저전력 센서 네트워크 노드용 SHA-1 해쉬함수 구현 분석

최용제*, 이항록**, 김호원***
한국전자통신연구원 정보보호연구단

Analysis of implementation of SHA-1 hash function for Low power Sensor Network

Yongje Choi*, Hangrok Lee**, Howon Kim***

Infomation Security Research Division
Electornics and Telecommunications Research Institute
E-mail : *choiyj@etri.re.kr, **neogauss@etri.re.kr, ***khw@etri.re.kr,

Abstract

In this paper, we achieved software and hardware implementation of SHA-1 hash function for sensor network. We implemented the software to be compatible with TinySec. In hardware design, we optimized operation logics for small area of hardware and minimized data transitions of register memory for low power design. Designed the software and hardware is verified on commercial sensor motes and our secure motes respectively.

I. 서론

센서 네트워크는 차세대 네트워크로써 많은 연구가 진행되고 있으며, 이의 응용분야로는 환경정보의 관리, 응급의료 시스템, 군사적 목적의 관리, 물건의 추적 및 제고 관리등이 있다. 이러한 센서 네트워크는 제한된 에너지, 대역폭, 계산능력등의 자원들을 가지고 있는 센서 노드들 사이에 무선 네트워크를 형성하는 것을 의미하며, 일반적으로 센서 노드들은 단거리 영역에서 브로드케스트를 기반으로 통신한다. 이러한 센서 네트워크가 브로드케스트 방식의 무선통신을 사용한다는 사실에 의해서 보안에 취약성을 갖는다. 이러한 센서 네트워크가 가지고 있는 보안상의 문제점을 해결하기 위해서 Berkeley 에서는 무선 센서 네트워크를 위한 Link Layer Security Architecture 를 제안하였다. TinySec 이라고 불리우는 Link Layer Security Architecutre 는 기존의 end to end 방식의 보안 프로토콜(SSL, SSH, IPSec)들이 센서 네트워크 내에서는 비효율적인 것에 착안하여, SkipJack 대칭키 암호에 기반한 CBC mode 및 CBC-MAC 를 운영함에 의해서 Link Layer 상에서 데이터에 대한 기밀성, 무결성, 인증 기능을 제공하는 메커니즘 이다[2].

본 논문에서는 TinySec 에서 제공하는 SkipJack 알고리즘을 이용한 CBC-MAC 대신에, SHA-1 전용 해쉬함수를 이용하여 메시지 해쉬 및 MAC 연산을 수행할 수 있도록 관련된 보안 소프트웨어와 하드웨어 모듈을 구현하여, 센서 네트워크 환경에서 전용 해쉬함수의 적용 가능성을 검토하고자 한다.

II. 본론

SHA-1 해쉬함수[3]의 연산은 32 비트 기반으로 수행되어 일반적인 컴퓨터 시스템에서 효율적으로 구현되는 것으로 알려져 있다. 하지만, 센서 네트워크 시스템의 센서 노드들은 주로 8 비트 기반의 저전력 프로세서를 기반으로 하고 있기 때문에 이에 적합한 구현이 필요하다. 이를 위하여 본 논문에서는 SHA-1 해쉬함수 연산을 8 비트 연산으로 변형하여 구현하였다. 또한 센서 네트워크 시스템으로 적용이 용이하도록

TinyOS[1]를 이용하여 구현하였으며, TinySec 구조와 호환을 위해서, TinySec 에 정의되어 있는 security components 들 사이의 관계와 유사하게 해쉬 관련 component 와 interface 를 구현하였다[2]. TinyHash 는 해쉬 알고리즘에 기반한 메시지에 대한 무결성과 인증을 제공한다. 인증을 위해서 HMAC 기법을 적용하였으며, 무결성을 위해서 SHA-1 해쉬 알고리즘을 사용한다. 상위 어플리케이션에서 다양한 종류의 해쉬 알고리즘을 사용할 수 있게 하기 위해서, Digest 라는 component 를 해쉬 알고리즘을 대표하는 component 로써 구현하였다. 이러한 구현은 TinyHash 가 해쉬 알고리즘과 독립적인 모듈이 되기 때문에 다른 여러 해쉬 알고리즘을 TinyHash 구조에 포함시킬 수 있도록 하며, 구현된 여러 해쉬 알고리즘을 편리하게 사용할 수 있도록 한다. HMAC 은 TinySec 에 정의 되어 있는 MAC interface 를 그대로 구현하였다. 구현된 해쉬함수 보안 소프트웨어 모듈은 그림 1 의 보안 센서 노드에서 검증하였다. 이때의 성능은 표 1 과 같다.

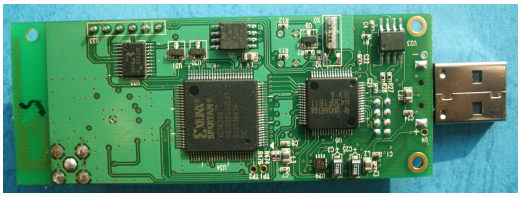

그림 1. 보안 센서 노드

표 1. SHA-1 S/W 보안 모듈 성능

연산 소요 시간	36 msec
RAM size	140 Byte
ROM size	3,644 Byte

SHA-1 알고리즘의 하드웨어 구현은 고속 연산에 중점을 두어 4 개의 덧셈기를 동시에 사용하여 라운드 함수를 한 클럭에 수행하도록 구현하며, 각 라운드에서 필요한 확장 메시지 값을 매 클럭마다 출력하기 위하여 쉬프트 레지스터를 이용한 구조로 구현한다[4,5]. 하지만, 4 개의 덧셈 연산과 쉬프트레지스터를 이용한 확장 메시지 연산은 많은 전력을 필요로 하며, 이는 저전력을 요하는 센서 네트워크 시스템에 적합치 않다. 이에 본 논문에서는 저전력 해쉬함수 구현을 위하여 해쉬 라운드 함수 연산을 위하여 하나의 덧셈기 만을 사용하여 4 클럭에 거쳐 라운드 함수가 연산되도록 구현하였다. 한 라운드를 4 클럭에 구현함으로써 확장 메시지를 메모리 구조로 바꾸어 연산할 수 있으며, 이는 확장 메시지 연산을 위한 전력 소모를 1/16 로 줄일 수 있다. 구현된 해쉬함수 하드웨어 모듈 역시 그림 1 의 보안 센서 노드의 FPGA 에서 구현하여 동작을 검증하였다. 또한 삼성 0.25u 공정에서 Synopsys 툴을 이용하여 소비전력을 측정하였다. 표 2 는 이때 동작 주파수에 따른 소비전력과 연산 시간을 보이고 있다.

표 2. SHA-1 H/W 보안 모듈 성능

	100kHz	1MHz	10MHz
소비 전력	19.5 uW	93 uW	1.68 mW
연산 시간	3.3 msec	330 usec	33 usec

Ⅲ. 결론

본 논문에서는 센서 네트워크에서의 전용 해쉬함수 적용을 위하여 소프트웨어 및 하드웨어 관점에서 각각 구현하고 이를 검증하였다. 전용 해쉬함수의 소프트웨어 구현은 현재 센서 노드들에서 충분히 수행할 수 있지만, 많은 자원 사용과 연산량으로 인하여 보다 light-weight 한 해쉬함수의 개발 및 적용이 필요할 것으로 보인다. 보안 하드웨어의 경우 타당한 소비전력과 성능을 가지는 구현이 가능할 것으로 보이며, 통신 시스템과의 유기적인 인터페이스를 취하여 보다 효율적인 시스템 구현이 가능할 것으로 보인다.

참고문헌

[1] David Gay, Philip Levis, Robert von Behren "The nesC Language : A Holistic Approach to Networked Embedded Systems" PLDI'03 June 9-11

[2] Chris Karlof, Naveen Sastry, David Wagner "TinySec : A Link Layer Security Architecture for Wireless sensor Networks" SenSys'04 November

[3] FIPS PUBS 180-2 " SECURE HASH STANDARD" U.S DoC/NIST, Augest 1,2002

[4] D. Zibin and Z. Ning. "FPGA Implementation of SHA-1 Algorithm" ASIC 2003. 1321-1324.

[5] H.E. Michail, A. P. Kakarountas. "Optimizing SHA-1 Hash Function for High Throughput with a Partial Unrolling Study" PATMOS 2005, LNCS 3728, PP. 591-600.