무선 센서네트워크에서의 소프트웨어 및 하드웨어 보안 모듈 성능 비교

오경희, 최용제, 최두호
한국전자통신연구원
e-mail : khoh@etri.re.kr

Performance Comparisons of Software and Hardware Implementations for Wireless Sensor Network

Kyunghee Oh, Yongjae Choi, Duho Choi
Electronics and Telecommunications Research Institute

요 약

무선 센서네트워크는 넓은 지역에 무선 네트워크로 설치된 센서들을 사용하여, 온도 습도 등의 환경을 감지하여 환경 감시, 대상 추적, 환자 모니터링, 군사적 목적 등 매우 다양한 분야의 서비스에 활용된다. 센서네트워크도 기존 네트워크와 마찬가지로 네트워크 보안 기능을 필요로 한다. 그러나 센서네트워크에 사용되는 장비가 사용할 수 있는 자원에 제약이 많아, 기존의 암호기술을 적용하는데 어려움이 있었다. 그러나, 최근의 연구결과들은 경량화 구현 기술을 적용하여 기존 네트워크에 적용하여 오던 보안 기술들을 센서네트워크에 적용하는 것이 실효성이 있다는 것을 보여준다. 본 논문에서는 대칭키 암호 기능과 비대칭키 암호 기능을 각각 소프트웨어와 하드웨어로 구현하여 성능을 측정한 결과를 비교한다.

1. 서론

센서네트워크는 넓은 지역에 무선 네트워크로 설치된 센서들을 통하여 감지된 정보들을 응용서비스 서버와 연동하는 기술로서, 환경 감시, 대상 추적, 환자 모니터링, 군사적 목적 등 매우 다양한 분야에 사용될 수 있다.

공개된 장소에 배치되어 자율적으로 네트워크를 형성하여 무선 통신을 하는 특성으로 인하여, 센서네트워크는 다양한 공격에 노출된다. 특히 도청에 의한 데이터 유출을 막기 위해서는 센서노드 간 암호 통신용 키 분배가 꼭 필요하다. 또한 허위 노드가 침입하여 허위 정보를 보내거나 라우팅 정보를 조작하는 등 다양한 공격이 가능하므로, 노드 간 인증도 필요하다.

센서노드의 하드웨어들은 작은 메모리에 제한된 연산 능력, 전력 사용의 제한이 있어, PKI와 같은 고급 보안 기능을 사용하는 것이 불가능한 것으로 여겨져, 보안 기능에 필요한 부하를 줄이기 위한 다양한 노드 인증 및 키 분배 방법들이 제안되었다. 그러나 이러한 방법들은 적용할 수 있는 환경에 제한이 있거나, 보안상 취약점을 어느 정도 감수하여야만 한다.

본 연구는 기존 네트워크에서 사용되어 온 공개키 및 대칭키를 사용하는 보안 기법들을 소프트웨어 및 하드웨어로 구현하여 센서네트워크에 적용하였다. 그리고 그 성능을 비교, 평가하여, 센서노드와 같이 한정된 자원을 사용하는 장비에서 강력한 보안 기능을 적용할 수 있는지 가능성을 검토한다.

†본 연구는 지식경제부 및 정보통신연구진흥원의 IT 산업원천기술개발사업의 기술개발사업 사업의 일환으로 수행하였음 [2009-F-055-01, 부채널 공격 방지 원천기술 및 안전성 검증기술 개발]

2. WSN 보안과 키 분배

센서네트워크에서의 보안 위협 특성을 고려할 때, 이에 특화된 보안 기법이 필요하다. 노드 인증과 암호화 통신을 위해 노드들이 사용할 키를 분배하는 기능은 꼭 필요하다.

자원제약 특성으로 인하여 비대칭 키의 사용 보다는 임의의 두 노드 사이에 대칭키를 분배하기 위한 연구들이 많이 이루어졌다. 만약 동일한 키를 여러 노드에서 같이 사용한다면, 공격자에 의한 노드 포획으로 센서네트워크 전체의 비밀 정보들이 유출되는 위험에 처해질 수 있다. 반대로 임의의 두 노드 간 통신에서 모두 다른 키를 사용한다면 센서네트워크 전체에서 모든 노드 수의 제곱에 해당하는 키가 필요하며, 센서네트워크와 같이 많은 수의 노드가 사용되는 환경에서 각 노드에서 필요한 모든 키들을 저장하고 관리한다는 것은 불가능하다. 이를 해결하기 위한 방법들을 살펴보면 다음과 같다.

임의 키 사전 분배 기법은 모든 가능한 키 공간에서 매우 큰 대칭키 풀을 임의로 선택하고, 각 노드들마다 여기서 일정한 개수의 키를 임의로 선택하여 노드의 키 저장 공간에 저장한다. 이후 두 노드가 키를 공유하기 위해서, 자신의 키 저장 공간에 있는 키들의 ID를 이웃 노드들에게 브로드캐스트하고, 이웃 노

드들에서 받은 키 ID 값과 자신이 가진 키들의 ID를 비교하여 공통된 키를 찾는다. 만약 동일한 키가 발견되면, 이 키를 사용하여 challenge-response 과정을 거쳐 세션키를 생성한다. 만약 공통된 키를 발견하지 못한다면, 이미 세션키를 생성한 다른 이웃 노드로 우회하여 경로키를 생성할 수 있다[1]. 이러한 방법은 birthday paradox 개념을 응용한 것으로, 완벽하지는 않더라도 임의의 두 노드 사이에 성공적으로 키를 생성할 확률이 매우 높다.

그러나 이러한 방법은 어느 한 노드가 공격자에게 노출되었을 때, 센서네트워크의 키 풀 일부가 공격자에게 누출되고, 따라서 노출된 키 풀의 양에 따라 센서네트워크 내의 다른 통신 내용들도 공격자에게 노출되는 단점이 있다. 이를 보완하기 위한 방법들 중 하나로, q-합성수 임의 키 사전 분배 기법이 있다. 위의 임의 키 사전 분배 기법에서 두 노드 사이에 단 하나의 공통키를 사용하여 세션키를 생성했던 것과는 달리, q개의 공통키를 찾은 다음 이를 조합하여 세션키를 생성한다. 만약 공격자가 어느 정도의 키들을 알고 있다 하더라도, 두 노드 사이에 사용된 q개의 공통키들을 모두 알고 있어야만 도청이 가능하게 된다[2]. 그러나 두 노드 사이에 공통키가 q개 미만이라면 세션키를 생성하는 것이 불가능하다는 단점이 있다.

직접적인 비밀 정보의 유출을 막기 위하여, 키와 관련된 행렬 정보들을 사용하는 Blom 스킴은 네트워크상의 어떤 임의의 두 쌍이라도 두 노드 사이의 비밀키 생성이 가능하며 노드 포획에 대해서도 이전의 방법보다 우수한 저항력을 가진다. $(\lambda+1)\cdot N$의 공개 행렬 G와 $(\lambda+1)\cdot(\lambda+1)$의 개인 행렬 D를 기본으로, $A=(DG)^T$를 비밀 행렬로 한다. 이때 D는 대칭행렬이어서, $AG=(AG)^T$의 특성을 가진다. 각 노드i는 A의 i번째 열과 G의 i번째 행을 저장하고, 노드 배치 후 노드i와 노드j가 키를 생성하고자 할 때, 서로 G의 행을 교환한 후, 각각 Kij=AiGj, Kji=AjGi 를 계산한다. Kij = Kji 이므로 두 노드는 동일한 키를 가지게 된다. Blom 스킴은 λ-security의 특성을 갖는다. 이는 개인 행렬에서 노출되는 열의 수가 λ 이하이면 행렬 D를 기반으로 생성된 다른 키들의 안전이 보장됨을 의미한다[3].

이러한 방법 이외에도 노드의 배치 정보를 적용하여 메모리 효율과 연결성을 높인 방법들도 제안되었다[4,5].

ZigBee[6]의 경우, Trust Center를 사용하여 키를 분배한다. 새로운 노드가 네트워크에 결합하기 위해서는 노드가 라우터에 association을 맺은 후, 이를 경유하여 Trust Center로부터 인증 및 링크키를 분배 받는 방식이다. 그리고 두 노드 사이의 링크키를 생성하는 경우에도 Trust Center가 개입한다. 이렇게 Trust Center를 사용한 인증 및 키 분배 방법은 인증 과정에서 아직 인증 받지 않은 노드로부터 여러 홉의 라우터를 거쳐 Trust Center까지 통신이 이루어져야 한다는 점에서, 노드의 수가 많은 센서네트워크에서는 인증 비용이 많이 소요되며, 공격 노드가 다수의 노드 ID를 가장하는 것과 같은 방법의 서비스 거부 공격을 할 수 있다는 점에 취약하다.

기존의 유무선망에서 널리 사용하여온 공개키 기반의 인증 및 키 분배 방식을 자원 제약이 있는 센서노드에 적용하는 것이 적합하지 않으리라는 견해가 많았으나, 최근의 연구에서는 공개키 기법을 적용하여 센서네트워크를 구현한 사례들이 많이 발표되었다.

공개키 암호 알고리즘을 사용하면 Trust Center와 같은 제3자의 개입 없이 두 노드가 직접 안전하고 신뢰성 있게 키를 생성할 수 있다. 타원곡선 암호 알고리즘 ECC를 사용하여 키를 분배하는 ECDH의 경우, 노드 A가 비밀키 a와 공개키 a*G를, 노드 B가 비밀키 b와 공개키 b*G를 가지고 있을 때, 서로 공개키를 알려준 후, 각각 자신의 비밀키를 사용하여 두 노드 사이에 공유되는 키 a*(b*G) = b*(a*G)를 연산할 수 있으므로, 공유키 a*b*G가 생성된다.

공개키 방식이 기존 컴퓨팅 환경에서 널리 사용되어 왔으나, 센서네트워크에 사용되는 디바이스들의 자원제약으로 인하여 연산량이 많은 공개키 암호화 방식은 부적합하리라 여겨졌다. 그러나 최근의 연구결과들은 공개키 암호화 방식이 센서네트워크 보안에서도 여전히 유효함을 보여준다. TinyOS 기반에 ECC 알고리즘을 구현한 TinyECC의 경우, ECDSA를 이용하여 전자서명에 1.6초, 검증에 2.0초가 소요된다[7]. 이는 대칭키 기반의 연산시간에 비하면 여전히 긴 시간이지만, 실재 응용에 키 분배 방법으로 구현되어 사용되더라도 충분히 실효성이 있는 시간으로 볼 수 있다.

또한, Truster Center를 이용하는 Kerberos 기반의 노드 키 분배 방법과 타원곡선 암호 알고리즘을 이용한 공개키 암호 프로토콜 ECDH/ECMQV 기반 키 분배 방법의 비교에서, Trust Center와 노드간의 연결 홉수가 3홉 이상이 되면 오히려 공개키 기반의 방식에 비하여 에너지 소모가 더 많아짐을 보여준다[8].

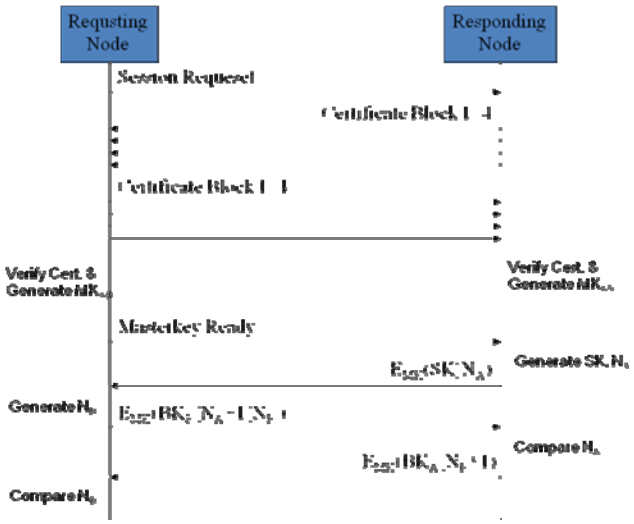
키 분배가 이루어졌다면, 이를 이용하여 데이터 프레임을 암호화하여 메시지를 주고받게 된다. 현재 ZigBee, TinyOS, 6LoWPAN 등, 대부분의 센서네트워크에서 IEEE 802.15.4 LR-WPAN을 MAC 계층으로 사용하고 있으며, 이 표준에 따라 AES 대칭키 알고리즘을 사용하여 프레임을 암호화하고 있다.

3. 암호 모듈 구현 및 비교

암호 모듈의 성능 분석을 위하여, ECC 기반 키 분배 프로토콜과 IEEE 802.15.4 MAC 프레임 암복호화 모듈을 각각 소프트웨어와 하드웨어로 구현하여 시험하였다.

그림 1은 TTA .KO-12.0092[9]에 따라 구현된 센서노드 간 인증 및 키 분배 프로토콜로서, 두 노드가 상호 인증 후 키를 생성하는 과정을 보여준다. 두 노드가 전자 서명이 포함된 공개키를 주고 받은 후, 서명을 검증하고 ECDH 키 교환 기법에 따라 마스터키 MK_{AB}를 생성한다. 다시 MK_{AB}를 사용하여 실제 데이터 암호화에 사용될 대칭키 SK를 생성하기 위한 키 생성 과정을 수행하게 된다. ECC 연산 모듈을 소프트

웨어 및 하드웨어로 각각 구현하여 이 프로토콜에 따라 수행한 결과를 측정하였다.



(그림 1) ECC 기반 키 분배 프로토콜

ECC 소프트웨어 모듈은 MSP430 MCU를 탑재한 TmoteSKY에 TinyOS 2.0을 탑재하고 수정된 TinyECC 1.0을 ECC 연산에 사용하였다. TinyOS가 단일 쓰레드 운영체제이기에, TinyECC에서 공개키 연산을 처리하는 중에는 센서노드가 다른 작업을 전혀 수행하지 못하는 문제점이 있다. 본 연구에서는 TinyECC 1.0을 기반으로, 공개키 연산 과정을 태스크 기반으로 수정하여, 공개키 연산 중에도 센서노드가 수행하여야 할 다른 작업들을 수행할 수 있도록 수정하였다.

ECC 하드웨어 모듈은 그림 2의 센서 노드로서 Xilinx FPGA에 ECC 연산 기능을 구현하였으며, MCU 등 다른 부속품과 구조는 TmoteSKY와 동일하다.

(그림 2) ECC 하드웨어를 장착한 센서노드

<표 1> ECC 기반 인증 및 키 분배 소요 시간

(단위:초)	S/W	H/W
인증서 교환	0.41	0.49
서명 검증 및 ECDH	6.72	0.93
세션키 생성	0.45	0.20
누적 합	7.58	1.62

표 1은 키 분배 과정에서 전송되는 프레임들을 무선 스니퍼를 사용하여 시간을 측정한 결과이다. ECC 소프트웨어 구현과 하드웨어 구현에서, 인증서 교환 및 세션키 생성과정은 동일하며, 서명 검증 및 ECDH 과정에서만 각각의 모듈이 사용된다. 서명 검증 및 ECDH 연산에서 하드웨어 모듈의 성능이 7배 이상 빠른 것을 확인할 수 있다

IEEE 802.15.4에 따르면 AES 대칭키 암호 알고리즘을 사용하여 프레임을 암호화한다. 이때 본 시험에서는 CCM 모드를 사용하는 security level 5를 적용하였다. 그리고, 소프트웨어 및 하드웨어 구현 모두 TmoteSKY를 사용하였다. 소프트웨어 구현의 경우, NanoQ+ 운영체제에서 AES 연산을 소프트웨어로 구현하였다. 하드웨어 구현은 TinyOS 운영체제에서 CC2420 칩 내부의 보안 기능을 사용하였다.

<표 2> AES S/W를 사용한 프레임 암호화 소요 시간

수행 절차	Payload 크기	연산 시간 (msec)
암호화	20	99.3
	107	263.8
복호화	20	99.5
	107	263.8

<표 3> AES H/W를 사용한 프레임 암호화 소요 시간

Payload 크기 (bytes)	평문 (msec)	암호문 (msec)	차이 (msec)
20	13.328	14.338	1.010
80	16.223	17.373	1.150
100	17.23	18.095	0.865
107	17.579	18.531	0.952

표 2는 각 MAC payload가 20, 107 바이트인 크기에 대하여 암호화 및 복호화 함수에서 소요된 시간을 측정한 값이다. 암호화와 복호화 과정 모두 20 바이트의 경우 약 100msec, 107바이트의 경우 약 263msec가 소요되었다. 암호화와 복호화에 거의 차이가 없는 것은 CCM 방식이 메시지의 암복호화에 상관없이 복호 연산 없이 AES 암호 연산만을 사용하는 특성 때문이다.

표 3은 CC2420 칩을 사용하여 MAC 프레임을 암호화하여 통신한 경우와 암호화 하지 않고 통신하였을 때의 소요 시간을 비교하여, 암호화에 따른 부하를 측정한 것이다. 암호 기능을 사용함으로써 payload 크기에 거의 상관없이 약 1msec의 추가 시간이 소요된 것을 확인할 수 있다.

이로써 대칭키 연산을 소프트웨어로 구현할 경우, 하드웨어를 사용한 경우에 비하여 100배 이상, 많게는 260배의 암호화에 따른 추가 지연이 있음을 보여준다.

4. 결론

센서네트워크의 키 분배에 있어, 사전 키 분배 방식은 노드가 공격자에게 포획되었을 때 여러 취약점이 발생한다는 단점이 있고, 제3자에 의한 키 분배의 경우에는 노드의 수가 증가함에 따라 확장성에 문제가 있다. 공개키 기법이 센서네트워크에 적합하지 않으리라는 기존의 우려와는 달리 실제 구현 사례들이

가능성을 보여준다. 공개키 기법의 경우 모든 센서노드들이 키 분배과정을 병렬적으로 수행함으로써, 본 연구 결과와 같은 수 초 정도의 소요시간은 실재로 감내할 수 있는 지연이라 여겨진다. 특히 보안 하드웨어를 사용하면 그 시간을 더욱 줄일 수 있다.

또한, 공개키 알고리즘의 복잡한 연산은 연산량뿐만이 아니라 프로그램의 코드 크기에도 부담을 준다. 그런데, 하드웨어 보안 모듈을 사용함으로써 코드 크기에 대한 부담을 줄일 수 있다.

대칭키 암호 연산의 경우, 소프트웨어 암호 연산에서 프레임의 크기에 따라 100~260msec 정도의 지연이 있음을 볼 수 있다. 이러한 성능으로는 초당 4~10 프레임 정도밖에 처리할 수 없게 된다. MCU가 암호화 연산뿐만 아니라 다른 작업들도 수행하여야 하므로, 실재 시스템에 적용하는 것은 적합하지 않아 보인다. 그런데, CC2420과 같이 현재 사용되고 있는 무선 통신 칩들에 암호 통신 기능이 하드웨어로 포함되어 있으며, 암호화로 인한 부하는 매우 적은 것으로 확인되었다.

본 연구를 통하여 공개키 기법과 대칭키 기법을 함께 사용하고 있는 기존의 유무선망처럼, 하드웨어 암호 모듈을 사용하여 센서네트워크를 안전하게 구현할 수 있음을 보였다.

참고문헌

[1] L. Eschenauer and V. D. Gligor. "A key-management scheme for distributed sensor networks," In Proceedings of the 9th ACM Conference on Computer and Communication Security, pages 41--47, Nov. 2002

[2] H. Chan, A. Perrig, and D. Song. "Random key predistribution schemes for sensor networks," In IEEE Symposium on Security and Privacy, May 2003

[3] R. Blom, "optimal class of symmetric key generation systems,"EUROCRYPT 84 workshop on advances in cryptology: theory. and application of cryptographic techniques, pp. 335-338, Dec. 1985, Paris

[4] W. Du, J. Deng, Y. S. Han, S. Chen, and P. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge," IEEE INFOCOM 04, Hong Kong. March, 2004

[5] D. Huang, M. Mehta, D. Medhi, L. Harn. "Location-aware Key Management for Wireless Sensor Networks," 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, 2004

[6] ZigBee Specification, ZigBee Document 053474r06, Version 1.0, ZigBee Alliance, June 27, 2005

[7] TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks, Ver 1.0, http://discovery.csc.ncsu.edu/software/TinyECC/, 11-02-2007.

[8] J. Johann Großschadl, A. Szekely, and S. Tillich, "The Energy Cost of Cryptographic Key Establishment in Wireless Sensor Networks," ASIACCS 2007, pp. 380–382. ACM Press, 2007.

[9] TTAK.KO-12.0092, USN 에서의 센서 노드 간 인증 및 키 분배 프로토콜, 한국정보통신기술협회, 2008.12