RL-003

Attack against WPA-TKIP using Vulnerability of QoS Packet Processing –WPA-TKIP is not safe in realistic environment–

藤堂 洋介 † 小澤 勇騎 † 大東 俊博 ‡ 森井 昌克 †
Yosuke Todo Yuki Ozawa Toshihiro Ohigashi Masakatu Morii

1 Introduction

WPA-TKIP (Wi-Fi Protected Access Temporal Key Integrity Protocol) is a security protocol that protects confidentiality and integrity for wireless LAN communication, and introduced the method that the vulnerability [2, 3, 4, 5] of WEP (Wired Equivalent Privacy) [1] is removed. Many researchers have discussed the security of WPA-TKIP [6, 7]. However, a realistic attack against WPA-TKIP was not known except a dictionary attack. Beck and Tews have proposed a message falsification attack on WPA-TKIP in 2008 [8]. Their attack (called the Beck-Tews attack) can recover the message integrity code (MIC) for 12-15 minutes, and a short encryption packet such as ARP packets can be forged. The Beck-Tews attack uses the chopchop attack, which is known as a replay attack on WEP. The attack works for only a network that supports IEEE 802.11e QoS features because WPA-TKIP has a preventing mechanism of the replay attack. WPA-TKIP has the TSC counter that grows each time the receiver receives the packet. If the received IV is less than or equal to the TSC counter, the received encrypted data is discarded. The communication using IEEE 802.11e has more access categories, and TSC counter is managed every access categories. The attacker selects the access category with a small TSC counter, and can execute the replay attack.

We have proposed the man-in-the-middle attack in JWIS2009 (the Ohigashi-Morii attack) [9, 10]. And, it works for a general network. However it is necessary to interrupt the communication between the access point and the client for executing the man-in-the-middle attack. It is not easy to execute the attack in a realistic environment. In this paper, we propose an executable attack in a realistic environment without requiring the man-in-the-middle attack. This attack uses the vulnerability of QoS packet processing of the IEEE 802.11e. Many wireless LAN implementations have this vulnerability. In addition, we show that the receiver receives a falsification packet made by our attack regardless of the setting of IEEE 802.11e. Therefore, almost all WPA-TKIP implementations cannot protect against the falsification attack in the realistic environment.

This paper is organized as follows: WPA-TKIP and IEEE 802.11e are shown in Sect. 2 and Sect. 3. In Sect. 4, we describe the Beck-Tews attack and the reverse chopchop attack. We propose a new falsification attack based on the vulnerability of QoS packet processing in Sect. 5, and show its experimental results in Sect. 6. The consideration of our results is shown in Sect. 7. Finally, we conclude this paper in Sect. 8.

2 Wi-Fi Protected Access

After various vulnerabilities were reported to WEP, IEEE Standards Association enacted a new encryption standard IEEE 802.11i [11]. IEEE 802.11i has chiefly three functions: user authentication function by EAP, integrity check function by TKIP, encryption function by AES. However, it is impossible to introduce AES into the existing model. Then, Wi-Fi Alliance [12] enacted WPA-TKIP to maintain security until spreading IEEE 802.11i. WPA-TKIP has user authentication function by EAP and integrity check function by TKIP.

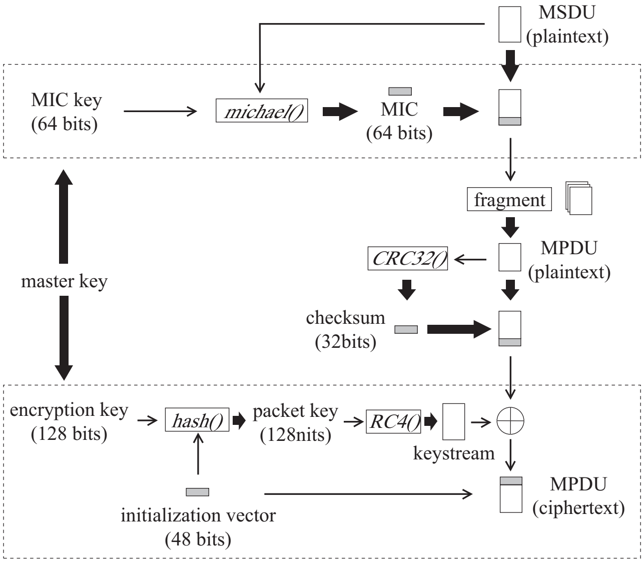


Fig.1 Processing of Sender

In WPA-TKIP, a 512-bit master key is shared between a client and an access point. This master key generate a 64-bit MIC key K^* and a 128-bit encryption key K. A MIC key is used to create a MIC. And, an encryption key is used to encrypt packets.

2.1 Processing of Sender

A sender calculates a MIC from a MIC key and a MAC Service Data Unit (MSDU) by using a message integrity check function MICHAEL. The MIC is added to the MSDU, as follows:

$$MSDU||michael(K^*, MSDU), \tag{1}$$

where $michael(K^*, MSDU)$ is a 64-bit MIC and $||$ is concatenation. The MSDU with the MIC is fragmented into MAC Protocol Data Units (MPDUs). A 32-bit checksum is calculated from each MPDU by using CRC32, and it is added to the MPDU, as follows:

$$MPDU||CRC32(MPDU), \tag{2}$$

where $CRC32(MPDU)$ is a 32-bit checksum.

Encryption of WPA is executed for each MPDU with the checksum. A packet key PK is generated from a 48-bit initialization vector (IV), an encryption key K, and a MAC address by using a specific hash function for WPA $hash()$. IVs for each MPDU are different, and the value of the IV is incremented by one when the IV is generated newly. In WPA, the IV is called the TKIP sequence counter (TSC).

† 神戸大学大学院工学研究科，Graduate School of Engineering, Kobe University

‡ 広島大学情報メディア教育研究センター，Information Media Center, Hiroshima University

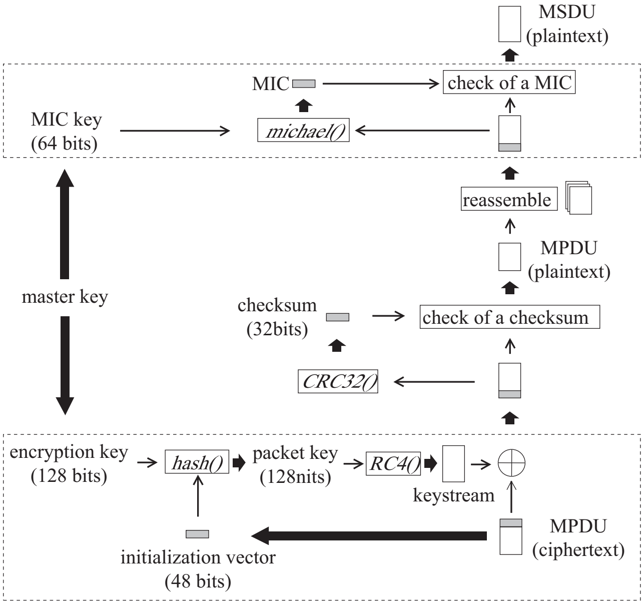


Fig.2　Processing of Receiver

A stream cipher RC4 is used as an encryption algorithm for WPA-TKIP. RC4 generates a pseudo-random sequence (called a keystream) $Z = (Z_1, Z_2, ..., Z_L)$ from a packet key and an IV, where Z_i is a byte variable and L is the length of a plaintext. The keystream is XOR-ed with a plaintext $P = (P_1, P_2, ..., P_L)$ to obtain a ciphertext $C = (C_1,\ C_2, ...,\ C_L)$ as follows:

$$C_i = P_i \oplus Z_i \quad (i = 1, 2, ..., L), \tag{3}$$

where C_i and P_i are a byte variable, respectively. Then, the encryption of WPA is written as follows:

$$C = (MPDU||CRC32(MPDU)) \oplus RC4(hash(K, IV)). \tag{4}$$

An encrypted MPDU and the IV are sent to the receiver. We show processes of sender on WPA in Fig. 1.

2.2　Processing of Receiver

The receiver receives an encrypted MPDU and an IV. The IV is compared with the TSC counter, which is a value of the IV corresponding to an encrypted MPDU accepted most recently. If the received IV is less than or equal to the TSC counter, the received encrypted MPDU is discarded. In the decryption of WPA, the receiver generates a keystream Z from a received IV and a packet key PK. The keystream Z is same as that of the sender Z. A plaintext P is obtained as follows:

$$P_i = P_i \oplus Z_i \oplus Z_i = C_i \oplus Z_i \quad (i = 1, 2, ..., L). \tag{5}$$

Then, the decryption of WPA is written as follows:

$$(MPDU||CRC32(MPDU)) = C \oplus RC4(hash(K, IV)). \tag{6}$$

The receiver calculates a checksum from the received MPDU, and the checksum is compared with the received checksum. If these checksums differ, the received MPDU is discarded. Note that the receiver does not send the error message of checksum to the sender.

Table 1　Access categories of IEEE 802.11e

Access category	priority	Description
Voice	7, 6	Highest priority Voice data such as VoIP
Video	5, 4	second priority video data
Best Effort	0, 3	third priority Traffic from legacy devices or applications
Background	2, 1	Low priority file downloads, print jobs

When all MPDUs are obtained, these are reassembled to the MSDU. The receiver calculates a MIC from the received MSDU and the MIC key by using Michael, and the MIC is compared with the received MIC. If these MICs differ, all the received MPDUs corresponding to the MSDU are discarded and the receiver sends the error message of MIC (a MIC failure report frame) to the sender. In WPA, the MIC key is changed if more than two error messages of MIC are sent to the sender in less than a minute. When the MSDU is accepted, the TSC counter is updated to the largest value in the IVs corresponding to all the MPDUs. We show processes of receiver on WPA in Fig. 2.

3　IEEE 802.11e and WMM

There is various technology that control a quality of service on the network. IEEE 802.11e is a technology that control a quality of service on the wireless LAN network. IEEE 802.11e has two methods for QoS control. The first method achieves the QoS control by adding the priority to each packet and the second method offers a priority to each implementation by handling the controller. The first method has the certification program named WMM by Wi-Fi Alliance [13]. In this paper, IEEE 802.11e indicates QoS control by WMM.

The mechanism of IEEE 802.11e is shown. The communication using IEEE 802.11e has four access categories. Table 1 shows the feature and the role of four kinds of access categories. An actual communication classifies data by using the value of the priority. Moreover, TSC counter is managed in each access category in IEEE 802.11e. Then each TSC counter is different in each priority. The attacker can capture the encryption packet of $IV = x$, and selects the priority of $TSC \leq x - 1$ and executes the replay attack like the chopchop attack.

4　Related Works

4.1　Beck-Tews Attack

The Beck-Tews attack [8] is a method that applies the chopchop attack [4] on WEP to WPA-TKIP. This attack recovers a MIC key and a plaintext from an encrypted short packet, and falsifies its

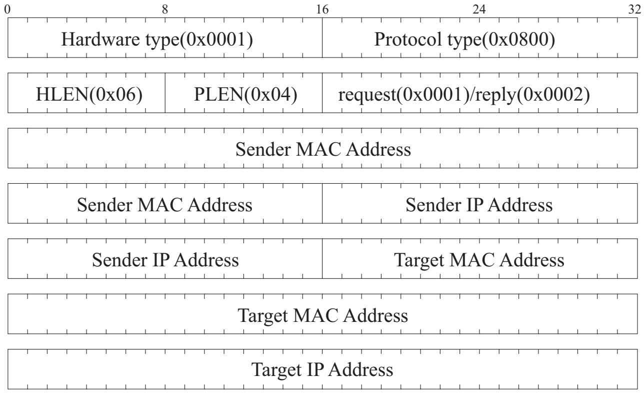


Fig.3 Structure of ARP packet

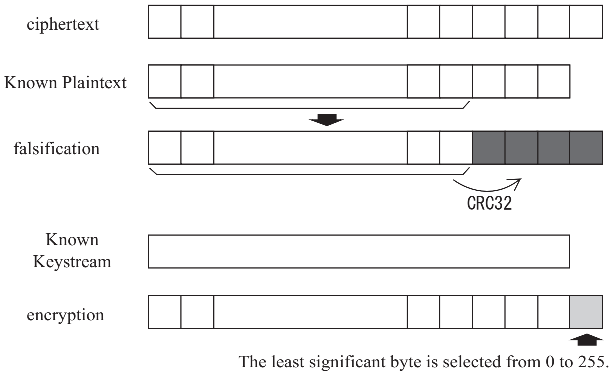


Fig.4 Reverse chopchop attack

packet, practically.

4.1.1 The Chopchop Attack on WEP

The purpose of the chopchop attack on WEP is to obtain the information of a plaintext from a given ciphertext. Note that this attack cannot obtain an encryption key of WEP.

Processes of WEP are different from WPA as follows:

1. The value of IV is not checked.
2. There is not a process of adding a MIC.
3. The receiver sends the error message of checksum to the sender.

An encrypted packet that is falsified from an encrypted packet accepted in the past is not discarded since the value of IV is not checked. Integrity check of a message is executed by only the checksum, and the receiver sends the error message of checksum to the sender if the checksum is incorrect.

The chopchop attack focuses on a property of CRC32. It is generally used with CRC32 as an error detecting code. In WEP, CRC32 plays the role as Message Authentication Code (MAC). However, if attacker knows the least significant byte, the attacker can restore CRC32 that chops off the least significant byte. The modification of a plaintext by XOR operation is executed easily in the encryption of the stream cipher. The chopchop attack restores the keystream by using this character. In WEP, the least significant byte of CRC32 is encrypted. Then, the attacker calculates CRC32 that chops off some candidate values of the least significant byte ($0xFF$ from $0x00$ for instance), and sends the falsification packet that attach CRC32 to the client or the access point. If the forecast of the least significant byte is mistaken, the receiver sends the error message of checksum. Therefore, the attacker can know that the predictive value is a correct value when the error message is not sent. In repeating this attack, the attacker can know all bytes of the keystream.

4.1.2 Applies to WPA-TKIP

The Beck-Tews attack is an application of the chopchop attack to WPA-TKIP. However, WPA-TKIP doesn't send the error message of checksum. Then, the attacker uses that the MIC check is executed when CRC32 agrees. The probability that MIC agrees by chance is very low($1/2^{64}$). Then, the attacker can know the value of a correct plaintext by observing the error message of MIC. However, the MIC key is changed if more than two error messages of MIC are sent to the sender in less than a minute. Then, the Beck-Tews attack needs the standby time for one minute after 1 byte is restored. Thus, it is not effective for the attack when the unknown bytes of the target packet are large. Then, they paid attention to the ARP packet. The ARP packet can guess the plaintext with a high probability. Figure 3 shows the structure of the ARP packet. They used that sender and receiver's IP addresses (respective lowest byte) can forecast in a high probability. This is appropriate assumption, because many users use wireless LAN implementation in the initial state. In this case, 14 bytes (data, MIC, and checksum) become unknown. The Beck-Tews attack executes the chopchop attack 12 times for the ARP packet, and MIC and checksum are restored. Sender and receiver's IP addresses (respective lowest byte) are restored by the comparison with checksum. Since, MICHAEL is a reversible function, the MIC key is easily restorable from the ARP packet and MIC. Now, the attacker obtained keystream corresponding to the MIC key and IV. The attacker can counterfeit the encryption packet, the size of which is the same as the keystream.

They showed in the method of shortening the execution time of the Beck-Tews attack when the attacker knows the MIC key. This works effectively when the attacker attacks it again for the period when the MIC key has not been renewed. If the attacker knows the MIC key, she/he is computable MIC from the MIC key and the ARP packet. The attacker executes the chopchop attack four times and restores correct checksum. Then, the attacker can obtain plaintext information since she/he compares correct checksum with the candidate of checksum. In addition, the attacker becomes possible the restoration of the keystream and can counterfeit the encryption packet. The execution time of the attack becomes about 4 minutes since waiting time for MIC error is 3 minutes.

4.2 Reverse Chopchop Attack

We have proposed the technique for executing the message falsification attack at high speed in SCIS2010. The reverse chopchop attack is one of the proposed attacks in SCIS2010 [14], and restores the keystream from higher bytes of the packet. The theory and the effect of the reverse chopchop attack are described in this section.

First, the theory of the attack, the attacker should know that higher bytes of the packet is already-known. At this time, CRC32 is calculated from the data that removes the lowest three bytes from already-known bytes. When this data is encrypted, all data except the least significant byte can be correctly encrypted. And, the falsification packet that tests 256 kinds of the least significant byte is sent. Then, passing the CRC32 check only becomes one kind for the receiver, and the MIC error is returned at a high probability because MIC becomes a disagreement. The attacker can restore one unknown byte of a keystream, because the attacker understands

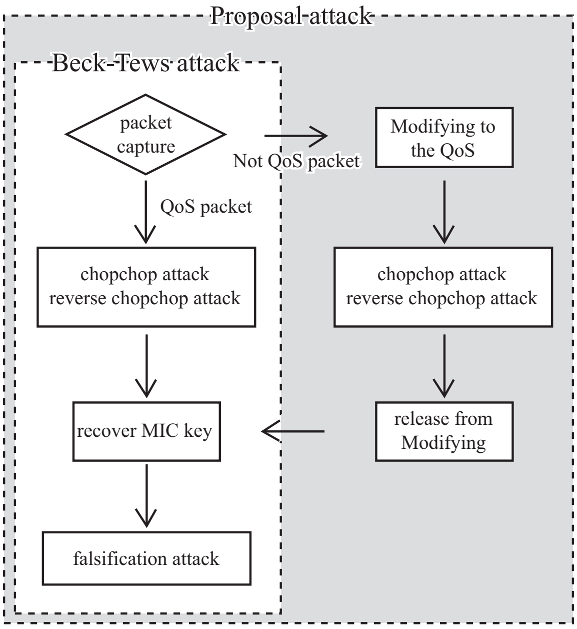


Fig.5 QoS forgery attack

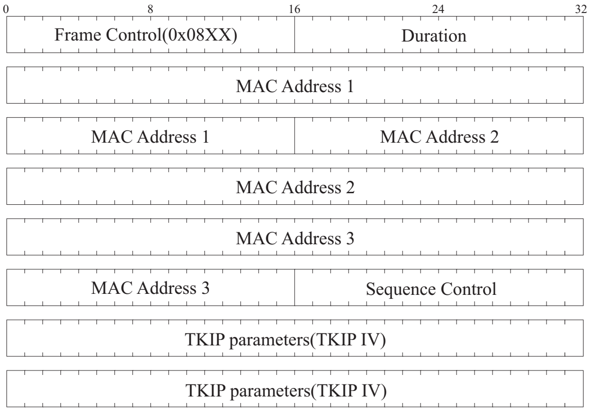


Fig.6 Structure of IEEE 802.11 header of usual packet

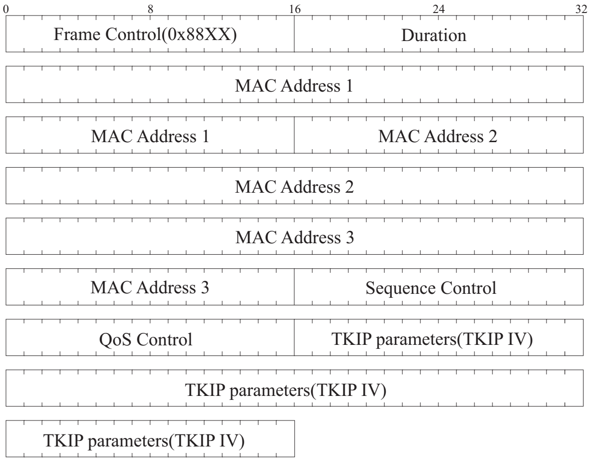


Fig.7 Structure of IEEE 802.11 header of QoS packet

that the packet when MIC error is detected was a correct ciphertext. Figure 4 shows the process of this attack.

Next, an actual effect of the attack show. First, this attack can shorten the execution time until restoring the MIC key. The reverse chopchop attack need not restore checksum by the chopchop attack. The effect equal with the Beck-Tews attack can be achieved for eight reverse chopchop attacks. Then, this attack can recover MIC for 7-9 minutes. Second, this attack can execute the falsification attack at high speed after the MIC key is restored, because the reverse chopchop attack can immediately restore Internet Protocol address of the client. Therefore, the falsification attack can be executed in 10 seconds on the average. Third, this attack can falsify a variable-length packet. The reverse chopchop attack can restore the keystream of length more than the keystream used for the chopchop attack. Therefore, this attack can falsify a variable-length packet, but it is necessary for one minute to enhance the keystream by one byte. Finally, this attack can execute the information gathering attack. The reverse chopchop attack can restore IP address of PC that belongs to a local network, before restoring CRC32 and MIC. This IP address is a significant value unlike MIC and CRC. Even if it is a network where the update interval of the MIC key is short, it is difficult to prevent the information gathering attack because the execution time is only ten seconds.

5 QoS Forgery Attack

The Beck-Tews attack is an executable attack only to the network where IEEE 802.11e is supported. However, IEEE 802.11e can be turned off depending on the setting of the access point, and the client connected with the access point which IEEE 802.11e was not supported couldn't attack. On the other hand, the Ohigashi-Morii attack can attack a general implementation. However it is necessary to interrupt the communication between the access point and the client for executing the man-in-the-middle attack. It is not easy to execute the attack in a realistic environment. Then, we propose an executable attack in a realistic environment without requiring the man-in-the-middle attack. The proposed attack doesn't depend on function that IEEE 802.11e enables or not, because this attack uses the vulnerability of processing it of the IEEE 802.11e function of the access point or the client. Many wireless LAN implementations have this vulnerability and can be attacked. The proposed attack includes the Beck-Tews attack, and Fig. 5 shows the flow of the entire attack. The next paragraph shows details of the proposed attack.

5.1 Packet Capture Section

First, the attacker checks the structure of the captured packet. Figure 6 shows the structure of IEEE 802.11 header of usual packet, and Fig. 7 shows the structure of IEEE 802.11 header of QoS packet [*1]. The following three contents are checked in the Packet capture section.

- It is checked whether the captured packet is ARP packet.
- It is checked whether the captured packet is packet which sends from the access point to the client.
- It is checked whether the captured packet is QoS packet.

The attacker uses ARP packet that can guess the plaintext and can be easily falsified. However, the received packet is encrypted, so whether it is ARP packet cannot be judged by using the protocol identifier. Therefore, the attacker judges ARP packet by the packet

[*1] Source MAC Address, Destination MAC Address, and BSSID are inserted in three MAC address fields.

length. Since ARP packet is a fixed length. Next, the attacker checks whether the captured packet is packet which sends from the access point to the client. Since the access point doesn't send the MIC error, and we cannot execute the chopchop attack for the access point. Finally, the attacker checks whether the captured packet is QoS packet. According to the comparison between Fig. 6 and Fig. 7, we notice the value of the frame control of the IEEE 802.11 header is different. ARP packet has structure of the data frame. The frame control of the QoS packet is “0x88”, but the frame control of the usual packet is “0x08”. This flow is similar to the Beck-Tews attack. However, if the attacker uses the Beck-Tews attack, she/he cannot attack packets other than the QoS packet. Then, the attacker removes the filter to judge whether this packets is QoS packet. Since, if the attacker uses the proposed attack, she/he can attack packets other than the QoS packet in many case.

5.2 Modifying to the QoS Section

We show the method of modifying to the QoS section. According to the comparison between Fig. 6 and Fig. 7, we notice two differences. First, most significant byte of the frame control is different. The attacker rewrites most significant byte to “0x88”. If the receiver receives this packet, he processes this packet as QoS packet regardless of the IEEE 802.11e function. Second, QoS packet has the QoS Control field. The priority shown in Sect. 2 is inserted in this field. The QoS Control field doesn't exist in a usual data packet. Then, the attacker inserts QoS control field in the captured packet. And the attacker sets the appropriate priority.

5.3 Recover MIC key Section

The attacker executes the chopchop attack or the reverse chopchop attack by using the QoS forgery packet, and she/he recovers MIC. We omit explaining, because the flow of this attack is an existing attack introduced in Sect. 4.1.2 and Sect. 4.2.

Next, the attacker recovers MIC key. In this time, the attacker must release from modifying. Because MIC is calculated from various data including priority, as follows:

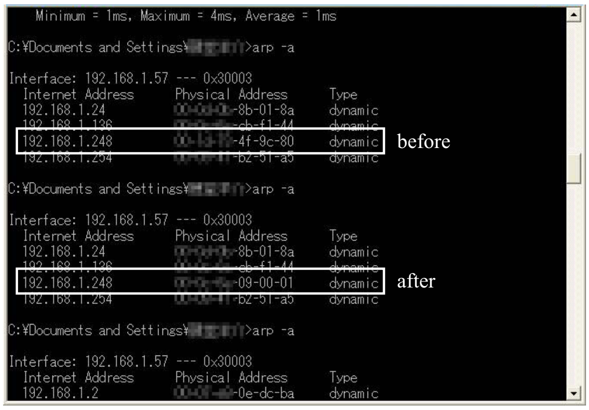
$$\begin{aligned} MIC = MICHAEL(&MICKey, DestinationMACAdress,\\ &SourceMACAdress, QoS\,priority, Data). \end{aligned} \quad (7)$$

Namely MIC that we recovered by using the chopchop attack or the reverse chopchop attack is MIC before modifying to the QoS. Then, the attacker uses the inverse function of Michael after releasing the QoS forgery and she/he recovers the MIC key. The attacker can falsify the packet with this MIC key, because the MIC key doesn't depend on the priority or the packet structure.

5.4 Packet Falsification Section

The attacker makes the falsification packet and sends it to the target. We discuss the ARP cache poisoning as an example of our attack.

We know various methods to execute ARP cache poisoning. In this paper, we execute ARP cache poisoning using forgery packet of ARP request. Since this method is executable without depending on the situation of ARP request of the client. We show the method that the attacker makes forgery packet of ARP request. First, the attacker modifies the ARP packet to ARP request, and she/he sets target MAC Address and target IP Address to MAC address and IP address that target owns. Thus, the attacker can send the target the ARP request. Next, the attacker sets sender MAC Address and sender IP Address to forgery MAC address and forgery IP address. The target memorizes this forgery MAC address and forgery IP address. In this case, it is necessary to set the value of three MAC Address in Fig. 7 appropriately. Figure 8 shows the result of ARP cache poisoning. We can confirm the MAC address of IP address (192.168.1.248) is poisoning from XX-XX-XX-4f-9c-60 to XX-XX-XX-09-00-01. The first three bytes (XX-XX-XX) identify the organization that issued the identifier, and we don't disclose this identifier in this paper. Moreover, the attacker can cause IP address conflict by setting the same value to Target IP Address and Sender IP Address.



```
  Minimum = 1ms, Maximum = 4ms, Average = 1ms

C:¥Documents and Settings¥[redacted]>arp -a

Interface: 192.168.1.57 --- 0x30003
  Internet Address      Physical Address      Type
  192.168.1.24          [redacted]-8b-01-8a     dynamic
  192.168.1.136         [redacted]-cb-f1-44     dynamic
  192.168.1.248         [redacted]-4f-9c-80     dynamic
  192.168.1.254         [redacted]-b2-51-a5     dynamic

C:¥Documents and Settings¥[redacted]>arp -a

Interface: 192.168.1.57 --- 0x30003
  Internet Address      Physical Address      Type
  192.168.1.24          [redacted]-8b-01-8a     dynamic
  192.168.1.136         [redacted]-cb-f1-44     dynamic
  192.168.1.248         [redacted]-09-00-01     dynamic
  192.168.1.254         [redacted]-b2-51-a5     dynamic

C:¥Documents and Settings¥[redacted]>arp -a

Interface: 192.168.1.57 --- 0x30003
  Internet Address      Physical Address      Type
  192.168.1.2           [redacted]-0e-dc-ba     dynamic
```

Fig.8 ARP cache poisoning

Table 2 Experimental result

	IEEE 802.11e*	release	result
Co.A(USB)	enable	2007	Success
Co.A(USB)	disable	2007	Success
Co.A(CardBus)	disable	2004	Failure
Co.B(USB)	disable	2009	Success
Co.B(CardBus)	disable	2006	Failure
Co.C(CardBus)	disable	2007	Success
Co.D(CardBus)	disable	2006	Success
Co.E(chipset)	enable	2008	Success
Co.F(chipset)	enable	2008	Success
Co.G(chipset)	enable	2006	Success

* Judgment from Web page or product specification, etc.

6 Experiment

In this section, we evaluate our attack to examine the kind of the product that can be attacked. We disable the IEEE 802.11e function of the access point. Namely, we evaluate our attack in the environment not to be able to execute the Beck-Tews attack, because QoS packet is not sent on the target network. On the other hand, we experimented to three kinds of clients (USB type, CardBus type, and Chipset with built-in PC). We execute ARP cache poisoning attack introduces in Sect. 5.4, and we judge that the proposed attack succeeded by having succeeded in rewriting the ARP table. Table 2 shows the result of the experiment.

From the result of the experiment, we can understood many wireless LAN implementations are the target. Moreover, we could attack many implementations assumed not to have the IEEE 802.11e function. On the other hand, we couldn't attack Co.B (CardBus,2006) because the communication is intercepted by one chopchop attack. But, this implementation is a breach of the pro-

Table 3 Compare the Beck-Tews attack and the proposed attack

	Access Point	client	Network
Beck-Tews attack	QoS enable	QoS enable	QoS enable
Proposal attack	-	IEEE 802.11e function (chipset)	QoS disable

tocol. In addition, we couldn't attack Co.A (CardBus,2004) because this client had been released before standardization of IEEE 802.11e. All implementations that we can attack have the IEEE 802.11e function in the chipset. Then, to prevent the attack, we should confirm the function of the chipset.

7 Consideration

In this section, we compare the Beck-Tews attack and the proposed attack. Next, we consider the technique for preventing the proposed attack.

First, we compare the Beck-Tews attack and the proposed attack. The target of the Beck-Tews attack was a network where IEEE 802.11e was supported. However, the proposed attack can be executed without depending on the setting of the access point. Therefore, the target of the proposed attack can be enhanced to the client that chipset corresponds to IEEE 802.11e. And many clients put on the market in recent years have the IEEE 802.11e function for the unit of the chipset. Namely, almost all implementations of WPA-TKIP can be attacked.

Next, we consider the technique for preventing the proposed attack. First, venders should immediately solve this vulnerability. However, we should consider the technique of preventing this attack until the vulnerability is removed. Then, we strongly recommend the shift to WPA2-AES, because, the user cannot receive the favor of IEEE 802.11e as long as the user uses WPA-TKIP. However we consider another technique for preventing the proposed attack. The method that we set the key update interval is shorter to prevent the proposed attack is known [8, 15]. However, we should pay attention to the method. Because this method that we set the key update interval is shorter cannot prevent the attack for the information gathering. If you use specific client utility, it may be possible to prevent the proposed attack. However, whether the attack was able to be prevented was influenced also by compatibility with the access point. Namely, reliability as method of preventing attack will not be able to be kept.

8 Conclusion

In this paper, we proposed an executable attack in a realistic environment without requiring the man-in-the-middle attack. This attack cannot execute all implementations, but this attack is easily executable by the realistic environment. Moreover, many wireless LAN implementations have vulnerability. Namely, if the attacker uses proposal attack, many wireless LAN implementations can be attacked.

ACKNOWLEDGEMENTS

This work was supported in part by Grant-in-Aid for Scientific Research (KAKENHI 21700018) of JSPS.

Reference

[1] IEEE Computer Society, "Wireless lan medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802.11, 1999.

[2] E. Tews, R. Weinmann, and A. Pyshkin, "Breaking 104 bit WEP in lessthan 60 seconds," Cryptology ePrint, 2007, available at http://eprint.iacr.org/2007/120.pdf

[3] R. Teramura, Y. Asakura, T. Ohigashi, H. Kuwakado, and M. Morii, "Breaking WEP with Any 104-bit Keys –All WEP Keys Can Be Recovered Using IP Packets Only–," Proc. of SCIS2009, CDROM, 1A2-6, Jan. 2009.

[4] KoreK, "chopchop (Experimental WEP attacks)," 2004, available at http://www.netstumbler.org/showthread.php?t=12489

[5] W. A. Arbaugh, "An Inductive Chosen Plaintext Attack against WEP/WEP2," available at http://www.cs.umd.edu/~waa/attack/frame.htm

[6] R. Moskowitz, "Weakness in Passphrase Choice in WPA Interface," 2003, available at http://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html

[7] V. Moen, H. Raddum, and K.J. Hole, "Weaknesses in the temporal key hash of WPA," ACM SIGMOBILE Mobile Computing and Communications Review, vol.8, pp.76–83, 2004.

[8] M. Beck and E. Tews, "Practical Attacks Against WEP and WPA," Proc. PacSec'08, pp.79–85, 2008.

[9] T. Ohigashi and M. Morii, "A practical message falsification attack on WPA," Proc. JWIS 2009, CDROM, 5A-4, 2009.

[10] Y. Ozawa, T. Ohigashi, M. Morii, "Weaknesses on WPA-TKIP and Application to the Message Falsification Attack," CSS2009, CDROM, vol.2009, no.11, pp.805–810, 2009. (in Japanease)

[11] IEEE Std 802.11i-2004, "Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Medium Access Control (MAC) Security Enhancements," IEEE, July 2004.

[12] Wi-Fi Alliance, available at http://www.wi-fi.org/

[13] Wi-Fi Alliance, "Wi-Fi CERTIFIED™ for WMM™ - Support for Multimedia Applications with Quality of Service in Wi-Fi® Networks," available at http://www.wi-fi.org/files/wp_1_WMM%20QoS%20In%20Wi-Fi_9-1-04.pdf

[14] Y. Todo, Y. Ozawa, T. Ohigashi, and M. Morii, "A Study on a Message Falsification Attack on WPA-TKIP," Proc. SCIS2010, CDROM, 2010. (in Japanease)

[15] Y. Oiwa, K. Kobara, R. Yamaguchi, G. Hanaoka, H. Watanabe, "RCIS Technical Notices 2009-01 (B)," Aug. 2009. (in Japanese)