the signal power spectra. Because the significance of coherence values depends on the number of averaged slices n – the more slices, the lower the coherence values are for the same signals – we reduce longer time series to a maximum length of 3 s. This is a compromise between sufficient variability for robust classification and quick user interaction. The final value is computed simply by averaging up to a cut-off frequency $f_{\max}$:

$$C_{xy} = \frac{1}{f_{\max}} \int_0^{f_{\max}} C_{xy}(f) \mathrm{d}f$$

With this heuristic, we threshold C_{xy} to create a binary decision of similarity for our authentication protocol. Our experiments have shown that, with a sampling rate of $r = 256$ Hz and windows of $w = 256$ samples with an overlap of 7/8 and a cut-off frequency of $f_{\max} = 40$ Hz, coherence provides good distinction between two devices being shaken together from two devices being shaken independently.

4.2.2. *Quantized FFT coefficients*

Keys must be bit-for-bit equal, and thus be based on discrete instead of continuous values. By retaining basic

features of the coherence measure and condensing them into discrete feature vectors, we can use those for a different way of comparing two accelerometer time series. We compared four variants of FFT-based feature vectors: linearly or exponentially quantized coefficients used either directly of added pairwise. Our experiments have shown that pairwise added, exponentially quantized FFT coefficients performed best, as also suggested by Huynh and Schiele (2005). When aiming for equivalence of feature vectors, there is, however, an additional complication: small differences of values near the boundaries of quantisation bands can lead to different feature values, although the FFT coefficients are only marginally different. Our solution is to quantise each FFT vector into multiple *candidate* feature vectors with different offsets. These offsets range from 0 to the value of the smallest quantisation band. The similarity criteria in this case is simply the percentage of matching candidate feature vectors out of all vectors sent to another device. Thresholding this percentage produces a binary decision for the authentication protocol. We achieved best results for distinguishing shaking together from shaking independently with $b=6$ exponentially scaled bands for quantization, $k=4$ candidates, and a cut-off frequency of $f_{\max}=20$ Hz at a sampling rate of $r=512$ Hz with FFT windows of $w=512$ samples, overlapping by 50%.

4.3. Authentication protocols

The two feature vectors generated in task 4 constitute, if equivalent, a shared secret password. This shared string is not directly suitable to act as a secret key for cryptographic primitives, because it is neither of defined length (e.g. 128 bits) nor distributed uniformly. But it is possible to create a cryptographically secure secret key via interactive protocols, authenticated by the feature vectors.

4.3.1. Protocol 1: Diffie–Hellman and interlock*

Fig. 2 shows our first authentication protocol, which is based on a standard Diffie–Hellman (DH) key agreement (introduced in their seminal article (Diffie and Hellman, 1976)) followed by an exchange of the condensed time series and comparison locally at each device.

Using DH key agreement, devices A and B generate two – supposedly – shared keys K^{Auth} and K^{Sess}, where it is impossible to infer one from the other (under the assumption that the hash function does not allow to find a pre-image). Creating two keys, one for authentication, one as session key, provides forward secrecy. Because DH is susceptible to MITM, the devices need to verify that their keys are equivalent. The unique key property of DH guarantees with a very high probability, that, if $K_a^{\text{Auth}}=K_b^{\text{Auth}}$, there can be no attacker E with $K_{e1}^{\text{Auth}}=K_a^{\text{Auth}}$ and $K_{e2}^{\text{Auth}}=K_b^{\text{Auth}}$, and subsequently, no $K_{e1}^{\text{Sess}}=K_a^{\text{Sess}}$ and $K_{e2}^{\text{Sess}}=K_b^{\text{Sess}}$.

This verification is done with an extended *interlock* protocol. Interlock (Rivest and Shamir, 1984) is not used widely, but is an efficient (in terms of message length) method to verify that two parties share the same key. By using this key as an input to a block cipher and splitting packets in halves, a MITM can only decrypt these packets after having received both halves. The interlock protocol then demands that A and B will only send their second halves after they have received the first halves from the respective other side. This has the effect that both sides must commit themselves to their values, by sending the first halves of the encrypted blocks, before they can receive, and subsequently decrypt, the other side's message. Thus, interlock can be seen as a commitment scheme (see, e.g., Vaudenay, 2005 for a definition) based on block ciphers. An attacker E is now left with only two options: either to forward the original packets, or to create packets on its own. In the former case, A and B will be unable to decrypt the messages properly, because they do not share the same key. In the latter case, E must guess the contents of the

A	Message	B
	Choose DH key	
$x \in \{1 \dots q-1\}$ $K_a^{Sess} :=$ $\quad H(\tilde{\mathbf{Y}}^x)$ $K_a^{Auth} :=$ $\quad H(\tilde{\mathbf{Y}}^x \vert C)$	$\mathbf{X} := g^x$ $\longrightarrow$ $\mathbf{Y} := g^y$ $\longleftarrow$	$y \in \{1 \dots q-1\}$ $K_b^{Sess} :=$ $\quad H(\tilde{\mathbf{X}}^y)$ $K_b^{Auth} :=$ $\quad H(\tilde{\mathbf{X}}^y \vert C)$
	Split and encrypt signal	
$(i=0:n-1, \quad j=1:n)$: $a_i := a[128i:128i+127]$ $c_0 \in \{0,\dots,2^{128}-1\}$ $c_j := E(K_a^{Auth}, c_{j-1} \oplus a_{j-1})$	$\mathbf{A}_1 = c_0[0:63] \vert \dots \vert c_n[0:63]$ $\longrightarrow$ $\mathbf{B}_1 = d_0[0:63] \vert \dots \vert d_m[0:63]$ $\longleftarrow$	$(i=0:m-1, \quad j=1:m)$: $b_i := b[128i:128i+127]$ $d_0 \in \{0,\dots,2^{128}-1\}$ $d_j := E(K_b^{Auth}, d_{j-1} \oplus b_{j-1})$
	Interlock and reassemable	
$(i=0:m-1)$: $\tilde{d}_i := \tilde{\mathbf{B}}_1[128i:128i+63] \vert \vert$ $\quad \tilde{\mathbf{B}}_2[128i:128i+63]$ $\tilde{b}_i := D(K_a^{Auth}, \tilde{d}_{i+1}) \oplus \tilde{d}_i$ $\tilde{b} := \tilde{b}_0 \vert \dots \vert \tilde{b}_m$	$\mathbf{A}_2 = c_0[64:127] \vert \dots \vert c_n[64:127]$ $\longrightarrow$ $\mathbf{B}_2 = d_0[64:127] \vert \dots \vert d_n[64:127]$ $\longleftarrow$	$(i=0:n-1)$: $\tilde{c}_i := \tilde{\mathbf{A}}_1[128i:128i+63] \vert \vert$ $\quad \tilde{\mathbf{A}}_2[128i:128i+63]$ $\tilde{a}_i := D(K_b^{Auth}, \tilde{c}_{i+1}) \oplus \tilde{c}_i$ $\tilde{a} := \tilde{a}_0 \vert \dots \vert \tilde{a}_n$
	Compare	
if $a \sim \tilde{b}$ then $K := K_a^{Sess}$ else $K := null$		if $b \sim \tilde{a}$ then $K := K_b^{Sess}$ else $K := null$

Fig. 2 – Shake-well-before-use protocol 1: Diffie–Hellman key agreement followed by exchange of the complete time series via interlock*.

messages, and encrypt them with the appropriate keys, before it has access to the actual messages. When the messages sent by A and B have an entropy of e bits,E is left with a single 2^{-e} chance of remaining undetected.

The original version of interlock is suitable for messages with the size of the cipher block length. Because in our case the vectors of the accelerometer sensor data, condensed into a time series of magnitudes, have arbitrary length, we introduce a slightly extended protocol that we call *interlock**. In this variant, A and B encrypt their complete messages, i.e. the (zero-padded) vectors a and b with lengths of n and m blocks, respectively, with any of the well-known block cipher modes. For our motion authentication protocol, we simply use the cipher block chaining (CBC) mode with a random initialization vector (IV). The resulting cipher texts c and d with lengths of $n+1$ and $m+1$ blocks are then split into two messages by concatenating the first halves of all cipher blocks into the first messages A_1 and B_1 and the second halves of all cipher blocks into the second messages A_2 and B_2. This ensures that E cannot decrypt any of the blocks, and can therefore not even learn parts of the plain text messages.

After exchanging their messages a and b, A and B verify that $a \sim b$, that is, that they are similar enough under their chosen criteria. We use coherence as described in Section 4.2, but other suitable features can be used without changes to the protocol. Because of this possibility, we do not try to minimize the message lengths as, e.g. suggested by Lester et al. (2004). In fact, A and B could use completely different similarity criteria, and could still authenticate using the same protocol. This is important for practical implementations, because different generations of devices will need to be compatible with each other.

4.3.2. Protocol 2: candidate key protocol

In our second protocol, which we call the *candidate key protocol* (CKP), the shared secret key is generated from sensor data instead of by DH. As depicted in Fig. 3, feature vectors v are hashed to generate *candidate key parts* h. If the feature extraction task produces multiple "parallel" feature vectors v^i for each time window, as suggested above in Section 4.2, then these yield multiple candidate key parts h^i. The one-way hashes are a simple way to communicate that a device has generated a certain feature vector without revealing it. To make dictionary attacks harder, we use the standard method of prepending random salt values s before hashing. When B receives such candidate key parts from A, it can use its own history of recently generated feature vectors LH to check for equals. When B has generated the same feature vector, it is stored in a list of *matching key parts* MC specific to each communication partner. As soon as enough entropy has been collected in this list, B concatenates all feature vectors, appends C, hashes the resulting string, and sends a *candidate key* K to A. If no messages have been lost in transit, A should be able to generate a key with the same hash, and thus the same secret key, which it acknowledges to B. If messages have been lost, A can simply ignore a candidate key and create its own later on.

CKP is again a general protocol and can be used with any feature vectors. Here we apply it to quantized FFT coefficients, which work well for accelerometer data. A more thorough analysis of CKP itself has been provided separately (Mayrhofer, 2007).

4.4. Experimental evaluation

For quantitative evaluation, extensive data sets were sampled in two user experiments (see Fig. 4 for the sensing devices). In the first one, two devices were shaken together in one hand, in the second one, pairs of users tried to shake one device each as similarly as possible. Fig. 5 shows the trade-off between

A	Message	**B**
Initialize $r_a := 0$		Initialize $r_b := 0$
$s_a^{r_a} \in \{0,\ldots,2^{128}-1\}$ **For each vector** v_a^i: $h_a^i := H(s_a \vert v_a^i)$ add $< r_a, v_a^i >$ to LH_a $r_a := r_a + 1$	$\mathbf{C}_a^{r_a} :=< r_a, s_a^{r_a}, h_a^1, h_a^2, \ldots >$ $\longrightarrow$ $\mathbf{C}_b^{r_b} :=< r_b, s_b^{r_b}, h_b^1, h_b^2, \ldots >$ $\longleftarrow$	$s_b^{r_b} \in \{0,\ldots,2^{128}-1\}$ **For each vector** v_b^j: $h_b^j := H(s_b \vert v_b^j)$ add $< r_b, v_b^j >$ to LH_b $r_b := r_b + 1$
For each hash $\tilde{h}_b^j$: if $\exists < \hat{r}_a, \hat{v}_a^i > \in LH_a$ s.t. $\tilde{h}_b^j = H(\tilde{s}_b^{r_b} \vert \hat{v}_a^i)$ then add $< \hat{r}_a, \hat{v}_a^i >$ to $MC_{a,b}$		**For each hash** $\tilde{h}_a^i$: if $\exists < \hat{r}_b, \hat{v}_b^j > \in LH_b$ s.t. $\tilde{h}_a^i = H(\tilde{s}_a^{r_a} \vert \hat{v}_b^j)$ then add $< \hat{r}_b, \hat{v}_b^j >$ to $MC_{b,a}$
if $\vert MC_{a,b} \vert >= N_a$ then $kh_a := H(s_a^{r_a} \vert last_{N_a}(MC_{a,b}))$ and $k_a := H(s_a^{r_a} \vert last_{N_a}(MC_{a,b}) \vert C)$	$\mathbf{K}_a :=< N_a, s_a^{r_a}, kh_a >$ $\longrightarrow$ $\mathbf{K}_b :=< N_b, s_b^{r_b}, kh_b >$ $\longleftarrow$	if $\vert MC_{b,a} \vert >= N_b$ then $kh_b := H(s_b^{r_b} \vert last_{N_b}(MC_{b,a}))$ and $k_b := H(s_b^{r_b} \vert last_{N_b}(MC_{b,a}) \vert C)$
if $\exists \hat{c}_a \subseteq MC_{a,b}$ s.t. $H(\tilde{s}_b^{r_b} \vert \hat{c}_a) = \widetilde{kh}_b$ then $k'_b := H(\tilde{s}_b^{r_b} \vert \hat{c}_a \vert C)$	ACK kh_b $\longrightarrow$ ACK kh_a $\longleftarrow$	if $\exists \hat{c}_b \subseteq MC_{b,a}$ s.t. $H(\tilde{s}_a^{r_a} \vert \hat{c}_b) = \widetilde{kh}_a$ then $k'_a := H(\tilde{s}_a^{r_a} \vert \hat{c}_b \vert C)$
if k_a is not set $k := k'_b$ if k'_b is not set $k := k_a$ if $kh_a = \widetilde{kh}_b$ then $k := k_a$ else $k := k_a \oplus k'_b$		if k_b is not set $k := k'_a$ if k'_a is not set $k := k_b$ if $kh_b = \widetilde{kh}_a$ then $k := k_b$ else $k := k_b \oplus k'_a$

Fig. 3 – Shake-well-before-use protocol 2: candidate key protocol for directly creating a secret key from common feature vector hashes.

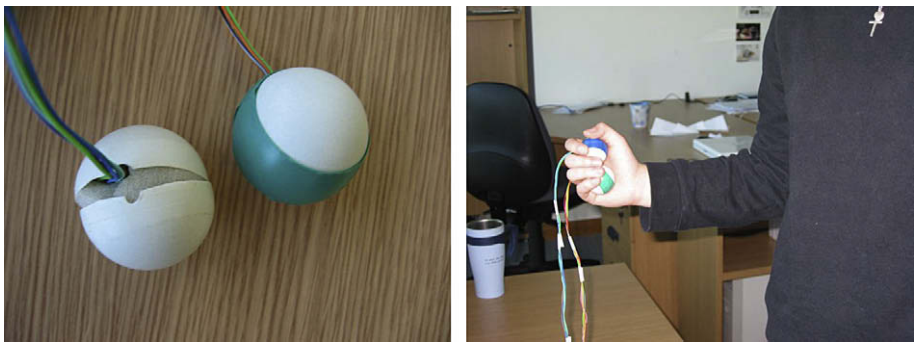


Fig. 4 – Experimental setup: devices with accelerometers and subject during data collection.

false positives (i.e. devices shaken independently by two people, but authentication would succeed) and false negatives (i.e. devices shaken together, but authentication would fail), depending on the thresholds. From a security point of view, we obviously prefer to restrict the number of false positives to zero, which can be achieved with false negatives rates of 10.24% and 11.96%, respectively. Because the feedback of a failed authentication is immediate and users just need to shake the devices again, these false negatives rates seem acceptable.

4.5. *Security analysis*

Both protocols depend on the entropy of active segments used for authentication from an attacker's point of view (threat I.b). The same data sets were also used to estimate the entropy of feature vectors used for our second protocol. If we assume an attacker to know which device, person, and hand are involved in a protocol run, we currently assume to generate around 7 bits entropy per second against offline analysis using our second protocol. Due to its design, the second protocol is limited to this level of security with a direct trade-off between speed of authentication and security: the longer users shake, the more entropy they create, with a current estimate of roughly 20 s of shaking for 128 bits.

In contrast, due to the use of Diffie–Hellman key agreement and interlock, the security level of our first protocol against offline attacks is only limited by DH and not by this entropy. Eve has only a single chance during an active MITM attack (to estimate the active segments transmitted by both devices with sufficient accuracy) to remain undetected. Although we cannot currently quantify the security level against such unlikely online attacks, the security level of protocol 1 against offline attacks is 128 bits even after only 3 s of shaking (assuming DH to be secure).

By introducing two protocols with different designs, application developers can decide on this well-known trade-off between security and performance according to their requirements. Protocol 2 offers benefits for devices with limited resources, large wireless networks, and quick interaction and supports group authentication, while we recommend using protocol 1 for higher security demands.

5. Security by spatial reference

Central to our second method is the concept of *Spatial References* (Mayrhofer et al., 2007). 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

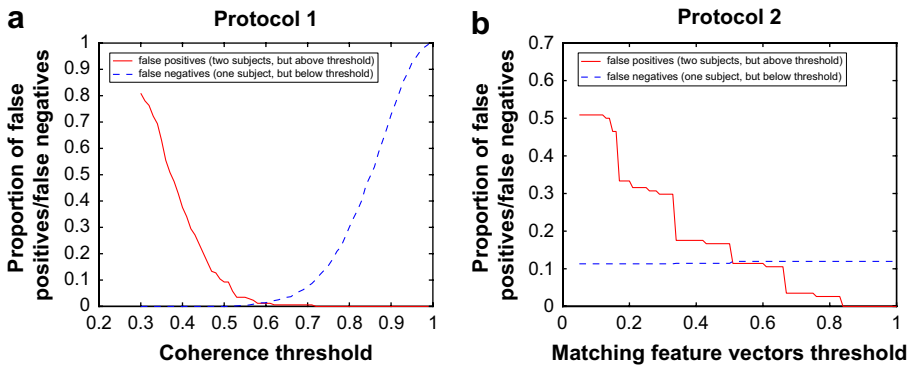


Fig. 5 – Thresholds for coherence and the number of matching FFT slices control the trade-off between false positives and false negatives w.r.t. all positive/negative samples.

devices, and a user can identify a target device in a way that their device can match with network entities.

During establishing and securing spontaneous interactions, spatial references are used both for selection of a target device and for verification that interaction is secured between the ''right'' devices. 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. (2005). 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 RF sensor network is separate from the wireless network that connects their host devices and used to co-ordinate ultrasonic sending). Sensing is performed by one node emitting ultrasound on its transducers, while all other nodes listen for a pulse on their transducers. Hazas et al. (2005) report a 90% precision around 8 cm in position and 25° in orientation: these figures and our practical experience suggest sufficient accuracy for reliable disambiguation of devices.

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. Fig. 6 shows two possible implementation. The one on the left is based on Guinard et al.'s (2007) *Gateways*: 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 Kortuem et al. (2005) 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.

5.1. *Authentication protocol*

As in the Shake-well-before-use protocol 1, we secure spontaneous interaction between two devices A and B in two phases, *key agreement* and *peer authentication*, as shown in Fig. 7.

5.1.1. *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 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 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 plain text 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.

5.1.2. *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. However, these distances can be tampered with by attacking the RF channel. We therefore introduce a method to embed information in ultrasound pulses, which (i) allows to use US as

Fig. 6 – Integration of spatial references to near-by devices in the mobile user interface; left, extension of Guinard et al.'s (2007) Gateways; right, Kortuem et al.'s (2005) map view.

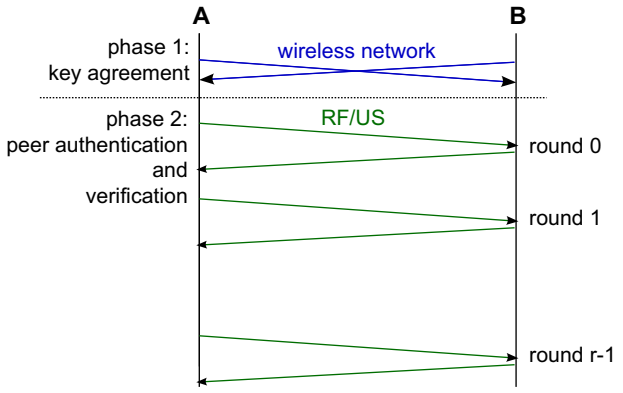


Fig. 7 – Devices A and B secure their interaction by key agreement over a wireless network channel, followed by peer authentication over the RF and US channels of their spatial sensors.

an out-of-band channel for message exchange, and (ii) makes the distances trustworthy.

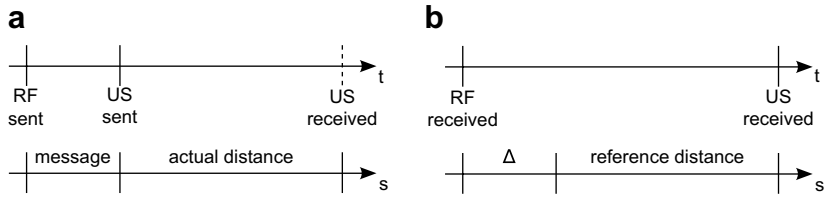
During authentication, the sender delays the sending of pulses to the effect of adding a certain perceived distance to the measurement, where the added distance represents information (in our protocol, a substring of the nonce). When for instance A receives a pulse and computes a distance, this distance 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 equal to the reference measurement, and therefore constitutes an implicit check of spatial integrity. Fig. 8 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.

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 again use an interlock protocol. In this case, the input to the interlock protocol is the random nonce that is transmitted both via RF (secured via interlock to prevent MITM attacks) and via US (to bind RF communication to spatial references and secured by the randomness of the input).

5.1.3. *Protocol specification*

An overview of the protocol phases is shown in Fig. 7. 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.

1. *Key agreement*, using the Diffie–Hellman key establishment protocol:
   (a) A chooses a random number $a \in \{1, \ldots, q-1\}$ and transmits $\mathbf{X} := g^a$, B chooses a random number $b \in \{1, \ldots, q-1\}$ and transmits $\mathbf{Y} := g^b$.
   (b) A computes $K_a^{\text{Sess}} := H(\widetilde{\mathbf{Y}}^a)$ and $K_a^{\text{Auth}} := H(\widetilde{\mathbf{Y}}^a \| C)$ with some secure hash algorithm, B generates K_b^{Sess} and K_b^{Auth} correspondingly from $\widetilde{\mathbf{X}}^b$. The numbers g, q and the string C 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



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

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

Fig. 8 – 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.

be advantageous to use long-term values for performance reasons. We use K^{Auth} ($= K_a^{\text{Auth}} = K_b^{\text{Auth}}$) for key verification in the peer authentication phase, and K^{Sess} ($= K_a^{\text{Sess}} = K_b^{\text{Sess}}$) for subsequent channel security if the verification succeeds.

2. *Peer authentication*:
   (a) A chooses a nonce $N_a \in \{1, \ldots, 2^{128} - 1\}$ and computes $M_a := E(K_a^{\text{Auth}}, N_a)$, B chooses N_b and computes M_b correspondingly with K_b^{Auth}.
   (b) *For each round* $i := 0, \ldots, r-1$:
   - A transmits an RF packet $\mathbf{M}_a^i := M_a[i \cdot b_m : (i+1) \cdot b_m - 1]$ and a 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 $\widetilde{\mathbf{M}}_a^i$ and US pulse $\widetilde{\mathbf{USP}}_a^i$, 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 $\widetilde{\mathbf{M}}_b^i$ and $\widetilde{\mathbf{USP}}_b^i$.

     A also verifies angle of arrival, computes $d_{a,b}^i$, uses the reference measurement $d_{a,b}$ to reconstruct $\Delta_b^i := d_{a,b}^i - d_{a,b}$.

     Finally A acknowledges B's messages for round i.

   (c) A reassembles all received RF packets $M'_b := \widetilde{\mathbf{M}}_b^0 \| \ldots \| \widetilde{\mathbf{M}}_b^{r-1}$.
   A decrypts the message $N'_b := D(K_a^{\text{Auth}}, M'_b)$.
   A 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 $K := K_a^{\text{Sess}}$ on match or $K :=$ null otherwise.
   B reassembles $M'_a := \widetilde{\mathbf{M}}_a^0 \| \ldots \| \widetilde{\mathbf{M}}_a^{r-1}$.
   B decrypts $N'_a := D(K_b^{\text{Auth}}, M'_a)$.
   B 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 $K := K_b^{\text{Sess}}$ 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.

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. More details on the RF protocol implementation of the interlock phase are presented in Mayrhofer et al. (2006).

Derived from the characteristics of the Relate sensors, we have set the number of bits transmitted in each round (with about 200 ms duration per 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%). 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.

5.2. *Security analysis*

5.2.1. *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. The nonce is also strictly used only once which rules out *replay* attacks.

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.

5.2.2. *Ultrasonic sensing and message transmission*

Our approach to coding random nonces (Section 5.1.2) and transmitting them via interlock (Section 5.1.3) prevents all attacks on the US channel: threat II.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 II.b is prevented by the random delays. As E cannot 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 cannot be arbitrarily small due to 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' 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 virtualised by sound forming, which is considered infeasible with the current state of the art in ultrasonic systems).

5.2.3. *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. The risk 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. 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.

5.2.4. *Speed versus security*

There is again 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. As in the case of shake-well-before-use, this number of rounds in our protocol only impacts on an attacker's one-off chance to guess the correct nonce to stage an undetected online MITM attack. It does not impact on the security level of 128 bits that will be provided after successful authentication against offline attacks.

6. Visible laser authentication

Our third method combines a wireless channel (RF) with a modulated laser (*L*) to create an authenticated secret key, similar to previous work (Kindberg and Zhang, 2003a). The difference is that we cannot use **L** for transmitting secret keys due to our assumption of *L* not providing confidentiality. Instead, *L* is used to transmit random numbers used only once (*nonces*) as part of a commitment scheme. Our protocol is designed so that an attacker would need to violate both the confidentiality and the integrity properties of the laser channel at the same time, i.e. to read what the user's personal device sends and to inject their own messages into the receiver.

From a user interaction point of view, we again combine two steps into one: device selection and implicit authentication. Nonetheless, this combined selection and authentication requires two user actions to prevent accidental selection of a "wrong" device. First the laser needs to be turned on to allow aiming, then the selection and implicit authentication needs to be performed. In our prototype implementation, user interaction is designed to be as simple as possible. We use a two-action button, similar to the buttons commonly used in digital cameras, to implement the two levels of action. By pressing the single button halfway, the laser lights up and allows proper aiming. By depressing the button fully, the target is selected and authenticated.

The protocol consists of the following steps between the user's personal device A and the remote device B:

1. The user presses the first button on A to turn on the laser and modulate it with a continuous stream of "ping" messages.
2. When the laser hits the receiver and the "ping" messages are detected, B switches to the "authentication in progress" state and broadcasts a "found" message over RF. In this state, B will only interact with a single personal device (the first to contact it in the next step).
3. By receiving the broadcast, A learns the network address of B. A and B agree to a secret key *K* via standard Diffie–Hellman key agreement (DH) over RF.
   B turns on its first LED (e.g. yellow).
4. When satisfied with the selection of B (and having seen its first LED turn on), the user presses the second button and the devices loop through the following steps until authentication is successful or the user stops the process by releasing the button:
   (a) A generates a fresh nonce *N*.
   (b) A computes $\mathbf{M}_1 := \mathrm{HMAC}_K(N|1)$ and sends it to B over RF.
   (c) B acknowledges the receipt by sending $\mathbf{M}_2 := \mathrm{HMAC}_K(\widetilde{\mathbf{M}}_1)$ to A over RF.
   (d) A verifies $\widetilde{\mathbf{M}}_2$ and transmits $\mathbf{M}_3 := N$ over *L* by modulating the laser.
   (e) B receives $\widetilde{N} := \widetilde{\mathbf{M}}_3$.
   B computes $\mathrm{HMAC}_K(\widetilde{N}|1)$ and verifies that it matches $\mathbf{M}_1$.
   B then sends $\mathbf{M}_4 := \mathrm{HMAC}_K(\widetilde{N}|2)$ over RF and turns on its second LED (e.g. green).
   (f) A verifies $\widetilde{\mathbf{M}}_4$ and notifies the user of successful verification, e.g. by turning on an LED (green).

The loop is necessary due to the possibility of transmission errors over **L**; it is important not to reuse nonces but to generate fresh nonces in each iteration. Only when both A and B signal success (e.g. with green LEDs) the user should continue with the interaction.

Note that the authentication part of the protocol does not rely on asymmetric primitives and is thus suitable for implementation on resource limited devices such as sensor nodes. However, when not assuming the laser channel to be confidential, asymmetric cryptography like DH or its Elliptic curve variant (ECDH) is necessary for creating a secret shared key (see step (2) in the protocol).

6.1. *Security analysis*

Our protocol uses both the (weak) confidentiality and integrity properties of the modulated laser channel:

- Integrity of L is exploited in steps (4b)–(4e): a MITM can only pass the check in (4e) when it can inject its own nonce N' so that the $\mathrm{HMAC}_K(N'|1)$ matches. Without such an injection on L, there are only two options: when the MITM simply relays $\mathbf{M}_1$, the HMAC will not match because of the different shared key. On the other hand, the MITM cannot generate a valid HMAC message because N has not yet been transmitted and is therefore unknown. Step (4b) thus serves to commit the sender A to the content that will be sent over L and to bind this commitment to the shared key K.
- Confidentiality of L is exploited in steps (4d)–(4f): a MITM can only pass the checks in (4e) and (4f) when they can eavesdrop on the laser, because only then will N be revealed.

Due to using long (i.e. $\geq$128 bits) nonces, this protocol is not susceptible to attacks against short codes on the out-of-band channel (Wong and Stajano, 2006, Section 3). Only when an attacker can perfectly overhear the original nonce N (sent by A over L) and inject an own nonce N' over L (as received by B) will the attack on RF go undetected. As outlined in Section 3, a laser channel is neither strictly confidential nor authentic. An attacker close to the target device B can observe the ''red dot'' at the sender and can shine a (possibly stronger and/or invisible IR) laser beam on the receiver, thus violating both the channel's confidentiality and authenticity. It remains to be shown how practical such attacks on both the confidentiality and the integrity are, taking the mobility of A and short interaction times into account.

7. Authentication proxies

All these methods as well as other suggested protocols assume that those devices that authenticate each other can experience the same context, but this is not always possible. Fig. 9 shows a device A, e.g. owned by Alice, trying to interact securely with a device B, e.g. a WLAN access point. Because the access point is physically inaccessible, Alice cannot benefit from direct context authentication with it to secure her communication. By introducing a *context authentication proxy* P, we give her this option. To facilitate an authentication between two devices, the authentication proxy experiences the same context as one of the devices, i.e. it shares some aspect of the context. With the other device, it is pre-authenticated. It will usually be desirable that context be shared with the more volatile side, i.e. with mobile devices, changing environments, or, generally speaking, with transient connections. Since we assume a more permanent relationship with the other end of the authentication, in this example between P and the access point, the necessary pre-authentication only needs to occur once during set-up of these devices. Any standard authentication protocol, e.g. password- or certificate-based ones or any means of conveying trust of B in P can be used. Due to this trust relationship, the possibly mobile authentication proxy P is assumed to be used or maintained by a trusted person, such as a system administrator.

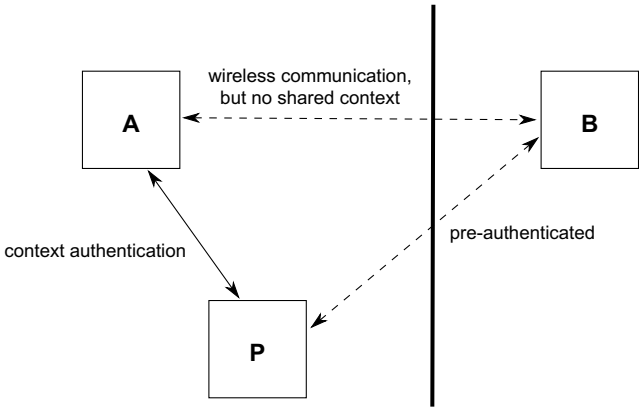


Fig. 9 – Using a context authentication proxy P allows physically separated devices A and B to benefit from context authentication even when they cannot experience the same context.

The main task of the authentication proxy is to create a shared secret between A and B, to enable secure communication between them over a wireless network. Depending on the initiator of the authentication, we can distinguish between two different approaches for user interaction with the proxy:

- We speak of a *passive authentication proxy* when P acts as an authentication service and simply waits for clients to initiate an interaction. The *client* takes the active role, starts context authentication with P to obtain a shared secret for communicating securely with B, and may need to engage in another authentication procedure with B over the now-authenticated wireless network. Instances of this approach are the closely related NiaB (Balfanz et al., 2004), and one of our previous works (Mayrhofer et al., 2003), which describes the use of RFID tags to secure communication over wireless ad-hoc peer-to-peer networks.
- For an *active authentication proxy*, the roles of waiting for and of initiating the context authentication are swapped between A and P. That is, the *proxy* takes the active role, starts context authentication with A to generate a shared secret for letting A communicate securely with B, and may take additional steps to register A with authorisation databases. In this case, A only waits to be authenticated and does not need to take any additional steps. This requires even less user interaction by offloading some steps to the proxy and thus can further decrease the burden placed on the user for setting up secure communication.

Choosing between a passive and an active authentication proxy also depends on the respective trust model. If the trust model can express transitive trust, i.e. delegating trust from one entity to another, then B can delegate authorisation decisions to P. We can further distinguish between an *offline* and an *online* connection between the P and B. In the former case, B delegates trust about authorisation to P by allowing all clients A authenticated by B to establish connections. This option has the advantage that, after pre-authentication, no further communication between B and P is necessary for authenticating arbitrary clients (except potential updates, e.g. of certificate revocation lists). In the latter case, P requests authorisation from B using an online connection. The

necessarily secure connection between B and P forms the pre-authentication between them with a slightly different trust model. B trusts P to authenticate A based on shared context and to forward machine information and certificates, but keeps decisions about authorisation local. For spontaneous interaction, the first option has the advantage that no connection between B and P is necessary.

8. Conclusions

In this article, we have presented three specific methods for spontaneous device authentication. All of them protect primarily against man-in-the-middle attacks on wireless communication using different out-of-band channels. Potential applications for these pairing protocols are manifold; coupling a mobile phone with a Bluetooth headset, establishing a transient secure connection between two smart cards for exchanging digital money, passing access rights between key chains, temporarily using a printer or larger display as part of the infrastructure, or general data transfer are prominent examples.

To be able to use the proposed out-of-band channels, devices require the respective sensors. For authentication based on shaking, simple and cheap accelerometers are sufficient. 3D accelerometers are already being embedded into off-the-shelf mobile devices like the ''Nokia 5550 Sport'', the ''Nokia N95'', the ''Apple iPhone'', or the ''FIC Freerunner'' and can immediately be used for authentication with both presented protocols. For authentication based on spatial references, devices require ultrasonic transducers such as those embedded in our ''Relate dongles'', which are not yet available in off-the-shelf products but offer additional benefits for device discovery and selection or in-door navigation. For authentication using visible lasers, mobile devices will need to be equipped with laser diodes, which are also simple, cheap, and small enough to be embedded without noticeable added cost.

In all presented methods, the explicit user interaction – taking two devices into one hand and shaking them as an indication that they should pair, selecting a device based on its relative location, or aiming a visible laser at it – is coupled with implicit authentication. This limits the burden placed on users. Connections are secured by default, not only as an option.

Full source code of our implementations including cryptographic protocols, demonstration applications, as well as test data sets are available as open source at http://www.openuat.org and http://ubicomp.lancs.ac.uk/relate.

Acknowledgments

We gratefully acknowledge support 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

Balfanz D, Durfee G, Grinter RE, Smetters DK, Stewart P. Network-in-a-box: how to set up a secure wireless network in under a minute. In: Proceedings of the 13th USENIX security symposium. USENIX; August 2004. p. 207–22.

Batina L, Mentens N, Verbauwhede I. Side-channel issues for designing secure hardware implementations. In: Proceedings of IOLTS: IEEE online testing symposium. IEEE CS Press; 2005. p. 118–21.

Bichler Daniel, Stromberg Guido, Huemer Mario, Löw Manuel. Key generation based on acceleration data of shaking processes. In: Proceedings of UbiComp 2007: ubiquitous computing. LNCS, vol. 4717. Springer-Verlag; September 2007. p. 304–17.

Čagalj M, Čapkun S, Hubaux J-P. Key agreement in peer-to-peer wireless networks. IEEE (Special Issue on Cryptography and Security) 2006;94:467–78.

Diffie W, Hellman ME. New directions in cryptography. IEEE Transactions on Information Theory 1976;22(6):644–54.

Eronen P, Tschofenig H. RFC4279: Pre-shared key ciphersuites for transport layer security (TLS); December 2005.

Ferguson N, Schneier B. Practical cryptography. Wiley Publishing; 2003.

Gehrmann C, Mitchell CJ, Nyberg K. Manual authentication for wireless devices. RSA Cryptobytes 2004;7(1):29–37.

Goodrich MT, Sirivianos M, Solis J, Tsudik G, Uzun E. Loud and clear: human verifiable authentication based on audio. In: Proceedings of ICDCS 2006. IEEE CS Press; July 2006. p. 10.

Dominique Guinard, Sara Streng, Hans Gellersen. Relategateways: a user interface for spontaneous mobile interaction with pervasive services. In: CHI 2007 workshop on mobile spatial interaction; 2007.

Hazas M, Kray C, Gellersen H, Agbota H, Kortuem G, Krohn A. A relative positioning system for co-located mobile devices. In: Proceedings of MobiSys 2005. ACM Press; June 2005. p. 177–90.

Hoepman J-H. The emphemeral pairing problem. In: Proceedings of eighth international conference on financial cryptography. Springer-Verlag; February 2004. p. 212–26.

Holmquist LE, Mattern F, Schiele B, Alahuhta P, Beigl M, Gellersen H-W. Smart-its friends: a technique for users to easily establish connections between smart artefacts. In: Proceedings of UbiComp 2001. Springer-Verlag; September 2001. p. 116–22.

Huynh T, Schiele B. Analyzing features for activity recognition. In: Proceedings of Soc-EUSAI 2005, ACM international conference proceeding series. ACM Press; October 2005. p. 159–63.

Kindberg T, Zhang K. Secure spontaneous device association. In: Proceedings of UbiComp 2003. Springer-Verlag; October 2003a. p. 124–31.

Kindberg T, Zhang K. Validating and securing spontaneous associations between wireless devices. In: Proceedings of ISC'03. Springer-Verlag; October 2003b. p. 44–53.

Kindberg T, Zhang K, Im SH. Evidently secure device associations, Technical report HPL-2005-40. Bristol: HP Laboratories; March 2005.

Darko Kirovski, Mike Sinclair, David Wilson. The Martini synch. Technical report MSR-TR-2007-123, Microsoft Research; September 2007.

Kortuem G, Kray C, Gellersen H. Sensing and visualizing spatial relations of mobile devices. In: Proceedings of UIST 2005. ACM Press; October 2005. p. 93–102.

Krawczyk H, Bellare M, Canetti R. RFC2104:HMAC: keyed-hashing for message authentication; February 1997.

Lester J, Hannaford B, Borriello G. ''Are you with me? '' – using accelerometers to determine if two devices are carried by the same person. In: Proceedings of Pervasive 2004. Springer-Verlag; April 2004. p. 33–50.

Mayrhofer R. A context authentication proxy for IPSec using spatial reference. In: Proceedings of TwUC 2006: first international workshop on trustworthy ubiquitous computing. Austrian Computer Society (OCG); December 2006. p. 449–62.

Mayrhofer R. The candidate key protocol for generating secret shared keys from similar sensor data streams. In: Proceedings of ESAS 2007: fourth European workshop on security and privacy in ad hoc and sensor networks. LNCS, vol. 4572. Springer-Verlag; July 2007. p. 1–15.

Mayrhofer R, Gellersen H. On the security of ultrasound as out-of-band channel. In: Proceedings of IPDPS 2007: 21st IEEE international parallel and distributed processing symposium. IEEE CS Press; March 2007. p. 321 [Track SSN 2007: third international workshop on security in systems and networks].

Mayrhofer R, Gellersen H. Shake well before use: authentication based on accelerometer data. In: Proceedings of Pervasive 2007: fifth international conference on Pervasive Computing. LNCS, vol. 4480. Springer-Verlag; May 2007. p. 144–61.

Mayrhofer R, Gellersen H, Hazas M. An authentication protocol using ultrasonic ranging, Technical report COMP-002-2006. Lancaster University; October 2006.

Mayrhofer R, Gellersen H, Hazas M. Security by spatial reference: using relative positioning to authenticate devices for spontaneous interaction. In: Proceedings of UbiComp 2007: ninth international conference on ubiquitous computing. LNCS, vol. 4717. Springer-Verlag; September 2007. p. 199–216.

Mayrhofer R, Ortner F, Ferscha A, Hechinger M. Securing passive objects in mobile ad-hoc peer-to-peer networks. In: Focardi R, Zavattaro G, editors. Electronic notes in theoretical computer science, vol. 85.3. Elsevier Science; June 2003.

Mayrhofer R, Welch M. A human-verifiable authentication protocol using visible laser light. In: Proceedings of ARES 2007: second international conference on availability, reliability and security. IEEE CS Press; April 2007. p. 1143–7 [Track WAIS 2007: first international workshop on advances in information security].

McCune JM, Perrig A, Reiter MK. Seeing-is-believing: using camera phones for human-verifiable authentication. In: Proceedings of IEEE symposium on security and privacy. IEEE CS Press; May 2005. p. 110–24.

Nicholson AJ, Smith IE, Hughes J, Noble BD. LoKey: leveraging the SMS network in decentralized, end-to-end trust establishment. In: Proceedings of Pervasive 2006. Springer-Verlag; May 2006. p. 202–19.

Patel SN, Abowd GD. A 2-way laser-assisted selection scheme for handhelds in a physical environment. In: Proceedings of UbiComp 2003. Springer-Verlag; October 2003. p. 200–7.

Ringwald M. Spontaneous interaction with everyday devices using a PDA. In: Proceedings of workshop on supporting spontaneous interaction in ubiquitous computing settings; September 2002.

Rivest RL, Shamir A. How to expose an eavesdropper. Commununications of ACM 1984;27(4):393–4.

Shaked Y, Wool A. Cracking the Bluetooth PIN. In: Proceedings of MobiSys 2005. ACM Press; June 2005. p. 39–50.

Stajano F, Anderson R. The resurrecting duckling: security issues for ad-hoc wireless networks. In: Proceedings of seventh international workshop on security protocols. Springer-Verlag; April 1999. p. 172–94.

Vaudenay S. Secure communications over insecure channels based on short authenticated strings. In: Proceedings of CRYPTO 2005. Springer-Verlag; August 2005.

Wong F-L, Stajano F. Multi-channel protocols. In: Proceedings of security protocols workshop 2005. Springer-Verlag; 2006.