An Authentication Protocol using Ultrasonic Ranging

Technical Report Number COMP-002-2006, October 2006

Rene Mayrhofer, Mike Hazas, and Hans Gellersen

Lancaster University, Infolab21, South Drive, Lancaster, LA1 4WA, UK
{rene,hazas,hwg}@comp.lancs.ac.uk,
WWW home page: http://ubicomp.lancs.ac.uk/

Abstract. This report presents a method for establishing and securing spontaneous interactions on the basis of spatial references which are obtained by accurate sensing of relative device positions. Utilising the Relate ultrasonic sensing system, we construct an interlocked protocol using radio frequency messages and ultrasonic pulses for verifying that two devices share a secret. This verification is necessary to prevent man-in-the-middle attacks on standard Diffie-Hellman key agreement.

1 Introduction

Key agreement over insecure channels, such as IEEE 802.11 wireless LAN, requires authentication of the generated keys to prevent man-in-the-middle (MITM) attacks. We approach the authentication problem with context sensing to obtain physical evidence for the authenticity of an encountered device, and we specifically employ sensing of the spatial relationship of the involved devices for this purpose. The principle of using physical evidence to bootstrap security for wireless ad hoc networked devices has been well established. In initial work on this problem, Stajano and Anderson proposed direct electrical contact for key exchange between devices [12]. Balfanz et al. [1] and Kindberg et al. [9] provided more general considerations of key exchange over communication channels with inherent physical limitations. The principle in using such channels is that a device has to be in a certain physical context in order to establish communication. This means the context is implicit, as a property of the channel. In our contribution we build on this principle but in addition use explicit measurement of spatial context for device authentication.

The contribution of this paper is a protocol for verifying that a secret key is shared with one specific device, identified by a *spatial reference*. These references are used to initiate interaction, and implicitly used in the presented protocol for device authentication and key exchange. The protocol is embedded seamlessly in the process of establishing an interaction, in the sense that it does not alter the sequence of events as far as user interaction with the target device is concerned.

2 Related Work

We are not aware of any previous implementation of peer authentication in combination with ultrasonic sensing. However, Kindberg et al. have proposed such an approach before us and outlined a protocol operating over radio frequency (RF) and ultrasound (US) channels [8]. Their proposed method is to prompt a network-discovered and user-selected device to send radio and ultrasonic beacons from which spatial parameters can be derived and presented to the user for verification that these correspond with the intended target. The idea is further to check during subsequent key exchange that messages are received from the corresponding spatial range. In our approach we provide an actual implementation based on ultrasound sensing, to explore in detail how spatial integrity of exchanged messages can be achieved. Specific conceptual differences in our approach are the use of an interlock protocol that more tightly couples RF and US communication, and a user interaction model in which spatial references are provided in the first place for device discovery and selection and then implicitly used for verification of device authenticity.

We combine the idea of using ultrasound for security purposes with the idea of using ultrasonic sensing in order to provide spatial references for user interaction. Hazas et al. [6], 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 Kortuem et al. [10] further discussed 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). 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. However, we adopt an advanced visualisation of device positions in space beyond the two-dimensional views proposed by Kortuem et al., and extend the approach by adding security in a seamless but transparent 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.

Technical requirements for our method include spatial sensing at an appropriate level of accuracy, and visualisation of device positions in appropriate detail. The sensors must be sufficiently accurate to allow reliable disambiguation of a target device from third devices, and the visualisation must be of a quality that

allows users to reliably match visualised references with device arrangements visible to them in the real world.

For illustration of our method consider the scenario of Alice and Bob in a meeting as introduced above. The devices of Alice and Bob as well as the devices of other meeting participants will discover each other as a result of the spatial discovery process. Note that devices of users in a next door meeting will not be discovered even though they may be connected to the same wireless network. The devices in our meeting perform measurements among themselves, compute relative positions, and each visualise these in a device map for their users. Alice will now be able to invoke file transfer to Bob's device by, for instance, dragging of the respective document icon in her user interface onto the position and icon in the device map that represents Bob's device. She will not need to perform any further action to secure the transfer, but she will have feedback through her interface first on progress in securing the link and then on transfer of the file.

3.1 Sensor Platform

As a fundament for our procedure we require a sensor platform that provides spatial discovery and relative positioning. We adopt the Relate sensor system introduced by Hazas et al. [6] and also used in the related work of Kortuem et al. [10]. The system is based on wireless sensors implemented as USB dongles that can be readily used to extend host devices 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 packets 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 to share and collect sensor data, and each sensor provides to its host device not only its own measurements but also those taken by other sensors in the network. 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 [6]: these figures and our practical experience suggest sufficient accuracy for reliable disambiguation of devices.

4 Key Agreement and Peer Authentication

A central component of our method is the protocol that we have designed for securing spontaneous interaction and specifically for verification of peer device authenticity. The protocol corresponds to step 5 in the overall procedure described above, and for this section we assume that the preceding steps have been successful. This means we assume for now that the user has selected a target device correctly and that as a result the user's device is in possession of a correct spatial reference associated with the intended target.

4.1 Key Agreement

We secure spontaneous interaction between two devices A and B in two phases, *key agreement* and *peer authentication*. In the first phase, we let A and B establish a shared key using an unauthenticated key agreement protocol, such as Diffie-Hellman (DH) [3]. If this is successful, then A and B can use the agreed key to protect their communication against eavesdropping and tampering, with an attacker being unable to gain sufficient knowledge of that shared key. However, unauthenticated key agreement in itself is open to 'man-in-the-middle' (MITM) attack: an attacking device M can pretend to A that it is B, and to B that it is A, and thus achieve key agreement 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 a 'man in the middle'. To protect A and B against this threat, 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).

4.2 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. As a starting point for authentication, A has a spatial reference to B as derived from Alice'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 couple this process with spatial sensing. The devices engage in a protocol designed to establish that (i) they have 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 nonce over the US channel in a series of smaller parts that are coded

(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. 1. 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.

as distance added to the actual distance between devices. 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.

Embedded in the described exchange is an implicit check of spatial integrity. When a device receives an ultrasonic pulse it computes a distance measurement based on the time-of-flight. However, during authentication the sender delays the sending of pulses to the effect of adding a certain distance to the measurement, where the added distance represents 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 a part of the nonce. A proceeds with subtracting the reference distance it has of B. This will let A retrieve the nonce information (i.e. the added distance) correctly only if the received pulse has been sent from a range that corresponds with the relative position of B. Figure 1 illustrates this mechanism for message transfer 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 M is present as MITM on the RF channel. M would be able to infer the nonces exchanged between A and B by taking its own US measurements, and it could then use its keys (maliciously agreed with A and B in the key agreement phase) to 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 [11]. 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 first before the message can be decrypted to retrieve the nonce. If M now receives a message part from A intended for B it can not retrieve any part of the nonce. M will also not receive

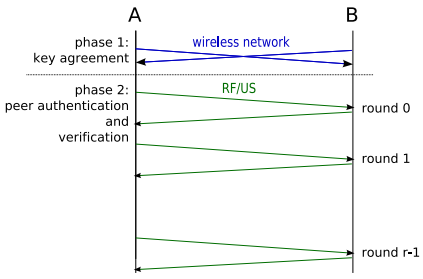


Fig. 2. Overview of the spatial authentication protocol

more message parts from A unless it passes the current one on to B, as A and B strictly adhere to turn-taking. M's only choices are then to guess the content for all message parts 'passing through' 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 there keys do not match (thereby detecting the presence of a MITM). The interlock protocol thus rules out that a MITM attack on the RF channel can succeed during peer authentication.

4.3 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 channels capacity. The US channel's 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. 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: $c := E_{EA}(K, m)$ describes the encryption of plain text m under key K with algorithm EA, $m := D_{EA}(K, c)$ the corresponding decryption, $H_{HA}(m)$ describes the hashing of the message m with algorithm HA, 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.

Figure 2 gives a graphical overview of the spatial authentication protocol and Appendix A presents the complete specification. It includes the following steps:

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 $K_a^{Sess} := H_{HA}(\mathbf{Y}^a)$ and $K_a^{Auth} := H_{HA}(\mathbf{Y}^a||PAD)$ with some secure hash algorithm HA,
   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 (in the sense as defined by [5, section 15.8.4]).
2. *Peer authentication*:
   (a) A chooses a nonce $N_a \in \{1, ..., 2^{128} - 1\}$ and computes $M_a := E_{EA}(K_a^{Auth}, N_a)$ using a secure block cipher EA,
   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 $\mathbf{USP_b^i}$, 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 $N_b' := D_{EA}(K_a^{Auth}, M_b')$, 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 $N_a' := D_{EA}(K_b^{Auth}, M_a')$, 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 [4]. 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.

4.4 Implementation

We have implemented the key agreement phase of our protocol over TCP/IP. As a secure hash HA we use $SHA_{DBL} - 256$ [5], which is a double execution of the standard $SHA - 256$ message digest to safeguard against length extension and partial-message collision attacks [7]: $SHA_{DBL} - 256(m) := SHA - 256(SHA - 256(m)||m)$.

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 the block cipher EA 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. And pulses emitted on the ultrasonic channel serve simultaneously for ranging and for transmission of nonce message parts (see Appendix B for details).

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 25.6cm which the sender adds as offset to the receiver-perceived distance by delaying the US pulse. At the receiver end, this allows for +/-12.8cm of measurement inaccuracy to retrieve the 3 bit correctly (note the reported precision of Relate sensors for this level of accuracy is over 95%). The duration of a round is about 200ms (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.

4.5 Security Analysis

We assume our environment to be open to *eavesdropping* on the RF and US channels. Over RF, all packets are encrypted with an authentication key, but over US the nonce will become gradually revealed as the protocol progresses. 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.

The interlock protocol also protects against *man-in-the-middle attack* during authentication. If a MITM is present on the RF channel than key verification will fail because the MITM will not be able decrypt and re-encrypt message parts exchange between the authenticating peers, unless the MITM achieves a concurrent attack on the US channel. However, our protocol guards against an

attack on the US channel by the coupling of message transfer with ultrasonic ranging. A MITM would have be positioned precisely in the line-of-sight between authenticating devices in order to attempt interception and manipulation of US pulses but their presence literally in the middle between devices would be obvious to the user. Moreover, this MITM can not be arbitrarily small due to a physical limit on the minimum size of ultrasound transducers. These two points make attacks on the US channel significantly harder compared to the RF channel.

Attacks on *communication integrity* are thus possible, for instance masking an US pulse with a stronger one, but it will not be practically feasible for an attacker to modify communication on the US channel such that this will remain undetected, and exploitable by the attacker to achieve authentication. All an attacker can achieve is to disrupt authentication, accounting for a *denial-of-service attack* to which RF and US are generally open.

In consideration of level of security 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 bit that will be provided after successful authentication. This difference is even more pronounced than the typical online vs. offline attack discussion (cf. [5, p. 103] for a short introduction into this topic). The reason is 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.

5 Conclusion

We have contributed and discussed a protocol for authenticating secret keys based on 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 exploits a novel mechanism for message transfer over ultrasound in a manner that ensures spatial integrity of the sender. The protocol is embedded in a spatial sensing scheme that more generally provides devices with accurate relative positions of peers discovered within interaction range. We have also provided a brief analysis of protocol security.

Acknowledgements

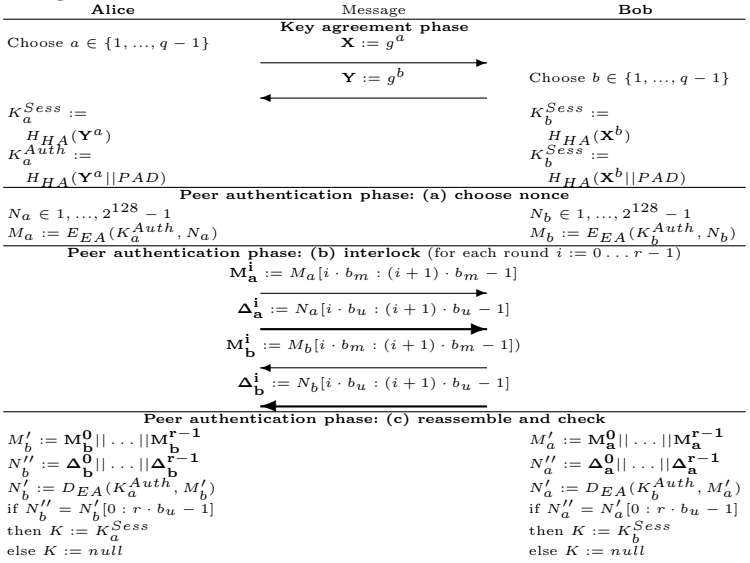
We gratefully acknowledge support for the presented research by the Commission of the European Union under contract 013790 "RELATE", and by the Engineering and Physical Sciences Research Council in the UK under grant GR/S77097/01.

References

1. Balfanz, D., Smetters, D. K., Stewart, P., and Wong, H. C. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. NDSS'02: 2002 Network and Distributed Systems Security Symposium* (February 2002), The Internet Society.
2. Decker, C., Krohn, A., Beigl, M., and Zimmer, T. The particle computer system. In *Proceedings of the ACM/IEEE 4th International Conference on Information Processing in Sensor Networks* (2005).
3. Diffie, W., and Hellman, M. E. New directions in cryptography. *IEEE Transactions on Information Theory IT-22*, 6 (1976), 644–654.
4. Eronen, P., and Tschofenig, H. RFC4279: Pre-shared key ciphersuites for transport layer security (TLS), December 2005.
5. Ferguson, N., and Schneier, B. *Practical Cryptography.* Wiley Publishing, 2003.
6. Hazas, M., Kray, C., Gellersen, H., Agbota, H., Kortuem, G., and Krohn, A. A relative positioning system for co-located mobile devices. In *Proc. MobiSys 2005: 3rd Int. Conf. on Mobile Systems, Applications, and Services* (New York, NY, USA, June 2005), ACM Press, pp. 177–190.
7. Kaminsky, D. MD5 to be considered harmful someday. Cryptology ePrint Archive, Report 2004/357, 2004.
8. Kindberg, T., and Zhang, K. Validating and securing spontaneous associations between wireless devices. In *Proc. ISC'03: 6th Information Security Conf.* (October 2003), Springer, pp. 44–53.
9. Kindberg, T., Zhang, K., and Shankar, N. Context authentication using constrained channels. In *Proc. WMCSA: 4th IEEE Workshop on Mobile Computing Systems and Applications* (June 2002), IEEE Computer Society, pp. 14–21.
10. Kortuem, G., Kray, C., and Gellersen, H. Sensing and visualizing spatial relations of mobile devices. In *Proc. UIST 2005: 18th ACM Symposium on User Interface Software and Technology* (October 2005), ACM Press, pp. 93–102.
11. Rivest, R. L., and Shamir, A. How to expose an eavesdropper. *Commununications of ACM 27*, 4 (1984), 393–394.
12. Stajano, F., and Anderson, R. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proc. 7th Int. Workshop on Security Protocols* (April 1999), Springer, pp. 172–194.

A Protocol specification

Both sides perform the same protocol in terms of steps and messages, i.e. the protocol is symmetric between the peers. Thin arrows indicate transmission of a message via RF, while thick lines indicate transmission via ultrasonic pulses.



B Sensor device interlock protocol

After a user triggers peer authentication by selecting a device for interaction, both Relate sensor devices are put into authentication mode. The host device (e.g. the laptop computer they are connected to) provides:

- The remote device ID to authenticate with.
- The 128 bit nonce N.
- The 128 bit message M to be sent over the RF network, i.e. the nonce encrypted with the authentication key K^{Auth}, which is only known to the host devices.
- The agreed number of rounds r and the number of bits b_u of M to code in US pulses in each round. b_m can be computed independently by the Relate sensor device.

Relate sensor devices are based on Particle Smart-Its and their AwareCon RF protocol stack [2]. A sensing device adapts its ultrasound transmission rate based on the number of devices present; thus the protocol is essentially round-robin. The sensor devices take turn in transmitting their RF packets and, triggered by them, US pulses. Each RF packet contains a part of b_m bits of M and the current round number $i \in \{0, \ldots, r-1\}$. It also acts as a trigger for sending the US pulse delayed by the respective b_u bits of N. Both the RF packet parts and the US delay parts are taken from M and N, respectively, starting at the least

significant bits. This bit order is conserved: lower order bits in the nonce are also lower order in the measurement. That is, in round i, the Relate sensor device sends $M[i \cdot b_m : (i+1) \cdot b_m - 1]$ and delays its US pulse by $U[i \cdot b_u : (i+1) \cdot b_u - 1]$ time units corresponding to units of 25.6cm in distance.

Insofar, this protocol only transmits the plaintext nonce N and its encryption M, taking care of the low-level details of transmitting and reassembling the bits. From a security point of view, it is critical that a device does not transmit an encrypted packet of round i before it has received a confirmation that the authentication partner has received its packet of round $i-1$. However, there is a catch: An attacker could easily fake explicit acknowledgements, making the Relate device believe that the other device has processed its packet of round $i-1$, and thus continuing to send its next round and subsequently giving an attacker the chance to receive, decrypt, and re-encrypt the packets. Requiring authentic acknowledgement packets in an authentication protocol would create a chicken-and-egg problem and is therefore not viable. The interlock protocol resolves this issue by requiring a device to have received and stored the remote packet of round $i-1$ before it sends its own packet i.

In the protocol implementation, we use a combination of explicit and implicit acknowledgement. As the RF packets carry the protocol round counter i, a Relate sensor device will only send part i if it has received part $i-1$ or part i from the remote side. This tolerance window of 1 round is required to start the protocol — either A or B will be the first to send part i. By sending packet i, a Relate sensor device implicitly acknowledges the receipt of either packet $i-1$ or packet i from the remote. To distinguish between these two cases, the RF packets contain an explicit acknowledgement flag, which is set when the packet acknowledges the current round i and cleared when it acknowledges the previous round $i-1$. Due to the possibility of packet losses, parts are retransmitted until the remote either implicitly acknowledges its receipt by incrementing its own round counter, or explicitly acknowledges by setting the acknowledgement flag (this distinction is mostly important for starting and ending the protocol properly, i.e. for rounds 0 and $r-1$).