PRF알고리즘 기반의 패킷 단위 암호화 구현 연구

김경호***, 김일준**, 김석우***

한세대학교 정보보호공학***,국가보안기술연구소**,

The Study of Packet Encryption Implementation based of PRF algorithm

KyoungHo Kim***, IlJun Kim**, SeokWoo Kim***

Dept. of Information Security, HanSei University***
National Security Research Institute**

요 약

무선 사용자의 증가에 따라 802.11 표준에 대한 취약점이 대두되었다. 무선랜에서 보안을 위한 IEEE 워킹 그룹에서는 EAP, 802.1x, RADIUS, TKIP, 802.11i등의 표준안을 제시 하여 최근 WPA 가 Wi-Fi 에 의해 802.11i의 중간 단계로써 제안되어 구현되고 있다. 본 논문에서는 향후 각 회사 별 WEP 취약점 보완을 위한 패킷단위의 IV변경을 PRF 알고리즘에 기반하여 암호화를 리눅스 디바이스 드라이버 레벨에서 시험 구현하였다. 임베디드 시스템에 적용하기 위한 구현 타당성을 위하여 PXA255 프로세서에서 client를, 인텔 PC에서 HOST AP 환경을 시험환경으로 사용하였다.

1. 서론

IEEE 802.11 표준을 따르는 무선랜 사용자가 급증함에 따라 무선 인터넷 사용자들은 공간에 제약 받지 않으면서 이동중에 전자상거래, 화상회의, 전자메일 및 홈 네트워크 등을 고속으로 편리하게 서비스 받고 싶은 욕구가 증가하게 되었다.[1] 그러나, 진행 중인 무선랜의 기술 발전에 대한 WEP 보안에 대한 여러 취약점이 대두되었다.[4-6] 무선랜 보안의 이와같은 취약점을 보완하기 위하여 IEEE 802.11 위킹그룹에서는 MAC 칩 자체를 AES-OCB로 변경하는 방법과 보안 문제를 소프트웨적으로 개선하려는 TKIP(Temporary Key Integrity Protocol)을 제시하여 기존의 무선랜과 호환하기위한 방법을 제시하였다.[1] 하지만 TKIP는 유선망의 인증서버의 키를 이용하여 키를 생성하는 의존성을 가지게 되어 AP와 무선 랜만으로 구성되는 소규모 무선 네트워크 망에는 적용하기에는 부적합하다.

본 논문에서는 무선 랜의 취약점과 이에 따른 보안 표준을 논의하고 이중 기밀성의 문제를 무선랜 자체에서 해결하기 위해 리눅스 환경에서 구현된 무선랜 및 AP 시스템을 기반으로 기존의 디바이스 드라이버를 수정하여 초기 값 IV와 PRF알고리즘을 이용해 생성된 IV'값을 Exclusive OR하여 새로운 IV값을 만들고, TKIP에서 제시한 패킷별 암호화를 위한 키를 생성여 취약한 기밀성을 보장하기 위하여 시험 수현하였다. 동 시험은 WEP IV를 패켓마다 변경하여 사용함을 목적으로 수행되었다.

2. 무선랜 WEP 취약점과 보안 표준

2.1 무선랜 보안 WEP

그림 1의 WEP은 무선랜 브로드캐스팅 방식으로 전

송 되어지는 데이터 스트림의 보안성을 제공하기 위하여 IEEE802.11표준에 정의된 암호화 스킴으로서, 데이터의 암복화에 동일키와 알고리즘을 사용하는 대칭형 구조이다. WEP 키는 단말을 인증하고, 데이터 프라이버시를 제공하는 데 사용된다. AP의 서비스를 받는 모든 단말은 40비트 크기의 암호키를 공유하고 있다. AP는 단말을 인증하기 위해 random challenge를 보내면, 단말은 40비트의 암호키와 24비트의 IV를 결합하여 이를 RC4 암호화 알고리즘에 입력시켜 의사 난수 키 스트림을 생성하고 이를 이용해 평문은 전송할 메시지 M과 메시지에 대한 CRC 검증값으로

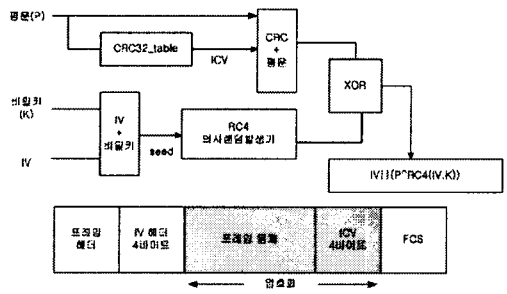

그림1. WEP 암호화 과정 및 프레임

구성되어 암호화 하여 전송한다.

수신측에서는 IV 값과 K값을 프레임에서 분리하여 RC4에 입력하여 키 스트림을 생성하여 암호문과 XOR하여 복호화를 실행하고 ICV 값을 분리 하여 복호화된 메시지를 CRC 알고리즘에 입력하여 생성된 ICV'값과 비교하여 무결성을 검사한다.[2]

2.2 무선랜 WEP 취약점

WEP 방식의 보안의 문제점은 IV는 공유키에 추가되어 패킷마다 2^{24}가지의 키 중 하나를 선택하여 암호화를 수행한다. 그런데 키 스트림은 재사용될 수 없기 때문에 공유키로부터 도출된 2^{24}가지의 키는 AP 1대로 이루어진 BBS가 11Mbps로 통상의 패킷을 송수신하는 경우 1시간 정도면 소비되어 재 사용 시 키 수열 소거 현상이 발생하여 암호문 단독 공격과 평문 공격에 취약하다. 또한 메시지 인증방식으로 32비트 CRC checksum을 사용하고 있다. 그러나 CRC는 전송 도중 발생할 수 있는 random error에 대한 대책일 뿐 이며 악의 있는 공격자의 메시지 변조 공격에 대한 대비책이 되지 못한다. 그리고 사용자 인증에서 공격자가 임의의 평문과 그에 해당하는 암호문쌍을 알고 있다면, 불법적인 인증이 가능하다.

2.3 취약점에 따른 보안 표준

WEP의 취약점을 해결하기 위해 IEEE802.11i에서는 두가지 접근 방식을 채택하고 있다.[1] 하나는 강도가 높은 알고리즘(AES-OCB)으로 바꾸는 것인데 이러한 방식은 MAC 칩 하드웨어를 변경해야 하는 것으로 기존의 무선랜 장비와 호환이 되지 않는다. 또 다른 방법으로는 소프트웨어적으로 개선하는 TKIP방식이다.

TKIP 프로토콜은 key-mixing 함수 Michal MIC함수, 키교환 프로토콜, re-keying 프로토콜로 구성된다. TKIP에서는 WEP을 이용한 암호화 이전에 무선단말과 인증서버의 상호 인증을 하기 위해서 생성된 키 PMK(Pairwise Master Key)가 AP로 전달 되어진다. PMK키는 TKIP에서 사용되는 512비트 키인 PTK(Pairwise Transient key)로 확대 되어 키생성 과정을 거치게 하여, WEP에 적용되는 키가 각 데이터 프레임마다 변경되도록 하였다. 그리고, 메시지 인증코드인 MIC를 프레임에 포함시켰다. 이러한 방법으로 알려진 WEP 알고리즘의 취약점을 해결하였다. 그림 2은 TKIP 알고리즘의 암호화 과정을 보여준다.

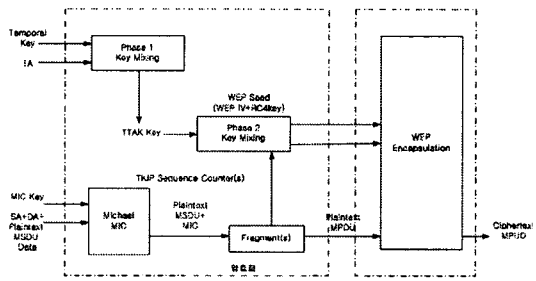

그림 2. TKIP 알고리즘의 암호화 과정

키 교환과정에서 액세스포인트와 스테이션에 동일한 temporal key와 MIC key가 생성되며, TSC는 각 데이터 프레임마다 1씩 증가하는 카운터이다. 프레임을 송신할 경우, temporal key와 TSC를 이용하여 데이터 프레임을 전송할 때마다 다른 WEP seed를 만들어, 메시지 인증코드인 MIC 코드와 함께 WEP으로 암호화 한다. 프레임을 수신한 경우, 암호화된 데이터 프레임의 확장 IV 필드에서 TSC 값을 읽어와 temporal key와 함께 WEP seed를 만들어 복호화한다. 그리고 자체적으로 계산한 MIC 값과 수신 프레임에 포함된 MIC 값을 비교하여 메시지의 무결성을 검증한다.[3]

2.3 IEEE802.11i 보안 표준의 문제점

IEEE802.11i에서 WEP의 보안 취약점을 해결하기 위한 방법 중에 TKIP는 유선망의 인증서버의 PMK키를 512비트인 PTK 키로 확대하여 WEP의 기밀성, 무결성, 인증등의 취약점을 소프트웨어적으로 보안하였으나 이것은 유선망에 부수적인 시스템을 가지고 있지 않는 AP와 무선랜만으로 구성되는 소규모 무선 네트워크망에서는 이러한 보안 서비스를 제공 받을수 없다. 이러한 문제점 중 기밀성의 취약점을 해결하기 위해서는 무선랜의 취약점에 따른 개선안을 제시하여 기존의 무선랜 디바이스 드라이버에 적용함으로써 해결하는 것이다.

3. 무선랜 보안 디바이스 드라이버 구현

3.1 무선랜 기밀성 문제를 위한 개선안

적은 크기의 초기 값 IV과 비밀키 K의 문제점으로 인한 기밀성의 취약성을 해결하기 위한 개선안으로 IV의 생성시 매번 다른 초기 값 IV값을 생성하고, 패킷별로 새로운 키를 생성하여 전송하려는 패킷마다 새롭게 생성된 키로 암호화를 할 수 있게 한다. 이러한 방법은 IV값과 키의 재 사용시 기밀성의 취약점을 점을 보안 할 수 있다.

3.2 기밀성을 보장하는 무선랜 드라이버 설계

본 논문에서는 무선랜 자체적으로 드라이버 단에서 기밀성을 보장하는 시스템을 설계 하였다. 아래의 그림.3에서처럼 블록들이 구성이 되어진다.

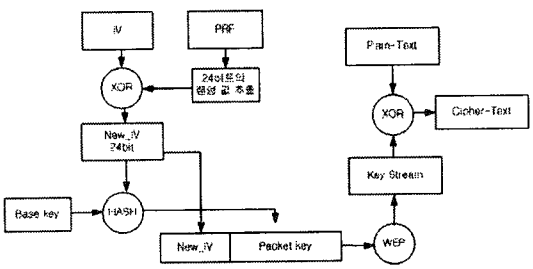

그림3. 기밀성을 보안한 무선랜 블록도

기존 무선랜에서 생성되어지는 초기 값 IV값에 EAP-Keying에 이용된 PRF알고리즘을 이용해서 랜덤한 값 24비트 추출하여 기존 IV값과 Exclusive OR 하여 새로운 값을 생성하여 초기 값 IV의 재사용을 막을수 있다. 또한 IEEE에서 패킷별로 전송 WEP키를 변경하는 Task Group I 드래프트의 WEP 성능 개선안을 구현하였다, 새롭게 생성된 New_IV가 임의로 생성되 WEP키와 연결되어 두 개의 값은 WEP 알고리즘으로 처리되어 키 스트림을 생성한다. 키 스트림은 일반 텍스트와 혼합되어 암호문을 생성한다.[4] 패킷별 키 생성 설계에서는 WEP 키와 New_IV를 해시하여 새로운 패킷 키를 생성하여 프로세스를 확장한다. 새로운 New_IV는 패킷 키와 연결되어 정상적으로 처리된다.

3. 구현

운영체제로 리눅스 커널 2.4.20를 사용하는 데스크탑 PC를 개발환경으로 사용하였으며, Prism2 계열의 칩을 사용하였으며, HostAP, linux-wlan의 오픈 무선 드라이버를 사용하였다.

linux-wlan와[5] HostAP의[6] 디바이스 드라이버는 Prism2 계열의 칩을 사용하는 무선랜 장비에 대한 소스코드가 공개 되어 있는 리눅스용 무선랜 디바이스 드라이버들이다.

이 소스코드를 기반으로 기밀성을 보장하기 위한 기능을 추가하기 아래의 그림.4 에서 보는것과 같이 구성되어진다. 암호화를 실행하는 블록도 각 각의 함

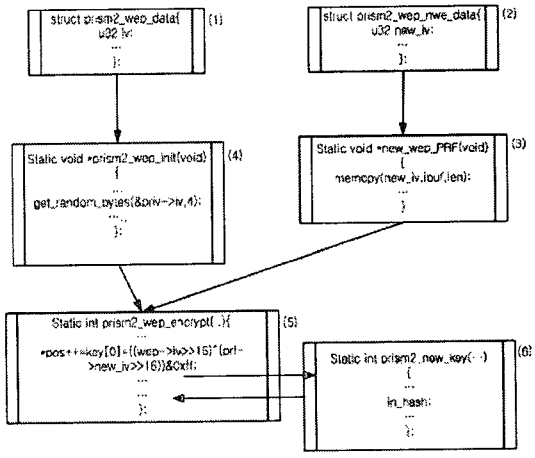

그림4. 기밀성을 보장하기 위한 소스 코드 블록도

수 중에서 (1)번째 블록은 초기값 IV와 키 길이 등과 같이 키에 대해서 정의 되어지고 있으며, 새로운 IV를 정의 위해서 (2)번째 블록에서 정의 하고 이후 (3)번째 블록에서 새로운 IV 값을 생성하기 위한 PRF 알고리즘이 정의 되어 (2)번째 블록에 정의 new_iv에 24비트를 저장 한다.. (4)번째 블록에서는 기존의 IV 값을 생성하여 iv에 저장이 되어진다. 이러한 값들을

(5)번째 블록에서 사용되어진다. 이 함수에서는 기존의 IV값과 New_iv값을

=key[0]=((wep->iv>>16)^(prf->new_iv>>16))&0xff;

=key[1]=((wep->iv>>8)^(prf->new_iv>>8))&0xff;

=key[2]=((wep->iv)^(prf->new_iv))&0xff;

Exclusive OR 하여 지정된 번지에 저장 되어지고 기존의 IV와 비밀키로 키 스트림을 생성하지 않고 (6)번째 블록에서 WEP키와 새로 생성된 New_IV를 Hash하여 새로운 패켓 키를 생성하여 다시 리턴하여 준다. 이후의 과정은 아래의 그림. 5처럼 리턴 받은

```
/* Setup RC4 state */
        for (i = 0; i < 256; i++)
                S[i] = i;    j = 0;
        kpos = 0;
        for (i = 0; i < 256; i++) {
                j = (j + S[i] + key[kpos]) & 0xff;
                kpos++;
                if (kpos >= klen)
                        kpos = 0;
                S_SWAP(i, j);
        }

        /* CRC32 */
        crc = ~0;
        i = j = 0;
        for (k = 0; k < len; k++) {
                crc = crc32_table[(crc ^ *pos) & 0xff] ^ (crc >> 8);
                i = (i + 1) & 0xff;
                j = (j + S[i]) & 0xff;
                S_SWAP(i, j);
                *pos++ ^= S[(S[i] + S[j]) & 0xff];
        }

crc = ~crc;
        /* Append little-endian CRC32 and encrypt it to produce ICV */
        pos[0] = crc;
        pos[1] = crc >> 8;
        pos[2] = crc >> 16;
        pos[3] = crc >> 24;
        for (k = 0; k < 4; k++) {
                i = (i + 1) & 0xff;
                j = (j + S[i]) & 0xff;
                S_SWAP(i, j);
                *pos++ ^= S[(S[i] + S[j]) & 0xff];
        }
```

그림5. 패킷 키 생성 이후의 순서 소스 코드

패킷 키와 New_iv로 키 스트림을 진행하여 일반 텍스트와 혼합되어 암호문을 생성한다.

4. 결론

리눅스 환경을 사용하여 유선망의 인증서버를 사용하지 않고 기밀성을 보장하는 무선랜을 제작되어, embedded 시스템으로 만들어 질 수 있으며. Prism2 칩을 사용하는 무선랜을 사용하였기 때문에 별도의 액세스포인트용 하드웨어를 사용하지 않고 일반 스테이션용 무선랜 카드를 사용하여 제작비를 줄일 수 있다. 단, 데이터 프레임의 암복호화 과정이 디바이스 드라이버에서 이루어지므로, 속도면에서 문제가 될 수 있으나, 암복호화 과정이 하드웨어 내에서 이루어지는 액세스포인트용 펌웨어를 사용하여 성능 문제를 해결할 수도 있을 것이다,

이 무선랜 디바이스 드라이버를 사용자 인증에 대한 부분만을 보안 한다면 유선망과 AP를 사용하지 않는 독립된 Ad-Hoc 네트워크 망의 취약한 기밀성을 보장할 수 있을 것이다. 향 후, 무선 랜 취약점을 보완할 수 있는 CCMP 암호알고리즘의 구현을 포함한 802.11i의 풀 구현이 기대되지만, 그 전초 단계로 기존의 무선랜 환경을 포함하는 과도기적 구현단계로써 본 논문의 의미를 부여코저 한다.

[참고문헌]

[1] IEEE Std 802.11i/D2.0."Specification for Enhanced Security", Mar.2002

[2] ANSI/IEEE Std 802.11,"Wireless LAN Medium Access Control(MAC)and Physical Layer(PHY) Specifications,"1999

[3] 오경희,강유성,정병호 "소프트웨어 구현을 통한 WPA 지원 무선랜 액세스포인트 개발", 2003년 정보과학회 춘계학술발표대회, 2003.4.24-26,제주대학교

[4] Pejman Roshan "802.11무선 LAN보안에 포괄적 검토 및 Cisco Wireless Security Suite", 2002.12, www.cisco.com/kr

[5] Gast, Matthew S "802.11 Wireless Networks: The Definitive Guide" O'RELLY, pp.141-175, 2002

[6] Bruce Potter ,Bob Fleck "802.11Security" O'RELLY, pp.148-157, 2003