센서네트워크에서 인-네트워크 프로세싱을 위한 경량 키 관리 프로토콜

김경태, 김형찬, R. S. Ramakrishna
광주과학기술원 정보통신공학과
e-mail : {ktkim, kimhc, rsr} @gist.ac.kr

A Lightweight Key Management for In-network Processing in WSNs

Kyeong Tae Kim, Hyung Chan Kim, R. S. Ramakrishna
Department of Information and Communications,
Gwangju Institute of Science and Technology (GIST)

요 약

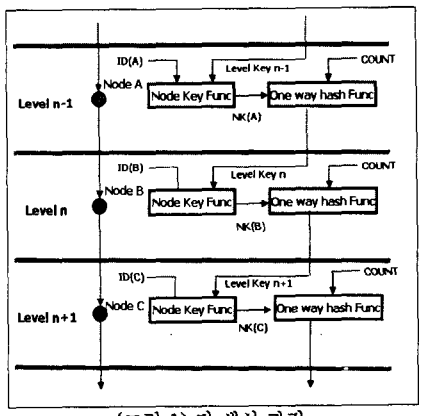
본 논문에서는 Wireless Sensor Networks(WSNs)에서 에너지 소모를 줄이기 위해 사용되는 In-network processing에 대하여 보안이 강화된 레벨 키 기반의 Infrastructure를 설계하여 노드의 전복 공격에 대해 안전한 패킷 포워딩을 보장하는 프로토콜을 제시한다. 이러한 계층적 구조를 가지는 보안 Framework는 노드의 추가 혹은 퇴거가 발생했을 때 Re-keying 비용을 획기적으로 줄일 수 있다. 시뮬레이션 결과, 전체 네트워크 중 전복된 노드가 40%를 차지할 때, 제안된 프로토콜을 사용하게 되면 약 3%의 추가적인 라우팅 오버헤드 비용으로 15%의 향상된 종단간 패킷 전송률을 보여준다. 또한 Re-keying을 할 때 OFT와 비교하여 통신비용을 현저하게 줄일 뿐만 아니라 서버의 도움 없이 키를 업데이트 하기 때문에 분산환경에 적합한 특징을 갖는다.

1. 서 론

WSNs에서 노드 사이에 메시지를 포워딩하는데 소모되는 에너지를 줄이기 위해 많은 연구자들이 Data aggregation에 관심을 갖기 시작했다. 계층구조를 갖고 데이터를 취합하는 Data aggregation과 동일한 이벤트를 검출했을 때 메시지 전송을 하지 않는 Passive participation 기술을 이용한 In-network processing은 향상된 Scalability을 제공할 뿐만 아니라 노드의 생명 연장에 크게 기여한다. 본 연구는 In-network processing에 보안이 강화된 가벼운 키 관리를 프로토콜을 제안한다. 제한된 리소스의 사용과 보안 요구사항인 개체 인증, 메시지의 기밀성 및 무결성을 보장하는 것 사이에는 항상 충돌이 발생하기 때문에 제시된 프로토콜은 이러한 두 가지 요소를 적절하게 만족시켜야 한다.

본 논문에서는 WSNs에서 In-network processing을 수행할 때 노드의 안전한 패킷 포워딩을 보장하고 노드의 추가 혹은 퇴거에 대한 Re-keying 비용을 획기적으로 줄이는 Framework를 개발하는데 초점을 맞춘다. 초기에 뿌려진 노드들은 라우팅을 형성함과 동시에 그림 1과 같이 어떤 레벨에 위치한다. 이것은 부모와 자식 간의 관계를 만들고 각 부모가 여러 다른 자식 노드들과 같은 레벨 키를 공유하는 트리 형태를 형성한다. 레벨 키는 일 방향 속성을 갖고 있는데, 상위 레벨(홉 카운트가 낮은)의 키로부터 하위 레벨의 키를 계산할 수 있지만, 반대의 작업은 불가능하다. 이것은 Data aggregation를 수행 하는 중에, 낮은 레벨의 조상 노드 A가 높은 레벨의 자손 노드 C의 키를 유추할 수 있게 하고, 종단간의 Pairwise key와 함께 사용되어 부모 노드 B가 메시지를 A에게 제대로 포워딩했는지를 확인할 수 있는 Secure Packet Forwarding(SPF) 메커니즘의 핵심이 된다.

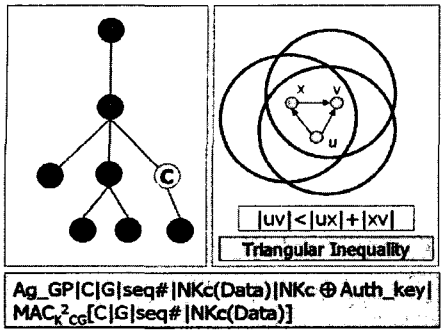
시뮬레이션 결과, SPF를 사용하게 되면 전복된 노드 가 전체



(그림 1) 키 생성 과정.

네트워크의 40%를 차지할 때, 종단간 패킷 전송 비율을 15%로 향상시키지만 추가적인 라우팅 오버헤드가 10% 필요하다. 하지만 개선된 Selective Secure Packet Forwarding(SSPF)을 사용하면 이를 3%까지 줄일 수 있다.

또한 노드가 추가되거나 퇴거할 때 마다 모든 지역에 Re-keying 메시지를 보내는 것이 아니라, 노드 이동이 발생한 해당 부모 노드가 자손에게만 새로운 레벨 키를 오직 한번만 뿌려준다. 이것은 기존의 OFT [1]와 비교했을 때, 전송하는 업데이트 메시지 개수를 약 40%정도 줄일 수 있다. 뿐만 아니라, LKH [1], OFT와 비교했을 때 중앙서버의 도움 없이 이동 노드의 부모 노드가 연산을 수행함으로써, Single Point Failure와 Congestion 문제를 해결하고 Latency를 줄여 분산환경에 적합한 특징을 가지고 있다.



(그림 2) Secure Packet Forwarding.

2. In-network Processing

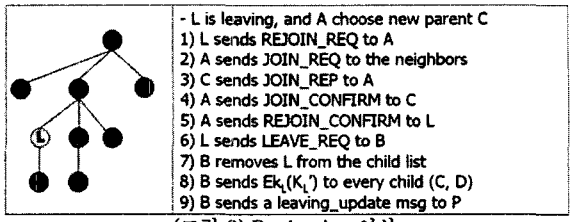
2.1 키의 생성

그림 1은 키 생성과정을 보여준다. 모두 세 가지 키가 사용되는데, 시드 키(Seed Key)로부터 레벨 키(Level Key)와 노트 키(Node Key)가 파생된다. 먼저, 시드 키는 베이스 스테이션(BS)으로부터 생성되며 보안 세션을 결정하는데 사용된다. 각 노드는 두 개의 함수를 관리하는데, 부모 노드로부터 자신의 레벨 키를 받으면, 이 때부터 두 개의 함수를 통해 키 생성이 시작된다. 첫 번째는 부모로부터 받은 레벨 키와 자신의 아이디로 노드 키를 생성하는 해쉬 함수이고, 두 번째는 생성된 노드 키에 임의의 카운트 값을 입력 값으로 해서 자식 노드의 레벨 키를 생성하는 일 방향 함수이다. Data-aggregation을 수행하기 위해서 센싱된 데이터를 자신의 노드 키로 암호화하는데, 하위 레벨의 노드는 상위 레벨의 노드로부터 받은 데이터 메시지를 해독할 수 있어야 한다. 이를 위해 상위 레벨의 노드는 하위 노드의 노드 키를 알고 있어야 하는데, 일 방향 함수는 이러한 요구조건을 만족한다. 반면에 하위 레벨의 노드는 상위 레벨의 레벨 키를 알 수가 없기 때문에, 부모 노드의 통신내용을 해독할 수 없다.

2.2 In-network Processing with Secure Packet Forwarding

그림 2는 Data aggregation 과정 중에 부모 노드 P가 C로부터 받은 패킷을 정확히 조상 노드 G에게 전달하는지를 모니터링 할 수 있는 패킷 구조를 나타낸다. 왼쪽으로부터 순차적으로 패킷 타입, 소스 ID, 목적지 ID, 시퀀스 넘버(Replay 공격 방지), 그리고 C의 노드 키로 암호화된 데이터 값이 포함 된다. C와 같은 레벨에 위치하는 A와 B는 C의 노드 키를 계산 할 수 있기 때문에 Passive participation을 허용하고, 부모 노드 P는 Auth_key [2]를 해독하여 자식 노드의 인증을 확인할 수 있다. 이 때, Auth_key와 함께 XOR된 노드 키(NKc)는 [3]에서 소개된 Triangular inequality를 무시할 수 있는 외부 공격 시나리오 대한 해결방법으로 사용된다. 즉, 그림 2에 오른쪽을 참조하면 노드 키를 알지 못하는 외부 공격 노드 x가 u인 것처럼 가장해서 v를 속일 수 없게 만든다. 마지막으로 조상 노드 G와 공유한 Pairwise 키(K^2_{CG})로 생성 된 MAC이 패킷 끝에 덧붙여진다. 이것은 노드 B가 패킷을 변조했을 때 G가 검출할 수 있도록 한다. G는 수신 받은 패킷이 변조되었는지 확인 한 후, 결과를 노드 P를 통해 C에게 ACK 메시지를 보낸다.

하지만 두 홉 떨어진 조상 노드에서 매번 패킷을 받을 때 마다



(그림 3) Re-keying 연산.

ACK 메시지를 자손에게 보내는 것은 라우팅 경로에 Traffic congestion을 야기하고 2배의 전송비용을 소모하기 때문에 이를 해결하기 위해 Selective Secure Packet Forwarding(SSPF)을 소개한다. 이것은 조상 노드가 매 수신마다 ACK 메시지를 보내는 대신에 Thresh값에 해당하는 개수만큼 패킷을 기다린 후 그 때까지 받은 데이터가 성공적이었음을 오직 하나의 ACK만을 전송함으로써 추가적인 Traffic을 크게 줄일 수 있다.

3. Re-keying

3.1 노드의 추가

그림 3에서 노드 D가 새롭게 추가 되었다고 가정하면, 노드 D는 인증절차와 Selection 공식을 이용하여, 적절한 부모 노드 B를 설정한 후, 노드 B, P와 Pairwise key를 생성한다 [2]. Re-keying 연산으로 레벨 키와 노드 키를 업데이트 시키는 알고리즘은 다음과 같다.

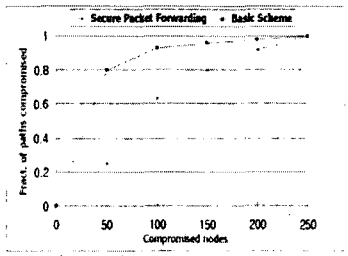
1) B generate new level key, K_L' .
   $K_L' = G_{COUNT}(NK_B)$
2) If (B is a leaf node) then
   B sends $E_{kBD}(KL')$ to D
   else
   B sends $E_{kL}(KL')$ to all its children, recursively.
3) B sends E_{kBP}(COUNT) to P

노드 키를 해쉬하는 일 방향 함수 G는 COUNT값을 입력 값으로 가지고 새로운 레벨 키(K_L')를 생성한다. 부모 노드 B는 K_L' 을 Pairwise key 혹은 기존의 레벨 키로 암호화 하여 자식 노드들에게 전송한다.

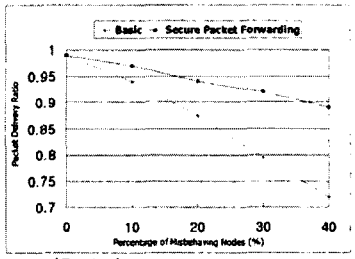
3.2 노드의 퇴거

그림 3은 노드 L이 퇴거할 때 적용되는 알고리즘을 순차적으로 기술하였다. 노드의 추가와 마찬가지로 퇴거하는 노드의 부모가 자식의 레벨 키와 노드 키를 재귀적으로 업데이트 시키고 있다. 일 방향 함수의 사용으로, 새롭게 추가된 노드는 오직 업데이트 된 키 값만을 알고, 퇴거 노드도 새로운 키 값을 계산할 수 없기 때문에 Backward secrecy[4]와 Forward secrecy[4]을 보장한다. 또한, 임의의 변수로 사용된 COUNT 값은 퇴거 노드에 알려지지 않기 때문에 OFT방법에서 문제점으로 지적된 Collusion attack을 해결한다 [5].

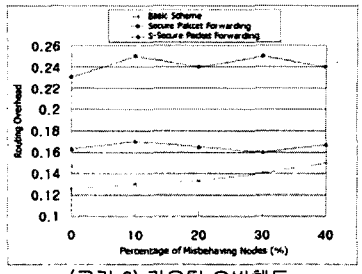
조상 노드가 부모 노드의 전복 공격을 감지할 경우 노드의 퇴거에 제시된 프로토콜은 Path repair 알고리즘 [2]과 함께 사용되어, 즉각적으로 전복된 노드를 고립시키고 새로운 라우팅을 형성하며 레벨 키와 노드 키뿐만 아니라 Pairwise 키를 업데이트 시킨다. 이것은 두 개의 연속된 전복 공격을 어렵게 만들며,



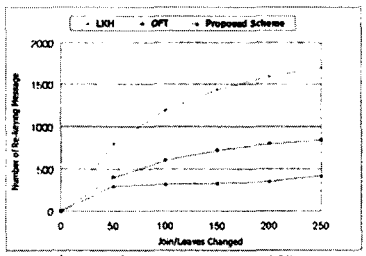
(그림 4) Network Fraction.



(그림 5) Packet delivery ratio.



(그림 6) 라우팅 오버헤드.



(그림 7) Re-keying 오버헤드.

안전한 경로로 종단 노드 간 통신이 이루어지도록 돕는다.

4. 시뮬레이션

4.1 Secure In-network Processing

우선 WSNs을 위한 시뮬레이터로 TinyOS 기반 위에 동작하는 TOSSIM을 사용하여 Data aggregation을 수행하는 중에 SPF를 사용했을 때 영향을 살펴보기로 한다. 그림 4는 250개의 노드로 이루어진 전체 네트워크에서 전복된 노드의 개수가 증가함에 따라서 종단간 경로가 끊어진 비율을 나타낸다. 각 종단 노드 사이에는 최적의 경로가 오직 하나만 존재하며, 중간에 전복이 발생했을 때, 경로가 끊어진다고 가정한다. 전체 네트워크 중에 50%의 경로를 차단하는데 31개의 노드 공격이 필요한 반면, SPF를 사용했을 때는 78개의 노드 공격이 필요하다. 전복된 노드의 숫자가 50개 일 때는 무려 55%의 성능 향상을 보여주고 있다.

그림 5는 소스에 의해 보내진 패킷의 개수와 목적지 노드에 정확히 도착한 패킷의 개수를 비율로 정의한 Packet Delivery Ratio를 나타낸다. 전복된 노드의 개수가 전체네트워크에 40%에 이를 때, SPF를 사용하지 않았을 경우와 비교해서 72%에서 89%로 15% 향상된 Throughput를 보여준다.

라우팅 오버헤드는 Bootstrapping, In-network processing, Re-keying, 그리고 Path repair을 포함하는 전체 데이터 전송량(bytes)와 SPF를 사용할 때 소모되는 전송량을 비율로 정의했다. 그림 6은 SPF를 사용할 때 전체 네트워크에서 약 10%의 추가적인 메시지 교환을 더 필요로 함을 보여주고 있다. 이것은 Data aggregation 과정 중 조상 노드가 매번 메시지를 수신할 때 마다 두 홉 떨어진 자손에게 보내는 ACK 메시지의 영향으로 기인한다. 이를 개선한 SSPF(Thresh = 3, Timeout = 0.15s)를 사용하게 되면 라우팅 오버헤드를 3%까지 줄일 수 있다.

4.2 Re-keying overhead

그림 7에서는 노드의 추가와 퇴거가 발생했을 때 Re-keying 연산을 위해 보낸 업데이트 메시지의 개수를 LKH, OFT, 그리고 제안된 프로토콜과 함께 비교한다. 시뮬레이션을 위해 전체 네트워크에 사용된 노드의 개수는 1024개이고, 제안된 프로토콜의 경우, 평균적으로 약 다섯 개의 이웃을 갖는 Topology를 구성했다. 제안된 프로토콜의 경우, 노드의 추가와 퇴거가 발생하는 트리 위치에 따라 업데이트 메시지의 개수가 차이가 나지만, 최악의 경우 OFT와 같고 평균적인 경우 OFT 보다 약 40% 향상된 Throughput을 보여준다.

5. 결론

본 연구에서는 WSNs에서 In-network processing을 위한 보안이 강화된 레벨 키 기반의 Infrastructure를 설계하고, 효율적인 그룹 통신을 위해 노드의 추가 혹은 퇴거가 발생했을 때 Re-keying 연산을 최소화 할 수 있는 가벼운 키 관리 프로토콜을 제시하였다. 시뮬레이션에 따르면 전체 네트워크에서 전복된 노드가 40%를 차지할 때, 제안된 프로토콜을 사용함으로써 약 3%의 추가적인 라우팅 오버헤드 비용으로 15% 향상된 종단간 패킷 전송률을 보여주었다. 또한 Re-keying을 수행할 때 LKH, OFT와 비교하여 현저하게 통신비용을 줄일 뿐만 아니라 OFT가 가지고 있는 Collusion attack 문제에 전혀 개의치 않았음을 확인하였다.

참고문헌

[1] A.T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," IEEE Transactions on Software Engineering, vol. 29, no. 5, pp. 444--458, 2003.

[2] 김경태, 김형찬, R. S. 라마크리쉬나, 고우이치 사쿠라이, "A Level-based Hop-by-hop Authentication Framework for In-network Processing in Wireless Sensor Networks," Abstracts of the 2006 Symposium on Cryptography and Information Security (SCIS2006), paper 4F2-3, pp. 328, 2006.

[3] S. Zhu, S. Setia: LEAP: efficient security mechanisms for large-scale distributed sensor networks. ACM Conference on Computer and Communications Security 2003: 62-72

[4] S. Mishra. Key management in large group multicast. Technical. Report CU-CS-940-02, Department of Computer Science, University of. Colorado, Boulder, CO., 2002.

[5] G. Horng, " Cryptanalysis of a Key Management Scheme. for Secure Multicast Communications" IEICE Trans. Commun., vol. E85-B, no. 5, pp. 1050- 1051, 2002.