MAC Layer Key Hierarchies and Establishment Procedures

Jukka Valkonen
Helsinki University of Technology and
Nokia Research Center
jukka.valkonen@tkk.fi

Abstract

To achieve confidential communication channel between devices they must share a security association that defines the keys and algorithms used to secure the communication. In this paper, methods to create such security associations based on a pre-shared key used in WLAN and WiMedia's UWB standards are discussed. The standards provide key agreement algorithms that operate on MAC layer based on a secret key shared by the devices. With the protocols, the devices are able to derive temporal keys for securing communication sessions. In addition to generating pair-wise keys, the standards also provide ways to distribute keys to secure group communications. In this paper, the key hierarchies and key negotiation principles are discussed. Also key generation principles on upper layers are shortly discussed.

KEYWORDS: security association, pre-shared key, key negotiation, WLAN, UWB

1 Introduction

The cryptographic keys and algorithms used to secure communications between devices are defined in security associations. To achieve a secure communication channel between two or multiple devices, these communicating devices must first form the security association to specify the needed parameters.

Different types of security associations can be negotiated by the devices on different layers of the protocol stack. In this paper, the methods used in WiMedia's UWB and IEEE's WLAN standards to negotiate security association in the MAC layer are discussed. In addition, some properties of combining the methods used in MAC layer and in the upper layers are discussed.

The security associations can be either short term or long lived. In the standards discussed in this paper, the basic idea is that the devices share a long term security association from which the short term associations are derived when communication channel is needed.

Generating session keys from a shared master key reduces the problem of creating keys for each session to creating the master key. Recently several methods to create authenticated shared secret between a pair of devices have been proposed. In the scope of this paper, these protocols are omitted, and the main target is to discuss MAC layer key negotiation methods using a shared master key.

In addition to providing confidentiality to communications between a pair of devices there is need for transmitting data securely for a group of devices. In such occasions, it is needed for all the devices to share one common key that is used to encrypt and decrypt the data. To solve these problems, both of the standards provide a way to distribute the group keys based on the pair-wise security associations.

The main goal of this paper is to explain the principles behind the generation of pair-wise and group keys used in WLAN and UWB. The details of the protocols are omitted on purpose to give clearer view on the key hierarchies of the standards.

The rest of the paper is organized as follows. In Sections 2 and 3 the key negotiation methods and the key hierarchies for WiMedia's UWB and the upcoming WLAN standard are discussed. In Section 4, key negotiation algorithms for upper layers are discussed. Finally, conclusions are given in Section 5.

2 WiMedia

WiMedia's [WiM06] ultra wideband (UWB) is a radio platform that provides short range wireless networks with speeds up to 480 Mb/s. The standard [UWB05] discussed in this paper is published by Ecma International [Ecm06]. The standard provides physical layer and medium access control specifications for UWB. Recently, for example, Wireless USB [Wir06] has adopted the UWB radio platform to be the transmission medium.

The standard describes 3 different security modes for the devices. In security mode 0, a device uses only non-secure frames to communicate with other devices. If a device is in security mode 1, it uses non-secure frames to communicate with devices in security mode 0 and with devices it does not have a secure communication channel. In security mode 2, the devices communicate only using secure frames. A device in security mode 2 never creates a secure communication channel with devices in security modes 0 or 1. To create a secure channel between two devices in security mode 2, they use the method described in Section 2.1.

The standard uses two kind of encryption keys. Encrypted channels can be built between a pair of devices or between a group of devices. If the communication is between two devices, the keys are negotiated using the handshake described in the following section. The key exchange is authenticated using a shared master key installed by some means in the devices prior to execution of the protocol. The group keys are created and distributed by the devices using the pair-wise security associations.

In the following two sections, methods to establish pair-wise and group keys are described.

2.1 Pair-wise keys

The key negotiation is performed using a 4-way handshake directly between the devices. In the handshake, a shared master key (PMK) is used to authenticate the entities to each other. During the handshake protocol, the devices derive a pair-wise temporal key (PTK) from the master key and random nonces. This PTK is then used to protect frames delivered between the devices. Neither the master key or the PTK is ever transmitted between the devices, encrypted or not. How the devices get the shared master key, is out of the scope of the standard. The procedure, where the acting devices are called initiator and responder, is as follows.

The first message sent by the initiator consists of the identifier MKID of the master key, a proposed identifier TKID of the pair-wise key to be derived and freshly generated 128-bit random value, called I-Nonce. The TKID must be unique at the moment; there can not be another pair-wise key with the same identifier nor is it possible to have an ongoing handshake with some other device with the same identifier.

Upon receiving the first message, the responder extracts the values and checks if the TKID proposed by the initiator is unique at that time. In case of a positive answer, the responder extracts the I-Nonce from the message and generates a 128-bit fresh random nonce called R-Nonce. At this point, the receiver is able to compute the PTK using the shared material. In addition to deriving the PTK, the responder also generates so called Key Confirmation Key (KCK). The keys are generated using the pair-wise master key, the device addresses of the communicating devices, the identifier of the PTK and the random nonces the devices have generated. These values are given as an input for a pseudo-random function. First 16 octets of the output of the function are interpreted as the KCK and the next 32 octets as the PTK. After the responder has generated the keys, the responder generates the second message of the protocol. This message contains a specific status code, R-Nonce and a message integrity code (MIC) calculated from the message using the key confirmation key. This message is then sent to the initiator.

When initiator receives the second message, it extracts the R-Nonce and computes both, PTK and KCK using the same method as described previously. Next, the initiator recalculates the integrity code (MIC) of the message using the keys the initiator just generated. If the recalculated integrity code does not match the code sent by the responder in the second message, the initiator shall discard the message and abort the handshake. Otherwise, if the integrity codes are equal, the initiator can be sure that the responder shares the same master key and the procedure can be continued. The initiator also checks the status code and aborts if the code indicates so. In addition, the status code sent by the responder can also indicate a collision in the identifier the initiator had suggested. In this case the handshake is restarted using a different TKID. If the status code indicates a normal status, the procedure is continued by the initiator who sends a message containing the same I-Nonce as in the first message and the integrity code (PTK MIC) computed for the message using KCK.

1. $\mathsf{D}_I \rightarrow \mathsf{D}_R$: `MKID, TKID, I-Nonce`
2. $\mathsf{D}_I \leftarrow \mathsf{D}_R$: `Status code, R-Nonce, PTK-MIC`
3. $\mathsf{D}_I \rightarrow \mathsf{D}_R$: `I-Nonce, PTK-MIC`
4. $\mathsf{D}_I \leftarrow \mathsf{D}_R$: `R-Nonce, PTK-MIC`

Figure 1: 4-way handshake used in WiMedia

Now, after the responder has received the third message of the protocol, it extracts the PTK MIC from the message and recalculates the same code using the keys possessed by the responder. If the codes are not equal, the responder aborts the procedure. Otherwise, it installs the PTK to start using it and creates and sends the fourth message of the protocol. In this message, the responder sends the same random nonce R-Nonce as in the second message and a message integrity code (PTK MIC) for the message computed using the secret KCK.

Upon receiving the fourth message, the initiator verifies the PTK MIC by recomputing it using its own KCK. If the values do not match, then the initiator aborts the procedure. Otherwise, it installs the PTK and starts using it.

To sum the procedure up, the handshake contains four messages and three different keys. These keys are pair-wise master key (PMK), pair-wise temporal key (PTK) and key confirmation key (KCK). Of these keys, PMK is a shared secret which is given for both of the device by some means before initiating the handshake. PTK and KCK are generated while the procedure is run and used only for a short period of time. None of the keys are sent between the devices. Only material sent are the random nonces which are used to derive the keys and the message integrity codes which are used to proof the integrity of the message and the possession of the shared master key. The procedure is depicted in Figure 1.

2.2 Group keys

The WiMedia UWB standard also provides a way to exchange keys for group communication, called Group Temporal Keys (GTKs). The GTKs are 128-bit random numbers and they are, as the name suggests, short lived and thus used only for a short period of time. The group keys are used only as one way keys, that is, the sender uses the key to encrypt multicast data and the recipients use the key to decrypt, but the recipients never encrypt using that key. The group keys are transmitted between the devices using pair-wise temporal keys. Thus in order to form a multicast group, the initiator (the sender) must first negotiate pair-wise keys with each of the intended recipients. The situation is depicted in Figure 2, where three devices have formed a multicast group allowing each device to send encrypted data to every other device.

The distribution of the group keys is handled using so called GTK command frames. After the devices have successfully performed the 4-round pair-wise handshake as described in Section 2.1, they distribute the group temporal keys used to send data between themselves. The messages

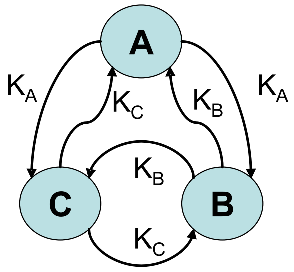

Figure 2: Keys used for group communication

are sent encrypted using the PTK.

Upon receiving GTK command frame, the device must check whether the identifier, called GTKID, of the corresponding GTK is unique. If it is, then the device acknowledges the GTK by responding with another GTK command frame with a status field indicating success. In case the GTKID is not unique, that is, there is a matching GTKID or TKID in use in the device, that device informs the sender of the situation. The sender of the GTK issues another GTKID until it receives a confirmation from the device that the identifier is unique. After the confirmation, the device distributes the new GTKID for the group temporal key to the devices possessing the same GTK.

3 WLAN

IEEE 802.11 denotes a set of standards to be used for creating wireless local area networks. Currently, the family of standards include 6 different techniques, that all use the same protocol. The protocols provide wireless networking techniques with maximum data rate of 540 Mbit/s (802.11n). [Wik06]

Currently, the IEEE 802.11 standard is under revision. In this paper, the unapproved draft version 8.0 [IEE06] of the standard is discussed.

The basic building block of an IEEE 802.11 network is call basic service set (BSS). BSS consists of a set of stations, that have successfully synchronized to communicate. A membership of a BSS does not imply that the devices in a BSS can communicate wirelessly. In the simples form, a BSS can consist only of two wireless devices.

For the devices to be able to communicate with each other, the devices can generate an ad-hoc network. This type of a network is called Independent BSS (IBSS). In such a network, the devices that need to communicate, must be close enough to each other to be able to communicate. In other words, there is no routing in an IBSS.

In case multiple basic service sets are connected, the network is called Extended Service Set (ESS). In such a network, the devices inside a BSS are connected to an access point, which is then connected to another access point using distribution system (DS). DS itself is not a part of the ESS. The devices in an ESS communicate only with the access point, not with each other.

The key hierarchy of WLAN standard consists of pairwise and group keys. A device can negotiate a shared temporal key with an access point or another station. For these negotiations, a shared master key is used to authenticate the devices. The standard also defines ways to create keys to secure multicast data. In addition, the standard defines negotiation keys that can be used between two stations communicating through an access point.

In the following subsections, the negotiation of different types of keys are discussed.

3.1 Pair wise keys

In both scenarios, that is, ESS and IBSS, the devices use both long term keys and short term temporal keys. The long term keys called pair-wise master keys (PMKs) are used to authenticate the short term keys. The negotiation of the PMKs can be handled for example using IEEE 802.1X authentication method, or the key can be some other form of pre shared key.

3.1.1 Station to Access Point

When a station needs to generate an association with an access point within an ESS, the station acts as so called supplicant and the access point as an authenticator. In addition, prior to key negotiation the access point must have generated a secure channel with an authentication server. This generation is out of the scope of the standard. The authentication server can be for example included in the access point.

The first phase of the key exchange is the negotiation of the long-term pair-wise master key (PMK). As already mentioned, this negotiation can be performed using IEEE 801.1X authentication which uses extensible authentication protocol (EAP) to negotiate a shared secret. The other possibility is to use pre-shared key as a PMK. The PMK is part of a security association called PMKSA.

After the PMK is successfully installed to the entities, they start a 4-way handshake to negotiate bidirectional pair-wise transient key association (PTKSA), which includes the pair-wise transient key (PTK), a temporal key used to encrypt traffic transmitted between two devices. The purpose of the handshake is to confirm that a live peer holds the PMK, confirm that the PMK is current, derive a fresh PTK, install the PTK, transport group temporal key (GTK) from the authenticator to supplicant and confirm the selection of the cipher suite. The handshake consists of the following steps.

In the first phase, the authenticator sends a message including random nonce, called ANonce. Upon receiving this message, the supplicant generates random nonce, called SNonce, and generates the pair-wise transient key from the random nonces and other shared data, such as the PMK. For this key generation, a pseudorandom function is used. From the PTK the device also generates three different keys, which are Key Confirmation Key (KCK), Key Encryption Key (KEK) and Temporal Key (TK) by taking certain bits of the PTK for each of the sub keys.

After generating the PTK (and the other keys from the PTK), the supplicant constructs the second message of the protocol which contains the SNonce the device just generated, a message integrity code computed using the key confirmation key, and some other data about the security association to be negotiated.

Upon receiving the second message, the authenticator is now able to extract the supplicant's random nonce SNonce

GTK	Group temporal key	GTKID	Group temporal key identifier
PTK	Pair-wise temporal key (UWB)	PMK	Pair-wise master key
MKID	Master key identifier	TKID	Temporal key identifier
MIC	Message integrity code	ESS	Extended Service Set
BSS	Basic Service Set	IBSS	Independent Service Set
PMKSA	Pairwise master key security association	DS	Distribution System
PSK	Pre-shared key	GTKSA	Group Temporal Key Security Association
GMK	Group master key	PTKSA	Pair-wise transient key security association
PTK	pair-wise transient key (WLAN)	KCK	Key confirmation key
TK	Temporal key		

Table 1: Abbreviations used

and compute first the PTK and derive the rest of the keys from the pairwise temporal key. The device is then also able to verify the integrity of the received message as it is now able to compute the same message integrity code that was included in the message.

The third message of the protocol is sent by the authenticator to the supplicant. This message includes the ANonce sent in the first message, a message integrity code (MIC), the group key and its identifier, and again some information of the negotiated association. Finally, the fourth message of the protocol, sent by the supplicant, ends the procedure. A simplified version of the protocol is depicted in Figure 3.

In case there is an error, such as non-matching integrity codes or counters with false value, during the procedure, the devices discard the flawed messages.

After successfully performing the handshake, the station and the access point can now send encrypted data using the pair-wise key between themselves. In addition, the devices can be sure that the recipient holds the same PMK and that the PMK is still valid.

1. $\mathsf{D}_A \rightarrow \mathsf{D}_S$: `ANonce, PMKID`
2. $\mathsf{D}_A \leftarrow \mathsf{D}_S$: `SNonce, MIC`
3. $\mathsf{D}_A \rightarrow \mathsf{D}_S$: `ANonce, MIC, GTK`
4. $\mathsf{D}_A \leftarrow \mathsf{D}_S$: `MIC`

Figure 3: Simplified 4-way handshake used in WLAN

3.1.2 Station to Station

In case two devices are communicating directly, that is, without an access point within an IBSS, one of the devices need to act as an authenticator and the other as a supplicant. After this point, the situation is the same as in the communication between a station and an access point.

The temporal key used to encrypt data sent between the devices is again derived using the 4-way handshake similarly as if the device was communicating with an access point. Prior to the execution of the handshake, the devices need to generate a PMKSA that includes the pair-wise master key. This can be done either using IEEE 802.1X authentication of a PSK shared some other way.

After the PMK is known for both of the participants, the 4-way handshake is initiated and performed as in the case when a device communicates with the access point described previously.

3.2 Group keys

The WLAN standard also provides a way to encrypt multicast data. For this, a group temporal key security association (GTKSA) that includes a group temporal key (GTK) is used. The security association can be built either using the pairwise 4-way handshake or a group handshake. In either case, the security association is built using the pair-wise associations as the starting point. Thus in order to create a multicast group, pair-wise associations must first be negotiated.

The group temporal key security association is unidirectional. Thus if bidirectional channel need to be created, two different associations must be created. In an ESS, the only station performing broadcasts is the access point, as the associations are always built between a station and the access point. Thus there is no need for bidirectional group security association; it is only needed for the access point to have a key to encrypt the broadcast data and the stations receiving the data to have the decryption key. The stations communicate only with the access point using the pair-wise keys. In an IBSS, the devices need to have encryption keys for each multicast groups the need to send data, and to have the decryption keys of each station that sends data to them.

If the group key is created using the 4-way handshake, the key is created by the authenticator and sent encrypted in the third message of the protocol (3 in Figure 3) to the supplicant, which then uses the key to encrypt data received from the authenticator. In addition, the authenticator provides the supplicant the identifier of the group key. The GTK is generated by the authenticator from group master key (GMK) installed in the authenticator using freshly generated random number GNonce, fixed string and the identifier of the authenticator. The key is taken as the output of a pseudorandom function that takes the values as input. The length of the key depends on the encryption method used. Actual encryption and decryption keys are derived from the GTK by taking the needed amount of bits from the beginning of the GTK.

The authenticator is also allowed to change an existing GTK. For this, group key handshake is used. This handshake consists of two messages. The first message is sent by the authenticator to the supplicant and it contains the new GTK with its identifier, an integrity check code and some other information of the derived key. The second message is sent by the supplicant and it is used to acknowledge the new GTK. After successfully exchanging the keys, the devices take the new group key into use.

4 Key Negotiation in Upper Layers

The Sections 2 and 3 discuss the key negotiation in the MAC layer. Both of the standards use pair-wise pre-shared keys to negotiate security associations between the devices. These associations are then used by the sender to transmit the group keys to encrypt data to the receivers of the multicast group.

In the standards discussed in this paper, creation of the pre-shared key is omitted. The WLAN standard provides a way to negotiate a key using IEEE 802.1X authentication protocol, but leaves also the possibility to use a pre-shared key negotiated using some other way.

This negotiation of the shared (and authenticated) secret can be performed on the upper layers of the protocol stack using methods such as MANA protocols [GMN04], Bluetooth Simple Pairing [Blu06] or visual channels [MPR05, SEKA06]. With such protocols, the devices are able to negotiate a shared master key between themselves using for example Diffie-Hellman key exchange [DH76] and authenticate the key exchange using auxiliary out-of-band channels such as the user comparing or entering short strings or taking snapshots using a camera phone. This key negotiation can be performed without encryption on the MAC layer as the properties of the protocols prevent an attacker to successfully eavesdropping or intervening the protocols. This kind of pair-wise key negotiation makes the procedure quite cumbersome for the users to handle when ad-hoc group associations where devices communicate directly need to be built as the users need to perform the authentication to each of the associations separately.

Valkonen, et al., describe in [VAN06] methods to create an authenticated shared secret between a group of devices without using pair-wise associations as the basis. The protocols use Diffie-Hellman key exchange modified for a group of devices to negotiate a shared secret, and authenticate it using methods based either on a passkey or numeric comparison. Other protocols that can be used to negotiate authenticated group keys can be found for example from [DB06, ABCP06].

The advantage of such a group protocol is the ability to negotiate group keys without using pair-wise associations thus making the procedure more straightforward for the users. In scope of WLAN or UWB standards, such a protocol could be used to create the shared secret between the devices used to authenticate the pair-wise associations on the MAC layer. In such a context, the procedure requires less actions from the users, as the users does not have to deal with the pair-wise associations.

Naturally if such a group key negotiation protocol is used to create the shared secret, then all of the devices participating in the multicast group share the same key. This becomes a problem, if a device needs to be expelled from the group. In such occasions, it is of course possible for the devices to use the pair-wise associations to redistribute the new GTK for the devices. This method is still unsecure; it is enough for the expelled device to have recorded one association between two other devices to know the temporal key they are using and thus to be able to find out the new group key. To avoid this, there should be some way for the upper layer protocol to redistribute the new PMKs to the remaining devices for them to recreate the pair-wise MAC layer associations and redistribute the new MAC layer group keys.

Still, some advantage can be gained if all of the devices in the group share the same PMK. First of all, in such occasions the master key identifies the group thus making it impossible for an attacker to trick the devices to sending data to some groups they don't know they belong. Also, after distributing the master key to all of the devices, the devices in the group do not have to rely on the associations generated on the upper layer. This makes the group more modular as all the devices sharing the same master key can now create temporal associations on the MAC layer and distributing broadcast keys to other devices.

5 Conclusions

In this paper, the key hierarchies and generation methods used in WLAN and UWB were discussed. Both of the standards provide ways to derive session keys from shared long term key provided by some means to the protocols. In addition, the standards also provide ways to secure multicast data by specifying methods to derive and distribute unidirectional group keys.

Both of the standards discussed in this paper use shared long secret keys, from which the session keys are derived. These shared secret keys are long term keys, that is, they are stored on the devices for a long period of time. The temporal, or transient, keys are used only to encrypt shorter communication sessions and thus new transient keys are derived from the master key when new sessions are needed.

It should be noted, that with the protocols used in the standards, Perfect Forward Secrecy is not achieved. If the pair-wise master key becomes available for an attacker, it can derive all the temporal keys created from the master key if the attacker has all the negotiations recorded.

The group keys are created and distributed using the pair-wise session keys. In WLAN, the group keys are distributed within the handshake that is used to create pair-wise security associations, or by using group key handshake. In UWB, the group keys are distributed separately from the pair-wise handshake. In both of the standards, these group keys are unidirectional and short term.

All in all, the constructions used in both of the standards are quite similar as both of them use master keys to derive session keys. Also, as these standards have adopted the method, it seems to be quite reasonable method to create session keys.

References

[ABCP06] Michael Abdalla, Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Password-based Group Key Exchange in a Constant Number of Rounds. In *Public Key Cryptography - PKC 2006*, 2006. LNCS 3958.

[Blu06] Bluetooth SIG. Simple Pairing Whitepaper. Technical report, Bluetooth SIG, 2006. `http://www.bluetooth.com/Bluetooth/Apply/Technology/Research/Simple_Pai%ring.htm`.

[DB06] Ratna Dutta and Rana Barua. Password-Based Encrypted Group Key Agreement. *International Journal of Network Security*, 3(1):23–34, 2006.

[DH76] Whitfield Diffie and Martin E. Hellman. New Directions In Cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, 1976.

[Ecm06] Ecma International, 2006. `http://www.ecma-international.org`.

[GMN04] Christian Gehrmann, Chris J. Mitchell, and Kaisa Nyberg. Manual Authentication for Wireless Devices. *RSA Cryptobytes*, 7(1), 2004.

[IEE06] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Unapproved Draft Standard v8.0, 2006.

[MPR05] Jonathan M. McCune, Adrian Perrig, and Michael K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2005.

[SEKA06] Nitesh Saxena, Jan-Erik Ekberg, Kari Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. Cryptology ePrint Archive, Report 2006/050, 2006. `http://eprint.iacr.org/`.

[UWB05] A Standard ECMA-368, High Rate Ultra Wideband PHY and MAC Standard, 2005. `http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-368.p%df`.

[VAN06] Jukka Valkonen, N. Asokan, and Kaisa Nyberg. Ad Hoc Security Associations for Groups. In *Proceedings of ESAS 2006 (to appear)*, 2006. LNCS 4357.

[Wik06] Wikipedia: IEEE 802.11, 2006. `http://en.wikipedia.org/wiki/IEEE_802.11`.

[WiM06] WiMedia Alliance, 2006. `http://www.wimedia.org`.

[Wir06] Certified Wireless USB from the USB-IF, 2006. `http://www.usb.org/developers/wusb`.