International Journal of Principles and Applications of Information Science and Technology
July 2008, Vol.2, No.1

A Secure 4-Way Handshake in 802.11i Using Cookies

Sung-Hyun Eum, Yae-Hoe Kim, and Hyoung-Kee Choi

Information and Communication Engineering, Sungkyunkwan University,
Jangan-gu, Suwon, 440-746, Korea
{sheum; weawen; hkchoi}@ece.skku.ac.kr

Abstract. The phenomenal popularity of the 802.11 network standards stems from the fact that they provide for wireless connections with ease and convenience. Recently, security holes have been identified in the operation of 802.11 networks, and the 802.11i protocol has been announced to protect such networks. However, there are still security issues that prevent the 802.11 network from becoming the best choice protocol for wireless LANs. We reviewed 802.11i security with a focus on a denial-of-service attack. This attack exhausts the client's memory using a vulnerability of the key derivation procedure in 802.11i. In this paper, a cookie-based mechanism is adopted in our solution to deriving the key securely. We define two classes of solution for the key derivation in 802.11i; encryption cookie and hash cookie. Both encryption cookie and hash cookie mechanisms can mitigate denial-of-service attacks in 802.11i networks.

Keywords : 4-way handshake, DoS attack, cookie, 802.11i

1 Introduction

The popularity of 802.11 networks stems from the fact that they provide for wireless connections to the Internet with ease and convenience. According to a report by the Gartner group, the wireless local area network (WLAN) market was expected to grow by twenty-two percent between 2004 and 2007 [1]. The WLAN has the advantages of mobility and a low installation cost, and it gives reasonable promise of sustainable growth.

WLANs are popular, but they are also exposed to inevitable attacks. Due to the nature of wireless communication, sensitive data is vulnerable to exposure to a third party. If the WLAN only provides confidentiality, but has weak authentication, an attacker can impersonate a user or an access point (AP) to eavesdrop on a user's secure communication. For this reason, the WLAN must provide both confidentiality and strong authentication.

The IEEE has published the 802.1x standards to provide secure communication in WLANs [2]. However, personal privacy was not fully secured by the 802.1x

protocol and its vulnerabilities to man-in-the-middle and Denial-of-Service (DoS) attacks have been identified. The IEEE has published the 802.11i protocol to mitigate these attacks. 802.1x is augmented by 802.11i for authentication and key distribution. The 802.11i protocol is well designed for the security of WLANs. However, the 802.11i protocol is still vulnerable to DoS attack in the key derivation procedure.

This paper reviews 802.11i security with a focus on the key derivation procedure. Because there is a vulnerability in this procedure, an attacker may be able to exhaust the client's memory. To mitigate these attacks, we propose two solutions using cookie-based mechanisms. Using our proposal, key derivation in 802.11i can be made more secure. The difference between the two solutions is only the assumption and the time required.

The remainder of this paper is organized as follows: In section 2, we describe key derivation in 802.11i. Section 3 discusses the vulnerability of this key derivation procedure. Section 4 proposes several possible solutions and compares their effectiveness. Finally, section 5 concludes the paper and provides some future directions.

2 Description of the 4-way handshake

The 802.11i key derivation procedure is based on a 4-way handshake. The primary activities in a 4-way handshake are to verify the existence of the same Pairwise Master Key (*PMK*) between the client and the AP and to derive the Pairwise Transient Key (*PTK*). The 4-way handshake consists of four messages, from Message-1 to Message-4. The 4-way handshake is not the only way to implement this process. For example Altunbasak and Owen [7] suggested performance improvements by reducing the number of messages and the time delay. Since the role of the last message in the 4-way handshake is only an acknowledgment, they suggested a 3-way handshake without the last message. They showed that a 2-Way Handshake is also possible, thereby simplifying the message exchange between a client and an AP. However this simplification does not mitigate the vulnerabilities described later in this paper.

As shown in Fig. 1, the 4-way handshake starts when the AP sends Message-1 to the client. Message-1 consist of three parameters; a MAC address of an AP (called *AA*), a random number chosen by an AP (called *ANonce*) and a counter to prevent a replay attack (called *SN*).

When the client receives Message-1, it generates two additional parameters: the MAC address of the client (called *SPA*) and a random number chosen by the client (called *SNonce*). Then the client derives the PTK from five parameters, i.e., *AA*, *ANonce*, *SPA*, *SNonce* and *PMK*. When Message-1 is sent, the PTK has not been derived yet between the client and the AP. This means that the first message cannot be secured by the PTK.

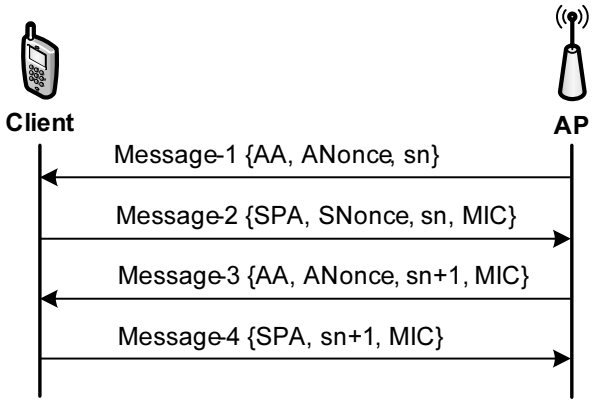

Fig. 1. The 4-way handshake in 802.11i

A client derives the PTK after receiving Message-2. Using the Message Integrity Code (*MIC*) in Message-2, the AP can verify that it has the same PTK as the one derived by the client. The MIC is a cryptographic digest used to provide the integrity of messages. Hence the MIC can be used not only to verify the integrity of the message but also to make sure that the client has the same PTK as the AP.

Message-3 is used in a similar manner to Message-2. When the AP receives the Message-3, the AP verifies that the client has the same PTK. In particular, Message-3 includes the Group Temporal Key (*GTK*). The GTK serves to protect broadcast frames in the network. Message-4 simply plays the role of an acknowledgment of Message-3.

In the 802.11i protocol, the 4-way handshake is indispensable for authentication and the derivation of a session key between a client and an AP. Regardless of whether the PMK is based on a Pre-shared Key (*PSK*), or reused from a cached PMK, a four-way handshake protocol must be able to ensure the freshness of the key. Therefore, a DoS attack to the 4-way handshake can be a fatal attack on the entire process of 802.11i authentication. A more detailed explanation of this phase can be found in Section 3.

3 Security analysis of the 4-way handshake

In order to enhance security in wireless LANs, the IEEE 802.11i standard defines a mutual authentication and session key distribution mechanism. He and Mitchell [2],[3] analyzed DoS attacks against the 802.11i protocol. This attack can occur during the 4-way handshake in the course of 802.11i authentication. In the 4-way handshake, a client and an access point exchange parameters to derive a temporary session key. The first two messages in the 4-way handshake are not protected by

the 802.11i protocol because the session key is not derived until the two messages are exchanged. The absence of the key allows a DoS attack to take place. An attacker may transmit a number of first messages, thereby deceiving a client into receiving the messages as if they came from legitimate APs [4].

In a 4-way handshake the client must accept the first message even if many first messages are received, because when the client receives the first message it cannot distinguish legitimate messages from forged messages. The client only checks for duplicates of the first message. In contrast, the client is able to accept the third message selectively because the client can verify the message using the MIC. It is quite difficult to manipulate the MIC without knowing the PTK.

The vulnerability in the 4-way handshake attack is attributed to the fact that the first message of the handshake is not secured by the session key because the session is not available at this time. An attacker can generate a number of first messages with different sequence numbers. The attack involves forging initial messages from the authenticator to the supplicant to produce inconsistent keys in peers. Therefore, the client is prone to a DoS attack; that is, generating a number of first messages (as shown in Fig. 2) and having the client keep each of the first messages in memory. This causes memory exhaustion in the client (called 4-way handshake blocking [4]). Because the attacker is capable of impersonating the authenticator composing a Message-1, and sending it to the user, there is a simple one-message attack that causes PTK inconsistency. This approach is a relatively easy attack and causes serious problems, because if the attack is successful, all previous authentication processes will be cancelled.

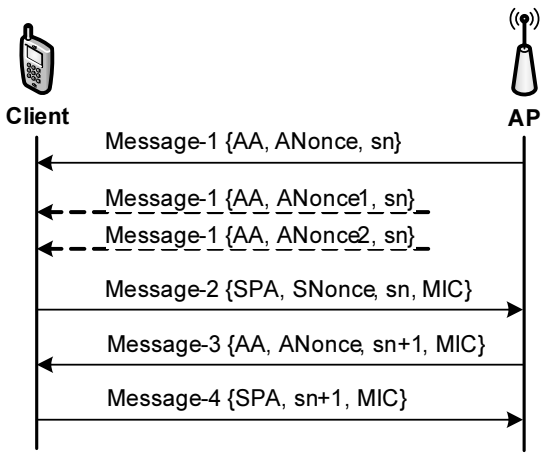

Fig. 2. 4-way handshake blocking

The attacker can cause problems for the user side by sending out more forged messages with different Nonces rather than only one. Therefore, in order to assure the handshake is non-blocking with the legitimate authenticator, the supplicant

must use sufficient memory to store all the received Nonces and the derived PTKs, until it finishes a handshake and obtains a legitimate PTK.

A memory exhaustion attack always potentially exists because the number of Message-1s can theoretically be unbounded. Though this memory exhaustion attack occurs on the supplicant side, which is not as severe as if it were the server, this is still a problem because it is quite easy for the attacker to forge and flood Message-1s.

Several solutions have been proposed in order to address 4-way handshake blocking [4]. First, with a more significant change a MIC calculated from the PMK can be added to Message-1 to prevent the attacker from forging Message-1. The authenticated Message-1 is still vulnerable to replay attacks since the PMK is static for a relatively long time. Second, the user can re-use SNonce until a legitimate handshake is completed. This approach may replace memory consumption with the consumption of computation power. Since clients increasingly have sufficient computation power, this can mitigate the attack. However, this solution has a potential problem by reusing the SNonce to provide key freshness in the authentication procedure.

4 Our proposed mechanism

In the research described in this paper, we adopted the cookie mechanism [5] to the 4-way handshake to mitigate memory depletion in the client. We defined two classes of approach for key derivation in 802.11i: Encryption cookie and hash cookie.

4.1 Encryption cookie

As we discussed in Section 3, the 4-way handshake uses the fact that the client stores both *ANonce* and a corresponding *PTK*. Hence we are able to mitigate this attack by eliminating stored parameters in the client. Fig. 3 depicts the message flows of our proposal. A more detailed explanation of our proposed mechanism is as follows.

In the beginning, the AP sends Message-1 to the client. Message-1 consists of three parameters: *AA*, *ANonce* and *SN*. When the client receives Message-1, it generates two additional parameters: *SPA* and *SNonce*. Then the client derives the PTK from five parameters, i.e., *AA*, *ANonce*, *SPA*, SNonce and *PMK*. In our proposal, the client does not store *ANonce* and *PTK*, but makes these parameters into a cookie and sends it to the AP. Because we make the cookie by encryption of some parameters, the client needs a secret key only used by the client itself. An AP after receiving Message-2 also derives the PTK. Using the MIC in Message-2, the AP can verify that it has the same PTK as the one derived by

the client. The cookie, that is the encrypted PTK, is sent back to the client. When the client receives the cookie, it decrypts the cookie using its secret key and has the same PTK as previously derived. In addition, the client can verify the MIC.

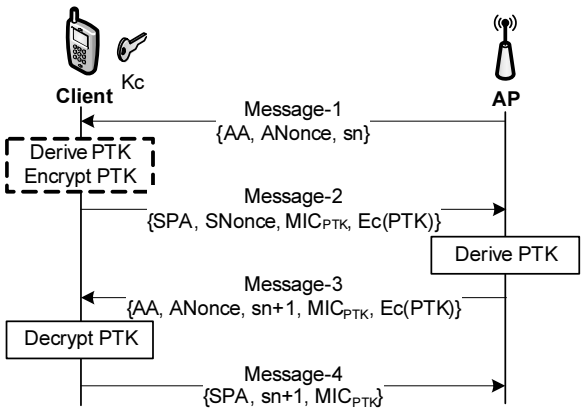

Fig. 3. Encryption cookie

Our proposal does not store both ANonce and PTK, thus an attacker may not cause a DoS attack using memory depletion in the client. However, this proposal needs additional computation power for encryption and decryption. Moreover, it needs the assumption that a client has a secret key, but an encryption using a symmetric key consumes little computation power. In addition, the Key (Kc as shown in Fig. 3) is used by only the client, thus it is easy to install the key in the client.

4.2 Hash Cookie

This approach, using the hash cookie, is similar to the encryption cookie approach. It does not transmit the PTK, but transmits the hash value of the PTK. Fig. 4 depicts the message flow of our approach. The client receives Message-1, generates the PTK and computes the hash value. The hash value is transmitted as a cookie in Message-2. The AP receives the cookie and sends back it to the client. When the client gets the cookie, it can verify that the cookie has the same value. In this approach, the client does not store SNonce and PTK, but it has to compute PTK again. Moreover, Message-3 needs an additional SNonce, because when the client receives the message, it cannot know SNonce.

This approach has the advantage of the assumption, which needs no key or encryption. In general, hash computation using this approach is more lightweight than encryption, but it has to derive the PTK again. We discuss the efficiency of these approaches in the following section.

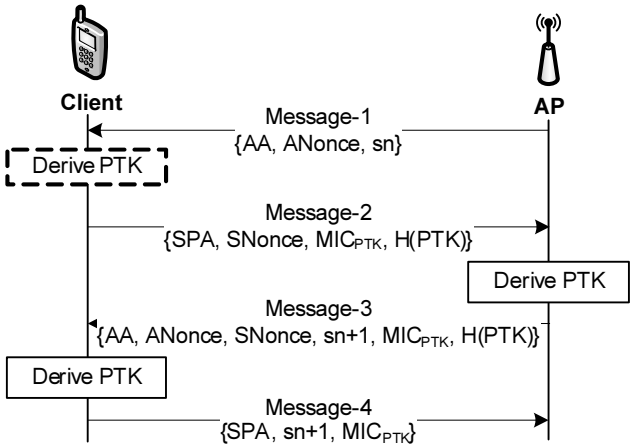

Fig. 4. Hash cookie

4.3 Comparison of our approaches

Table 1 depicts the comparison of our approaches. Our approaches have no memory consumption in the client during the 4-way handshake, thus they can mitigate the 4-way handshake blocking attack. However, they need more computation power to generate the cookie.

Table 1. Comparison of our approaches

Approach	4-way handshake	Encryption cookie	Hash cookie
Memory consumption in the client	ANonces, PTKs	None	None
Computation in the client with PTK	Derivation	Derivation Encryption Decryption	Derivation Hash Derivation
Handshake blocking	Weak	Robust	Robust
Assumption	None	Secret key in the client	None
Delay of key installation	T_{PTK}	$T_{PTK}+T_{EN}+T_{DE}$	$T_{PTK}+T_{HASH}+T_{PTK}$

Our approaches have a trade off. To mitigate the 4-way handshake attack, we need more computation and this will be time consuming. First, the encryption cookie approach needs time to encrypt and decrypt the PTK (called T_{EN} and T_{DE}), and the assumption of a secret key. Second, the hash cookie approach

needs the time to generate a hash value of the PTK (called T_{HASH}) and to derive the PTK (called T_{PTK}) one more time. However, because the delays T_{EN} and T_{DE} have insignificantly small values, we may tolerate this delay compare with the total delay during a 4-way handshake.

We simulated the AES process ten thousand times in Fedora 7 GNU/Linux, using the 2.6 kernel on a standard x86 processor. The results showed T_{EN} and T_{DE} to be only 1.5 and 2.3 micro-seconds respectively. Another trustworthy measurement can be found in [6]. The total delay of a 4-way handshake has been measured at about 60 milliseconds [8] hence the additional delay (3.5 micro-seconds) caused by our proposal is tolerable. We do not mention the time for PTK derivation, because it is flexible with the size of PTK. Consequently, our approaches to mitigate the attack cause only a small delay when compared with the overall authentication procedure.

5 Conclusion

802.11i is a well-designed standard, promising to improve the security of wireless LANs. However, security holes have been identified in the key derivation procedure in 802.11i. We have analyzed and studied the 4-way handshake protocol, that is, the key derivation procedure in 802.11i. The protocol has a vulnerability, where unprotected messages are used. This vulnerability may cause DoS attacks, exhausting the client's memory. A successful attack will cancel all previous efforts in 802.11i authentication.

To mitigate this vulnerability, we adopt a cookie-based mechanism. Our proposal is classified into two related solutions that use the information in the cookie. However, they use the same idea; that is, making the cookie contain the information required by a client and delivering it to another entity.

The 4-way handshake is an essential part of 802.11i. Thus, a DoS attack against the 4-way handshake is serious in the 802.11i authentication procedure. Using our proposals, the network can have a more secure 4-way handshake. Consequently, through a secure 4-way handshake, the network may provide a secure wireless LAN service for a client and an AP.

Acknowledgements

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA -2008-C1090-0801-0028)

References

1. Gartner Dataquest, "Global Telecommunications Market Take, 4Q03," 2004.
2. IEEE Std 802.11i, "Wireless Medium Access Control(MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements," July 2003.
3. Changhua He, John C. Mitchell, "Security analysis and improvements for IEEE 802.11i," The 12th Annual Network and Distributed System Security Symposium 2005.
4. Changhua He, John C. Mitchell, "Analysis of the 802.11i 4-Way Handshake," In Proceedings of the Third ACM International Workshop on Wireless Security 2004.
5. RFC 4987, "TCP SYN Flooding Attacks and Common Mitigations," August 2007.
6. Speed Comparison of Popular Crypto Algorithms, available at http://gd.tuwien.ac.at/privacy/crypto/libs/cryptlib/benchmarks.html, July 2004.
7. Hayriye Altunbasak, Henry Owen, "Alternative Pair-wise Key Exchange Protocols for Robust Security Networks (IEEE 802.11i) in Wireless LANs," IEEE SoutheastCon 2004.
8. tacci, R., Maccari, L., Pecorella, T. and F. Frosali, "A secure and performant token-based authentication for infrastructure and mesh 802.1X networks," IEEE Conference on Computer Communications, June 2006.