WEP 프로토콜의 FPGA 구현

하 창 수, 최 병 윤
동의대학교 컴퓨터공학과
전화 : 051-890-1704 / 핸드폰 : 011-882-2577

FPGA Implementation of WEP Protocol

Chang-Soo Ha, Byeong-Yoon Choi
Dept. of Computer Engineering, Dongeui University
E-mail : systemonchip@hanmail.net

Abstract

In this paper a FPGA implementation of WEP protocol is described. IEEE 802.11 specifies a wired LAN equivalent data confidentiality algorithm. WEP(Wired Equivalent Privacy) is defined as protecting authorized users of a wireless LAN from casual eavesdropping. WEP use RC4 algorithm for data encryption and decryption, also it use CRC-32 algorithm for error detection.

The WEP protocol is implemented using Xilinx VirtexE XCV1000E-6HQ240C FPGA chip with PCI bus interface.

I. 서론

무선 LAN에서 컴퓨터는 무선 기지국의 범위 내에 있는 내부 및 외부 장소로 데이터를 전송, 수신할 수 있다. 무선 LAN을 이용하려면 컴퓨터에 Wi-Fi(802.11) 인증 전파 수신기가 장착되어 있어야 한다. Wi-Fi 인증을 사용하면, 집, 회사와 같이 Wi-Fi 인증 제품이 설치되어 있는 곳, 핫스팟(Hot Spot)이 있는 곳이면 어디든지 연결이 가능하다. Wi-Fi 인증은 무선 LAN 제품의 상호 운용성을 인증하는 비영리 국제 협회인 Wi-Fi 협회에서 주관하고 있다. WEP는 Wi-Fi 협회(802.11b) 표준에서 정의하고 있는 무선 로컬 영역에서의 네트웍에 대한 보안 프로토콜로서 유선 랜과 동등한 수준의 보안을 유지시키기 위한 것이다. WEP 프로토콜의 핵심 기능은 데이터의 암호화/복호화 연산과 전송 데이터의 에러 발생 유무를 검사하는 연산을 수행하는 것이다. 각각의 기능들을 구현하기 위해 사용하는 알고리즘은 암호화/복호화에 사용되는 RC4-PRNG(Pseudo Random Number Generator)와 데이터 오류검출에 사용되는 CRC-32 이다.

본 논문은 무선 LAN 상의 데이터 량이 증가하고 전송속도가 고속화 됨에 따라 소프트웨어로 구현된 WEP의 병목현상을 줄이기 위한 하드웨어 설계가 필수적이라는 판단아래 다양한 크기의 키 값들을 가질 수 있는 RC4와 고속으로 동작 가능한 32-Bit CRC 회로를 중심으로 WEP 프로토콜을 Verilog HDL로 설계하고 PC와 PCI 인터페이스를 사용하여 만들어진 회로의 올바른 동작을 검증하였다.

II. WEP 프로토콜 분석

2.1 WEP 프로토콜

WEP 프로토콜은 무선 랜 환경에서 유선 랜과 동등한 수준의 보안을 유지하기 위해 사용되는 프로토콜이다. 데이터의 보안성을 제공하기 위해 RC4 알고리즘을 사용하여 암호화/복호화를 수행하며 데이터의 무결성을 제공하기 위해 32-Bit CRC 알고리즘을 사용한다.

WEP는 몇 가지 특징을 가지고 있으며 아래 표 1에 나타내었다.

1) 데이터의 기밀성은 키의 관리에 의존한다
2) Brute-Force(주먹구구식) 공격에 충분히 강하다
3) 하드웨어나 소프트웨어로 구현하기에 효율적이다
4) IEEE 802.11 의 Option 이다

표 1. WEP 프로토콜의 특징

WEP는 40-bit(또는 104-Bit, 128-bit)의 키를 사용하여 데이터의 암호화/복호화를 수행하는데 이 키는 비밀 키의 형태로 관리되어야하며 데이터의 기밀성은 키의 관리에 의존하게 된다. 이것은 LAN의 규모가 작을 때는 문제가 되지 않지만 LAN의 규모가 커질 경우 키의 관리가 어려워진다는 단점이 있다. 키의 관리가 잘 이루어질 경우, 액세스 포인트에서 수행되는 SSID, MAC 주소 필터링 등의 기술과 같이 사용되어 높은 수준의 안전성을 제공한다.

2.2 WEP Encipherment/Decipherment

송신측과 수신측에서 데이터가 WEP를 통해 처리되는 과정을 그림 1과 2에 나타내었다.

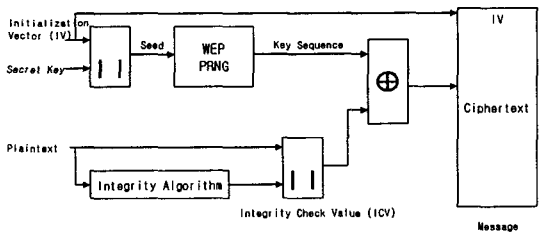

그림 1. WEP Encipherment Block Diagram

송신측에서는 송신하려고 하는 데이터와 외부에서 제공하는 키 값과 초기 값을 사용하여 수신측에서 받게될 메시지를 생성하게 된다. 데이터의 암호화를 수행하기 위해 IV(Initialization Vector)와 Key를 사용하여 PRNG로 들어갈 Seed를 생성하며, PRNG로 사용되는 RC4는 생성된 Seed를 사용하여 가상 난수를 생성한다. 송신하려고 하는 데이터는 무결성을 제공하기 위해 CRC-32를 거쳐 ICV(Integrity Check Value)를 생성하게 된다. 송신하려는 데이터와 ICV는 암호화된 메시지를 생성하기 위해 RC4를 통해 만들어진 가상 난수와 XOR 연산을 수행하게 되며 최종적으로 수신측에 전송되는 데이터는 IV와 암호화된 데이터이다.

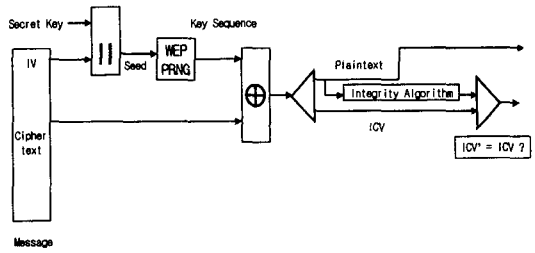

그림 2. WEP Decipherment Blcok Diagram

수신측에서는 송신측에서 수행했던 작업을 역순으로 수행하여 암호화된 데이터를 복원시킨다. 키와 iV를 사용하여 Seed를 생성하고 RC4를 통하여 가상 난수를 생성한 뒤 암호화된 데이터와 XOR 연산을 수행하여 송신측에서 보내고자 했던 원래의 데이터를 복호화 시킨다. 복원된 데이터는 ICV를 포함하고 있는데 수신측에서는 복원된 데이터 중 ICV를 제외한 나머지들의 ICV'을 다시 계산하여 전송 중 오류가 발생하였는지를 검사한다. 만약 오류가 발생하였다면 수신된 데이터는 버려지고 송신측에 재전송을 요구하게 된다.

2.3 PDU 분석

WEP 프로토콜에서 사용되는 PDU는 MAC 프로토콜의 PDU이며 그림 3에 PDU의 형식을 나타내었다.

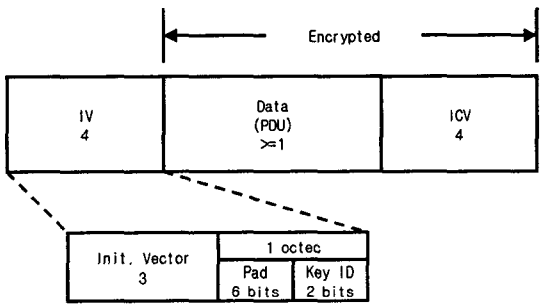

그림 3. WEP MPDU

키와 함께 사용되어 RC4의 Seed를 생성하는 IV는 총 4-byte이며 그 중 3-byte는 초기값을 가지며 나머지 1-byte는 현재 IV에 대한 정보를 가진다. 1-byte중 6-bit는 PAD이며 2-bit는 key id 값을 가진다. 이 id는 WEP 프로토콜이 사용하는 것은 아니며, 외부의 키 관리자가 가질 수 있는 여러 가지의 키들 중 임의의 키를 구별하기 위해 사용하는 ID이다. 외부 키 관리자는 임의의 연결에서 4가지의 키들을 가질 수 있으며 각 IV는 자신을 구별하기 위한 key id를 가진다.

Data는 1-byte이상의 크기를 가져야 하며, ICV는

Data에 대해서만 CRC를 수행하여 생성된다.

III. WEP 프로토콜의 하드웨어 설계

WEP 프로토콜을 하드웨어로 설계하기 위해 필요한 회로는 앞에서 살펴본 바와 같이 RC4, CRC-32가 필요하며 XOR 단과 ICV를 검사하기 위한 비교기 등 프로토콜의 수행에 필요한 나머지 연산들을 구현할 회로가 필요하다. 그림 4에 WEP 하드웨어의 전체 블록도를 나타내었다.

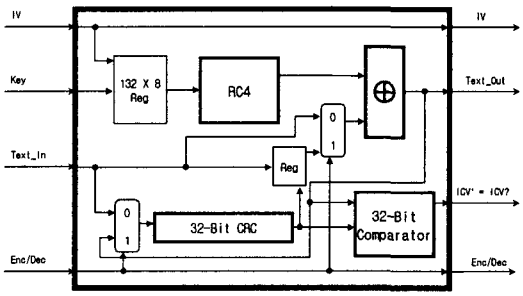

그림 4. WEP 하드웨어 블록도

RC4와 CRC-32 회로는 각각을 따로 설계하여 Core로 삽입하였으며 WEP 동작에 필요한 레지스터들을 추가하였다. Enc/Dec 제어 신호를 사용하여 송신/수신에 따라 필요한 연산을 수행할 수 있도록 하였으며 XOR 단과 32-bit 비교기는 상위모듈에서 조합회로로 구현하였다.

WEP 프로토콜이 하드웨어로 구현하기가 효율적이라고 앞서 언급하였는데 그 이유를 위 블록도에서 살펴볼 수 있다. 암호화/복호화를 위해 수행되는 RC4 동작과 ICV를 생성하고 검사하는 CRC의 동작이 같은 하나의 회로로 수행 가능하기 때문에 하드웨어의 공유가 쉽게 이루어질 수 있다. 그러므로 하드웨어 구현이 간결하고 쉬워지며 면적을 줄일 수 있고 제어회로 또한 복잡하지 않게 구현할 수 있다.

IV. RC4와 CRC의 하드웨어 설계

4.1 RC4 알고리즘

RC4는 RSA Data Security의 Rivest에 의해서 설계된 스트림 암호이며 바이트 단위 연산을 가지는 가변 길이 키 스트림 암호이다. 아래의 코드는 RC4의 알고리즘이다.

```
for i=0 to 255 {
    Si=i
    Ki=key(i mod key_length)
}
j=0
for i=0 to 255 {
    j=(j+Si+Ki) mod 256
    swap Si and Sj
}
for each pseudo-random byte to be generated {
    i=(i+1) mod 256
    j=(j+Si) mod 256
    swap Si and Sj
    t=(Si+Sj) mod 256
    pseudo-random byte is St
}
```

그림 4. RC4 알고리즘

4.2 RC4의 하드웨어 설계

그림 4 의 알고리즘을 하드웨어로 구현하기 위해 2개의 메모리와 3개의 피연산자를 가지는 덧셈기가 필요하다. 그림 5에 RC4에 대한 데이터 패스를 나타내었다.

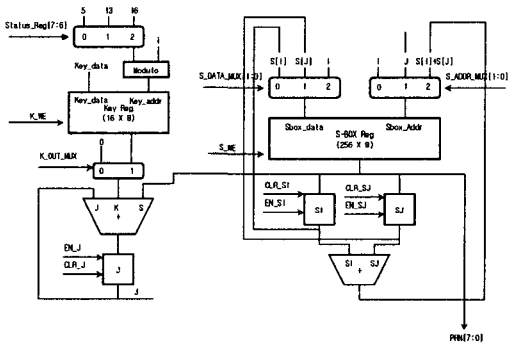

그림 5. RC4 Data Path

4.3 CRC-32

CRC는 데이터 통신에 사용되는 4가지 Cyclic Code (VRC, LRC, CRC, CheckSum)들 중 가장 강력하며 인코더와 디코더의 구현이 매우 간단하다는 특징을 갖는다. 또한 버스트 에러를 포함하여 에러검출에 매우 좋은 성능을 가지므로 신뢰성 있는 데이터 전송을 위한 에러 제어 코딩 방식으로 널리 사용되고 있다.

WEP 프로토콜에서 사용되는 CRC-32의 다항식은 아래와 같다

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + x^{0}$$

그림 6. WEP에서 사용되는 CRC-32의 다항식

CRC는 1의 보수 덧셈 방식을 사용하여 데이터를 주어진 다항식으로 나누며 나누어진 결과에서 나머지를 취하는 알고리즘이다. 아래에 CRC 연산 예제를 나타내었다.

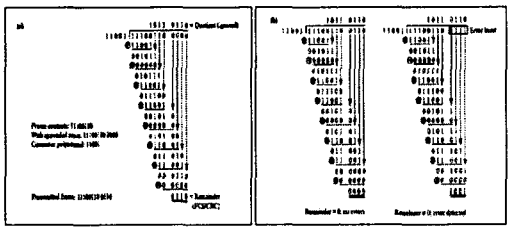

그림 7. CRC 연산 예제

4.4 CRC-32의 하드웨어 구현

CRC를 구현하는 여러 가지 방법이 있으나 본 논문에서는 면적을 작게 차지하고 고속으로 동작할 수 있는 LFSR(Linear Feedback Shift Register) 구조를 사용하여 구현하였다. 그림 8에 CRC-32의 하드웨어를 나타내었다.

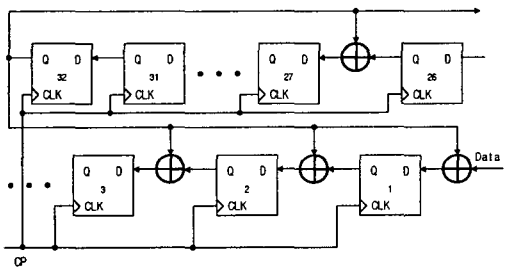

그림 8. LSFR 구조의 CRC-32 Data Path

4.5 하드웨어의 검증

설계된 회로를 FPGA에 구현하고, 구현된 회로를 검증하기 위해 PC를 사용하였다. PC와 FPGA 간의 인터페이스는 PCI 버스를 사용하였으며 이를 위해 필요한 회로가 추가되었다. 그림 9에 PC와의 인터페이스를 위한 회로가 추가된 전체 블록도를 나타내었다.

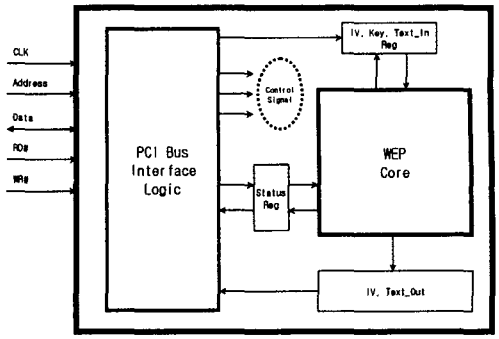

그림 9. PC 인터페이스를 위한 하드웨어 블록도

V. 결 론

무선 LAN에서의 WEP 프로토콜은 Option 사항으로 되어있지만 보안의 중요성과 필요에 의해 꼭 필요한 사항이다. 본 논문에서는 WEP 프로토콜을 하드웨어로 설계하여 FPGA 칩에 구현하고 PC와 인터페이스를 통하여 구현된 회로의 올바른 동작을 확인하였다.

본 논문에서 설계된 WEP 프로토콜은 무선 랜 환경에서 증가하고 있는 데이터 량과 전송속도로 기존의 소프트웨어 WEP 프로토콜로 인한 병목현상을 제거하고 성능 향상에 기여할 수 있을 것으로 판단된다.

감사의 글

본 연구에 IDEC 지원 소프트웨어를 사용하였습니다.

Reference

[1] IEEE Std 802.11-1997.

[2] James LaRocca, Ruth LaRocca, "802.11 DEMYSTIFIED", 2002

[3] 하창수, 최병윤, "40, 104, 128-Bit 길이 Key를 갖는 RC4 Pseudo Random Number Generator의 FPGA 구현", 2003년도 대한전자공학회 CAD 및 VLSI 설계 연구회 학술발표회 논문집, 2003, pp.160-163

[4] 하창수, 최병윤, "PLX 9050 칩을 사용한 PCI 인터페이스 환경의 구현", 2002 한국 신호처리 시스템학회 추계 학술대회 논문집 3권 2호, 2002, pp. 85-88

[5] 최병윤, "Verilog HDL을 사용한 디지털 설계 및 실습", 부산대학교 IDEC 강좌자료, 2003.1.20 -2003.1.22