



Research Center

NRC-TR-2007-004

Security Associations in Personal Networks: A Comparative Analysis

Jani Suomalainen¹, Jukka Valkonen² and N. Asokan²

¹VTT Technical Research Centre of Finland
<http://www.vtt.fi/>

²Nokia Research Center Helsinki
<http://research.nokia.com>
9.1.2007

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 and discuss how to mitigate them.

Index Terms:

personal networks
security association
standards
comparative survey
attacks

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 Bluetooth¹, Wi-Fi², Wireless Universal Serial Bus (WUSB)³, and HomePlugAV⁴ 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.⁵

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 schemes utilizing short passwords/checksums [5, 9, 21, 22] or out-of-band channels, such as physical [19], audio [6], visual [11, 17] or very short-range wireless 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 more than one association model. 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 Simple Pairing [18], Wi-Fi Protected Setup [23], Wireless USB Association Models [24], and HomePlugAV security modes [7].

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 the standards 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. Finally, Section 6 presents a discussion on potential countermeasures against these attacks.

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

¹ <http://bluetooth.org>

² <http://wi-fi.org>

³ <http://usb.org/wusb>

⁴ <http://homeplug.org>

⁵ First, when there are many devices or networks in the scanned neighborhood, 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.

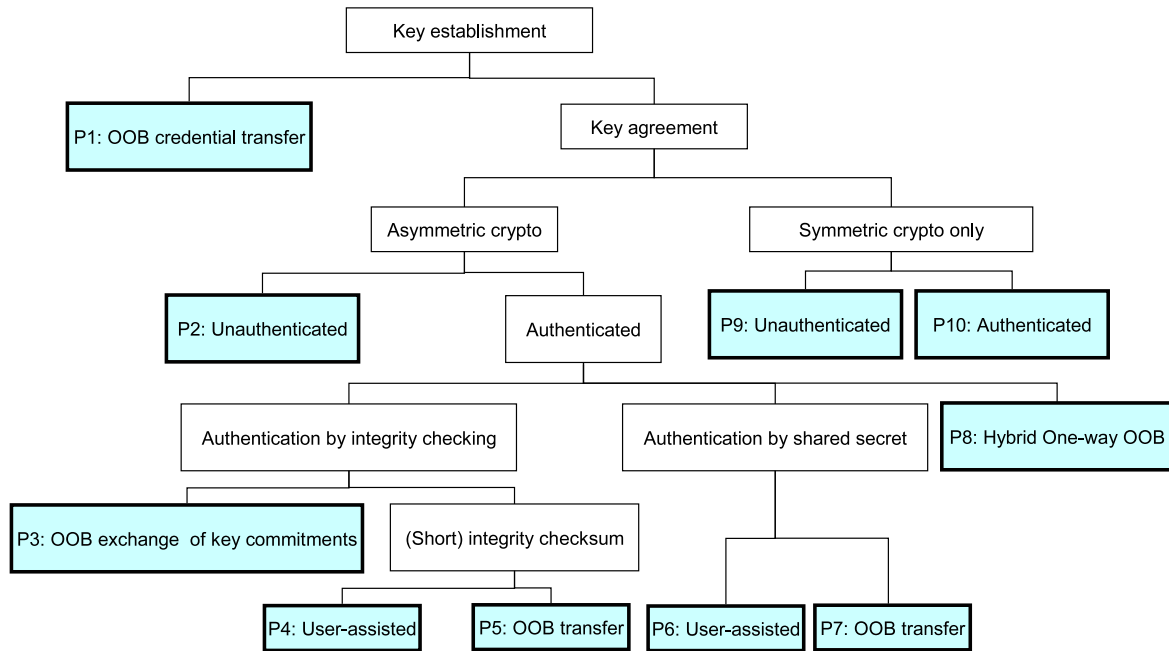


Fig. 1. Classification of Key Agreement Protocols

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.

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 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 man-in-the-middle (MitM).

The only way to authenticate key agreement based on symmetric key cryptography is 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 [16, 9, 22, 21], 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 [9] 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 ($\hat{\cdot}$) symbol is used to denote the receiver’s view of a value sent in protocol message.

1. D_1 generates a long random value R_1 , computes commitment $h = h(R_1)$ and sends it to D_2
 $D_1 \rightarrow D_2: h$
2. D_2 generates a long random value R_2 and sends it to D_1
 $D_1 \leftarrow D_2: R_2$
3. D_1 opens its commitment by sending R_1 to D_2
 $D_1 \rightarrow D_2: R_1$
4. D_2 checks if $\hat{h} \stackrel{?}{=} h(\hat{R}_1)$. If equality holds, D_2 computes $v_2 = f(\hat{PK}_1, PK_2, \hat{R}_1, R_2)$, otherwise it aborts.
 D_1 computes $v_1 = f(PK_1, \hat{PK}_2, R_1, \hat{R}_2)$.
5. Both devices check if v_1 equals v_2 .

Fig. 2. Authentication by Short Integrity Checksum

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**): Each device displays its own string to the user and ask whether it is the same as what the other device is displaying. If the checksums 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 a physical out-of-band channel (**P5**) as in [17].

To succeed a man-in-the-middle attacker has to choose such R'_1 and R'_2 that $f(PK'_1, PK_2, R'_1, R_2)$ is the same as $f(PK_1, PK'_2, R_1, R'_2)$ where PK'_1 and PK'_2 are attacker’s public keys. 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 man-in-the-middle attacker succeeding is 2^{-n} because the attacker cannot influence the outcome of $f()$. This success probability is unconditional; it does not rely on any assumptions about the computational capabilities of the attacker. See [10] 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 have been proposed for password-authenticated key exchange since Bellare and Merritt introduced the idea in [3]. In Figure 3 we describe a variant of the MANA III protocol by Gehrman, et al., in [5]. It uses a one-time passkey P to authenticate PK_1 and PK_2 . P is split into k pieces, labelled $P_1 \dots P_k$. 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 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 $P_i, i = 2 \dots k$, 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 man-in-the-middle attack is $2^{-(n-\frac{n}{k})}$. As in the case of short authentication string, the man-in-the-middle success probabilities are unconditional.

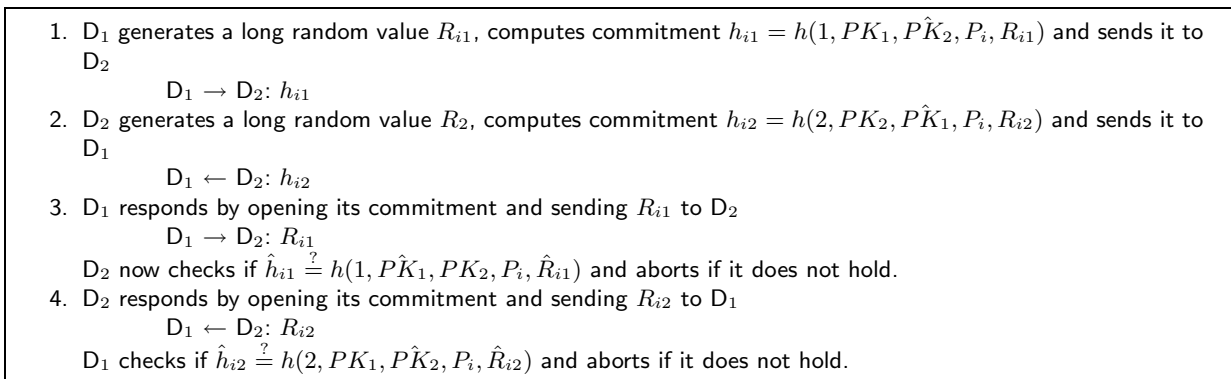


Fig. 3. Round i of Authentication by (Short) Shared Secret

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**). In such methods, as there is no need for a human to transfer the shared secret between the devices, it can be longer, thus making probability for a successful attack smaller. Note that the passkey 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-of-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 .

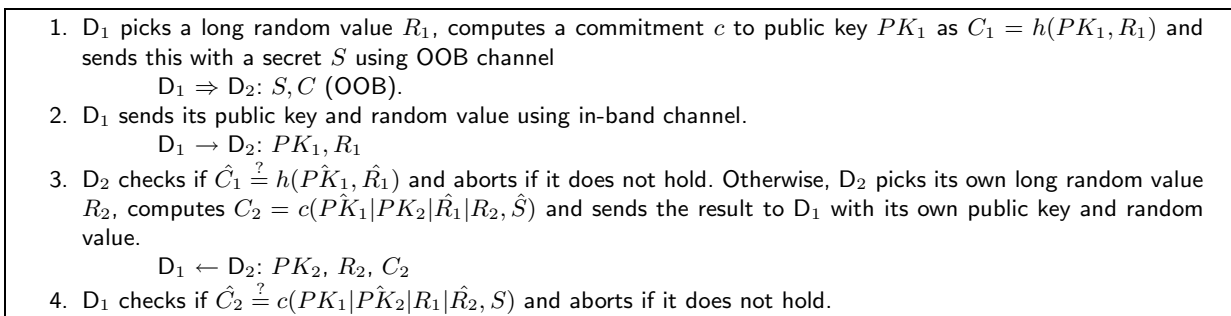


Fig. 4. Hybrid Authentication Protocol

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 $c()$.

3 Association Models in Standards for Personal Networks

In this section, we survey the association models proposed in four emerging standards [18, 23, 24, 14]. We then compare them by referring to the classification presented in Section 2.

3.1 Bluetooth Simple Pairing

Bluetooth Simple Pairing [18] 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 [15]. 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, as we will see below).

Simple Pairing 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. The exact instantiation is depicted in Figure 5. At this point the devices have already completed phase 1 and possess both public keys PK_a and PK_b .

1. D_B computes a commitment $C_b = f1(PK_b, \hat{PK}_a, N_b, 0)$ using the one-way function^a $f1$ with 128-bit output and 128-bit fresh random nonce N_b and sends the value to D_A .
 $D_A \leftarrow D_B: C_b$
 2. D_A responds by sending 128-bit fresh random nonce N_a to D_B .
 $D_A \rightarrow D_B: N_a$
 3. D_B opens the commitment by sending N_b to D_A .
 $D_A \leftarrow D_B: N_b$
 4. D_A recomputes C_b as $f1(\hat{PK}_b, PK_a, \hat{N}_b, 0)$ and checks if $C_b \stackrel{?}{=} \hat{C}_b$. If it is, D_A computes short a checksum using one-way function^b g as $V_a = g(PK_a, \hat{PK}_b, N_a, \hat{N}_b)$, otherwise D_A aborts.
 D_B computes checksum $V_b = g(\hat{PK}_a, PK_b, \hat{N}_a, N_b)$.
 Each device displays the six least significant digits of its own checksum.
 5. Each device prompts the user to check and confirm if the checksum it displays is the same as the checksum displayed by the peer device.
- ^a $f1(U, V, X, Z) = \text{HMAC-SHA-256}_X(U|V|Z)/2^{128}$
^b $g(U, V, X, Y) = \text{SHA-256}(U|V|X|Y) \bmod 2^{32}$

Fig. 5. Bluetooth Simple Pairing: Numeric Comparison Model

The protocol is straightforward implementation of authentication protocol for **P4** depicted in Figure 2, where D_B plays the role of D_1 , D_A plays the role of D_1 . Similar to Figure 2, the protocol structure ensures that the D_A and D_B have to choose N_a and N_b , respectively, independently of each other.

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. The exact instantiation of the phase 2 protocol is depicted in Figure 6. As before, phase 1 has been completed and both devices know PK_a and PK_b . This is essentially the protocol in Figure 3 executed 20 times.

‘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 man-in-the-middle attacks.

Let r_a and r_b denote the value of the 6-digit passkey as seen by D_A and D_B respectively (in the normal case, r_a and r_b have the same value). r_{ai} and r_{bi} denote the i^{th} most significant bit of r_a and r_b respectively. The devices execute the following 20 times:

1. D_A generates a 128-bit random value N_{ai} , computes commitment $C_{ai} = f1(PK_a, P\hat{K}_b, N_{ai}, r_{ai})$ and sends it to D_B .
 $D_A \rightarrow D_B: C_{ai}$
2. D_B generates a 128-bit random value N_{bi} , computes commitment $C_{bi} = f1(PK_b, P\hat{K}_a, N_{bi}, r_{bi})$ and sends it to D_A .
 $D_A \leftarrow D_B: C_{bi}$
3. D_A sends N_{ai} to D_B .
 $D_A \rightarrow D_B: N_{ai}$
 D_B recomputes C_{ai} as $f1(P\hat{K}_a, PK_b, \hat{N}_{ai}, r_{bi})$ and checks if $C_{ai} \stackrel{?}{=} \hat{C}_{ai}$. If it is not, D_B aborts.
4. Otherwise D_B sends N_{bi} to D_A .
 $D_A \leftarrow D_B: N_{bi}$
 D_A recomputes C_{bi} as $f1(P\hat{K}_b, PK_a, \hat{N}_{bi}, r_{ai})$ and checks if $C_{bi} \stackrel{?}{=} \hat{C}_{bi}$. If it is not, D_B aborts the protocol.

Fig. 6. Bluetooth Simple Pairing: Passkey Entry Model

The protocol used in this model is the same as in the numeric comparison model, but the integrity check values are accepted by the devices without checking for equality. The specification allows a device to optionally ask the user for a confirmation to accept the connection, without displaying the checksum or asking for an equality check.

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**).

The phase 2 protocol instantiation is depicted in Figure 7. As before, the devices are expected to know PK_a and PK_b at the end of phase 1.

D_A sets r_a to a fresh 128-bit random value and r_b to 0; D_A computes commitment C_a as $f1(PK_a, PK_a, r_a, 0)$.
 D_B sets r_b to a fresh 128-bit random value and r_a to 0; D_B computes commitment C_b as $f1(PK_b, PK_b, r_b, 0)$.

1. D_A then sends its device address along with r_a , and C_a via the out-of-band channel.
 $D_A \Rightarrow D_B: C_a, r_a, A$
 If D_B receives an out-of-band message, it updates r_a to be the received value \hat{r}_a , recomputes C_a as $f1(P\hat{K}_a, P\hat{K}_a, \hat{r}_a, 0)$ and checks if $C_a \stackrel{?}{=} \hat{C}_a$. If the equality does not hold, D_B aborts.
2. D_B similarly sends its own device address along with r_b , and C_b .
 $D_A \leftarrow D_B: C_b, r_b, B$
 If D_A receives an out-of-band message, it updates r_b to be the received value \hat{r}_b , recomputes C_b as $f1(P\hat{K}_b, P\hat{K}_b, \hat{r}_b, 0)$ and checks if $C_b \stackrel{?}{=} \hat{C}_b$. If the equality does not hold, D_A aborts.
3. D_A chooses a fresh random nonce N_a and sends it to D_B in-band.
 $D_A \rightarrow D_B: N_a$
4. D_B chooses a fresh random nonce N_b and sends it to D_A in-band.
 $D_A \leftarrow D_B: N_b$

Fig. 7. Bluetooth Simple Pairing: Out-of-band Model

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 Simple Pairing 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. The confirmation phase is depicted in Figure 8. The notations are as follows: A and B denote the device addresses of D_A and D_B respectively; $IOCapA$ and $IOCapB$ are the user input/output capabilities exchanged between D_A and D_B in phase 1 of Simple Pairing; $DHKey$ is the Diffie-Hellman key computed using the public keys exchanged between D_A and D_B in phase 1. Both devices set the values of r_a and r_b already in phase 2: In the numeric association model, both devices set their r_a and r_b values to 0. In the passkey model, both devices set r_a and r_b to the value of the shared 6-digit passkey. In the out-of-band model D_A resets its r_a value to 0 either if it cannot send an out-of-band message, or if it receives r_b via out-of-band but learns via in-band that D_B was not able to read r_a out of band. D_B follows similar rules.

To see how this serves to complete mutual authentication in the case of one-way out-of-band channels, suppose the out-of-band channel in Figure 7 had been unidirectional from D_A to D_B . In this case, D_B would have received the secret r_a which will be included in the computation of E_b ⁶. E_b thus serves as a proof-of-knowledge of the shared secret r_a . In terms of the notations in the hybrid authentication protocol (Figure 4), r_a is the shared secret S and E_b serves as the message authentication code C_2 .

1. D_A computes a confirmation message E_a as $f_3(DHKey, N_a, \hat{N}_b, r_b, IOCapA, A, B)$ using a one way function^a f_3 and sends E_a to D_B .
 $D_A \rightarrow D_B: E_a$
 D_B recomputes E_a . If the recomputed value does not match the received value \hat{E}_a , D_B aborts.
 2. Otherwise, D_B computes E_b as $f_3(DHKey, N_b, \hat{N}_a, r_a, IOCapB, B, A)$ and sends E_b to D_A .
 $D_A \rightarrow D_B: E_b$
 D_A recomputes E_b . If the recomputed value does not match the received value \hat{E}_b , D_A aborts. Otherwise, the devices have successfully performed the exchange and can continue.
- ^a $f_3(X, A, B, C, D, E, F) = \text{HMAC-SHA-256}_X(A|B|C|D|E|F)/2^{128}$

Fig. 8. Bluetooth Simple Pairing: Confirmation Phase

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 Simple Pairing 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. Simple Pairing 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 Simple Pairing 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 Simple Pairing specification describes how these capabilities should be used to select the association model.

⁶ The Simple Pairing White Paper[18] incorrectly showed that r_b was included in the computation of E_b . This was reported to the Bluetooth SIG and will be corrected in the actual specification.

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) [12, 13] 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 to 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 an instantiation of the modified MANA III protocol in Figure 3 with two rounds ($k = 2$). The exact instantiation described in Figure 9, where the following notation is used:

- N_1, N_2 : 128-bit nonces chosen by enrollee and registrar respectively.
- PK_a and PK_b : D-H public keys (for the 1536-bit MODP group 5 defined in [8]) of enrollee and registrar respectively.
- M_j^* : The message M_j without the HMAC authenticator.
- $AuthKey$ and $KeyWrapKey$: Keys derived from the Diffie-Hellman key.
- $ENC_{Key}(\cdot)$: AES-CBC encryption using a 128-bit key Key .
- ES_i and RS_i : The random values used to prove knowledge of the i^{th} component of the passkey (similar to R_{i1} and R_{i2} in Figure 3).
- $EHash_i$ and $RHash_i$: Commitments used in the proof of knowledge (similar to h_{i1} and h_{i2} in Figure 3),

Unlike in Figure 3 each party sends both of its passkey commitments in a single message (M_3 and M_4 respectively). The commitment $EHash_i, i = 1, 2$ is computed as

$$PSK_i = 128 \text{ bits of HMAC-SHA256}_{AuthKey}(i^{th} \text{ half of passkey})$$

$$EHash_i = \text{HMAC-SHA256}_{AuthKey}(ES_i || PSK_i || PK_1 || PK_2)$$

Messages $M_3 - M_6$ constitute the two rounds of the authentication protocol. In Message M_8 , the access key for the network is delivered to the enrollee as ‘ConfigData’.⁷

As in the other passkey authentication mechanisms (Figures 3 and 6), 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 commitments $EHash_i$ includes $AuthKey$, which is derived from the Diffie-Hellman key).

In-band configuration can also be authenticated using hybrid authentication (Figure 1: **P8**) by transmitting the passkeys and key commitments using NFC-tokens or USB flash drives. This way, longer passkeys can be supported, as the users do not need to type the passkey into a device.

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. The OOB channel is used to transmit messages M_1 and M_2 (Figure 9) between the devices. No in-band communication takes place. The access keys are delivered in message M_2 .
2. 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. The same out-of-band channel can be used to configure multiple enrollees.

⁷ ConfigData also appears in Message M_7 . This is used when a registrar, acting as D_A takes ownership of an access point, acting as D_B , and initializes the network access key.

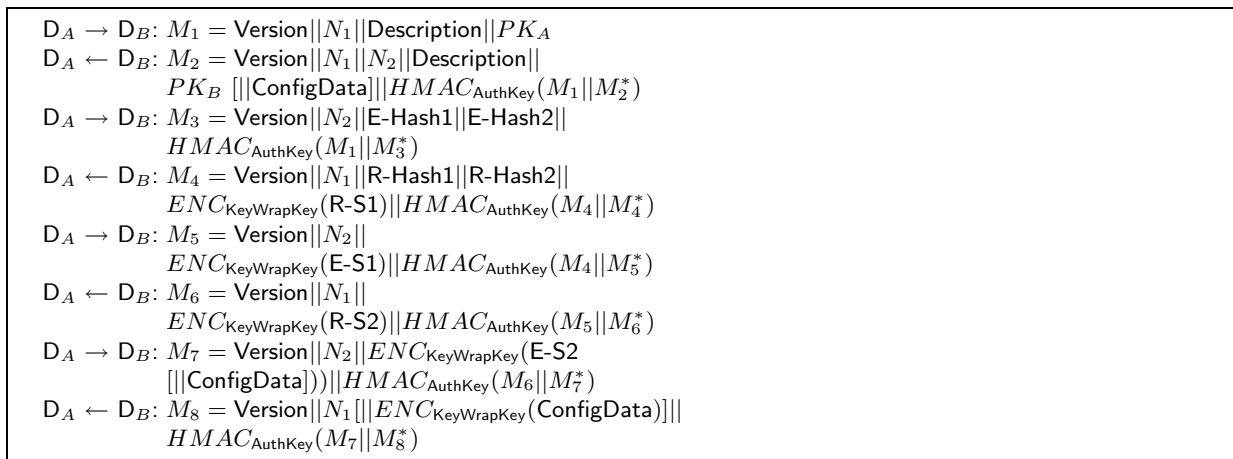


Fig. 9. Windows Connect Now-NET Registration Protocol

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. (The in-band interaction consists of messages M_1 and M_2 in Figure 9). From a security perspective, this is essentially out-of-band key transfer (Figure 1: **P1**). The advantage of the method is that if the flash drive is lost, no one except a device holding the encryption key is able to get the access key. The flash drive should be still kept secret since a man-in-the-middle does have the encryption key and can able to decrypt the access key from the drive.

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, they are required to abort the process and inform the user. Otherwise, they carry out the basic protocol, without a passkey.

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 [24] 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 (for the 3072-bit MODP group 15 defined in [8]) 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.

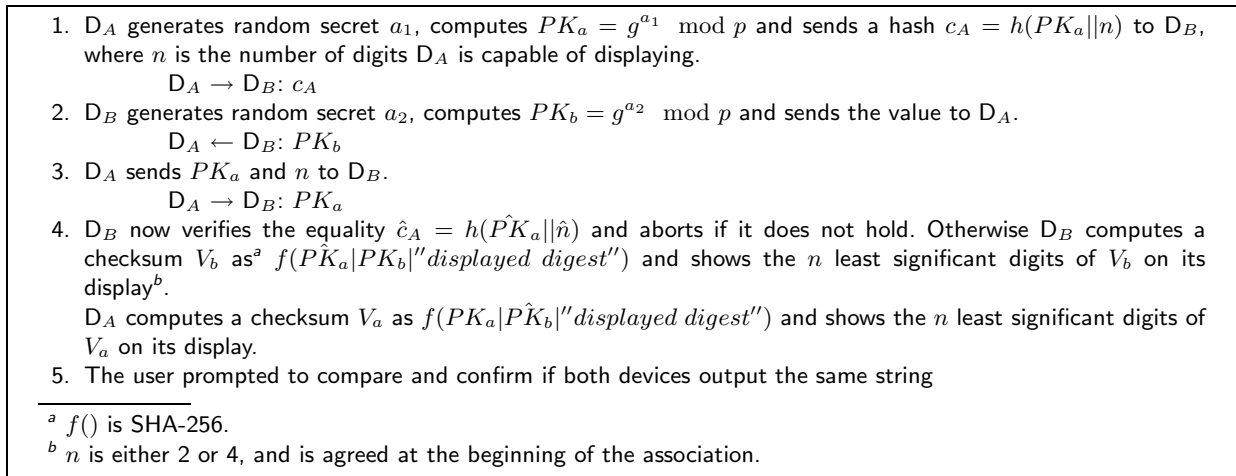


Fig. 10. Wireless USB: Numeric Association Model

The specification requires that WUSB hosts must support 4-digit checksums whereas WUSB devices must support either 2 or 4-digit checksums. The exact instantiation is described in Figure 10.

The protocol is similar to the ones described in Figures 2 and 5. The main difference is that in Figures 2 and 5, the commitments are computed from fresh random values, whereas in WUSB numeric association model the commitments are computed from the public Diffie-Hellman values. This implies that in WUSB numeric association models, each run of the protocol requires the use of fresh Diffie-Hellman keypairs.

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 numeric model the user sets a USB device to search for devices and a USB host to accept connections. The host advertises their 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 [14]:

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.

If a new device notices more than one controller, it uses signal strength to choose the right one. Still, there is a possibility that it may connect to the wrong controller. The user will notice this if/when a device does not work as expected, and must retry.

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 network membership or encryption keys 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 such as Bluetooth Pairing or Windows Connect Now.

Man-in-the-middle 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 man-in-the-middle 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. devices sharing the NMK can access the NEKs and thus join the network.

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. Figure 11 presents how the models can be mapped into the classification presented in Section 2. A comparative summary of models’ security characteristics are presented in Table 1.

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 effort” subcolumn under the “Offline Attacks” column of Table 1, we use some recent sources [8, 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 ballpark estimates only. Offline attack protection in HomePlugAV relies on the characteristics of the powerline 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’ in Simple Pairing, and ‘Push Button’ in WiFi Protected Setup) trade off protection against man-in-the-middle attacks, in return for increased ease-of-use. HomePlugAV Simple Connect also falls into this category.

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

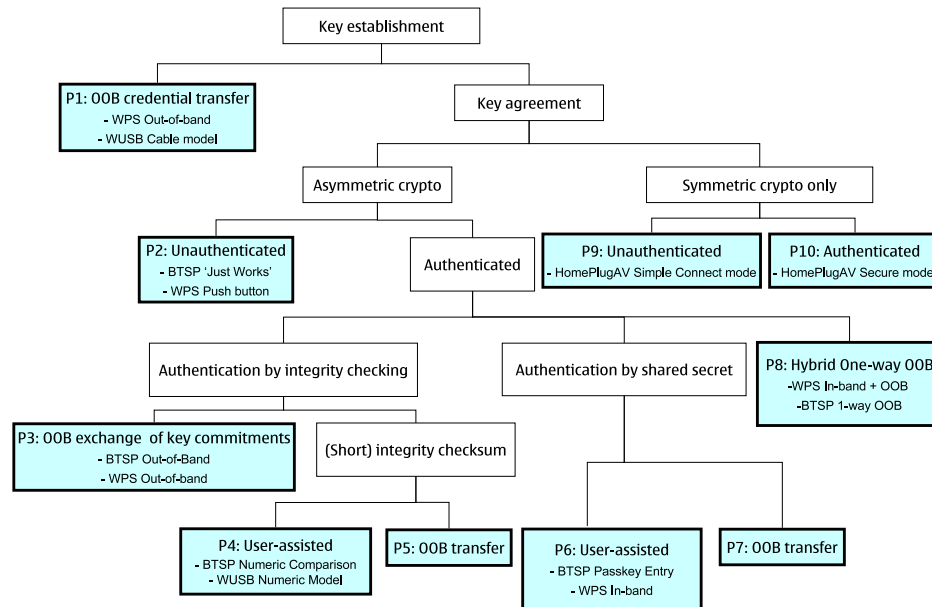


Fig. 11. Classification of Standardized Association Protocols

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 are unconditional since they 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 effort” 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 bits and k rounds the success probability for an online active attack 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 $\frac{1}{1000000}$ unconditional success probability. WPS network uses essentially the same protocol, but in two rounds only. This leads to unconditional 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. In 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} .

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.

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

Association Model	Offline Attacks		Online Active Attacks		
	Protection	Work effort ¹	Protection	Success Probability	Work effort ²
<i>Bluetooth simple pairing</i>					
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}
<i>Wi-Fi Protected Setup</i>					
In-band	DH	2^{90} [8]	8 digit passkey ³	10^{-4}	2^{256}
In-band + OOB for passkey, pubkey hash	DH	2^{90} [8]	OOB security	2^{-128}	2^{256}
Out-of-band	OOB	2^{90} [8]	OOB security	-	-
PushButton	DH	2^{90} [8]	-	1	0
<i>WUSB Association Models</i>					
Numeric model	DH	2^{128} [2]	2/4 digit checksum	10^{-2} or 10^{-4}	2^{256}
Cable model	OOB	2^{128} [2]	OOB	-	-
<i>HomePlugAV Protection Modes</i>					
Simple Connect	SNR	Assumed high	Traffic monitoring	Assumed low	Assumed high
Secure Mode	AES	2^{72}	12 char passkey ⁴	2^{-72}	2^{72}

Table 1. Comparison of security characteristics of association models

¹ Rough estimates based on Table 2 of [2] and Section 8 of [8]

² Work effort to break commitments exchanged

³ 4 digit passkeys are allowed, too

⁴ Permanent long-term secret

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 can 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.

5.1 Man-in-the-Middle between Different Association Models

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 12 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 has been educated to detect if displayed checksums are different. However, now, when only another device displays a checksum, the user may easily accept association without noticing any attack.

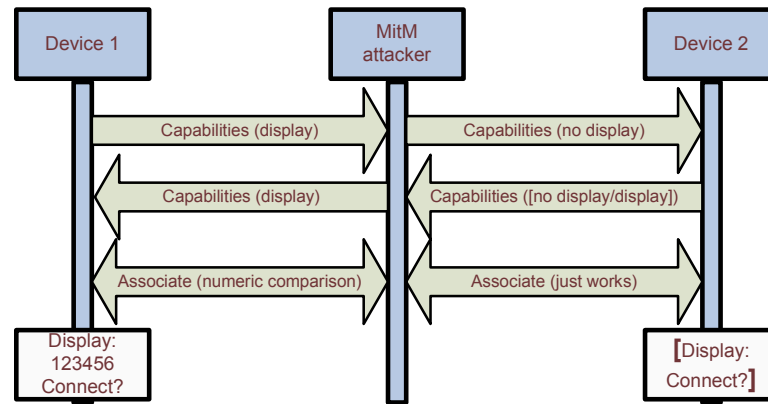


Fig. 12. Man-in-the-middle between Different Association Models

To get an idea about whether such user confusion is likely, we included the situation depicted in Figure 12 as a test scenario in one round of an on-going series of usability testing [20]. 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. We expect to include more details and analysis in a forthcoming report.

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 confirmations.

5.2 Unconditioned Associations

A scenario related to the attack on Figure 12 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, as illustrated in Figure 13. 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.

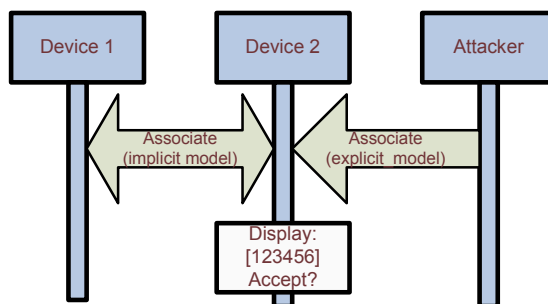


Fig. 13. Initiating Explicit Association in the Side of Implicit

5.3 Jamming Attacks

A man-in-the-middle attacker may try to prevent associations until a frustrated user decides to try the alternative less secure model as illustrated in Figure 14. The attack is applicable to situations where the end-user is allowed to select the association model. Particularly, when detecting that the HomePlugAV secure mode is used, an attacker may disturb communication until the user selects Simple Connect mode.

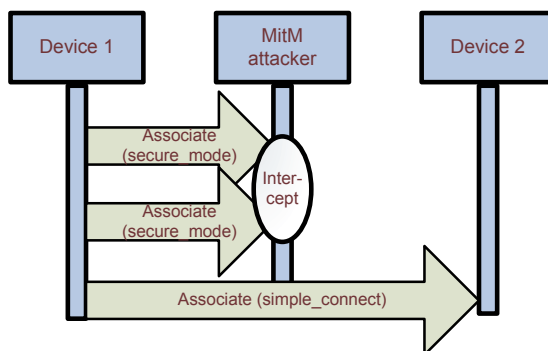


Fig. 14. Jamming a Secure Model to get the User to Switch into a Less Secure Model

6 Strengthening Devices

In this section, we discuss some implementation guidelines that can help address the kind of attacks identified in Section 5.

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 BTSP ‘Just Works’ or HomePlugAV Simple Connect models

should not be allowed to install or configure software, at least, without explicit authorization from the user. This precaution would help with bidding down attacks.

The man-in-the-middle attack between numeric comparison and unauthenticated protocols (Figure 12) could be addressed with two alternative strategies:

1. 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.
2. 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.

7 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, and discussed possible measures to reduce their impact. 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.

8 Acknowledgments

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

References

1. Dirk Balfanz, Diana K. Smetters, Paul Stewart, and H. Chi Wong. Talking to strangers: authentication in ad-hoc wireless networks. In *Proceedings of the Network and Distributed System Security Symposium*, pages 207–222, 2002.
2. Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. Recommendation for key management - part 1: General (revised), 2006. http://csrc.nist.gov/CryptoToolkit/kms/SP800-57Part1_6-30-06.pdf.
3. Steven M. Bellovin and Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *SP '92: Proceedings of the 1992 IEEE Symposium on Security and Privacy*, page 72, Washington, DC, USA, 1992. IEEE Computer Society.
4. Whitfield Diffie and Martin E. Hellman. New Directions In Cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, 1976.
5. Christian Gehrmann, Chris J. Mitchell, and Kaisa Nyberg. Manual authentication for wireless devices. *RSA Crypto-Bytes*, 7(1):29 – 37, Spring 2004.
6. Michael T. Goodrich, Michael Sirivianos, John Solis, Gene Tsudik, and Ersin Uzun. Loud and clear: Human-verifiable authentication based on audio. In *ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, page 10, Washington, DC, USA, 2006. IEEE Computer Society.
7. HomePlug AV whitepaper. HomePlug Powerline Alliance. <http://www.homeplug.org/>, 2005.
8. Tero Kivinen and Markku Kojo. RFC3526: More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), May 2003. <http://www.ietf.org/rfc/rfc3526.txt>.
9. Sven Laur, N. Asokan, and Kaisa Nyberg. Efficient Mutual Data Authentication Using Manually Authenticated Strings. Cryptology ePrint Archive, Report 2005/424, 2005. <http://eprint.iacr.org/>.

10. Sven Laur and Kaisa Nyberg. Efficient mutual data authentication using manually authenticated strings. In *Proceedings of the 5th International Conference on Cryptology and Network Security, Suzhou, China*, number 4301 in Lecture Notes in Computer Science, pages 90–107. Springer, 2006.
11. Jonathan M. McCune, Adrian Perrig, and Michael K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 110–124, Washington, DC, USA, 2005. IEEE Computer Society.
12. Windows Connect Now-NET. Version 1.0. Microsoft. [Http://www.microsoft.com/whdc/Rally/WCN-Netspec.mspix](http://www.microsoft.com/whdc/Rally/WCN-Netspec.mspix), 2006.
13. Windows Connect Now-UFD and Windows Vista Specification. Version 1.0. Microsoft. [Http://www.microsoft.com/whdc/Rally/WCN-UFD_Vistaspec.mspix](http://www.microsoft.com/whdc/Rally/WCN-UFD_Vistaspec.mspix), 2006.
14. Richard Newman, Sherman Gavette, Larry Yonge, and Ross Anderson. Protecting domestic power-line communications. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 122–132, New York, NY, USA, 2006. ACM Press.
15. NIST: National Institute of Standards and Technology. *DIGITAL SIGNATURE STANDARD (DSS)*. U.S. DEPARTMENT OF COMMERCE, January 2000. <http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf>.
16. Sylvain Pasini and Serge Vaudenay. Sas-based authenticated key agreement. In *Public Key Cryptography - PKC'06: The 9th international workshop on theory and practice in public key cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 395 – 409. Springer, 2006.
17. Nitesh Saxena, Jan-Erik Ekberg, Kari Kostiaainen, and N. Asokan. Secure device pairing based on a visual channel (short paper). In *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 306–313, Washington, DC, USA, 2006. IEEE Computer Society.
18. Simple Pairing Whitepaper. Bluetooth Special Interest Group. [Http://www.bluetooth.com/Bluetooth/Apply/Technology/Research/Simple_Pairing.htm](http://www.bluetooth.com/Bluetooth/Apply/Technology/Research/Simple_Pairing.htm), 2006.
19. Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*, pages 172–194, 1999.
20. Ersin Uzun, Kristiina Karvonen, and N. Asokan. Usability analysis of secure pairing methods. Technical Report NRC-TR-2007-xyz, Nokia Research Center, 2007.
21. Serge Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 309 – 326. Springer, 2005.
22. Mario Čagalj, Srdjan Čapkun, and Jean-Pierre Hubaux. Key agreement in peer-to-peer wireless networks. In *Proceedings of the IEEE (Special Issue on Cryptography and Security)*, volume 94, pages 467–478, 2006.
23. WiFi Alliance. Wi-Fi Protected Setup Specification. Wi-Fi Alliance Document, January 2007.
24. Wireless USB Specification. Association Models Supplement. Revision 1.0. USB Implementers Forum. [Http://www.usb.org/developers/wusb/](http://www.usb.org/developers/wusb/), 2006.