情報保護學會論文誌
第16卷 第6號, 2006. 12

인증지연 없는 멀티-레벨 μTESLA의 새로운 구성

임 채 훈
세종대학교 컴퓨터공학부

New Constructions of Multi-level μTESLA with Immediate Authentication

Chae Hoon Lim
Dept. of Computer Engineering, Sejong University

요 약

멀티-레벨 μTESLA는 다수의 키 체인을 이용하여 μTESLA를 긴 수명의 센서 네트워크에 효율적으로 확장시킨 프로토콜이다. 본 논문에서는 멀티-레벨 μTESLA의 변형으로 하위 키체인 초기값의 즉시 인증이 가능한 새로운 멀티-레벨 μTESLA의 구성방안들을 제안한다. 제안방식들은 원래의 멀티-레벨 μTESLA의 즉시 인증 가능한 변형에 비해 계산/통신 효율성이나 패킷 분실에 대한 저항성 등 모든 면에서 보다 우수한 특성을 제공한다.

ABSTRACT

Multi-level μTESLA is an efficient extension to μTESLA to provide an extended lifetime for long-lived sensor networks. This paper presents new constructions of multi-level μTESLA with immediate authentication of key chain commitments. The proposed constructions are shown to be more efficient and robust than the previous multi-level μ TESLA extension.

Keywords : *Sensor networks, Broadcast authentication, multi-level μTESLA*

Ⅰ. 서 론

센서 네트워크는 매우 제약된 자원만을 갖는 많은 수의 센서노드와 하나 혹은 소수의 강력한 베이스 스테이션(BS: Base Station)으로 구성된다. 센서노드들은 BS로부터 오는 명령과 데이터에 의해 통제되며, BS는 인터넷을 통해 센서 네트워크 메니저와 연결되어 센서노드들과의 게이트웨이 역할을 하게 된다. 따라서 BS로부터 센서노드들에게 브로드캐스트되는 명령과 데이터에 대한 인증은 센서 네트워크의 안전한 동작을 위해 가장 필수적인 보안 서비스의 하나이다. 브로드캐스트 인증은 디지털 서명과 같은 공개키 암호를 이용하면 쉽게 해결될 수 있지만 매우 제약된 자원을 갖는 센서 네트워크에서 무거운 공개키 암호의 빈번한 사용은 네트워크 수명을 단축시킬 수 있으므로 가능한 피하는 것이 바람직하다.

비밀키 암호만을 이용한 브로드캐스트 인증 기법으로 멀티캐스트 인증을 위해 개발된 TESLA를 센서 네트워크 환경에 최적화시킨 μTESLA가 대표적이다[1,2]. μTESLA는 일방향 함수를 통해 생성된 키체인을 역방향으로 이용하면서 동기화된 클럭을 기반으로 인증키의 지연노출을 통해 비밀키 암호만으로 공개키 암호와 같은 비대칭성을 제공한다. μ TESLA는 그러나 긴 수명을 갖는 센서 네트워크에

접수일: 2006년 11월 2일 : 채택일: 2006년 11월 24일
† 주저자, chlim@sejong.ac.kr

적용하기에는 지나치게 긴 키체인을 필요로 하여 비현실적이다. 이를 보완하기 위해 복수개의 키체인을 이용한 멀티-레벨 μTESLA가 제안되었다[3].

멀티-레벨 μTESLA는 상위 레벨의 키체인을 이용하여 하위 레벨 키체인의 초기값들을 인증하며 최하위 레벨의 키체인이 실제로 데이터 인증에 사용된다. 그러나 멀티-레벨 μTESLA의 가장 큰 약점의 하나로 인증지연의 문제가 지적된다. 즉 상위 레벨 키체인을 통해 분배되는 키체인 파라미터들에 대해 수신 즉시 인증이 불가능하여 서비스 거부(DoS: Denial of Service) 공격의 주요 목표가 될 수 있다. 이러한 문제를 해결하기 위해 전체 하위 레벨 키체인 초기값들에 대한 Merkle hash tree를 구축하여 그 루트(root)를 각 센서노드에게 사전분배하고, 각 키체인의 초기값 인증을 위해 루트로부터의 인증경로를 전송하는 방법이 제안되었다[4]. 그러나 이 방법은 인증경로의 길이가 센서 네트워크의 수명에 로가리즘적으로 의존하여 1년 정도의 수명을 갖는 센서 네트워크만 하더라도 키체인 초기값의 분배를 위해 20개 전후의 해쉬값을 전송하여야 하므로 실용성이 떨어진다.

멀티-레벨 μTESLA에서도 키체인 초기값의 즉시 인증이 가능한 변형된 버전을 제안하였으나, 본 논문에서는 이를 분석, 보다 효율적이며 공격 저항성이 강한 멀티-레벨 μTESLA의 구성방법들을 제안한다.

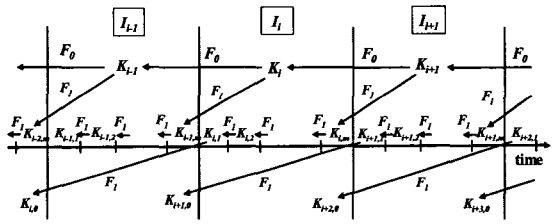
II. μTESLA와 멀티-레벨 μTESLA

우선 μTESLA의 동작원리를 간략히 살펴보자. F를 PRF(pseudo-random function)라 두고 키체인의 길이를 n이라 두자. BS (혹은 네트워크 서버)는 키체인의 마지막 값 K_n을 랜덤하게 선택하여 나머지 키체인값 $K_i=F(K_{i+1})$ $(0\le i\langle n)$를 생성한다. 각 센서노드는 BS와 클럭을 동기화시킨 후 키체인 초기값 K_0와 다른 변수들 (키체인의 시작시점 T_0, 한 구간의 시간간격 Δ, 송/수신자 사이의 최대 클럭 차이 δ, 키의 지연노출 간격 d 등)로 초기화된다. 키 K_i는 시간간격 Δ의 i-번째 구간 I_i에서의 인증키로 할당된다.

BS는 구간 I_j에서 전송할 메시지 m이 있으면, 구간 인덱스 j, K_j로 생성된 MAC $\sigma_j=MAC_{K_j}(m)$, 구간 I_{j-d}에 할당된 키 K_{j-d} 등을 m과 함께 전송한다. 즉 $P_j=\langle j,\ m\ ,\sigma_j,\ K_{j-d}\rangle$. 각 센서노드는 패킷 P_j를 받으면 일단 수신시간 T_c와 함께 저장해 두었다가 구간 I_{j+d}에서 키 K_j가 노출되면 이를 이용하여 보안조건과 MAC을 검사하여 메시지의 진실성 여부를 결정한다. 우선 이 MAC이 인증키 K_j가 노출되기 전에 전송되었는지를 확인하기 위해 수신시간에 대해 보안조건 $\lfloor (T_c+\delta+T_0)/\Delta \rfloor \langle j+d$를 검사하고, 이 조건이 만족되면 K_j를 이용하여 MAC σ_j를 검증한다.

하나의 키체인 만을 이용하는 μTESLA는 긴 수명의 센서 네트워크에 적용하기에는 너무 긴 키체인이 필요하여 BS나 센서노드 모두에게 많은 자원의 소모를 요구한다. 멀티-레벨 μTESLA는 복수개의 상위 레벨 키체인을 이용하여 짧은 길이의 하위 레벨 키체인 초기값들을 분배함으로써 μTESLA를 효율적으로 확장시킨 것이다[3]. 편의상 Two-level μTESLA를 예로들어 설명한다.

우선 네트워크 서버는 네트워크의 예상 수명을 시간간격 Δ_0를 갖는 n개의 구간으로 나눈다. 이를 $I_1,\ I_2,\ \ldots,\ I_n$이라 두고 각 구간 I_i의 시작시점을 T_i라 두자. 이 상위 레벨 키체인의 구간 I_i에서 사용될 키 K_i는 K_n을 랜덤하게 선택한 후 $K_i=F_0(K_{i+1})$ $(0\le i\langle n)$와 같이 생성한다. 상위 레벨의 각 구간 I_i는 다시 더 작은 시간간격 Δ_1을 갖는 m개의 소구간 $I_{i,j}$ $(1\le j\le m)$로 나누어져 하위 레벨 키체인을 형성하며, 여기에 사용될 인증키는 $K_{i,j}=F_1(K_{i,j+1})(0\le j\langle m)$, $K_{i,m}=F_1(K_{i+1})$과 같이 상위



〔그림 1〕 Two-level μTESLA의 키체인 구성[3]

레벨 구간 I_{i+1}의 인증키 K_{i+1}로부터 생성된다 (F_0와 F_1은 서로 다른 PRF). 편의상 모든 하위 레벨 소구간의 시간간격 Δ_1 및 키 지연 간격 d는 동일하다고 가정한다 (〔그림 1〕 참조).

상위 레벨 키체인의 시간간격 Δ_0은 네트워크 지연이나 최대 클럭 차이 δ에 비해 충분히 길므로 여기서의 키 노출 지연간격은 1로 가정한다 (즉 K_i는

구간 I_{i+1}에서 노출된다). 따라서 상위 레벨 키체인의 구간 I_i에서의 보안조건은 수신시간 T_c에 대해 $T_c+\delta < T_{i+1}$을 만족하는지를 검사하는 것이다. 하위 레벨 키 $K_{i,j}$로 인증되는 메시지에 대한 보안조건은 상위 레벨 키체인의 시작시점 T_1을 기준으로 조건 $i' < (i-1)*m+j+d$, $i' = \lfloor (T_c - T_1 + \delta)/\triangle_1 \rfloor$을 만족하는지 검사하면 된다.

센서노드들이 하위 레벨 키체인 $\langle K_{i,0} \rangle$을 이용하기 위해서는 키체인의 초기값 $K_{i,0}$을 T_i 전에 인증하여야 한다. BS는 각 구간 I_i에서 다음과 같은 *CDM* (commitment distribution message)을 방송함으로써 이를 수행한다 (MAC키는 K_i로부터 유도하여 사용하는 것이 바람직하나 편의상 같은 키로 표기한다):

$$CDM_i = i \mid K_{i+2,0} \mid MAC_{K_i}(i \mid K_{i+2,0}) \mid K_{i-1} \quad (1)$$

따라서 구간 I_i에서 키체인 $\langle K_{i,0} \rangle$을 이용하기 위해서는 BS가 구간 I_{i-2}에서 CDM_{i-2}를 분배해야 하며, 센서노드는 구간 I_{i-1}에서 CDM_{i-1}를 받으면 여기서 노출된 키 K_{i-2}를 이용하여 CDM_{i-2}에 포함된 MAC을 확인함으로써 인증된 키 $K_{i,0}$를 얻을 수 있게 된다.

위의 CDM 분배 방식에서 현재의 CDM은 다음 CDM을 수신한 후에야 검증이 가능한다는 인증지연이 문제가 된다. CDM은 한번 놓치면 상위 레벨 한 구간 (상당히 긴 시간) 동안 메시지 인증이 불가능한 매우 중요한 메세지이므로 DoS 공격의 주요 목표가 될 수 있다. 통신상의 에러나 DoS 공격에 저항성을 주기 위해 CDM를 랜덤하게 여러 번 반복 전송하며, DoS 공격으로 정해진 수 이상의 많은 CDM이 수신되는 경우 버퍼의 고갈을 막기 위해 각 센서노드는 이들 중 랜덤하게 하나 또는 소수의 CDM만을 확률적으로 선택하는 방법 (random selection method)을 사용할 수 있다[3]. 보다 적극적인 방법으로는 모든 CDM을 사전에 생성하여 이들을 상호 연쇄시킴으로써 즉시 인증이 가능한 변형을 사용할 수도 있다[3]:

$$CDM_i = CDM_i' \mid MAC_{K_i}(CDM_i') \mid K_{i-1},$$
$$CDM_i' = i \mid K_{i+1,0} \mid H(CDM_{i+1}) \quad (2)$$

위의 변형에서는 센서노드가 이전 CDM을 제대로 수신했을 때에만 여기에 포함된 다음 CDM의 해쉬값을 이용해 수신 CDM을 즉시 인증 할 수 있다. 대부분의 센서노드들이 CDM을 정상적으로 수신한다고 보면, 이 변형은 대부분의 경우 CDM의 즉시 인증을 가능하게 한다. 그러나 역시 이전 CDM의 분실시에는 DoS 공격이 가능하므로 random selection method 등을 같이 적용해 대비해야 한다.

Ⅲ. 제안된 멀티-레벨 μTESLA

먼저 TinyOS가 지원하는 표준 패킷 포맷은 5바이트 헤더, 최대 29바이트 데이터, 그리고 2바이트 트레일러로 구성됨을 상기하자. 식 (2)의 즉시 인증 가능한 CDM을 전송하기 위해서는 두개의 Tiny OS 패킷이 필요하므로 식 (1)의 CDM에 비해 상당한 통신량 증가를 수반한다. 키체인의 레벨을 구분하기 위한 Level(1바이트)과 구간 인덱스 i(4바이트)를 포함하면 식 (1)의 CDM은 정확히 Tiny OS 패킷의 최대 페이로드 사이즈 29바이트를 갖지만, 식 (2)의 CDM은 증가된 8바이트로 인해 새로운 패킷을 하나 더 형성해야 하므로 추가로 15바이트의 전송이 필요하여 41.7%의 통신량 증가를 초래한다.

우선 식 (2)의 CDM에서 MAC값을 포함시킨 것은 이전 CDM이 전혀 없는 상태에서 CDM_i와 CDM_{i+1}을 받았을때 모든 센서들이 항상 검증 가능한 키체인 $\langle K_0 \rangle$만을 이용해 $K_{i+1,0}$와 $H(CDM_{i+1})$를 인증할 수 있도록 하기 위함임을 주목하자. 따라서 식 (2)에 의한 CDM 분배는 다음과 같이 이중 키체인의 형태로 단순화시킬 수 있다 (키체인 $\langle K_0 \rangle$는 식 (2)에서처럼 CDM에 대한 loss tolerance 기능을 주로 제공하며, 키체인 $\langle S_0 \rangle$는 식 (2)의 MAC을 대신하여 즉시 인증 기능을 주로 제공한다):

$$\begin{aligned} CDM_i &= i \mid K_{i+1,0} \mid S_i \mid K_i \\ K_i &= F_0(K_{i+1}, i+1) \\ K_{i,j} &= F_1(K_{i,j+1}, j+1) \end{aligned} \quad (3)$$

여기서 F_0, F_1, F_2의 입력값들은 이전과 다르게 키와 메세지를 콤마로 분리하여 표기하였다. 센서와 같은 제약된 환경에서는 해쉬함수보다 블럭암호를 이용하여 PRF를 구현하는 것이 보다 효율적인데, 이

경우 F_0, F_1은 암호화 함수 $X_i = E_{X_{i+1}}(C_j|i+1)$ (C_j는 F_0, F_1을 구분하기 위한 서로 다른 상수)로, F_2는 $S_i = MAC_{S_{i+1}}(K_{i+1,0}|K_{i+2,0}|K_{i+1})$와 같이 구현할 수 있다. 식 (3)의 키체인은 먼저 S_{n-1}, K_n을 랜덤하게 선택한 후 ($K_{n+1,0}$, S_n은 NULL로 설정, 마지막 구간 I_n에서는 K_n만 노출해 주면 됨) F 함수들을 이용하여 $K_{1,0}$, S_0, K_0까지 순차적으로 생성하고 이 마지막 값들을 각 센서노드에게 사전 분배하면 된다.

식 (3)에 따른 CDM 분배는 식 (2)에 의한 CDM 분배와 동등한 효과를 주지만 CDM을 하나의 TinyOS 패킷만으로 전송 가능하므로 식 (2)의 방식에 비해 통신량을 40% 이상 줄일 수 있다. 그러나 식 (3)에 의한 CDM 분배는 식 (2)를 이용하는 경우와 마찬가지로 이전 CDM을 정상적으로 (즉 에러 없이) 수신한 경우에만 현재 수신된 CDM을 즉시 인증할 수 있다. 따라서 이전 CDM을 제대로 수신하지 못한 경우에는 식 (1)에 의한 CDM 분배에서와 마찬가지로 인증지연이 발생하게 된다.

이제 식 (3)의 CDM 분배를 다음과 같이 변경해 보자 (K_i, $K_{i,j}$의 생성은 식 (3)에서와 동일):

$$\begin{aligned} CDM_i &= i \mid K_{i+1,0} \mid S_i \mid K_{i+1} \oplus S_{i+1} \\ S_i &= F_2(S_{i+1}, K_{i+2,0} | K_{i+2} \oplus S_{i+2}) \end{aligned} \quad (4)$$

식 (4)의 키체인은 S_n, K_n을 랜덤하게 선택한 후 ($K_{n+1,0}$은 NULL, $K_{n,0}$은 임의의 값으로 설정) F 함수들을 이용하여 중간 값들을 차례로 생성하고 마지막 값 $K_{1,0}$, S_0, K_1을 각 센서노드에 사전 분배하면 된다. 식 (4)에 의한 CDM 분배의 장점은 중간에 하나의 CDM을 놓친 경우라도 그 전 CDM만 있으면 다음 CDM을 즉시 인증할 수 있다는 점이다. 현재 구간에서 CDM_i를 수신하였다고 하자. 만일 CDM_{i-1}을 정상적으로 수신한 경우라면 $\langle S_0 \rangle$ 키체인을 통해 즉시 인증이 가능하며, 또한 인증된 CDM_{i-1}의 마지막 필드 $K_i \oplus S_i$로부터 K_i를 복구하여 키체인 $\langle K_0 \rangle$를 통해 검증함으로써 인증된 K_i값을 얻을 수 있다. 만일 CDM_{i-1}은 놓쳤지만 인증된 CDM_{i-2}를 가지고 있는 경우라면 현재 수신된 CDM_i로부터 S_{i-1}을 계산하여 저장중인 CDM_{i-2}의 마지막 필드 $K_{i-1} \oplus S_{i-1}$로부터 K_{i-1}을 복구한 후 키체인 $\langle K_0 \rangle$를 통해 검증함으로써 현재 수신된 CDM_i을 즉시 인증할 수 있다.

위의 CDM에 대한 loss-tolerance 특성은 패킷 에러 확률이 높은 센서 네트워크 환경에서 대단히 중요한 성질이다. 다양한 이유로 한 구간의 반복 전송되는 모든 CDM들이 분실될 확률을 p라고 하면, 식 (2)나 (3)의 경우 CDM의 즉시 인증 확률이 $1-p$인 반면 식 (4)에서는 이 확률이 $1-p^2$으로 높아진다. 예를들어 $p=0.1$, 즉 패킷 분실 확률이 10%라면 식 (2)나 (3)의 경우 CDM의 즉시 인증 확률은 90%이지만 식 (4)의 경우는 그 확률이 99%로 높아진다.

그러나 식 (4)에 의한 CDM 분배는 다른 측면에서 단점도 존재한다. 그 하나는 두개 이상의 연속된 CDM을 모두 놓친 경우 식 (3)이 제공하던 메시지의 loss tolerance 기능, 즉 이전 구간 I_{i-1}의 모든 소구간 메시지를 저장해 둔다면 다음 CDM_i 수신 즉시 여기에 포함된 K_i로부터 이전 소구간의 모든 $K_{i-1,j}$값을 생성하여 저장된 메시지들을 인증할 수 있었던 기능을 제공할 수 없다는 점이다. 이는 이전 CDM의 수신 여부와 상관없이 항상 인증이 가능하던 키체인 $\langle K_0 \rangle$상의 현재 값이 CDM을 통해 노출되지 않기 때문이다. 또 다른 관련 단점으로는 식 (3)의 경우 이전 CDM의 수신 여부와 상관없이 임의의 연속된 두 CDM, CDM_{i-1}과 CDM_i를 정상적으로 수신한 경우 두 CDM에 포함된 $K_{i,0}$와 $K_{i+1,0}$를 모두 인증할 수 있지만 (따라서 구간 I_i의 모든 메시지들을 저장해 둔다면 CDM_i 수신 후 이들을 인증할 수 있지만), 식 (4)에 의한 CDM의 경우는 $K_{i+1,0}$만을 인증할 수 있다 (다음 구간의 메시지들만 인증할 수 있다). 이는 S_{i-1}의 계산에 $K_{i+1,0}$만을 포함하고 있기 때문이다. 그러나 한 구간의 모든 메시지들을 버퍼링한 후 다음 구간의 CDM을 이용하여 한꺼번에 인증하는 것은 최하위 레벨 μTESLA에서의 메시지 지연인증보다 훨씬 DoS 공격에 취약하므로 실제로 그 효용성은 의문이며 오히려 공격이나 에러에 대한 저항성이 강한 식 (4)에 의한 CDM 분배가 더 바람직한 응용도 존재할 수 있다.

마지막으로 식 (3)의 CDM에 $K_{i,0}$을 추가로 전송하는 다음과 같은 CDM 분배를 생각해 보자:

$$\begin{aligned} CDM_i &= i \mid K_{i,0} | K_{i+1,0} \mid S_i \mid K_i \\ S_i &= F_2(S_{i+1}, K_{i+1,0} | K_{i+2,0} | K_{i+1}) \end{aligned} \quad (5)$$

식 (5)에 의한 CDM 분배는 추가된 $K_{i,0}$로 인해 식

(2)에서처럼 TinyOS 패킷 포맷 기준으로 두개의 패킷으로 나누어 전송해야 하므로 통신량의 증가를 수반하지만, loss tolerance면에서는 식 (3)과 식 (4)에 의한 CDM 분배의 장점들을 모두 제공하는 우수한 특성을 갖는다. 더욱이 $k(k\geq 2)$개 이상의 연속된 CDM이 모두 소실된 경우라도 $(k-1)(m+1)$번의 F_1 연산만 추가적으로 수행한다면 현재 수신된 CDM의 즉시 인증이 항상 가능하다. 이는 임의의 구간 I_{i+1}에서 CDM에 포함된 K_{i+1}로부터 I_{i-1} 구간의 $K_{i,0}$는 $(m+1)$번의 F_1 연산만 수행하면 계산할 수 있고 이는 키체인 $\langle K_0\rangle$를 통해 임의의 하위구간까지 확장 가능하다는 사실로부터 쉽게 알 수 있다. 효율성을 위해 최하위 레벨의 키체인 길이는 통상 작게 잡는 것이 일반적이며 또한 $(k-1)(m+1)$번의 F_1 연산은 한번만 수행하면 되므로 많은 경우 k를 3이나 4 정도까지 확장하는 것은 그다지 문제가 되지 않을 수 있다. 따라서 식 (5)는 패킷 에러 확률이 높은 열악한 환경이나 DoS 공격에 대한 높은 저항성이 요구되는 응용 등에서는 이전 방식들보다 오히려 더 효율적인 CDM 분배방법이 될 수 있을 것이다.

마지막으로 BS에서 키체인의 사전생성에 대한 계산/저장 복잡도를 생각해 보자. 이는 분명 적은 양은 아니지만 BS의 상대적으로 강력한 컴퓨팅 환경이나 혹은 필요하면 중앙의 네트워크 메니저에서 대신 계산하여 (안전한 유선채널을 통해) 주기적으로 BS로 전송할 수도 있음을 고려하면 이것이 현실적으로 문제가 될 소지는 별로 없다. 예를들어 좀 극단적인 예로, 5년 정도의 긴 수명을 갖는 센서 네트워크에서 레벨 2의 키체인 길이를 가능한 길게 잡고, 레벨 3 키체인을 사용하는 대신 레벨 2 키체인 초기값의 해쉬값을 모든 센서노드에 초기화시키는 시나리오를 가정보자. 구체적으로 레벨 1의 한 구간간격을 100msec, 키체인 길이를 600, 레벨 2의 한 구간간격을 1분, 키체인 길이를 43200으로 가정하면, 매 30일마다 레벨 2의 키체인 초기값을 전송해야 하고 이는 센서가 저장하고 있는 해쉬값으로 즉시 인증 가능하다. 이 경우 센서노드는 60개의 해쉬값을 저장하면 되며 (480바이트), BS는 60개의 독립적인 레벨 2 키체인의 마지막 값 (K_n, S_n)을 저장하고 있다가 매달 43200개의 $\langle K_0\rangle$, $\langle S_0\rangle$ 키체인 전체를 생성하여 저장하고, 또한 매 분마다 600개의 레벨 1 키체인 $\langle K_{i,0}\rangle$을 생성하여 저장한 후 사용하면 된다 (총 약 1.04M 바이트). 하나의 레벨 2 키체인 생성에는 약 2600만번의 PRF 연산이 필요하나 이는 노트북 정도의 BS를 가정하면 몇 분이내에 계산 가능한 양이다. 만일 이것이 BS에 지나친 부담이라면 매달 해당 키체인을 중앙의 네트워크 메니저로부터 다운받을 수도 있다. 물론 레벨 3 키체인을 두어 센서의 메모리를 절약하고 BS의 부하를 줄이는 것이 보다 바람직할 것이다.

Ⅳ. 결 론

μTESLA는 비록 클럭 동기화 문제나 메시지의 지연인증으로 인한 DoS 공격 가능성 등 본질적인 문제점들이 지적되기는 하지만, 매우 제한된 자원을 갖는 센서 네트워크 환경에서 그래도 가장 현실적이며 에너지 효율적인 브로드캐스트 인증 프로토콜의 하나로 인식되고 있다. 본 논문에서는 μTESLA의 멀티-레벨 확장버전에서 CDM 분배시 발생되는 인증지연 문제를 해결할 수 있는 방안으로 기존의 즉시 인증 가능한 변형보다 더 효율적이며 패킷 분실에 대한 저항성을 높인 새로운 형태의 상위 레벨 키체인 구성방안들을 제안하였다. 제안방식들은 효율성이나 DoS 공격에 대한 저항성면에서 기존 방식에 비해 보다 우수한 특성을 가지므로 멀티-레벨 μTESLA를 위한 CDM 분배의 새로운 대안이 될 수 있을 것이다.

참 고 문 헌

[1] A.Perrig, R.Canetti, D.Song and D.Tygar, "Efficient authentication and signing of multicast streams over lossy channels," *IEEE Symp. on Security and Privacy*, Feb.2001.

[2] A.Perrig, R.Szewczyk, V.Wen, D. Culler, and D.Tygar, "SPINS: Security protocols for sensor networks," *MobiCom 2001*, Jul. 2001.

[3] D.Liu and P.Ning, "Multi-level μTESLA: Broadcast authentication for distributed sensor networks," *ACM Trans. in Embedded Computing Systems*, 3(4), pp.800-836, 2004.

[4] D.Liu, P.Ning, S.Zhu and S.Jajo dia. "Practical broadcast authentication in sensor networks," *Mobi Quitous 2005*, Jul. 2005.