Authentication – Based Medium Access Control to Prevent Protocol Jamming: A-MAC

Jaemin Jeung, Seungmyeong Jeong, and Jaesung Lim*

Graduate School of Information and Communications, Ajou University, South Korea
{mnsg,aflight,jaslim}@ajou.ac.kr

Abstract. Recently, Wireless Local Area Network (WLAN) is used by enterprises, government, and the military, as well as small office and home offices. Although it is convenient, it has inherent security weaknesses due to wireless characteristics. For this reason, security-sensitive groups are still unwilling to use WLAN. There are several types of attacks to degrade the wireless network throughput using security weakness, especially Protocol Jamming Attacks are critical. These attacks consume little energy and can be easily implemented. In this paper, we introduce Authentication-based Medium Access Control (A-MAC) to prevent Virtual Carrier Sense (VCS) Jamming Attack and De-authentication / Disassociation Jamming Attack that are typical Protocol Jamming Attacks in 802.11 based wireless systems. The proposed scheme can authenticate frames using Universal Hashing Message Authentication Codes (UMAC-32) and Hidden Sequence Number (SN). A-MAC frequently changes the key and SN using shift row and shift column processes to overcome the weakness of short 32 bits hashing codes. A-MAC can achieve integrity, authentication, and anti-replay attack security features. It can prevent Protocol Jamming Attacks that degrade wireless network throughput. Our simulation shows A-MAC can sustain throughput under Protocol Jamming Attacks.

Keyword: Protocol Jamming, Authentication, Medium Access Control.

1 Introduction

Wireless Local Area Network (WLAN) provides users with mobility within a broad coverage area with connection to networks. In addition, it is cost-efficient, allowing ease of integration with other networks and network components. For these reasons, WLAN is used by enterprises, government, and the military, as well as small office and home offices. However, it has inherent security weaknesses due to wireless characteristics. Furthermore, it uses the Industrial Scientific Medical (ISM) band to be highly susceptible to interference. Security-sensitive groups are still unwilling to use WLAN due to these weaknesses. If we use open source scanning software, such as

* This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-C1090-0902-0003).

D. Taniar et al. (Eds.): ICCSA 2010, Part III, LNCS 6018, pp. 72–84, 2010.

KISMET, we can easily analyze the packets [1]. Many researchers have been studying these issues, since these weaknesses are well known. Thus, WEP, WPA, TKIP, 802.11i standards were formed. However, the standards cannot resolve Protocol Jamming Attacks. Furthermore, Protocol Jamming Attacks consume little energy, and can be easily implemented. An adversary can decrease the wireless networks throughput for a long time, especially, when jammers are hidden.

Virtual Carrier Sense (VCS) jamming and De-authentication / Disassociation jamming are typical Protocol Jamming Attacks. VCS jamming attacks exploit a weakness in the Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA). CSMA/CA systems use VCS to avoid collision. An adversary exploits the VCS process to decrease the wireless networks throughput. De-authentication / Disassociation jamming attacks exploit the protocol concerned with connection in the mobile station and AP. All stations have to go through the process of authentication / association before sending data. If an AP sends a De-authentication / Disassociation frame to a station, the stations have to terminate the connections. Unfortunately, based on the 802.11 standards, the station cannot reject the notification of these frames. Therefore, these attacks are more effective and powerful than VCS attacks [2].

We consider that the cause of Protocol Jamming Attacks is authentication problems. That is, APs and mobile stations cannot authenticate RTS / CTS / De-authentication / Disassociation frames. Even if there is an authentication process, it can be attacked by replay-attacks. Replay attacks also decrease network throughput. Thus, we need authentication, integrity and anti-replay-attacks features to prevent Protocol Jamming Attacks. In this paper, the proposed A-MAC substitutes digested message for Cyclic Redundancy Check (CRC). The digested message is created when the Universal Hashing Message Authentication Codes (UMAC-32) are XORed with a Hidden Sequence Number. UMAC-32 ensures the authenticity and the integrity of transmitted messages. In addition, it can check the transmission error. So we can substitute the digested message for CRC. The Hidden Sequence Number for anti-replay attacks is not seen in the transmitted frames. It is stored in a Sequence Table of each correspondent. In this approach, we can achieve integrity, authentication and anti-replay attacks security features, and prevent Protocol Jamming Attacks that degrade network throughput, as well as checking for transmission errors, without frame size overhead. We also introduce Key Shift & Sequence Number Select Notification (KSSN) that notify key and SN change by shift row and shift column processes, so that we compensate for the short 32bits digested message. Using the KSSN, each correspondent can change the key and SN, if necessary. That is, the KSSN is verified by the A-MAC, each correspondent changes the key and SN. This method is suitable for control and management frames that are short frame and occur frequently, because the KSSN can use a one-way key exchange mechanism, instead of four-way handshaking. Thus, we can prevent VCS and De-authentication / Disassociation jamming attacks using the A-MAC. Our simulation shows its performance and effectiveness.

The remainder of this paper is organized as follows. Section 2 describes the background of the proposed scheme and related work. We introduce adversary models, A-MAC and KSSN process in Section 3. Security analysis and performance analysis are presented in Section 4. Finally we conclude the paper and outline directions for future work in Section 5.

2 Background and Related Work

2.1 Background

802.11 wireless networks use Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The Distributed Coordination Function (DCF) is the basis of the standard CSMA/CA access mechanism to avoid collisions. It first checks to see that the radio link is clear before transmitting. Stations use a random back-off after each frame, with the first sender seizing the channel, to avoid collisions. The DCF may use the Request To Send (RTS) / Clear To Send (CTS) clearing technique to further reduce the possibility of collisions.

Carrier sensing is used to determine if the medium is available. Two types of carrier sensing functions in 802.11 manage this process: Physical Carrier Sensing and Virtual Carrier Sensing. If either carrier sensing function indicates that the medium is busy, the MAC reports this to higher layers [3]. Unlike wired communications, Physical Carrier Sensing cannot provide all the necessary information, due to the hidden node problem. In Fig. 1, A is about to send frames to B. However, D cannot recognize the situations, because A's physical signal cannot reach D. Since A and D cannot hear each other, they may sense the media is free at the same time and both try to transmit simultaneously; this causes collision in the network. This is the hidden node problem. To reduce this problem, 802.11 standards adopt RTS, CTS, and VCS mechanisms. Fig. 1 shows these mechanisms. Activity on the medium by stations is represented by the shaded bars, and each bar is labeled with the frame type. Interframe spacing is depicted by the lack of any activity. DIFS is Distributed Inter-Frame Space and SIFS is Short Inter-Frame Space. If A wants to send frames to B, A sends the RTS frame to B, and then B replies with the CTS frame. Neighbor stations, C and D can overhear RTS and CTS frames, and then defer access to the medium until the

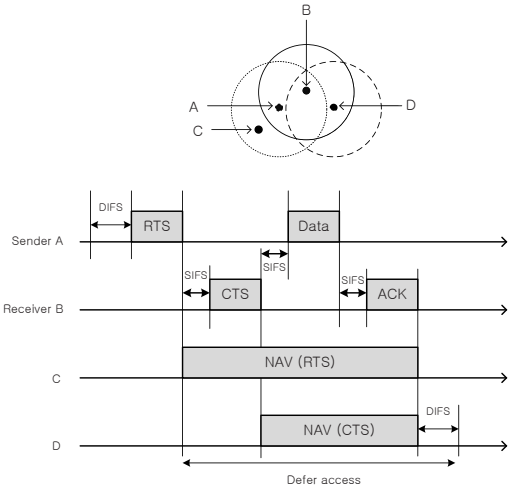


Fig. 1. Virtual Carrier Sense

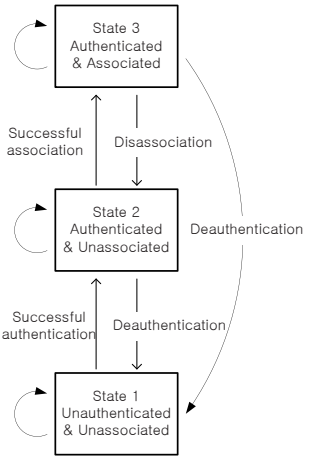


Fig. 2. 802.11 state diagram

Network Allocation Vector (NAV) elapses. VCS is provided by the NAV. The NAV is a timer that indicates the amount of time the medium will be reserved, in microseconds. It is usually presented by 2 bytes. If a station sets the timer for which they expect to use the medium, other stations count down from the NAV to 0. When the NAV reaches 0, the VCS indicates that the medium is idle. These RTS, CTS, and VCS mechanisms will prevent collisions.

Stations have to go through three states for connection to Access Point (AP). Fig. 2 shows the state diagram of 802.11. Stations are either authenticated or unauthenticated and can be associated or unassociated. These situations can be combined into three states. State 1 is not authenticated and not associated, State 2 is authenticated but not associated, and State 3 is authenticated and associated. Each state is a successively higher point in the development of an 802.11 connection. All stations start in State 1, and data can be transmitted only in State 3. That is, any stations cannot send data in Sate 1 and State 2. When a station transfers from one state to another state, it would be vulnerable.

2.2 Related Work

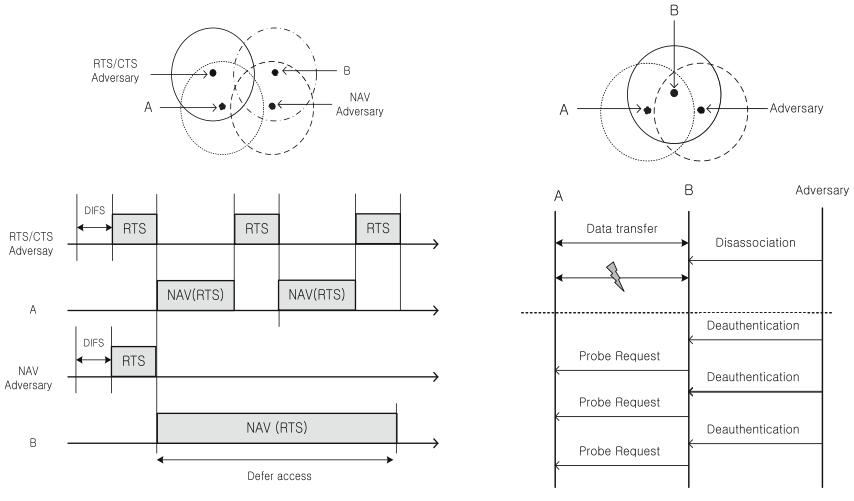
Any current or upcoming 802.11 standards would not help mitigate the risk of Protocol Jamming [4]. Researchers have tried to resolve these problems recently. Zhou proposed a packet-by-packet authentication method [5]. The author encrypted all frame content by a secret key, and then attached the encrypted content to the end of the original frame. For example; A → B: {RTS, $E(sID, dID, TS, SN)_k$}, B → A: {CTS, $E(sID, dID, TS, SN+1)_k$}. User A decrypts the encrypted attachment, verifies sID, dID, TS, and SN+1. If the frame is correctly decrypted, A can commence data transmission. However, this authentication scheme increases transmission overhead. Assume TS and SN are 4 bytes each, the total encrypted attachment is 20 bytes (6 bytes each for sID and dID). Considering RTS / CTS frame are just 20 bytes / 16 bytes, the encrypted attachment is a considerable overhead. Karlof proposed the TinyAuth packet format for wireless sensor networks [6]. However, it has a weakness in a replay attack, because it does not use any kind of sequence number or time-stamp in the TinySec-Auth packet. So, TinySec is not suitable in WLAN. Bellardo has a different view point [7]. In contrast to the authentication scheme, this scheme places two different limits on the duration values accepted by stations. The low limit has a value equal to the amount of time required to send an ACK frame, plus media access back-off for that frame. The high limit has a value equal to the amount of time required to send the largest data frame, plus the media access back-off for that frame. However, this scheme still has a weakness in RTS/CTS flooding. Since each station cannot verify the adversary that frequently send false RTS/CTS frames, the stations defer their transmission, and consequently the wireless networks throughput is drastically lowered. Some research prevents Protocol Jamming Attacks using the creation of a series of protocol extension and replacements (e.g., WEP, WPA, 802.11i, 802.11w) [2]. However, these schemes need complicated key management, powerful computational process and frame size overhead. They are not easy to implement.

3 Authentication-Based Medium Access Control (A-MAC)

3.1 Malicious Adversary Models

Malicious adversary models are VCS jamming attacks and De-authentication / Disassociation jamming attacks. VCS jamming attacks exploit the weakness of the virtual carrier sensing mechanism, especially RTS/CTS process. De-authentication / Disassociation jamming attacks exploit the weakness of the state diagram in Fig. 2. When a station transfers from one state to another state, a vulnerable point occurs, especially in the De-authentication, Disassociation frames.

Fig. 3(a) shows RTS/CTS adversary and NAV adversary in VCS jamming attacks. If a malicious adversary frequently sends a RTS frame to A, A sets the NAV timer, as much as the duration field of RTS frame indicates. According to the VCS mechanism, A has to defer access to the medium until the NAV elapses, since the VCS mechanism indicates that the medium is busy. If a malicious adversary repeatedly sends a RTS frame to A, station A cannot have the opportunity to access the medium. This scenario will decrease the total wireless network throughput. We define this adversary as the RTS/CTS Adversary. In the case of Station B, a malicious adversary sets the NAV timer for a longer period. According to VCS mechanism, B simply thinks that the other station is about to send a large data set. As a result, B has to defer access to the medium during the time of the duration field. The duration field has 16 bits. An adversary can set the NAV duration for a 2^{16} time slot. Assume that one NAV time slot is 1 μs; B has to wait approximately 0.06 second. This delay is considerable. If there are many stations around the malicious adversary, the total wireless network throughput is drastically lowered. We define this adversary as the NAV adversary. These two adversary models apply to the CTS transmission process too.



(a) RTS/CTS, NAV adversary (b) Dis-ass, De-auth adversary

Fig. 3. Malicious adversary models

Fig. 3(b) shows Disassociation/De-authentication jamming attacks. These attacks exploit the weakness of the connection process. We have already learned that each station goes through States 1, 2, and 3. If a station gets a de-authentication frame when transferring from state 1 to State 2, the station cannot transfer to State 2. For the same reason, if a station gets a disassociation frame when transferring from State 2 to State 3 or sending data to the AP, the station reverts to State 2. According to the 802.11 standards, stations cannot reject the notification of disassociation and de-authentication frames. Unfortunately, a malicious adversary can easily fake disassociation and de-authentication frames. In Fig. 3(b), assume that station B is connected to Access Point A. If B gets a disassociation frame, B has to terminate the data communication. We define this attacker as the Disassociation Adversary. The adversary model is simple but, powerful. The total wireless throughput is to be a zero by only a few frames / second. After terminating the connection, B tries to reestablish a new connection. If a malicious adversary frequently sends a de-authentication frame at that time, B cannot transfer to State 3. We define this attacker as the De-authentication Adversary.

We define four adversary models. The characteristics of these models are easy implementation, use low energy, and decrease the total wireless network throughput. It is not easy for wireless IDS to detect the jammer, given that the jammer is hidden, uses low signal power, and sends fake frames not frequently but appropriately. Thus, the jammer can sustain a low network throughput that irritates network users and managers for a long time.

3.2 A-MAC Scheme

We propose the Authentication-based Medium Access Control (A-MAC) that achieves authentication, integrity and anti-replay attack without overhead to prevent Protocol Jamming Attacks that degrade throughput. A-MAC compresses the frames into a 32 bits digested message using Universal Hashing Message Authentication Codes (UMAC-32). Then, the digested message is masked by XOR with Hidden Sequence Numbers. Finally, we substitute the results for Cyclic Redundancy Check (CRC). In this scheme, we easily detect some abnormal frames that considerably decrease throughput. Thus, the total wireless network throughput does not decrease by discarding the instruction of abnormal frames. Fig. 4 details the proposed A-MAC process.

In Fig. 4, we assume that sender A transfers a message (M) to receiver B, and M is RTS, CTS, Disassociation, or De-authentication frame without CRC. A-MAC compresses the M into a 32 bits digested message by the UMAC-32 function and the secret key. Then, the compressed message is XORed with the Hidden Sequence Number. The Hidden SN is not seen in the transmitted frames. It is stored in a Sequence Table. The Hidden SN is very important in this scheme. It prevents a replay attack and complicates the digested message. Finally, the XORed message {UMAC (K, M) $\oplus$ SN} is concatenated with M. Sender A sends this frame to receiver B. B also compresses the received M into a 32 bits digested message by the UMAC-32 function and the secret key. Then, the compressed message is XORed with the received {UMAC (K, M) $\oplus$ SN}. Receiver B derives SN from this process [UMAC (K, M) $\oplus$ {UMAC (K, M) $\oplus$ SN} = SN]. The derived SN is compared with SN of B's Sequence Table. If

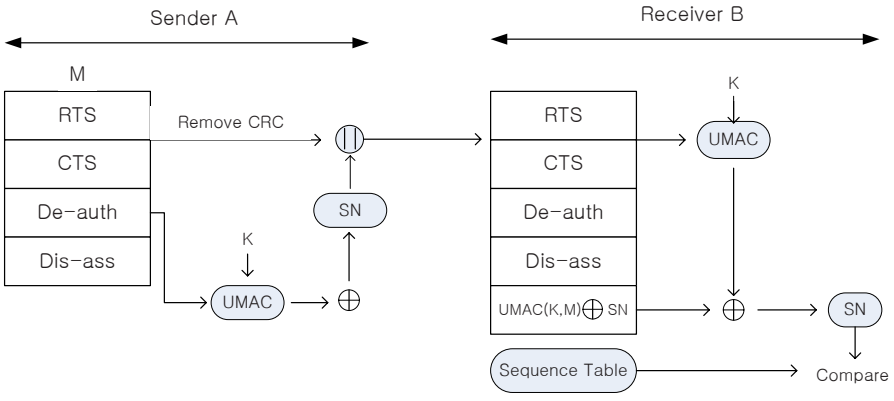


Fig. 4. Authentication-based Medium Access Control Process

the SNs agree with each other, the sender A is proved to be a legitimate station. If the SN does not agree with each other, there are either transmission errors or Protocol Jamming Attacks. Transmission errors are closely related to Signal to Noise Ratio (SNR). We can easily check the SNR in 802.11 systems by checking the Received Signal Strength Indication (RSSI). Therefore, although the RSSI is good, the SN does not agree with each other for several times. The probability of Protocol Jamming Attacks becomes high.

We can apply the A-MAC to an infrastructure WLAN. For example, an adversary sends de-authentication and disassociation frames to a certain station, the station can verify the frames using the A-MAC mechanism. If those frames are verified, they follow the instruction of each frame, otherwise they discard the frames. In case of sending lots of false RTS / CTS frames to a certain station, counter measures will be different. That is, key distribution schemes can make the method different. For example, an AP and stations share the same secret key, the stations can derive the SN from the frames using A-MAC. If the SNs of successive frames are incremented by one, the station will be legitimate, otherwise the stations will be suspected as an adversary. In contrast to the prior method, if the AP and each station share the 1:1 matching secret key, neighbor stations cannot derive the SN due to a different key. But the AP can derive the SN, and verify the frames. In that case, suppose the AP recognizes the adversary, it can notify the protocol jamming situation to the stations. The stations received the notification can discard the frames and cope with the situation.

A-MAC uses UMAC-32 for the digested message, instead of CRC, since UMAC-32 ensures the authenticity and the integrity of transmitted messages. Furthermore, it is the fastest message authentication code reported in cryptographic literature [8]. We can derive Tag from the UMAC-32 process. Tag's size is 32 bits, 64 bits, 96 bits, 128 bits. In this paper, A-MAC use a 32 bits Tag so that the digested message is XORed with a 32 bits Hidden SN. Tag = H_K1 (M) $\oplus$ F_K1 (nonce), M is an input message, H is a secret hash function, F is a pseudo random function, and K1 is secret random key shared by sender and receiver. Nonce is a value that changes with each digested message. In this paper, M is the content of the CRC removed frame, nonce is the Hidden SN. The proposed scheme can achieve authentication, integrity and anti-replay

attack without overhead. Therefore, A-MAC can overcome the weakness of Karlof's scheme [6] and Bellardo's scheme [7]. Especially, it does not expand the original frame size. This characteristic also overcomes the problem of Zhou's scheme [5].

3.3 Key Shift and SN Select

As using a short 32 bits digested message, A-MAC may be attacked by brute-force attack that is a technique to defeat the authentication mechanism by trying successively all the words in an exhaustive list. We propose two methods that complicate the attached 32 bits message of A-MAC to mitigate this vulnerability.

At first, A-MAC does not count the initial SN from zero. If the initial SN starts from zero, a malicious adversary can calculate {UMAC (K, M)}. Thus, the probability of detecting the key becomes high. A-MAC derives the initial SN from the 128 bits key to mitigate this probability. Table 1**.** is an example of Sequence Table. We make a 4 × 4 byte matrix using the 128 bits secret key. Each row and each column is 32 bits. The first row becomes the initial SN of the RTS frame, the second row becomes the initial SN of the CTS frame, the third row becomes the initial SN of the deauthentication frame, the fourth row becomes the initial SN of the disassociation frame. After being set, the initial SN is incremented by one every frame exchange. In this scheme, it is not an easy for a malicious adversary to estimate the key and the digested message.

Second, each correspondent (station or AP) can change the secret key and SN if necessary. We define this process as Key Shift & SN Select Notification (KSSN). Fig. 5 shows that each correspondent changes the secret key and SN simultaneously.

Table 1. Sequence Table

Sequence Table					
Node ID	Key	RTS	CTS	De–auth	Dis–ass
A	ABCD EFGH I JK L MNOP	ABCD	EFGH	I JK L	MNOP
B	BCDE FGH I JKLM NOPA	BCDE	FGH I	JKLM	NOPA
* *	* *	* *	* *	* *	* *

In Fig. 5, we assume that sender A wants to change the secret key and SN. A sets the subtype of Frame Control Header to 0111. In 802.11 standards, management subtype 0111 and control subtype 0111 are reserved and not currently used [3]. In this paper, we use this subtype 0111 to send the KSSN message. After setting the subtype, sender A transmits {UMAC (K, M) ⊕ SN ⊕ KSSN} to receiver B. The received {UMAC (K, M) ⊕ SN ⊕ KSSN} message is XORed with {UMAC (K, M) ⊕ SN} by receiver B. B derives the KSSN from this process. B compares the last 13 bits of KSSN with the last 13 bits of SN.

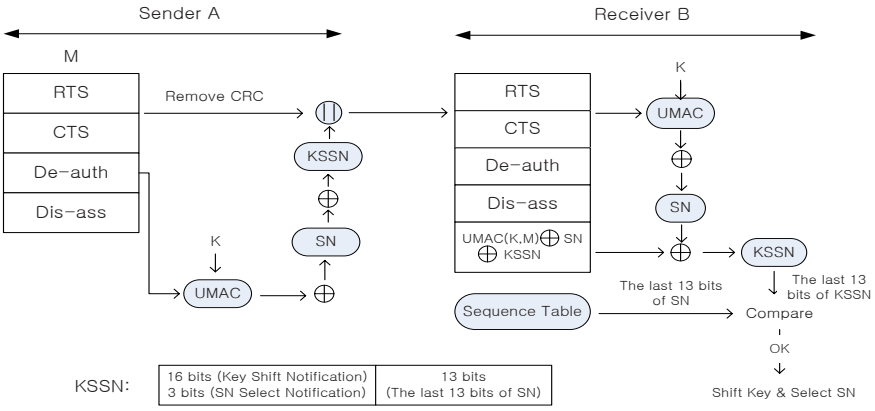


Fig. 5. Key Shift & SN Select Notification Process

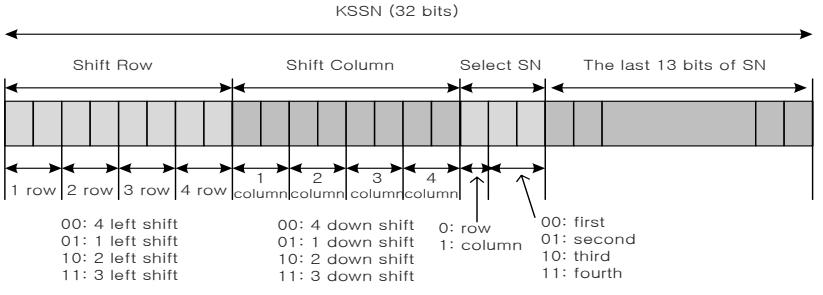


Fig. 6. KSSN format

Fig. 6 shows the 16 bits Key Shift Notification, 3 bits SN Select Notification, and the last 13 bits SN KSSN format. We form the 4 × 4 byte matrix using the 128 bits secret key, as in Fig. 7. The front 8 bits are related to Shift Row. The first 2 bits indicate the first row, 3 ~ 4 bits indicate the second row, 5 ~ 6 bits indicate the third row, 7 ~ 8 bits indicate the fourth row. 00 indicates four bytes left shift, 01 indicates one byte left shift, 10 indicates two bytes left shift, 11 indicates three bytes left shift. The 9 ~ 16 bits are related to the Shift Column. The allocation method is similar to Shift Row. The 17 ~ 19 bits indicate which row or column is selected as the new SN. If the 17th bit is 0, it selects the new SN from the row. If it is 1, it selects the new SN from the column. 00 indicates the first row or column, 01 indicates the second row or column, 10 indicates the third row or column, 11 indicates the fourth row or column. The last 13 bits are used for authentication. If the last 13 bits of KSSN and the last 13 bits of SN are verified, the receiver can change the secret key and SN. The sender transmits the KSSN several times to overcome transmission error. This depends on channel states.

Fig. 7 shows the Key Shift and SN Select operations. If the receiver derives the KSSN (KSSN: 00011011 00011011 010 + 13 bits) from the received message, the receiver verifies the last 13 bits of KSSN comparing them to the last 13 bits of SN.

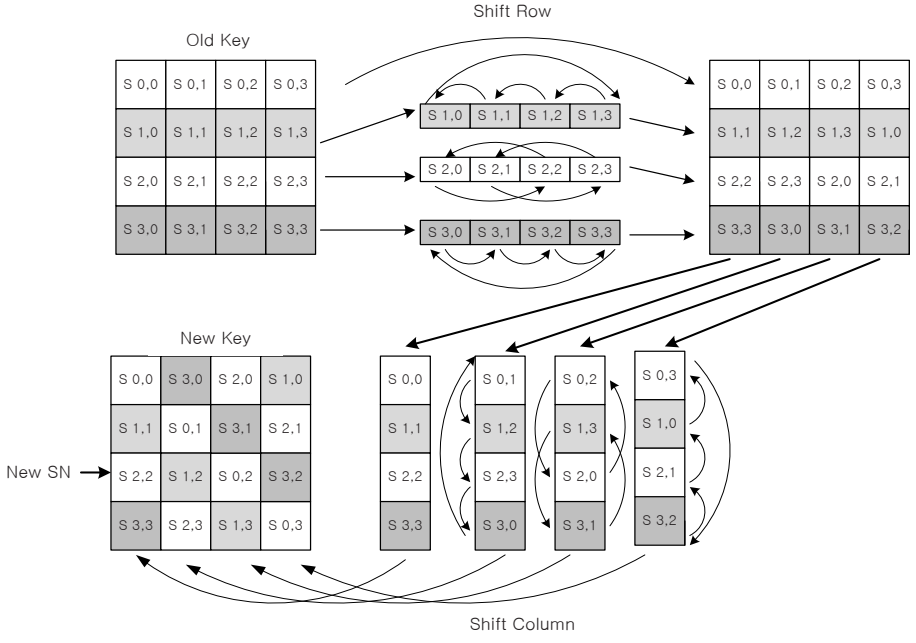


Fig. 7. Key Shift & SN Select Operations

If they are verified, the receiver changes the secret key and SN. The first 8 bits (00011011) indicate the first row is shifted by 4 bytes, the second row is shifted by 1 byte, the third row is shifted by 2 bytes, and the fourth row is shifted by 3 bytes. The next 8 bits (00011011) indicate that the first column is down-shifted by 4, the second column is down-shifted by 1 byte, the third column is down-shifted by 2 bytes, and the fourth column is down-shifted by 3bytes. As a result of the operations, the new key is generated. 17~19 bits (010) indicates that the third row is the new SN of the RTS frame. Then, other rows are sequentially allocated to other frames (e.g. the fourth row is the new SN of CTS frame, the first row is the new SN of de-authentication frame, and the second row is the new SN of association frame).

4 Security and Performance Analysis

4.1 Security Analysis

We designed the A-MAC to achieve authentication, integrity and anti-replay attack without overhead. The A-MAC compresses the frames into a 32 bits digested message using UMAC-32, and then the digested message is masked by XOR with Hidden Sequence Numbers. We obscure the digested message using KSSN operations to overcome the weakness of the short digested message (e.g. brute-force attack). In this process, we can obtain some security advantages. First, to prevent replay-attacks, we use a Hidden SN that makes it hard to estimate the next frame. If it were not for the

Hidden SN, a malicious adversary could perform Denial of Service (DoS) attacks. Assuming that A transmits a RTS control frame to B. A → B: {FC (Frame Control Header), D (Duration), RA (Receiver Address), TA (Transmitter Address), SN, CRC}. An adversary can attempt to predict the SN, and then frequently sends a RTS control frame like this. Adversary → B: {FC, D, RA, TA, SN + (i = i+1), CRC}. B will frequently send CTS or ACK frame. As a result, the total wireless network throughput will be lowered. However, the A-MAC can prevent these DoS attacks using the Hidden SN. Furthermore, is uses the 4 byte initial SN from the 128 bits secret key, instead of zero. This makes it more difficult to estimate the SN. Second, A-MAC can change the secret key and SN using the KSSN. This can prevent brute-force attacks. A malicious adversary can try to break the digested message with the probability of a 1 over 2^{32}. Assuming that an adversary repeatedly sends a forged frame, he could be accepted by AP after about 2^{31}. On 11Mbps 802.11b, an adversary can send about 27,000 forgery attempts per second (where RTS = 15 μs, SIFS = 10 μs) without considering transmission delay. Therefore, it would take over 44 hours. However, the A-MAC can change the secret key and SN frequently in a short time without a complex key exchange mechanism. In addition, A-MAC does not have any frame size overhead. These characteristics can overcome the weakness of Zhou's scheme [5] and Bellardo's scheme [7]. Therefore, the proposed A-MAC is useful for government, the military, and enterprises.

4.2 Performance Analysis

We analyze the performance and effectiveness of the proposed A-MAC. In the simulation environment, we do not consider complex key distribution systems (e.g. Remote Authentication Dial-In User Service: RADIUS), because the key distribution process does not affect our simulation, the secret keys are presumed to have been delivered to the AP and stations. The jammer and AP send RTS, CTS, de-authentication, and disassociation frames. When stations receive the frames, they process the A-MAC operation. If those frames are verified, they follow the instruction of each frame, otherwise they discard the frames.

Fig. 8 shows the simulation topology. There are one AP, four stations and one jammer. Especially, the jammer is hidden and near the AP. All stations can hear the RTS, CTS, de-authentication, and disassociation frame of the AP and Jammer. The jammer starts to attack at 6 seconds. Under transmission errors, although the station is a legitimate user, it cannot access the medium. In the real world, we can check the RSSI level and adjust the threshold. However, we just set the threshold value to one in this simulation. That is, we do not consider transmission error. Fig. 9 shows the throughput under the RTS/CTS Adversary model. The jammer transmits the faked RTS frame at the rate of 100, 200 Frames Per Second (FPS). As the faked RTS frame increases, the throughput lowers. Under a rate of 200 FPS, the throughput falls to 5.25 Mbps. However, we can sustain the throughput under the A-MAC. Fig. 10 shows the throughput under the NAV Adversary model. The Jammer sets the duration field to the length of 2^{11}, 2^{12}. This attack is more serious than the previous attack. Under a length of 2^{12}, the total throughput drastically falls to 50%, since the NAV at 2^{12} ($2^{12} \times 1\mu s = 0.004$ sec) has a considerable delay in 802.11 systems. However, we can also sustain the throughput under the A-MAC. Fig. 11 shows the throughput under the

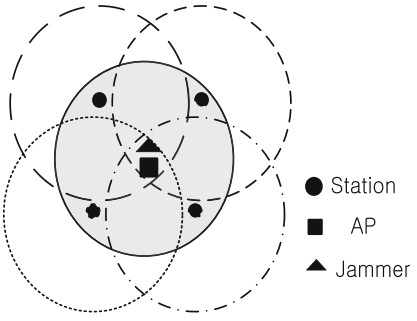


Fig. 8. Simulation Topology

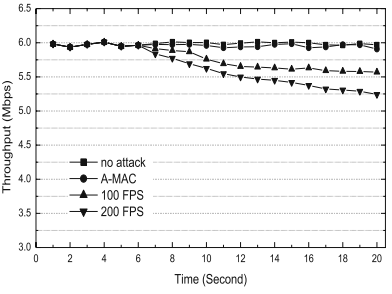


Fig. 9. RTS/CTS Adversary

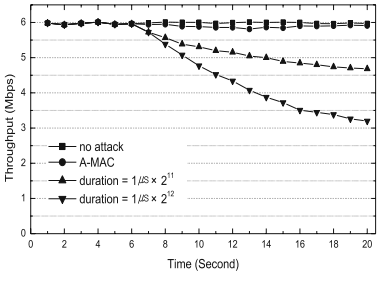


Fig. 10. NAV Adversary

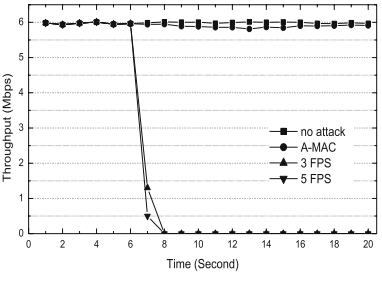


Fig. 11. De-auth / Dis-ass Adversary

De-authentication and Disassociation Adversary. The simulation results of the De-authentication Adversary model are similar to the results of the Disassociation Adversary model. This is due to the de-authentication frame-received station also terminating the connection during data transmission. So, we just show the results of the De-authentication Adversary model. The jammer transmits the faked frames at rates of 3, 5 FPS. These attacks are the most powerful and efficient attacks. Just a few frames can decrease the throughput to almost zero, since all stations undergo the state transition to State 1 or State 2, as soon as they get the frames. In the four cases, the A-MAC can prevent the attacks.

5 Conclusions

In this paper, we introduced the A-MAC to prevent VCS Attacks and De-authentication / Disassociation Attacks that are typical Protocol Jamming Attacks. A-MAC can authenticate frames using UMAC-32 and Hidden SN. A-MAC frequently changes the secret key and SN by using shift row and shift column operations to prevent the weakness of short 32 bits hashing codes. Therefore, A-MAC can achieve integrity, authentication, and anti-replay-attacks security features, and can prevent Protocol Jamming Attacks that degrade wireless network throughput. Our simulation

shows that A-MAC can sustain the network throughput under Protocol Jamming Attacks. In the near future, we will study a physical layer anti-jamming scheme in 802.11 based systems. If we mitigate Protocol Jamming Attacks and physical layer attacks in 802.11 based systems, security-sensitive groups (e.g. government, enterprises, and the military) have a preference to use WLAN. We hope that our research will help that situation come true.

References

1. http://www.kismetwireless.net
2. Liu, C., Yu, J.: Rogue Access Point Based DoS attacks against 802.11 WLANs. In: The Fourth AICT (2008)
3. Gast, M.S.: 802.11 Wireless Networks: The Definitive Guide, pp. 33–66. O'Reilly Publisher, Sebastopol (2002)
4. Malekzadeh, M.: Empirical Analysis of Virtual Carrier Sense Flooding Attacks over Wireless Local Area Network. Journal of Computer Science (2009)
5. Zhou, Y., Wu, D., Nettles, S.: Analyzing and Preventing MAC-Layer Denial of Service Attacks for Stock 802.11 Systems. In: Workshop on BWSA, Broadnets (2004)
6. Karlof, C., Sastry, N., Wagner, D.: TinySec: A link Layer Security Architecture for Wireless Sensor Networks. In: SenSys (2004)
7. Bellardo, J., Savage, S.: 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions. In: USENIX Security Symposium (2003)
8. Krovetz, T.: RFC4418-UMAC: Message Authentication Code using Universal Hashing. In: IEEE Network Working Group (March 2006)