Standards for security associations in personal networks: a comparative analysis

J. Suomalainen*

VTT Technical Research Centre of Finland,
P.O. Box 1000, FI-02044 VTT,
Espoo, Finland
E-mail: Jani.Suomalainen@vtt.fi
*Corresponding author

J. Valkonen

Department of Information and Computer Science,
Helsinki University of Technology,
P.O. Box 5400, FI-02015 TKK,
Espoo, Finland
E-mail: Jukka.Valkonen@tkk.fi

N. Asokan

Nokia Research Center,
P.O. Box 407, FI-00045 Nokia Group,
Helsinki, Finland
E-mail: N.Asokan@nokia.com

Abstract: Introducing a new device to a network or to another device is one of the most security critical phases of communication in personal networks. It is particularly challenging to make this process of *associating* devices easy-to-use, secure and inexpensive at the same time. A cornerstone of this process is key establishment. In this paper, we first present a taxonomy of protocols for key establishment in personal networks as well as describe and analyse specific protocols. We then use this taxonomy in surveying and comparing association models proposed in several emerging standards from security, usability and implementability perspectives.

Keywords: networks; security; personal networks; security association; standards; Bluetooth; Wi-Fi; WUSB; HomePlugAV; comparative survey; attacks.

Reference to this paper should be made as follows: Suomalainen, J., Valkonen, J. and Asokan, N. (2009) 'Standards for security associations in personal networks: a comparative analysis', *Int. J. Security and Networks*, Vol. 4, Nos. 1/2, pp.87–100.

Biographical notes: Jani Suomalainen is a Research Scientist in VTT Technical Research Centre of Finland where he has been working since 2000. He received his MSc Degree from Lappeenranta University of Technology in 2001 and is currently a postgraduate student at the Helsinki University of Technology. His research interests include information security in the areas of home and spontaneous networks as well as mobile and embedded devices.

Jukka Valkonen received his MSc Degree in 2006 from Helsinki University of Technology, where he has been affiliated with Laboratory for Theoretical Computer Science since 2005. He started his postgraduate studies in 2006.

N. Asokan is a Principal Scientist with Nokia Research Center in Helsinki. He has been conducting research in building secure systems for over ten years. He received his doctorate in Computer Science from the University of Waterloo. He has been working in the IBM Zurich Research Laboratory and served as a professor at the Helsinki University of Technology. His research interests include cryptographic techniques to design secure protocols for distributed systems, the use of Trusted Computing technologies, and ways to make secure systems usable.

1 Introduction

Short-range communication standards have brought a large number of new services to the reach of ordinary users. For instance, standards for personal networking technologies such as Bluetooth,[1] Wi-Fi,[2] Wireless Universal Serial Bus (WUSB),[3] and HomePlugAV[4] enable users to easily introduce, access, and control services and devices both in home and mobile environments.

The initial process of introducing a new device to another device or to a network is called an *association*. Association consists of the participating devices finding each other, and possibly setting up a *security association*, such as establishing a shared secret key, between them.

The part of the association procedure that is visible to the user is called an *association model*. Association models in today's personal networks such as those based on Wi-Fi or Bluetooth, typically consist of the user scanning the neighbourhood from one device, selecting the other device or network to associate with, and then typing in a shared passkey. These current association procedures have several usability and security drawbacks arising primarily from the fact that they are used by ordinary non-expert users. First, when there are many devices or networks in the scanned neighbourhood, users find it difficult to choose the correct one from a, possibly long, list of choices. Second, the security of the association protocol depends on the strength of the shared passkey. Making the passkey long and hard-to-guess impacts usability. Using a short or memorable passkey leaves the protocol vulnerable to dictionary attacks, even by passive eavesdroppers. Also, over the last few years several other cryptographic weaknesses have also been discovered in the association protocols used in Wi-Fi and Bluetooth.

To address these concerns, various new ideas have been proposed with the intent of providing a secure yet usable association model. For instance, there have been proposals for key establishment schemes utilising short passwords/checksums (Čagalj et al., 2006b; Gehrmann et al., 2004; Larsson, 2001; Laur et al., 2005; Vaudenay, 2005; Zimmermann, 1996) or various types of Out-Of-Band (OOB) channels (Balfanz et al., 2002; McCune et al., 2005; Saxena et al., 2006; Stajano and Anderson, 1999; Soriente et al., 2007). In reality, it is impractical to mandate a single association model for all kinds of devices because different devices have different hardware capabilities. Also, different users and application contexts have different usability and security requirements. Because of this, forthcoming standards are adopting multiple association models. Although low-end devices like headsets and wireless access points may be limited to one association model, richer devices like mobile phones and personal computers will naturally support several. The security of individual association models has been studied widely. But new kinds of threats may emerge when several models are supported in personal devices and several standards, both new and old, are in use simultaneously.

This paper is an extended version of a paper presented in the ESAS 2007 workshop (Suomalainen et al., 2007). In this paper, we present and analyse various protocols for key establishment in personal networks and present a taxonomy for classifying them. We then make a comparative analysis of association models proposed in different standards from a practical point of view. The surveyed standards are Bluetooth Secure Simple Pairing (SSP) (Bluetooth SIG, 2007), Wi-Fi Protected Setup (Wi-Fi Alliance, 2007), Wireless USB Association Models (USB Implementers Forum, 2006), and HomePlugAV security modes (Newman et al., 2006, 2007). We show the similarities between the protocols in different standard specifications by relating them to our taxonomy. We point out other similarities as well: All of the them can address the problem of finding the right peer device usually by supporting some variation of the notion of *user-conditioning*: a device participates in the association only when it is in a special association mode; typically a device enters the association mode in response to an explicit user action, such as pressing a button. All of the surveyed standards are targeted for personal networks and support multiple association models.

The rest of this paper is organised as follows. In Section 2 we provide a systematic taxonomy of different protocols for key establishment and describe some basic protocols. In Section 3 we look at how different types of secure channels and physical interfaces can be used to implement the protocols discussed in Section 2. In Section 4 we explain how and which key establishment protocols and related association models are used in the surveyed standards. In Section 5 we evaluate and analyse the various association models described in these standards. Finally, in Section 6 we provide a summary and contemplate possible future developments in this area.

2 Key establishment protocols

2.1 Classification of key establishment methods

All of the association models we will survey in Section 4 are based on one or more protocols for human-mediated establishment of a shared key between two devices. The shared key is typically used to protect subsequent communication over the otherwise insecure communication channel and, possibly, in authentication for other access control decisions. We show that the same basic protocols are used in different standard specifications, even though the exact instantiations naturally differ.

The attacker model for key establishment is as follows. The two devices involved in key establishment are capable of communicating over an insecure communication channel. The devices themselves are assumed to be secure and trustworthy. The attacker has the standard Dolev-Yao capabilities (Dolev and Yao, 1983) over the insecure

channel: the attacker can insert, delete, modify or delay messages sent over the insecure channel. The security objective of the participating devices is to establish a common key shared only between the two devices, which they can use to protect subsequent communication between them. The goal of the attacker is to intervene in this process so that either it can read subsequent communication between the participating devices, or act as an active man-in-the-middle. In the latter case, the attacker can generate or modify messages and fool one or both of the devices into accepting these messages as originating from the peer device.

As a prelude to identifying and comparing these different instantiations, we present a systematic classification of human-mediated key establishment protocols that can be used in personal networks. Figure 1 provides an overview of this classification.

At a high level, key establishment may be a simple *key transport* or involve running a *key agreement* protocol. In the context of personal networks where the devices are likely to be in close proximity, an additional key establishment method is *key extraction* from the common shared environment.

Key transport. In key transport, one device chooses the key and transmits it directly to the second device using an OOB secure communication channel (**P1**). Typical OOB channels used for key transport include a direct USB cable connection or the use of removable memory, like flash drives. The security of key transport depends on the OOB channel being secret and unspoofable: a man-in-the-middle must not be able to modify the data transmitted OOB between the devices.

Key extraction. Devices in personal networks are in close proximity to one another and thus share a common ambient environment. This gives rise to an interesting possibility for key establishment: measurements of certain environmental parameters, such as the signal strengths of radio beacons in the vicinity (Varshavsky et al., 2007) or ambient noise, may be similar in devices that are close to each other but hard to predict from devices that are not in the same place at the same time. By measuring such parameters, and using them in a key agreement protocol, the devices may be able to *extract an authenticated shared secret* (**P12**).

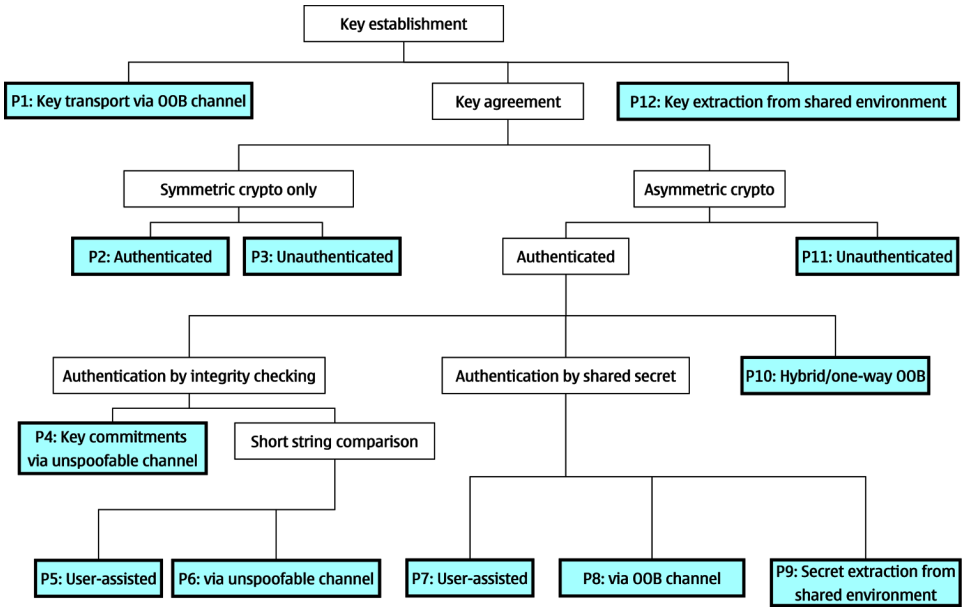
Key agreement. Key agreement protocols may be based purely on symmetric key cryptography, or may be based on asymmetric key cryptography as well. In the latter case, the typical protocol is the key exchange presented by Diffie and Hellman (1976).

Key agreement may be *unauthenticated* or *authenticated*. Unauthenticated symmetric key agreement (**P3**) is vulnerable even to passive eavesdroppers. Unauthenticated asymmetric key agreement (**P11**) is secure against passive eavesdroppers but is vulnerable to active man-in-the-middle.

2.2 Authentication methods

There are a number of ways to authenticate key agreement. Key agreement based on symmetric key cryptography is authenticated by using a sufficiently long *pre-shared secret* (**P2**). The security of such protocols depend on the length of the pre-shared secret. Authentication of asymmetric key agreement can be performed using some form of *integrity checking*, or by using a pre-shared secret or using a combination of these two. Authentication by integrity-checking can be done either by exchanging and comparing commitments to public keys, or by exchanging and comparing short integrity checksums. Now we take a closer look at the protocols involved in each case.

Figure 1 Classification of key establishment methods for personal networks (see online version for colours)



Authentication by exchanging key commitments. A simple folklore protocol to authenticate the public keys of two devices is to use an auxiliary channel to exchange commitments to the public keys (**P4**) (Balfanz et al., 2002). The auxiliary channel is unspoofable in that it is difficult for an attacker to insert, modify or delete messages in the channel without being detected. When the devices exchange public keys via the in-band channel, they can validate the authenticity of these keys by using the information exchanged via the auxiliary channel.

The security of the protocols depends on the auxiliary channel being unspoofable. There are two ways to realise such auxiliary channel. The first is to use a separate, OOB, physical channel which is resistant to spoofing. Several such OOB channels have been proposed in the literature including audio (Goodrich et al., 2006), visual (McCune et al., 2005; Saxena et al., 2006), infrared (Balfanz et al., 2002) and Near-Field Communication (NFC). Both devices involved in the association are assumed to support the same type of physical hardware interfaces. The second way is to use the *I-Codes* (Čagalj et al., 2006a) technique which uses the anti-blocking property inherent in some otherwise insecure in-band channels[5] to construct a logical auxiliary channel which is difficult to spoof.

The security also depends on the commitments of public keys being strong enough (e.g., a cryptographic hash function with at least 80 bits of output) to resist the attacker finding a second pre-image to the commitment.

Authentication by short integrity checksum. The idea of using short checksums to authenticate a key agreement was originally proposed in PGPfone by Zimmermann (1996). Subsequently several researchers have proposed variations and enhancements (Čagalj et al., 2006b; Laur et al., 2005; Pasini and Vaudenay, 2006; Vaudenay, 2005). In these protocols, each device computes a short checksum from the messages exchanged during the key agreement protocol. As we shall see in the example protocol below, the messages are structured such that if the two checksums are the same, the exchange is authenticated. This is sometimes referred to as "Short Authenticated String" (SAS) protocols. A basic three round mutual authentication protocol from (Laur et al., 2005) is depicted, in a simplified form, in Figure 2. Devices D_1 and D_2 first exchange their public keys PK_1 and PK_2. The protocol is used to mutually authenticate public keys. The notations are as follows: in practice, h is a cryptographic hash function like SHA-256; f is also a hash function, but with a short output mapped to a human-readable string of digits. The hat '$\hat{}$' symbol is used to denote the receiver's view of a value sent in protocol message over the insecure in-band channel.

The check in the last step can be done in many different ways. One way is to ask the user to do the comparison (**P5**): Each device 'shows' its own string to the user and ask whether it is the same as what the other device is showing. 'Showing' can use any applicable user interface: displaying the string on a screen, or having a voice synthesiser read out the characters in the string. If the checksum strings are identical, the user indicates this to both devices and both devices conclude that the authentication is successful. Otherwise, the user indicates a mismatch to both devices and both conclude that the authentication did not succeed. An alternative way is to do the check using an auxiliary unspoofable channel (**P6**). As before, the unspoofable channel can be a physical OOB channel, as presented by Saxena et al. (2006), Soriente et al. (2007), or an I-Codes channel by Čagalj et al. (2006a).

Figure 2 Authentication by short integrity checksum

1. D_1 generates a long random value R_1, computes commitment $h = h(R_1)$ and sends it to D_2

   $\mathsf{D}_1 \rightarrow \mathsf{D}_2$: h

2. D_2 generates a long random value R_2 and sends it to D_1

   $\mathsf{D}_1 \leftarrow \mathsf{D}_2$: R_2

3. D_1 sends R_1 to D_2

   $\mathsf{D}_1 \rightarrow \mathsf{D}_2$: R_1

4. D_2 checks if $\hat{h} \stackrel{?}{=} h(\hat{R}_1)$. If equality holds, D_2 computes $V_2 = f(P\hat{K}_1, PK_2, \hat{R}_1, R_2)$, otherwise it aborts.

   D_1 computes $V_1 = f(PK_1, P\hat{K}_2, R_1, \hat{R}_2)$.

5. Both devices check if V_1 equals V_2.

To break this protocol, a man-in-the-middle has to choose random numbers R'_1, R'_2 and public keys PK'_1, PK'_2 so that $f(PK'_1, PK_2, R'_1, R_2)$ equals $f(PK_1, PK'_2, R_1, R'_2)$. The security of the protocol depends on the quality of the functions h and g. If h is collision-resistant, the attacker has to choose R'_1 without knowing anything about R_2. If h is one-way, attacker has to choose R'_2 without knowing about R_1. If the output of f is a uniformly distributed ℓ-bit value, then the chance of a man-in-the-middle succeeding is $2^{-\ell}$ because the attacker cannot influence the outcome of g. This success probability does not depend on any additional assumptions about the computational capabilities of the attacker beyond that he cannot break h in real time. The formal proofs were presented by Laur and Nyberg (2006).

Authentication by (short) shared secret. Key exchange can also be authenticated using a short pre-shared secret passkey. A number of different methods have been proposed for password-authenticated key exchange since the idea was introduced by Bellovin and Merritt (1992). In Figure 3 we describe a variant of the MANA III protocol by Gehrmann et al. (2004) originally described by Larsson (2001). It uses a one-time passkey P to authenticate PK_1 and PK_2. P is split into k pieces, labelled $P_1 \ldots P_k$. The steps in the protocol are repeated k times. The figure shows the exchanges in the ith round.

In each round, each party demonstrates its knowledge of P_i. A man-in-the-middle can easily learn P_1 by sending garbage in message 2, and figuring out P_1 by

exhaustive search once D_1 reveals R_1 in message 3. However, without knowing $P_i, i = 2 \ldots k$, the attacker cannot successfully complete the protocol run (recall that P is a *one-time* passkey). With ℓ-bit passkey and k rounds the probability for a successful man-in-the-middle attack is $2^{-(\ell - \frac{\ell}{k})}$. As in the case of short authentication string, the man-in-the-middle success probabilities do not depend on additional assumptions about the attacker's computational capabilities.

Figure 3 Round i of authentication by (short) shared secret

1. D_1 generates a long random value R_{i1}, computes commitment $h_{i1} = h(1, PK_1, P\hat{K}_2, P_i, R_{i1})$ and sends it to D_2

   $\mathsf{D}_1 \rightarrow \mathsf{D}_2$: h_{i1}

2. D_2 generates a long random value R_2, computes commitment $h_{i2} = h(2, PK_2, P\hat{K}_1, P_i, R_{i2})$ and sends it to D_1

   $\mathsf{D}_1 \leftarrow \mathsf{D}_2$: h_{i2}

3. D_1 responds by sending R_{i1} to D_2

   $\mathsf{D}_1 \rightarrow \mathsf{D}_2$: R_{i1}

   D_2 checks if $\hat{h}_{i1} \stackrel{?}{=} h(1, P\hat{K}_1, PK_2, P_i, \hat{R}_{i1})$ and aborts if it does not hold.

4. D_2 responds by sending R_{i2} to D_1

   $\mathsf{D}_1 \leftarrow \mathsf{D}_2$: R_{i2}

   D_1 checks if $\hat{h}_{i2} \stackrel{?}{=} h(2, PK_1, P\hat{K}_2, P_i, \hat{R}_{i2})$ and aborts if it does not hold.

There are three different ways for arranging for both devices to know the same P. One way is to have the user as the intermediary (**P7**): one device may show a value for P which the user is asked to enter into the second device, or the user may choose P and enter it into both devices. Alternatively, P may be transported from one device to another using a OOB channel providing communication secrecy (**P8**). A third possibility is to extract P from the shared environment (**P9**) (Varshavsky et al., 2007). In the latter two methods, there is no need for a human to transfer P between the devices. Consequently P can be longer, thus making probability for a successful attack smaller. Note that P is still used only to authenticate the key agreement, rather than as the long term secret.

Hybrid authentication. Hybrid authentication protocols are used to achieve mutual authentication when only a one-way out-band-channel is available (**P10**). The one-way channel is used to transmit the shared secret value and a hash of the public key from the first device to the second. The second device authenticates the first based on the public key hash. The first device authenticates the second based on its knowledge of the shared secret. A basic protocol is depicted in Figure 4. The function $c(M, K)$ is a Message Authentication Code (MAC) on message M using a key K.

Figure 4 Hybrid authentication protocol

1. D_1 picks two long random values R_1 and K, computes commitment h to public key PK_1 as $h = h(PK_1, R_1)$ and sends h and K using OOB channel

   $\mathsf{D}_1 \Rightarrow \mathsf{D}_2$: S, C_1 (sent via the OOB channel)

2. D_1 sends its public key and random value using in-band channel.

   $\mathsf{D}_1 \rightarrow \mathsf{D}_2$: PK_1, R_1

3. D_2 checks if $h \stackrel{?}{=} h(P\hat{K}_1, \hat{R}_1)$ and aborts if it does not hold. Otherwise, D_2 picks its own long random value R_2, computes $C = c(P\hat{K}_1 | PK_2 | \hat{R}_1 | R_2, K)$ and sends the result to D_1 with its own public key and random value.

   $\mathsf{D}_1 \leftarrow \mathsf{D}_2$: PK_2, R_2, C

4. D_1 checks if $\hat{C} \stackrel{?}{=} c(PK_1 | P\hat{K}_2 | R_1 | \hat{R}_2, K)$ and aborts if it does not hold.

The security of the protocol depends on the OOB communication being both secret and unspoofable, as well as on strength of the hash function h and the message authentication code function c.

3 Secure channels and physical interfaces

In this section, we survey various types of secure communication channels and physical interfaces and how they can be used for key establishment in the various methods we looked at in Section 2.

OOB channels are communication channels distinct from the insecure channel over which the devices normally communicate. Using OOB channels to aid in association and key establishment can greatly improve usability by minimising user actions. Therefore, from very early on (Stajano and Anderson, 1999) researchers have looked for ways of using OOB channels in key establishment.

Various types of OOB channels have been considered in the literature including physical contact (Stajano and Anderson, 1999), infrared (Balfanz et al., 2002), audio channels (Soriente et al., 2007), visual channels (McCune et al., 2005; Saxena et al., 2006), very short-range wireless communication channels like NFC.[6] Different types of channels have different characteristics which affect their applicability to the different methods we saw in Section 2. The characteristics that are relevant for key agreement are the following:

- *Channel security.* All useful types OOB channels are assumed to provide *integrity*: an attacker is assumed incapable of modifying, inserting or deleting messages sent via the channel. Some types are assumed to provide *secrecy* as well: an attacker is assumed incapable of reading the information sent via the channel. Usually physical connections and

NFC channels are assumed to provide secrecy; however the validity of these assumptions have been questioned (Heydt-Benjamin et al., 2007).

- *Directionality.* Depending on the hardware available on the devices, the OOB channel may be unidirectional or bidirectional.
- *Bandwidth.* Bandwidth of a channel is the rate at which it can transfer data. The bandwidth of an OOB channel is relevant in key establishment because it influences the time it takes to complete the association process.

Table 1 lists the protocols from Section 2 that can be implemented using OOB channels. Footnotes in the table list papers which describe how different types of OOB channels are used with that protocol. The table gives also characteristics that these protocols require from OOB channels.

Although the promise of better usability is the motivation for using OOB channels in key establishment, the downside is the need to have the necessary hardware interfaces on both devices. There is no universal OOB channel guaranteed to be available on all devices. The vast majority of personal devices are low-cost commodity devices. Therefore adding a new hardware interface simply for the purpose of easing the association process is usually not an economically viable option. Researchers have therefore investigated ways to establish associations while maximising security, usability and cost. One approach is to design the association procedures taking the resource asymmetry between the devices involved in the association. Typically one device, like a laptop or phone, has greater capabilities, while the other, like an access point or headset, is extremely resource constrained and cost-sensitive. Saxena et al. (2006) describe setting up a security association using a visual channel: one device is assumed to have a video camera while the other device needs to have only a single light source (such as a light-emitting diode) and mechanisms for user confirmation (like buttons for indicating yes and no).

Characteristics of in-band communication channels have been utilised by some key establishment protocols to strengthen security level. These schemes are based on the fact that signal quality is different in different locations. For instance, Newman et al. (2006) observed that signals on power-line channel must be adapted for each receiver and because of that eavesdropper cannot receive good enough signal. Further, they argue that active online attacks can be easily detected in a narrowband power-line channel. Azimi-Sadjadi et al. (2007) proposed generation of shared keys from signal envelopes in wireless networks.

4 Association models in standards

In this section, we survey the association models proposed in four emerging standards for personal networks. We then compare them by referring to the classification presented in Section 2.

4.1 Bluetooth Secure Simple Pairing

Bluetooth SSP from Bluetooth SIG (2007) is intended to provide better usability and security than the original Bluetooth pairing mechanism, and is expected to replace it. Simple pairing consists of three phases. In the first phase, the devices find each other and exchange information about their user input/output capabilities and their elliptic curve Diffie-Hellman public keys for the FIPS P-192 curve (National Institute of Standards and Technology, 2000). In the second phase, the public keys are authenticated and the Diffie-Hellman key is calculated. The exact authentication protocol, and hence the association model, is determined based on the device user-I/O capabilities. In the third phase, the agreed key is confirmed (in one association model, the authentication spans both the second and third phase).

SSP supports four different association models: Numeric Comparison, Passkey entry, 'Just Works' and OOB models. Now we will examine each of these models and the protocols they use for authentication in phase 2.

- *Numeric comparison model* is where the user manually compares and confirms whether the short integrity checksum displayed by both devices are identical (Figure 1: **P5**). The compared checksum is 6 digits long. The phase 2 protocol is an instantiation of the protocol in Figure 2.
- *Passkey entry model* is targeted primarily for the case where only one device has a display but the

Table 1 Applicability of Out-Of-Band channels

Method	*Integrity*	*Secrecy*	*Directionality*	*Data size*
P1: Key transport[1]		√	1-way	128–256 bits
P4: Exchange of key commitments[2]	√		2-way	128–256 bits
P6: Short string comparison[3]	√		1-way[4]	12–20 bits
P8: Transfer of (short) secret		√	1-way	12–20 bits
P10: Transfer of key commitment and secret	√	√	1-way	256–512 bits

[1]Stajano and Anderson (1999).
[2]Balfanz et al. (2002), McCune et al. (2005) and Soriente et al. (2007).
[3]Saxena et al. (2006).
[4]For mutual authentication, the method relies on the user as the return channel.

other device has a keypad. The first device displays the 6-digit secret passkey, and the user is required to type it into the second device. The passkey is used to authenticate the Diffie-Hellman key agreement (Figure 1: **P7**). The protocol is based on user-assisted authentication by shared secret in Figure 3 with 20 rounds ($k = 20$). Devices prove knowledge of one bit of the passkey in each round.

- '*Just works*' *model* is targeted for cases where at least one of the devices has neither a display nor a keypad. Therefore, unauthenticated Diffie-Hellman key agreement is used (Figure 1: **P11**) to protect against passive eavesdroppers but not against man-in-the middle attacks.
- *Out-Of-Band model* is intended to be used with different OOB channels, in particular with Near Field Communication technology. Device D_A uses the OOB channel to send a 128-bit secret r_a and a commitment C_a to its public key PK_a. Similarly, D_B uses the OOB channel to send r_b and C_b. If OOB communication is bidirectional, mutual authentication is achieved by each party verifying that the peer's public key matches the commitment received via the OOB channel. (Figure 1: **P4**).

  If the OOB channel is only one way, the party receiving the OOB message can authenticate the public key of its peer. However, the party sending the OOB message must wait until the third, key confirmation, phase of SSP which we now describe.

In phase 3, the same key confirmation protocol is executed in all association models to confirm successful key exchange by exchanging message authentication codes using the newly computed Diffie-Hellman key. Each device includes the random value r received from the peer in the calculation of its MAC. In the one-way OOB case, the MAC serves as a proof-of-knowledge of the shared secret r received OOB. This is the hybrid authentication protocol **P10** (Figure 4).

Peer discovery. In original Bluetooth pairing, peer discovery is left to the user: the user initiates pairing from one device which constructs a list of all other Bluetooth devices in the neighbourhood that are publicly discoverable and asks the user to choose the right one to pair with. In the OOB association model, device addresses are sent via the OOB channel. This makes it possible to uniquely identify the peer to pair with, without requiring user selection. SSP does not contain any new mechanisms to make peer discovery easier in the other association models. Individual implementations could use existing Bluetooth modes, like the "limited discoverable mode" and 'pairable mode' to support user-conditioning on the peer device. However, since such user-conditioning is not mandated by the specification, it is quite possible that the implementations of SSP may still need to resort to asking the user to choose the right peer device from a list.

Model selection. The association model to be used is uniquely selected during the initialisation of the session. If the association process is initiated by OOB interaction, and security-information is sent through the OOB channel, then the OOB model is chosen automatically. Otherwise, in phase 1, the devices exchange their input-output capabilities. The SSP specification describes how these capabilities should be used to select the association model.

4.2 Wi-Fi Protected Setup

Wi-Fi Protected Setup (WPS) is Wi-Fi Alliance's specification for secure association of wireless LAN devices. Microsoft's Windows Connect Now (WCN) includes a subset of association models described in WPS. The deployment of WPS has already started: According to Wi-Fi Alliance (2007), there are currently almost 200 products which are certified for WPS. The products range from WLAN access points to USB WLAN adapters. These products are provided by multiple different manufacturers.

The objective of WPS is to mutually authenticate the enrolling device with the Wi-Fi network and to deliver network access keys to the enrolling device. This is done by having the enrolling device interact with a device known as the 'registrar', responsible for controlling the Wi-Fi network. The registrar may be, but does not have to be, located in the Wi-Fi access point itself. WPS supports three configuration methods: In-band, OOB, and push-button configurations.

- *In-band configuration* enables associations based on a shared secret passkey (Figure 1: **P7**). The user is required to enter a passkey of enrollee to the registrar. This passkey may be temporary (and displayed by the enrollee) or static (and printed on a label). 8-digit passkeys are recommended but 4-digit passkeys are allowed. The passkey is used to authenticate the Diffie-Hellman key agreement between the enrollee and the registrar. The protocol used is a variation of the modified MANA III protocol in Figure 3 with two rounds ($k = 2$).

  As in MANA III (Figure 3), once a passkey is used in a protocol run, an attacker can recover the passkey by dictionary attack (although in this instantiation, the attacker needs to be active since the computation of the used commitments includes a key derived from the Diffie-Hellman key).
- *Out-Of-Band configuration* is intended to be used with channels like USB-flash drives, NFC-tokens or two-way NFC interfaces. There are three different scenarios:
  - Exchange of public key commitments (Figure 1: **P4**), typically intended for two-way NFC interfaces, where the entire Diffie-Hellman exchange and the delivery of access keys takes place over the OOB channel.
  - Unencrypted key transfer (Figure 1: **P1**). An access key is transmitted from a registrar to

enrollees in unencrypted form, either using USB-flash drives or NFC-tokens.

- Encrypted key transfer. This is similar to the previous case, except that the key is encrypted using a key derived from the (unauthenticated) Diffie-Hellman key agreed in-band. From a security perspective, this is essentially OOB key transfer (Figure 1: **P1**).

- *Push button configuration* is an optional method that provides an unauthenticated key exchange (Figure 1: **P11**). The user initiates the Push button configuration by conditioning the enrollee (e.g., by pushing a button), and then, within 120 s the user has to condition the registrar as well. The enrollee will start sending out probe requests to all visible access points inquiring if they are enabled for push button configuration. Access points are supposed to respond affirmatively only when their registrar has been conditioned by the user for this configuration. If a device or registrar sees multiple peers ready to start push button method, it is required to abort the process and inform the user.

Peer discovery. Enrollees start association in response to explicit user conditioning. They scan the neighbourhood for available access points and send Probe Request messages. The Probe Response message has a 'SelectedRegistrar' flag to indicate if the user has recently conditioned a registrar of that access point to accept registrations. This is mandatory for push button configuration but is optional for other models. Thus it is possible that user may have to be asked to select the correct Wi-Fi network from a list of available networks.

Model selection. The model is explicitly negotiated at the beginning.

4.3 Wireless USB association models

Wireless USB (WUSB) is a short-range wireless communication technology for high speed data transmission. WUSB Association Models Supplement 1.0 specification from USB Implementers Forum (2006) supports two association models for creating trust relationships between WUSB hosts and devices:

- *Cable model* uses OOB key transfer (Figure 1: **P1**) and utilises wired USB connection to associate devices. Connecting two WUSB devices together is considered as an implicit decision and, hence, the standard does not require users to perform additional actions like accept user prompts.

- *Numeric model* relies on the users to authenticate the Diffie-Hellman key agreement by comparing short integrity checksum values (Figure 1: **P5**). The protocol is an instantiation of the protocol in Figure 2. First D_A and D_B negotiate the length of the checksum to be used. The specification requires that WUSB hosts must support 4-digit checksums whereas WUSB devices must support either 2 or 4-digit checksums.

These two association models were selected to handle all possible usage cases. The basic assumption is that most of the WUSB devices are equipped with a USB cable thus being able to use the cable model. Numeric model was chosen to handle situations where cable model could not be used. WUSB hosts need to implement both association models, whereas in devices only one may be implemented. This way it is ensured that devices can always be associated.

A passkey model similar to Bluetooth SSP was considered but was not chosen because of users' preference for comparing digits instead of typing them. According to USB Implementers Forum (2007) usage of NFC for association is being actively investigated, and may be included as an association model in later WUSB specifications.

Peer discovery. The association is initialised by implicit or explicit user conditioning. Attaching a USB-cable is interpreted as an implicit conditioning. The user pressing a button is an example of explicit user conditioning. In the numeric model the user sets a USB device to search for hosts and a USB host to accept connections. The host advertise its willingness to accept a new association in the control messages it transmits on the WUSB control channel. In case multiple devices are simultaneously advertising their accepting states, the searching device either selects a host randomly or ends the association procedure in a failure. In future revisions of Wireless USB, some preassociation information about hosts and devices may be included. This would allow the searching device to display a list of user friendly host names accepting connection. The user could then select the desired one from the list.

Model selection. The choice of the association model is based on the type of user conditioning done. In case a cable is plugged, the devices exchange information on whether they support cable association. If so, they use cable model. If conditioning is explicit, they use numeric model.

4.4 HomePlugAV protection modes

HomePlugAV is a power-line communication standard for broadband data transmission inside home and building networks. In addition to protecting deliberate attacks, association mechanisms are used to create logically separate subnetworks by distributing an 128-bit AES Network Encryption Key (NEK) for devices in each subnetwork. As with WPS, each HomePlugAV network has a controller device. HomePlugAV supports the following association models (Newman et al., 2006):

- *Secure mode* allows new devices to have a secret passkey, of at least 12 alphanumeric characters long, typically printed on a label. The user is required to

type in this passkey to the controller device. The controller device uses it to construct an encryption of NMK and send it to the new device. The keys for devices joining in secure mode is different from the keys for devices joining in simple connect mode. This is an example of authenticated symmetric crypto key agreement (Figure 1: **P2**).

- *Optional modes* enable use of alternative models for distributing NMKs or NEKs between devices. These include 'manufacturer keying' where a group of devices have a factory installed shared secret, and external keying, where trust is bootstrapped from other methods.
- *Simple connect mode* uses symmetric crypto based key agreement to agree on a shared key. This Network Membership Key (NMK), is used to transport NEK to the new device. The key agreement process is as follows. To admit a new device, the user is required to first condition the controller device, and then condition the new device, e.g., by turning on its power. The devices find each other and exchange nonces. A Temporary Encryption Key (TEK) is formed by hashing the two nonces together. The controller encrypts the NMK using the TEK and sends it to the new device. The model is an unauthenticated (Figure 1: **P3**) as any cryptographic authentication mechanisms are not used. However, some level of authentication has been achieved with communication engineering as described below.

Man-in-the-middle attacks can be prevented in simple connect mode by utilising characteristics of powerline medium. Before two nodes can communicate, they must negotiate tone maps, which enable devices to compensate disturbances caused by powerline channel. This negotiation is done in a reliable, narrow-band broadcast channel. Thus a man-in-the-middle trying to negotiate tone maps with the legitimate endpoints can be detected.

Passive eavesdropping in the broadband point-to-point channel is difficult since an attacker, even with the knowledge of the tone maps used between the legitimate endpoints, will not be able to extract the signal from the channel because the signal-to-noise ratio will be too poor at different locations, particularly, when the attacker is outside a building and the legitimate end points are inside. Also, licensees of HomePlugAV technology do not provide devices that can extract signal without negotiating tone maps. Hence, attackers must be able to build expensive devices for eavesdropping.

Peer discovery. In simple connect mode the peer discovery is performed by the user conditioning the devices into a suitable modes, and the new device scanning the network to find a controller that is willing to accept new devices.

Model selection. The model is selected by user conditioning. There is no automatic negotiation.

5 Evaluation and analysis

In this section, we analyse the association models described in Section 4 from different perspectives and point out some problematic areas.

5.1 Comparison of security levels

First we summarise and compare the security levels provided by the different association models discussed in Section 4. A comparative summary of models' security characteristics is presented in Table 2.

5.1.1 Offline attacks

The OOB association models rely on the secrecy of OOB communication to protect against passive attacks against key agreement. The in-band and hybrid models in all of the standards except HomePlugAV use Diffie-Hellman key agreement to protect against passive attacks. The level of protection depends on the strength of the algorithms and the length of the keys used. In the 'Work' subcolumn under the 'Offline Attacks' column of Table 2, we use some recent sources (Kivinen and Kojo, 2003) and (Barker et al., 2006) to estimate the amount of work an attacker has to do in order to be successful. The figures correspond to approximate lower bounds, and should be treated as rough ballbark estimates only. Offline attack protection in HomePlugAV relies on the characteristics of the power-line communications: the signal-to-noise ratio is assumed it to make it difficult for an attacker to eavesdrop. The HomePlugAV secure mode uses symmetric key encryption as protection.

5.1.2 Online active attacks

Mounting an online active attack as a man-in-the-middle against key agreement is significantly more difficult than passive eavesdropping. Several of the models ('Just Works', 'Push Button', and 'Simple Connect') trade off protection against man-in-the-middle attacks, in return for increased ease-of-use.

Other in-band association models rely on authentication as the means to protect against online active attacks. The probability of success for an online active attack depends on the length of the key as well as the protocol. The Bluetooth SSP numeric comparison model uses 6-digit checksums leading to a success probability of $\frac{1}{1000000}$. The WUSB numeric model allows a success probability of $\frac{1}{100}$ when two digit checksum is used, and $\frac{1}{10000}$ when four digit checksum is used. These probabilities do not rely on any assumptions about the computational capabilities of the man-in-the-middle.

Association models based on numeric comparison use cryptographic hash functions as the commitment function. In principle, a man-in-the-middle who can break the hiding property of the hash commitment function *during* the key agreement process can also succeed by figuring out the nonce used in the commitment. We show this

Table 2 Comparison of security characteristics of association models

	Offline attacks		*Online active attacks*		
Association model	*Protection*	*Work*[1]	*Protection*	*Success probability*	*Work*[2]
Bluetooth Secure Simple Pairing					
Numeric comparison	DH	2^{80}	6 digit checksum	2^{-20}	2^{148}
Just Works	DH	2^{80}	–	1	0
Passkey entry	DH	2^{80}	6 digit passkey	2^{-19}	2^{147}
OOB	DH	2^{80}	OOB security	–	2^{128}
Wi-Fi Protected Setup					
In-band	DH	2^{90}	8 digit passkey	$2^{-13.2}$	$2^{141.2}$
In-band + OOB[3]	DH	2^{90}	OOB security	2^{-128}	2^{196}
OOB	OOB	2^{90}	OOB security	–	–
Push Button	DH	2^{90}	–	1	0
WUSB Association Models					
Numeric model	DH	2^{128}	2/4 digit checksum	$2^{-6.6}$ or $2^{-13.2}$	$2^{262.6}$ or $2^{269.2}$
Cable model	OOB	2^{128}	OOB	–	–
HomePlugAV Protection Modes					
Simple Connect	SNR	High	traffic monitoring	Low	High
Secure mode	AES	2^{72}	passkey	2^{-72}	2^{72}

[1]Rough work effort estimates based on (Barker et al., 2006, Table 2) and (Kivinen and Kojo, 2003, Section 8).
[2]Work effort to break commitments exchanged, with probability 1.
[3]OOB passkey + checksum.

in Table 2, in the 'Work' subcolumn under the "Online Active Attacks" column by indicating the amount of *online* work the attacker has to perform in order to succeed with probability 1. In this case, assuming that the hash function is strong, and requires exhaustive search to find the correct pre-image, the work factor depends on the size of the nonce and the size of the checksum. Bluetooth SSP uses 128-bit nonces and 20-bit checksum; therefore we use the figure 2^{148}. WUSB numeric model uses the Diffie-Hellman public value as the hidden nonce, which is based on a 256-bit long private value. It uses 2- or 4-digit checksums. Hence, we use a work factor figure of $2^{262.6}$ or $2^{269.2}$. These figures correspond to the amount of online work required for the attacker to succeed with probability 1.

Association models based on passkeys also use cryptographic hash functions as the commitment function. An attacker who can break the hiding property of the hash function can figure out the nonce and the passkey component used in a given round. The work factor depends on the size of the nonce plus the size of the passkey component. For Bluetooth SSP the work factor is 2^{147} (128-bit nonce and 19-bit passkey component), whereas for WPS in-band model the work factor is $2^{141.2}$ (128-bit nonce and 4-digit passkey component). Alternatively, an attacker who can break the binding property of the hash function can send a randomly chosen value as h_{i2} in Step 2 of the protocol (Figure 3), learn the passkey after receiving message 3 and then calculate a suitable R_{i2} that matches the alleged commitment sent earlier in Step 2. The work factor depends on the size of the commitment. Bluetooth SSP uses 128-bit commitments, leading to a work factor of 2^{128}. WPS uses 256-bit commitments, but the size of the random input is only 128-bit. Thus, although 2^{128} amount of work is sufficient to break the binding property, the attacker cannot always succeed, since he may have used a value in Step 2 for which there is no 128-bit pre-image. Therefore, we stick with the $2^{141.2}$ work factor discussed above.

Recall from Section 2 that with n bit passkeys and k rounds the success probability for an online active attack against the passkey protocols is $2^{-(n-\frac{n}{k})}$. Bluetooth SSP passkey entry model uses 6-digit ($n \approx 20$) one-time passwords in $k = 20$ rounds. This leads to approximately $\frac{1}{1000000}$ success probability. WPS network uses essentially the same protocol, but in two rounds only. This leads to success probabilities of $\frac{1}{100}$ when 4-digit passkeys are used, and $\frac{1}{10000}$ when 8-digit passkeys are used. In both cases, the passkey must be single-use. If the passkey is re-used, the success probability of man-in-the-middle rises dramatically, reaching 1 after the kth re-use, where k is the number of rounds in the original protocol. In other words, if the same fixed passkey in WPS network model is re-used even *once*, the man-in-the-middle can succeed in the next attempt with certainty. As before, we can estimate the online work effort the attacker has to do to break the hash commitments. HomePlugAV secure mode uses a 12 character passkey which is used to generate a key for AES encryption, leading to a probability of 2^{-72} and the amount of online work effort is 2^{72}.

The reader may notice that resistance against breaking the hash commitment appears to be over-engineered. To see this in context, assume a hash commitment function with hash values of length a. Then it takes about 2^a online work to do pre-image search and break the hiding and binding properties of the hash function with probability close to one. Let t be the upper bound for the amount of online work a real world adversary is capable of, where $t < 2^a$. Then the probability that the adversary succeeds in pre-image search is about $t/2^a$. When this hash

commitment function is used in a passkey authentication scheme with a success probability of $2^{-\ell}$, the parameters are in balance (i.e., not over-engineered) when $t/2^a = 2^{-\ell}$. For example, in the case of Bluetooth SSP, if $a = 128$ and $\ell = 20$, then the choice of parameters is balanced if $t = 2^{108}$.

As pointed out by Kuo et al. (2007), we can infer a similar upper bound for amount of offline work implied by the choice of parameters for offline protection. For example, assuming that the Diffie-Hellman key agreement used in Bluetooth SSP requires 2^{80} amount of offline work to break (Table 2), and that the design balances the protection against offline attacks with that of online guessing, the implied upper bound for offline work that the attacker is capable of is given by $t' = 2^{80}/2^{20}$. This figure is based on the assumption that the particular offline attack technique used by the attacker allows the work done by him to be cumulative: that is, partial work done by the attacker reduces the space from which he has to guess.

The hybrid models using a one-directional OOB channel, the random secret transferred using the OOB channel is 128 bits long leading to a computational security of 2^{-128}.

Attack probability against HomePlugAV simple connect mode is assumed to be small as attackers can be detected by monitoring communication on narrowband channel (Newman et al., 2006).

Wi-Fi and Bluetooth have legacy association models. If a device supports both the improved and the legacy association models, it is vulnerable to a bidding down attack, which is difficult to detect without relying on the user.

5.1.3 *Associations with wrong peers*

Unauthenticated association models face the risk of a device being associated with a wrong peer. For instance, in WPS push button model, the user may condition first the enrollee to search for registrars before conditioning the registrar. If the attacker sets a bogus registrar to accept connections before the users does it with the legitimate registrar, the enrollee associates with the attacker's registrar. Only in the case when both registrars, the bogus and the legitimate one, are simultaneously accepting connections, is the procedure aborted.

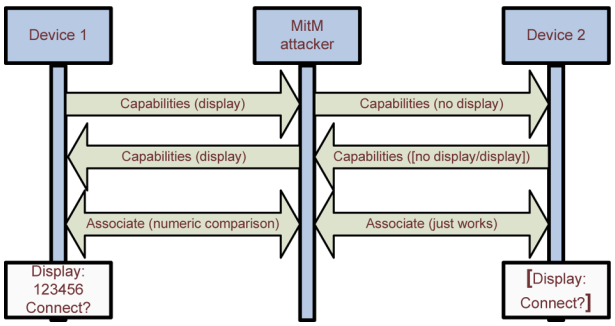
In HomePlugAV Simple Connect mode, the user sets the control device to accept connections before starting the joining device up. This could be used to reduce the probability for an attacker to successfully masquerading as a bogus control device because since, if the new device sees multiple control points, it can abort association. However, the mode is potentially vulnerable for fatal errors where the user is slow to switch power to the new device. In this case an attacker may connect to user's control point and get the network encryption key. The more longer walking distance there is between power-line devices, the more likely this attack is to succeed.

5.2 *Further challenges in implementing multiple association models*

Above, we saw how naive implementations of user interaction could increase the likelihood of fatal errors. In this section, we look at further similar challenges in implementation arising out of the fact that the standards invariably support multiple association models simultaneously.

Consider specifications that support an unauthenticated association model as well as user-assisted comparison of integrity checksums. An example is a Bluetooth device that supports the numeric association model and the 'Just Works' model. Figure 5 illustrates a man-in-the-middle attacker who can intercept messages exchanged during an association. The first associated device has a display and the second may or may not have a display. The attacker changes device capability information so that the first device will be using the numeric comparison model and that the second device will be using 'Just Works' model. This leads to a situation where the first device shows a 6-digit checksum and the second device, using 'Just Works' model, does not display a checksum, even if it would have a display. The user may have been educated to detect a mismatch in checksums. But now, when only one device displays a checksum, the user is likely to be confused and may just go ahead and accept the association.

Figure 5 Man-in-the-middle between different association models (see online version for colours)



To get an idea about whether such user confusion is likely, Valkonen et al. (2007) included the situation depicted in Figure 5 as a test scenario in one round of an on-going series of usability testing. Out of 40 test users, 6 accepted the pairing on both devices, 11 noticed the problem and rejected the pairing on both devices, and the rest rejected pairing on Device 1 but accepted it on Device 2.

This attack has two implications. Firstly, when the second device has a display, it is a bidding down attack against this device. The second device will know that the association is unauthenticated. However, the user may still allow the association to happen. Secondly, it is a bidding up attack against the first device since it believes that the association is made using a secure protocol resistant to man-in-the-middle attacks. Consequently, the first

device may choose to trust this security association more than it would trust a 'Just Works' security association. For instance, it may have a policy rule, which allows more trustworthy devices to initiate connections without user confirmation.

A scenario related to the attack on Figure 5 arises with devices that are willing to participate in setting up a security association without immediate user conditioning. Public printers and access points are examples of devices that may be permanently conditioned for association. Suppose a user starts associating Device 1 with Device 2 using an association model that does not require any user dialog (e.g., WUSB cable model, or HomePlugAV Simple Connect mode) and that Device 2 is permanently conditioned to accept incoming association requests. If an attacker now initiates association with Device 2, say using Bluetooth SSP numeric comparison, a user dialog will pop up on Device 2. Since the user is in the middle of associating Device 1 and Device 2, he might answer the dialog thinking that it is a query about Device 1. Depending on the nature of the dialog, the attacker may end up gaining unintended privileges on Device 2.

Strengthening devices. Now we discuss some implementation guidelines that can help address the kind of attacks identified above. When a security association is stored persistently, information about its level of security should be stored as well. HomePlugAV already does this indirectly by using different keys with different association models. Furthermore, this security-level information should be used in deciding the level of trust granted to the peer device. For instance, devices associated using Bluetooth SSP 'Just Works' or HomePlugAV Simple Connect models should not be allowed to install or configure software, at least, without explicit authorisation from the user. This precaution would help to prevent bidding down attacks. The man-in-the-middle attack between numeric comparison and unauthenticated protocols (Figure 5) could be addressed with two alternative strategies:

- Bidding down the second device from using numeric comparison to the 'Just Works' model could be addressed by requiring that devices believing to be in 'Just Works' association would anyway show the checksum if they are able to do so. However, this solution does not prevent the bidding up attack against the first device.
- Bidding down and bidding up attacks can both be countered by querying the user appropriately to confirm the I/O capabilities of the peer device. For instance, if the capability negotiation messages indicate that the peer device has no display, a device could ask the user if the peer device does indeed have a display. If the user gives answers affirmatively, it is an indication of a man-in-the-middle. However, such an additional dialogue is likely to impair usability.

6 Conclusions

The problem of designing ways to set up security associations in personal networks is a challenging one because it calls for balancing usability, security and cost. A number of innovative solutions have been proposed in recent research literature. Some of these have been incorporated into new standards for associating devices in personal networks. The objective of the new standards is to make the association process more user-friendly while improving the security at the same time without incurring significant cost penalties.

We surveyed various protocols in the research literature and association models used in different standards specifications. We presented a systematic classification of protocols for human-mediated establishment of session keys and provided formal analyses of some of them. We showed how the different protocols in standard specifications are related by using our classification.

The flexibility of the new proposals also introduce potential for some new attacks. We described some such threats. Careful design of user dialogs may reduce the likelihood of these attacks. However, how exactly to design the user dialogs to preserve security without harming usability remains an open issue.

Devices implementing the new standards are beginning to be deployed. All of them provide better security than the old procedures they replace. However, how well they are accepted by users remains to be seen. Unauthenticated key agreement (as in the 'Just Works' model of Bluetooth SSP and the 'Pushbutton' model of WPS) incur virtually no additional cost and optimal in usability. Therefore it may turn out to be more preferred and more widely deployed than authenticated key agreement. However, unauthenticated key agreement will not be sufficient for certain scenarios. One example is associating input devices (like keyboards and mice) with a computing device – a malicious input device can cause significant damage to the computing device. Another example is associating personal medical devices, or other similar contexts that may be subject to privacy regulation. Thus, the need for extremely inexpensive (and yet secure and usable) solutions for this problem remains. In-band integrity channels (Čagalj et al., 2006a) and extracting secrets from the shared environments using existing sensors (Varshavsky et al., 2007) seem to be promising avenues to conduct further research.

Acknowledgements

We thank Kaisa Nyberg for valuable input, which has improved the paper significantly.

References

Azimi-Sadjadi, B., Kiayias, A., Mercado, A. and Yener, B. (2007) 'Robust key generation from signal envelopes in wireless networks', *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ACM, New York, NY, USA, pp.401–410, http://portal.acm.org/citation.cfm?id=1315295.

Balfanz, D., Smetters, D., Stewart, P. and Wong, H.C. (2002) 'Talking to strangers: authentication in ad-hoc wireless networks', *Proceedings of the Network and Distributed System Security Symposium*, http://www2.parc.com/csl/members/balfanz/publications/loclim.pdf

Barker, E., Barker, W., Burr, W., Polk, W. and Smid, M. (2006) *Recommendation for Key Management – Part 1: General (Revised)*, http://csrc.nist.gov/CryptoToolkit/kms/SP800-57Part1_6-30-06.pdf

Bellovin, S.M. and Merritt, M. (1992) 'Encrypted key exchange: password-based protocols secure against dictionary attacks', *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, pp.72–84, ieeexplore.ieee.org/iel2/412/5566/00213269.pdf.

Bluetooth SIG (2007) *Bluetooth 2.1 Specifications. Bluetooth Special Interest Group*, http://www.bluetooth.com/NR/rdonlyres/F8E8276A-3898-4EC6-B7DA-E5535258B056/6545/Core_V21__EDR.zip

Čagalj, M., Hubaux, J-P., Čapkun, S., Rengaswamy, R., Tsigkogiannis, I. and Srivastava, M. (2006a) 'Integrity (I) codes: message integrity protection and authentication over insecure channels', *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pp.280–294, ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1624018

Čagalj, M., Čapkun, S. and Hubaux, J-P. (2006b) 'Key agreement in peer-to-peer wireless networks', *Proceedings of the IEEE*, Vol. 94, No. 2, pp.467–478, ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1580514

Diffie, W. and Hellman, M.E. (1976) 'New directions in cryptography', *IEEE Transactions on Information Theory, IT-22*, Vol. 22, pp.644–654, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1055638

Dolev, D. and Yao, A.C. (1983) 'On the security of public key protocols', *IEEE Transactions on Information Theory*, Vol. 29, No. 2, pp.198–208, ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1056650.

Gehrmann, C., Mitchell, C. and Nyberg, K. (2004) 'Manual authentication for wireless devices', *RSA Crypto-Bytes*, Vol. 7, No. 1, pp.29–37, http://www.rsa.com/rsalabs/cryptobytes/Spring_2004_Cryptobytes.pdf.

Goodrich, M.T., Sirivianos, M., Solis, J., Tsudik, G. and Uzun, E. (2006) 'Loud and clear: human-verifiable authentication based on audio', *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1648797

Heydt-Benjamin, T.S., Bailey, D.V., Fu, K., Juels, A. and OHare, T. (2007) 'Vulnerabilities in first-generation RFID-enabled credit cards', *Proceedings of Eleventh International Conference on Financial Cryptography and Data Security*, Volume 4886 of Lecture Notes in Computer Science, Springer-Verlag, Lowlands, Scarborough, Trinidad/Tobago, pp.2–14, http://prisms.cs.umass.edu/~kevinfu/papers/RFID-CC-manuscript.pdf

Kivinen, T. and Kojo, M. (2003) *RFC3526: More Modular Exponential (MODP) Diffie-Hellman Groups for Internet Key Exchange (IKE)*, http://www.ietf.org/rfc/rfc3526.txt

Kuo, C., Walker, J. and Perrig, A. (2007) 'Low-cost manufacturing, usability, and security: an analysis of bluetooth simple pairing and Wi-Fi protected setup', *Proceedings of the Usable Security 2007 Workshop*, Lowlands, Scarborough, Trinidad/Tobago, http://usablesecurity.org/papers/kuo.pdf

Larsson, J-O. (2001) 'Higher layer key exchange techniques for bluetooth security', *Open Group Conference*, Amsterdam, 24 October.

Laur, S., Asokan, N. and Nyberg, K. (2005) *Efficient Mutual Data Authentication Using Manually Authenticated Strings*, Cryptology ePrint Archive, Report 2005/424, eprint.iacr.org/2005/424.pdf.

Laur, S. and Nyberg, K. (2006) 'Efficient mutual data authentication using manually authenticated strings', in Pointcheval, D. (Ed.): *The 5th International Conference on Cryptology and Network Security*, Volume 4301 of Lecture Notes in Computer Science, Springer, Suzhou, China, pp.90–107, http://www.springerlink.com/content/w152n0455673652k/

McCune, J.M., Perrig, A. and Reiter, M.K. (2005) 'Seeing-is-believing: using camera phones for human-verifiable authentication', *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pp.110–124, ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1425062

National Institute of Standards and Technology (2000) *Digital Signature Standard (DSS)*, US Department of Commerce, http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf

Newman, R., Gavette, S., Yonge, L. and Anderson, R. (2006) 'Protecting domestic power-line communications', *Proceedings of The Second Symposium on Usable Privacy and Security*, pp.122–132, http://portal.acm.org/citation.cfm?id=1143120.1143136

Newman, R., Yonge, L., Gavette, S. and Anderson, R. (2007) 'HomePlug AV security mechanisms', *Proceedings of The International Symposium on Power Line Communications and Its Applications*, pp.366–371, ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4231726

Pasini, S. and Vaudenay, S. (2006) 'SAS-based authenticated key agreement', *Proceedings of The 9th International Workshop on Theory and Practice in Public Key Cryptography*, Volume 3958 of Lecture Notes in Computer Science, Springer-Verlag, pp.395–409, http://www.springerlink.com/content/r42826j7335254q2/

Saxena, N., Ekberg, J-E., Kostiainen, K. and Asokan, N. (2006) 'Secure device pairing based on a visual channel (short paper)', *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pp.306–313, http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1624021

Soriente, C., Tsudik, G. and Uzun, E. (2007) *HAPADEP: Human Asisted Pure Audio Device Pairing*, Technical report, Cryptology ePrint Archive, Report 2007/039, eprint.iacr.org/2007/093.pdf

Stajano, F. and Anderson, R. (1999) 'The resurrecting duckling: security issues for ad-hocwireless networks', *Proceedings of the 7th International Workshop on Security Protocols*, Volume 2133 of Lecture Notes in Computer Science, Springer-Verlag, pp.172–194, http://www.springerlink.com/content/ru2015q381304428/

Suomalainen, J., Valkonen, J. and Asokan, N. (2007) 'Security associations in personal networks: a comparative analysis', *Proceedings of the 4th European Workshop on Security and Privacy in Ad-hoc and Sensor Networks*, Volume 4572 of Lecture Notes in Computer Science, Springer-Verlag, pp.43–57, http://www.springerlink.com/content/dk04356586jg4g00/

USB Implementers Forum (2006) *Wireless USB Specification. Association Models Supplement. Revision 1.0*, http://www.usb.org/developers/wusb/

USB Implementers Forum (2007) *Association Models Supplement to the Certified Wireless Universal Serial Bus Specification – Frequently Asked Questions*, http://www.usb.org/developers/wusb/WUSB_AM_FAQ_2007_06_19.pdf

Valkonen, J., Toivonen, A. and Karvonen, K. (2007) 'Usability testing for secure device pairing in home networks', *UbiComp 2007 Workshop Proceedings*, September, Innsbruck, Austria, pp.457–462.

Varshavsky, A., Scannell, A., LaMarca, A. and de Lara, E. (2007) 'Amigo: proximity-based authentication of mobile devices', *Proceedings of the Ninth International Conference on Ubiquitous Computing*, Volume 4717 of Lecture Notes in Computer Science, Springer-Verlag, pp.253–270, http://www.springerlink.com/content/37v827165x571333/

Vaudenay, S. (2005) 'Secure communications over insecure channels based on short authenticated strings', *Advances in Cryptology – CRYPTO 2005*, Volume 3621 of Lecture Notes in Computer Science, Springer-Verlag, pp.309–326, http://www.springerlink.com/content/5wak8q5hedk2fe4n/

Wi-Fi Alliance (2007) *Wi-Fi Protected Setup Specification*, Wi-Fi Alliance Document, available at http://www.wi-fi.org/wifi-protected-setup/

Zimmermann, P.R. (1996) *Pgpfone: Pretty Good Privacy Phone Owner's Manual*, version 1.0 beta 5, appendix c. http://web.mit.edu/network/pgpfone/manual/#PGP000057

Notes

[1]http://bluetooth.org
[2]http://wi-fi.org
[3]http://usb.org/wusb
[4]http://homeplug.org
[5]In such channels the standard Dolev-Yao attacker model is too strong.
[6]http://www.nfc-forum.org