WSN에서 변형된 Merkle 트리를 이용한 공개키 인증 기법

김은주[O,†], 김현성[††], 이원진[†], 전일수[†]
[†]금오공과대학교 전자통신공학과, [††]경일대학교 컴퓨터공학부
{candycore[O], wjlee, isjeon}@kumoh.ac.kr, kim@kiu.ac.kr

Public Key Authentication Scheme using Transformed Merkle-Tree in WSNs

Eun-Ju Kim[O,†], Hyun-Sung Kim[††], Won-Jin Lee[†], Il-Soo Jeon[†]
School of Electronic Communication Engineering, Kumoh National Institute of Technology,
School of Computer Engineering, Kyungil University

요 약

최근 Du 등은 공개키 암호 기반의 센서네트워크에서 주요 난제인 공개키 인증 문제를 해결하기 위해 해쉬 함수를 사용하여 계산비용을 줄일 수 있는 트리 기반의 인증기법[1]을 제안하였다. 그러나 이 기법은 공개키 인증을 위해 노드의 수 N에 대해 $O(\log N)$의 수행시간이 필요하고 전송되는 메시지의 양도 $O(\log N)$이 되어 네트워크의 크기에 따라 인증비용이 커지는 문제점이 있다. 이러한 문제를 해결하기 위하여 본 논문에서는 공개키 인증을 위한 안전하고 효율적인 경량의 인증 기법을 제안한다. 제안한 기법은 해쉬 연산과 XOR 연산을 이용하여 네트워크의 크기에 관계없이 상수복잡도의 수행시간과 전송되는 메시지의 양이 상수복잡도인 향상된 공개키 인증을 수행한다.

1. 서 론

센서네트워크는 군사목적의 감시, 생태계 모니터링, 환자 관리 시스템 등 다양한 목적으로 활용 범위가 더욱 확대되고 있는 기술이다. 이는 자원이 제한된 센서노드로 구성되며 물리적으로 안전하지 않은 환경에 배치되기 때문에 보안이 매우 취약하거나 다양한 형태의 공격들이 잠재될 수 있다[2]. 이러한 위협에 대해 안전성을 제공하기 위해서는 센서노드 간 비밀성과 무결성을 보장하여야하므로 인증에 필요한 키 수립이 가장 먼저 고려되어야 한다.

센서노드 간 비밀키를 수립하는 가장 단순한 방법은 모든 노드가 하나의 그룹키를 공유하는 방식이다. 하지만 이 방식은 센서노드가 하나만 공격자에게 포획된다고 하더라도 그룹키가 유출되어 전체 네트워크가 보안적 위험에 처하게 된다. 다른 극단적인 방법으로는 모든 센서노드의 쌍마다 유일한 키들을 할당하는 방식이 있다. 그러나 이 경우에도 각각의 센서노드는 전체 노드 수, N중 N-1개의 키를 저장해야 하므로 메모리 제한을 가지는 센서네트워크에서는 비현실적인 방법이다.

대칭키 기반의 암호시스템을 이용한 센서네트워크에서 보안의 최종적인 목표는 높은 레벨의 연결성과 탄력성을 제공하는 것이다. 하지만 센서네트워크는 여러 가지 제약사항이 있어서 동시에 이 두 가지 요구조건을 충족시키기는 어렵다. 그러나 최근에 여러 가지 연구들에서 효율적이면서 안전한 키 수립 방법들이 제안되고 있다[3-9]. Perrig 등은 베이스 스테이션을 두고 각 센서노드 간에 하나의 키를 생성하는 기법[3]을 개발하였고, 몇몇 연구들에서는 Eschenauer와 Gligor의 랜덤키 사전 분배 방법[4]을 기반으로 하여 다양한 기법[5,6]을 제안하였다.

이외에도 공개키 암호의 비대칭 속성이 가지는 장점을 기반으로 하여 센서네트워크에서의 공개키 암호시스템 실용성에 관한 연구[7,8]가 활발히 진행되고 있다. 한 예로, 타원 곡선 암호(ECC)의 경우, 서명 검증에 걸리는 시간이 크로스보우사의 모트 플랫폼인 ATmega128 8MHz 프로세서 상에서 160bit의 키를 가질 때 1.62초가 걸림을 보였다[9].

Du 등의 논문[1]에서는 센서네트워크에 적합한 공개키 암호시스템을 사용하기 위해서 공개키 암호시스템에서 많은 연산의 오버헤드가 필요한 공개키 인증에서 연산의 효율성을 제시하기 위한 연구를 제시하였다. Du 등의 공개키 인증 기법에서는 Merkle 트리[10]의 commitment 값을 해쉬 연산만으로 검증하는 기법을 제안하였다. 하지만 이 기법에서는 단방향 인증을 제공하고, 통신 시 주고받는 인자들이 그대로 노출되어 재전송 공격의 가능성이 있으며, 노드 수 증가에 따라 연산과 통신에 필요한 인자의 수도 늘어나므로 이를 해결하기 위하여 배치정보를 이용한 해결책을 제시하였다.

본 논문에서는 Du 등이 제시한 기법의 문제점을 해결하기 위하여 공개키 인증을 보다 안전하고 효율적으로 하기위해 변형된 Merkle 트리를 이용한 양방향 인증 기법을 제안한다. 제안된 인증 기법에서는 해쉬 연산과 XOR 연산을 기본으로 한다. 제안한 기법은 기존의 기법의 재전송 공격에 대응하기 위하여 challange/response 방법을 사용한다. 특히, 기존 기법에서의 해쉬 연산으로 인한 오버헤드에 따른 문제를 해결하기 위하여 연산의 오버헤드가 작은 XOR 연산을 효율적으로 적용한다. 변형된 Merkle 트리를 이용한 공개키 인증을 수행함으로써 Du 등의 기법에서 필요로 한 배치정보에 대한 추가적인 정보의 필요성을 없애고, 센서네트워크를 구성하는 노드 수에 상관없이 통신 시 평균 6개의 인자를 사용함으로써 통신 및 연산의 오버헤드를 효율적으로 줄일 수 있었다.

본 본문의 구성은 다음과 같다. 2장에서는 Du 등이 제안한 기본적인 공개키 인증 기법과 배치정보를 이용한 추가적인 인증 기법에 대해서 살펴보고, 3장에서는 Du 등의 프로토콜의 문제점을 해결하기 위한 변형된 Merkle 트리를 기반으로 하는 새로운 공개키 인증 기법을 제안한다. 4장에서는 제안된 기법에 대하여 분석을 제시하고 마지막으로 5장에서는 결론을 맺는다.

2. 관련 연구

본 장에서는 본 논문에서 사용하는 용어의 표기법을 제시하고 관련 연구로서 Du 등의 Merkle 트리에 기반 한 공개키 인증 기법에 대해서 간단히 살펴본다.

2.1 표기법 정의

본 논문에서 제안된 기법 및 관련 연구에서 사용되는 용어의 표기법은 표 1과 같다.

표 1. 용어 표기법

기호	설명
N	전체 노드의 수
L_i	i번째 잎 노드
id_i, id_a, id_b	i번째 잎 노드와 A 노드 그리고 B 노드의 식별자
pk_i, pk_a, pk_b	i번째 잎 노드와 A 노드 그리고 B 노드의 공개키
V	잎 노드와 루트 노드를 제외한 내부 노드
V_{left}	내부 노드의 왼쪽 자식 노드
V_{right}	내부 노드의 오른쪽 자식 노드
R	루트 노드
λ	잎 노드로부터 루트 노드까지의 경로
H	λ의 최고 높이
v_j	λ에 포함되는 노드 중, j번째 노드
$h()$	일방향 해쉬 함수
$\|$	연결 연산자
$\oplus$	XOR 연산자
S	N을 x그룹으로 나눌 때, 한 그룹 내의 노드 수
a,b,c,d	노드위치정보에 기반하여 동일, 수직/수평, 대각선, 나머지의 그룹으로 나눈 그룹별 서브트리 높이
$\lceil x \rceil$	x보다 작지 않은 최소 정수
w_0, w_1, w_2, w_3	두 이웃 노드의 관계가 동일, 수직/수평, 대각선, 비이웃으로 정해질 수 있는 확률

2.2 Du 등의 공개키 인증 기법

Du 등은 논문 [1]에서 Merkle 트리에 기반한 공개키 인증 기법을 제안하고 있다. 본 절에서는 Du 등의 기법에서 사용하는 Merkle 트리의 구축 방법을 살펴보고, 이러한 Merkle 트리에 기반 한 공개키 인증 기법에 대해서 기술한다. 이러한 공개키 인증 기법은 노드의 수가 증가할수록 연산의 오버헤드가 커지는 문제점이 있다. 이러한 문제를 해결하기 위해서 Du 등이 제안한 노드의 배치정보를 이용한 추가적인 방법을 마지막으로 살펴본다.

2.2.1 Merkle 트리의 구축

일반 네트워크에서 사용되는 공개키 인증 기술은 센서 네트워크의 제약사항들로 인해서 변형된 방식이 제시되어야 한다. 그러므로 대칭형 암호 시스템이나 속도가 빠른 해쉬 함수를 사용하여 인증 기술을 구성하는 것이 바람직하다. Du 등은 Merkle의 인증 트리 기법[10]을 이용하여 해쉬 함수만으로 효율적인 인증이 가능한 기법을 제안하였다. Assignment Φ를 통해서 내부 노드 및 루트 노드로 구성된 완전이진트리 구조의 Merkle 트리를 구성한다.

그림 1에서 보여주는 바와 같이 Merkle 트리를 구축하기 위해 N개의 노드로 구성되는 센서네트워크에서는 N개의 잎 노드(leaf node)로 구성된다. 잎 노드와 그 외의 노드들이 가지게 되는 Φ 값은 다음 식에 의해 정의된다.

$$\Phi(L_i) = h(id_i, pk_i), \text{ for } i=1,\ldots,N$$
$$\Phi(V) = h(\Phi(V_{left}) \| \Phi(V_{right}))$$

$\Phi(L_i)$는 i번째 노드의 식별자와 공개키를 해쉬 함수로 매핑한 값이고, 추후에 계산될 $\Phi(R')$와 $\Phi(R)$ 값이 동일할 경우, 사용자의 공개키를 인증하는 방식이다. 여기서, V는 전체 Merkle 트리에서 내부 노드를 가리킨다. 부모 노드들은 자식 노드들의 Φ 값의 접합을 다시 해쉬하여 자신의 Φ 값을 구한다. 루트 노드도 다른 부모 노드와 동일하게 자신의 Φ 값을 구한다. Merkle 트리의 생성이 완료되면 모든 노드는 Merkle 트리의 루트인 $\Phi(R)$과 각 노드(잎 노드)를 기준으로 루트까지의 경로 λ에 포함되는 노드들의 형제 Φ 값들을 부가인자로 저장한다. 이에 따른 각 노드의 메모리 사용량은 (해쉬의 길이×(트리 높이+1)) 가 된다.

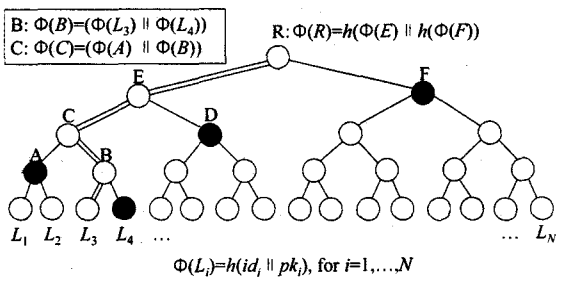

그림 1. 기존 기법의 공개키 인증을 위한 Merkle 트리

2.2.2 공개키의 인증

A노드의 공개키가 pk_3이고, A의 상응하는 잎 노드를 L_3라고 할 때 L_3부터 루트까지의 경로를 λ로, 그 경로의 길이를 H로 표현한다. 인증 과정은 다음과 같이 요약할 수 있다.

Step 1. A는 자신의 공개키 pk_3와 경로λ에 포함되는 노드들의 형제 Φ 값들을 B에게 전송한다.

$$A \rightarrow B : \{pk_3, \Phi(L_4), \Phi(A), \Phi(D), \Phi(F)\}$$

수신자B는 검증을 위해 A의 공개키 pk_3와 A의 식별자 id_3(미리 알고 있다고 가정)를 통해서 λ_1인 해쉬 값 Φ

(L_3')를 먼저 계산한다.

$$B : \Phi(L_3') = h(id_3, pk_3)$$

Merkle 트리의 루트, R'를 재구성하기 위해 앞서 나온 결과와 나머지 인자 값들을 다시 해쉬를 취함으로써 $\lambda_2,\ldots,\lambda_H$를 구성한다.

$$\begin{aligned} B : (1)\ \Phi(B') &= h(\Phi(L_3') \| \Phi(L_4')) \\ (2)\ \Phi(C') &= h(\Phi(B') \| \Phi(A')) \\ (3)\ \Phi(E') &= h(\Phi(C') \| \Phi(D')) \\ (4)\ \Phi(R') &= h(\Phi(E') \| \Phi(F')) \end{aligned}$$

$\Phi(R')$가 B에게 저장된 $\Phi(R)$과 일치하면 B는 id_3와 pk_3의 바인딩을 신뢰할 수 있다. 그러나 Du 등의 프로토콜에서는 Step 1에서 $\Phi(R')$을 계산하는데 있어서 중요한 인자들이 노출되어 전송됨으로 해서 재전송 공격의 잠재적인 위험을 가진다. 즉, 공격자가 도청을 통해 메시지를 캡쳐하여 다른 노드에게 이용하더라도 수신자의 검증에서는 유효한 메시지로 확인되므로 공격자를 인증할 수 밖에 없는 문제가 발생하는 것이다.

그림 1에 나오는 검은색 원들은 A의 경로 상에 있는 노드들의 형제 값으로 공개키와 함께 인자로 보내게 된다. 여기서 노드의 수가 늘어날수록 인증 시 전송되어야 할 인자의 수도 늘어나 통신의 오버헤드가 생기고, 이러한 정보들은 각 노드의 메모리에 저장되어야 하므로 메모리의 효율성에도 문제점이 있다.

2.2.3 배치 정보를 이용한 개선된 기법

Merkle 트리를 단순히 구성함으로써 생긴 메모리 문제를 해결하기 위하여 Du 등의 기법에서는 배치정보를 이용한 해결을 제안하였다. 센서노드들은 배치정보를 통해 배치된다. 이러한 배치과정에서 이웃이 될 수 있는 노드들을 그룹화 하여 그룹단위로 확률(w_0,w_1,w_2,w_3)을 구하고 높이를 산출하여 Merkle 서브트리의 높이를 최적으로 선택한다.

$$m_{max} = \lceil \frac{S}{2^a} \rceil + \lceil \frac{4S}{2^b} \rceil + \lceil \frac{4S}{2^c} \rceil + \lceil \frac{N}{2^d} \rceil \qquad (식\ 1)$$

$$C = w_0 \cdot a + w_1 \cdot b + w_2 \cdot c + w_3 \cdot d \qquad (식\ 2)$$

식 1을 통해 노드의 최대 메모리 요구량을 구할 수 있고, 식 2를 통해서 통신 오버헤드를 구할 수 있다. 이 두 가지 식을 이용하여 최적의 트리 높이를 구한다.

이웃이 될 확률이 높은 수직/수평, 대각선의 그룹들은 다른 그룹에 비해 짧은 높이를 가지며 그 높이에서 제일 위의 조상 노드를 루트 노드로 가진다. 노드들이 선택적으로 최적화된 다른 높이를 가짐으로써 저장해야 하는 인자들의 수도 최적화된다고 볼 수 있다.

3. 새로운 공개키 인증 기법

본 장에서는 변형된 Merkle 트리를 이용하여 재전송 공격에 강한 개선된 공개키 인증 기법을 제안한다.

3.1 변형된 Merkle 트리의 구축

기존 기법에서는 해쉬 함수만으로 Merkle 트리를 구축하였다. 해쉬 함수를 반복함으로써 인자들의 길이는 짧게 고정될 수 있었지만 네트워크의 크기가 확장될 때 인자 수가 함께 증가되는 문제를 야기하였다.

그림 2에서는 본 논문에서 제안한 변형된 Merkle 트리를 보여준다. Du 등의 Merkle 트리를 효율적으로 개선하기 위하여 다음과 같이 본 기법에서는 잎 노드의 Φ 값과 루트 노드의 Φ 값을 구할 경우에만 해쉬 함수를 이용하고, 나머지 내부 노드들의 Φ 값은 XOR연산을 이용한다.

$$\begin{aligned} \Phi(L_i) &= h(id_i \| pk_i),\ \text{for } i=1,\ldots,N \\ \Phi(V) &= \Phi(V_{left}) \oplus \Phi(V_{right}) \\ \Delta &= (v_1\text{'s} \oplus v_2\text{'s} \oplus \ldots \oplus v_{H-1}\text{'s}) \\ \Phi(R) &= h(\Phi(L_i) \oplus \Delta) \end{aligned}$$

여기서, Δ은 λ에 포함되는 노드들의 형제 값들을 포함하는 것으로 λ에는 루트 노드의 정보는 제외된다. 모든 노드들은 자신의 공개키와 루트의 해쉬 값, Δ(자신 노드로부터 루트까지의 경로에 존재하는 내부노드들의 Φ 값들을 모두 XOR한 하나의 인자 값)만 저장한다.

3.2 변형된 Merkle 트리에 기반한 인증 기법

A노드의 공개키가 pk_a이고, A가 선택하는 nonce를 n_a, A의 Δ를 Δ_a라고 가정한다. A는 $R_a(\Phi(R)$의 해쉬를 취하기 전 값)를 $h(id_a\|pk_a)\oplus\Delta_a$로 계산할 수 있다. 이와 마찬가지로 B노드의 공개키는 pk_b, 선택한 nonce는 n_b, 이 등을 이용하여 A와 동일한 가정을 한다. 본 기법에서는 재전송 공격에 대응하기 위하여 challange/response 방식을 이용한 양방향 인증을 수행하며 세부적인 과정은 다음과 같다.

Step 1. A는 인증을 요청하기 위해 먼저 R_a와 선택한 n_a를 XOR하여 전송한다. 이를 수신한 B는 메시지에 자신의 R_b를 XOR하여 n_a'를 유도한다.

$$\begin{aligned} A \rightarrow B &: (R_a \oplus n_a) \\ B &: (R_a \oplus n_a) \oplus R_b \Rightarrow n_a' \end{aligned}$$

Step 2. B는 A로부터 인증을 받기 위해 자신의 공개키 pk_b와 유도된 n_a', 그리고 R_b와 Δ_b의 XOR 값, R_b와 n_b의 XOR 값을 인자로 A에게 전송한다. A는 B로부터 수신된 메시지를 통해 다음의 조건이 만족되면 B를 인증하고 n_b'를 유도한다.

$$\begin{aligned} B \rightarrow A &: \{pk_b,\ n_a',\ (R_b \oplus \Delta_b),\ (R_b \oplus n_b)\} \\ A &: \text{If } n_a' == n_a \text{ and} \\ &\quad h(h(id_b\|pk_b) \oplus (R_b \oplus \Delta_b) \oplus R_a) == \Phi(R) \\ &\quad (R_b \oplus n_b) \oplus R_a \Rightarrow n_b' \end{aligned}$$

Step 3. 마지막으로 A가 B로부터 인증받기 위하여 공개

키 pk_a와 n_b', R_a와 Δ_a의 XOR 값을 인자로 B에게 보낸다. B는 인자들의 계산을 통해 다음의 조건이 만족되면 A를 인증한다.

$$A \rightarrow B : \{pk_a,\ n_b',\ (R_a \oplus \Delta_a)\}$$
$$B : \text{If } n_b' == n_b \text{ and}$$
$$h(h(id_a \| pk_a) \oplus (R_a \oplus \Delta_a) \oplus R_b) == \Phi(R)$$

위의 단계들은 재전송 공격의 문제를 해결하기 위해 각 노드가 하나의 인증 세션 안에서만 유효한 nonce를 생성하고 challenge/response 방법을 통해 확인하는 과정을 포함한다. 그리고 이를 위해서 각 노드의 nonce(n_a, n_b)들은 각 R_a와 R_b로 XOR 연산을 하여 비밀성을 유지한다. A와 B의 메모리에 저장된 $\Phi(R)$이 동일한 값이므로 A, B가 각각 계산하게 되는 R_a과 R_b도 동일하게 일치하며 이를 통해 전송된 XOR 값에 자신의 R_a, R_b를 한번 더 XOR하면 원래의 nonce를 구할 수 있다. 이렇게 유도된 nonce가 서로 일치하고 $\Phi(R)$로 복호한 값 역시 일치하면 A와 B는 역시 서로를 인증할 수 있다. 그림 2는 변형된 Merkle 트리의 구성을 보여준다. 특히 루트의 $\Phi(R)$ 은 잎 노드들로부터 계산된 값과 XOR의 결합법칙에 의해 노드들의 형제 값, 즉 여러 개의 인자들이 결합된 하나의 인자 값으로 나타낼 수 있다. 여러 인자 값이 포함되었더라도 XOR연산으로 결합되었기 때문에 처음 잎 노드의 해쉬된 값과 같은 길이를 갖는다.

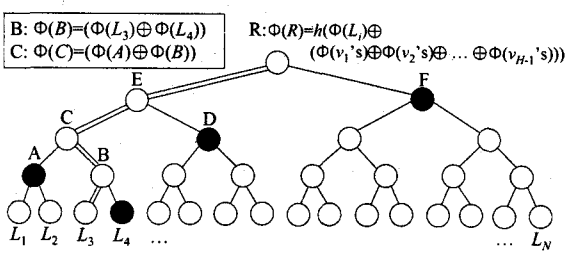

그림 2. 변형된 Merkle 트리의 구성

3.3 제안된 인증 기법의 특징

본 논문에서 제안된 두 개체 간에 이루어지는 인증 기법은 다음과 같이 3.2절에서 언급한 3번의 전송에 의해서 이루어진다.

Step 1. $A \rightarrow B : (R_a \oplus n_a)$
Step 2. $B \rightarrow A : \{pk_b,\ n_a',\ (R_b \oplus \Delta_b),\ (R_b \oplus n_b)\}$
Step 3. $A \rightarrow B : \{pk_a,\ n_b',\ (R_a \oplus \Delta_a)\}$

A노드는 임의의 노드 B에게 루트의 정보가 포함된 자신이 생성한 n_a를 보냄으로써 인증을 요청한다.

인증 요청을 받은 B노드는 유도된 n_a'로 응답함과 동시에 $(R_b \oplus \Delta_b)$와 자신이 생성한 n_b의 두 인자를 포함하여 A에게 다시 인증을 요청한다. A는 n_a와 n_a'의 일치를 확인한 후에 $\Phi(L_b)$를 계산하고, $(R_b \oplus \Delta_b)$와 R_a를 XOR 연산하여 Δ_b를 구하여 $\Phi(L_b)$와 Δ_b를 입력으로 한 해쉬 연산을 거쳐 $\Phi(R')$값을 계산한다. 그 값이 A가 가지고 있던 $\Phi(R)$값과 같을 경우에만 B의 공개키를 확신한다. 마찬가지로 B도 유도한 $\Phi(R')$과 $\Phi(R)$를 비교하여 상대방인 A의 공개키를 인증한다.

$\Phi(R) = h(h(id_i \| pk_i) \oplus (R_i \oplus \Delta_i) \oplus R_j)$의 증명은 쉽게 할 수 있다. XOR로 연결되어 있는 R_i와 R_j는 동일한 값이므로 소거가 가능하며, $\Phi(R) = h(h(id_i \| pk_i) \oplus \Delta_i)$가 최종적으로 계산된다.

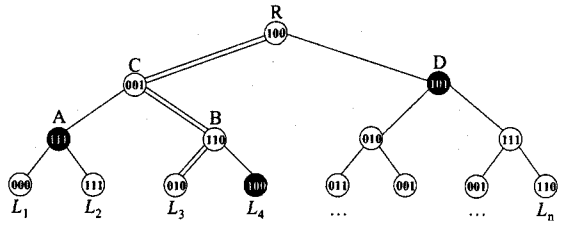

그림 3. 변형된 Merkle 트리의 XOR 비트 연산

그림 3에서 XOR의 비트 연산을 보면 검은색 원은 이미 잎 노드들의 XOR한 값을 가지고 있는데 이는 조상 노드에 잎 노드의 값들이 포함이 되어 있는 것이다.

XOR에서는 $A \oplus B \oplus C = A \oplus (B \oplus C)$의 결합법칙이 성립하기 때문에 다음의 과정에 의해 $\Phi(R)$이 같은 값으로 복원되는 것을 증명할 수 있다.

(1). 순차적인 XOR의 결과에 해쉬 연산을 취하는 형태는 다음과 같다.

$$\Phi(R) = h(v_1\text{'s} \oplus v_2\text{'s} \oplus \ldots \oplus v_{H-1}\text{'s})$$
$$\Phi(R) = h(\Phi(L_3) \oplus \Phi(L_4) \oplus \Phi(A) \oplus \Phi(D))$$
$$\Phi(R) = h((010) \oplus (100) \oplus (111) \oplus (101))$$
$$\Phi(R) = h(100)$$

(2). pk_i를 제외한 나머지 인자들을 하나로 결합하여 해쉬를 취하는 과정이다.

$$\Phi(R) = h((010) \oplus ((100) \oplus (111) \oplus (101)))$$
$$\Phi(R) = h((010) \oplus (110))$$
$$\Phi(R) = h(100)$$

(1)과 (2), 두 과정 모두 동일한 값으로 나타나는 증명을 통하여 1번의 연산으로 인증이 가능함을 알 수 있다. 그러나 재전송 공격에 강한 양방향 인증을 위해 nonce 값을 인자로 따로 포함시켜야한다. 또한, 비밀성이 유지되어야 할 Δ와 nonce의 노출을 막기 위해 각 값을 R_i로 XOR 연산하는 것이 필요하다. 본 논문에서 제안한 기법은 노드 수가 증가하더라도 고정된 연산을 수행하게 된다. 즉, 2번의 해쉬 연산을 제외한 주 연산은 XOR 연산이기 때문에 해쉬로만 이루어진 연산보다 비교적 연산오버헤드가 적게 든다. 특히, 기존 기법에서 제안되었던 배치정보를 이용하여 트리의 높이를 선택적으로 최적화하는 일련의 과정은 필요하지 않다.

4. 분 석

본 장에서는 기존 기법과 본 논문에서 제안한 기법의 트리 생성과 인증 시 요구되는 연산량과 메시지 요구량을 비교 분석한다.

4.1. 트리 생성 오버헤드 분석

기존 기법과 제안한 기법은 상이한 값들로부터 commitment 값을 유도하기 위해 비슷한 형태의 Merkle 트리를 이용한다. Merkle 트리의 생성 계산비용은 해쉬 연산 및 XOR 연산의 횟수의 총계로 구할 수 있다. 깊이가 H인 이진트리의 노드들의 최대수는 $2^{H+1}-1$이므로 총 $2^{H+1}-1$번의 연산이 필요하다. 본 논문에서는 해쉬 함수로서 SHA-1알고리즘을 통해 분석을 제시한다. SHA-1은 덧셈, 논리 연산(AND, OR, XOR), Shift, Rotate으로 구성된다. 표 2는 트리를 생성하기 위해 필요한 기존 기법과 제안한 기법에서의 연산의 오버헤드의 비교를 보여준다. 제안한 기법은 기존 기법에서 사용하는 해쉬 연산의 약 절반가량을 XOR 연산으로 대치하였기 때문에 그만큼의 계산 성능 향상을 기대할 수 있다.

특히, 기존 기법에서는 $\Phi(R)$을 계산하는데 있어서 높은 레벨의 값은 낮은 레벨의 값에 의존적으로 처리되어야 하므로 데이터에 의존적인 연산이 발생해야 한다. 하지만 본 논문에서 제안한 변형된 Merkle 트리는 XOR 연산의 속성을 이용하기 때문에 데이터 의존적이지 않아서 처리의 효율성을 증대시킬 수 있다.

표 2. 기존 기법과 제안한 기법의 트리 생성 오버헤드

Property \ Scheme	Existing scheme (using SHA1)	Proposed scheme (using XOR)
Key or hash size (bit)	160	160
Amount of hash operations	$2^{H-1}-1$	$2^{H}+1$
Amount of XOR operations	0	$2^{H}-2$

이와 더불어, 기존 기법은 모든 노드의 동등한 관계에 기인하여 통신과 메모리 효율성간에 극단적인 상충성(trade off)을 가지고 있으므로 이웃이 될 확률에 의한 배치정보를 이용하여 관계를 다시 효율적으로 그룹화한다. 재구성된 그룹 관계를 통해 각 그룹 간 높이를 선택하고 완성된 트리를 다시 서브트리로 줄이게 되는데, 이 작업은 $O((\log N)^3)$이상의 연산을 필요로 한다. 이것은 표2에서 나타난 결과에 부가적으로 $O((\log N)^3)$의 계산복잡도가 더 늘어나는 것으로 기존 기법에 비하여 본 논문에서 제안한 기법의 성능이 상대적으로 향상되었음을 보인다.

4.2. 인증 오버헤드 분석

공개키 인증에 있어서 실제적인 성능 비교를 위해서는 인증에 소요되는 통신비용과 계산비용을 분석해야 한다. 기존 기법에서는 공개키 암호 알고리즘인 RSA와 ECC을 센서프로세서 상에서 실험하여 인증에 따르는 오버헤드 분석결과를 제시하고 있다. 제시된 결과에서 RSA와 ECC의 오버헤드에 비하여 Du 등의 기법이 상당히 개선되었음을 확인할 수 있다[1]. 그러나 기존 기법에서는 k(검증자에게 전송하는 인자의 수)의 변수가 존재하므로 각 서브 Merkle 트리의 선택된 높이에 비례하여 통신비용과 계산비용이 증가함을 알 수 있다.

본 논문에서 제안된 기법은 암호학적 강도 유지와 인자들의 수를 줄이기 위하여 2번의 해쉬 연산만 사용하도록 하였으며 나머지 Merkle 트리의 commitment 값을 구하기 위한 연산으로 XOR 연산을 사용하였다. 그리고 challange/response 기반의 양방향 인증을 모델로 하여 3번의 전송으로 두 개체간의 공개키를 인증한다. 기존의 통일을 위해 두 기법을 양방향 인증으로 가정하여 비교한다. 이러한 가정으로 인해 제안된 기법은 기존 기법으로부터 다음의 두 가지 차이점을 갖는다.

먼저, 메시지의 길이로 양방향 전송 3단계에 쓰이는 부가인자가 6개로 고정되면서 인자의 해쉬 비트 수인 160bit에 6배된 길이가 인자들의 총합이 된다. 이는 네트워크의 크기에 무관하며 기존 기법에 $k/3$배 줄여진 결과를 나타내므로 네트워크가 커질수록 두 기법은 확연한 차이를 가지게 될 것이다. 센서네트워크에서 노드가 가장 많은 에너지를 소비하는 부분은 패킷 전송과 관련되므로 에너지 절약의 효과를 기대할 수 있다.

둘째는 계산비용으로, 기존 기법은 해쉬 연산들을 연계하여 Merkle 트리를 구성한다. 그러나 제안한 기법에서는 해쉬 연산은 최소로 사용하고 계산속도가 빠른 연산자인 XOR로 대치하였다. 단일 XOR 연산이 해쉬(SHA-1 알고리즘) 연산보다 단연 계산비용이 적으므로 오버헤드를 많이 줄일 수 있을 것이다.

표 3의 결과에서 기존 기법은 각 서브트리의 높이만큼 연산을 하고 양방향 인증을 위해 전체를 한번 더 연산해야 하기 때문에 SHA-1의 연산 1회당 7.2초의 시간[1]이 걸릴 때 k배 증가한 결과에 다시 2배의 시간이 걸림을 알 수 있다. 이에 반하여 제안한 기법은 동일한 양방향 인증을 기준으로 기존 기법보다 전송 수가 1번 더 증가하지만 고정된 11번의 XOR와 2번의 해쉬 연산만이 필요하므로 11회의 XOR 연산이 SHA-1의 1회 속도와 같다는 최악의 가정을 고려(SHA-1: 1라운드 당 최소 6번의 XOR연산이 20단계, 총 4라운드)하더라도 총 3번의 해쉬 연산 시간이 걸림을 보인다. 또한, 기존 기법은 네트워크 크기에 의존적으로 복잡도가 증가하지만, 본 논문에서 제안한 기법은 네트워크 크기와 연산의 복잡도는 관계가 없다.

표 3. 기존 기법과 제안한 기법의 양방향 인증 오버헤드

Property \ Scheme	Existing scheme (using SHA-1)	Proposed scheme (using XOR)
Key or hash size (bit)	160	160
Communication overhead (bit)	$(160\times k)\times 2$	160×6
Computation time (ms)	$(7.2\times k)\times 2$	$(7.2\times 2)+T_{XOR}$

이와 같은 결과는 해쉬 연산과 XOR 연산의 결합법칙의 차이로 인한 것으로 해쉬 연산은 순서에 의존적이기 때문에 서브트리의 높이에 따라 발생하는 인자들을 개별로 저장해야 하는 특성을 가진다. 또한, 연산 자체도 순

서에 따라 계산하여야 하므로 꼭 서브트리의 높이만큼 연산을 거쳐야 한다. 그러나 XOR는 순서에 독립적이므로 다수로 발생하는 인자들을 하나로 결합할 수 있으며 그 인자를 이용해 단 한번의 연산이면 복호가 가능하다. 본 논문에서 제안한 인증 기법은 배치정보가 불필요하였으므로 Merkle 트리만을 비교한다면 기존 기법은 $O(\log N)$개의 부가 인자와 역시 $O(\log N)$번의 연산이 필요한데 반해, 제안한 논문은 6개의 인자와 2번의 해쉬 연산 및 11번의 부수적인 XOR 연산이 필요하기 때문에 $O(\log N)$복잡도에서 상수복잡도로 계산 복잡도가 향상되었음을 확인 할 수 있다.

5. 결론 및 향후 과제

본 논문에서는 센서네트워크 환경에서 변형된 Merkle 트리를 이용한 공개키 인증 기법을 제안하였다. 우선, 기존 기법에서 제안한 Merkle 트리를 기반으로 하는 공개키 인증의 오버헤드를 줄이기 위하여 변형된 Merkle 트리를 제안하였다. 변형된 Merkle 트리는 식별자와 공개키 간의 바인딩을 위해 가벼운 일방향 해쉬 함수를 이용하고 내부 노드들의 값은 역시 경량의 XOR연산으로 결합하여 다시 마지막에 해쉬 함수를 취하였다. 일방향 해쉬 함수가 처음과 마무리 단계에서 쓰였기 때문에 역으로 원래의 값을 알아내기는 어렵다. 인증은 공개키와 부가 인자를 통해 상수복잡도의 연산을 취해도 검증이 가능하며 재전송 공격에도 강한 특성을 보인다. 이러한 특징을 통해 무선 센서 네트워크에서 많은 오버헤드를 차지하는 공개키 인증의 연산을 줄여 공개키 암호 사용의 가능성을 제시하였다.

그러나 공개키 암호를 전면적으로 사용하려면 암호화에 드는 연산비용이 더 고려되어야 하기 때문에 한정적인 연산에서만 사용하고 나머지 통신에서는 비밀키를 사용할 수 있다. 이에 적합한 키 생성과 키 관리 기법을 추가하는 것이 과제로 남아있으며 다른 경량 프로토콜, 공개키 기법들과의 성능을 비교하고 제안한 기법의 모듈 성능을 테스트 하는 작업도 수반되어야 한다.

6. 참고 문헌

[1] W. Du, R. Wang and P. Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks", *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 58-67 5. 2005.

[2] A. Perrig, D. Wagner and J. Stankovic, "Security in Wireless Sensor Networks", *Communications of the ACM*, Vol. 47, No.6, pp. 53-57, 6. 2004.

[3] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. D. Tygar, "SPINS: Security protocols for sensor networks", *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 189-199, 2001

[4] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks", *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41-47, 11. 2002.

[5] H. Chan, A. Perrig and D. Song, "Random key predistribution schemes for sensor networks", *Proceedings of 2003 IEEE Symposium on Security and Privacy*, pp. 197-213, 5. 2003.

[6] W. Du, J. Deng, Y. S. Han and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks", *Proceedings of the 10th ACM conference on Computer and communications security*, pp. 42-51, 8. 2003.

[7] D. J. Malan, M. Welsh and M. D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography", *The First IEEE International Conference on Sensor and Ad Hoc Communications and Networks*, 8. 2004.

[8] 서석충, 김형찬, R. S. Ramakrishna, "무선 센서 네트워크에서 타원곡선 암호를 이용한 공유키 설정에 기반한 보안 프로토콜", *한국정보처리학회 추계학술발표대회*, 제12권, 2호, pp. 873-876, 11. 2005

[9] Crossbow Technology Inc, *Wireless sensor networks*, http://www.xbow.com/. 2004.

[10] R. C. Merkle, "A Certified Digital Signature", *CRYPTO'89*, pp 218-238, 1989