Hardware 기반 ARIA 블록 암호를 활용한 MACsec Core 구현

박종욱*, 이준호*, 윤동욱**, 김호원*
*부산대학교, **부산대학교 블록체인 플랫폼 연구센터
jonguk@islab.re.kr, junho@islab.re.kr, dongwook@islab.re.kr, howonkim@pusan.ac.kr

Implement MACsec Core using Hardware-based ARIA Block Cipher

Park Jong Uk*, Lee Jun Ho*, Yun Dong Wook**, Kim Ho Won*
*Pusan National Univ, **Pusan National Univ Blockchain Platform Research Center

요 약

본 논문은 국가 기간망 및 스마트공장등 산업 전반에 MACsec을 활용하기 위해 KCMVP 인증 암호 ARIA-128/256-GCM을 활용하여 MACsec Core를 하드웨어 기반으로 구현 하였다. ARIA 암호는 AES 암호에 비해 하드웨어 구현상의 면적 및 속도의 이점을 가질 수 있으며 MACsec은 Layer3 보안인 IPsec 등이 완벽하게 제공해 주지 못하는 ARP, DHCP와 같은 Layer2 프로토콜에 대한 취약성에 대해 보호를 제공할 수 있다. 본 논문의 결과물은 Wireshark를 통해 MACsec Core가 제공하는 Point to Point 송수신 단에서 Layer2 보안성을 검증하였다.

Ⅰ. 서 론

MACsec의 경우 AES-128/256-GCM을 표준으로 사용하고 있다. 하지만 국가 기간망 및 공공기관 정보 통신망에서 사용하기 위해 KCMVP 인증 암호 중 블록 암호 ARIA-128/256-GCM을 사용하였다. 또한 스마트공장과 같은 외부와 독립되어 있는 망은 외부의 공격으로부터 안전하다고 생각되어 왔다. 하지만 4차 산업 혁명의 도입으로 내부망이 아닌 클라우드 환경에서 다양한 기술을 요구하게 되면서 하드웨어가 외부의 공격으로부터 노출되었다. 이의 해결책으로 VPN(Virtual Private Network)이나 IPsec과 같은 보안 솔루션을 이용하여 데이터를 보호할 수 있지만, 이는 OSI 모델의 Layer 3에서 동작하며, Layer 2 기반 프로토콜인 ARP나 DHCP에 대한 ARP Spoofing, DHCP Starvation과 같은 공격에 취약하다. MACsec은 Layer 2에 해당하는 데이터를 암호화하여 전송간에 중간자에 대한 공격으로부터 보호 할 수 있다.

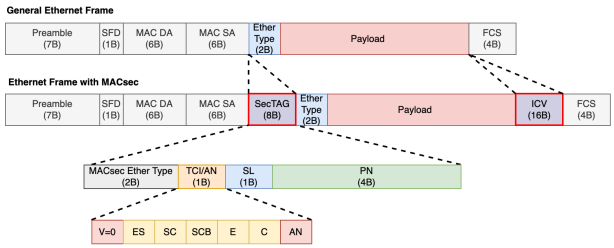
Ⅱ. 배경 지식

1) ARIA-GCM

MACsec에서는 AES-128/256-GCM을 표준으로 사용하며 XPN 구조를 확장으로 제공한다. 본 논문에서는 ARIA-128/256-GCM을 사용하였으며 국산 블록암호인 ARIA는 ISPN(Involutional SPN)구조를 가지며 Add Round Key, Substitution Layer, Diffusion Layer의 구성을 가진다. 데이터의 입출력 크기는 128bit, 라운드는 128/256 키 길이에 따라 12/16 라운드 수를 가진다. AES는 Round Key 생성을 미리 구현하여 저장하는 방식에 비해 ARIA는 Round가 진행되는 순간 연산을 통해 Round Key를 생성하여 하드웨어 구현상의 면적 관점에서 이득을 볼 수 있다. GCM의 경우 Mode Of Operation으로 블록암호를 이용하여 데이터 기밀성 및 무결성을 제공할 수 있게 해준다. GCM에서는 GHASH라는 Field 곱셈을 활용하며 Tag 값을 생성한다. 또한 데이터는 CTR Mode Of Operation을 사용함으로써 병렬성을 높일 수 있다.

2) MACsec 전송 Frame 및 SecTAG

MACsec은 Layer2 보안이므로 하드웨어 관점에서 데이터는 MII(Media Independent Interface)로부터 데이터를 받아 처리하게 된다. 따라서 Layer2 전송 Frame의 형태를 보면 Preamble와 Start Of Frame(SFD)도 받아서 처리를 해줘야하며 기본 Ethernet Frame에 비해서 SecTAG의 MACsec 설정에 대한 정보와 SCI(Secure Channel Identifier)의 정보, ICV(Integrity Check Value), CRC32가 추가로 전송되게 된다.



[그림 1] Ethernet/MACsec Ethernet Frame 구성

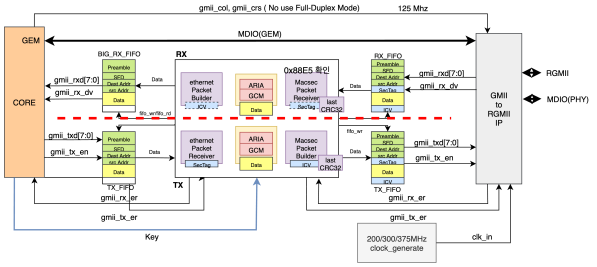
SecTAG 같은 경우에는 MACsec Ether Type과 TCI(Tag Control Information)/AN(Association Number), Short Length, Packet Number로 구성되는데 TCI/AN 내부에는 Point-to-Point 통신이 아닌 통신에 대해서 지원을 하기 위한 추가 구성들이다. Switch/Router의 내부망 통신에서는 SCI가 필요 없지만 라우터를 지나게 되면 MAC 및 IP Address의 변경이 필요하게 되어 MACsec을 다시 재구성하게 된다. 따라서 ARIA-GCM에 필요한 IV(Initialization Vector)값을 유지해주기 위해 사용된다. 이 과정에서 TCI/AN 내부의 ES, SC, AN 설정이 변경된다. AN 같은 경우에는 1개의 CA에서 최대 4개의 SA를 구분할 수 있으므로 4개의 SAK(Secure Association Key)를 가질 수 있다. PN 같은 경우는 ARIA-GCM에서 IV의 값을 Unique하게 구성되도록 IV의 하위 32bit를 구성할 때 사용된다.[1]

Ⅲ. 본론

본 논문에서 설명하는 Hardware 기반 MACsec Core의 전체 구조는 [그림 2]와 같으며 Key Agreement protocol은 Software를 통해 진행 되었으며 CSR(Control and Status Register) 형식을 통해 외부에서 Key, SecTAG 정보를 입력 받는 구조다. RX, TX를 위한 각각 2개의 FIFO, 그리고 각 2개씩의 CRC32 Core가 존재하며 각 1개씩의 ARIA-GCM 모듈로 구성되어 있다.

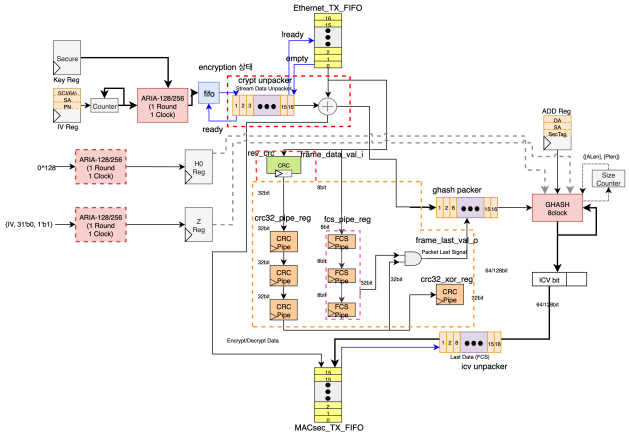
TX단에서는 Core로부터 전송하고자 하는 기본 Frame 형태를 Ethernet Packet Receiver가 Preamble & SFD, SA, DA 확인 및 MACsec의 동작을

확인하고 SecTAG 정보를 추가한다. MACsec Packet Builder 같은 경우에는 ICV와 FCS(CRC32) 값을 Frame에 추가한다. RX 같은 경우에는 외부네트워크로부터 MACsec Packet Receiver가 Preamble & SFD, SA, DA 확인 및 Ether Type을 확인하여 MACsec의 수신 과정을 동작할지 결정한다.



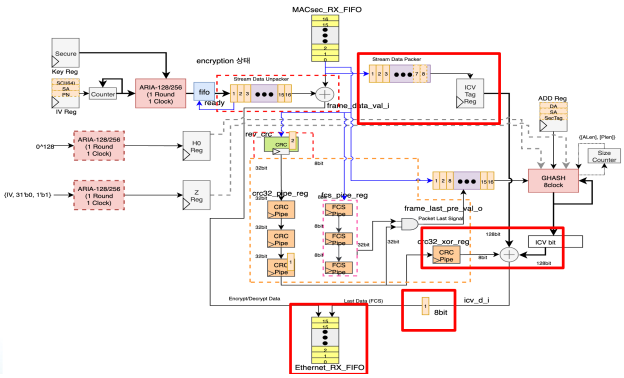
[그림 2] MACsec Core 구조도

세부적인 MACsec TX 구현은 [그림 3]와 같으며 ARIA 모듈의 경우에는 Key가 설정이 되는 동시에 내부 Key Initialization 및 GHASH 함수에 필요한 H값과 마지막 Tag값 연산에 필요한 Z(Y0)값을 생성한다. 그리고 SecTAG 및 SA를 통해 IV를 구성하였다면 들어오는 TX_FIFO 데이터에 상관없이 사전에 FIFO에 ARIA 결과를 write를 수행하게 된다. Crypt Unpacker가 FIFO의 128bit 데이터를 8bit 형식으로 분할하여 FIFO Data와 XOR 연산을 통해 CTR을 수행하고 8bit 형태로 다시 MACsec TX FIFO로 전송하게 된다. GHASH 모듈 또한 GHASH Packer의 8bit 데이터가 축적되어 128bit로 된다면 GHASH 연산을 수행한다. CRC 모듈의 같은 경우에는 Pipe 구조를 가지며, 실제 CRC32 연산과 Frame Data의 수신 결과 Reverse 형태를 갖기 때문에, Frame Data의 마지막을 데이터를 받아야 계산된 CRC32와 비교를 할 수 있다.



[그림 3] MACsec Core TX Schematic

MACsec의 RX 같은 경우에는 [그림 4]와 같으며 MACsec TX와 구조가 비슷하나 MACsec Frame형태인 경우 ICV 데이터 및 최종 FCS를 수신하기 위한 레지스터를 구성 하였다. ARIA-GCM의 Decryption과정에서 알고리즘대로라면 Data를 활용하여 Tag 값을 비교한 후, ARIA 연산을 통해 출력을 전송하게 된다. 이는 많은 Latency를 가지게 되고 이를 효율적으로 처리하기 위해 ARIA-GCM의 Encryption과 동일하게 동작하되 우리가 전송하고자 하는 FCS(CRC32)의 마지막 1byte값을 이용하여 처리하였다. 수신받은 ICV 값과 Decryption이 끝난 데이터의 ICV값, 마지막 1byte FCS 값을 XOR 한 결과의 ICV값과 수신한 ICV값이 같다면 0으로 바뀌기 때문에 원래의 FCS 값을 전송할 수 있다. 실제 이더넷 통신상에서 FCS의 값이 변경되면 수신하는 Switch/Rounter는 다른 수신자의 내부에서 Frame을 Drop 하는 방식을 활용하여 효율적인 처리를 할 수 있도록 구현하였다.



[그림 4] MACsec Core RX Schematic

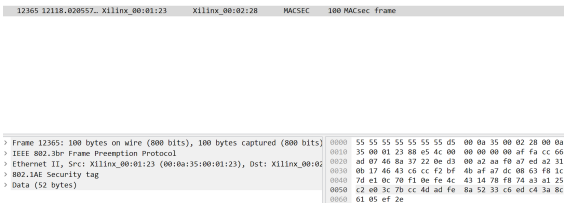
본 논문에서 제안하는 MACsec Core 구현을 위해 Xilinx Zynq Ultrascale+ XCZU7EV-2FFVC1156 칩이 내장되어 있는 ZCU104 FPGA Board를 사용와 Ethernet 다중 연결을 위한 확장 보드 Ethernet FMC(FPGA Mezzanine Card)를 사용했다. Ethernet FMC와의 MDIO(Management Data Input/Output) 통신 및 패킷 생성을 위한 Host Processor는 ZCU 104 FPGA Board에서 지원하는 PS(Processing System) 및 운영체 제는 Ubuntu 20.04 LTS를 이용하였다. 또한 MACsec Core 검증을 위한 블록 디자인을 구현 하였는데, Ethernet FMC의 Interface는 RGMII 이며, ZCU104 PS GEM의 Interface는 GMII를 사용하므로 GMII to RGMII IP를 활용하여 구성하였다.

[표 1] MACsec Core 구현에 따른 하드웨어 자원사용량

	LUT	FF	BRAM	MCMM
Utilized	8,662	5,773	2.5	2
Available	230,400	460,800	312	8
Percent	3.76%	1.25%	0.80%	25.00%

Ⅳ. 결론

FPGA 보드 운용환경에서 검증을 위해 FPGA 보드 2대를 이용하여 송신측, 수신측 환경을 구성하였다. 송신측 보드의 출력을 Ethernet Debugger와 Wireshark를 이용하여 분석한 결과 [그림 5]와 같이 성공적으로 암호화된 것을 검증하였다. 또한 수신측 보드에서는 성공적으로 복호화하여 전송한 원본 Frame이 Wireshark에 탐지 되는 것을 검증 하였다.



[그림 5] MACsec으로 암호화된 Ethernet Frame

본 연구에서는 ARIA-GCM을 이용한 FPGA 기반 MACsec Core를 구현 하였다. 다중 Ethernet 속도를 지원하기 위해 타이밍 최적화를 진행하여 최대 1000Mbps를 만족하였다. 또한, 하드웨어 자원은 최대 4%가량 사용되었으므로 저면적으로 구현 및 검증을 완료 하였다.

ACKNOWLEDGMENT

The EDA tool was supported by the IC Design Education Center(IDEC), Korea.

참 고 문 헌

[1] IEEE std. 802.1AE, Media Access Control(MAC) Security, IEEE, 2018