계층적인 센서 네트워크에서 확장성을 제공하는 분산 키 관리 방법

김미희°, 채기준
이화여자대학교, 컴퓨터학과
mihui°@ewhain.net, kjchae@ewha.ac.kr

Distributed Key Management Supporting Scalability on Hierarchical Sensor Networks

Mihui Kim°, Kijoon Chae
Ewha Womans University, Department of Computer Science and Engineering

요 약

본 논문에서는 계층적인 센서 네트워크에서 하위 센서 노드의 인증이나 센싱된 정보의 암호화를 위해 사용할 수 있는 키를 관리하기 위하여 키 선분배를 기본으로 키 재분배 방법을 제공하는 키 관리 메커니즘을 제안한다. 본 키 관리의 특징은 첫째, 중앙 관리의 약점을 극복하기 위해 키 관리를 다른 aggregator 노드들에 분산시켰다. 둘째, SINK 노드는 키의 재분배를 위한 키 스페이스를 제외하고, 이미 분배된 키에 대해서는 어느 노드에게 어떤 키를 분배했는지 또는 그 키 자체를 저장하지 않고, 키 계산을 위한 일부 정보만 저장하고 있다가 노드가 메시지에 첨부하여 주는 키 정보를 이용해 사용된 키를 간단히 계산하며, 키 풀의 확장이 용이하여 확장성을 제공한다. 마지막으로 계산 및 메모리 측면에서의 오버헤드 분석을 통해 제안된 키 관리의 확장성 제공을 입증한다.

1. 서 론

최근 유비쿼터스(ubiquitous) 컴퓨팅 구현을 위한 기반 네트워크로서 초경량, 저전력의 많은 센서들로 구성된 무선 센서 네트워크에 대한 연구가 활발히 진행되고 있다. 특히 센서 네트워크에서는 공격자가 트래픽을 쉽게 엿들을 수 있고 주변 노드에게 잘못된 정보를 제공함으로써 센서 네트워크의 노드로 흡내낼 수 있기 때문에, 보안성을 제공하는 것이 중요하다. 이러한 서비스를 위하여 고려해야 할 사항은 노드 캡쳐에 의한 위협의 가능성, 노드의 제한적인 자원, 동적 토폴로지 변화 등이다.

안전한 센서 네트워크의 통신을 위한 기본적인 보안 서비스로서 키 관리 방법에 대한 다양한 연구가 진행되어 왔다. 첫째, 관리적인 측면에서 SINK 노드나 베이스스테이션(BS)을 안전하다고 가정하고 중앙 키 관리를 하는 기법[1]이 제안되었는데, 실제로는 중앙 노드에 대한 공격 및 장애가 가능하며 이것이 미치는 영향이 그 중앙 노드가 관리하는 네트워크 전역에 미칠 수 있다는 점과 키 관리를 위한 트래픽 및 프로세싱 로드가 중앙으로 밀집된다는 단점을 갖고 있다. 둘째, 랜덤 키 선분배 방식을 통해 임의 노드 간에 간단한 계산으로써 pairwise 키를 계산할 수 있고, 노드 노출 시에도 어느 정도의 resilience를 제공하는 방법[2]이 제안되었는데, 이 또한 선분배 방식을 사용하기 때문에 노드 배치 후 키의 갱신에 대한 고려가 제공되지 않았고, 관리하는 노드가 확장되는 경우 관리하는 키 정보를 확장하기 용이하지 않다는 단점이 있다.

센서 네트워크는 응용에 따라 다양한 구조가 가능한데, 본 논문에서는 중앙 노드 SINK와 Aggregator 노드(AN)들과 다수의 센서 노드(SN)들이 계층적으로 구성된 네트워크를 가정한다. 이러한 구조의 트래픽 특성에 맞추고 기존 키 관리 기법들의 문제점을 보완하기 위하여, 본 논문에서는 센서들이 센싱한 정보의 암호화나 인증을 위하여 사용할 수 있는 키를 관리하되 SINK 뿐 아니라 AN에 분산 관리하는 키 관리 기법을 제안한다. 본 키 관리의 특징은 중회귀 모델을 사용하여 키를 생성 관리하고, 키 관리 노드에서는 키의 재분배를 위한 키 스페이스를 제외하고, 이미 분배된 키와 노드와의 관계를 저장하지 않고, 키 계산을 위한 정보만 저장하고 있다가, 노드가 메시지에 첨부한 키 정보를 사용하여 키를 계산하여, 관리 노드의 위협에 위한 피해를 최소화 하였다. 이를 통해 키를 계산할 수 있는 노드를 소수의 노드(SINK, AN)로 한정하였지만 이를 계층적으로 분산화하였고, 센서노드 캡쳐에 대해 λ-security 특성 제공한다. 또한, 용이한 키 풀 확장을 통해 관리 노드 수 확장에 따른 키의 확정성을 제공하며, 주기적으로 키 재분배를 수행함으로써 키의 freshness를 제공한다.

본 논문의 구성은 1장의 서론에 이어, 2장에서는 본 논문의 키 관리에서 기본으로 사용하고 있는 회귀모델을 설명하고, 3장에서는 본 논문에서 제안하고 있는 분산 키 관리 메커니즘을 기술한다. 4장에서는 제안된 메커니즘의 성능 분석하여, 마지막으로 결론으로써 본 논문을 마치고자 한다.

2. 회귀모델과 키 계산

회귀분석이란 이미 알려진 독립변수 (independent variable)들로부터 하나의 종속변수 (dependent variable)의 값을 예측하는데 사용되는 통계 방법 중의 하나이다. 예를 들어 어떤 플라스틱 제품의 견고도가 이 제품을 만드는 기계의 온도와 만드는 시간에 어떤 연관성이 있다면, 이들 변수간의 함수관계를 규명하기 위해 사용되는 방법이다. 이 예에서 견고도는 종속변수라 부르고, 온도와 시간과 같이 종속변수에 영향을 주는 변수를 독립변수라고 한다[3].

대표적인 회귀분석 방법인 최소 제곱법(least square)은 종속변수의 측정치 y와 b 행렬에 의해 계산된 값인 y_k와의 차이에 대한 제곱값이 최소($\min \sum (y-y_k)^2$)가 되도록 하는 최소 제곱법에 의해 추정된다. 최소 제곱법에 의해 추정되는 b 행렬은 다음과 같다.

$$b=(X'X)^{-1}X'y, \quad X' \text{는 } X \text{의 전치행렬}$$

그림 1은 독립변수의 계수가 $(\lambda+1)$, 표본 수가 n일 때의 중회귀(Multiple Regression) 모형으로 본 논문에서는 다음과 같은 중회귀 모형을 사용하는데, 이는 종속변수를 예측하기 위함이 아니라 노드가 송신해 주는 $(1\times(\lambda+1))$ 키 정보 X_i를 이용하여 사용된 키 y_{ki}를 계산해 내는 새로운 응용으로써 사용한다.

$$y_k = X(X'X)^{-1}X'y = X\times b$$

$$y_k=\begin{Bmatrix} y_{k1} \\ y_{k2} \\ y_{k3} \\ \vdots \\ y_{kn} \end{Bmatrix} = \begin{Bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{\lambda 1} \\ 1 & x_{12} & x_{22} & \cdots & x_{\lambda 2} \\ 1 & x_{13} & x_{23} & \cdots & x_{\lambda 3} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{\lambda n} \end{Bmatrix} \times \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_\lambda \end{Bmatrix} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_n \end{Bmatrix} \times \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_\lambda \end{Bmatrix} = X\times b$$

(그림 1) 일반 중회귀 모형

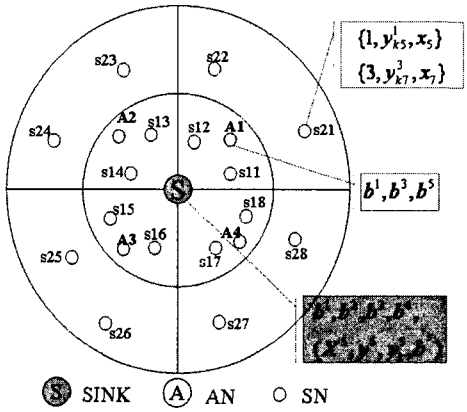
3. 제안하는 분산 키 관리 방법

3.1 키스페이스 생성, 선분배, 키 계산

본 논문에서 제공되는 키는 중회귀 모델을 사용하여 생성되고 관리된다. 즉, 그림 1에서 하나의 키 스페이스 y_k는 $(n\times(\lambda+1))$의 키 정보 행렬(Key Information Matrix) X와 $(n\times 1)$의 키 생성 행렬(Key Generation Matrix) y로부터 계산된다. 여기에서 행렬 X와 y는 유한체인 $GF(q)$ 위에서 정의되며, q는 하나의 키 y_{ki}의 길이가 64 비트일 때, 2^{64}보다 큰 소수 중 가장 작은 소수로써 선택하면 된다[2]. 이렇게 생성한 키와 키 정보 값을 노드에게 같이 선분배 해 주고, 키 관리 노드는 이에 대한 관계 정보는 전혀 저장하지 않고 키를 계산할 수 있는 b 행렬만을 저장하고 있다가, 노드가 키 사용 시 메시지와 함께 제공해 주는 키 정보를 가지고 간단한 행렬 곱셈 연산을 통해 (λ 번의 모듈러 곱셈) 키를 계산하여 사용하게 된다.

키 관리라는 새로운 응용을 위한 중회귀 모델의 적용 가능성은 첫째, 공격자(adversary)에게 키 정보 행렬 X에서 $(\lambda+1)$개의 행, 즉 $(\lambda+1)$개의 키 정보가 노출되지 않는 한 다른 키를 계산해 내기 위한 b 행렬을 알아낼 수 없다는 키 관리의 안정성을 제공할 수 있고, 둘째 키 관리 노드는 분배된 키와 노드와의 관계 정보를 저장하지 않고 노드가 제공하는 키 정보를 가지고 간단히 키를 계산함으로써 키 관리 노드의 위협에 의한 전체 키의 노출이라는 취약점을 보완할 수 있다. 이러한 키 스페이스는 노드 캡쳐에 의한 키 스페이스의 노출 정도를 낮추어 안전한 키 관리를 행할 수 있도록 다수의 행렬 y로부터 계산된 다수의 키 스페이스를 키 풀로 생성하여 관리할 수 있다.

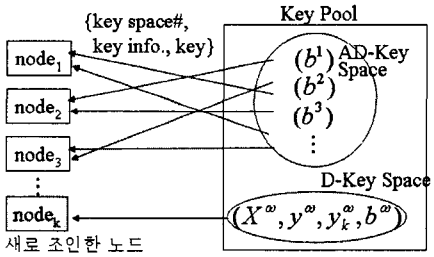
메시지에 첨부되는 키 정보의 크기를 줄이기 위해 [2]에서와 같이 시드(seed) 값 s의 곱에 의해 키 정보 행렬을 구성하면, s^i 키 정보 값만 전달해 주면 키 계산시 전체 키 정보인 $(1 \;\; s^i \;\; (s^i)^2 \;\; (s^i)^3 \;\; \Lambda \;\; (s^i)^\lambda)$을 계산하여 사용할 수 있다. 그림 2는 본 논문에서 가정한 계층적인 센서 네트워크와 각 노드가 갖는 정보를 예시하였다.



(그림 2) 계층적인 센서 네트워크의 예와 노드가 갖는 정보의 예

3.2 키 풀 관리와 키 스페이스 확장

키 관리 중앙 노드인 SINK는 키 풀을 관리할 때, 관리 노드의 노출에 대한 위협을 줄이기 위해 이미 분배된 키에 대한 스페이스(AD-Key Space, Already Distributed -Key Space)와 새로 네트워크에 추가된 노드나 키 갱신을 목적으로 사용되는 키 스페이스(D-Key Space, Distributing-Key Space)로 나누어 관리할 수 있다. AD-Key Space에 대해서는 해당 키 스페이스에 포함된 키가 사용되었을 때, 그 키의 계산을 위해 b 행렬만을 저장하고 있고, 해당 키 스페이스의 X, y 행렬 및 어느 노드에게 어떤 키셋을 분배하였는지에 관한 관계 정보는 저장하지 않는다. 이에 반해 D-Key Space에 대해서는 분배하고 있는 키 스페이스 y_k 및 키 정보 행렬 X, 키 생성 행렬 y, 키의 용이한 계산을 위한 b 행렬을 저장하고 있고, 그러나 어느 노드에게 어떤 키셋이 분배되었는지는 저장하지 않는다. 그림 3은 키 풀 관리에 대한 예시이다.



(그림 3) 키 풀 관리

본 논문의 키 관리 방법에서 키 freshness를 높이기 위하여 SINK 노드 키 스페이스 확장 방법을 제공하는데 설명한 키 스페이스 생성과 마찬가지로 하나의 키 스페이스 (X^6, y^6, y_k^6, b^6)를 생성한 후, 현재 D-Key Space (X^5, y^5, y_k^5, b^5)를 대신하여 새로운 D-Key Space로서 이후의 키 분배 및 재분배에 사용되고, 기존에 D-Key Space는 AD-Key Space로 포함되어 키 계산을 위한 b^5 행렬만 남기고 나머지 X^5, y^5, y_k^5 행렬은 삭제하게 된다. 그리고 새로 생성된 키 스페이스의 b^6 행렬을 AN들

에게 전달하되, 각 AN들로부터 최근에 수신된 메시지에 사용된 키셋을 가지고 암호화하여 안전하게 전달하고, 이를 수신한 AN은 자신의 하단에 또 다른 AN이 존재하면 같은 방법으로 안전하게 b^6 행렬을 전달한다.

또한 키 관리 노드의 저장 공간의 유한성으로 인해 일부 키 스페이스를 제거하고자 하는 경우, AD-Key Space 중에 가장 오래된 순서대로 혹은 랜덤하게 선택하여 삭제할 키 스페이스를 선정한 후, 안전하게 삭제할 키스페이스 번호를 전달하면, 노드는 이 노드에 해당하는 키셋 정보를 삭제한다.

3.3 키 재분배

전체 키 풀을 관리하는 노드(그림 2에서는 SINK)에서는 주기적으로 키를 갱신시키기 위한 타이머를 갖고 있고, 해당 타이머가 타임아웃 되면 일정 시간(T_{redist}) 동안 키 재분배 기능을 수행하게 된다. 그림 2를 가지고 예를 들어 설명하면, SINK가 키 재분배 시간 동안 센서 노드 SN으로부터 직접 센싱된 정보를 수신하면, 수신된 정보에 포함된 키 스페이스 번호와 키 정보를 가지고 키 K_{pre} 를 계산하여 정보를 처리하고, 새로운 키셋 정보(복수개 키셋도 가능)를 D-Key Space인 b^5 에서 랜덤하게 선택하여 수신한 정보에 사용했던 기존 키 K_{pre} 를 가지고 암호화 및 인증코드를 작성하여 전송한다. 이를 수신한 SN에서는 복호화 및 인증 과정을 거친 후 정당한 키 재분배 메시지이면, 자신이 갖고 있던 다수의 키셋 중에서 새로 받은 키셋의 수만큼 랜덤하게 제거하고 새로운 것으로 대치하여 관리하게 된다.

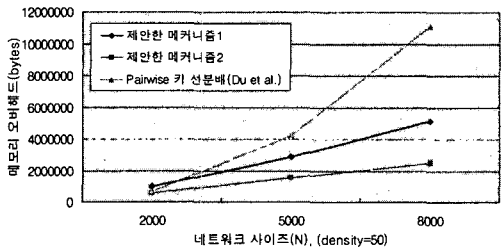
이와는 달리 SINK가 키 재분배 시간 동안 AN으로부터 통합된 센싱 정보를 수신하면, 이에 대한 복호화, 인증 및 데이터 처리 후, 새로운 키를 분배하기 위해 다수의 키셋 정보를 D-Key Space인 b^5 에서 랜덤하게 선택하고 남은 키 재분배 시간(T_{redist})과 함께 수신한 정보에 사용했던 기존 키를 가지고 암호화 및 인증코드를 작성하여 전송한다. 이를 수신한 AN에서는 남은 키 재분배 시간 동안, 하단 AN이나 SN들에게 새로이 수신한 다수의 키셋 정보를 이용하여 SINK와 유사한 키 재분배 과정을 수행하고, 키 재분배 시간 T_{redist} 가 끝나면, AN은 재분배를 위한 키셋 정보를 삭제하게 된다. 이로써 키 재분배 시간 동안 센서 네트워크의 노드들의 키가 랜덤하게 갱신되게 된다.

3. 성능분석

본 논문에서 제안된 키 관리 메커니즘의 통신 오버헤드는 키 스페이스의 개수가 ω 이고 키 정보 행렬의 한 원소가 x_{qi} 라고 했을 때, $\log_2 \omega + |x_{qi}|$ 정도로 사용되는 키 길이 정도의 오버헤드만 발생하게 된다.

또한 계산 오버헤드는 수신된 키 정보의 시드값에서 키 정보를 생성하는데 필요한 $(\lambda - 1)$ 번의 모듈로 곱셈과 사용된 키의 계산을 위해 생성된 키 정보와 b 행렬과의 $(\lambda + 1)$ 번의 모듈로 곱셈이 필요하므로 총 2λ 모듈로 곱셈의 계산 오버헤드가 필요하여, 이는 pairwise 키 선분배 기법[2]과 동일한 양이며, RSA 공개키 기반의 암호화 알고리즘에 비해 훨씬 작은 계산양이다.

그림 4는 메모리 오버헤드 비교를 위해 Du et al.[2]이 제안한 방법과의 비교한 그림으로 한 AN이 50개의 SN을 처리하는 환경에서 네트워크 사이즈를 증가시켰을 때의 필요한 메모리 양을 나타내었다. 사용된 파라미터 값은 [2,4]논문에서 사용한 값을 참고하였고, 키 길이는 64 비트로 하였다 또한, 본 논문의 AN에서 갖는 키 스페이스 수 μ 를 전체 키 스페이스 수인 ω 와 동일하게 한 경우가 " 제안한 메커니즘1" 이고(메모리 측면에서 최악의 경우), ω 의 절반의 값을 μ 의 값으로 한 경우가 " 제안한 메커니즘2" 의 결과이다. 결과적으로 본 논문의 방법이 Du et al. 방법의 메모리 오버헤드보다 네트워크 노드 증가에 따라 영향이 적음을 나타내어 노드에 대한 확장성 제공을 입증할 수 있다.



(그림 4) 네트워크 사이즈 증가에 따른 메모리 오버헤드

4. 결론

본 논문에서는 안전한 계층적인 센서 네트워크를 위한 기본적인 메커니즘인 키 관리에 대해 키 관리 기능을 분산시켜 제안하였고, 중회귀 모델을 사용하여 키를 생성, 관리하며, 키 자체에 대한 정보를 저장하지 않고 간단한 계산으로써 사용된 키를 계산함으로써 중앙 키 관리에 대한 단점 및 위협에 따른 피해를 완화시켜 주었다. 마지막으로 통신, 계산, 메모리 측면의 오버헤드 분석을 통해 제안된 키 관리의 효율성을 입증하였다.

Acknowledgement

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

참고문헌

[1] Adrian Perrig, et al., " SPINS: Security Protocols for Sensor Networks," WINE 2002.

[2] Wenliang Du, et al., " A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," CCS 2003.

[3] 박선형, " 제 3 판 회귀분석" , 민영사, 1998 년 9 월.

[4] Haowen Chan, et al., " PIKE:Peer Intermediaries for Key Establishment," INFOCOM 2005.