

Authenticating public wireless networks with physical evidence

Tim Kindberg, James Mitchell, Jim Grimmett, Chris Bevan, Eamonn O’Neill

Abstract— Users of public Wi-Fi networks risk being tricked into connecting to ‘evil twin’ access points set up by attackers to launch man-in-the-middle attacks. We present a system which employs post hoc validation of an anonymous Diffie-Hellman key exchange undertaken as part of an 802.1X / EAP-TTLS network association process. Our system utilises an additional secure auxiliary channel to run a modified version of the Interlock protocol based on physical evidence in the network location. By using keying information generated during the network joining process, we allow spontaneous network users to detect man-in-the-middle attacks as well as avoiding the need for pre-shared keys. We report on implementations of our system which utilise physical evidence of authenticity in the alternative forms of public displays and 2D barcodes embedded in the environment and read by mobile phones.

Index Terms— Wireless LAN, security, authentication

I. INTRODUCTION

PUBLIC wireless (Wi-Fi) access points are becoming increasingly ubiquitous, yet they present significant security risks to their users. Many such locations offer encrypted connections, typically employing pre-shared keys using WEP or WPA-PSK. These methods offer some level of protection against passive eavesdropping. Alongside these link-layer security methods, public locations often employ web-based “captive portal” systems, which require users to supply identification credentials in order to provide access control and accounting.

However, public wireless networks very seldom provide a means by which the wireless access point (AP) can be authenticated to the user. As such, the user risks connecting to a rogue AP set up by an attacker. This rogue AP may offer a captive portal which appears identical to that provided by the location’s legitimate AP and may accept the same pre-shared key for link layer encryption. However, the rogue AP may download malware to the user’s device via its captive portal and may act as a man in the middle: relaying traffic to the legitimate access point, or to its own onward connection to the internet.

Manuscript received June 21, 2009. This work was supported in part by the UK Engineering and Physical Sciences Research Council under Grant EP/C547683/1.

T. Kindberg is with Hewlett-Packard Laboratories, Bristol, BS34 8QZ, UK. timothy@hpl.hp.com, +44 (0)117 312 9920.

J. Mitchell, J. Grimmett, C. Bevan and E. O’Neill, are with the Computer Science Department, University of Bath, Bath, BA2 7AY, UK. [jrm25 | j.w.grimmett | c.r.bevan | e.o.neill]@bath.ac.uk, +44 (0)1225 386 811.

Once a user has connected to a rogue AP, the attacker may eavesdrop on or tamper with her traffic, or present fake versions of well-known web pages e.g. to gather sensitive personal information. Such attacks occur in practice [2], and may even be mounted against residential networks. It is relatively straightforward to mount the attack using a conventional laptop loaded with software easily downloadable from the web.

Use of public Wi-Fi networks tends to be spontaneous: users connect to them opportunistically without prior planning. Therefore it is not reasonable to assume that users will have pre-loaded certificates or other data on their mobile devices with which to authenticate APs directly. Moreover, trusted third parties and browser-based protection mechanisms have limited value. An attacker can purchase an SSL certificate for a plausible domain name relatively cheaply. Most users will not check the URL of a captive portal or even have any firm basis for checking whether the URL is bona fide. Most users do not understand browser warnings about certificates signed by unrecognized third parties, and may simply click through warnings even if a certificate is self-signed [3].

Fortunately, the above issues can be overcome using the fact that public wireless networks are embedded in the physical world. More specifically, Wi-Fi is an example of a situated service [9]: it is provided in association with a physical venue such as a café or airport lounge, over a relatively short range. Our approach is based on proving whether a given wireless network does in fact belong in a physical sense to the venue, rather than whether it is provided by a certified entity. The proof, which the user runs through, is based on interaction with an artefact that is physically part of the venue.

Based on that approach, we present a system to improve the security of spontaneous connections to public wireless networks by allowing users to authenticate the access point they have connected to without requiring any pre-existing certificates, keys or trust relationships. Our contribution is to detect any rogue AP (including a man in the middle) via an interaction with an artefact embedded in the premises where the AP is based. For example, if the Wi-Fi network is provided in a café, then the protocol can exploit a large (e.g. LCD) display attached to the wall of the café. The user connects to the AP and then verifies its authenticity before using it. Our system is implemented using standard encryption and authentication techniques with slight modifications, and exploits a proven cryptographic scheme. Whilst we consider

the use case of a free public wireless network which any user can join, established methods of access control and accounting may trivially be added.

The remainder of the paper is organized as follows. Section II describes related work, including the Interlock protocol which forms the basis of our authentication protocol. Section III reports on the design of our solution, including the system model and protocol. Section IV describes the prototype implementation of our system. Finally in section V we examine the benefits and limitations of our system and suggest areas for further exploration.

II. RELATED WORK

Well-established methods exist to protect against connection to rogue APs. For example, 802.1X access control [6] coupled with certain EAP authentication methods [1] allow bidirectional authentication between the client and access point. Alternatively, a user could connect to an untrusted network and subsequently establish a VPN or SSH tunnel to a trusted endpoint. However, methods such as these typically involve pre-existing configuration or trust relationships, which typically do not exist in the context of spontaneous association with wireless networks.

Perhaps the closest problem to spontaneous AP authentication that has been tackled by the research community is that of creating a secure connection spontaneously between two devices – for example, to transfer a document securely between two mobile devices belonging to users who trust one another but who are in an untrusted environment. Many approaches to this problem involve a secure auxiliary channel in addition to the wireless network.

In one class of solutions, the devices themselves carry out the authentication over the auxiliary channel, which is used to transmit secret data or authenticating data. For example, the beam of a laser pointer may be used as a secret channel between devices for certain practical purposes [7], or a 2D barcode may be used to display authenticating material on one mobile phone which is read using the camera on the other device [12].

In another class of solutions, the auxiliary channel is used to provide evidence for humans to compare between the devices so as to validate the association between them. This involves a two-stage process:

1. The devices establish a shared secret key over the insecure wireless channel; secret keys are generated so as to be unique to the run of the key exchange protocol, e.g. as a result of using Diffie-Hellman key exchange.

2. The resulting shared secret keys on each device are compared using the auxiliary channel, in order to expose any man in the middle. If there is a man in the middle, then the keys exchanged with each device will be different and the comparison will demonstrate that.

There are various mechanisms for key comparison over the auxiliary channel. Perhaps the simplest is to hash each key and represent it as a digit-string for comparison by humans. This includes for example the MANA protocols [5]. Alternatively, the hashes may be represented in visual form as images

synthesised from them [13], or represented in multimedia form such as blinking lights and sounds [7].

However, one problem with the foregoing techniques is that the data being compared is not meaningful to, or chosen by, the users. Since it is not meaningful, it is relatively hard to compare accurately. Moreover, as our studies of users with authentication protocols have demonstrated [8], users consider authentication mechanisms to be more secure if the data is chosen by them, and this may be an important factor for the protocol's acceptance.

Fortunately, a protocol exists that enables secure key comparison in order to detect a man in the middle, and also allows the user to choose the data to be compared between the two devices. This is a prior solution by a team including one of the authors [10] which in turn is a variation on the Interlock protocol of Rivest and Shamir [14]. This solution involves the comparison of keys and thus authentication of a secure association by comparing an arbitrary message chosen by the user of one device and displayed on the other device. However, the protocol is not directly applicable to the case of authenticating wireless access points. First, it assumes that both devices have a display to render the message, which is plainly not the case with a wireless AP. Second, the particular formulation in [10] assumes a human operator of each of the two devices concerned. Again, this is not the case with a wireless AP. Third, the man in the middle is not the only attack in the case of a wireless AP: someone with a single network interface could still construct a fake access point and use it to launch a malware attack. Such an attacker could not be detected by key comparison, since only one key would be involved. Lastly, the Interlock protocol used in [10] is difficult for users to understand when transcribed literally into user interactions, as we discovered from trials [8].

We have devised a variation on the protocol in [10] that addresses all the foregoing issues and we now describe it.

III. DESIGN

A. System Model

The scenario we consider involves a user, Alice who wishes to use her personal wireless device A to connect spontaneously to a public wireless network in a location, such as a café, that she has not visited before. The café offers a Wi-Fi network through an infrastructure mode access point, B. The café has no restrictions on who may join the network, and other users may be attempting to join it at the same time as Alice, or have already joined the network previously.

Our goal is for Alice to establish an authenticated connection from her mobile device A (e.g. a laptop) to the access point B. We consider a threat model in which an attacker Mallory may be operating another access point M, which may itself be associated with the legitimate access point B. To Alice this access point M may appear indistinguishable from the access point B. If Alice connects her device A to Mallory's access point M, her network traffic can be eavesdropped or altered before being relayed to or from B; M may launch malware attacks via its captive portal page, even if

no connection is provided onto the internet.

Our system aims to provide Alice with evidence that she has in fact connected her device A to the legitimate access point B, rather than Mallory's access point M. To achieve this, B is connected to a display device D. Display D must be visible to Alice, physically associated with the fabric of the café, and securely connected to B. For example, D is a large public display firmly bolted to the wall of the venue and connected by a secure cable to the café infrastructure. We discuss implementation choices in more detail in Section IV.

B. The Display Interlock protocol

We first describe the Display Interlock protocol, an adaptation of the Interlock protocol of Rivest and Shamir [14], between Alice's mobile device A and the access point B. In the following subsection, we will explain the user interaction between A and D that constitutes the auxiliary channel by which authentication takes place. For now, we assume only that D can display any data instructed by B. Recall that D and its connection to B is secure, so that no attacker may instruct D to display data or alter the data instructed by B or displayed by D.

The goal of this protocol is for A to establish a shared secret key at the time of joining the network and to verify, before proceeding to transmit or receive data over the network, that the key has been established with B and not with M. Once this validation has taken place, A will encrypt all traffic with the secret key.

1. Using a symmetric key exchange protocol such as Diffie-Hellman, A establishes a key K_{AB} , supposedly with B. Whichever protocol is used, it must have the unique key property of Diffie-Hellman: the probability that the same key will be generated by any two runs of the protocol is vanishingly small.
2. By the same protocol, B establishes a key K'_{AB} , supposedly with A.
3. A chooses a random plaintext message P , from a set of N possible messages.
4. A generates the ciphertext $C = \{P\}_{K_{AB}}$, using a symmetric cipher which has the quality that partial ciphertexts cannot be decrypted.
5. A splits the ciphertext C into two non-empty parts C_1 and C_2 : $C = C_1 \oplus C_2$ where $C_1, C_2 \neq NULL$. The original Interlock protocol was based on a single block of ciphertext, however we need to support messages that may occupy several blocks when encrypted. The operation \oplus denotes blockwise composition from half-blocks (also used in [11]), i.e. C_1 consists of the first half of each block of the ciphertext appended in sequence, and C_2 consists of the second half of each block of the ciphertext appended in sequence.
6. A sends C_1 to B.
7. A waits until D displays an indication of receipt of a message by B. If no such indication appears after a timeout of a few minutes (as determined by A), then validation fails.
8. A sends C_2 to B.

9. Let C'_1 and C'_2 be the two messages received by B, and the blockwise concatenation of these be $C' = C'_1 \oplus C'_2$.
10. B decrypts C' with its key K'_{AB} , to obtain $P' = K'_{AB} \{C'\}$.
11. B instructs D to display P' .
12. If no message P' appears on D after a timeout of a few minutes, then validation fails. Otherwise, Alice compares P' with her original plaintext message P . If $P' \neq P$ then validation fails. If, conversely, $P' = P$ then validation succeeds.

To see how this protocol exposes an attack, consider the case where A has in fact established a key with Mallory rather than B. If M has no network connection to B, then no indication of receipt of the first part of the ciphertext will appear on D in step 7, and the validation will fail. Otherwise, let us assume that M does have a connection to B and that M has exchanged a key K_{MB} with B and a key K_{AM} with A, with $K_{MB} \neq K_{AM}$ due to the unique-key property of the exchange protocol.

If Mallory simply forwards the two blocks of ciphertext, then B will obtain garbage when it decrypts the combined ciphertext in step 10, since B possesses K_{MB} but it was encrypted with key K_{AM} . The displayed message will therefore not match the original message at step 12, and validation will fail.

Since the first ciphertext message C_1 consists of only half-blocks of the original ciphertext, Mallory cannot decrypt it. Moreover, the messages are freely generated by the user or chosen from a sufficiently large set so that Mallory cannot feasibly compute all the possible ciphertexts in the short time available to him (the key is not known in advance of the protocol run). If the block size is chosen to be sufficiently large – at least 128 bits – then neither does he have time to guess the other half of the ciphertext by exhaustive enumeration and decryption.

If Mallory waits for the second half of the ciphertext, no indication of receipt of the first half will occur at step 7, so validation will fail. Therefore Mallory must guess the message blindly, encrypt it with K_{MB} and forward half of the resulting ciphertext. The probability that his guess will succeed is $1/N$, where N is the size of the set from which P was chosen. This can be made arbitrarily small by choosing a suitably large value for N .

If the messages match in step 12 and validation therefore succeeds, then Alice may either go on to use the network on the grounds that the level of validation is sufficient (that the probability $1/N$ that there is a man in the middle is sufficiently small) or to repeat steps 3-12 with fresh choices of message as many times as she wishes, in order to decrease the probability that she has connected to M rather than B.

Once validation is complete, all traffic between the user's device and the AP is fingerprinted and encrypted with the key chosen in the first stage, so that Alice cannot be subjected to eavesdropping or injection attacks.

Finally, as described, the protocol supports at most one user at any one time. Should several users participate concurrently due either to a malfunctioning implementation or to Mallory's contrivance, then, by the unique key property, no false

positive could result. However, this could result in denial of service. That is, validation may fail even for a user who is in fact connected to a *bona fide* access point.

C. Using the system

On arriving at the café, Alice switches on her device and scans for available wireless networks. Based on the capabilities advertised by the various access points, the device will assist Alice in her choice of SSID by reporting that enhanced authentication may be available. Alice chooses the SSID matching the café and joins the network without the need for credentials or keys.

Once the device A has successfully joined the network, the Display Interlock protocol is offered to Alice. When she selects it, Alice is presented with a GUI application on her device which features an avatar. The avatar is provided by the AP, and is tied to Alice's user session. This allows Alice to distinguish her protocol run on the display D from other users who may be joining the network at the same time.

Alice's device A exchanges a secret key as described in the protocol in III.B. It then instructs her to choose a message at random. Once she has chosen a message, she is asked to check the public display D to see whether her avatar appears there to confirm that the system is ready for her message. When Alice sees the confirmation from her avatar on display D, she confirms this on her device A. Having received confirmation, Alice's device shows the message she chose whilst the display D shows the decrypted message it has received. Alice is asked to compare the two messages. If the messages match, then Alice has authenticated her connection from A to B with the probability specified in the protocol. This varies according to the exact means by which Alice chose her message. If they do not match, or if no message is displayed on the display D, then the connection cannot be authenticated, and device A has joined an access point other than B.

Whilst it is not made explicit to Alice, the interaction above supports a run of the Display Interlock protocol. After Alice chooses her message, the device A encrypts it using the keying material resulting from the network joining process (which is shared with the access point). It splits the resulting ciphertext into two parts, and sends the first part automatically to the AP. In order for the protocol to succeed, her device A must not send the second part of the split ciphertext until the AP B has received the first part. In the user interaction this corresponds to Alice's device A instructing her to wait and to confirm when the avatar on display D is "ready". On receiving this confirmation from Alice, device A sends the second part. The AP B then recombines the ciphertext, decrypts it with the keying material that relates to Alice's user session, and sends it to D for display.

Note that Alice experiences this interaction as the following sequence: "choose the message, wait until the Avatar on display D is ready to receive it, confirm when it is ready, then view the message it displays and compare it with the original message". This sequence is chosen to be broadly comprehensible to the user, and can be explained in such terms as "only the café's genuine network will be able to

repeat your message correctly onto the café's display". It is not explained why the avatar may not be immediately ready, but users are used to waiting while processing of various kinds takes place. A prior implementation in which we transcribed the protocol literally up to the level of the user interaction was found to be largely incomprehensible [8]. It is very difficult for ordinary users to understand why their message should be split into two and sent in stages.

The Display Interlock protocol, as described, supports at most one user at a time since messages from different users cannot be distinguished. It is straightforward to adapt the protocol to support several concurrent users, each using a distinct avatar. However, the appearance of several active avatars responding to users on the display D may cause confusion. The simplest approach to avoiding this confusion is for at most one user to run the protocol at one time, and for the avatars of users who are waiting to run the protocol to be placed in a graphical queue on display D until it is their turn. Their devices inform them to wait until their avatar reaches the front of the queue before they may proceed.

IV. IMPLEMENTATION

Our system operates in two stages. Firstly the user joins the wireless network and keying material is generated to support an encrypted connection to the access point. Once this stage is completed, we perform an authentication process which presents the user with evidence that their device and the legitimate access point are (or are not) in possession of the same keying material.

To avoid confusion, whilst we refer to (and use) authentication methods in the network joining stage, we note that no real authentication of the user or the network takes place until after the user has connected to the network.

In the spontaneous use case we wish to support, neither the user nor the network has any prior information about the other on which to base an authentication decision.

A. Network joining stage

To support spontaneous secure connection we use 802.1X access control and EAP-TTLS authentication [4] (with tunnelled EAP-MD5/Challenge) through a secure supplicant on the user's mobile device. This choice of authentication method fulfils two main requirements. Firstly, session-specific keying material can be generated using an anonymous Diffie-Hellman key exchange, which is then used to support encryption between the client and access point. Secondly, the tunnelled EAP-MD5/Challenge authentication allows the user to specify an identity (chosen at random by the supplicant). This tuple of user identity and keying material is exported for subsequent use in our authentication application.

On the infrastructure side, access points typically have restricted functionality. Therefore we assume that B in our protocol is implemented as an 802.1X-capable access point connected to an authentication server running *freeRADIUS*. Display D is connected securely to the authentication server, rather than the access point. To support our requirements for network joining, *freeRADIUS* is configured to allow only

EAP-TTLS / EAP-MD5/Challenge authentication (granting access to any user identity using that protocol), and mandates anonymous Diffie-Hellman key exchange in the TLS handshake of EAP-TTLS. Another application on the authentication server watches the activity of *freeRADIUS*, and as new users successfully join the network, their user identity and keying material is exported to a database which supports the subsequent authentication phase. At this stage a pictorial avatar is assigned to each new user. To avoid confusion and possible spoofing attacks, we enforce a minimum time period within which a given avatar will not be re-used.

On the client, network joining is provided by a patched version of the *wpa_supplicant* application running on Linux. Our modified supplicant broadcasts a Probe Request to discover access points with the necessary capabilities for our authentication service to be present. On finding one, the supplicant connects using an identity chosen at random. If the connection is successful, the supplicant further checks for the presence of our authentication server on the network. If the server is found, the supplicant application launches the *authentication client* application, and exports the chosen user identity and keying material.

B. Authentication

After the user has successfully joined the network, the user's authentication client and the network's authentication server are both in possession of the user's identity and the session-specific keying material, which is unique to each run. The purpose of the authentication stage is to demonstrate to the user that both her client and the authentication server hold the same information.

In addition to the information exported from the supplicant, the authentication client communicates with the authentication server over HTTP. In making requests to the authentication server, the client identifies itself using the user's identity.

Once it is launched, the authentication client requests the visual avatar which was previously assigned by the server to the user's identity. In the client application, this avatar appears throughout the interaction and provides the user with instructions to guide them through the protocol.

Once the user has chosen a message the authentication client encrypts the message using the session keying material exported from the supplicant. We use the AES cipher in cipher-block chaining (CBC) mode. The resulting ciphertext is split into two parts (following the procedure defined in step 5 of the Display Interlock protocol) and the first part of the ciphertext is sent to the authentication server in the background without user intervention, as explained above.

Once the message has been sent, the user is instructed to look at the display D. The server instructs display D to show the user's avatar along with a message "Not yet ready to receive your message – please wait" until the first ciphertext part is successfully received. Once it is received, display D alters its display to read "Ready for your message – please send it". It is essential that the user delays sending it until the Avatar indicates a state of readiness – that is, until the first part of the ciphertext has been received by B. To mitigate

against the possibility of a user "clicking through" this stage in the client GUI without reading the instructions or checking the display D, we implement a time delay in the client GUI during which time the controls are not visible.

Once both parts of the ciphertext have been received by the authentication server they are combined according to step 9 of the protocol and the complete ciphertext is decrypted using the keying material stored for the user session, as exported in the network joining stage. The decrypted message is then shown on display D, along with the user's avatar. At the same time, the client GUI application displays the message that the user chose, and instructs the user to compare the two messages and confirm whether or not they match. If the user confirms the match, a dialogue asks here whether she would like to choose another message to strengthen the validation or continue to use the network. If she indicates a mismatch, the client application informs her that validation has failed.

C. The display

In the initial user trials with the Display Interlock protocol, we used a large (42") LCD display fixed to the wall of the café where the trials were held [8]. This configuration is plainly authoritatively attached to the premises, and therefore provides good evidence of authenticity. Our trial users confirmed that this was also their perception. Many venues have a public display anyway, so it does not necessarily add expense. However, the use of a public display has limitations: it may not be visible from all parts of the premises; even if it is visible, it may be difficult to compare text accurately between a local device display and a remote wall-mounted display; a public display such as this cannot easily support concurrent users; and the owners may prefer to avoid a large display for reasons of expense or space. More subtly, the public nature of the display may not be congruent with users' desire for privacy and their existing mental models of security protocols, where secrecy is typically an important element [8].

We are currently testing an alternative implementation of the system in which the large public display is replaced by the screen on the user's mobile phone. For the purposes of this description, the target device that the user wishes to connect to the network is assumed to be a laptop or other device in addition to the mobile phone, rather than the mobile phone itself. Issues of expense, comparability and privacy are removed with this configuration. The user can hold the mobile phone next to the target device's display for accurate comparison.

However, we must bind the phone's display to the premises in such a way that the binding is secure and seen by the users to be secure – as it is in the case of a public display attached to the wall. To achieve this, we use a 2D barcode such as a QR code. This may be printed or electronically displayed in such a way as to be firmly attached to the venue so that an attacker could not substitute another barcode without detection – for example, it could be on a poster behind the bar but in read-range of customers. Alternatively, it may be handed to the user by a member of staff on a till receipt or a card from a stack in the till drawer. Using portable media as in the latter case (but

from a trusted origin) enables the user to read the barcode in relatively discreet circumstances at her table.

The QR code acts as a physical hyperlink – the user reads a URL from it with her mobile phone camera, which causes her browser to connect to a website that manages the authentication process for the particular venue. In this implementation, we rely on the user’s cellular network provider to provide a secure SSL session with the café’s authentication server. The mobile phone configuration is as secure as the public display configuration. A properly anchored barcode (on the till receipt for the coffee that the user is drinking, say) provides about as much security as the bolts holding a display to the wall. In each case, the effort required to substitute a fake artefact without being detected would be considerable. The SSL connection between the mobile phone and the authentication server is similarly secure to the connection between a wall-mounted display and the authentication server.

Since the QR code is not specific to her user session (but is specific to the café location), the user is directed to a page showing all of the user avatars waiting to carry out the authentication process in that café. She selects the avatar that matches the one shown in her target mobile device, from which point the protocol continues as described previously, with the mobile phone browser taking the role of the display screen D.

V. DISCUSSION

We have described the Display Interlock protocol for authenticating a wireless access point. More precisely, the protocol enables a user to validate a connection by verifying, to within an arbitrarily small probability, that a secret key is shared with an access point provided by the venue. The evidence for this is a physical link between a display and a part of the physical premises that can reasonably be expected to be secured by the management. We have described two configurations of display and physical linkage: a public display fixed to a wall in the venue, and the user’s mobile phone connected to the venue’s authentication service by reading a barcode printed or displayed with a robust association to the venue, such as on a till receipt or card handed to the user by a member of staff from behind the counter.

The Display Interlock protocol enables spontaneous connections (requiring no preconfiguration), which is an important requirement for accessing public wireless networks. In our trials with the public display configuration, we found that the evidence of authenticity was convincing to users. However, we have proposed the mobile phone configuration to address the usability and acceptance issues caused by the public display. We are engaged in further user trials with the mobile phone version to measure its usability and user acceptance in terms of perceptions of its security and other factors arising in its use.

REFERENCES

- [1] Aboba, B., Blunk, L., Vollbrecht, J. Carlson, J., and Levkowitz, H., Extensible Authentication Protocol (EAP), Request for Comments 3748, Network Working Group, <<http://www.ietf.org/rfc/rfc3748.txt>>, 2004, accessed 8 Oct. 2008.
- [2] Bugzilla@Mozilla – MITM in-the-wild. https://bugzilla.mozilla.org/show_bug.cgi?id=460374.
- [3] Dhamija, R., Tygar, J. D., and Hearst, M. Why phishing works. In Proc. CHI 2006. ACM Press (2006), 581-590.
- [4] Funk, P. and Blake-Wilson, S. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0), Request for Comments 5281, Network Working Group, <<http://www.ietf.org/rfc/rfc5281.txt>>, 2008, accessed 8 Oct. 2008.
- [5] Gehrmann, C., Mitchell, J. and Nyberg, K. Manual authentication for wireless devices. RSA Cryptobytes 7(1), (2004), 29-37.
- [6] IEEE LAN / MAN Standards Committee, IEEE 802.1X-2004 IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control, IEEE Standards for Information Technology, <<http://standards.ieee.org/getieee802/download/802.1X-2004.pdf>> 2004, accessed 8 Oct. 2008.
- [7] Kindberg, T., and Zhang, K. Secure Spontaneous Device Association. Proc. UbiComp 2003, Seattle, USA, October 2003.
- [8] Kindberg, T., O’Neill, E., Bevan, C., Mitchell, J., Woodgate, D., and Grimmett, J., Authenticating ubiquitous services: a study of wireless hotspot access. To appear in proc. Ubicomp 2009.
- [9] Kindberg, T., O’Neill, E., Bevan, C., Kostakos, V., Stanton Fraser, D., and Jay, T. Measuring trust in wi-fi hotspots. In Proc. CHI 2008. ACM Press (2008), 173-182.
- [10] Kindberg, T., Zhang, K., and Im, S. Evidently secure device associations. HP Labs tech report HPL-2005-40.
- [11] Mayrhofer, R., and Gellersen, H. Shake well before use: Authentication based on accelerometer data. In Proc. Pervasive 2007, Springer-Verlag, May 2007, 144-161.
- [12] McCune, J.M., Perrig, A., and Reiter, M.K. Seeing-is-believing: using camera phones for human-verifiable authentication. In IEEE Security and Privacy, (2005), 110-124.
- [13] Perrig, A., and Song, D. Hash visualization: A new technique to improve real-world security. In Proceedings of the Workshop on Cryptographic Techniques and E-Commerce (CrypTEC), pages 131–138, July 1999.
- [14] Rivest, R., and Shamir, A. How to expose an eavesdropper. Communications of the ACM, 27(4), (1984).