Key Management in Ad-Hoc Networks

Jukka Valkonen

Helsinki University of Technology
Laboratory for Theoretical Computes Science
jukka.valkonen@tkk.fi

Abstract. Key management is crucial part of security in communications. In wireless ad-hoc network, the issue is even bigger. As there is no infrastructure in the network, the distribution of encryption keys in an authenticated manner is a difficult task. In this paper, key management aspects are discussed by first describing different key management approaches followed by some ways to authenticate key exchange protocols. Finally short overview of keys in Wireless LAN is given.

1 Introduction

The wireless nature of ad-hoc networks enables easy eavesdropping of communications for adversaries. Thus it is crucial for the devices to share encryption keys to enable encryption of all data transmitted through the wireless medium. As ad-hoc networks lack all kind of an infrastructure, sharing these encryption keys is not an easy task. The devices need to be sure that they are actually communicating with the intended recipient and not some other device masquerading as a legitimate one.

The lack of infrastructure is not the only difficulty for key generation in ad-hoc networks. In many cases, the devices are very constrained, both in computational capacity and input/output capabilities.

In this paper, some key management aspects of ad-hoc networks are described. This paper concentrates on basis of authenticated key agreement, and the limitations of computationally constrained devices has been left away. See [Ekb06] for discussion on constrained devices.

The paper is organized as follows. Section 2 discusses different approaches to key management. In Section 3 some protocols for authenticated key agreement are discussed. As an example of a real world system, Section 4 discusses key management in Wireless LAN. In Section 5 conclusions are given.

2 Key Management Approaches

Different approaches for key management are listed in [MM04]. These are key predistribution, key transport, key arbitration and key agreement. These approaches are discussed next.

2.1 Key Predistribution

In key predistribution the keys are distributed to the devices participating in the communication beforehand. The system is inflexible, all the devices sharing the key must be known when the system is initiated.

2.2 Key Transport

In key transport methods, the key is transmitted when needed. The transport is initiated by one device; a device generates a key that is then transmitted to the recipients. This transmission is usually encrypted using some prior shared key encryption key (KEK). In such methods, some way for the devices to know the KEK is needed. With public key infrastructure (PKI), the key transmission could be done using the public keys of recipients. However, in ad-hoc networks, PKI can not be held as a requirement.

One possible method for key transport utilizing key encryption keys given in [MM04] is called Shamir's three-pass protocol, illustrated in Figure 1. The protocol is based on invertible functions f and g that commute, that is $f(g(z)) = g(f(z))$.

1. D_1 generates random key K and encrypts it using f with random key x and sends the value to D_2
   $\mathsf{D}_1 \rightarrow \mathsf{D}_2$: $f_x(K)$
2. D_2 encrypts the received message using g and a random key y and sends the value to D_1
   $\mathsf{D}_1 \leftarrow \mathsf{D}_2$: $g_y(f_x(K))$
3. D_1 decrypts the received value using f^{-1} and x and sends the value to D_2
   $\mathsf{D}_1 \rightarrow \mathsf{D}_2$: $f^{-1}(g_y(f_x(K))) = f_x^{-1}(f_x(g_y(K))) = g_y(K)$
4. D_2 decrypts the received value using g^{-1} and y.

Fig. 1. Shamir's three-pass protocol

Recently, standardization organizations have adopted key transport as one possible way to transmit keys. Both Bluetooth Simple Pairing [Blu06] and Wi-Fi Protected Setup [Wi-07] can use some out-of-band channels to transmit network credentials between the devices. The out-of-band channel can be implemented for example using Near Field Communication (NFC) or USB Flash drives (UFD).

2.3 Key Arbitration

In key arbitration, one device in the network is used to create and transmit keys to devices. Key arbitration methods are not very suitable for ad-hoc networks, as one central node is needed to be accessible for all the other nodes all the time. In a network with infrastructure, the access point acts usually as the arbitrator.

2.4 Key Agreement

In key agreement protocols, some asymmetric key algorithm is used to negotiate the key. Probably the most common algorithm is called Diffie-Hellman key agreement protocol [DH76]. The protocol provides an unauthenticated key agreement; a passive attacker is not able to derive the shared secret, but an active attacker is able to masquerade as a man-in-the-middle an participate in the key agreement without the legitimate devices noticing. The protocol is depicted in Figure 2.

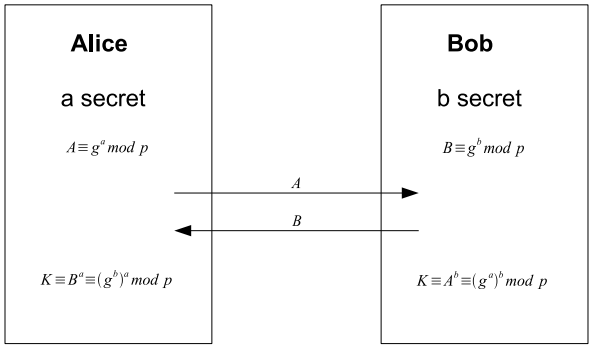

Fig. 2. Diffie-Hellman Key Exchange

3 Authenticated Key Management

3.1 Methods Based on a Passkey

One possibility to authenticate key agreement is to use a shared passkey. Such methods are proposed for example in [GMN04,BM92]. The basic principle behind the protocols is that the attacker is not able to masquerade as a legitimate device as it does not know the passkey. A version of a MANA III protocol described in [GMN04] is depicted in Figure 3, where PK_1 and PK_2 are the public keys of the devices. The protocol needs to run at least twice with different passkeys P_i; first to authenticate the values and then to verify the authentication. For example, Bluetooth Simple Pairing [Blu06] uses 20 rounds. Other possibility is to use the human as the verifier: after the protocol has run, the user indicates the success or failure to both devices.

A natural extension for a pairwise protocol is to create a shared key between a group of devices. Asokan and Ginzboorg present in [AG00] group key agreement methods based on Encrypted Key Exchange (EKE) -protocol proposed in [BM92]. The protocol is depicted in Figure 4, where R_i denotes a random value selected by ith device, $P(data)$ denotes encryption using the shared secret, $K(data)$ denotes encryption using the negotiated shared key K and $H(data)$ is a one-way hash function. As a result of the protocol, all n devices share the same key $g^{R_1 R_2 \ldots R_n} \mod p$.

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 $\hat{h}_{i1} \stackrel{?}{=} h(1, P\hat{K}_1, PK_2, P_i, \hat{R}_{i1})$
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

1. $\mathsf{D}_i \rightarrow \mathsf{D}_{i+1} : g^{R_1 R_2 \ldots R_i} \mod p$, $i = 1, \ldots, n-2$
2. $\mathsf{D}_{n-1} \rightarrow$ ALL: $\pi = g^{R_1 R_2 \ldots R_{n-1}} \mod p$
3. $\mathsf{D}_i \rightarrow \mathsf{D}_n$: $P(c_i)$, $i = 1, \ldots, n-1$, where $c_i = \pi^{\frac{\tilde{R}_i}{R_i}}$ and $\tilde{R}_i$ is a fresh random number generated by D_i
4. $\mathsf{D}_n \rightarrow \mathsf{D}_i$: $c_i^{R_n}$, $i = 1, \ldots, n-1$
5. $\mathsf{D}_i \rightarrow$ ALL: D_i, $K(\mathsf{D}_i, H(\mathsf{D}_1, \mathsf{D}_2, \ldots, \mathsf{D}_n))$ for some i

Fig. 4. EKE-based Group Diffie-Hellman key agreement method

3.2 Methods Based on a Verification String

If the devices have more restricted input capabilities but good displays, the key negotiation can be authenticated using so called short authenticated strings, as proposed for example by [ČČH06] and [LAN05]. In these methods, the devices compute a short verification string from the negotiated material. This string is displayed to the user, whose task it is to compare the strings. In case the strings are equal, the devices share the same key and thus the association was successful. Otherwise, the devices have negotiated a shared key with an attacker or some other device, and thus the association failed. The basic three-round protocol from [LAN05] is depicted in Figure 5.

It should be noted, that protocols based on a verification string are inherently less secure than methods based on a passkey, as the user is able to do fatal errors by signaling false acceptance to the devices. With passkey based protocols, this is not the case, as if the user enters wrong passkey, the key agreement algorithm fails.

As was the case with the with passkey based protocols, also numeric comparison protocols can be extended to handle group key negotiation, as is described in [VAN06]. The protocol utilizes Diffie-Hellman key agreement protocol extended to multi-party case. As a result of the protocol, all devices compute a short verification string from the keying material and random values. Each of the devices

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. 5. Authentication by Short Integrity Checksum

display the string for the users to compare. If and only if all the devices display the same string, the users acknowledge the procedure for all of the devices.

3.3 Threshold Cryptography

Contradicting the properties of ad-hoc networks, the authors of [MM04] describe also usage of public key infrastructure as a possible way for key agreement. In such methods, the network includes certifying authorities (CAs) that binds keys to specific nodes by means of certificates.

In the method described in [MM04] the networks includes n servers providing the certificates, out of which $t + 1$ are needed for creation of the certificate but t is not enough. The system requires that $n \geq 3 \cdot t + 1$. When signing a public key, the servers create partial signatures and send them to combiner, which then generates the signature using $t + 1$ partial signatures and verifies the result. If the verification fails, another subset of partial signatures is tried.

3.4 Self-Organized Key Management

In [MM04] describe also self-organized public-key management originally proposed by Čapkun, et al., in [ČBH03]. The system does not need any kind of infrastructure to authenticate keys. Their method is based on the users issuing certificates to each other based on personal acquaintance. These certificates are used to bind a public key and a node. The devices store in a local repository all certificates the other devices have issued to the device and the certificates the devices has issued to other devices. The devices periodically exchange information about these repositories by requesting new certificates from neighbors. If a conflict is found, some method for resolving such conflicts is used.

The devices are organized into a certificate graph, where vertices are public keys of nodes edges are certificates issued by the users. When a device wishes to obtain a public key of some other device, it finds a chain of certificates between themselves.

4 Case: WLAN

The key management approaches described previously in this paper are used on the upper layers of the protocol stack. In addition to these keys, lower level keys are used to actually encrypt transmitted data. To give an insight about what actually happens after the upper layer keys have been negotiated, this section briefly discusses key hierarchies of Wireless LAN [IEE06,IEE].

After the devices have negotiated a pair-wise master key using some method described previously, they install it into MAC-layer. This key is then used to derive pair-wise transient keys and group temporal keys (GTK). The pair-wise transient key (PTK) is negotiated using four way handshake depicted in Figure 6, where ANonce and SNonce are random nonces used to guarantee freshness, PMKID is identifier for pair-wise master key negotiated, MICs stand for message integrity codes and KCK (key confirmation key) and KEK (key encryption key) are subkeys derived from PTK.

As a result of the handshake, the devices share pair-wise encryption keys, key confirmation keys and key encryption keys used to secure pair-wise communications.

1. $\mathsf{D}_A \rightarrow \mathsf{D}_S$: `ANonce, PMKID`
2. $\mathsf{D}_A \leftarrow \mathsf{D}_S$: `SNonce,` $\mathtt{MIC}_{\mathtt{KCK}}$
3. $\mathsf{D}_A \rightarrow \mathsf{D}_S$: `ANonce,` $\mathtt{MIC}_{\mathtt{KCK}}$`,` $\mathtt{E}_{\mathtt{KEK}}$`(GTK)`
4. $\mathsf{D}_A \leftarrow \mathsf{D}_S$: $\mathtt{MIC}_{\mathtt{KCK}}$

Fig. 6. Simplified 4-way handshake used in WLAN

During the four way handshake, the devices also transmit group temporal keys encrypted using the output of the handshake. This group temporal key is sender specific; the sender negotiates a 4-way handshake with all the intended recipients and transmits the key to these. Only the sender uses this key to encrypt data, the recipients use the key only to decrypt. This kind of construction works well when the network is constructed in infrastructure mode, where all devices are attached to base station performing all the multicasts. In ad-hoc mode, the situation is different, as in order to generate a multicast group where all devices need to transmit to all devices, they all need first to generate pair-wise keys on the upper level, run the MAC-layer handshake and transmit the group key to other devices. See [NV07] for more thorough discussion on multicast aspects on MAC-layer.

5 Conclusions

Key management in ad-hoc networks is a difficult problem. In this paper, some aspects of key management approaches based on [MM04] have been discussed.

The classification described in Section 2 is just one possible, see [SVA07] for a different kind of classification.

From the key management point of view, it is not enough to create a method that can be used to create a shared secret between the devices, the underlying communications protocol must also be taken into consideration.

As it can be seen, there is no single method for key management that can be used in all cases. If some way works in some kind of network, it might not work with another kind.

References

[AG00] N. Asokan and Philip Ginzboorg. Key Agreement in Ad-hoc Networks. *Computer Communications Review*, 23(17):1627–1637, November 2000.

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

[BM92] Steven M. Bellovin and Michael Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In *1992 IEEE Computer Society Symposium*, pages 72–84, 1992.

[ČBH03] Srdjan Čapkun, Levente Buttyan, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, 2003.

[ČČH06] Mario Čagalj, Srdjan Čapkun, and Jean-Pierre Hubaux. Key Agreement in Peer-to-Peer Wireless Networks. *Proceedings of the IEEE (Special Issue on Security and Cryptography)*, 92(2):467–478, February 2006.

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

[Ekb06] Jan-Erik Ekberg. Key establishment in constrained devices. `http://www.tcs.hut.fi/Studies/T-79.7001/2006AUT/seminar-papers/Ekberg-paper-final.pdf`, 2006.

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

[IEE] IEEE 802.11TM WIRELESS LOCAL AREA NETWORKS - The Working Group for WLAN Standards. `http://grouper.ieee.org/groups/802/11/`.

[IEE06] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specif ications. Unpublished Draft v8.0, 2006.

[LAN05] 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/`.

[MM04] C. Siva Ram Murthy and B. S. Manoj. *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall, 2004.

[NV07] Kaisa Nyberg and Jukka Valkonen. Wireless Group Security Using MAC Layer Multicast, 2007. Accepted to WoWMoM 2007.

[SVA07] Jani Suomalainen, Jukka Valkonen, and N. Asokan. Security Associations in Personal Networks: A Comparative Survey. Technical Report NRC-TR-2007-004, Nokia Research Center and VTT Technical Research Centre of Finland, 2007. `http://research.nokia.com/tr/NRC-TR-2007-004.pdf`.

[VAN06] Jukka Valkonen, N. Asokan, and Kaisa Nyberg. Ad Hoc Security Associations for Groups. Accepted to ESAS 2006, 2006.

[Wi-07] Wi-Fi Alliance. Wi-Fi Protected Setup Specification. Wi-Fi Alliance Document, January 2007.