Security Associations in Personal Networks: A Comparative Analysis*

Jani Suomalainen[1], Jukka Valkonen[2,3], and N. Asokan[2,3]

[1] VTT Technical Research Centre of Finland
Jani.Suomalainen@vtt.fi
[2] Helsinki University of Technology
Jukka.Valkonen@tkk.fi
[3] Nokia Research Center
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. There have been several different proposals to make this process of *associating* devices both easy-to-use and secure. Some of them have been adapted by emerging standard specifications. In this paper, we first present a taxonomy of protocols for creating security associations in personal networks. We then make use of this taxonomy in surveying and comparing association models proposed in several emerging standards. We also identify new potential attack scenarios.

Keywords: Personal networks, security association, survey.

1 Introduction

Short-range communication standards have brought a large number of new services to the reach of common users. For instance, standards for personal networking technologies such as Bluetooth, Wi-Fi, Wireless Universal Serial Bus (WUSB), and HomePlugAV 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 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 neighborhood 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.

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

* The full version of this paper is a Nokia Research Center technical report [14].

F. Stajano et al. (Eds.): ESAS 2007, LNCS 4572, pp. 43–57, 2007.

been proposals for schemes utilizing short passwords/checksums [5,7,15,16] or out-of-band channels . 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.

In this paper, we make a comparative analysis of proposed association models in different standards from a practical point of view. The surveyed standards are Bluetooth Secure Simple Pairing [13], Wi-Fi Protected Setup [17], Wireless USB Association Models [18], and HomePlugAV security modes [9].

The standards have some similarities. 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 them are targeted for personal networks and support multiple association models. Also, all of them utilize some sort of key establishment procedure for agreeing on a shared secret key between the devices.

The rest of the paper is organized as follows. Section 2 provides a systematic taxonomy of different protocols for key establishment. Section 3 describes how and which key establishment protocols and related association models are used in the surveyed standards. Section 4 presents a comparative analysis on the security of these standards. Section 5 describes novel attack scenarios where attackers utilize simultaneous availability of different association models.

2 Association Protocols

All of the association models we will survey in Section 3 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 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.

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.

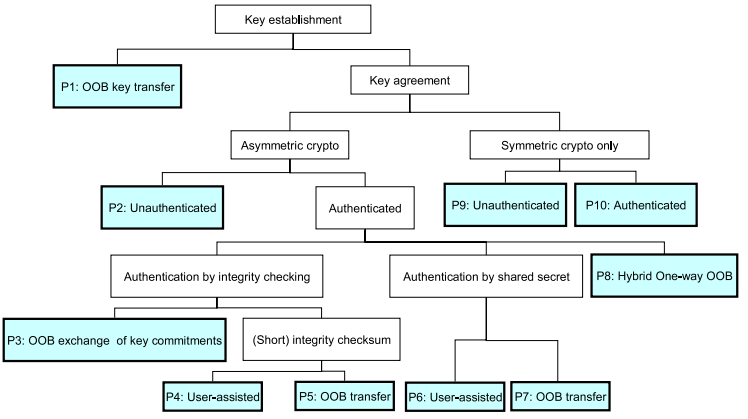

Fig. 1. Classification of Key Agreement Protocols

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

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 Diffie-Hellman key exchange [4].

Key agreement may be *unauthenticated* or *authenticated*. Unauthenticated symmetric key agreement **(P9)** is vulnerable even to passive eavesdroppers. Unauthenticated asymmetric key agreement **(P2)** is secure against passive eavesdroppers but is vulnerable to active MitM.

Key agreement based on symmetric key cryptography is authenticated by using a sufficiently long *pre-shared secret* **(P10)**. 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. There are two ways to authenticate by integrity-checking: by exchanging commitments to public keys, or by verifying a short integrity checksum. Now we take a closer look at the protocols involved in the different ways of authenticating key agreement based on asymmetric key cryptography.

Authentication by exchanging key commitments: Balfanz, et al., propose in [1] to exchange commitments to public keys using an out-of-band channel (**P3**). The commitments can be the public keys of the devices or their hashes. When the devices exchange public keys via the in-band channel, they can validate the authenticity of these public keys by using the information exchanged via the out-of-band channel.

The security of the protocols depends on the out-of-band channel being unspoofable. Also, the commitments of public keys must be 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: Several researchers have proposed authentication by using short checksums [11,7,16,15], sometimes referred to as "short authenticated string" protocols. In such protocols, each device computes a short checksum from the messages exchanged during the key agreement protocol. If the two checksums are the same, the exchange is authenticated. A basic three round mutual authentication protocol from [7] 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 cryptographic 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.

The check in the last step can be done in many different ways. One way is to ask the user to do the comparison **(P4)**. An alternative way is to do the check using a physical out-of-band channel **(P5)** as in [12].

To succeed, a MitM attacker has to choose random mumbers 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 $f()$. If $h()$ is collision-resistant, 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 n-bit value, then the chance of a MitM attacker succeeding is 2^{-n} because the attacker cannot influence the outcome of $f()$. 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. See [8] for a formal proof.

Authentication by (short) shared secret: Key exchange can also be authenticated using a short pre-shared secret passkey. A number of different methods

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 opens its commitment by sending 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.

Fig. 2. Authentication by Short Integrity Checksum

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 opening its commitment and 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 opening its commitment and 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.

Fig. 3. Round i of Authentication by (Short) Shared Secret

have been proposed for password-authenticated key exchange since Bellovin and Merrit introduced the idea in [3]. In Figure 3 we describe a variant of the MANA III protocol by Gehrmann, et al., originally described in [5]. 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 i^{th} round.

In each round, each party demonstrates its knowledge of P_i. A MitM 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 $P_i, i = 2 \ldots n$, the attacker cannot successfully complete the protocol run (recall that P is a *one-time* passkey). With n-bit passkey and k rounds the probability for a successful MitM attack is $2^{-(n-\frac{n}{k})}$. As in the case of short authentication string, the MitM success probabilities do not depend on additional assumptions about the attacker's computational capabilities.

There are many different ways for arranging for both devices to know the same P. One way is to have the user as the intermediary **(P6)**: the user may choose P and enter it into both devices, or one device may show a value for P which the user is asked to enter into the second device. Alternatively, P may be transported from one device to another using an out-of-band channel **(P7)**.

Hybrid authentication: Hybrid authentication protocols are used to achieve mutual authentication when only a one-way out-band-channel is available **(P8)**. 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 on message M using a key K.

The security of the protocol depends on the out-of-band being secret and unspoofable, as well as on strength of the commitment function $h()$ and the message authentication code function $c()$.

1. D_1 picks a long random value R_1, computes a commitment c to public key PK_1 as $C_1 = h(PK_1, R_1)$ and sends C_1 and secret S using OOB channel
   $\mathsf{D}_1 \Rightarrow \mathsf{D}_2$: S, C (OOB)
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 $\hat{C}_1 \stackrel{?}{=} h(\hat{PK}_1, \hat{R}_1)$ and aborts if it does not hold. Otherwise, D_2 picks its own long random value R_2, computes $C_2 = c(\hat{PK}_1|PK_2|\hat{R}_1|R_2, \hat{S})$ 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_2
4. D_1 checks if $\hat{C}_2 \stackrel{?}{=} c(PK_1|\hat{PK}_2|R_1|\hat{R}_2, S)$ and aborts if it does not hold.

Fig. 4. Hybrid Authentication Protocol

3 Association Models in Standards for Personal Networks

In this section, we survey the association models proposed in four emerging standards [13,17,18,9]. We then compare them by referring to the classification presented in Section 2.

3.1 Bluetooth Secure Simple Pairing

Bluetooth Secure Simple Pairing (SSP) [13] is a standard developed by Bluetooth Special Interest Group. It 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 [10]. 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 Out-of-band 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: **P4**). 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 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: **P6**). 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: **P2**) to protect against passive eavesdroppers but not against MitM attacks.

Out-of-band model is intended to be used with different out-of-band channels, in particular with Near Field Communication technology. Device D_A uses the out-of-band 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 out-of-band channel to send r_b and C_b. If out-of-band communication is bidirectional, mutual authentication is achieved by each party verifying that the peer's public key matches the commitment received via the out-of-band channel. (Figure 1: **P3**).

If the out-of-band channel is two way, then message 1 and message 2 will both be sent. Mutual authentication is complete at the end of step 2.

If the out-of-band channel is only one way, the party receiving the out-of-band message can authenticate the public key of its peer. However, the party sending the out-of-band 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 message authentication code. In the one-way out-of-band case, the message authentication code serves as a proof-of-knowledge of the shared secret r received out-of-band. This is the hybrid authentication protocol **P8** (Figure 4).

Peer discovery: In current 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 neighborhood that are publicly discoverable and asks the user to choose the right one to pair with. In SSP out-of-band association model, device addresses are sent via the out-of-band 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 SSP implementations 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 initialization of the session. If the association process is initiated by out-of-band interaction, and security-information is sent through the out-of-band channel, then the out-of-band 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.

3.2 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 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, out-of-band, and push-button configurations.

In-band configuration enables associations based on a shared secret passkey (Figure 1: **P6**). 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:

1. Exchange of public key commitments (Figure 1: **P3**), typically intended for two-way NFC interfaces, where the entire Diffie-Hellman exchange and the delivery of access keys takes place over the out-of-band channel.
2. Unencrypted key transfer (Figure 1: **P1**). An access key is transmitted from a registrar to enrollees in unecrypted form, either using USB-flash drives or NFC-tokens.
3. 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 out-of-band key transfer (Figure 1: **P1**).

Push button configuration is an optional method that provides an unauthenticated key exchange (Figure 1: **P2**). The user initiates the Push button configuration (PBC) by conditioning the enrollee (e.g., by pushing a button), and then, within 120 seconds 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 PBC. Access points are supposed to respond affirmatively only when their registrar has been conditioned by the user for PBC. If a device or registrar sees multiple peers ready to start PBC, 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 neighborhood 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.

3.3 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 [18] supports two association models for creating trust relationships between WUSB hosts and devices:

Cable model uses out-of-band key transfer (Figure 1: **P1**) and utilizes 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: **P4**). 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.

Peer discovery: The association is initialized 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.

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.

3.4 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 [9]:

Simple connect mode uses unauthenticated symmetric crypto based key agreement to agree on a shared key (Figure 1: **P9**). 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.

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: **P10**).

Optional modes enable alternative 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.

MitM attacks are prevented in simple connect mode by utilizing 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 MitM trying to negotiate tone maps with the legitimate endpoints will be detected.

Passive eavesdropping in the 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.

4 Comparison of Proposed Association Models

In this section, we summarize and compare the security levels provided by the different association models discussed in Section 3. A comparative summary of models' security characteristics are presented in Table 1.

Table 1. Comparison of Security Characteristics of Association Models

Association Model	**Offline Attacks**		**Online Active Attacks**		
	Protection	Work[1]	Protection	Success Probability	Work[2]
Bluetooth Simple Pairing					
Numeric Comparison	DH	2^{80} [2]	6 digit checksum	10^{-6}	2^{128}
Just Works	DH	2^{80} [2]	-	1	0
Passkey Entry	DH	2^{80} [2]	6 digit passkey	10^{-6}	2^{128}
Out-of-band	DH	2^{80} [2]	OOB security	-	2^{128}
Wi-Fi Protected Setup					
In-band	DH	2^{90} [6]	8 digit passkey	10^{-4}	2^{256}
In-band + OOB [3]	DH	2^{90} [6]	OOB security	2^{-128}	2^{256}
Out-of-band	OOB	2^{90} [6]	OOB security	-	-
PushButton	DH	2^{90} [6]	-	1	0
WUSB Association Models					
Numeric Model	DH	2^{128} [2]	2/4 digit checksum	10^{-2} or 10^{-4}	2^{256}
Cable Model	OOB	2^{128} [2]	OOB	-	-
HomePlugAV Protection Modes					
Simple Connect	SNR	Assumed high	Traffic monitoring	Assumed low	Assumed high
Secure Mode	AES	2^{72}	passkey	2^{-72}	2^{72}

[1] Rough work effort estimates based on Table 2 of [2] and Section 8 of [6].
[2] Work effort to break commitments exchanged.
[3] OOB passkey + checksum.

4.1 Offline Attacks

The out-of-band association models rely on the secrecy of out-of-band 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 1, we use some recent sources [6,2] 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: namely the signal-to-noise ratio (SNR) make it difficult for an attacker to eavesdrop. The HomePlugAV Secure Mode uses symmetric key encryption as protection.

4.2 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. Bluetooth Simple Pairing numeric comparison model uses 6-digit checksums leading to a success probability of $\frac{1}{1000000}$. 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. All of these use hash functions with 128-bit outputs to compute commitments. In principle, a man-in-the-middle who can find a second pre-image of a hash commitment, *during* the key agreement process can also succeed. We show this in Table 1, in the "Work" subcolumn under the "Online Active Attacks" column by indicating the amount of *on-line* work the attacker has to perform in order to succeed. In this case, assuming that the hash function is strong, and requires exhaustive search to find a second pre-image we use the figure 2^{128}.

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 Simple Pairing 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 k^{th} 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 on-line 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 on-line work effort is 2^{72}.

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

An interesting implication of Table 1 is that in all the systems (except HomePlug AV), the work factor for online active attack *far exceeds* the work factor for offline attack. This reflects the difficulties in comparing the relative security of cryptographic hash functions with that of public key algorithms.

4.3 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.

5 Attacks Against Multiple Association Models

Simultaneous support for multiple association models may be utilized in different attacks. In this section, we examine such threats.

Consider specifications that support an unauthenticated association model as well as user-assisted comparison of integrity checksums. An example is a Bluetooth Simple Pairing device that supports the numeric association model and the 'just works' model. Figure 5 illustrates a MitM 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.

To get an idea about whether such user confusion is likely, we 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

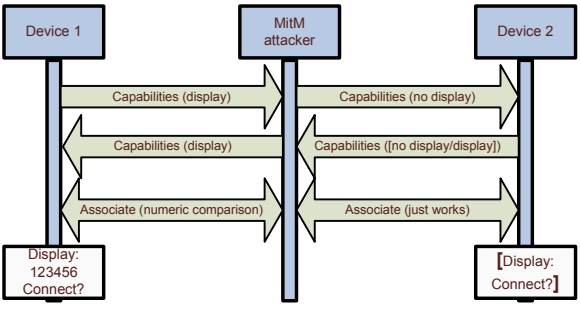

Fig. 5. Man-in-the-middle between Different Association Models

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 MitM 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 confirmations.

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 Simple Pairing numeric association, 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.

6 Conclusions

New standards for associating devices in personal networks are emerging. The objective of the new standards is to make the association process more user-friendly while improving the security at the same time. We surveyed the protocols and association models used in different standards specifications. We presented a systematic classification of protocols for human-mediated establishment of session keys. 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, as discussed in the full version of this paper ([14] Section 6). However, how exactly to design the user dialogs to preserve security without harming usability remains an open issue.

Acknowledgments

We thank Dan Forsberg, Kristiina Karvonen, Janne Marin, Seamus Moloney, Kaisa Nyberg and Gene Tsudik for highly valuable feedback. We are particularly grateful to Kaisa for her many suggestions for improving the paper.

The work of the second author is supported by the InHoNets project funded by TEKES.

References

1. Balfanz, D. et al.: Talking to strangers: authentication in ad-hoc wireless networks. In: Proceedings of the Network and Distributed System Security Symposium (2002)
2. Barker, E. et al.: Recommendation for key management - part 1: General (revised), (2006) http://csrc.nist.gov/CryptoToolkit/kms/SP800-57Part1_6-30-06.pdf
3. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: Steven, M. (ed.) Proceedings of the 1992 IEEE Symposium on Security and Privacy, pp. 72–84 (1992)
4. Diffie, W., Hellman, M.E.: New Directions In Cryptography. IEEE Transactions on Information Theory IT-22, 644–654 (1976)
5. Gehrmann, C. et al.: Manual authentication for wireless devices. RSA CryptoBytes (2004)
6. Kivinen, T., Kojo, M.: RFC3526: More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE) (May 2003) http://www.ietf.org/rfc/rfc3526.txt
7. Laur, S. et al.: Efficient Mutual Data Authentication Using Manually Authenticated Strings. Cryptology ePrint Archive, Report 2005/424 (2005)
8. Laur, S., Nyberg, K.: Efficient mutual data authentication using manually authenticated strings. In: Proceedings of the 5th International Conference on Cryptology and Network Security, pp. 90–107 (2006)
9. Newman, R., et al.: Protecting domestic power-line communications. In: Proc. of The Second Symposium on Usable Privacy and Security, pp. 122–132 (2006)
10. NIST: National Institute of Standards and Technology. Digital Signature Standard (DSS). U.S. Department of Commerce (January 2000)
11. Pasini, S., Vaudenay, S.: SAS-based Authenticated Key Agreement. In: Proceedings of The 9th International Workshop on Theory and Practice in Public Key Cryptography, pp. 395–409 (2006)
12. Saxena, N., et al.: Secure device pairing based on a visual channel (short paper). In: Proc. of the 2006 IEEE Symposium on Security and Privacy, pp. 306–313 (2006)
13. Simple Pairing Whitepaper. Bluetooth Special Interest Group (2006) http://www.bluetooth.com/Bluetooth/Apply/Technology/Research/Simple_Pai ring.htm
14. Suomalainen, J. et al.: Security associations in personal networks: A comparative analysis. Technical Report NRC-TR-2007-004, Nokia Research Center (2007) http://research.nokia.com/tr/NRC-TR-2007-004.pdf
15. Vaudenay, S.: Secure communications over insecure channels based on short authenticated strings. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 309–326. Springer, Heidelberg (2005)
16. Čagalj, M., Čapkun, S., Hubaux, J.-P.: Key agreement in peer-to-peer wireless networks. In: Proceedings of the IEEE (Special Issue on Cryptography and Security), pp. 467–478 (2006)
17. Wi-Fi Alliance. Wi-Fi Protected Setup Specification. Wi-Fi Alliance Document (January 2007)
18. Wireless USB Specification. Association Models Supplement. Revision 1.0. USB Implementers Forum (2006) http://www.usb.org/developers/wusb/