PAPER

# Falsification Attacks against WPA-TKIP in a Realistic Environment

**Yosuke TODO**[†a)], **Yuki OZAWA**[†], ***Student Members***, **Toshihiro OHIGASHI**[††], ***Member***, ***and*** **Masakatu MORII**[†],

**SUMMARY** In this paper, we propose two new falsification attacks against Wi-Fi Protected Access Temporal Key Integrity Protocol (WPA-TKIP). A previous realistic attack succeeds only for a network that supports IEEE 802.11e QoS features by both an access point (AP) and a client, and it has an execution time of 12–15 min, in which it recovers a message integrity code (MIC) key from an ARP packet. Our first attack reduces the execution time for recovering a MIC key. It can recover the MIC key within 7–8 min. Our second attack expands its targets that can be attacked. This attack focuses on a new vulnerability of QoS packet processing, and this vulnerability can remove the condition that the AP supports IEEE 802.11e. In addition, we discovered another vulnerability by which our attack succeeds under the condition that the chipset of the client supports IEEE 802.11e even if the client disables this standard through the OS. We demonstrate that chipsets developed by several kinds of vendors have the same vulnerability.

***key words:*** *wireless LAN network, WPA-TKIP, falsification attack, QoS, vulnerability*

## 1. Introduction

Wi-Fi Protected Access Temporal Key Integrity Protocol (WPA-TKIP) is a security protocol that protects data confidentiality and integrity in wireless LAN communication. This protocol was introduced in order to prevent the vulnerability [1]–[4] of Wired Equivalent Privacy (WEP) [5]. Several researchers have deliberated about the security aspects of WPA-TKIP [6], [7]. However, thus far, no realistic attack against WPA-TKIP, except for the dictionary attack, had been known. In 2008, Beck and Tews proposed a falsification attack on WPA-TKIP [8]. Their attack (called the Beck-Tews attack) has an execution time of 12–15 min, in which the message integrity code (MIC) can be recovered, enabling the forging of short encryption packets such as ARP packets. The Beck-Tews attack is based on the chopchop attack, known as the replay attack, on WEP. Since WPA-TKIP has a mechanism for preventing the replay attack, the Beck-Tews attack succeeds only in the case of a network that supports IEEE 802.11e features. WPA-TKIP has a TSC counter that increases every time the receiver receives regular data. If the received initialization vector (IV) is less than or equal to the TSC counter, the received encrypted packet is discarded. The communication using IEEE 802.11e has four access categories, and the TSC counter is managed for each access category. The attacker selects the access category with a small TSC counter and executes the replay attack. In addition, Beck and Tews also proposed an attack based on a man-in-the-middle attack, which can be executed under the condition that a network does not support IEEE802.11e [8]. In 2009, Ohigashi and Morii considered a concrete procedure of the man-in-the-middle attack against WPA-TKIP and we call this attack the Ohigashi-Morii attack [9]. These attacks expand their targets to other products that do not support IEEE 802.11e. However, it is necessary to interrupt communication between the access point (AP) and the client to execute the man-in-the-middle attack, which makes it difficult to execute the Ohigashi-Morii attack in a realistic environment.

In this paper, we propose two new attacks against WPA-TKIP. The first is a high-speed falsification attack that can recover a MIC key within 7–8 min. The second is an attack that expands its targets in a manner different from that of the Ohigashi-Morii attack (man-in-the-middle attack). This proposed attack exploits the newly discovered vulnerability in the QoS packet processing feature of IEEE 802.11e. The receiver receives a falsification packet despite the fact that the communication does not support IEEE 802.11e. This vulnerability removes the condition that the AP supports IEEE 802.11e. We execute this proposed attack against clients chosen at random. As a result, our attack succeeds under the condition that the chipset of the client supports IEEE 802.11e even if the client disables this standard through the OS. Many chipsets of clients available in the market in recent years support IEEE 802.11e. Therefore, if other clients have this vulnerability, almost all WPA-TKIP implementations would fail to protect a system against the falsification attack in a realistic environment.

This paper is organized as follows. WPA-TKIP and three systems for preventing the falsification attack are discussed in Sect. 2. In Sect. 3, the Beck-Tews attack is described. Next, the two proposed attacks are discussed. The high-speed falsification attack is described in Sect. 4, and the new falsification attack that is based on the vulnerability of QoS packet processing is described in Sect. 5, respectively. In Sect. 6, our proposed attacks are evaluated. Finally, the paper is concluded in Sect. 7.

Manuscript received October 29, 2010.
Manuscript revised September 21, 2011.

[†]The authors are with the Graduate School of Engineering, Kobe University, Kobe-shi, 657–8501 Japan.

[††]The author is with Information Media Center, Hiroshima University, Higashihiroshima-shi, 739–8511 Japan.

a) E-mail: todo@stu.kobe-u.ac.jp

DOI: 10.1587/transinf.E95.D.588

## 2. Wi-Fi Protected Access

After the vulnerability of WEP was reported [1]–[4], the IEEE Standards Association formulated a new encryption standard, IEEE 802.11i [10]. This standard has three main functions: user authentication function by the Extensible Authentication Protocol (EAP), integrity check function by TKIP, and encryption function by the Advanced Encryption Standard (AES). However, it is impossible to introduce AES in the legacy model. Then, the Wi-Fi Alliance [11] introduced WPA-TKIP as an interim measure until the use of IEEE 802.11i becomes widespread. As mentioned above, WPA-TKIP has a user authentication function by EAP and an integrity check function by TKIP.

In WPA-TKIP, a 256-bit pairwise master key (PMK) is shared between the AP and the client. In WPA-PSK, a secret key is pre-shared between the AP and the client, and the PMK is calculated by the pre-shared key [10]. On the other hand, in WPA-802.1X, the authentication server authenticates the client and negotiates a secure PMK [12]. In each association, a 512-bit pairwise transient key (PTK) is shared between the AP and the client by using the PMK and four-way handshake. The PTK generates a 64-bit MIC key $K^*$ and a 128-bit encryption key $K$. The MIC key is used to generate a MIC, and the encryption key is used to encrypt packets.

### 2.1 Processing for Sender

A sender calculates a MIC from the 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:

$$MSDU\|michael(K^*, MSDU),$$

where $michael(K^*, MSDU)$ is a 64-bit MIC and $\|$ denotes 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 is added to the MPDU as follows:

$$MPDU\|CRC32(MPDU),$$

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

Encryption of WPA is executed for each MPDU with the checksum. A packet key $PK$ is generated from a 48-bit IV, an encryption key $K$, and a sender MAC address by using a specific hash function for WPA, $hash()$. Each MPDU has a different IV and the value of the IV is incremented by 1 each time a new IV is generated. In WPA, the IV is called the TKIP sequence counter (TSC).

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

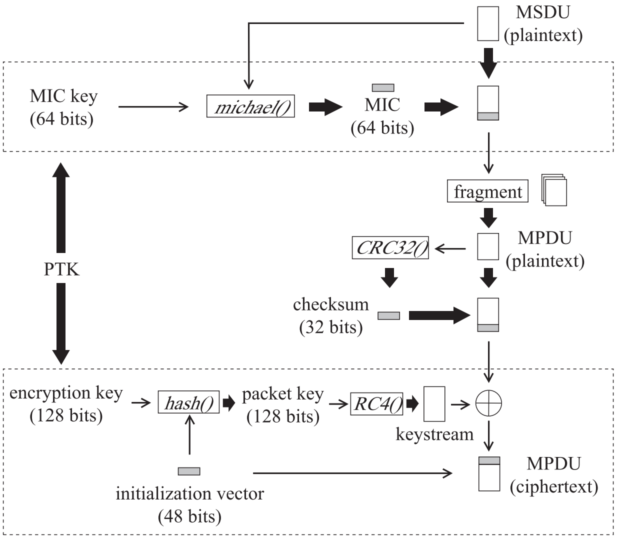

**Fig. 1** Processing for sender.

$$C_i = P_i \oplus Z_i \quad (i = 1, 2, \ldots, L),$$

where both $C_i$ and $P_i$ are one-byte variable. The encryption of WPA is then expressed as follows:

$$C = (MPDU\|CRC32(MPDU)) \oplus RC4(PK).$$

The encrypted MPDU and the IV are sent to the receiver. The processing carried out for the sender in WPA is shown in Fig. 1.

### 2.2 Processing for Receiver

The receiver receives an encrypted MPDU and an IV. Then, the received IV is compared with the TSC counter, which is an IV value corresponding to the encrypted MPDU accepted most recently. If the received IV is less than or equal to the TSC counter, the received encrypted MPDU is discarded. As a result, attackers cannot execute the replay attack. We call this method by which the packet is discarded the *TKIP-IV check*.

In the decryption of WPA, the receiver generates a keystream $Z$ from the received IV and the packet key $PK$. The keystream $Z$ of the receiver is the same as that of the sender. A plaintext $P$ is obtained as follows:

$$P_i = P_i \oplus Z_i \oplus Z_i = C_i \oplus Z_i \quad (i = 1, 2, \ldots, L).$$

The decryption of WPA is expressed as follows:

$$(MPDU\|CRC32(MPDU)) = C \oplus RC4(PK).$$

The receiver calculates a checksum from the received MPDU, and this calculated checksum is compared with the received checksum. If their values are different, the received MPDU is discarded. It should be noted that the receiver does not send an error message of the checksum to the sender. We

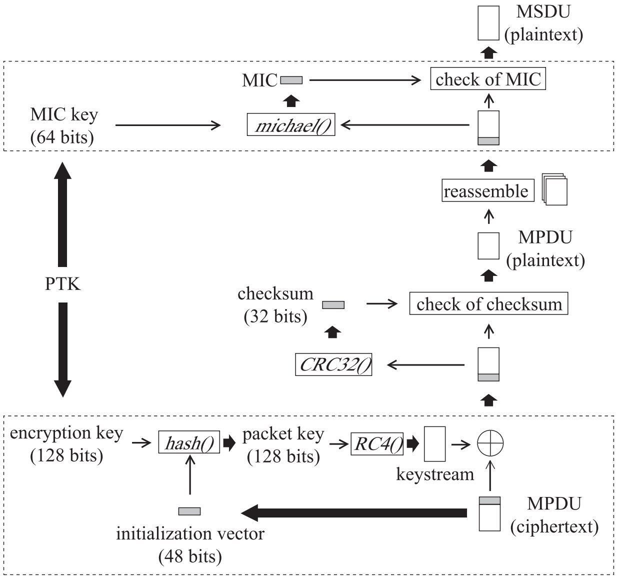

**Fig. 2** Processing for receiver.

call this method by which the packet is discarded the *checksum check*.

When all MPDUs are obtained, they are reassembled to form the MSDU. The receiver calculates a MIC from the received MSDU and the MIC key by using the function Michael, and then, the calculated MIC is compared with the received MIC. If their values are different, all the received MPDUs corresponding to the MSDU are discarded and the receiver sends an error message of MIC (MIC failure report frame) to the sender. In WPA, the MIC key is changed if more than two MIC failure report frames are sent to the sender in less than 1 min. When the MSDU is accepted, the TSC counter is updated to the largest value of the IVs corresponding to all the MPDUs. We call this method by which the packet is discarded the *MIC check*. The processing carried out for the receiver in WPA is shown in Fig. 2.

## 3. The Beck-Tews Attack

As explained Sect. 2.2, WPA-TKIP includes three methods that prevent the falsification attack (the TKIP-IV check, the checksum check, and the MIC check). The Beck-Tews attack [8] can break these three methods. For breaking the TKIP-IV check, Beck and Tews used a special feature of IEEE 802.11e. For breaking the checksum check, they proposed a method in which the chopchop attack [3] on WEP is applied to WPA-TKIP. For breaking the MIC check, they proposed a reversible function of MICHAEL. Then, this attack recovers the MIC key and plaintext from an encrypted short packet and falsifies the packet.

### 3.1 The Method for Breaking the TKIP-IV Check

Here, we describe IEEE 802.11e (QoS control for a wireless LAN network) against which the Beck-Tews attack can execute. QoS control is a technology by which the quality of service in a network is controlled. Various technologies have been proposed for QoS control. IEEE 802.11e is a technology that controls the QoS in a wireless LAN network. IEEE 802.11e includes two methods for QoS control. In one method, QoS control is achieved by assigning priority to each packet, whereas in the other method, a priority is assigned to each implementation by handling the controller. The first method has been certified as Wi-Fi Multimedia (WMM) by the Wi-Fi Alliance [13]. In this paper, QoS control by WMM is referred to as IEEE 802.11e.

**Table 1** Access categories of IEEE 802.11e.

| Access category | Priority | Description |
|---|---|---|
| Voice | 7, 6 | **Highest priority**<br>Voice data such as VoIP |
| Video | 5, 4 | **Second-highest priority**<br>Video data |
| Best effort | 0, 3 | **Third-highest priority**<br>Traffic from legacy devices or applications |
| Background | 2, 1 | **Lowest priority**<br>File downloads, print jobs |

The mechanism of IEEE 802.11e is as follows. Communication using IEEE 802.11e has four access categories. Table 1 lists the specifications and roles of the four different access categories. An actual communication classifies data on the basis of priority. Moreover, the TSC counter is managed in each access category in IEEE 802.11e. Therefore, each priority has a different TSC counter. When attacker captures the encryption packet of $IV = x$, she/he selects the priority that TSC counter is less than or equal to $x - 1$ and executes the replay attack like the chopchop attack. In this time, an IV of the received packet is less than the TSC counter of the priority and the TKIP-IV check does not operate. The Beck-Tews attack can execute a falsification attack by using abovementioned technique.

### 3.2 The Method for Breaking the Checksum Check

For breaking the checksum check, the Beck-Tews attack uses the chopchop attack executed against WEP. The purpose of the chopchop attack is to obtain information about a plaintext from a given ciphertext. It should be noted that this attack cannot acquire the encryption key of WEP.

We now explain the chopchop attack against WPA-TKIP. This attack focuses on the properties of CRC32, which is generally used as an error detection code. If the attacker knows the least significant byte of CRC32, she/he can restore CRC32 that chops off the least significant byte of the payload. A plaintext can be easily modified by an XOR operation in the encryption of the stream cipher. The chopchop attack restores the keystream by using this charac-

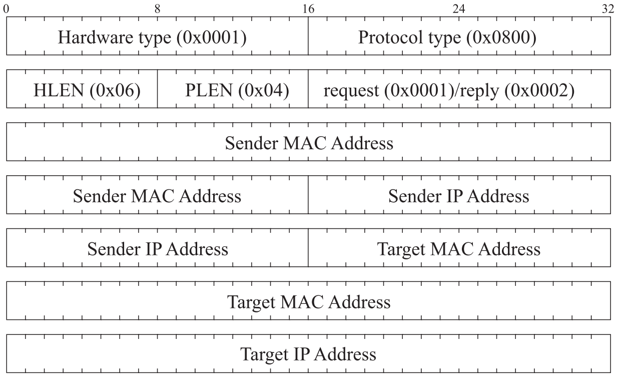

**Fig. 3** Structure of ARP packet.

ter of the stream cipher. In WPA-TKIP, the least significant byte of CRC32 is encrypted. Hence, the attacker calculates CRC32 that chops off some possible values of the least significant byte (0x$FF$ from 0x00); then, she/he sends the falsification packet that attaches CRC32 to the client. If the predicted least significant byte is accurate, the MIC check is executed. Further, the probability that the MIC coincidentally agrees with is very low ($1/2^{64}$). Then, the attacker can determine the correct plaintext value by observing the error message of MIC. However, the MIC key is changed if more than two error messages of MIC are sent back to the sender in less than 1 min. Thus, the Beck-Tews attack requires a standby time of 1 min after 1 byte is restored. Thus, this attack is not effective when the target packet has a large number of unknown bytes. Hence, Beck and Tews focused on the ARP packet. The ARP packet can estimate plaintext with high probability. Figure 3 shows the structure of the ARP packet. Beck and Tews assumed that the IP addresses of the sender and the receiver (excluding respective lowest bytes) can be predicted with high probability. This is an appropriate assumption, because many users use wireless LAN implementations in the default configuration. In this case, 14 bytes (data, MIC, and checksum) are unknown. The attacker executes the chopchop attack 12 times against the ARP packet and restores the MIC and checksum. The IP addresses of the sender and receiver (respective lowest bytes) are restored by comparison with the checksum. By repeating this attack, the attacker can determine almost all bytes of the keystream.

### 3.3 The Method for Breaking the MIC Check

Finally, we explain the method for breaking the MIC check. In WPA-TKIP, the MIC is calculated by using the message integrity check function MICHAEL as follows:

$$\begin{aligned} MIC &= michael(MICKey, DestinationMACAddress,\\ &\quad SourceMACAddress, QoS\,priority, Data). \end{aligned}$$

Beck and Tews used a reverse function of MICHAEL as follows:

$$\begin{aligned} MICKey &= reverse\_michael\\ &\quad (MIC, DestinationMACAddress,\\ &\quad SourceMACAddress, QoS\,priority, Data). \end{aligned}$$

Then, the MIC key is easily restorable from the ARP packet and MIC. Once the attacker has obtained the keystream corresponding to the MIC key and the IV, she/he can counterfeit the encryption packet, whose size is the same as that of the keystream.

## 4. The Reverse Chopchop Attack

In this section, we describe our proposed attack that improves the execution time for recovering the MIC key from an ARP packet. We call this attack the *reverse chopchop attack*. This attack is used to break the checksum check, and it can recover the MIC key within 7–8 min.

### 4.1 Principle of the Reverse Chopchop Attack

The Beck-Tews attack sequentially restores the unknown bytes of the keystream from the lower byte of the packet. However, if all the bytes of the packet, except for CRC32, are already known, the restoration of CRC32 is unnecessary. Then, we apply a technique for restoring the keystream using higher bytes of the packet to WPA-TKIP. This technique for WEP has been proposed in [4].

We next describe the principle of the reverse chopchop attack. We assume that the higher bytes of a packet are known to attacker. Generally, this assumption holds because the higher bytes of the packet of IEEE 802.11 are fixed or guessable values. This attack recovers the next unknown byte of this packet by using the known bytes and ciphertext. The attacker can calculate CRC32 corresponding to 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 encrypted correctly. In addition, the attacker sends falsification packets created using all 256 candidates of the least significant byte. If attacker sends falsification packets created using the correct least significant byte, the receiver passes the checksum check for this packet and execute the MIC check. However this packet cannot pass the MIC check at a high probability. Then receiver sends a MIC failure report frame. The attacker can restore one unknown byte of a keystream by detecting the MIC failure report frame. Figure 4 shows the process of this attack.

The effect of the reverse chopchop attack is explained as follows:

1. This attack can execute an information gathering attack. The reverse chopchop attack can restore the IP address belonging to a local network before restoring CRC32 and MIC. Namely, the attacker can determine the connection between the IP address and the MAC address, because the attacker can know the MAC address from the unencrypted IEEE 802.11 header. This IP address is a significant value, unlike those of the MIC and CRC. Even if the update interval of the MIC

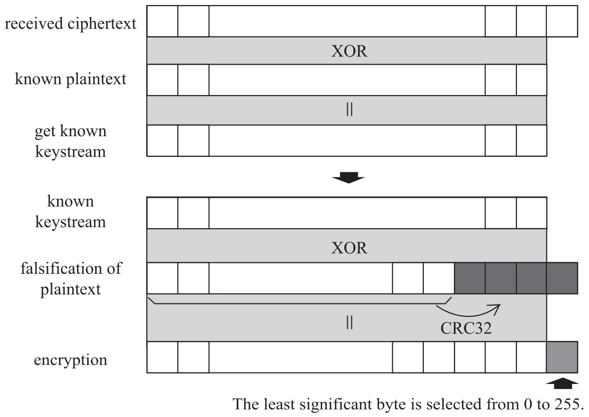

**Fig. 4** Reverse chopchop attack.

key is short in the target network, it is difficult to prevent the information gathering attack because the execution time is only 5 sec, as will be determined in Sect. 4.2.

2. The reverse chopchop attack can reduce the execution time required to restore the MIC key. This attack does not necessarily restore the checksum. Moreover, source and destination IP addresses can be recovered beforehand by using the information gathering attack. Therefore, an effect equivalent to that of the Beck-Tews attack can be achieved by executing eight reverse chopchop attacks; the reverse chopchop attack can recover the MIC key within 7–8 min.
3. The reverse chopchop attack can execute the falsification attack at a high speed after the MIC key is restored. If the attacker knows the MIC key, she/he can recover the MIC and checksum from IP addresses. Therefore, if attacker knows the source and destination IP addresses, the falsification attack can be executed in 1 sec or less[†]. Even if the attacker does not know source and destination IP addresses, she/he can recover the IP addresses by using the information gathering attack.
4. The reverse chopchop attack can falsify a variable-length packet. This attack can also restore a keystream with a length more than that of the keystream used for the chopchop attack. Therefore, this attack can falsify a variable-length packet, but a time of 1 min is required to enhance the keystream by 1 byte.

### 4.2 The Execution Time and Experimental Result

Next, we discuss the execution time required for recovering the MIC key. If an attacker executes the chopchop attack in order to recover the MIC key, then the execution time is expressed as follows:

$$T = 11 + 12 \cdot T_{average} \ \ (\text{min}), \tag{1}$$

where $T_{average}$ is the average execution time for one chopchop attack and “11 min” is standby time for the attack executed 11 times to prevent the AP from shutting down. On the other hand, if an attacker executes the reverse chopchop attack in order to recover the MIC key, then the execution time is expressed as follows:

**Table 2** Experimental result.

| $T_{interval}$ (s) | Success (%) | Failure A (%) | Failure B (%) |
|---|---|---|---|
| 1.75 | 7 | 89 | 4 |
| 2 | 82 | 13 | 5 |
| 5 | 88 | 1 | 11 |
| 10 | 89 | 1 | 10 |
| 15 | 90 | 1 | 9 |
| 20 | 93 | 1 | 6 |

$$T = 7 + 8 \cdot T_{average} \ \ (\text{min}), \tag{2}$$

where $T_{average}$ is the average execution time for one reverse chopchop attack and “7 min” is standby time for the attack executed 7 times to prevent the AP from shutting down. This execution time ($T_{average}$) is similar to that for one chopchop attack. Then, the reverse chopchop attack can recover the MIC key faster than the chopchop attack. Next, we evaluate the execution time for one (reverse) chopchop attack, because this execution time has not yet been evaluated comprehensively.

First, we explain the execution time $T_{average}$. The correct value recovered by one (reverse) chopchop attack appears between 0x00 and 0x$FF$ with the same probability, because the data is encrypted. Therefore, the average time for one (reverse) chopchop attack is

$$T_{average} = \frac{1}{2} \cdot T_{interval}, \tag{3}$$

where $T_{interval}$ is the execution time required to send 256 packets.

Next, we explain the execution time $T_{interval}$. In the (reverse) chopchop attack, the attacker has to observe a MIC failure report frame in a short time. The shorter this time is, the greater the probability of recovering the wrong key is. Then, we experiment with 6 different execution times ($T_{interval}$): 1.75 sec, 2 sec, 5 sec, 10 sec, 15 sec, and 20 sec. We attack the target located at a distance of 10 meters and place a barrier between the targets and us. Namely, we performed the experiment in a realistic environment for attackers. Table 2 lists the experimental results.

Failure A shows the probability of the wrong key being detected, and Failure B shows the probability of the attacker missing the MIC failure report frame. Failure B does not depend on the value of $T_{interval}$. We think that the reason for this abnormality is the noise between the AP and the client. On the other hand, Failure A depends on the value of $T_{interval}$, and the shorter the $T_{interval}$ value, the larger is Failure A. As can be seen from Table 2, the attacker should spend a minimum time of 5 sec to send 256 packets. Of course, this execution time $T_{interval}$ should be changed according to the performance of the target. Then, the attacker can select a suitable execution time $T_{interval}$ by executing

[†] This execution time includes only the time in which the attacker creates and sends a falsification packet.

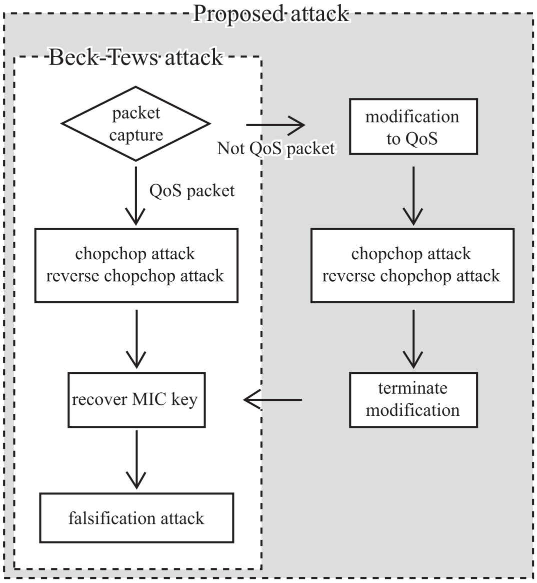

**Fig. 5** QoS forgery attack.

the information gathering attack. If the attacker spends a time of 10 sec to send 256 packets, $T_{average}$ becomes 5 sec by Eq. (3). Therefore, the chopchop attack can recover the MIC key in about 12 min as Eq. (1). In contrast, calculated from the reverse chopchop attack can recover the MIC key in about 7 min 40 sec as calculated from Eq. (2).

## 5. The QoS Forgery Attack

In this section, we describe our proposed falsification attack that is based on the vulnerability of QoS packet processing. We call this attack the *QoS forgery attack*. The QoS forgery attack is used to break the TKIP-IV check.

The Beck-Tews attack can be executed against only a network that supports IEEE 802.11e features. However, IEEE 802.11e can be disabled by setting the AP appropriately, and a client connected to an AP that does not support IEEE 802.11e cannot be attacked. The proposed attack does not depend on whether the network supports IEEE 802.11e, because this attack employs the new vulnerability of the wireless LAN implementation in the processing of the IEEE 802.11e function by the client. Many wireless LAN implementations may have this vulnerability and are prone to be attacked. In the next paragraph, we describe the principle of this attack. Figure 5 shows the flowchart of the entire attack.

### 5.1 Principle of the QoS Forgery Attack

First, an attacker checks the structure of the captured packet. Figure 6 shows the structure of the IEEE 802.11 header of a regular packet, and Fig. 7 shows that of the IEEE 802.11 header of a QoS packet[†]. A comparison of these structures reveals that the values of frame controls of the headers are different. Similar to the Beck-Tews attack, the main target of our proposed attack is an ARP packet; which has the structure of a data frame. The frame control of the QoS ARP packet is "0x88," but that of the regular ARP packet is "0x08." If the captured packet is a regular ARP packet, the attacker modifies it to a QoS ARP packet.

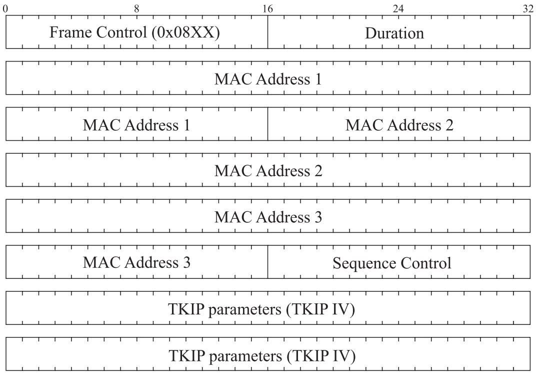

**Fig. 6** Structure of IEEE 802.11 header for regular packet.

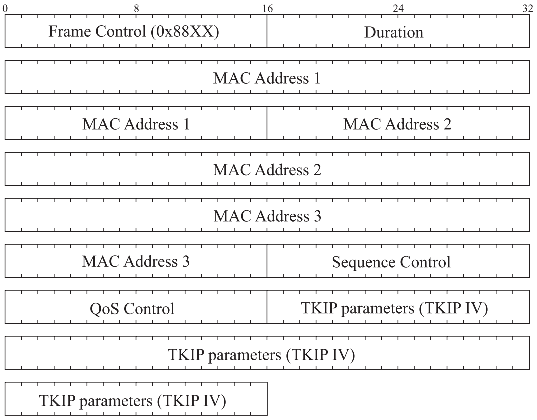

**Fig. 7** Structure of IEEE 802.11 header for QoS packet.

Next, we describe the method for modifying a regular packet to a QoS packet. The comparison between Fig. 6 and Fig. 7 reveals two differences. The first is the most significant byte of the frame control. The attacker rewrites this byte to "0x88." The second is that the QoS packet contains the QoS control field. The priority shown in Sect. 3.1 is inserted in this field. The QoS control field does not exist in a regular packet. Therefore, the attacker inserts a QoS control field in the captured packet. The attacker also sets an appropriate priority and then executes the chopchop attack or the reverse chopchop attack by using the QoS forgery packet and recovers the MIC. We omit the explanation of this process, because the chopchop attack and the reverse chopchop attack have already been described in Sect. 3 and Sect. 4, respectively.

Finally, the attacker recovers the MIC key. After this recovery, the attacker terminates the modification. This is because the MIC is calculated from various data, including priority, and the MIC recovered by the attacker is that

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

**Table 4** Comparison of the Beck-Tews attack and the proposed attack.

| | Access Point | Client | Network |
|---|---|---|---|
| Beck-Tews attack | QoS enabled | QoS enabled | QoS enabled |
| Proposed attack | - | IEEE 802.11e function (chipset) | QoS disabled |

**Table 3** Experimental result.

| Chipset vendor | Release vendor | Type | Release | Result |
|---|---|---|---|---|
| Co.A | Co.a | USB | 2008 | Success |
| Co.A | Co.b | CardBus | 2006 | Success |
| Co.A | Co.c | chipset | 2009 | Success |
| Co.B | Co.d | USB | 2009 | Success |
| Co.B | Co.e | chipset | 2009 | Success |
| Co.B | Co.f | chipset | 2008 | Success |
| Co.C | Co.a | USB | 2007 | Success |
| Co.C | Co.d | CardBus | 2006 | Failure |
| Co.D | Co.g | chipset | 2009 | Success |
| Co.D | Co.h | chipset | 2006 | Success |

of original packet (not QoS packet). Therefore it should be noted that if the attacker recovers the MIC key from the modification packet instead of the original packet, the MIC key is an incorrect key.

The attacker uses the inverse function of Michael after releasing the QoS forgery, and then recovers the MIC key. The attacker can falsify the packet with this MIC key, because the key does not depend on the priority or the packet structure.

### 5.2 Experimental Result

In this section, we evaluate the QoS forgery attack to examine the type of product that can be attacked. We disable the IEEE 802.11e function of the AP. Namely, we evaluate our attack in an environment in which the Beck-Tews attack cannot be executed, because the QoS packet is not sent to the target network. Further, we experiment with three types of clients (USB type, CardBus type, and Chipset with built-in PC) that support the IEEE 802.11e with a chipset. We execute the ARP cache poisoning attack and judge that the proposed attack will succeed if the ARP table is rewritten in the experiment. Table 3 lists the experimental results. Chipset vendor indicates the company that developed the chipset of the target product, whereas, release vendor indicates the company that sells the product. If the client type is chipset, the release vendor is a company that sells computers.

The experimental results reveal that several wireless LAN implementations are potential attack targets. However, we were unable to attack Co.C (CardBus, 2006) because the communication was intercepted by one chopchop attack. However, this implementation is a breach of protocol. Namely, our attack succeeds under the condition that the chipset of the client supports IEEE 802.11e. Many chipsets of clients available in the market in recent years support IEEE 802.11e. Therefore, if other clients have this vulnerability, almost all WPA-TKIP implementations would fail to protect a system against the falsification attack in a realistic environment.

## 6. Consideration

In this section, we compare the Beck-Tews attack with the proposed attack. We also consider a technique for preventing the proposed attack.

First, we compare the Beck-Tews attack with the proposed attack. The target of the Beck-Tews attack is a network that supports IEEE 802.11e. However, the proposed attack can be executed independently of the setting of the AP, and it does not require the QoS packets to be sent on the network. Therefore, the target of the proposed attack can be explained to a client whose chipset corresponds to IEEE 802.11e. Table 4 lists the differences between the Beck-Tews attack and the proposed attack. According to our experiment, we can attack all clients that support IEEE 802.11e with a chipset. Further, as mentioned above, many clients introduced in the market in recent years have the IEEE 802.11e function for the chipset. In other words, if other clients have this vulnerability, almost all implementations of WPA-TKIP can be attacked.

Next, we consider a technique for preventing the proposed attack. First, vendors should immediately take steps to overcome this vulnerability. If vendors implement a client that discards the QoS packet when it does not use IEEE 802.11e, the attacker cannot use our proposed attack. However, we should consider the technique for preventing this attack until the vulnerability is overcome. Then, we strongly recommend the shift to WPA2-AES [14]. However, we consider another technique for preventing the proposed attack; it involves reducing the key update interval for preventing the Beck-Tews attack [8], [14]. Note that the technique cannot prevent the information gathering attack proposed in Sect. 4.1. The information gathering attack can obtain a relation between an IP address and a MAC address. Although this relation cannot cause a realistic damage compared with the MIC key, it may give the attacker some information that can be used for another attack. For example, in Sect. 4.1, we abuse this relation to reduce the execution time required to restore the MIC key.

## 7. Conclusion

In this paper, we proposed two new falsification attacks. First, we proposed the reverse chopchop attack, which is an improvement over the chopchop attack and can recover a MIC key within 7–8 min. Next, we proposed the QoS forgery attack, which is a falsification attack based on the vulnerability of QoS packet processing. In this attack, the condition for the Beck-Tews attack that the AP supports IEEE 802.11e is negated. In addition, we discovered another vulnerability by which our attack succeeds under the condition that the chipset of the client supports IEEE 802.11e even if the client disables this standard through the OS. In other words, if other clients have this vulnerability, almost all implementations of WPA-TKIP can be attacked. Therefore, WPA-TKIP is not secure in a realistic environment.

## Acknowledgements

This work was supported in part by Grant-in-Aid for Scientific Research (KAKENHI 21700018, 23560455) of JSPS. We thank anonymous reviewers and Associate Editor for their helpful comments.

### References

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

[2] R. Teramura, Y. Asakura, T. Ohigashi, H. Kuwakado, and M. Morii, "Fast WEP-key recovery attack using only encrypted IP packets," IEICE Trans. Fundamentals, vol.E93-A, no.1, pp.164–171, Jan. 2010.

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

[4] W.A. Arbaugh, "An inductive chosen plaintext attack against WEP/WEP2," available at http://www.cs.umd.edu/~waa/attack/frame.htm

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

[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] 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.

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

[12] IEEE Std 802.1X-2001, "Port-based network access control," IEEE, July 2001.

[13] Wi-Fi Alliance, "Wi-Fi CERTIFIED$^{\text{TM}}$ for WMM$^{\text{TM}}$-support for multimedia applications with quality of service in Wi-Fi$^{\circledR}$ networks," available at http://www.wi-fi.org/files/wp_1_WMM%20QoS%20In%20Wi-Fi_9-1-04.pdf

[14] Y. Oiwa, K. Kobara, R. Yamaguchi, G. Hanaoka, and H. Watanabe, "TN2009-01 (B)," RCIS Technical Notices 2009-01, Aug. 2009, available at http://www.rcis.aist.go.jp/TR/TN2009-01/wpa-compromise.html (in Japanese).

**Yosuke Todo** received the B.E. degree from Kobe University, Japan, in 2010. Since 2010, he has been a master's student in Graduate School of Engineering, Kobe University. His current research interests are in information security and cryptography.

**Yuki Ozawa** received the B.E. degree from Kobe University, Japan, in 2010. Since 2010, he has been a master's student in Graduate School of Engineering, Kobe University. His current research interest is in information security.

**Toshihiro Ohigashi** received the B.E. and M.E. degrees from the University of Tokushima, Japan, and the Ph.D. degree from Kobe University in 2002, 2004, and 2008, respectively. Since 2008, he has been an Assistant Professor in Information Media Center, Hiroshima University. His current research interests include cryptography, information security, and network protocol. He received the SCIS 20th Anniversary Award from ISEC group of IEICE in 2003. He is a member of the Information Processing Society of Japan.

**Masakatu Morii** received the B.E. degree in electrical engineering and the M.E. degree in electronics engineering from Saga University, Saga, Japan, and the D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor in the Department of Electronics and Information Science, Kyoto Institute of Technology, Japan. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering at Ehime University, Japan. From 1995 to 2005 he was a Professor at the Department of Intelligent Systems and Information Science, Faculty of Engineering at the University of Tokushima, Japan. Since 2005, he has been a Professor at the Department of Electrical and Electronics Engineering, Faculty of Engineering at Kobe University, Japan. His research interests are in error correcting codes, cryptography, discrete mathematics and computer networks and information security. He is a member of the IEEE.