무선 센서 네트워크에서의 안전한 네트워크 재프로그래밍 기법

신승목*, 최임성*, 김광조*
*한국정보통신대학교 공학부

Secure Clustering-based Network Reprogramming in Wireless Sensor Network

Sung-mok Shin*, Im-sung Choi* and Kwangjo Kim*
*Division of engineering, Information and Communications University.

요 약

무선 센서 네트워크 환경에서 센서 노드는 버그 수정 또는 업데이트가 요구 될 시에 원격으로 재프로그래밍이 가능해야 한다 [11]. 기존에 제안된 네트워크 재프로그래밍 기법인 Sluice [9]는 공개키 암호화 기법을 사용하기 때문에 제한된 리소스를 가진 센서 노드에서는 현실적으로 사용하기 어렵다 [6]. 이에 본 논문에서는 무선 센서 네트워크에서 적용할 수 있는 안전하고 효율적인 네트워크 재프로그래밍 기법을 제안한다. 제안 방식은 기존 방식인 Sluice에 비해 다음과 같은 장점을 가진다. 첫째, 인증방식을 공개키 연산 대신 랜덤-키 사전분배 방식을 사용하여 연산 오버헤드를 줄일 수 있다. 실제 노드에서의 수행된 평균 에너지 소비 결과를 통해 수행된 정성적 평가에서 제안방식은 Sluice의 총 소비 에너지의 4%정도만을 사용하여 업데이트의 기밀성을 지킬 수 있는 것으로 나타났다. 둘째, 프로그램 이미지를 암호화 하여 전송함으로써 업데이트의 기밀성을 보장한다. 셋째, 암호화를 위해 사용되는 그룹키의 분배와 생성 과정이 클러스터링 과정에 포함되기 때문에 키 생성을 위한 추가적인 통신비용이 발생하지 않는다.

I. 서론

무선 센서 네트워크(Wireless Sensor Network, 이하 WSN)는 저가의 제한된 리소스를 가지고 동작하는 수많은 초소형 센서노드들의 집합이다. 환경 정보를 센싱하여 얻은 정보를 가공 및 전송해야하는 센서 노드는 그와 같은 작업들을 수행할 수 있게 해주는 운영체제 및 응용프로그램의 탑재가 요구된다. 그러나 해당 프로그램의 버그 수정이나 업데이트가 요구될 시에, WSN의 특성 상 한번 대상 지역에 배치된 센서 노드의 프로그램을 물리적으로 접근하여 업데이트 하는 방식이 일반적이나, 대규모 또는 접근이 어려운 지역에 배치되는 경우 물리적인 업데이트는 불가능하다.

이를 극복하기 위해 네트워크를 통한 원격에서의 노드 재프로그래밍 기법 [9, 10, 12]들이 제안되었다. 이들 중 안전하게 설계된 기법 중의 하나인 Sluice [9]은 전송되는 업데이트 코드 이미지의 인증 [5]과 무결성에 초점을 맞추어 설계되었다.

그러나 Sluice는 업데이트 이미지의 인증을 위하여 노드의 개수만큼의 공개키 연산을 해주어야 하기 때문에 센서 노드에서는 사용하기에 부적합하다. 또한 업데이트 이미지의 인증 및 무결성은 보장하는 반면, 기밀성은 보장하지 못하고 있다. 만약 공격자가 통신 채널 모니터링을 통해 프로그램 이미지를 획득할 수 있다면 이를 이용해 다양한 공격 방식을 위한 정보 획득이 가능하다 [2, 8].

본 논문에서는 기존의 인증 및 무결성을 보장해주는 동시에 기밀성 또한 보장해줄 수 있는 네트워크 재프로그래밍 기법을 제시한다. 제

안방식에서는 기존 방식의 단점을 보완하기 위해 다음과 같은 사항을 개선하였다.

첫째로 기존 방식에서 사용되었던 공개키 연산을 없애고 랜덤-키 사전분배 방식을 사용하여 공개키 연산의 오버헤드를 제거하였다.

둘째, 블록 암호 기법을 사용하여 전송되는 프로그램 이미지를 암호화하여 공격자가 도청한 정보의 내용을 확인할 수 없게 한다. 이를 위해 제안 방식에서는 두 개의 그룹키가 초기화 과정을 통해 분배된다.

셋째, 제안 방식에서는 업데이트 암호화를 위해 사용되는 그룹키의 분배와 생성 과정이 클러스터링 과정에 포함되기 때문에 그룹키 생성을 위한 추가적인 통신비용이 발생하지 않는다.

본 논문의 내용 구성은 다음과 같다. 2장에서는 기존의 네트워크 재프로그래밍 기법과 그 문제점에 대해 살펴보고 제안 방식에서 사용된 기법에 관해 설명한다. 3장에서는 랜덤-키 사전분배 방식과 블록 암호 기법을 사용해 개선된 프로토콜을 제안한다. 4장에서는 제안된 프로토콜의 보안 평가에 대해 살펴보고 5장에서 결론을 맺는다.

II. 관련 연구

본 장에서는 WSN에서의 안전한 노드 클러스터링을 수행해주는 SecLEACH [7]알고리즘과 그 기반이 되는 랜덤-키 사전분배 방식 [3]에 대해 알아본다. 또한 기존에 제안된 네트워크 재프로그래밍 기법인 Sluice에 대해서 살펴보고 이를 기반으로 개선된 알고리즘을 제안한다.

2.1 랜덤-키 사전분배 방식

랜덤-키 사전분배 방식은 Eschenauer와 Gligor [3]에 의해 처음 제안되었다. 본 방식은 3단계의 과정을 거쳐 노드 간의 안전한 인증을 보장한다. 첫 번째, 키 사전분배 과정에서 각 노드는 대상 지역에 배치되기 전에 복수의 키를 가진 키 풀(pool)로부터 랜덤으로 선택된 키의 집합을 할당받고, 두 번째, 공유키 발견 과정에서 모든 노드는 자신들의 키 집합의 키 ID를 브로드캐스트한다. 이를 통해 노드들은 어떤 이웃 노드가 자신과 키를 공유하고 있는지를 알게 되고, 공유키를 사용하여 안전한 링크를 설정하게 된다. 세 번째, 경로-키 설정 단계에서 키를 공유하고 있지 않은 노드 쌍들은 그들만의 유일한 키를 생성하여 나누어 가진다.

2.2 SecLEACH [7]

WSN에서의 안전한 클러스터링을 담당하는 SecLEACH는 기존의 LEACH 기법 [13]에 랜덤-키 사전 분배 방식을 적용하여 보안성을 높인 기법이다. 기존의 LEACH는 클러스터에 조인하는 노드들의 인증을 수행하지 않았다. 그러나 SecLEACH에서는 노드에 사전 분배된 키 리스트를 사용하여 브로드캐스트함으로써 자신의 주변 노드들과 공유하는 키를 이용하여 인증에 사용되는 MAC 정보를 생성할 수 있다.

[표 1] 기호 표기

기호	설명
A_i, CH, BS	일반 노드, 클러스터 헤더, 베이스 스테이션
ϑ	망 내의 모든 노드의 집합
$\Rightarrow, \rightarrow$	브로드캐스트, 유니캐스트
id_x	노드 x의 id
adv	CH 선출 광고 메시지
join_req	클러스터에 조인한다고 알리는 메시지
k_{id}	키 집합 내의 키의 id
$k_{[r]}$	id r에 해당하는 대칭키
R_x	노드 x의 키 링의 키 아이디의 집합
$MAC_k(msg)$	키 k를 사용하여 계산된 MAC

1. H $\Rightarrow$ ϑ : $id_H, nonce, adv$
   A_i : $r \square (R_H \cap R_{A_i})$인 r을 선택
2. $A_i \rightarrow$ H : $id_{A_i}, id_H, r,$
   $join\_req,\ MAC_{k_r}(id_{A_i} \mid id_{CH} \mid r \mid nonce)$
3. H $\Rightarrow$ ϑ : $id_H, (..., < id_{A_i}, t_{A_i} >, ...), sched$

[그림 1] Setup 단계

4. $A_i \rightarrow$ H : $id_{A_i}, id_H, d_{A_i},$

$MAC_{k_r}(id_{A_i} | id_{CH} | d_{A_i} | nonce + j)$

5. H $\Rightarrow$ BS : $id_H, id_{BS}, F(...,d_{A_i},...),$

$MAC_{k_H}(F(...,d_{A_i},...) | C_H)$

[그림 2] Steady-state 단계

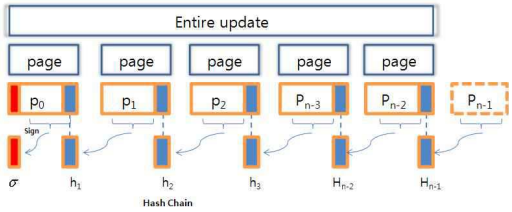
SecLEACH는 클러스터 셋업을 위한 setup 단계와 센서 정보를 취합하여 기지국(Base Station, 이하 BS)에게 전달하는 Steady-state 단계로 구성된다.

Step 1에서 CH는 자신이 CH로 선출되었다는 것을 모든 노드들에게 알리기 위해 자신의 id와 nonce 값, adv 메시지를 브로드캐스트한다. 브로드캐스트 메시지를 받은 노드들은 메시지의 전력 세기를 측정하여 자신과 가까이 위치하는 노드를 CH로 정하는 방식을 취한다. 브로드캐스트 메시지를 받은 후, 일반 노드는 자신의 노드 id와 CH 후보로부터 수신된 노드 id의 비교를 통해 두 노드 사이에 공통으로 공유하는 키를 찾는다. 그리고 Step 2에서 일반 노드는 자신의 id, CH의 id, 사전 공유키와 nonce값을 사용하여 MAC값을 계산한 후 join_req 메시지와 함께 전송한다. 이 과정을 통해 하나의 클러스터가 생성되게 되고, CH는 MAC값을 인증하여 올바른 노드라면 적절한 타임 스케줄을 분배하여 센싱 데이터를 받을 준비를 하게 된다.

2.3 Sluice

Sluice는 업데이트 이미지의 인증과 무결성에 초점을 맞춘 네트워크 재프로그래밍 기법이다. 업데이트 이미지가 신뢰받는 소스라는 것을 보장하기 위해 Sluice에서는 공개키 암호 기법과 일방향 해쉬 체인을 사용한다.

그림 3은 Sluice의 업데이트 검증 과정을 보여준다. MNP, Deluge [10, 12]와 같은 기존의 네트워크 재프로그래밍 프로토콜은 전체 업데이트를 페이지의 집합으로 나누어 전송한다. Sluice에서는 전체 업데이트를 페이지의 집합으로 나눈 뒤, 각 페이지의 해쉬 값을 그 이전 페이지의 payload에 함께 싣는다. (예를 들어 page p_{n-1}의 해쉬 값 n_{n-1}은 이전 페이지의 payload와 함께 p_{n-2}에 함께 실어진다. 마지막 페이지는 해쉬 값을 보내지 않고, 첫 번째 페이지 p_0는 페이지 p_1의 해쉬 값 h_1과 전자 서명 σ를 함께 싣는다.



[그림 3] Sluice - 업데이트 과정

해쉬 체인의 헤드인 h_1은 σ로 전자 서명된 페이지의 한 부분이므로 전송된 페이지가 신뢰받는 대상이라는 것을 보장해준다. 그러므로 전자 서명은 업데이트 이미지가 신뢰받는 BS로부터 전송되었다는 것을 인증할 수 있다. 한번 h_1를 신뢰할 수 있으면 나머지 페이지의 해쉬 값들은 one-way hash chain의 특성에 의해 무결성을 확인할 수 있다.

2.4 Sluice의 문제점

- 공개키 연산의 오버헤드

Sluice에서는 BS로부터 전송받는 업데이트 이미지의 신뢰성을 보장하기 위해 공개키 암호 기법을 사용한다. 그러나 제한된 리소스를 가진 센서 노드에서 공개키 연산을 사용하는 것은 센서 네트워크 전반으로 큰 오버헤드를 발생시킨다. Sluice에서 사용된 전자 서명은 인증하는 데에 30~35초가량의 시간이 소비되는 것으로 나타났다 [9]. 이는 소규모 네트워크에서는 가능하나 수백, 수천 개의 WSN에서는 사용 불가능하다.

- 업데이트 이미지의 노출

기존 네트워크 재프로그래밍 방식에서는 업데이트 이미지의 인증과 무결성에 초점을 맞추어 설계되었다. 그러나 기존 방식은 업데이트 이미지의 기밀성에 대해서는 고려하지 않았다. 만약 공격자가 노드간의 통신을 모니터링하여

업데이트 이미지를 획득한다면 센서 노드에 업데이트 되는 프로그램의 내용에 대해 알 수 있게 된다. 이러한 공격은 많은 노력을 들여 개발한 벤더의 손실뿐만 아니라, 여러 가지 보안 위협을 만들어 낼 수 있다. 예를 들어 도청을 통해 획득된 프로그램을 역공학 과정을 거쳐 공격 목적을 가진 프로그램으로 만들어 배포할 수 있다. 또한, 이를 통해 센서 네트워크의 동작 방식, 인증 메커니즘 등에 대해서도 알아 낼 수 있을 것이다. 업데이트 이미지에 비밀 정보(예를 들면, 인증을 위한 정보 및 키)가 실려 있을 경우 그 정보에 대한 노출 또한 큰 문제가 될 수 있다. 따라서 전송되는 업데이트 이미지의 암호화가 필요하다.

그러나 Sluice 프로토콜은 오직 BS가 개인키와 공개키를 소유하고 있고 각 노드들은 오직 공개키 만을 보유하고 있기 때문에 위와 같은 암호화를 수행할 수 있는 공유키가 존재하지 않는다.

III. 제안 방식

본 장에서는 기밀성과 효율성을 고려한 네트워크 재프로그래밍 기법에 대해 제안한다. 제안 방식에서는 업데이트 이미지의 암호화를 통해 Sluice 프로토콜에서 고려하지 못했던 업데이트 이미지의 기밀성을 고려하였다. 또한 공개키 연산 대신 랜덤-키 사전분배 방식을 통한 그룹키 분배를 통해 공개키 연산의 오버헤드를 제거하였다.

3.1 기호 표기

[표 1] 추가 기호 표기

기호	설명
update_Adv	새로운 업데이트 알림 메시지
join_req	클러스터 조인 메시지
SK_req	공유키를 요청하는 메시지
ACK	전송 확인 메시지
SK_{intra}	클러스터 멤버들 사이에 공유되는 암호화 용 대칭키
SK_{inter}	BS와 CH사이에 공유되는 암호화 용 대칭키
$Ek_k(msg)$	키 k로 암호화된 msg
Updates	바이너리 이미지

3.2 프로토콜 설명

제안 방식은 3개의 과정으로 나뉜다. 첫 번째, Intra-Cluster setup 과정은 CH와 센서 노드 간에 공유할 수 있는 그룹키를 분배하는 과정이다. 두 번째, Inter-Cluster 초기화 과정은 BS와 CH 간에 공유하는 그룹키를 분배하는 과정이다. 마지막으로 업데이트 과정은 위에서 분배된 그룹키를 사용하여 업데이트를 암호화하여 전송하는 과정이다.

1. BS $\Rightarrow \vartheta$: update_adv 메시지 전송
2. CH $\Rightarrow \vartheta$: $id_{CH}, nonce, adv$
   A_i : $k_{id} \square (R_{CH} \cap R_{A_i})$인 r을 선택
3. A_i → CH : $id_{A_i}, id_{CH}, k_{id}, join\_req,$
   $MAC_{k_r}(id_{A_i} \mid id_{CH} \mid k_{id} \mid nonce)$
   CH : 가장 많은 노드들이 공유하는 사전키를 그룹키로 정한다.
   $k_{id}^{\square} \square (R_{A_1} \ldots \cap \ldots R_{A_i})$ 인 $r^{\square}$을 선택
4. CH $\Rightarrow$ A_i : $id_{CH,} EK_{k_{id}^{\square}}(SK_{intra})$
5. A_i $\Rightarrow$ CH : ACK message

[그림 4] Intra-Cluster 단계

6. CH → BS : ACK message
7. BS $\Rightarrow$ CH : $id_{BS}, nonce$
   CH : R_{CH}에서 키 ID k_{id}를 선택
8. CH → BS : $id_{CH}, id_{BS}, k_{id}, SK\_req,$
   $MAC_{k_r}(id_{CH} \mid id_{BS} \mid k_{id} \mid nonce)$
   BS : 공유 세션키 SK_{inter}를 생성
9. BS → CH : $EK_{k_{id}^{\square}}(SK_{inter}), id_{BS}$

[그림 5] Inter-Cluster setup 단계

SK_{inter} 는 BS에서 생성되어 사전 분배된 키 k_r을 사용하여 암호화 되어 CH에게 전송된다. 이때 전송되는 그룹키는 BS에서 생성된 128비트의 난수라고 가정한다.

10. CH → BS : 그룹키 전송 확인 ACK전송
11. BS $\Rightarrow$ CH : $EK_{SK_{inter}}(Updates)$
12. CH $\Rightarrow$ A_i : $EK_{SK_{intra}}(Updates)$

[그림 6] Code dissemination 단계

IV. 안전성 분석

본 장에서는 제안 방식과 Sluice의 비교를 통하여 개선된 점에 대해서 알아본다.

4.1 효율적인 네트워크 재프로그래밍

표 2에서는 제안 방식과 기존방식의 소비 에너지에 대한 정성적 평가를 수행하였다. 표에서 n은 총 노드 개수, t는 CH의 개수, i는 전송되는 페이지의 개수를 의미한다. 센서 노드에서의 블록 암호화 연산 수행에 소비되는 에너지양에 관한 결과인 [14]를 참고하여 각 암호화 연산에 요구되는 CPU 사이클 및 에너지 소비량을 정하였다. 센서 노드에서 한 CPU 사이클이 수행될 때 소비되는 전력량은 평균 1.26(nJ)이다. KASUMI 알고리즘을 사용했을 경우, 암호화에 드는 CPU 사이클은 568회, 복호화는 576회이다. 반면, 공개키 알고리즘을 사용하였을 때, 서명 생성에 드는 CPU 사이클은 18757회, 서명 검증은 480427회의 CPU 사이클이 필요하다. Sluice의 경우, n회의 공개키 복호화 연산이 수행된다. 제안방식의 경우, 각 CH에서 t*i번의 암호화가, 그리고 각 CH와 일반 노드에서 (t*i)+(n*i)번의 복호화가 수행된다. 그림 7은 표 2에서 도출된 소비 전력량을 노드 개수의 증가 추이를 그래프로 나타낸다. 그림 7에서 제안방식이 소비 전력 부분에서 WSN에 쓰이기에 적절하다는 것을 보여준다.

[표 2] 소비 에너지 비교

	Sluice	제안방식
암호화 횟수	0회 (BS)	t*i회 (BS)
복호화 횟수	n회	(t*i) + (n*i)회
CPU cycle	480427n	((t*i)*568*2) + ((n*i)*576)
소비 에너지	480427n * 1.26(nJ)	(((t*i)*568*2) + ((n*i)*576))* 1.26(nJ)

4.2 인증, 무결성 및 기밀성 보장

제안 방식에서는 기존 방식에서 사용되었던 공개키 암호 대신, 랜덤-키 사전 분배 방식을 사용하여 노드 및 업데이트의 신뢰성을 인증하고. 또한 일방향 해쉬 체인을 통한 무결성 검증도 그대로 제공한다. 더 나아가, 업데이트의 기밀성을 제공함으로써 공격자가 도청에 성공하더라도 획득한 정보를 알아볼 수 없게 한다.

[그림 7] 소비 에너지 그래프

4.3 노드 탈취에 대한 대응

노드 탈취는 대규모 무선 센서 네트워크의 성공적인 배치를 위협하는 심각한 보안 문제이다 [1]. 제안 방식은 BS와 CH 사이에 암호화 전송을 위한 공통의 그룹키를 보유하고 있기 때문에 하나의 클러스터 헤더 노드가 탈취된다면 전체 센서 네트워크의 보안이 위협받을 수 있다.

이러한 노드 탈취에 대해 [4]에서는 이웃 노드간의 전송 전력 감지를 통해 탈취 노드에 대한 탐지 기법을 제시하였다. 이를 통해 업데이트를 브로드캐스트 하는 과정에서의 탈취 노드 탐지가 가능할 것이라 예상된다. 이에 대한 과제는 향후 연구로 남기도록 한다.

V. 결론

본 논문에서는 기존에 제안된 Sluice를 기반으로 업데이트의 기밀성을 보장하며 효율적으로 업데이트를 전송할 수 있는 개선된 무선 센서 네트워크 재프로그래밍 기법을 제안하였다. 제안된 기법은 앞서 언급한 Sluice의 단점을 해결하기 위해 랜덤-키 사전분배를 통한 클러스터링을 통해 그룹키를 안전하게 분배하고, 그 그룹키를 사용하여 전송되는 데이터를 암호화하여 안전한 노드 재프로그래밍을 가능하게 해준다.

[참고문헌]

[1] C. Hartung, J. Balasalle, and R. Han, “Node Compromise in Sensor Networks: The Need for Secure Systems.” Technical Report, 2005

[2] C. Karlof, and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures." Elsevier's AdHoc Networks Journal, Sep 2003, pp 293 - 315

[3] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks.” IEEE Symposium on Security and Privacy, 2003

[4] H. Song, L. Xie, S. Zhu, and G. Cao, “Sensor node compromise detection: the location perspective.” Proceedings of the International Conference on Wireless Communications and Mobile Computing (IWCMC '07), Honolulu, Hawaii, USA, Aug 2007, pp 242

[5] I. Krontiris, and T. Dimitriou, “Authenticated in-network programming for wireless sensor networks.” 5th International Conference on AD-HOC Networks & Wireless (Adhoc-Now ‘06), Ottawa, Canada, Aug 2006

[6] Krzysztof Piotrowski, Peter Langendoerfer, and Steffen Peter, "How public key cryptography influences wireless sensor node lifetime." Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks (SASN 2006), Alexandria, Virginia, USA, 2006

[7] L.B. Oliveira, H.C. Wong, M. Bern, R. Dahab, and A.A.F. Loureiro, "SecLEACH: A Random Key Distribution Solution for Securing Clustered Sensor Networks." 5th IEEE International Symposium on Network Computing and Applications (NCA'06), Cambridge, MA, July 2006, pp 145-154

[8] Patrick E. Lanigan, "Secure Network Reprogramming for Sensor Networks." Master's thesis, Information Networking Institute, Carnegie Mellon University, Pittsburgh, PA, May 2006.

[9] P. E. Lanigan, R. Gandhi, and P. Narasimhan, "Sluice: Secure dissemination of code updates in sensor networks." ICDCS, Lisbon, Portugal, Jul 2006.

[10] P. K. Dutta, J.W. Hui, D. C. Chu, and D. E. Culler, "Securing the Deluge network programming system." Information Processing in Sensor Networks, Apr 2006

[11] Q. Wang, Y. Zhu, and L. Cheng, "Reprogramming wireless sensor networks: Challenges and approaches." IEEE Network Magazine, May/June 2006, pp 48 - 55

[12] S. S. Kulkarni, and L. Wang, "MNP: Multihop network programming for sensor networks." International Conference on Distributed Computing Systems, June 2005, pp 7 - 16

[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks." IEEE Hawaii International Conference on System Sciences, Jan 2000, pp 4 - 7

[14] Y. Law, J. Doumen, and P. Hartel, " Survey and benchmark of block ciphers for wireless sensor networks." Technical Report TKCTIT-04-07, Centre for Telematics and Information Technology, University of Twente, the Netherlands, 2004