情報保護學會論文誌
第15卷 第4號, 2005. 8

IEEE 802.11에서의 접근 제어를 위한 Lightweight 패킷 인증*

이 근 순,†‡ 김 효 진, 송 주 석
연세대학교 컴퓨터과학과

Lightweight Packet Authentication for Access Control in IEEE 802.11*

KeunSoon Lee,†‡ HyoJin Kim, JooSeok Song
Dept. of Computer Science, Yonsei Univ.

요 약

IEEE 802.11은 보안상의 문제점이 많다는 것이 알려지게 되어, 강한 보안성을 제공하는 IEEE 802.11i 표준이 제안, 채택되었다. 하지만, 이는 WLAN을 단순히 웹 서핑을 하는데 사용하는 대부분의 사용자들에게 너무 많은 오버헤드를 요구한다. 한편, 통신을 하기 위해서는 노드 인증(node authentication)뿐만이 아니라 패킷 인증(packet authentication)도 필요하다. IEEE 802.11i에서는 이를 위해 TKIP(Temporal Key Integrity Protocol)과 CCMP(CTR with CBC-MAC Protocol)가 사용되지만, 오버헤드가 크다. 이 논문에서는 단순한 웹 서핑을 위한 오버헤드가 적은 패킷 인증 방법인 Lightweight Packet Authentication(LIPA)을 제안한다. 또한, LIPA의 성능을 TKIP, CCMP와 비교해보고, 특히 패킷 전송 시에 LIPA가 효율적이라는 것을 알아본다.

ABSTRACT

Because IEEE 802.11 has several security vulnerabilities, IEEE 802.11i was proposed and accepted. But IEEE 802.11i has much overhead for most of users for the web surfing. Besides not only node the authentication but also the packet authentication is needed to communicate. Although IEEE 802.11i uses TKIP(Temporal Key Integrity Protocol) and CCMP(CTR with CBC-MAC Protocol), they have a lot of overheads. In this paper, Lightweight Packet Authentication(LIPA) is proposed. LIPA has less overhead and short delay so that it can be affordable for simple web-surfing which does not need stronger security. After comparing performances of LIPA with those of TKIP and CCMP, LIPA is more efficient than other schemes for transmitting packets.

Keywords : *access control, IEEE 802.11, LIPA, packet authentication, WLAN*

Ⅰ. 서 론

휴대용 기기의 발전으로 무선 통신의 필요성이 부각되면서 때맞춰 등장한 IEEE 802.11은 IEEE 802.11b[1] 표준의 성공으로 널리 쓰이게 되었지만, IEEE 802.11b는 보안상의 취약점이 많이 지적되었으므로[2], 좀 더 확실한 보안을 제공해 줄 수 있는 표준이 필요하게 되었다. 이러한 이유로 IEEE 802.11 WLAN(Wireless Local Area Network)에서 필요한 모든 보안에 관련된 사항들을 정리해 놓은 IEEE 802.11i 표준이 제안되었고, 2004년 6월에 승인되었다. 그러나 IEEE 802.11i는 강한

접수일 : 2005년 2월 11일 ; 채택일 : 2005년 7월 14일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성·지원사업의 연구결과로 수행되었음.
† 주저자, ‡ 교신저자, soonlee@emerald.yonsei.ac.kr

보안성을 제공해 주는 만큼 시간지연(delay)이나 많은 계산상의 오버헤드(computational overhead)가 있어서, 배터리와 컴퓨팅 파워가 제한되어 있는 대부분의 무선기기에는 적합하지 않을 수 있다. 게다가 대부분의 사용자들은 WLAN을 단순히 웹 서핑을 하는 데 사용하기 때문에[3], 인터넷 뱅킹과 같이 보안이 매우 중요한 경우와는 다른, 좀 더 적은 오버헤드를 갖는 방법을 제공할 필요가 있다.

한편 IEEE 802.11을 포함한 무선 환경에서는 STA(Station)과 AP(Access Point) 사이를 무선으로 연결하므로, 누구나 쉽게 AP에 접근할 수 있어서 노드 인증(node authentication)으로 정당한 사용자만을 가려내는 작업이 필요하다. 그러나 공격자가 인증된 STA인 것처럼 가장해서 AP에 패킷을 보낼 수도 있으므로, AP에 도달한 패킷이 인증된 STA에서 보내졌다는 것을 확인하는 패킷 인증(packet authentication)도 역시 필요하다. 이는 공격자가 AP인 것처럼 가장하는 적대적 AP (rogue AP) 문제를 해결할 수 있다.

이와 같은 패킷 인증을 위해서 여러 가지 방법이 연구되었다[4-8]. [4]의 저자들은 MAC(Message Authentication Code)을 이용한 간단한 패킷 인증 방식을 제안했다. 기존의 HMAC[9]이나 UMAC[10] 등의 MAC 계산은 계산상의 오버헤드를 많이 필요로 하므로, 이를 모든 패킷마다 계산하면 성능이 많이 떨어지게 된다. 그러므로 [4]에서는 패킷들을 클래스 단위로 나누어 처리하여 그 해결책을 제시했지만, 전송 호스트(sending host)가 보내는 정보에 따라야 하는 한계가 있다. 또한, IP층(IP layer)에서 동작하는 패킷 인증 방법도 제시되었다[5]. 송신자와 수신자는 인증 스트림(authentication stream)을 공유하고, IPSec(IP Security)[11]의 슬라이딩 윈도우(sliding window)를 링크드 리스트(linked list)로 대치한 뒤 인증 스트림의 위치를 보낸다. 이는 각 데이터 패킷에 약간의 비트(bits)를 붙여서 보내는 것으로 패킷 인증을 가능하게 한다. SOLA[6-8]는 [5]를 발전시켜서 IEEE 802.11에서의 패킷 인증을 위한 방법을 제시했다. SOLA는 수신자와 송신자가 같은 비트 스트림(Bit Stream)을 갖고, 송신 시 그 스트림을 한 비트씩 차례대로 패킷에 붙여 송신하는 방법을 제안했다. 그러나 이는 오직 한 비트만을 이용하여 패킷을 인증하기 때문에 공격자가 유추하기 쉽고, 패킷이 손실되었을 때 스트림의 동기화(synchronization)가 힘들다는 단점이 있다.

IEEE 802.11i에서는 패킷 인증을 제공하기 위해서 TKIP(Temporal Key Integrity Protocol)과 CCMP(CTR with CBC-MAC Protocol)를 사용한다[12]. TKIP은 IEEE 802.11b의 보안 기법인 WEP(Wired Equivalent Privacy)을 기본으로 하므로 하드웨어를 교체할 필요가 없지만, CCMP에 비해 Related Message Attack과 같은 공격에 취약한 단점이 있다[12-13]. CCMP는 상당한 수준의 보안성을 제공하지만 하드웨어를 교체해야하고 오버헤드가 크다. 또한, TKIP과 CCMP는 유선 도메인에서 IPv6(Internet Protocol version 6)나 VPN(Virtual Private Networks) 등을 이용하여 제공되는, 종단 간(end-to-end) 보안을 위해 암호화한 패킷을 다시 암호화하므로 시간지연이 많이 발생한다.

본 논문에서는 IEEE 802.11i를 기본으로 인터넷 뱅킹 등 강력한 보안을 필요로 하지 않는 경우에 한하여, WLAN 서비스를 제공해주기 위한 최소한의 요건인 패킷 인증만을 제공하는 Lightweight Packet Authentication(LIPA)을 제안한다. LIPA는 종단 간 보안에서 이미 제공하는 무결성(integrity)과 기밀성(confidentiality)을 위한 메시지 암호화(message encryption) 과정을 생략하고 패킷 인증을 위한 새로운 방법을 제시함으로써 패킷을 보내기 전의 시간지연과 계산상의 오버헤드를 획기적으로 줄일 수 있다.

본 논문의 2장에서는 IEEE 802.11i에서의 패킷 인증 방법인 TKIP과 CCMP에 대해 간략히 설명하고, 3장에서는 LIPA의 동작 과정을 구체적으로 설명한다. 또한, 4장에서는 LIPA의 보안상 강점과 성능을 TKIP, CCMP와 비교해본다. 마지막으로 5장에서는 이 논문에 대한 결론을 맺는다.

Ⅱ. IEEE 802.11i에서의 패킷 인증 방법

2.1 Temporal Key Integrity Protocol (TKIP)

TKIP의 동작 과정은 다음과 같다[12].

1) 송신자(Source)는 MSDU(Medium Access Control Service Data Units)의 SA(Source Address), DA(Destination Address), MSDU priority, MSDU plaintext, 그리고 64-bit MIC(Message Integrity Code) key를 Michael 함수의 입력으로 넣은 뒤

64-bit MIC를 계산한다.

1-1) 64-bit MIC key를 32-bit K_0와 K_1으로 나눈다.

1-2) SA, DA, priority, plaintext를 포함하고 있는 MSDU를 32-bit M_0, ... M_{n-1}로 나눈다.

1-3) Michael의 계산 과정은 [12]와 같으며, MSDU를 N개의 32-bit 블록(block)으로 나누었으므로 n번의 계산이 필요하다.

2) 위에서 계산한 MIC를 MSDU에 붙인 뒤, 이 MSDU를 여러 개의 MPDU(Medium Access Control Protocol Data Unit)로 쪼갠다.

3) 각각의 MPDU에 TSC(TKIP Sequence Counter)를 하나씩 증가시켜서 붙인다.

4) Phase 1과 2 key mixing 과정을 통해서 WEP seed(IV와 RC4 key)를 만들어 낸 뒤, WEP으로 암호화해서 송신한다.

5) 수신자(Receiver)는 하나의 MSDU에서 쪼개져서 동일한 IV(Initialization Vector)를 가진 MPDU들을 모은 뒤, Michael을 계산해서 유효하지 않은 MIC를 포함하고 있는 MSDU는 폐기한다.

6) 60초 내에 두 번 이상 유효하지 않은 MIC가 나타나면, TKIP는 그 이후 60초 동안 어떤 수신도 받지 않으므로 공격자는 다량의 위조 공격(forgery attack)을 시도하지 못한다.

2.2 CTR with CBC-MAC Protocol (CCMP)

CCMP의 동작 과정은 다음과 같다[12].

1) 송신자는 PN(Packet Number)을 유지하면서, MPDU를 보낼 때마다 PN을 하나씩 증가시킨다. PN은 한 세션(session) 내에서 반복되지 않으므로 재전송 공격(replay attack)을 막을 수 있다.

2) MPDU의 헤더에서 AAD(Additional Authentication Data)를 만들어 낸다. 이는 다른 수신자가 재전송 공격을 하는 것을 방지해준다.

3) PN, A2(Address 2), MPDU의 Priority에서 Nonce를 계산해 낸다.

4) TK(Temporal Key), PN, AAD, Nonce를 이용해서 MIC를 계산한다.

4-1) plaintext MPDU의 첫 번째 블록을 AES로 암호화해서 CBC 모드의 IV를 만든다.

4-2) MPDU 헤더와 plaintext MPDU 데이터를 블록 단위로 쪼갠 뒤 CBC-MAC 모드의 AES 연산에 사용한다. 이 값과 다음 블록을 exclusive OR (XOR) 한 값을 AES에 넣는다. 쪼갠 블록이 소진될 때까지 이를 반복한다.

4-3) 위 연산의 결과 값인 MIC를 plaintext MPDU 뒤에 붙인다.

5) MIC가 붙어있는 plaintext MPDU와 TK, MPDU의 counter를 이용해서 암호화된 MPDU를 계산한다.

5-1) 1-byte flag, 1-byte Quality of Service(QoS) 정보, 6-byte address, 6-byte PN, 2-byte counter를 이용해서 CTR Preload를 만든다.

5-2) CTR Preload를 AES에 넣는다. MIC가 붙어있는 plaintext MPDU를 블록 단위로 쪼갠 뒤, 위 AES의 결과 값과 XOR 해서 ciphertext 블록을 얻는다. 더 이상 블록이 없을 때까지 이를 반복한다.

5-3) 위 연산으로 얻어진 ciphertext 블록들을 연결해서 암호화된 MPDU를 얻는다.

III. Lightweight Packet Authentication(LIPA)

이 장에서는 위에서 언급한 방법들의 문제점을 보완하고 웹 서핑과 같은 강한 보안성을 필요로 하지 않는 경우를 위해 제안된 오버헤드가 적은 Lightweight Packet Authentication(LIPA)에 대해 알아본다.

LIPA는 IEEE 802.11i 환경에서만이 아니라, 두 노드 사이의 패킷 인증이 필요한 경우에는 모두 적용할 수 있다. 즉, LIPA는 노드 인증이 이루어져 두 노드 간에 공유하는 키가 있다면 어느 환경 하에서도 사용가능하지만, 이 논문에서는 노드 인증을 위해서 IEEE 802.11i를 이용하므로 구체적인 패킷 인증 역시 IEEE 802.11i에 적합한 방식으로 설명한다.

3.1 LIPA의 가정 사항

LIPA에서 송신자와 수신자는 패킷이 이동해가는 방향에 따라 달라진다. 즉, STA이 AP로 패킷을 보내는 경우에는 STA이 송신자, AP가 수신자가 되고, 그 반대의 경우에는 AP가 송신자, STA이 수신자가 된다.

LIPA가 동작하기 위해서는 다음과 같은 가정과 사전 작업이 필요하다.

- 송신자와 수신자는 IEEE 802.11i의 인증 방법[12]을 이용하여 서로를 인증하고, 세션 키(session key)인 동일한 TK를 각각 유지한다고 가정한다.
- 송신자와 수신자는 세션이 시작되면 각각 다음과 같은 과정을 수행한다.
  1) TK에서 128 bit-Data Encryption/Integrity key를 추출한다.
  2) Data Encryption/Integrity key를 128 bit-BBS(Blum-Blum-Shub) generator [14]의 seed로 사용하여 계산한다. (gcd (seed, n)=1을 가정)
  3) 2)의 결과로 1024 bytes의 Authentication Stream을 얻는다.

3.2 패킷 인증 과정

먼저 송신자는 패킷을 보낼 때 패킷 인증을 고려하지 않고 IEEE 802.11 패킷을 만든다. 이 패킷 헤더에는 Sequence Control 필드가 포함되는데, 각 STA과 AP들은 single modulo 4096 counter를 유지하고[1], 각 패킷을 보낼 때마다 1씩 증가시켜서 이 필드에 실어 보낸다.

송신자는 패킷 인증을 위해 그림 1과 같이 패킷 인증 생성 과정을 수행한다.

- 패킷 인증 생성 과정 -

1) 송신자는 TK에서 Data Encryption/Integrity key를 뽑아낸다.
2) 송신자는 그림 2와 같이 보내고자 하는 패킷의 헤더에서 Source Address(SA)와 Sequence Control 필드를 뽑아서 AES 연산의 평문으로, Data Encryption/Integrity key를 AES 연산의 key로 입력한다.
   2-1) AES 연산의 결과는 128 bits 문자열(stream)이다.
   2-2) 이 문자열에서 가장 왼쪽의(leftmost) 13 bits를 SKey라고 부른다.
3) 그림 3과 같이 송신자가 갖고 있는 Authentication Stream에서 SKey만큼 오른쪽으로 떨어진 비트를 시작점으로 8 bytes의 AStream을 뽑아낸다. 예를 들어, SKey가 1이라면, Authentication Stream의 첫 번째 비트를 시작점으로 8 bytes의 AStream을 뽑아낸다. 이때 SKey가 8128보다 큰 수이면 랩어라운드(wrap-around) 방식을 사용한다.
4) 위에서 뽑아낸 AStream을 패킷에 붙여서 전송한다.

수신자는 전송 받은 패킷을 다음과 같은 패킷 인증 확인 과정으로 패킷 인증 여부를 확인한다.

- 패킷 인증 확인 과정 -

1) 받은 패킷의 Sequence Control 필드를 확인해서 현재까지 받은 패킷의 바로 다음이 아니면 폐기한다.
2) 수신자는 받은 패킷의 헤더에서 Fragment Number와 Sequence Number로 구성된 Sequence Control과 Source Address(SA) 필드를 뽑아서 그림 2와 같이 이것을 AES 연산의 평문으로 입력하고, Data Encryption/Integrity key는 AES 연산의 key로 사용한다.
   2-1) AES 연산의 결과는 128 bits 문자열

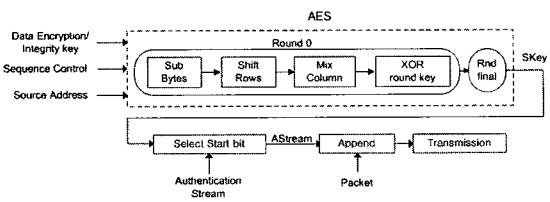


그림 1. LIPA의 packet authentication generation 과정

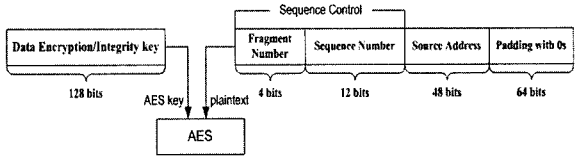


그림 2. Data Encryption/Integrity key와 Sequence Control, Source Address를 AES에 입력

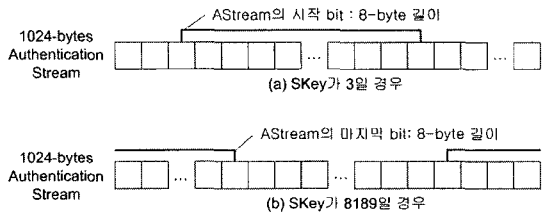


그림 3. SKey 값에 따른 Authentication Stream 선택

(stream)이다.

2-2) 이 문자열에서 가장 왼쪽(leftmost)의 13 bits를 SKey′라고 부른다.

3) 수신자가 갖고 있는 Authentication Stream에서 SKey′만큼 오른쪽으로 떨어진 비트를 시작점으로 8 bytes의 AStream′를 뽑아낸다. 이 과정은 패킷 인증 생성 과정에서 SKey와 AStream을 생성하는 과정과 동일하다. 이때 SKey′가 8128보다 큰 수이면 랩어라운드(wrap-around) 방식을 사용한다.

4) 이 AStream′가 받은 패킷에 붙어있는 AStream과 동일하지 않으면 받은 패킷을 폐기한다.

이 과정을 알아보기 쉽도록 수식으로 나타내면 수식 (1)과 같다.

$$\begin{aligned} &\textit{Authentication Stream} \\ &\quad = \textit{BBS(Data Encryption/Integrity key)} \\ &\textit{SKey} \\ &\quad = \textit{AES((Sequence Control || SA),} \\ &\qquad \textit{Data Encryption/Integrity key)} \quad (1) \\ &\textit{AStream (8 bytes)} \\ &\quad = \textit{Authentication Stream(SKey)} \\ &\textit{sending packet} \\ &\quad = \textit{original packet || AStream} \end{aligned}$$

3.3 LIPA의 특징

LIPA의 가장 큰 특징은 패킷이 만들어지기 전에 AStream이 미리 계산될 수 있다는 것이다. TKIP과 CCMP의 경우 보낼 패킷이 연산하는데 포함되므로 패킷이 만들어진 뒤에만 조합(encapsulation)이 가능하다. 즉, 패킷 전송을 위해 각 패킷이 만들어진 뒤 TKIP과 CCMP를 위한 연산 시간만큼 기다렸다가 전송해야한다. 반면, LIPA의 경우 각 패킷을 위한 AStream을 미리 계산(pre-computing)할 수 있으므로, 패킷이 만들어지면 LIPA 연산을 위한 시간지연 없이 전송이 가능하다. 그러므로 한 세션 당 전송되는 패킷의 수가 많을수록 TKIP, CCMP와 LIPA의 속도 차이는 커진다.

LIPA의 두 번째 특징은 AES 연산을 획기적으로 줄였다는 것이다. 58 bytes의 메시지를 CCMP와 TKIP, 그리고 LIPA를 이용하여 인증한다고 가정했을 때, CCMP의 경우 MIC를 만들기 위해서 7번, 암호화된 MPDU를 만들기 위해서 5번의 AES 연산이 필요하므로 패킷을 보내기 위해서 총 12번의 AES 연산이 필요하다. TKIP의 경우에도 MIC를 만들기 위해서 15번의 Michael 연산이 필요하다. 반면 LIPA의 경우 메시지 길이에 상관없이 SKey를 만들기 위해 오직 한번의 AES 연산이 필요하므로 AES 연산의 횟수는 메시지의 길이가 길어질수록 더 큰 차이를 보인다.

마지막으로 LIPA는 하드웨어를 교체할 필요가 없다. AES 연산은 소프트웨어와 하드웨어 두 가지 방식으로 작동이 가능하다. AES 연산을 위한 하드웨어가 갖추어진 노드의 경우 그 하드웨어를 사용하고, AES 연산을 위한 하드웨어가 없더라도 소프트웨어 업데이트를 통해 AES 연산을 수행할 수 있다. LIPA의 경우 한 패킷 당 단 한번의 AES 연산이 사용되므로 전체 시간에 미치는 영향력이 적어서 두 가지 중 어떤 방법을 사용해도 무방하다.

한편, LIPA는 단지 패킷의 출발지(origin)를 인증할 뿐이고, 패킷의 내용을 암호화하지 않으므로 무결성과 기밀성을 제공하지 않는다. 그러므로 LIPA는 빠른 속도가 필요하고 보안이 상대적으로 중요하지 않은 어플리케이션과 배터리와 컴퓨팅 파워가 적은 노드에 적합하다.

Ⅳ. Performance Analysis

이 장에서는 무선 환경에서 강한 보안성을 필요로 하지 않는 경우에 제안된 LIPA를 사용하여 패킷 인증을 제공할 때의 보안상 안전성과 인증 속도, 오버헤드를 TKIP, CCMP와 비교해보았다.

4.1 LIPA의 보안상 안전성

4.1.1 재전송 공격(Replay attack)에 대한 안전성

일반적으로 IEEE 802.11은 유니캐스트(uni-

cast) 패킷의 경우 재정렬(reordering)을 지원하지 않는다. 그러므로 송신자는 ACK(Acknowledgement)을 받지 못하면 그 다음 패킷을 전송하지 않기 때문에[1], Sequence Control 필드의 SN이 큰 패킷이 SN이 작은 패킷을 받기 전에 먼저 도착할 수 없다. 또한, LIPA는 패킷 헤더 내의 Sequence Control 필드의 값을 사용하는데, 이 필드는 TKIP의 TSC나 CCMP의 PN과 같이 한 세션 내에서는 반복되지 않는다. 그러므로 LIPA는 재전송 공격을 방지할 수 있다.

4.1.2 Brute-force attack에 대한 안전성

TKIP의 경우 WEP의 결함을 보완하기 위하여 여러 가지 방법을 사용했다[12]. 그 결과 WEP seed의 크기는 128 bits로 초기 WEP key의 크기인 40 bits, 또는 104 bits에 비해서 크게 늘었지만, IV 자체의 크기는 초기 WEP IV의 크기에서 변하지 않았다. 그러므로 TKIP은 초기 WEP과 마찬가지로 IV의 크기가 작아서 매 패킷마다 선택되는 IV가 중복될 수 있다는 결점을 갖고 있다. 이 경우 brute-force attack에 대해 안전하지 않다.

반면 CCMP의 경우 DES의 발전된 형태인 AES를 사용한다. AES는 그 보안 강도를 round key expansion의 복잡도(complexity)에 의존하며 키 길이가 128 bits이므로, 현재의 컴퓨팅 환경에서는 brute-force attack에 대해 안전하다고 할 수 있다[15]. 하지만 패킷마다 AES 연산을 여러 번 반복하는 CCMP는 오버헤드가 많으므로, LIPA는 SKey만을 AES key expansion을 이용하여 연산한다. 그러므로 LIPA는 CCMP에 비해 훨씬 적은 계산상의 오버헤드를 가지며, AES가 높은 계산상의 효율성(high computational efficiency)을 가지므로, 고속 어플리케이션(high-speed applications)에 적합하다.

4.1.3 SKey와 Authentication Stream에 대한 안전성

어떤 송신 노드 H에 대하여 Data Encryption/Integrity key의 크기를 k, SKey의 크기를 s, 그리고 AStream의 크기를 a라고 가정할 때 공격자의 유형은 다음과 같이 두 가지로 나누어질 수 있다.

첫째로 공격자가 아무런 정보도 갖고 있지 않다고 가정할 때, 다음에 전송될 패킷에 사용될 AStream을 알기 위해서는 brute-force attack을 시행해야 한다. AStream의 크기는 8 bytes이므로, 공격자는 각 패킷에 대하여 수식 (2)의 확률로 AStream을 추측할 수 있다.

$$Pr(AStream|H=attacker\ w/o\ information) = 2^{-a} = 2^{-64} \quad (2)$$

둘째로 공격자가 Authentication Stream을 알고 있지만, Data Encryption/Integrity key는 모른다고 가정할 때, 공격자는 다음에 전송될 패킷에 사용될 AStream을 알기 위해서 SKey만 추측해내면 된다. SKey는 13 bits로 AStream에 비해 크기가 작다. 그러므로 공격자가 brute-force attack을 이용하여 SKey를 추측할 경우, 수식 (3)의 확률을 갖는다.

$$Pr(next\ AStream\ |\ H=attacker\ w/\ Authentication\ Stream) = Pr(next\ SKey\ |\ H=attacker\ w/\ Authentication\ Stream) = 2^{-s} = 2^{-13} \quad (3)$$

수식 (3)의 확률 값이 일반적으로 작지 않기 때문에 brute-force attack에 취약하다고 생각할 수 있으나, 공격자가 이와 같은 확률을 얻기 위해서는 반드시 Authentication Stream을 알고 있어야 하므로 실제 확률은 수식 (4)과 같다.

Authentication Stream은 확률 테스트(statistical test)를 통과한 단방향 함수(one-way function)인 BBS PRBG(Blum-Blum-Shub Pseudo Random Bits Generator)로 얻을 수 있다[16]. 그러므로 공격자가 패킷을 스니핑(sniffing)해서 여러 개의 AStream을 획득하여도 Authentication Stream을 유효한 시간 내에 복원할 수 없다. 그러므로 결론적으로 SKey를 추측하기 위한 실제 확률은 수식 (4)과 같이 현시점에서 0이라고 할 수 있다.

$$\begin{aligned} &Pr(next\ SKey|H=attacker) \\ &= Pr(next\ SKey\ |\ H=attacker\ w/o\ information) \times Pr(Authentication\ Stream) \\ &= Pr(next\ SKey\ |\ H = attacker\ w/o\ information) \times 0\ at\ the\ present\ time \\ &= 0\ at\ the\ present\ time \end{aligned} \quad (4)$$

한편, LIPA는 공격자에게 패킷이 유출되어도 TK의 Data Encryption/Integrity key를 알 수 없도록 설계되었다. 공격자가 패킷을 스니핑할 경우 알아낼 수 있는 것은 그 패킷의 Source Address, Sequence Control 필드, 그리고 AStream이다. 공격자가 Authentication Stream을 알고 있다고 가정할 때, 공격자는 AStream을 이용하여 스니핑한 패킷의 SKey를 추측할 수 있다. 그러나 SKey는 AES 연산의 결과 값의 일부분이므로 동일한 SKey를 갖는 패킷이더라도 AES 연산의 결과 값은 다를 수 있다. 만약 AES 연산의 결과 값을 추측할 수 있다고 가정하더라도, 현시점에서 AES 연산의 결과 값을 알 때 입력 값을 추측할 수 없으므로 공격자는 TK의 Data Encryption/Integrity key를 알아낼 수 없다.

4.2 LIPA의 패킷 인증 속도와 오버헤드

TKIP은 MSDU마다 MIC를 계산하고, 메시지가 길어질수록 MSDU에서 쪼개지는 MPDU의 개수가 많아지므로 계산해야하는 WEP seed와 WEP 암호화의 횟수도 많아진다. CCMP 역시 메시지 자체와 key를 AES로 암호화하므로, 특히 메시지 길이가 길고 여러 개인 경우에는 패킷 인증을 하는 데 많은 오버헤드와 긴 시간지연, 즉 속도 저하를 가져온다. 하지만, LIPA는 SKey만을 AES로 암호화하고 단순한 shifting 연산을 이용하므로 TKIP과 CCMP 보다 훨씬 오버헤드가 적고 빠르게 패킷 인증을 수행한다.

그림 4는 한 세션 내에서 전송된 패킷의 개수에 따른 누적 시간지연을 보여준다. 전송되는 패킷의 개수가 증가할수록 시간지연이 빠르게 증가하는 CCMP와 TKIP에 비해 시간지연이 거의 증가하지 않는 LIPA를 볼 수 있다. 메시지의 길이는 260 bytes를 가정하였다.

그림 5는 한 세션 내에서 전송된 패킷의 개수에 따른 패킷 하나 당 시간지연을 보여준다. CCMP와 TKIP에 비해 적은 LIPA의 시간지연을 볼 수 있다. 이는 LIPA가 메시지 하나 당 필요한 AES 연산의 횟수가 한번인데 반해, CCMP의 경우 여러 번의 AES 연산이 필요하기 때문이다. TKIP 역시 하나의 MPDU 패킷을 보내기 위해서 Michael과 WEP seed를 만들기 위한 연산, 그리고 WEP 암호화 연산이 필요하기 때문에 LIPA 보다 시간지연이 크다. 그래프에서 100개의 패킷이 전송된 경우 LIPA의 시간지연이 더 많은 개수의 패킷이 전송된 경우에 비해서 더 큰 이유는 Authentication Stream의 연산 때문이다. LIPA는 세션이 시작될 때 BBS를 이용해서 Authentication Stream을 연산하는 데, 그에 따른 시간지연은 한 세션 당 전송되는 패킷의 수가 많아질수록 전체 시간지연에 영향을 적게 미친다. 메시지의 길이는 260 bytes를 가정하였다.

그림 6은 메시지의 길이에 따른 누적 시간지연을 보여준다. CCMP는 메시지의 길이가 길수록 AES 블록의 개수가 많아지기 때문에 연산해야 할 AES의 횟수가 늘어나는 반면, 메시지가 연산에 포함되지 않는 LIPA의 경우는 메시지의 길이가 길어져도 동일한 시간지연을 볼 수 있다. TKIP 역시 메시지 길이에 따라 블록의 개수가 많아지는 것이 아니기

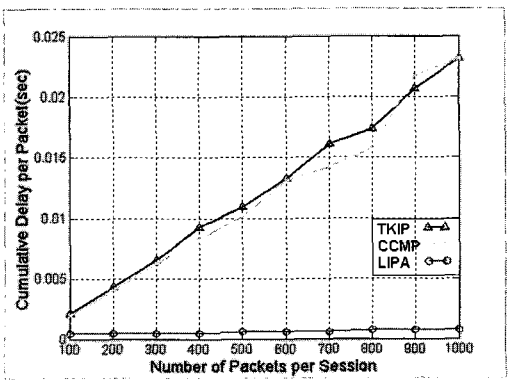


그림 4. 한 세션 내에 전송되는 패킷 개수에 따른 누적 시간지연

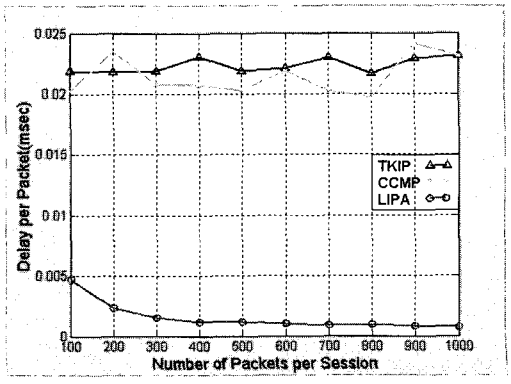


그림 5. 한 세션 내에 전송되는 패킷 개수에 따른 메시지 한 개당 시간지연

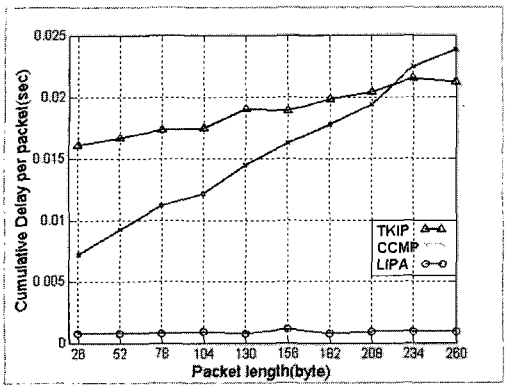


그림 6. 메시지 길이에 따른 누적 시간지연

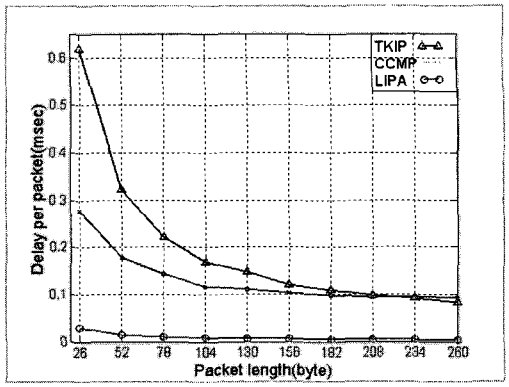


그림 7. 메시지 길이에 따른 byte당 시간지연

때문에 CCMP에 비해 그래프의 기울기가 상대적으로 완만하지만, TKIP의 연산량은 LIPA보다 많기 때문에 시간지연이 더 큰 것을 알 수 있다. 한 세션 당 패킷의 개수는 100개를 가정하였다.

그림 7은 메시지의 길이에 따른 byte 당 시간지연을 보여준다. 메시지의 길이가 길어져도 byte 당 시간지연은 LIPA에 비해 CCMP와 TKIP의 경우가 더 큰 것을 볼 수 있다. 한 세션 당 패킷의 개수는 100개를 가정하였다.

V. 결 론

본 논문은 IEEE 802.11 환경에서 동작하는 패킷 인증 방법인 Lightweight Packet Authentication(LIPA)을 제안했다. LIPA는 AES와 BBS를 이용하므로, 현시점에서는 공격자가 Data Encryption/Integrity key와 다음에 전송될 패킷에 사용될 AStream을 추측할 수 없다. 그림 4에서부터 그림 7에서와 같이 IEEE 802.11i에서 제공하는 패킷 인증 방법인 TKIP과 CCMP는 전송되는 패킷의 개수가 많아질수록, 또 메시지의 길이가 길어질수록 시간지연이 선형적으로 증가하지만, LIPA는 더 이상 증가하지 않고 동일한 시간지연을 보인다. 패킷의 개수와 메시지 길이에 따라 연산이 증가하는 TKIP과 CCMP와 달리, LIPA는 패킷의 개수와 메시지의 길이와는 독립적으로 Authentication Stream을 계산하고, 한번의 AES 연산으로 SKey를 구한다. LIPA는 IEEE 802.11i와 호환이 가능하고, TKIP과 CCMP에 비해 더 효율적인 패킷 인증을 제공해주므로 TKIP과 CCMP를 대체하여 사용될 수 있다. 그러나 LIPA는 강력한 보안을 필요로 하지 않는 경우에 WLAN 서비스를 제공하기 위한 최소한의 요건인 패킷 인증만을 제공하기 위해 고안되었으므로, 패킷의 출발지만을 인증하고 패킷의 기밀성과 무결성을 보장하지 않기 때문에, 다른 층에서 종단 간 보안을 제공해주어야 한다는 가정이 필요하다.

참 고 문 헌

[1] "Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements-part 11: Wireless Lan Medium Access Control (MAC) And Physical Layer (PHY) Specifications," *IEEE Std 802.11-1997*, pp. i-445, Nov. 18, 1997.

[2] N. Cam-Winget, R. Housley, D. Wagner, J. Walker, "Wireless networking security: Security flaws in 802.11 data link protocols," *Communications of the ACM*, 46(5), pp. 35-39, May 2003.

[3] 한국 인터넷 정보 센터, "2003 하반기 정보화 실태조사(요약보고서)," 정보통신부, site at: http://www.mic.go.kr/notice/index_view.jsp?idx=3400&page_no=1&mode=&selOption=&keyword=.

[4] K.L Calvert., S. Venkatraman, J.N. Griffioen., "FPAC: fast, fixed-cost authentication for access to reserved re-

sources," *Proceedings of IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies.(INFOCOM)*, 2, pp. 1049-1058, Jun. 23-27, 2002.

[5] F. Zhao, Y. Shin, S.F. Wu, H. Johnson, A. Nilsson, "RBWA: an efficient random-bit window-based authentication protocol," *Proceedings of Global Telecommunications Conference(GLOBECOM '03)*, 3, pp. 1379-1383, Dec. 1-5, 2003.

[6] H. Johnson, A. Nilsson, J. Fu, S.F. Wu, A. Chen, H. Huang, "SOLA: A One-bit Identiry Authentication Protocol for Access Control in IEEE 802.11," *Proceeding of IEEE Global Telecommunications Conference 2002*, 1, pp. 768-772, 2002.

[7] F. Wu, H. Johnson, A. Nilsson, "SOLA: lightweight security for access control in IEEE 802.11," *IT Professional*, 6(3), pp. 10-16, May-June 2004.

[8] Kui Ren, Hyunrok Lee, Kyusuk Han, Jaemin Park, Kwangjo Kim, "An Enhanced Lightweight Authentication Protocol for Access Control in Wireless LANs," *IEEE International Conference On Networks(ICON '04)*, Hilton, Singapore, Nov. 16-19, 2004.

[9] H. Krawcyk, M. Bellare, R. Canetti, "HMAC: Keyed-hashing for message-authentication," RFC 2104, Feb. 1997.

[10] J. Black, S. Halevi, H. Krawcyk, T. Krovets, P. Rogaway, "MAC: Fast and secure message authentication," Proceedings of Lecture Notes in Computer Science, *Springer-Verlag*, CRYPTO '99, 1666, pp. 216-233, 1999.

[11] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, Nov. 1998.

[12] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Medium Access Control (MAC) Security Enhancements," IEEE Std 802.11i-2004, pp. 1-175, 2004.

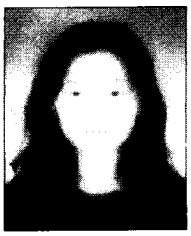
[13] A. Wool, "A Note on the Fragility of the "Michael" Message Integrity Code," *IEEE Transactions on Wireless Communications*, 3(5), pp. 1459-1462, Sep. 2004.

[14] L. Blum, M. Blum, M. Shub, "A Simple Unpredictable Pseudo-Random Number Generator," *SIAM Journal on Computing*, 15(2), pp. 364-383, 1996.

[15] W. Stallings, "Cryptography and Network Security Principles and Practice," *Prentice Hall*, 2, pp. 74-75, 1999.

[16] A. Menezes, P. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography," *CRC Press Inc.*, pp. 175-184, 1997.

〈著者紹介〉



이 근 순 (KeunSoon Lee) 정회원
2003년 8월: 연세대학교 컴퓨터산업시스템공학과 졸업
2003년 9월~현재: 연세대학교 컴퓨터과학과 석사과정
〈관심분야〉 정보보호, 무선통신

김 효 진 (HyoJin Kim) 정회원
2002년 2월: 연세대학교 기계전자공학부 정보산업공학과 졸업
2004년 2월: 연세대학교 컴퓨터산업시스템공학과 석사 졸업
2004년 3월~현재: 연세대학교 컴퓨터과학과 박사과정
〈관심분야〉 유·무선통신, 정보보호

송 주 석 (JooSeok Song) 정회원
1976년 2월: 서울대학교 전기과 졸업
1979년 2월: KAIST 전기과 석사 졸업
1988년 2월: Dept. of Computer Science, University of California at Berkeley 졸업
1989년 3월~현재: 연세대학교 컴퓨터과학과, 정교수
〈관심분야〉 정보보호, 유·무선통신