무선 센서 네트워크 환경에서의 안전한 키 분배 기법

장지용[1], 김형진[1], 권태경[2], 송주석[1]
[1]연세대학교 컴퓨터과학과, [2]세종대학교 컴퓨터공학과

Secure Key Pre-distribution Scheme for Wireless Sensor Network

Ji-Yong Jang[1], Hyoung-Jin Kim[1], Taekyoung Kwon[2], JooSeok Song[1]
[1]Dept. of Computer Science, Yonsei University
[2]Dept. of Computer Engineering, Sejong University

요 약

센서 네트워크는 앞으로 다방면에서 활용되어질 것으로 기대되고 있다. 그러나 센서 노드들은 악의적인 공격자에 의해 통신내용이 수집되고 공격을 받거나 조작될 수 있는 적대적이고 안전하지 못한 환경에 위치할 수 있다. 더욱이 센서 네트워크는 메모리, 전력, 계산능력에 있어서 상당히 제약적인 노드들로 구성되기 때문에 기존의 키 관리 기법을 적용하기 어려우며, 따라서 별도의 키 관리 기법이 요구되어진다. 본 논문에서는 센서 네트워크에서의 키 관리에 관련한 연구들을 소개하고 좀 더 안전한 키 분배 기법을 제안하고자 한다.

I. 서론

센서 네트워크는 다가올 유비쿼터스 컴퓨팅 환경에서 중요한 역할을 할 것으로 기대되고 있다. 개인용 휴대 단말기들을 주변의 센서 네트워크와 연동시킴으로써 그 활용도를 넓힐 수도 있을 것이다.

일반적으로 무선 센서 네트워크는 제한적인 저장공간, 전력, 계산능력, 대역폭을 갖는 수많은 센서 노드들로 구성된다. 이러한 무선 센서 네트워크는 노드들을 배치한 이후, 네트워크를 확장하기 위해 노드를 추가하거나 문제가 생긴 노드들을 제거해야한다는 면에서 유동적 네트워크이다. 또한 악의적인 공격자에 의해 센서 노드들 간의 통신내용이 수집되고 공격을 받거나 조작될 수 있는 위험에 노출되어 있다. 더욱이 센서 네트워크의 규모가 클 경우, 노드 하나하나를 수작업을 통해 배치할 수 없기에 비행기 등을 이용하여 노드를 뿌리게 된다. 따라서 전체 네트워크 구성을 노드를 배치하기 전에 미리 정확하게 알 수가 없다.[1)]

센서 네트워크의 보안문제는 센서 노드의 하드웨어적 제약사항과 배치방법상의 특징으로 인해 복잡해진다. 첫째로 센서 노드의 제한적인 계산능력, 저장공간, 전력으로 인해 전통적인 비 대칭키(공개키) 알고리즘을 사용하기가 어렵기 때문에 효율적인 대칭키 알고리즘의 사용이 요구되어진다. 그리고 센서 노드들은 물리적 공격에 노출될 위험이 높다. 상당수의 노드들을 배치하기 위해서는 각각의 노드들의 가격이 저렴해야할 필요가 있는데 이로 인해 센서 노드들을 물리적 손상에 대해 대비할 수 있게 만들기가 어렵게 된다. 만약 센서 노드가 공격자에 의해 캡쳐되면 노드 메모리상의 모든 정보가 노출되게 된다. 또한 센서 노드들이 무선 통신을 사용하기 때문에 이를 엿듣거나 네트워크에 거짓된 메시지를 쉽게 뿌릴 수 있다.[1][2]

본 논문은 다음과 같이 구성된다. II 장에서는 키 관리 방식에서의 기존 연구에 대해서 알아보고, III 장에서는 개선된 키 관리 기법을 제안하며, IV 장에서는 성능 및 보안 측면에 대한 분석을 하며, V 장에서 마무리한다.

II. 기존연구

일반적으로 pair-wise key 분배 과정은 3단계로 이루어진다: (i) 배치 전 키 설정단계, (ii) 배치 후 인접한 노드 간의 공유 키(shared-key) 발견단계, (iii) 인접한 두 노드가 공유 키를 갖지 않을 경우의 경로 키(path-key) 설정단계.[1]

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음

(공유 키란 직접 이웃한 두 노드 간에 쓰이는 키를 말하며, 경로 키 설정이란 한 개 이상의 다른 노드를 거쳐 가는 다중 홉 링크를 통해서 키를 설정하는 과정을 말한다.) 최근 이 분야에 많은 논문이 발표되었으며, 다양한 기법이 연구되었다.[3][4][5][6] 본 논문에서는 효율적이고 결정적인 프로토콜인 LEAP (Localized encryption and authentication protocol)의 취약점을 분석한 후 이를 바탕으로 안전한 키 분배 기법을 제시한다.[5]

LEAP은 노드들 간의 교환 메시지 종류가 서로 다르며, 각 종류별로 보안 요구사항이 다르다는 면에서 다중 키의 사용을 제안하였다. 즉, 독립적 노드 키(기지국과 센서 노드 간에 사용), pair-wise 공유 키(센서 노드 간에 사용), 클러스터 키(센서 노드와 모든 주변 이웃 노드들 간에 사용), 그룹 키(센서 네트워크 전체를 커버)의 4가지 종류 키를 제시하였다. pair-wise 키 설정 단계에서 노드는 initial key K_I를 받는다. 노드 S_u는 K_I 와 pseudo -random 함수 f를 사용하여 master key $K_u = f_{K_I}(ID_u)$를 생성한다. 공유 키 발견단계에서 노드 S_u는 $(ID_u, Nonce_u)$ 메시지를 브로드캐스트 하고 이 메시지를 받은 이웃 노드인 S_v는 이에 대한 응답을 보내준다.

$S_u \rightarrow *$: $ID_u, Nonce_u$

$S_v \rightarrow S_u$: $ID_v, MAC_{K_v}(Nonce_u | ID_v)$

이를 통해 S_u는 $K_v = f_{K_I}(ID_v)$를 계산할 수 있으며, 두 노드 S_u, S_v 모두 세션 키 $K_{u,v} = f_{K_v}(ID_u)$를 생성할 수 있게 된다. 키 생성과정이 끝나고 나면, 노드들은 최초의 initial key K_I와 자신의 것이 아닌, 계산된 다른 노드들의 master key를 모두 삭제한다. 결국 노드들은 자신의 master key와 주변 이웃 노드들과의 pair-wise key 정보만을 갖게 된다.

노드와 클러스터 헤드 간에는 다중 홉 pair-wise 키가 사용될 수 있다. 이를 위해 노드 S_u는 키 $K_{u,c}$를 만들고 m개의 중간 노드를 찾는다. $K_{u,c}$를 $K_{u,c} = sk_1 \oplus sk_2 \cdots \oplus sk_m$와 같이 m개로 분할하고 각각을 중간 노드 S_{v_i} $(1 \le i \le m)$를 통해서 S_c로 보낸다.

$S_u \rightarrow S_{v_i} : \{sk_i\}_{K_{u,v_i}}, f_{sk_i}(0)$

$S_{v_i} \rightarrow S_c : \{sk_i\}_{K_{v_i,c}}, f_{sk_i}(0)$

노드가 주위 모든 이웃 노드와 통신하기 위한 클러스터 키를 만들기 위해서 S_u는 클러스터 키 K_u^c를 만든다. 그런 다음 K_u^c를 이웃 노드 S_{v_i}에게 이미 만들어진 pair-wise 키로 암호화하여 $(K_u^c)_{K_{u,v_i}}$ 보내준다.

이 프로토콜의 보안성은 initial key K_I에 의존하게 되는데, initial key K_I를 지움으로 해서 노드에 대한 물리적 캡쳐를 성공하더라도 그 피해가 일부 지역으로 제한되게 된다. 이러한 보안성을 제공하기 위해 논문에서 가정하기를, 키를 생성하는데 소요되는 시간(T_{est})이 악의적인 공격자가 노드를 공격하여 노드로부터 정보를 빼내는 데 필요로 하는 최소의 시간($T_{\min}$)보다 작다고 하였다. 즉, $T_{est} < T_{\min}$ 이다.

III. 제안하는 기법

1. 기존 연구에서의 문제점

LEAP 프로토콜이 보안상 안전한 이유가 $T_{\min}$시간이 지나기 전에 키 생성 과정을 모두 마치고 initial key K_I를 지우기 때문이다. 그러나 대역폭이 좁은 센서 네트워크의 특성상 노드 간의 통신이 원활하지 않거나 packet이 드롭될 수 있기 때문에 키를 설정하는 데 소요되는 시간 T_{est}가 길어져서 악의적인 공격자가 노드를 캡쳐하는데 필요한 최소의 시간 $T_{\min}$보다 커지는 상황이 발생할 수 있다. 이런 경우, 공격자는 노드의 메모리로부터 initial key K_I를 획득할 수 있고, 이를 이용하여 데이터를 도청, 변조하거나 임의의 노드를 추가할 수 있는 등 전체 네트워크의 보안이 무너지는 문제가 발생하게 된다.

2. 제안하는 기법

이에 본 논문에서는 initial key K_I가 노출되는 경우가 발생하더라도 그로 인한 피해를 최

소화할 수 있는 좀 더 안전한 기법을 제시하고자 한다. 이를 위해 기존의 LEAP 프로토콜에 시간 축의 개념을 도입한 기법을 제시한다.

- 센서 노드는 뿌려지기 전에 노드가 뿌려지는 period의 initial key와 이후 시간대의 master key m개를 사전에 저장한다.
- 센서 노드는 노드가 뿌려진 해당 period 동안은 initial key를 이용하여 계산된 master key를 가지고 이미 뿌려진 다른 노드들과의 키 생성과정을 수행한다.
- 이후의 period 동안 뿌려지는 노드들과는 미리 계산되어 저장된 m개의 master key 중 해당되는 master key를 이용하여 키를 생성한다.
- 직접적으로 공유 키를 설정할 수 없는 경우, proxy 노드를 이용하여 경유 키를 생성하는 과정을 거친다.

예를 들어 $m=5$인 경우, 각 period 마다 저장되는 키 정보는 그림 1과 같다. T_3시간에 뿌려지는 노드의 경우에는 initial key K_{I_3}과 master key $K_{u_4} \sim K_{u_8}$을 저장하고 뿌려지게 된다. $T_1 \sim T_3$시간에 뿌려진 노드들과는 K_{I_3}를 이용하여 키 생성과정을 수행하며, $T_4 \sim T_8$시간에 뿌려지게 될 노드들과는 해당 period 때의 master key를 사용하여 키를 생성하게 된다.

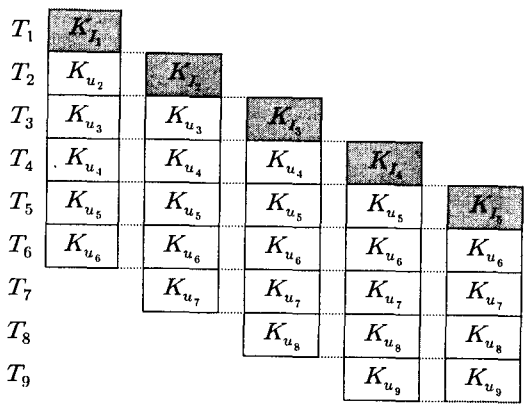

그림 1. 저장되는 키 정보($m=5$인 경우)

IV. 성능평가 및 보안성 분석

1. 성능평가

t_1시간에 뿌려진 노드 u와 t_2시간에 뿌려진 노드 v가 서로 pair-wise key를 생성하여 연결될 수 있는 키 연결성에 대해서 알아본다. 각 노드는 initial key외에 추가로 m개의 master key를 저장하고 있기 때문에 노드가 뿌려진 period 시간대 전후로 각각 m개의 period 동안 뿌려진 다른 노드들과 키 생성과정을 수행할 수 있다. 각 키가 유효하게 쓰이는 한 period의 시간을 T라 할 때 키 연결성은 다음과 같다.

$$C = \Pr[\,|t_2 - t_1| \leq m \cdot T\,] \qquad (1)$$

즉, 두 노드가 뿌려진 시간차이가 $m \cdot T$시간보다 작을 때 직접적으로 키를 연결할 수 있다. 위의 수식에서 보는 바와 같이, 노드마다 추가로 저장하는 master key의 개수 m을 증가시키거나 각 키가 유효한 시간 T를 늘리게 되면 두 노드가 직접적으로 공유 키를 설정할 수 있는 확률이 커지게 된다. 직접적으로 공유 키를 설정할 수 없더라도 이미 공유 키를 갖고 있는 다른 노드를 proxy 노드로 하는 경유 키를 설정할 수 있으며, 이러한 연결까지 고려한다면 전체 네트워크의 키 연결성은 더 높아진다.

높은 키 연결성을 위해 m값을 증가시키면 그만큼의 저장공간의 부하가 늘어나게 되며 T값을 늘리게 되면 하나의 키가 유효한 시간이 길어지게 되어 하나의 K_I 노출로 인한 피해가 늘어나는 측면이 있다.

만약 동일한 키 연결성을 유지하면서 저장공간의 부하를 줄이고자 m을 감소시키려면 period의 길이 T를 늘려야 하며, 반대로 한 개의 키가 유효한 시간을 짧게 하고자 T를 줄이려면 m을 증가시켜 더 많은 키를 저장해야 한다. 수식 (1)을 보더라도 m과 T는 동일한 키 연결성 C에 대해서 서로 반비례 관계에 있다는 것을 알 수 있다.

2. 보안성 분석

공격자가 센서 노드를 공격하여 노드 내의 정보를 획득하거나 노드를 악의적으로 이용하려 하는 것이 감지되면, 해당 노드와 관련된 키

정보를 폐기하여 더 이상 공격당한 노드를 악용하지 못하도록 해야 한다. 그러나 노드가 공격받았을 때, 공격을 받았다는 사실을 바로 감지해내는 일이 쉽지는 않다. 따라서 노드가 악의적인 공격자에게 캡쳐되서 키 정보 등의 메모리 내용이 노출되었을 때, 그로 인해 전체 센서 네트워크 중에서 보안이 무너지게 되는 부분이 얼마나 되느냐가 중요한 문제로 거론된다. 이는 반대로 노드의 캡쳐가 발생하였을 때 전체 센서 네트워크 중에서 얼마의 부분이 생존하는가 하는 문제로 볼 수도 있다.

기존의 LEAP 프로토콜에서는 T_{est} 시간 후에 initial key K_I를 지우기 때문에, 이 시간 후에 노드가 캡쳐되더라도 인근 노드와 설정된 pair-wise key, cluster key 정도를 이용할 수 있을 뿐이다. 즉, 이 노드를 다른 지역에 가져가서 활용할 수 없기 때문에 보안이 무너지는 범위를 최소화할 수 있다. 또한 wormhole attack이나 sinkhole attack 또한 비슷한 맥락에서 막을 수 있다. 그러나 이미 위에서 언급한 대로 T_{est} 시간 이전에 노드 캡쳐가 이루어지게 되면 initial key K_I가 노출되게 되고, 이는 곧 전체 센서 네트워크의 보안이 무너지는 것을 의미한다.

이에 반해 본 논문에서 제시하는 기법은 이러한 피해를 시간적으로 지역화 시킬 수 있다. 즉, 임의의 period T_a에 공격자가 노드를 캡쳐하여 initial key K_{I_a}를 알아내게 되더라도, 그로 인한 피해는 T_a 시간대로 국한되게 된다. 또한 공격자가 K_{I_a}를 안다고 하더라도 이전 시간대의 initial key를 알 수 없기 때문에, 그 동안의 트래픽을 저장해두었다 해도 그 내용을 해독할 수 없다는 면에서 backward confidentiality를 제공한다고 볼 수 있다. T_a 이후의 시간대에도 initial key가 아닌 master key만을 저장하기 때문에 해당 노드를 이용하여 다른 공격을 하기 어렵다는 면에서 forward confidentiality를 제공한다고 볼 수 있다.

V. 결론

LEAP 프로토콜은 센서 노드에서 비교적 구현하기 쉬우면서도 높은 보안성을 제공해 준다. 그러나 initial key K_I가 노출되는 경우 전체 센서 네트워크의 보안이 무너지게 된다는 문제점을 갖고 있다. 이를 해결하기 위해 시간 축의 개념을 도입하여 보안이 무너지는 범위를 시간적으로 최소화하였으며 향상된 보안성을 제공할 수 있도록 하였다.

[참고문헌]

[1] L. Eschenauer and V.D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," ACM conference on Computer and Communications Security, Nov. 2002.

[2] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," IEEE Symposium on Research in Security and Privacy, 2003.

[3] W. Du, J. Deng, Y. Han, and P. Varshney, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," Proceedings of the 10th ACM conference on Computer and Communications Security, Oct. 2003.

[4] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," ACM conference on Computer and communications security, 2003.

[5] S. Zhu, S. Setia and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," ACM Conference on Computer and Communications Security, Oct. 2003.

[6] D. Huang, M. Mehta, D. Medhi and L. Harn, "Location-aware Key Management Scheme for Wireless Sensor Networks," ACM SASN, Oct. 2004.