情報保護學會論文誌
第20卷 第2號, 2010. 4

데이터 손실에 강하고 효율적 연산을 지원하는 XOR 체인을 이용한 트리기반 μTESLA 프로토콜 개선*

여 돈 구,† 장 재 훈, 최 현 우, 염 흥 열‡
순천향대학교

Improved Tree-Based μTESLA Broadcast Authentication Protocol Based on XOR Chain for Data-Loss Tolerant and Gigh-Efficiency*

Don-Gu Yeo,† Jae-Hoon Jang, Hyun-Woo Choi, Heung-Youl Youm‡
Soonchunhyang University

요 약

센서 네트워크에서 송신자와 수신자 간의 인증을 위한 μTESLA 브로드캐스트 인증 기법은 다수의 연구자들에 의해 연구되어 왔지만, 인증지연 문제를 내포하고 있다. Tree-based μTESLA는 μTESLA의 대표적인 인증지연 문제를 해결하였지만, 머클 해쉬 트리 기반 인증서 구조로 인해 메시지 인증시 송신자나 키 체인의 수에 따라 가변적인 오버헤드를 가지는 문제점이 있다. μTPCT-based μTESLA는 하위 머클 해쉬 트리 기반 인증서 구조를 해쉬 체인으로 변경함으로써 고정된 연산량 가지는 인증서 구조를 제안하였다. 하지만, 순차적으로 분배되는 해쉬 체인 값만으로 μTESLA 파라메타를 인증하므로 데이터 손실률이 높은 네트워크에서는 성공적인 인증을 보장할 수 없다. 본 논문에서는 XOR 체인 기반 인증서 구조를 이용함으로써 μTPCT-based μTESLA의 장점인 고정된 연산량과 Tree-based μTESLA의 장점인 데이터 손실에도 관대한 인증서 체인 구조를 제안하고자 한다.

ABSTRACT

μTESLA broadcast authentication protocol have been developed by many researchers for providing authenticated broadcasting message between receiver and sender in sensor networks. Those cause authentication delay Tree-based μTESLA[3] solves the problem of authentication delay. But, it has new problems from Merkel hash tree certificate structure. Such as an increase in quantity of data transmission and computation according to the number of sender or parameter of μTESLA chain. μTPCT-based μTESLA[4] has an advantages, such as a fixed computation cost by altered Low-level Merkel has tree to hash chain. However, it only use the sequential values of Hash chain to authenticate μTESLA parameters. So, It can't ensure the success of authentication in lossy sensor network. This paper is to propose the improved method for Tree-based μTESLA by using XOR-based chain. The proposed scheme provide advantages such as a fixed computation cost with μTPCT-based μ TESLA and a message loss-tolerant with Tree-based μTESLA.

Keywords: XORC-based μTESLA, μTPCT-based μTESLA, Tree-based μTESLA, Broadcast Authentication

접수일(2009년 10월 19일), 수정일(2009년 12월 21일), 게재확정일(2010년 2월 23일)
*본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음. (NIPA-2009-(C1090-0902-0016))
† 주저자, h7ei@sch.ac.kr
‡ 교신저자, hyyoum@sch.ac.kr

I. 서 론

센서 네트워크(Sensor Network)는 주변 환경 정보를 수집하는 수신자(Sensor Node)와 수신자를 컨트롤하며 수신자로부터 수집된 정보를 모으고 센서 네트워크 외부와 통신할 수 있는 송신자(Base Station)로 구성된다. 센서 네트워크의 특징 중 하나는 수신자의 파워, 저장, 연산 능력이 제약적이라는 것이다. 이러한 특징은 센서 네트워크 전체의 수명에도 영향을 미치게 되기 때문에, 송·수신자간 통신 중 수신자의 부하를 최대한 줄이려는 연구가 활발하게 진행되어왔다.

센서 네트워크에서는 송신자와 수신자 간에 메시지 전달에 있어 무선 통신 기술을 이용한다. 무선 통신에 사용되는 브로드캐스트 방식은 송신자의 전파 범위 안에 있는 수신자라면 누구든지 메시지를 획득할 수 있다. 이러한 통신 방식은 광범위한 센서 네트워크에서의 센서 노드들을 관리하는데 있어 효율적이다. 다만, 브로드캐스트된 메시지들을 인증하는데 있어 센서 네트워크의 낮은 대역폭, 무선 통신의 간헐적인 통신 두절, 센서 노드의 자원 제약적인 특징 등이 제약사항으로 작용한다. 그렇기 때문에 기존의 유선 네트워크에서 사용하던 브로드캐스트 방식에서의 인증 방식이나, 기존 보안 프로토콜들을 센서 네트워크에 그대로 적용할 수가 없다.

2001년 ACM에 게재된 SPIN(Security Protocol for Sensor Networks)[1]은 일반 PC급에서 디지털 서명을 이용한 브로드캐스트 인증 방식을 지원하는 TESLA 프로토콜을 센서 네트워크에 적용한 μTESLA를 선보였다. μTESLA는 해쉬 함수를 이용하여 해쉬 체인을 만들고, 체인이 생성된 방향과 반대 방향으로 생성된 키를 브로드캐스트 한다. 한번 공개된 키는 다음 키가 공개되기 이전 시점까지만 사용함으로서 비대칭 키와 유사한 특징을 갖는다. 하지만, 하나의 키 체인으로 센서 네트워크의 수명을 포함(cover)하기에는 키 체인의 각 구간($Interval \triangle_0$)이 상당히 길어지기 때문에 인증지연 문제가 발생한다. 이후, 다수의 짧은 구간의 키 체인을 계층적으로 연결하여 인증지연 시간을 줄인 Multi-Level μTESLA 프로토콜[2]를 비롯하여, 광범위한 센서 네트워크에서 다수의 송신자를 고려한 Tree-based μTESLA 프로토콜[3] 등이 제안되었다.

μTPCT-based μTESLA 프로토콜[4]은 Tree-based μTESLA 프로토콜에서 송신자가 소유하는 키 체인의 수가 늘어남에 따라 센서 노드에서의 연산량이 증가하는 문제를 해결하였다. 이 기법은 ITU-T와 ISO/IEC에서 국제 표준화로 추진 중인 USN (Ubiquitous Sensor Network)을 위한 보안 프레임워크(X.usnsec-1)[6]에서 센서 네트워크에서의 브로드캐스트 인증 기법으로 채택되었다.

본 논문에서는 μTPCT-based μTESLA 프로토콜의 해쉬 체인 기반 인증서 구조가 연속적인 통신 두절시 메시지 인증이 불가능하다는 점을 보이고, 이 문제를 해결하기 위해 [그림 3]의 해쉬 체인 기반 인증서 구조를 [그림 4]의 XOR 연산을 이용해 값을 연결한 XORC(eXclusive-OR Chain) 기반 인증서 구조로 변경함으로써 센서 노드 측면에서 메시지 인증을 위해 고정적인 연산량을 소비하며, 즉시인증도 지원한다. μTESLA 프로토콜에 XOR 연산을 적용한 연구는 XORC가 최초이다. 2장에서는 이전에 연구되었던 μTESLA 기법들에 대해 알아보고, 3장에서는 제안하고자 하는 XORC 구조가 갖는 연산적 특성 및 구조적 특성에 대해 알아본다. 이후, 4장에서는 제안하는 기법의 효율성 및 안전성에 대해 분석하고, 5장에서 결론을 맺고자 한다.

II. 관련 연구

이번 장에서는 이전에 제안되었던 μTESLA 프로토콜의 특성들에 대해 살펴보고자 한다. 들어가기에 앞서, 각 논문에서 사용하는 용어에 차이가 있으므로 이해를 돕기 위해 용어와 약어에 대해 다음과 같이 정의를 한다.

[표 1]은 본 논문에서 사용되는 용어와 약어에 대한 설명을 나타내고, [표 2]는 참조한 논문들에서 사용한 용어와 약어들이다. 각 참조 논문에서의 역할을 기준으로 유사 용어를 묶어 분류하였으며, 제안하는 논문에서는 대표 용어로 사용하도록 한다.

다음으로 센서 네트워크의 구조를 프로토콜이 단일 송신자를 고려하는지, 다중 송신자를 고려하는지에 따라서 아래와 같이 분류하고, 각 프로토콜의 구조적 차이와 이용하는 인증 기법에 대하여 알아보고자 한다.

(1) 단일 송신자를 고려한 센서 네트워크 : 센서 네트워크에 하나의 유선 혹은 무선으로 연결된 단일 송신자가 있고, 주변에 수신자가 위치한 네트워크 구조로 소규모의 센서 네트워크에 적합하다. 적용 가능한 μTESLA 프로토콜은 Original μTESLA, Multi-Level μTESLA이다.

〔표 1〕 용어 및 약어 설명

용어 및 약어	설명
N	센서 네트워크에 있는 송신자의 수를 의미한다.
n	센서 네트워크에 있는 한 송신자가 갖는 긴 $Interval$을 갖는 키 체인의 수를 의미한다.
m	하위 트리에 포함되는 짧은 $Interval$을 갖는 키 체인의 수를 의미한다.
h	HASH 함수를 1회 연산하는데 걸리는 시간을 의미한다.
x	XOR 함수를 1회 연산하는데 걸리는 시간을 의미한다.
Δ_0, $Interval_j$	상위 레벨 키 체인의 긴 구간을 나타낸다.
Δ_1, $Interval_{ji}$	하위 레벨 키 체인의 긴 구간을 나타낸다.
$x\|\|y$	x와 y의 연접(concatenation)을 의미한다.
CS	센서 네트워크를 관리하는 중앙 서버(Central Server)를 의미한다.
BS	센서 네트워크에서 수신자에게 메시지를 배포하는 기지국을 의미한다.
SN	센서 네트워크에서 주변의 정보를 수집하는 장치(Sensor Node)를 의미한다.
$Tree_R$	센서 네트워크의 송신자 각각의 인증서를 리프 노드로 구성한 중앙서버의 인증서 트리를 의미한다.
$Tree_j$	송신자j가 사용할 키 체인 정보를 리프 노드로 구성한 송신자j의 인증서 트리를 의미한다.
$Root_R$	중앙서버의 인증서 트리의 루트 값으로 N개의 송신자가 있을 경우 K_{1N}로 표현한다.
$Root_j$	송신자j의 인증서 트리의 루트 값으로 m개의 키 체인이 있을 경우 R_{1m}로 표현한다.
μTP_{ji}	CDM_i와 유사한 의미를 갖는 기호로 송신자j의 i번째 키 체인의 파라메타를 의미한다.
K_{ji}	수신자j가 이용하는 키 체인의 $Interval_i$에서 사용될 메시지의 키 값을 의미한다.
$K_{i,m}$	해쉬 체인을 구성하기 위해 송신자가 랜덤하게 선택한 마지막 키 값을 의미한다.
$K_{i,0}$	만들어진 해쉬 체인의 가장 초기 키 값으로, 이후 분배된 키들을 인증하는데 사용된다.
T_s	센서 네트워크의 송·수신자 간에 시간 동기화를 위한 현재 시간을 의미한다.
T_i	전달된 초기 키 값이 이용되는 시작 시간을 의미한다.
T_{int}	키 체인의 동기화 구간 크기를 나타낸다.
d	메시지 키 노출 지연 시간으로, 분배된 K_{ji}는 d 구간만큼 지연된 이후에 사용된다.
$H(x)$	트리 생성시 사용되는 일 방향 해쉬 함수로 중앙서버, 송신자, 수신자가 사전에 공유하는 함수이다.
$F_i(x)$	키 생성시 사용되는 의사 난수 생성 함수로 정의한다[2].
CDM_i	Commitment Distribution Message의 약자로 각 N구간 마다 송신자가 주기적으로 브로드캐스트되는 값이다. 각 N구간에서 사용될 초기 키 값($K_{i,0}$) 및 통신에 필요한 파라메타 값을 포함한다.
$MAC_{K'_i}(x)$	Message Authentication Code의 약자. 입력 값 x에 대해 K'_i를 키로 대한 특정 비트열의 이미지 값을 생성한다.
ID_j	송신자j를 식별 값을 의미한다.
R'_j	송신자j이 소유한 키 체인의 초기 키 값들로부터 만들어진 인증서의 부분정보를 의미한다.
$ParaCert_{ji}$, U_{ji}	센서 네트워크에 다수의 송신자를 고려하는 경우, 송신자j의 i번째 키 체인에 대한 인증서 값을 연산하는데 필요한 파라메타들의 값을 의미한다. (단, μTPCT-based μTESLA의 경우, U_{ji}와 동일한 의미로 사용됨)
$ParaCert_j$	센서 네트워크에 다수의 송신자를 고려하는 경우, 수신자가 송신자j의 인증서를 연산하는데 필요한 파라메타들의 값을 의미한다.
$PCert$	$ParaCert$의 파라메타로 해쉬 값으로 구성되는 인자를 의미한다.
S_{ji}	송신자j의 i번째 키 체인에 대한 인증서 값을 의미한다.(단, [1]의 경우, μTP_{ji}와 동일)
S_j	송신자j에 대한 인증서 값으로 R'_j와 ID_j의 연접으로 구성된다.
XORC	제안하고자 하는 eXclusive-OR Chain의 약자이다.
L_i	i개의 리프 노드를 갖는 이진 트리의 높이를 의미한다.

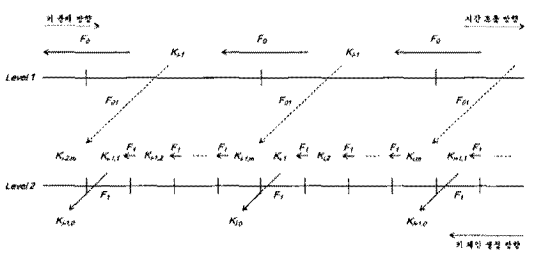
〔표 2〕 유사 용어 및 대표 용어 선택

대표 용어	유사 용어	설명
중앙서버 CS	Central Server, Base Station, BServer	센서 네트워크에서 게이트웨이 역할을 하는 중앙 서버. (단일 송신자를 고려한 네트워크의 경우 BS가 CS의 역할을 동시에 수행함)
송신자 BS	Base Station, Sender, Sink node, Solider, BNode	센서 네트워크에서 게이트웨이와 연결된 송신자.
수신자 SN	Sensor Node	센서 네트워크에서 주변 정보를 수집하는 수신자.
$\mu TP_{j,i}$	CDM_i, $\mu TP_{j,i}$	키 체인의 초기 키 값과 체인을 구성하는데 필요한 파라메타들로 구성된 메시지.
$ParaCert_{j,i}$	$U_{j,i}$, $ParaCert_{j,i}$	$\mu TP_{j,i}$와 송신자 인증서

(2) 다수의 송신자를 고려한 센서 네트워크 : 센서 네트워크에 하나의 중앙 서버가 위치하고 유선 혹은 무선으로 연결된 다수의 송신자가 있으며, 그 주변에 수신자가 위치한 네트워크 구조로 광대역의 센서 네트워크에 적합하다. 적용 가능한 μTESLA 프로토콜은 Tree-based μTESLA, μTPCT-based μTESLA, 제안하는 프로토콜이다.

2.1 단일 송신자를 고려한 센서 네트워크를 지원하는 μTESLA 프로토콜

TESLA(the Timed, Efficient, Streaming, Loss-tolerant, Authentication) 프로토콜[7]은 워크스



〔그림 1〕 2-Level μTESLA의 키 체인 구조

테이션 급 PC간의 효율적인 브로드캐스트 기법을 제공하는 프로토콜이다. 인증을 위해 송신자의 디지털 서명 기법을 이용하기 때문에, 자원이 제약적이고 무선을 이용하는 센서 네트워크에 적용하기에는 부적합하다[1].

2.1.1 Original μTESLA

Original μTESLA에서 이용되는 키 체인 구조는 [그림 1]에서 Level1에 해당한다. 전체 센서 네트워크의 생명 주기를 $Interval\,\Delta_0$ 길이의 n개의 구간으로 서로 다른 인증키(K_j)를 할당한다. 송신자는 n 구간마다 전달되는 메시지에 인증키 값(K_j)을 포함하여 전달하고, 수신자는 이를 수신하였다가 수식(2)의 검증식을 거쳐 인증되었을 경우 수신한 메시지를 저장한다. 수집한 정보가 있어 송신자에게 전달할 경우, 약속된 지연시간(delay) 이후에 저장해둔 인증키 값(K_j)과 수집 정보를 함께 전달한다. 송신자가 수신한 메시지에 이전에 전달했던 인증키 값(K_j)이 있을 경우에만 메시지를 저장하게 된다.

$$K_{i-1} = F_1(K_i), (0 \le i \le n) \quad (1)$$

Original μTESLA는 디지털 서명 대신에 송·수신자 간에 공유되는 초기 키(K_n)를 의사 난수 생성 함수($F_1(\ )$)의 입력 값으로 사용함으로써 일정 시간 구간($Interval_j$)마다 서로 다른 키(K_j)를 만들어내는 해쉬 체인을 구성한다.

$$K_i = F_1^{j-i}(K_j), (i < j) \quad (2)$$

수식(2)는 Original μTESLA의 메시지 검증식이다. 수신자는 현재 구간($Interval_i$)에 수신한 인증키(K_j)의 검증을 위해 의사 난수 생성 함수($F_1(\ )$)에 인증키(K_j)를 입력 값으로 하여 $(j-i)$회 반복 연산한다. 결과 값이 가장 최근($Interval_i$)에 이용하였던 인증키(K_i)값과 동일한지 확인한다. 만약, 값이 동일하다면 올바른 인증키로 믿고 인증키를 K_j로 대체한다.

2.1.2 Multi-Level μTESLA

Multi-Level μTESLA에서는 Original μTESLA를 좀 더 큰 규모의 센서 네트워크에 적용할 수 있도록 개선하였다. 그 특징으로 첫째, 동일한 μTESLA 파라메타의 경우 사전결정 방법을 이용하여 전달할 데이터

의 양을 줄였다. 둘째, 상위 레벨에 구간이 긴 키 체인을 두고, 구간이 짧은 키 체인을 하위 레벨에 두고 서로를 계층적으로 연결함으로써 인증키의 갱신 주기를 줄였다. 셋째, 메시지 손실과 DoS(Denial of Service) 공격의 피해를 줄이기 위해 메시지를 반복 전송하는 방법과 메시지 인증 지연을 줄이기 위해 다음 구간의 인증키를 현재 구간의 CDM_i에 추가하는 방법을 이용하였다.

[그림 1]은 2-Level μTESLA의 키 체인 구조를 나타낸다. Multi-Level μTESLA는 Level1의 n개의 긴 구간을 m개의 짧은 구간($Interval \triangle_1$)으로 나누어 메시지 인증지연 시간을 줄이고, 인증키의 갱신 주기를 줄였다.

$$CDM_{i-1} = \begin{Bmatrix} i \| K_{i,0} \| H(K_{i+1,0}) \| K'_{i-2} \\ \| MAC_{K'_{i-1}}(i \| K_{i,0} \| H(K_{i+1,0})) \end{Bmatrix} \quad (3)$$

$$CDM_i = \begin{Bmatrix} i \| K_{i+1,0} \| H(K_{i+2,0}) \| K'_{i-1} \\ \| MAC_{K'_i}(i \| K_{i+1,0} \| H(K_{i+2,0})) \end{Bmatrix} \quad (4)$$

수식(3)의 3번째 파라메타가 다음 구간($Interval_i$)에서 사용될 인증키($K_{i+1,0}$)의 이미지 값을 담고 있으므로, CDM_i을 수신하고 $H(K_{i+1,0})$을 연산하여 값이 동일하면 CDM_{i-1}을 인증하게 된다. 이후 CDM_{i-1}의 무결성 검사를 위해서 CDM_i의 마지막 파라메타인 K'_{i-1}를 이용한다. i번째 구간의 하위 레벨 키 체인의 마지막 키($K_{i,n}$)의 분실시 복구가 가능하도록 상위 레벨 키 체인과 하위 레벨 키 체인을 $F_{01}(\ )$로 서로 연결하였다.

2.2 다수의 송신자를 고려한 센서 네트워크를 지원하는 μTESLA 프로토콜

센서 네트워크에 송신자가 하나만 위치한 경우, 다수의 수신자들로부터 전달되는 데이터로 인하여 병목현상이 발생하며 송신자 인근에 배치된 중계 노드들의 에너지 소모가 많아지게 된다. 결과적으로 센서 네트워크의 수명을 짧아지게 만든다. Tree-based μTESLA는 하나의 센서 네트워크에 다수의 송신자를 고려함으로써 병목현상을 줄일 수 있으며 광범위한 센서 네트워크에 적용 가능하다.

앞으로 설명할 두 프로토콜 및 제안하는 프로토콜은 [그림 5]의 키 체인 구조와 [그림 2]의 상위 레벨 트리($Tree_R$)를 기반으로 한다. 본 절에서는 각 프로토콜이 갖는 장점과 단점에 대해 기술하고, 키 체인 구조 및 트리를 이용한 인증서 구조에 대한 자세한 설명은 3.3절에서 살펴보도록 한다.

2.2.1 Tree-based μTESLA

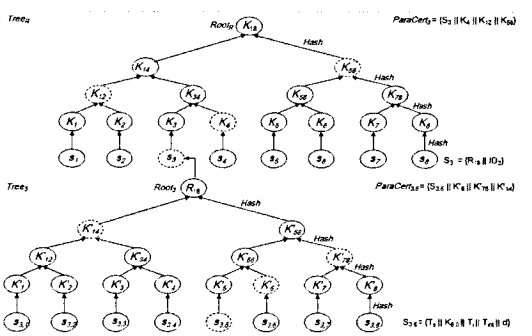
Tree-based μTESLA는 센서 네트워크에 다수의 송신자를 고려하기 위하여 각 송신자j에 대한 인증서(S_j)와 i번째 구간에 대한 송신자j의 키 체인에 대한 인증서($S_{j,i}$)를 생성하고, 이를 연산할 수 있는 인증서 파라메타($ParaCert_j, ParaCert_{j,i}$)를 수신자에게 전달한다. 트리의 각 노드는 이웃한 하위 레벨 트리 2개를 연접하고 해쉬 함수를 적용함으로써 생성된다. [그림 2]는 송신자3을 위한 5번째 키 체인의 $\mu TP_{3,5}$의 인증서($S_{3,5}$)를 생성했을 경우를 나타낸다.

Tree-based μTESLA는 짧은 구간($Interval \triangle_1$)의 키 체인을 이용하여 메시지 인증지연 시간을 줄이고, 사전 분배된 상위 트리의 루트 값($Root_R$)을 이용하여 송신자로부터 인증서 파라메타($ParaCert_j$, $ParaCert_{j,i}$)를 수신할 경우, 즉시인증이 가능하다. 또한, 송신자를 위해 상위 레벨의 트리 기반 인증서 구조를 이용함으로써 센서 네트워크에 다수의 송신자를 설치할 수 있는 장점이 있다.

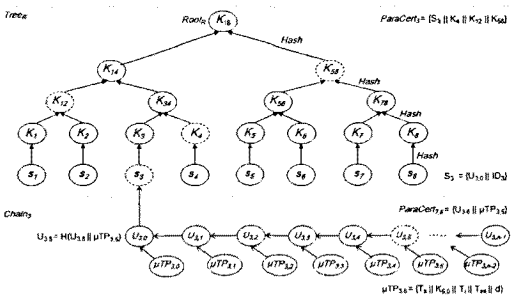
하지만, 송신자의 수나 소유하는 키 체인의 수가 많은 경우, 트리 높이가 증가하기 때문에 전달해야 하는 $ParaCert_{j,i}$의 데이터 량이 증가하게 된다. 따라서 송·수신자 사이의 통신량 및 연산량이 증가하게 되는 단점을 가지고 있다.

2.2.2 μTPCT-based μTESLA

μTPCT-based μTESLA는 트리 높이 증가에 따른 연산량이 증가하는 문제점을 해결하기 위해 제안되



[그림 2] Tree-based μTESLA의 인증서 구조



〔그림 3〕 μTPCT-based μTESLA 인증서 구조

었다. 이 프로토콜은 현재 ISO/IEC JTC1/SC6 WG7에서 진행 중인 USN을 위한 보안 프레임워크(X.usnsec-1)에서 브로드캐스트 인증 기법으로 채택된 프로토콜이다[7]. 〔그림 3〕은 송신자3이 소유한 6번째 키 체인의 $\mu TP_{3,5}$를 위한 인증서($ParaCert_{3,5}$)를 생성했을 경우를 나타낸다.

Tree-based μTESLA의 하위 트리의 구조를 μTPC(μTESLA Parameter Chain)라 불리는 해쉬 체인 구조로 변경하여, 수신자 측에서의 메시지 인증시 연산에 필요한 인증서 파라메타($ParaCert_{j,i}$)의 데이터양을 줄임으로써 통신량과 수신자의 연산량을 고정적으로 줄이는 효과를 가져왔다.

반면, 해쉬 체인 구조는 긴 2구간($2\times\Delta_0$) 이상의 통신 두절시 더 이상 메시지 인증이 불가능한 특성을 가지고 있다. 만약, 수신자가 초기 인증서 값($U_{3,0}$)만을 가지고 있는 상태에서 2구간 이상 데이터를 수신하지 못한 경우, 수신자가 받을 수 있는 파라메타들은 인증서 값 $U_{3,3}$와 μTESLA 파라메타 $\mu TP_{3,2}$가 된다. 이때, 수신한 파라메타들을 이용해 $U_{3,2}$값을 계산할 수 있지만, $\mu TP_{3,1}$을 도출할 수 없으므로 수신자3이 전달하는 이후의 메시지들을 인증할 수 없게 된다.

III. 제안 프로토콜

이번 장에서는 위에서 언급한 목적을 달성하기 위해 이용되는 연산적인 특성과 제안한 XORC 인증서 구조 및 생성 과정에 대해 알아보고자 한다. 본 논문에서 제안하는 기법은 μTPCT-based μTESLA의 장기간 통신 두절시 메시지 인증이 불가능하다는 문제점을 해결하기 위해 해쉬 함수를 이용한 XORC 구조를 제안하며, XORC 구조를 이용한 목적은 다음과 같다.

3.1 브로드캐스트 인증 방식의 목적 및 요구사항

① 해쉬 체인의 순서성이 장기간 통신 두절시 메시지 인증불가 문제를 해결한다.
② 장기간 통신 두절 이후, 언제라도 인증서 파라메타를 수신하면 인증이 가능해야 한다.
③ 키 체인의 수와 관계없이 적은양의 고정된 연산만으로 송신자와 메시지를 인증할 수 있어야 한다.
④ 만들어진 키와 인증서의 값은 각 구간마다 서로 상이해야 하고, 공개되지 않은 키에 대한 정보를 유추할 수 없어야 한다.

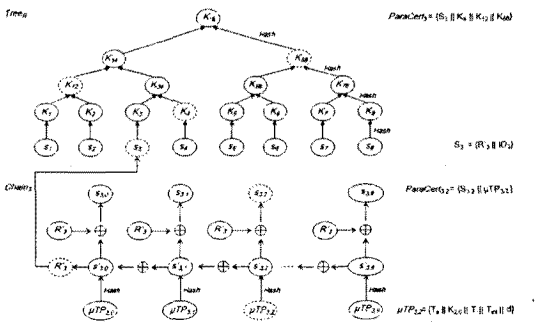
3.2 제안하는 인증서 구조에서 이용하는 연산자 측면의 특성 분석

트리 기반 인증서 구조는 트리 높이가 높아질 경우 통신량과 연산량이 증가하는 문제점을 가지고 있다. 해쉬 체인 기반 인증서 구조는 트리 기반 인증서 구조의 문제를 해결했지만, 장기간의 통신 두절이 발생할 경우에는 인증이 불가능하다는 문제점이 발생한다. 제안하고자 하는 인증서 구조는 이러한 문제를 해결하면서 트리 기반 인증서 구조의 장점과 체인 기반 인증서 구조의 장점을 고루 갖춘 XORC 생성을 위해 XOR 연산과 해쉬 연산을 이용하였다, 각 연산은 〔표 3〕에 나열된 특성을 가진다.

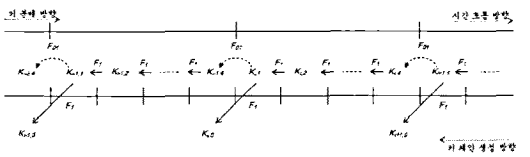
해쉬 체인 기반 인증서 구조의 경우, 장기간 통신 두절시 메시지 인증이 불가능하게 되는데 그 이유는 해쉬 함수의 순서성 때문이다. (2.2.2 참조). 제안하는 기법에서는 해쉬 연산의 순서성을 없애기 위해 XORC을 이용한다. 위에서 살펴본 XOR 연산의 특성①과 특성②는 장점으로 작용하지만, 특성③과 특성④는 보안상 취약한 요소로 작용할 수 있다. 3.1절의 목적 달성과 특성③과 특성④를 보완하기 위해서 XORC 생성 이전에 1회의 해쉬 연산을 추가하였다. XORC에 대한 안전성 분석은 4.4절에서 다룬다.

〔표 3〕 XOR 체인 구조가 이용하는 연산의 특성

HASH 연산의 특성[6]	XOR 연산의 특성
①일방향성 ②약한 일방향성 ③강한 일방향성 ④충돌 회피성 ⑤순서성	①가벼운 연산량(해쉬 연산 대비) ②순서성 없음 ③짝수 번 반복된 입력 값 상쇄 ④마지막 값 유추 가능성



〔그림 4〕 XOR-based μTESLA 인증서 구조



〔그림 5〕 XOR-based μTESLA의 키 체인 구조

3.3 제안하는 인증서 구조 분석

본 절에서는 제안하고자 하는 인증서 구조가 이용하는 키 체인과 상위 인증서 트리 및 하위 XORC 기반 인증서 구조에 대해 알아보고자 한다. 설명은 키 체인 생성을 시작으로 하여 최종 값인 $Root_R$를 생성하는 순서로 진행한다. 〔그림 4〕의 상위 트리는 송신자를 나타내는 $N=8$, $Interval\Delta_0$ 키 체인의 수를 나타내는 $n=10$, 한 개의 $Interval\Delta_0$ 키 체인이 갖는 $Interval\Delta_1$ 키 체인의 수를 나타내는 $m=5$인 경우를 나타낸다. 〔그림 5〕의 키 체인은 송신자를 지원하는 μTESLA 프로토콜에서 공통적으로 이용하고 있는 구조이고, 〔그림 4〕의 하위 체인이 XORC 기반의 인증서 구조를 나타낸 것이다. 이해를 돕기 위해 〔그림 4〕에서 인증서에 포함되는 파라메타들을 회색으로 표현하도록 한다. (단, 송신자를 나타내는 $j=3$, 긴 체인의 인덱스를 나타내는 $i=2$인 경우를 나타낸다.)

(1) 키 체인 구조의 생성 과정

중앙 서버가 키 체인과 관련된 파라메타를 사전에 결정하고, 관련 정보를 송신자에게 전달한다.

단계①,3번째 송신자의 2번째 키 체인 생성을 위해서 중앙 서버가 랜덤하게 선택한 마지막 키 값($K_{2,4}$)을 송신자에 전달한다.

단계②, $K_{i,t-1}=F_1(K_{i,t}),(0\le t\le m-1)$ (5)

수신된 마지막 키 값을 $F_1(\ )$의 입력 값으로 하여 4회 반복 연산하여 1개의 짧은 키 체인을 만들고, $K_{2,0}$을 초기 키 값으로 선택한다.

단계③, $K_{t-1,m-1}=F_{01}(K_{t,1}),(0\le t\le n)$ (6)

이웃한 짧은 키 체인을 서로 연결하기 위해, 수식 (5)에서 $K_{2,1}$값을 $F_{01}(\ )$의 입력 값으로 하여 다음 키 체인의 마지막 키 값($K_{1,4}$)을 만든다.

단계④,단계②와 단계③을 반복하여 송신자3을 위한 한 개의 키 체인을 생성한다.(〔그림 5〕)

단계⑤, $\mu TP_{j,i}=\{T_s\|K_{i,0}\|T_i\|T_{int}\|d\}$ (7)

센서 네트워크 내의 시간 동기화를 위해 다음의 파라메타들을 각 체인의 초기 키 값과 함께 전달한다. (여기서, 각 짧은 키 체인의 초기 키 값($K_{i,0}$)을 제외한 값들이 동일한 경우, 중앙 서버는 이를 사전에 결정할 수 있다.)

단계⑥, 공개되는 키의 방향은 키 체인 생성 방향과 반대 방향이다.

(2) 하위 XORC 기반의 인증서 생성 과정

XORC은 중앙 서버에서 생성하여 관련 정보를 각 송신자에게 전달한다.

단계①, $R_j=\begin{Bmatrix}H(\mu TP_{j,0})\oplus H(\mu TP_{j,1})\oplus\cdots\\ \oplus H(\mu TP_{j,n-2})\oplus H(\mu TP_{j,n-1})\end{Bmatrix}$ (8)

송신자3의 인증서($S_3=R_3\|ID_3$)의 부분 정보로, 송신자3이 소유한 키 체인의 μTESLA 파라메타 값들에 의해 생성된다.

단계②, $S_{j,i}=\{R_j\oplus H(\mu TP_{j,i})\}$ (9)

송신자3의 인증서 부분 정보(R_3)를 이용하여, 송신자3의 두 번째 짧은 체인의 μTESLA 파라메타 값에 대한 인증서 값을 생성한다.

단계③, $ParaCert_{j,i}=\{S_{j,i}\|\mu TP_{j,i}\}$ (10)

송신자3은 수신자에게 자신이 소유한 $K_{3,0}$ 정보를 알리기 위해, μTESLA 파라메타와 인증서로 구성된 $ParaCert_{3,2}$를 주기적으로 브로드캐스트 한다.

(3) 상위 트리 기반 인증서 생성 과정

상위 트리 기반 인증서 구조는 중앙 서버에서 생성하여 관련 정보를 각 송신자에게 사전분배 한다.

단계①, $S_j=\{R_j\|ID_j\}$ (11)

송신자3의 인증서(S_3)는 키 체인의 초기 키 값들의 정보(R_3)와 송신자3의 식별자(ID_3) 로 구성된다.

$$\text{단계②}, K_j = H(S_j) \quad (12)$$

각 송신자의 인증서(S_3) 정보가 노출되지 않도록 해쉬 연산을 한다.

$$\text{단계③}, H(K_1 \| K_2), H(K_3 \| K_4), \ldots, H(K_{N-1} \| K_N) \quad (13)$$

각 트리 높이 별로 이웃한 해쉬 연산 결과를 연접하여 해쉬 연산을 한다. 이와 같은 연산을 통해 노드의 수가 1/2씩 줄게 되므로, L_N회를 반복하면 $Root_R$값을 생성할 수 있다. $Root_R$ 값은 센서 네트워크에 참여하고 있는 송·수신자에게 사전 분배된다.

$$\text{단계④}, ParaCert_j = \{S_j \| K_{12} \| K_4 \| K_{58}\} \quad (14)$$

중앙 서버가 송신자3에게 전달하려는 정보가 있을 경우, 송신자3의 인증서 파라메타를 함께 전달한다. ($N=8$, $j=3$으로 가정)

(4) 송신자의 메시지 인증 과정

중앙 서버로부터 송신자가 메시지를 수신한 경우. ($N=8$, $j=3$으로 가정)

$$\text{단계①}, DATA \| ParaCert_3 = \{S_3 \| K_4 \| K_{12} \| K_{58}\} \quad (15)$$

송신자3은 중앙서버로부터 데이터와 자신의 인증서 파라메타를 수신한다.

$$\text{단계②}, \begin{matrix} Root'_R = \begin{Bmatrix} H(H(K_{12} \| \\ H(H(S_3) \| K_4)) \| K_{58}) \end{Bmatrix} \\ Root_R \stackrel{?}{=} Root'_R \end{matrix} \quad (16)$$

송신자3은 사전분배 받은 $Root_R$값을 이용하여, 인증서 파라메타의 연산결과($Root'_R$)와 동일한지 검사한다. 일치할 경우, 수신했던 데이터를 저장한다.

(5) 수신자의 메시지 인증 과정

송신자로부터 수신자가 메시지를 수신한 경우. ($N=8$, $j=3$으로 가정)

$$\text{단계①}, DATA \| ParaCert_3 \| ParaCert_{3,2} \quad (17)$$

$$\text{단계②}, \begin{matrix} Root'_R = \begin{Bmatrix} H(H(K_{12} \| \\ H(H(S_3) \| K_4)) \| K_{58}) \end{Bmatrix} \\ Root_R \stackrel{?}{=} Root'_R \end{matrix} \quad (18)$$

$$\text{단계③}, \begin{matrix} R'_3 = H(\mu TP_{3,2}) \oplus S_{3,2} \\ S_3 \stackrel{?}{=} R'_3 \| ID_3 \end{matrix} \quad (19)$$

송신자3이 데이터를 보내는 경우, $ParaCert_3$을 이용해서 올바른 송신자인지를 확인하고, 수식(19)에서 계산한 R'_3을 이용하여 송신자3의 ID_3와 연접한 값이 송신자3의 인증서 값S_3와 동일한지 검사한다. 일치할 경우, 수신했던 데이터를 저장한다.

IV. 성능 분석

이번 절에서는 지금까지 알아본 브로드캐스트 인증 기법들의 효율성을 확인하기 위해서 각 기법에서의 저장, 통신, 연산 오버헤드 및 메시지 복원력과 제안한 인증서 구조의 안전성에 대해 분석을 하고자 한다. 파라메타의 크기는 기법①, ②, ③이 모두 Tree-based μTESLA[1]를 송신자 인증서 생성을 위해 사용하였으므로 [1]에서 가정한 패킷 페이로드 크기인 29*bytes*로 가정하고, $|HASH|$, $|PCert|$, $|S|$, $|\mu TP|$ 모두 8*bytes*로 가정한다. $|\mu TP|$의 파라메타$\{T_s \| T_i \| T_{int} \| d\}$가 중앙 서버에서 사전에 정의될 경우, $|\mu TP| = \{K_{i,0}\}$가 되므로 8*bytes*의 크기를 가질 수 있다.

4.1 효율성 분석

비교에 앞서 체인 전체를 생성해 내는 중앙서버, 송신자의 능력은 PC급이므로 저장과 연산 능력에 제한이 없는 것으로 고려한다. 다수의 송신자를 고려하고 있는 3가지 기법이 동일한 키 체인 구조와 트리 기반 인증서 구조를 이용하고 있으므로 CS에서 인증서를 생성하는데 드는 연산량을 비교하고, BS와 SN에서는 저장량, 통신량, 연산량 등을 비교하도록 한다. 분석 대상을 원문자를 할당하여 각 기법을 구분한다. ① Tree-based μTESLA, ② μTPCT-based μTESLA, ③ XOR Chain-based μTESLA. 센서 네트워크에 N명의 송신자는 각각 n개의 긴 키 체인을 소유하고 있다고 가정한다.

(1) 중앙 서버(CS)의 연산량 분석

기법①, ②, ③ 모두 다중 송신자를 지원하기 위해 트리 기반의 인증서 구조를 이용하고 있다. N명의 송신자로 구성된 상위 레벨 트리 인증서 구조 생성을 위해 $(2^{L_N}-1)$회의 해쉬 연산이 필요하다. 기법①의 경우, n개의 긴 키 체인으로 구성된 하위 레벨 트리 인증서 구조 생성을 위해 $(2^{L_n}-1)$회의 해쉬 연산이 필요하다. 기법②의 경우, μTPC을 이용하기 때문에 n회의

해쉬 연산만 필요로 한다. 기법③의 경우, XORC을 이용하기 때문에 n회의 해쉬 연산과 XORC 생성을 위한 $(2n-1)$의 XOR 연산이 추가로 요구된다. 효율성을 비교하자면 기법② 〉 기법③ 〉 기법① 순이다.

(2) 송신자(BS)의 저장 오버헤드 분석

기법①, ②, ③ 모두 상위 레벨 트리 인증서 구조를 사용하기 때문에, 사전 분배된 상위 레벨의 루트 값 ($Root_R$)과 키 체인 정보를 담고 있는 하위 레벨 트리 (기법②의 경우 μTPC, 기법③의 경우 XORC)에 대한 정보를 저장하고 있어야 한다. 따라서 기법①의 경우 $(2^{L_n}-1)$개의 $ParaCert$, 기법②와 기법③의 경우 n개의 $ParaCert$을 추가로 저장해야 한다. 효율성을 비교할 때 기법② = 기법③ 〉 기법① 순이다.

(3) 중앙 서버와 송신자 간의 통신량 분석

각 송신자는 중앙 서버로부터 데이터나 쿼리를 받을 경우, 자신이 소유한 키 체인 정보로 만들어진 송신자의 인증서를 받게 된다. 기법①, ②, ③ 모두 상위 레벨 트리 인증서 구조를 사용하기 때문에, (L_N-1)개의 $PCert$와 S_j을 수신해야 한다.

(4) 송신자의 연산량 분석

송신자는 중앙 서버로부터 수신한 메시지를 인증하기 위해 $(1+log_2N)$회의 해쉬 연산을 수행해야 한다. 기법①, ②, ③ 모두 상위 레벨 트리 인증서 구조를 사용하기 때문에, 모두 동일한 연산량을 갖는다고 볼 수 있다.

(5) 수신자의 저장량 분석

기법①, ②, ③ 모두 상위 레벨 트리 인증서 구조를 사용하기 때문에, $Root_R$을 사전에 분배 받는다. 다만, 기법②의 경우, 하위 레벨 체인으로 μTPC을 이용하기 때문에 μTPC의 초기 값인 $U_{j,0}$값을 추가로 저장하고 있어야 한다.

(6) 송신자와 수신자 간의 통신량 분석

송신자가 수신자에게 정보를 전달하려면, 송신자 자신을 증명할 수 있는 인증서 값(S_j)을 포함하고 있는 $ParaCert_j$을 전달해야 하고, 자신이 소유한 키 체인에 대한 인증서 값($S_{j,i}$)와 μTESLA 파라메타 값 ($\mu TP_{i,m}$)을 포함하고 있는 $ParaCert_{j,i}$을 주기적으로 전달해야 한다. 모든 기법에서 $ParaCert_j$의 데이터 크기는 $(L_N-1)|PCert|+|S|$로 표현 가능하다. $ParaCert_{j,i}$의 데이터 크기는 기법①의 경우에는 $(L_n-1)|PCert|+|\mu TP|$이고, 기법①와 ②의 경우에는 $|PCert|+|\mu TP|$이다. 효율성을 비교해보면 기법② = 기법③ 〉 기법① 순이다.

(7) 수신자 측면에서의 연산량 분석

수신자는 송신자로부터 송신자를 인증할 수 있는 정보를 담고 있는 $ParaCert_j$와 키 체인 정보를 담고 있는 $ParaCert_{j,i}$을 수신 받는다. 기법①, ②, ③ 모두 상위 레벨 트리 인증서 구조를 사용하기 때문에, 송신자 인증을 위해서 $(1+log_2N)$회의 해쉬 연산이 공통적으로 필요하고, 기법①의 경우 하위 트리의 인증을 위해 $(1+log_2n)$회의 해쉬 연산이 추가적으로 필요하다. 반면, 기법②의 경우에는 1회의 해쉬 연산이 추가되고, 기법③의 경우에는 1회의 해쉬 연산과 1회의 XOR 연산이 추가적으로 필요하다. 효율성을 비교해보면 기

[표 4] 기존 방법과의 효율성 분석

(단위: $|HASH|=|PCert|=|S|=8bytes, |\mu TP|\geq 8bytes$)

		Tree-based μTESLA(1)	μTPCT-based μTESLA(2)	XORC-based μTESLA(3)
CS	인증서 연산량	$(2^{L_N}-1)h+(2^{L_n}-1)h$	$(2^{L_N}-1)h+nh$	$(2^{L_N}-1)h+nh$ $+(2n-1)x$
BS	저장량	$\|HASH\|+(2^{L_n}-1)\|PCert\|$	$\|HASH\|+n\|PCert\|$	$\|HASH\|+n\|PCert\|$
	CS-BS 통신량	$(L_N-1)\|PCert\|+\|S\|$	$(L_N-1)\|PCert\|+\|S\|$	$(L_N-1)\|PCert\|+\|S\|$
	인증 연산량	$(1+log_2N)h$	$(1+log_2N)h$	$(1+log_2N)h$
SN	저장량	$HASH$	$2HASH$	$HASH$
	BS-CS 통신량	$(L_N-1)\|PCert\|+\|S\|$ $+(L_n-1)\|PCert\|+\|\mu TP\|$	$(L_N-1)\|PCert\|+\|S\|$ $+\|PCert\|+\|\mu TP\|$	$(L_N-1)\|PCert\|+\|S\|$ $+\|PCert\|+\|\mu TP\|$
	인증 연산량	$((log_2Nn)+2)h$	$((log_2n)+2)h$	$((log_2n)+2)h+x$

법② > 기법③ > 기법① 순이다.

〔표 4〕는 앞서 분석한 내용을 도표로 정리한 것이다. 제안하고자 기법③은 1회의 XOR 연산만을 추가함으로써 기법②의 방식처럼 수신자 측면에서 효율적인 메시지 인증 연산량을 갖는다.

4.2 메시지 복원력 분석

기법①, ③에서는 모든 송·수신자가 신뢰하는 $Root_R$ 값을 전달하기 위해 사전분배 기법을 이용하였다〔3〕. 따라서 네트워크 상태가 장기간 끊어지더라도, 메시지 인증을 위해 $ParaCert_j$, $ParaCert_{j,i}$만 수신한다면 언제라도 메시지 인증이 가능하다.

기법②에서는 모든 송·수신자가 신뢰하는 $Root_R$값을 사전분배 받음으로써 〔표 5〕와 같이 적은 양의 데이터만을 수신하여 즉시 인증이 가능하다. 하지만, 순서성을 갖는 해쉬 함수 체인(μTPC)를 이용하기 때문에 연속된 입력 값을 알지 못하는 경우 출력 값을 알 수 없다. 〔그림 3〕에서 나타난 것처럼 추가 인증을 위해서는 최근에 인증된 $U_{j,i}$값을 저장하고 있어야만 한다. 그러므로 2.2.2에서 언급한 바와 같이 2구간($2\times\Delta_0$)간 이상 $ParaCert_{j,i}$을 수신하지 못하면 이후 수신한 메시지를 모두 인증할 수 없는 문제점이 발생하게 된다.

〔표 5〕는 각 기법에서 최대 복구 가능한 분실 기간을 표로 정리한 것이다. 기법①도 분실 기간에 상관없이 메시지 인증이 가능하지만, 키 체인의 개수에 따라 통신량이 증가하는 단점이 있다. 제안하는 기법③는 적은 양의 데이터 통신만으로 언제든 메시지 인증이 가능함을 확인할 수 있다.

4.3 인증서 구조의 안전성 분석

지금까지 알아본 기법①, ②, ③에서 송신자 인증을 위해 동일한 상위 트리 기반 인증서 구조를 이용하고, 트리 기반 인증서 구조는 해쉬 함수의 안전성에 기반을 둔다. 마찬가지로 각 기법이 이용하는 하위 레벨의 인증서 구조 또한 해쉬 함수의 안전성에 기반을 두고 있다. 다만, 기법③의 경우 XOR 체인을 이용함에 있어, 3.2절에서 언급한 문제점이 발생할 수 있다. 이를 보완하는 방법으로 해쉬 함수를 이용하였다. 이번 절에서는 각 기법이 안전성에 기반을 두고 있는 해쉬 함수의 안전성에 대해 알아보고, XOR 체인의 문제점을 보완했음을 확인한다. 해쉬 함수의 특성을 자세히 알아보면 다음과 같다〔6〕.

〔표 5〕 허용 가능한 최대 메시지 분실 기간 비교

구분	최대 구간	인증 기준	데이터 통신량
Tree	-	$Root_R$	$(L_n-1)PCert+1\mu TP$
μTPC	1	$U_{j,m-1}$	$1PCert+1\mu TP$
XORC	-	$Root_R$	$1PCert+1\mu TP$

*최대 구간 : 인증이 가능한 최대 메시지 손실 구간 기간
*인증기준 : 수신자가 m번째 메시지 인증을 위해 신뢰하고 있는 신뢰하는 값.
*데이터 통신량 : 메시지 인증을 위해 전달되어야 하는 최소한의 데이터 통신량

특성① 약한 일방향성 : 해쉬 함수 $H(x)=y$에서 y가 주어질 때, 연관된 입력 x를 계산할 수 없어야 한다.

특성② 강한 일방향성 : 해쉬 함수 $H(x)=y$에서 $H(x')=y$가 되는 입력 $x'\neq x$를 찾을 수 없어야 한다.

특성③ 충돌 회피성 : 해쉬 함수 $H(x)=y$에서 $H(x)=H(x')$ 인 값이 없어야 한다.

하지만, 해쉬 함수는 m비트의 메시지를 n비트의 해쉬 값으로 사상시킴으로서 같은 결과 값을 가질 수 있는 확률이 존재하게 된다. 생일공격(Birthday Attack)〔6〕에서 충돌쌍이 발생할 수 있는 확률은 $2^{n/2}$이다. 아래 식은 각 기법의 브로드캐스트 메시지 인증 과정을 나타내고 있다. (단, $N=8$, $n=10$, $m=5$, $j=3$, $i=2$일 경우 아래와 같이 표현할 수 있다.)

$$\text{기법①}: H'_j = H(H(K'_{14}\|H(H(S_{j,5})\|K'_6)\|K'_{78}) \quad (20)$$

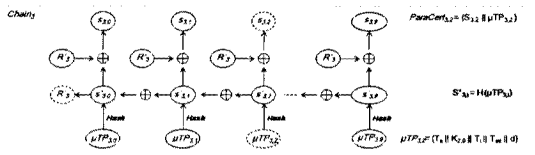
$$\text{기법②}: U_{i-1} = H(U_i\|\mu TP_{i-1}) \quad (21)$$

$$\text{기법③}: H'_j = H(\mu TP_{j,i})\oplus S_{j,i} \quad (22)$$

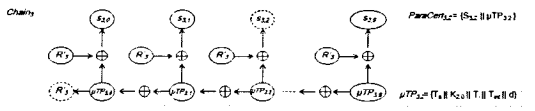
기법①에서는 $2n$비트의 입력으로 n비트의 출력이 발생하고, 기법②에서는 $2n+\alpha$비트 입력으로 n비트의 출력이 발생하고, 기법③에서는 $2n$비트의 입력으로 n비트의 출력이 발생한다. 각 기법 모두에서 n비트의 출력이 발생하는 해쉬 연산을 이용하고 있으므로 모든 기법에서 해쉬 함수를 공격하여 인증서를 위조할 확률은 $2^{n/2}$이다.

4.4 XOR Chain의 안전성 검증

제안하는 XOR 체인 구조에서는 $\mu TP_{j,i}$, $(1\le j\le N, 0\le i\le n-1)$일 때, 사용자$j$를 위해 $\mu TP_{j,i}$ 모두를 XOR 연산한 결과를 기반으로 $\mu TP_{j,i}$의 인증



〔그림 6〕 HASH 연산이 포함된 XORC 생성 과정



〔그림 7〕 HASH 연산이 포함되지 않은 XORC 생성 과정

서 $S_{j,i}$를 생성하게 된다. 이 과정에서 XOR만을 이용하게 될 경우, 3.2절에서 알아본 XOR 연산의 특성 중 ③짝수 번 반복된 입력 값 상쇄에 의해 공개되지 않은 $\mu TP_{j,i}$ 값이 노출될 수 있는 취약점이 발생한다. 이런 문제점을 해결하기 위해 XOR 체인 연산에 해쉬 연산을 추가하였다. 이번 절에서는 해쉬 연산을 추가한 XOR 체인 구조가 공개되지 않은 $\mu TP_{j,i}$의 노출 문제를 해결할 수 있음을 증명하고자 한다.

〔그림 4〕에서 해쉬 연산이 포함되지 않은 XOR 체인의 생성 과정을 수식 및 그림으로 표현하면 다음과 같다.

증명: 만약, $n=5$, 전체 구간은 0~4, 현재 구간을 2로 가정할 경우, 구간 3이 되는 순간 아직 공개되지 않은 구간 4의 $\mu TP_{3,4}$값을 유추할 수 있다. $\mu TP_{3,4} = R_3 \oplus (\mu TP_{3,0} \oplus \mu TP_{3,1} \oplus \mu TP_{3,2} \oplus \mu TP_{3,3})$이므로 공개되지 않은 $\mu TP_{3,4}$값을 사전에 알 수 있다.

반면, XOR 체인을 생성하기 이전에 해쉬 연산을 적용한다면 $\mu TP_{3,4}$값이 노출되지는 않는다.

$$\text{단계①}, s'_{j,i} = H(\mu TP_{j,i}), (0 \le i \le n-1) \quad (27)$$

$$\text{단계②}, R'_3 = \{s'_{j,0} \oplus s'_{j,1} \oplus \cdots \oplus s'_{j,n-2} \oplus s'_{j,n-1}\} \quad (28)$$

$$\text{단계③}, S_{3,2} = \{R'_3 \oplus s'_{3,2}\}, (j=3, i=2\text{인 경우}) \quad (29)$$

$$\text{단계④}, ParaCert_{3,2} = \{S_{3,2} \| \mu TP_{3,2}\} \quad (30)$$

$$\text{단계⑤}, R'_3 = \mu TP_{3,2} \oplus S_{3,2}, S_3 \xrightarrow{+} R'_3 \| ID_3 \quad (31)$$

$$\text{증명} : H(\mu TP_{3,4}) = R'_3 \oplus \begin{Bmatrix} H(\mu TP_{3,0}) \oplus H(\mu TP_{3,1}) \\ \oplus H(\mu TP_{3,2}) \oplus H(\mu TP_{3,3}) \end{Bmatrix}$$

일 때, $H(\mu TP_{3,4})$값을 얻을 수 있다. 하지만 $\mu TP_{3,4} = \{T_s \| K_{4,0} \| T_4 \| T_{int} \| d\}$ 값이 유출되지 않으므로, 공개되지 않은 μTESLA 파라메타를 보호할 수 있다. XORC에 사용된 $H(\,)$의 안전성은 4.3절에서 알아본 특성①에 의해 보장 받는다.

V. 결 론

센서 네트워크에서 수신자의 자원 제약적인 측면은 센서 네트워크의 전체적인 수명에 크게 영향을 미치기 때문에 송신자와 수신자 간에 전달되는 메시지의 양을 최대한 줄여야 하며, 전달된 메시지를 인증하기 위해 소모되는 연산량 또한 최소화시켜야 한다. 더욱이 대부분의 센서 네트워크에서 통신으로 이용하는 브로드캐스트 방식은 수신을 보장할 수 없으므로 반복적인 전송을 기본으로 하고 있다. 손실률이 높은 무선 네트워크라도 수신한 메시지에 대한 인증은 보장되어야 한다. 새롭게 제안한 XOR 체인 기반 인증서 체인 구조는 수신자 측면에서 1회의 XOR 연산이 증가하기는 하지만 고효율의 인증 연산을 지원하고, 장기간의 통신 두절이 발생하더라도 언제든 즉시인증이 가능하다. 향후 연구를 통해 2-레벨의 XOR 체인 기반 인증서 구조를 이용하면 상위 레벨 트리 인증서 구조가 가지는 저장, 연산, 통신 오버헤드를 획기적으로 줄일 수 있을 것으로 예상된다.

참 고 문 헌

〔1〕 A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J.D. Tygar, "SPINS: security protocols for sensor networks," Proceedings of the seventh annual international conference on Mobile computing and networking, pp. 189-199, July 2001.

〔2〕 D. Liu and P. Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," ACM Transactions on Embedded Computing Systems (TECS), vol. 3, no. 4, pp. 800-836, Nov. 2004.

〔3〕 D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical broadcast authentication in sensor networks," the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 118-132, July 2005.

〔4〕 Z. Du, K. Wang, and L. Zhou, "Efficient broadcast authentication in wireless

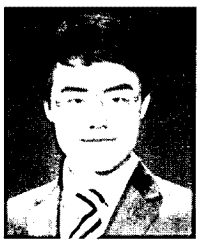
sensor networks," Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference, pp. 187-192, Dec. 2008.

[5] ITU-T, "Proposal for the 4th revised text on ITU-T X.usnsec-1|ISO CD 29180: Security framework for ubiquitous sensor network," http://www.itu.int/md/T09-SG 17-C-0125/en, Sep. 2009.

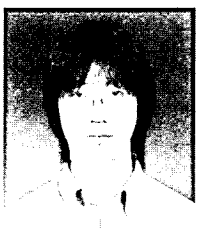
[6] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, pp. 321-383, Oct. 1996.

[7] A. Perrig, R. Canetti, J.D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," In IEEE Symposium on Security and Privacy, pp. 56-74, May 2000.

〈著者紹介〉



여 돈 구 (Don-Gu Yeo) 학생회원
2009년 2월: 순천향대학교 정보보호학과 졸업
2009년 3월: 순천향대학교 정보보호학과 석사과정
〈관심분야〉 정보보호, USN 보안, 클라우드 컴퓨팅 보안, IPTV 보안, 역추적



장 재 훈 (Jae-Hoon Jang) 학생회원
2009년 2월: 순천향대학교 정보보호학과 졸업
2009년 3월: 순천향대학교 정보보호학과 석사과정
〈관심분야〉 역추적, IPTV 보안, USN 보안

최 현 우 (Hyun-Woo Choi) 학생회원
2009년 2월: 순천향대학교 정보보호학과 졸업
2009년 3월: 순천향대학교 정보보호학과 석사과정
〈관심분야〉 IPTV 보안, 스마트그리드 보안, USN 보안, 역추적

염 흥 열 (Heung-Youl Youm) 종신회원
1981년 2월: 한양대학교 전자공학과 졸업(학사)
1983년 2월: 한양대학교 대학원 전자공학과 졸업(석사)
1990년 2월: 한양대학교 대학원 전자공학과 졸업(박사)
1982년 12월~1990년 9월: 한국전자통신연구소 선임연구원
1990년 9월~현재: 순천향대학교 공과대학 정보보호학과 정교수
1997년 3월~2000년 3월: 순천향대학교 산업기술연구소 소장
2000년 4월~2006년 2월: 순천향대학교 산학연컨소시엄센터 소장
1997년 3월~현재: 한국정보보호학회 총무이사, 학술이사, 교육이사, 총무이사, 논문지편집위원 위원장(역), 수석부회장(현)
2005년~2008년: ITU-T SG17 Q.9 Rapporteur(역)
2006년 11월~2009년 2월: 정보통신연구진흥원 정보보호전문위원
2009년 5월~현재: 국정원 암호검증위원회 위원
2009년~현재: ITU-T SG17 부의장/SG17 WP2 의장
〈관심분야〉 인터넷보안, USN 보안, IPTV 보안, 홈네트워크 보안, 암호 프로토콜