USING A SPATIAL CONTEXT AUTHENTICATION PROXY FOR ESTABLISHING SECURE WIRELESS CONNECTIONS*

RENE MAYRHOFER
Computing Department, Lancaster University, South Drive
Lancaster LA1 4WA, United Kingdom
rene@comp.lancs.ac.uk

ROSWITHA GOSTNER
Computing Department, Lancaster University, South Drive
Lancaster LA1 4WA, United Kingdom
gostner@comp.lancs.ac.uk

Abstract

Spontaneous interaction in wireless ad-hoc networks is often desirable not only between users or devices in direct contact, but also with devices that are accessible only via a wireless network. Secure communication with such devices is difficult because of the required authentication, which is often either password- or certificate-based. An intuitive alternative is context-based authentication, where device authenticity is verified by shared context, and often by direct physical evidence. Devices that are physically separated cannot experience the same context and thus cannot benefit directly from context authentication. We introduce a *context authentication proxy* that is pre-authenticated with one of the devices and can authenticate with the other by shared context. This concept is applicable to a wide range of application scenarios, context sensing technologies, and trust models. We show its practicality in an implementation for setting up IPSec connections based on spatial reference. Our specific scenario is ad-hoc access of mobile devices to secure 802.11 WLANs using a mobile device as authentication proxy. A user study shows that our method and implementation are intuitive to use and compare favourably to a standard, password-based approach.

1 Introduction

Spontaneous interaction is a desirable feature for many ubiquitous computing scenarios. It is typically seen as a process between users or devices that are in direct contact with each other, and often implies spatial proximity. However, spontaneous interaction can also be important between users or devices that are physically or virtually separated, but can communicate over some common channel like a wireless network. A similar situation

*This is an extended version of "Rene Mayrhofer: *A Context Authentication Proxy for IPSec using Spatial Reference*, Proc. TwUC 2006, Austrian Computer Society (OCG), 449–462, December 2006".

arises when interacting with devices that do not feature any user interface, but only communicate wirelessly. One prominent example is IEEE 802.11 WLAN itself: users, represented by their client devices, engage in spontaneous interaction with access points that usually neither have a user interface nor are physically accessible (they might be built into building infrastructure).

The problem with such settings is to authenticate users or devices. Wireless networks are particularly vulnerable to attacks, ranging from simple eavesdropping to more sophisticated man-in-the-middle (MITM) attacks. Although there are well-known protocols to secure communication over wireless networks, they all depend on some form of authentication. Only after authenticating the communication partner, further steps to create a secure channel make sense. More specifically, the problem is to authenticate intuitively and efficiently.

From a user point of view, secure channel setup should be as transparent as possible and should cause minimal, if any, overhead to the desired spontaneous interaction. Any additional burden that is caused by authentication is not part of the intended interaction, and thus collides with spontaneity. Obvious and often deployed solutions for authentication are typically either secure or convenient. Password-based authentication like Bluetooth-style PINs, WEP, and WPA-PSK is one example, which is unfortunately neither particularly secure nor user friendly; another well-known solution is certificate-based authentication like X.509 public key infrastructures (PKIs).

An example of a secure channel implementation is IPSec. It is currently considered one of the most secure communication protocols, supports both password- and certificate-based authentication, and has been designed for cross-platform interoperability, but is daunting to set up even for technically skilled users. Although it has desirable properties from a security point of view, many users may choose not to use it for spontaneous and ad-hoc interactions. Giving credit to its wide-spread use and practical problems, also investigated by others [1, 3], we therefore use the setup of IPSec connections as our motivating example. More specifically, our demonstration application is to grant secure access to a WLAN access point – and consequently the network it manages – to new clients such as laptops via IPSec connections.

Context based authentication, or *context authentication*, allows secure and intuitive authentication without introducing unreasonable overhead that would be incompatible with spontaneous interaction. It uses shared context between devices to create shared secrets. These shared secrets can consequently be used as cryptographic tokens for creating secure channels. However, devices such as WLAN access points that are physically separated from user devices or that have no sensors or user interfaces are unable to experience the same context.

Our approach to allow such devices to authenticate via shared context is to introduce a *context authentication proxy*. The proxy is pre-authenticated to the device that does not have sensors or a user interface itself, and authenticates to other devices on behalf of it. This concept is independent of the underlying infrastructure for expressing trust, and can work in online and offline settings and with existing password- or certificate-based authentication mechanisms. Our example application uses a mobile context authentication proxy in the form of a personal digital assistant (PDA) for better ease of use.

The contribution of this work is twofold: we examine the general concept of a context authentication proxy in more detail, discuss different options of implementing it, and we show a specific application for a widely used protocol. A user study shows that authentication based on relative location — one aspect of shared context — is a viable alternative to standard, password-based authentication. Our implementation also confirms a user study presented in related work [1], anecdotally showing a significant improvement of ease of use in setting up IPSec connections due to use of context authentication. We also argue that, although demonstrated by an application for securing wireless networks,

context authentication proxies are of wider applicability.

In the following, we first discuss related work in Section 2 and the previously introduced concept of context authentication in Section 3. Our main contribution is the general notion of an authentication proxy presented in Section 4 and our specific implementation for authenticating IPSec connections shown in Section 5 which we investigate in Section 6. Finally, we discuss further alternatives for implementing context authentication and for using the established shared secrets in secure communication protocols in Section 7.

2 Related work

Our chosen example of securing IEEE 802.11 WLAN using context authentication has also been discussed by Balfanz et al. [1]. They present a system called "Network-in-a-box" (NiaB) that uses an infrared channel to transmit authentic cryptographic tokens and automates the set-up of secure wireless communication in much the same way as our example application does. This infrared connection is established between the client device and either the WLAN access point itself, or, in case of a distributed infrastructure, an "enrollment station", which can be regarded as a stationary instance of a context authentication proxy. Furthermore, they show in a user study that context authentication can, for end-users, significantly lower the time required to set up a secure wireless network. The major difference to our work is the role of the authentication proxy. In NiaB, the authentication proxy is described as a permanent station that authenticates all devices that are able to establish infrared connections to it. On the other hand, we specifically assign the authentication proxy an active role, in which it triggers the authentication process to a selected client, as described in more detail in Section 4. A specific advantage of our approach is that the authentication proxy can be mobile — and for our demonstration application, it explicitly is. Instead of forcing users to bring their devices to fixed stations, administrators can authenticate devices wherever it is necessary and appropriate. This can include authentication of new fixed stations, which is not possible with the less flexible enrollment station described by NiaB.

Kindberg et.al. [8] describe "channel proxies", which may be seen as a low-level implementation of a context authentication proxy. These channel proxies selectively forward messages depending on some constraints, like location of the sender or the receiver. In contrast, our concept of context authentication proxies explicitly includes high-level processing of messages. In our example, this allows the complete authentication protocol to be performed between the proxy and the WLAN client, while the WLAN access point will typically be unaware of the whole process.

Godber and Dasgupta [3] describe another implementation that is closely related to the demonstrative application we discuss in Section 5. Their system called "Secure Wireless Gateway" (SWG) uses IPSec to secure IEEE 802.11b WLAN, and also provides a captive portal to redirect unauthenticated users to a web page with instructions on how to authenticate. They suggest to use a common shared key for guest users, which is to be considered insecure against MITM attacks, and individual shared keys for registered users. However, they explicitly do not investigate generation and distribution of these individual shared keys or the use of certificate-based authentication and define it as out of scope of their work. In the present article, we focus on this key distribution problem and present an implementation similar to SWG as an example application making use of easy key distribution.

(a) USB dongles for sensing relative spatial positions can be added to typical off-the-shelf laptops, PDAs, or even mobile phones

(b) User interface for spatial selection

Figure 1: Current implementation of context authentication by spatial reference

3 Context authentication

Context authentication tries to provide intuitive means of authenticating users or devices by verifying that they are in some specific context, e.g. at some specific location. The possibilities for sharing context are obviously constrained by the sensors available to the involved devices. These sensors are used to verify some properties of the device to authenticate, i.e. to verify that the other device is in the same context. Context authentication then aims to create shared secrets for setting up secure communication, usually in the form of cryptographic key material.

In earlier work we reported on using *spatial reference* for authenticating spontaneous interaction [13] and on the security properties of our underlying localization method [12]. Selecting devices based on direct line of sight has also been explored with the "gesturePen" [18]. The gesturePen has the intention of selecting devices by pointing at them, in much the same way as we select devices based on their relative spatial position in this work. Location is just one option for context authentication, and for the description of additional options, we refer to others [8, 2]. In the present article, we investigate connections that are initiated in an ad-hoc manner but that might yield longer-lived security associations. Specifically, we establish IPSec connections on first contact, but continue to use these connections once established.

Building upon our current implementation [13], we assume devices to be equipped with sensors in the form of USB dongles. These dongles provide accurate sensing of relative spatial positions using ultrasound. Figure 1a shows two of them attached to a laptop and attached to a PDA — both can sense each others position with an accuracy of better than 10 centimeters in distance and 25° in angle [6].

In an office space with many laptops, PDAs, and other devices communicating over the same wireless network within a small area, this fine-grained sensing of shared context offers distinct advantages in selecting specific devices. If an administrator wants to allow "that device over there" to access the wireless network[†], then other devices in the same room should not automatically be allowed too. Solutions based on infrared connections can not easily provide such a fine-grained selection because infrared beams often span the whole area.

Our context authentication protocol integrates secure authentication transparently

[†]The same method can be used to allow access to private parts of the network, or, more generally speaking, to specific resources. We use access to the wireless network only as an example.

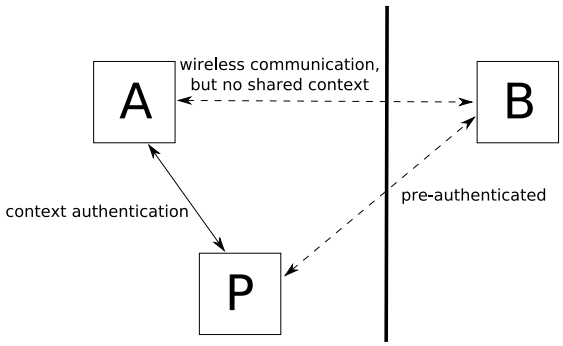


Figure 2: Using a context authentication proxy P allows physically separated devices A and B to benefit from context authentication even when they can not experience the same context

and seamlessly with device selection, as shown in Figure 1b. Simply by selecting a visualized device position, corresponding to the physical device as visible to the user, the authentication process is triggered. User interaction is thus changed from selecting devices from a network-discovered list to a spatially-discovered environment; authenticating selected devices happens automatically without further user interaction. This seamless integration makes the protocol well suited for spontaneous interaction. Security properties of our authentication protocol [13] and ultrasound as an out-of-band channel [12] have been discussed previously. Here we simply assume the protocol to provide a shared secret to both devices that perform the spatial authentication.

Although we build upon this specific authentication protocol for our demonstrative application, the concept of authentication proxies is independent of the underlying sensing platform for context authentication.

4 Authentication proxy

Previous work on context authentication assumes that those devices that authenticate each other can experience the same context, but this is not always possible. Figure 2 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 can not benefit from direct context authentication with it to secure her communication. By introducing a *context authentication proxy* P, we give her this option. 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.

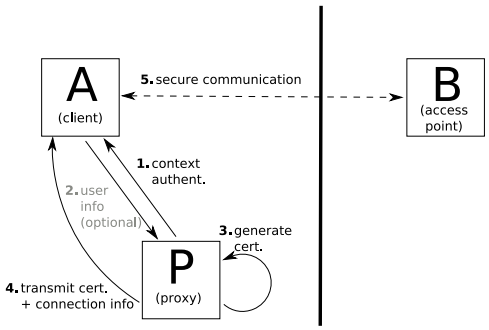
The main task of the authentication proxy is to create a shared secret between A and B, to enable secure communication between them over a wireless network. Depending on the initiator of the authentication, we can distinguish between two different approaches

for user interaction with the proxy:

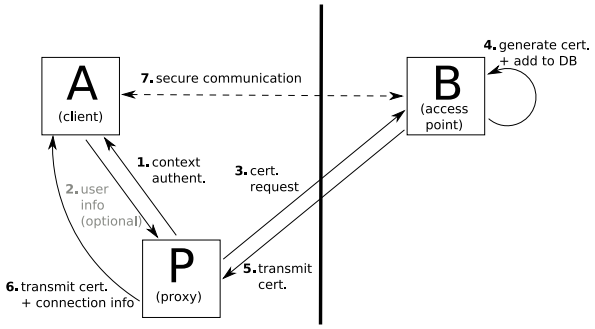
- We speak of a *passive authentication proxy* when P acts as an authentication service and simply waits for clients to initiate an interaction. The *client* takes the active role, starts context authentication with P to obtain a shared secret for communicating securely with B, and may need to engage in another authentication procedure with B over the now-authenticated wireless network. Instances of this approach are the closely related NiaB [1], and one of our previous works [14], which describes the use of RFID tags to secure communication over wireless ad-hoc peer-to-peer networks. The former requires a further offline authentication step performed by the in-house certificate authority when used for "enterprise" WLAN access, or relies on the infrared channel authentication for the simpler "home" WLAN setting. In the latter, we store public keys of network peers on associated RFID tags that can be read for secure spontaneous interaction. Note that in this previous work, we termed the RFID tags "objects" and the associated devices with which the interaction takes place the "proxies" because of a slightly different focus on the interaction.
- 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 authorization 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 can thus further decrease the burden placed on the user for setting up secure communication. We point out that the interaction between A and P, and subsequently between A and B, is still spontaneous. However, the change in roles relieves the client from going through additional steps after the initial context authentication and shifts this task to the proxy. P is in a better position to perform them, because it is part of the existing network and is thus assumed to know more about it than the new client.

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 authorization decisions to P. Without the ability to delegate trust, an active authentication proxy can still initiate the context authentication, but a subsequent authorization step might be necessary before A can access resources on B. In this case, the choice of authentication proxy should match the interaction style of the application, i.e. who initiates the spontaneous interaction. Note that arbitrary trust models can be used, including the sharing of passwords — which is clearly not recommended from a security point of view — and that most can be used to delegate trust in some way. A concept for delegating restricted trust over potentially multiple hops is described e.g. by Steffen and Knorr [17] and could be used in combination with context authentication proxies.

One secure and standardized option to delegate trust is to use X.509 certificates signed by a certificate authority (CA) managed by P and trusted by B. Every certificate that P creates and signs will be trusted by B, allowing P to make decisions about authorizing clients to use B's services. In this sense, our approach of a context authentication proxy is an implementation of the plug-and-play PKI [5]: a client device is automatically provided with an X.509 certificate that allows it to use some services. But instead of initially authenticating with the suggested username/password combination, we authenticate client devices based on context, specifically based on their relative spatial position. This makes the approach more usable for spontaneous interaction.



(a) The proxy implements the CA — no connection between the proxy and the access point is necessary



(b) The access point (or infrastructure) implements the CA — the proxy needs an online connection to request certificates for and forward them to the client

Figure 3: Two options for delegating trust with a context authentication proxy

5 Application for establishing IPSec connections

In this section we present *IPSecME* (IPSec made easy), an application to delegate trust for authorizing IPSec connections that uses an active authentication proxy and standard X.509 certificates. It uses our secure spatial authentication protocol described in earlier work [13] and does not depend on software being pre-installed on the client like NiaB.

5.1 Concept

Our IPSecME application can be used for setting up arbitrary IPSec connections by providing appropriate connection details in the form of an XML configuration file to the authentication proxy. IPSec tunnels over an otherwise open 802.11 WLAN are a practical example without loss of generality. For simplifying the discussion, we also assume the access point to act as an IPSec gateway, but it could be easily split into different devices without any change to our work.

IPSecME consists of two parts, one running on the client and one on the proxy device. Figure 3 shows two options for implementing this application using an active context authentication proxy P: the CA can either run directly on P, or it can run on the access point B (or any other infrastructure device). In the former case, B delegates

trust about authorization to P by allowing all clients A that present a certificate signed by the proxy's CA to establish IPSec tunnels. As illustrated in Fig. 3a:

1. P authenticates A via shared context, in this application via spatial reference.
2. A can optionally send information about the logged in user, the machine name, etc., if this should be encoded in the X.509 certificate.
3. P generates a new X.509 certificate with the information provided by A and/or locally entered data and signs it with its CA key. Note that the certificate is bundled with the matching private key.
4. P forwards the new certificate, the private key, its CA certificate, and details about the IPSec connection, i.e. the IP address of the gateway, the remote subnet, etc. to A. The private key is encrypted with the shared key generated in step 1.
5. A uses its new certificate and the IPSec connection description to establish a secure connection to B.

This option has the advantage that no online connection between the P and B is required. The trust between them is formed by B importing P's CA certificate. After this, no further communication between B and P is necessary for authenticating arbitrary clients[‡].

In the latter case, P requests certificates from the CA running on B using an online connection. As illustrated in Fig. 3b:

1. equal to step 1 in the former case
2. equal to step 2 in the former case
3. P generates a certificate request with the information provided by A and/or locally entered data and sends it to B.
4. B decides if A should be authorized and, if yes, signs the certificate request with its CA key and adds the new certificate to its authorization database.
5. B sends the new certificate to P.
6. equal to step 4 in the former case
7. equal to step 5 in the former case

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 authorization local. For spontaneous interaction, the first option has the advantage that no online connection between B and P is necessary, and we therefore implement this one.

5.2 Implementation

The implementation currently runs on a standard Laptop running Windows XP SP2 or Linux with any of the available IPSec implementations as the client A and a PDA running Pocket PC 2003 (or a laptop running any supported operating system) as the authentication proxy P. Because our context authentication protocol using spatial reference and the IPSecME application have been implemented in Java, other platforms can be supported fairly easily. All platform-specific parts,

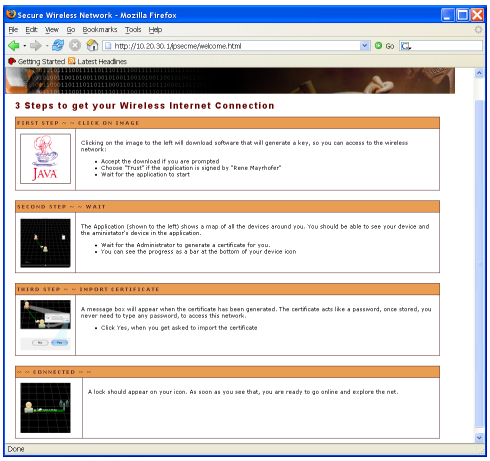
[‡]Note that revoking a certificate that P generated will require an update of its associated certificate revocation list (CRL) on B, and consequently communication between B and P. However, for spontaneous interactions, short-lived certificates can be used to alleviate the need for CRL updates.

i.e. managing certificates, establishing IPSec connections, and access to the ultrasound sensing devices, have been implemented for Windows XP, Linux, and Mac OS/X. The combination of access point and IPSec gateway, depicted as B in the above concept, has been implemented in two different versions. A standard access point connected to a PC running Gibraltar firewall [11] represents an enterprise scenario where the functionality of B is distributed in the infrastructure. An embedded implementation using the OpenWrt distribution [16] on an Asus WL-500GP access point represents the home/small office scenario with a single, combined device. Both implementations use Openswan [19] as IPSec implementation and ChilliSpot [7] to provide the captive portal. These two scenarios show that our approach can be used with arbitrary implementations of WLANs and IPSec gateways as long as they support external X.509 CAs.

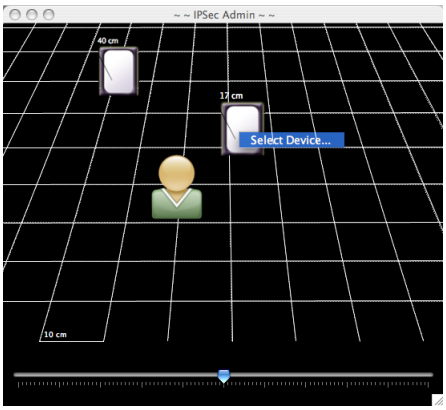
Figure 4 shows how users experience the whole process. The client does not need to have any special software pre-installed and does not need any a priori information about the environment. When it first connects to the WLAN, which is publicly accessible, its web browser gets redirected to a local web page in the same way as it is used by the currently popular WLAN hot spots (see Fig. 4a). From this web page, the user can start the client part of the application via Java Webstart and then simply waits for the proxy to initiate authentication. We assume that devices are either equipped with ultrasound sensing or that the USB dongles are attached at this stage. For ease of use, we skip the optional step 2 and omit to use client-provided information for generating the certificate. With spontaneous interaction, any such information tends to be meaningless anyway due to the lack of a globally accepted naming scheme. An administrator using the context authentication proxy can then select the client based on spatial reference (see Fig. 4b) and specify the validity period of the certificate and optionally a name describing the client for later use (see Fig. 4c). This name only needs to be meaningful within this environment, e.g. to the administrator. After initiating the context authentication protocol, the certificate is generated and signed automatically, and the IPSec connection details along with the certificate are sent to the client (see Fig. 4d). Note that the private key contained in the PKCS#12 format used for transmitting the certificate is encrypted with the shared secret that has been established between the client and the proxy during context authentication. Thus, they can communicate over the public, insecure WLAN without worrying about attacks. Finally, after receiving the certificate and connection details, the client can, when accepting them, immediately import its new certificate and establish the IPSec connection (see Fig. 4e and 4f). Further communication is automatically secured by the IPSec tunnel, which in our case includes all traffic to and from the client.

6 Experimental Evaluation

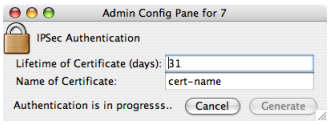
Although our spatial authentication method in general and our authentication proxy application for establishing IPSec connections in particular have been designed to make user interaction as intuitive as possible, they are new and to this time unknown to potential users. In contrast, users have already been trained to use existing, typically password-based methods to get access to WLANs. We



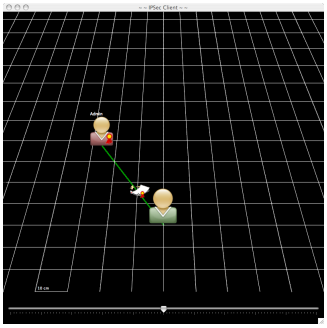
(a) **Client:** When first trying to access the network, the client is redirected to a page that delivers the client application



(b) **Authentication proxy:** selecting the client to authenticate based on spatial reference

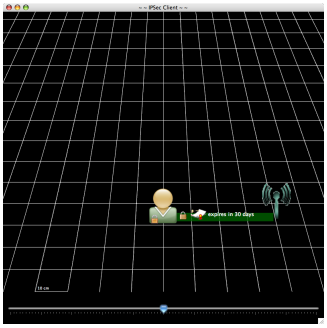


(c) **Authentication proxy:** after setting a name and the validity period for the new certificate, the client is authenticated



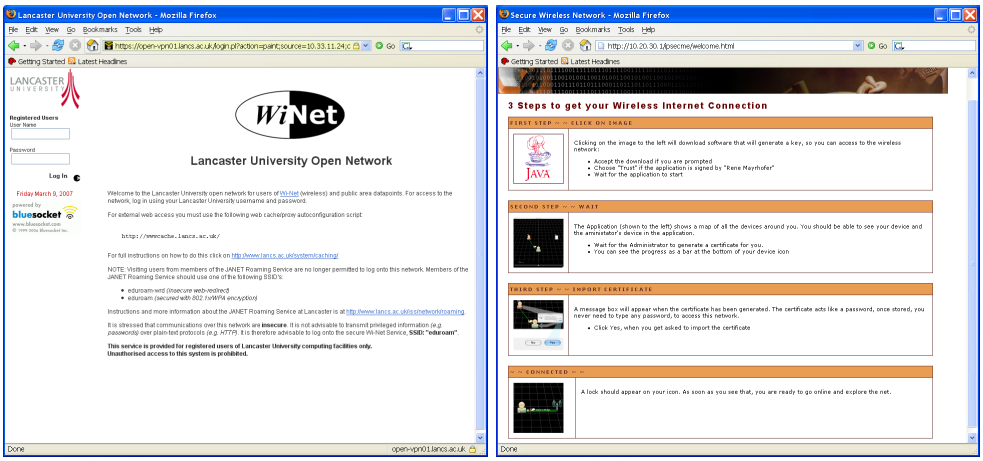
(d) **Client:** authentication in progress

(e) **Client:** accepting the new certificate and the CA of the IPSec gateway to establish the connection



(f) **Client:** IPSec connection successfully established

Figure 4: Screen shots of the IPSecME application



(a) Method 1: authentication with standard network accounts

(b) Method 2: authentication with IPSecME

Figure 5: Screen shots of respective captive web pages

therefore conducted two user studies to evaluate how end-users react to our method and to discover potential issues, and one informal study from the administrator point of view.

All subjects were office workers, either researchers from various fields or administrative staff in an academic environment. They generally had extensive experience with typical desktop applications and Internet usage, some also from a more technical point of view.

6.1 Study 1: Comparison to WLAN with captive portal and password authentication

Experimental design Our first study directly compared the end-user experience and satisfaction between a currently deployed solution and our IPSecME research prototype. Subjects were asked to get access to the respective WLAN with a standard laptop running Windows XP. In our study, neither of the variants assumed any a priori information to be shared between the subjects, who acted as guests in a new environment, and the WLAN environment itself. For simplifying the study, two laptops were used, one set to the WLAN ESSID of the first, the other to the ESSID of the second network. When opening the web browser (Mozilla Firefox) in the respective unauthenticated states, both displayed a captive web page with instructions on how to gain access to the network.

The aim of this study was to compare usability and end-user experience, and therefore subjects were not explicitly educated about the underlying principles and differences between the methods. Specifically, they were not told that the existing method only authorizes their laptop to access the network, while our IPSecME method additionally provides a secure IPSec channel for all IP connections.

Figure 5a shows the WLAN captive portal web page used at Lancaster University. Users can enter their normal network account details in the form of username and password to gain access to the network. Guests new to this environment would not have such an account, and therefore our subjects were asked to use one specific account and given the username and password (12 randomized characters, mixed upper and lower case letters and digits).

Figure 5b shows the captive portal web page as displayed to unauthenticated guests by our IPSecME WLAN gateway. This page simply allows to download the Java Webstart client application as shown earlier in Fig. 4. The Relate dongles were already plugged into the laptop, and subjects were told to follow the instructions on the web page. These instructions proved to be sufficient for subjects to use IPSecME for gaining network access.

For finishing the defined task of accessing the Google web page using both methods, the subjects required around 5 minutes on average. Measured variables were the number of errors, required time to read the captive web page, and time from starting the respective first step of authentication until successfully finishing it. For both methods, the investigator then closed the browsers and asked the subjects to perform the same task a second time, simulating subsequent network access, e.g. the next day of a visit. Variables were only recorded for the first access, the second aimed at examining user satisfaction. Finally, subjects were asked the following questions for both methods:

- I found the method easy to use for one-time access.
- I found the method fast to use for one-time access.
- I found the method easy to use for subsequent access.
- I found the method fast to use for subsequent access.

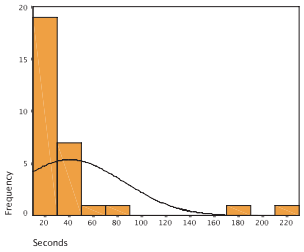
Answers to the above questions were a seven-point Likert scale with ratings from 1 ("strongly agree") to 7 ("strongly disagree"). Additionally, users were asked which method they liked more and which method they felt was more secure. On the next page, i.e. only after answering these questions, subjects were asked if:

- Were you aware that the IPSecME method provides encrypted connections? (Yes/No)
- Are you concerned about someone recording your Wireless Network usage (web sites, email)? (Yes/No)

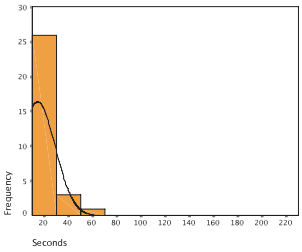
Results Due to the complexity of the whole process of gaining network access and the large underlying differences between the methods, study 1 was split into two phases.

A preliminary study with 15 subjects, 26.7% female, 73.3% male, was used to exploratively discover issues in user interaction and understanding, and to refine the exact study procedure and questionnaire so as to reduce any study bias towards either of the methods as far as possible. As a result, we were able to improve the user interface for making the underlying steps of IPSecME clearer. Especially the involvement and usage of the certificate is now visualized in multiple places, as this turned out to be an unknown concept to many users.

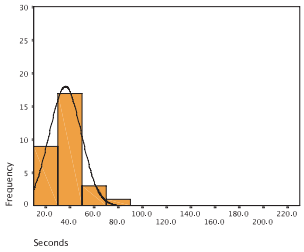
The main study was conducted with 30 different subjects (non-overlapping with the preliminary study), 40% female, 60% male. 80% had used the existing



(a) Password-based method: authentication



(b) IPSecME method: reading captive web page



(c) IPSecME method: authentication

Figure 6: Results: required times for gaining network access

Question	Access	Median existing	Median IPSecME	z	$p <$
"easy to use"	one-time	2	3	-1.21	0.226
"fast to use"	one-time	2	2.5	-0.81	0.42
"easy to use"	subsequent	2	1	-3.85	**0.0001**
"fast to use"	subsequent	3	1	-3.98	**0.0001**

Table 1: Results: rated answers to questionnaire and Wilcoxon Signed Rank test results

Lancaster WLAN before. For the existing Lancaster password-based method, 2 subjects made 1 error during entering the password and 3 subjects made 3 errors. For the IPSecME method, there were no user errors at all, but for 12 subjects the Relate authentication protocol failed due to distance measurement errors on the ultrasound channel and had to be repeated (we refer to [13] for a more detailed description of false negatives in the protocol). In this case, users (on the client) needed to acknowledge a dialog box stating that the protocol failed and the investigator (on the authentication proxy) restarted the device authentication.

Figure 6 shows the times people took for reading the IPSecME captive web page (with a mean of $\mu = 14.76$ seconds and a standard deviation of $\sigma = 14.58$ seconds) and to complete the respective authentication methods ($\mu = 40.95$ and $\sigma = 44.39$ for the existing, $\mu = 36.63$ and $\sigma = 13.27$ for the IPSecME method). For the existing password-based method, reading times were negligable due to the high familarity of most subjects. In the direct comparison, 15 subjects preferred IPSecME, 4 preferred the existing method, and the remaining 11 had no clear preference, but acknowledged advantages and disadvantages of both methods. Although common tests for significance can not be applied in this case, it is interesting to note that all 5 subjects who made errors during typing in the password for the existing method preferred IPSecME. 15 subjects felt that IPSecME was more secure (10 of which also stated that they preferred IPSecME), 9 felt that the existing method was more secure, and 6 could not decide. Without explicitly pointing it out, 14 subjects were aware of the fact that IPSecME provided encrypted connections after authentication, and 16 were not. 24 were generally concerned about anybody recording their wireless network usage when using insecure access methods.

For first time access, a Wilcoxon Signed Rank test does not show statistically significant differences for either of the questions (see Table 1: a rating of 1 means "strongly agree", 2 means "agree", and 3 means "slightly agree"). However, there are statistically significant differences for subsequent accesses, indicating that our subjects rated IPSecME higher than the existing password-based method for subsequent accesses both in terms of ease of use and of speed.

Discussion The general impression of study 1 is that, even though IPSecME is a new and unknown method and sometimes produces authentication errors that require a retry, our subjects were comfortable using it for the first time and rated it similarly to the existing method. For subsequent access, IPSecME is rated significantly better, which is unsurprising due to the automatic reconnects within the lifetime of the certificate, compared to the need for re-authentication on each access using the existing method.

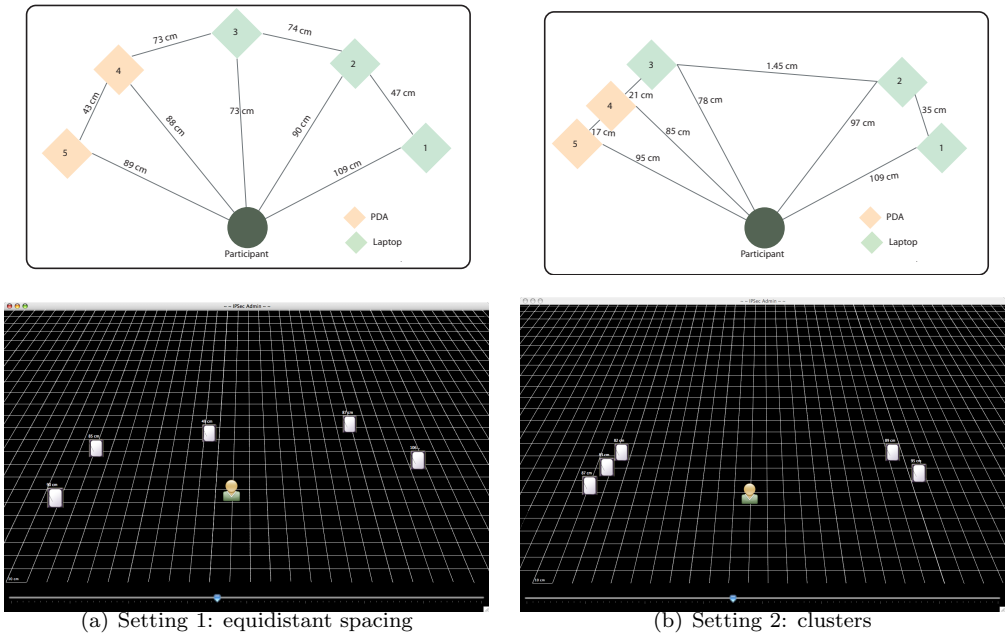
When accumulating reading and authentication times, IPSecME takes on average about 10 seconds longer, but it can be argued that reading the web page is a one-time task, while the authentication process itself is quicker. Our subjects seem to implicitly have taken this into account, as IPSecME was rated only slightly lower in terms of one-time authentication speed.

From additional, informal answers given by the subjects, we found that this convenience provided by installing a certificate on the client machine is seen as a major advantage. In the preliminary study, a few users expressed concerns about running additional software (the IPSecME client application) on their machines, while this did not appear as an issue in the main study. This is presumably due to improvements in the user interface to more clearly indicate what the client application does and how the certificate is being used.

6.2 Study 2: Selecting real-world devices with a spatial GUI

Experimental design The second study examines our method from the authentication proxy point of view and investigates how well people deal with our spatial selection method and user interface. 30 subjects were seated at a specific place in front of a meeting table and asked to use a laptop with 15" display, mouse, and attached Relate dongle for selecting different devices using our spatial user interface.

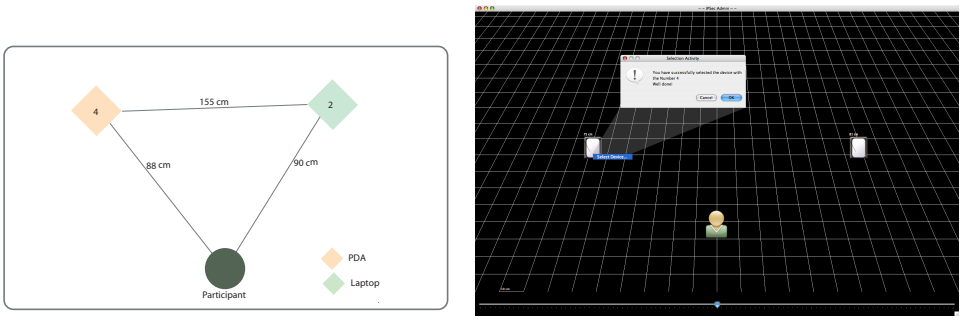
Figure 7 shows the two investigated settings with slightly different placement of the other 5 devices that were equipped with Relate dongles. Every device had its number printed on the case and clearly visible to the subject. To alleviate the influence of a training bias, an initial task used setting 1 for training purposes. With only devices 2 and 4 present, the procedure for the following tasks was explained: after the investigator mentioned a device number, the subject should select the corresponding device icon in the spatial user interface by right-clicking on it and then clicking on the pop-up menu item. Actions and errors were not recorded for the training task of selecting device number 4. Figure 8 shows the simplified placement of devices 2 and 4 and the spatial user interface as it was seen by the subjects.



(a) Setting 1: equidistant spacing

(b) Setting 2: clusters

Figure 7: Placement of devices on the table and in the spatial user interface



(a) Simplified setting 1 for training

(b) Spatial user interface during training task

Figure 8: Training task

To focus on the spatial selection, our user interface uses the same icon for all devices and does not show any identification information; the only textual information shown is the respective measured distance to the device. Therefore, different positions are the only distinguishing criteria. Informally, we discovered that subjects did not use the printed distances as an aid, but mostly the placement of the other devices relative to each other.

The task was for subjects to select 4 different devices, 2 in each setting. First they were asked to select device 2 followed by device 5 in setting 1, then device 1 followed by device 3 in setting 2. While changing the setup from setting 1 to setting 2 by slightly moving the devices, subjects were able to watch the laptop screen and follow the movement in the spatial user interface. Figure 7 also shows the respective spatial user interfaces.

An additional task was used to investigate correlations between the ability to estimate distances to and between real-world objects and perceived difficulties in mapping spatial relationships with our user interface. Subjects were asked to estimate the distances:

- between themselves and the door (2.60 m)
- the width of the door (1 m)
- between devices 2 and 3 (1.45 m)
- the width of the table (2.8 m)

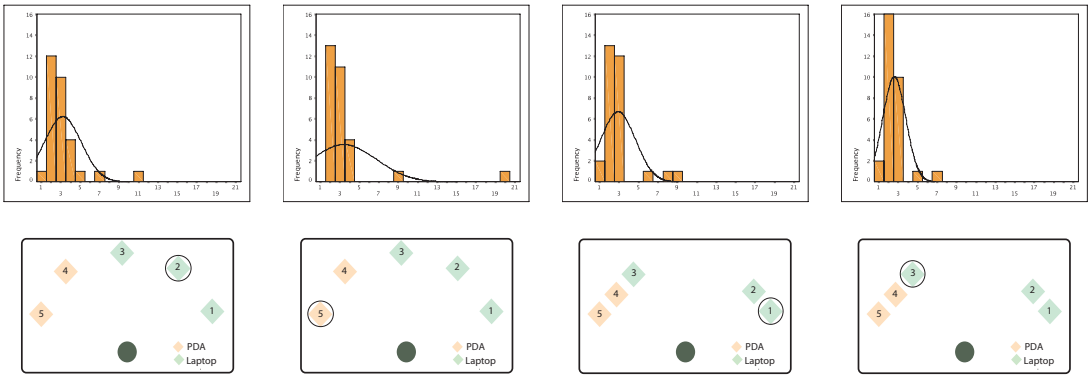
The absolute distances between the estimates given by the subjects and the real distances were accumulated for each subject. In addition, the investigator asked how easy the subjects found the mapping task (rating 1 to 7). The whole study took around 7 minutes per subject on average. Measured variables were the time from when the investigator named the real-world device until the subject right-clicked on the correct item in the spatial user interface and the number of errors until the correct device was selected.

Results 30 subjects, 23.3% female, 76.7% male, participated in the second study. 25 of these subjects are researchers in computer science, 1 is a researcher from a different area, and 4 belong to the University administrative staff.

Figure 9 shows the measured times the subjects needed to select the correct devices. Mean times for task 1 were $\mu = 3.2$ with $\sigma = 1.91$, for task 2 $\mu = 3.39$ with $\sigma = 3.34$, for task 3 $\mu = 2.89$ with $\sigma = 1.77$, and for task 4 $\mu = 2.58$ with $\sigma = 1.19$ seconds. The numbers of subjects making errors were, for each of the tasks, 2 (with 1 error per subject), 1 (the subject made 2 errors at this task), 1 (only 1 error), and 0, respectively, and the errors are fully disjoint, i.e. made by 4 different subjects. All 4 subjects who made errors answered that the tasks were easy to perform.

We performed three tests to examine statistical correlations:

- The hypothesis of a correlation between the subjects making any error during the mapping tasks and their answer to the question on ease of use was neither accepted not rejected with statistical significance. This is an expected results considering the generally low error rate.



(a) Task 1 (b) Task 2 (c) Task 3 (d) Task 4

Figure 9: Results: required times for device selection

- The hypothesis of a correlation between the accumulated absolute error of distance estimates and the the subjects making any error during the mapping tasks was also neither accepted not rejected with statistical significance.

- For the third test, the accumulated absolute errors were classified into two groups: $[0;1]$ m and $]1;\inf[$ m. A Mann-Whitney test shows a correlation with the answer to the question on ease of use with $U = 49.5$ and $p < 0.08$. Therefore, people who were better at estimating real-world distances found the task easier to perform, which matches intuition.

Discussion Due to the small error rate, we can not quantitatively characterize the errors. One likely influence seems to be a training effect, because two subjects made an error during task 1, one subject each in tasks 2 and 3, but there were no errors during task 4. There were two surprising findings: First, that task 2 took on average longer than task 1, which is contrary to any learning effect. One possible explanation is that the target device in task 2 was to the left of the subjects and most probably outside their primary field of sight when facing straight; they had to turn slightly to find it. Additionally, the device was smaller (a PDA instead of a laptop). Second, that setting 2 seemed to be generally easier for subjects than setting 1, although it contained partial occlusions of devices from the subject point of view. The most probable explanation is that people can more easily deal with clusters of small numbers than with a homogeneous group of a large number of devices. In setting 2, there are two clusters, 3 devices to the left and 2 to the right, and the numbers of devices are small enough so that people did not need to count for selecting the target device.

The single outlier with a large time in task 2 was caused by the subject who made 2 errors in the same task.

6.3 Study 3: Comparison from an administrative point of view

Another important aspect of wireless network access is its administration and management. Our third study examines IPSecME from an administrator's point of view. With informal demo and interview sessions, we explained our approach to the two network administrators responsible for the Computing Department at Lancaster University.

In an interactive questionnaire, both administrators stated that the currently deployed system was problematic for spontaneous guest access; although the creation of guest accounts, e.g. for meetings hosted at the department, was supported, it was a cumbersome and slow process that was unsuitable for spontaneous access. Standard practice is therefore for the hosts to either share their password or enter it at the guest's mobile device to grant access, both of which is questionable in terms of network security. An additional issue is that MacOS/X and Linux users are not well supported by the policy of periodic password changes. When not logging on to the Windows domain but only to the wireless network, passwords can not be changed and thus expire, forcing users to find a Windows client to re-gain wireless network access.

Although no clear preference for permanent, registered users has been mentioned, the administrators would prefer IPSecME over the current system for managing guest access. The major two reasons for this preference are security and spontaneous access. Security is improved by creating client-specific X.509 certificates on the fly for network access, and guests no longer need to use access credentials of registered users, which significantly improves accountability. Spontaneous access for clients is made easier by allowing local administrators, hosts of meetings and events, or secretarial staff to quickly grant network access while restricting it to specific guest devices, instead of having to interact with centralized authorization databases.

7 Discussion

The concept of an authentication proxy is generally applicable to arbitrary ways of authentication via shared context, and NiaB has already shown that the use of a special instance of an authentication proxy with infrared works well. It has yet to be investigated how well this concept integrates with other options such as cameras or microphones for sensing shared context. Our software [10] has been designed to make the context authentication protocol exchangeable. It is a simple task to change our application to use IrDA like in NiaB, for example, or to use something different like authentication over an audio channel [4] or with mobile phone cameras [15]. Even though practical applications have not yet made use of authentication proxies in those cases, we do not anticipate any major obstacles.

There are also other options for implementing the secure channel after successful authentication. In this work, we use the well-known IPSec protocol, but the different TLS suites, IEEE 802.1x, or IEEE 802.11i are also considered to be secure protocols and may be more appropriate for different application scenarios. Securing WLANs has been chosen as a scenario due to its practicality and wide applicability. By leaving the WLAN itself open and publicly accessible, we can

provide public services usable without authentication, and additional access to authenticated users. We already use this possibility to deliver the authentication application to new clients, thus making it unnecessary to require any pre-installed software. This combination of two (or multiple) levels of service is more difficult to achieve with IEEE 802.1x. For purely spontaneous interaction, IPSec transport connections can be used between just two hosts instead of tunnel connections for securing all traffic a host generates.

For trust delegation, there are again multiple possibilities. In our application, we rely on standard PKI techniques, but shared passwords, OpenPGP keys, or even hardware tokens are other examples that can be used with the same concept. The decision of using online or offline relationships between the service and the authentication proxy is also highly dependent on both the application and the trust model. If the trust model allows delegation of trust, then an active authentication proxy can have distinct advantages, especially when a wireless connection to the actual service is not available ubiquitously. The trust relationship then allows pre-authentication of a client to the service, via the authentication proxy, even before any wireless contact to the actual service is possible. This gives more freedom in performing the authentication, because it can be done at any time for later use. Our application demonstrates this by pre-authenticating IPSec connections for accessing a private network securely over an otherwise public WLAN or from the Internet. This use of IPSec connections is often termed "road warrior" support, because the home network can be accessed from anywhere.

The security of our approach builds upon three parts: First, our context authentication protocol is considered secure against known attack scenarios; it uses multiple rounds of an interlock protocol to verify that only a device at a specific relative position can successfully authenticate. Ultrasound sensing is used as a side channel for transmitting information, in a way that is tightly interwoven with the spatial relationship between devices and that prevents man-in-the-middle attacks on the wireless channel (see [13] and [12] for a more detailed analysis). Second, IPSec as a protocol for secure channels is currently considered as one of the most secure standards. Third, well-known PKI techniques delegate trust to the context authentication proxy. We explicitly point out that the security of our proposed use of authentication proxies relies on the physical security of the proxy devices; when attackers can access these proxies physically, they can access resources as defined by the respective trust model. This is not a new restriction — the security of most protocols relies on physical security of some of its components. An active authentication proxy, like the PDA in our example application, might be small and mobile, and thus even more care needs to be taken to protect it.

When comparing our method with others from a user point of view, we need to distinguish two different aspects. One is the ease of use for gaining access to a protected WLAN. Our user study presented in Section 6.1 shows that spatial authentication compares favourably even to a method already known to and used by the subjects, mostly because of forming a longer-lived security association that can be re-used for subsequent network access.

The second aspect is to establish secure IPSec connections, which is not supported by the standard password-based approach, and thus not currently used by most subjects. Due to this lack of real-world comparability, we only have anecdotal evidence that IPSec connection set-up is significantly eased by our use of an

active authentication proxy: For comparable security using e.g. the web administration interface of Gibraltar firewall, an administrator first needs to log in and navigate to the certificate management module (4 steps), create a new certificate for the client (10 fields in a web form), and download it. Then this certificate needs to be imported on the client machine (manual transfer of the file, e.g. with a USB storage device, followed by 14 steps under Windows XP) and an IPSec connection needs to be created (8 steps with the Windows XP wizard). In contrast, using our demonstration application, a new client needs to start the application (1 step, Fig. 4a), an administrator needs to spatially select the client device (1 step, Fig. 4b) and enter the certificate details (2 fields, Fig. 4c). After automatically transmitting the new certificate to the client and importing it, the user only needs to start the IPSec connection (1 step, Fig. 4e). Intuitively it seems clear that it is a considerable improvement over manual configuration. By explicitly assigning the authentication proxy an active role, the end user is relieved from dealing with the connection set-up details at all. This combines into a single step two tasks that are usually separate: the selection, often called *identification*, of a device followed by a proper authentication, and the *authorization* to use some service. We argue that only one step, namely deciding about authorization, is necessary from an administrator point of view and that the authentication step should be made implicit for spontaneous interaction to become viable.

It might become difficult to distinguish devices on the visualized map when too many are presented at once. However, in our user study this did not appear as a problem, and the issue would be implementation specific and is not inherent to the concept of an authentication proxy. We point out that the use of spatial reference for context authentication assumes the availability of appropriate sensors, either built into a device, or attached to it. For example in a meeting scenario, spatial reference is a generally useful tool [9] and using it for granting temporary access to resources – with the approach described in this article – thus integrates seamlessly. In other scenarios, ultrasound sensing might not be readily available for current mobile devices. Although our USB dongles make it easy to attach them, it is an additional step that needs to be done. But, as mobile devices begin to include more sensors, context authentication will be more easily possible in the near future.

8 Conclusions

In this article, we argue that context authentication is more intuitive then typical password- or certificate-based methods, especially for spontaneous interaction. The example of setting up secure WLAN connections shows clearly that these often-used methods do not scale with regards to the number of wireless connections used by a single person. A direct comparison between the number of steps that need to be executed by a user and an administrator for creating such a secure connection between a password-, a certificate-, and a context-based authentication procedure is obviously biased; our demonstration application has been designed specifically to make this as easy as possible, while other methods are usually not aimed at supporting spontaneous interaction. Nonetheless, practical experience shows that those WLANs where simple, spontaneous interaction is desired, such as WLAN hot spots in hotels or airports, either do not use any authentication at

all or tend to be seen as awkward by most users. Context authentication allows to provide secure wireless connections without demanding user attention "just for security". Our main contribution is the general concept of a context authentication proxy, which allows devices to use context authentication when they can not actually experience the same sensor values for any suitable aspect of context. A first demonstration application implements this concept for a prominent example, namely WLAN access. The fact that other projects have also approached this scenario shows the practical importance of the problem.

Compared to SWG, we benefit from the use of certificates to provide better security for larger scenarios, where re-keying of the whole system to disable access for a single client is not reasonable. We extend the results of the NiaB project in three areas: First, by making the context authentication proxy active, we give both the clients and the administrator more flexibility in the authentication process. By running a CA on the proxy, the decisions about authentication and authorization can be condensed into only one spatial device selection step to improve ease of use. Second, the proxy is made mobile and supports offline authentication where connectivity to the target network is not available. Third, ultrasound sensing provides more fine-grained selection of devices, and the same granularity is used in the spatial authentication protocol. This allows multiple devices in the same area to be distinguished better, e.g. to grant temporary network access in a meeting scenario with multiple laptops and PDAs on one desk. With an infrared channel like the one used in NiaB, there is no protection against active man-in-the-middle attacks. Therefore, the context authentication needs to be run in a secure environment where such attacks are prevented by organizational restrictions (e.g. that only one device is allowed to enter the authentication room at any time). With our proposed spatial authentication protocol, context authentication is secure even in public and untrusted environments.

Complete source code of our client and proxy implementations is available at `http://www.openuat.org/spatial-ipsec-proxy`, including configuration files for the gateway using Gibraltar firewall and using OpenWrt.

Acknowledgments

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

References

[1] D. Balfanz, G. Durfee, R. E. Grinter, D. K. Smetters, and P Stewart. Network-in-a-box: How to set up a secure wireless network in under a minute. In *Proc. 13th USENIX Security Symp.*, pages 207–222. USENIX, August 2004.

[2] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. NDSS'02: 2002 Network and Distributed Systems Security Symp.* The Internet Society, February 2002.

[3] A. Godber and P. Dasgupta. Secure wireless gateway. In *Proc. WiSE'02: 3rd ACM workshop on Wireless security*, pages 41–46. ACM Press, 2002.

[4] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human verifiable authentication based on audio. In *Proc. ICDCS 2006: 26th Conf. on Distributed Computing Systems*, page 10. IEEE CS Press, July 2006.

[5] P. Gutmann. Plug-and-play PKI: A PKI your mother can use. In *Proc. 12th USENIX Security Symp.*, pages 45–58, August 2003. published at `http://www.cs.auckland.ac.nz/~pgut001/pubs/usenix03.pdf`, shorter version appeared in IEEE Computer Magazine, August 2002.

[6] M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for co-located mobile devices. In *Proc. MobiSys 2005: 3rd Int. Conf. on Mobile Systems, Applications, and Services*, pages 177–190. ACM Press, June 2005.

[7] Jens Jakobsen. Chillispot web page. `http://www.chillispot.org`, 2006.

[8] T. Kindberg, K. Zhang, and N. Shankar. Context authentication using constrained channels. In *Proc. WMCSA: 4th IEEE Workshop on Mobile Computing Systems and Applications*, pages 14–21. IEEE CS Press, June 2002.

[9] G. Kortuem, C. Kray, and H. Gellersen. Sensing and visualizing spatial relations of mobile devices. In *Proc. UIST 2005: 18th ACM Symp. on User Interface Software and Technology*, pages 93–102. ACM Press, October 2005.

[10] R. Mayrhofer. Towards an open source toolkit for ubiquitous device authentication. In *Workshops Proc. PerCom 2007: 5th IEEE International Conference on Pervasive Computing and Communications*, pages 247–252. IEEE CS Press, March 2007. Track PerSec 2007: 4th IEEE International Workshop on Pervasive Computing and Communication Security.

[11] R. Mayrhofer and Esys GmbH. Gibraltar firewall web page. `http://www.gibraltar.at`, 2006.

[12] R. Mayrhofer and H. Gellersen. On the security of ultrasound as out-of-band channel. In *Proc. IPDPS 2007: 21st IEEE International Parallel and Distributed Processing Symposium*, page 321. IEEE CS Press, March 2007. Track SSN 2007: 3rd International Workshop on Security in Systems and Networks.

[13] R. Mayrhofer, H. Gellersen, and M. Hazas. An authentication protocol using ultrasonic ranging. Technical Report COMP-002-2006, Lancaster University, October 2006.

[14] R. Mayrhofer, F. Ortner, A. Ferscha, and M. Hechinger. Securing passive objects in mobile ad-hoc peer-to-peer networks. In R. Focardi and G. Zavattaro, editors, *Electronic Notes in Theoretical Computer Science*, volume 85.3. Elsevier Science, June 2003.

[15] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *Proc. IEEE Symp. on Security and Privacy*, pages 110–124. IEEE CS Press, May 2005.

[16] OpenWrt. OpenWrt web page. `http://openwrt.org`, 2006.

[17] R. Steffen and R. Knorr. A trust based delegation system for managing access control. In *Advances in Pervasive Computing: Adjunct Proc. Pervasive 2005*, volume 191, pages 1–5. Austrian Computer Society (OCG), April 2005.

[18] C. Swindells, K. M. Inkpen, J. C. Dill, and M. Tory. That one there! pointing to establish device identity. In *Proc. UIST '02: 15th ACM Symp. on User interface software and technology*, pages 151–160. ACM Press, 2002.

[19] Xelerance Corporation. Openswan web page. `http://www.openswan.org`, 2006.