情報保護學會論文誌
第17卷 第1號, 2007. 2

무선센서네트워크 환경에서 서명 기반 브로드캐스트 인증

구 우 권,† 이 화 성, 김 용 호, 황 정 연, 이 동 훈‡
고려대학교 정보경영공학전문대학원

Signature-based Broadcast Authentication for Wireless Sensor Networks

Woo Kwon Koo,† Hwaseong Lee, Yong Ho Kim, Jung Yeon Hwang, Dong Hoon Lee‡
Graduate School of Information Management and Security, Korea University

요 약

무선센서네트워크 환경에서 브로드캐스트 인증은 필수적인 보안 요소이다. 대표적인 브로드캐스트 인증 방식은 키 체인을 사용한 μ-TESLA 이다. 하지만 이 기법은 인증 시간의 지연이 불가피함으로 실시간 처리가 중요한 무선센서네트워크 환경에서 적합하지 않다. 따라서 본 논문에서는 인증 시간의 지연이 없을 뿐 아니라 re-keying 문제가 없는 효율적인 브로드캐스트 인증 기법을 제안하고 이에 대한 안정성 및 효율성을 분석한다. 이 기법은 무선센서네트워크 환경에 실질적 적용이 가능하고 장기적인 네트워크를 구성할 수 있다.

ABSTRACT

A broadcast authentication is important and fundamental consideration for security in wireless sensor networks. Perigg et al suggests μ-TESLA used a key chain. But it is unavoidable the delay of time to authenticate packets. so it is hard to meet the property that most application of sensor are performed in real-time. To cope with these problems we propose an efficient broadcast authentication scheme which has no delay of time and provides re-keying mechanism. we also describe an analysis of security and efficiency for this scheme.

Keywords : *Wireless Sensor Networks, Broadcast Authentication, Security*

Ⅰ. 서 론

무선센서네트워크(Wireless Sensor Networks)는 베이스 스테이션(Base Station)과 자원 제약이 있는 수 많은 센서노드(Sensor Node)들로 구성되어 있다. 무선센서네트워크 환경에서의 브로드캐스트 인증(Broadcast Authentication)은 네트워크 상의 기본적이고 중요한 보안 요소 중에 하나이다. 베이스 스테이션이 각 센서노드에게 명령, 지시, 경고 메시지를 보내야 할 경우 브로드캐스트는 통신량과 패킷 길이 측면에서 유니캐스트나 멀티캐스트보다 매우 효율적으로 수행할 수 있다. 그러나 만약 브로드캐스트 메시지에 대한 인증이 이루어지지 않는다면 잘못된 명령의 수행으로 인해 자원 낭비가 발생할 뿐 아니라 다양한 공격에 노출될 수 있어 네트워크 손실이 발생한다. 따라서 무선센서네트워크와 같은 브로드캐스트 네트워크 환경에서는 메시지를 보내는 송신자와 메시지에 대한 인증이 필수적이다.

무선센서네트워크 환경에서 브로드캐스트 인증을 위해 μ-TESLA[3] 방법이 제안되었다. 그러나 μ-TESLA

접수일: 2007년 1월 16일; 채택일: 2007년 2월 6일
* 이 논문은 2006년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No.R01-2004-000-10704-0)
† 주저자, kwk4386@korea.ac.kr
‡ 교신저자, donglee@korea.ac.kr

방법은 인증 시간의 지연이 발생한다. 이로 인해 센서노드는 인증키가 노출되기 전까지 인증되지 않은 패킷을 버퍼에 저장해야 하므로 자원 낭비가 발생한다. 이러한 μ-TESLA의 특징은 실시간 어플리케이션이 대부분인 무선센서네트워크 환경에서 적합하지 않다.

따라서 본 논문은 아래의 요구사항을 만족하는 효율적인 브로드캐스트 인증 방법을 제안하고 그에 대한 안전성과 효율성을 면밀히 분석한다.

1.1 요구사항

무선센서네트워크 환경에서 브로드캐스트 인증은 다음과 같은 요구 사항을 만족해야 한다.

- 즉각적인 인증 : 무선센서네트워크 애플리케이션은 대부분 실시간으로 이루어지기 때문에 인증지연이 발생하지 않아야 한다.
- 작은 계산비용 : 센서노드들은 제한된 계산능력을 가지고 있기 때문에 인증의 계산적 효율성을 필요로 한다.
- 작은 통신비용 : 센서노드들은 전송용량의 제약이 있으므로 인증 시 필요한 패킷 크기가 작아야한다.
- 작은 저장량 : 센서노드들은 저장 공간의 제약이 있기 때문에 사전 저장되는 정보가 작아야한다.

Ⅱ. 기존의 일회성 서명 스킴

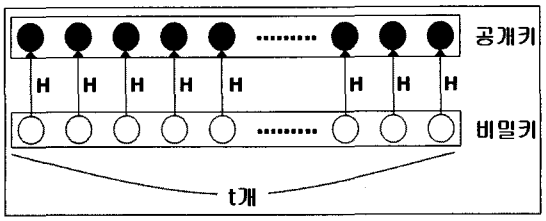
2.1 HORS[4]

HORS는 일회성 서명 기법을 이용하여 브로드캐스트 인증을 하지만 모든 센서 노드들이 저장하고 있어야 하는 공개키의 크기가 너무 크다는 단점을 가지고 있다.

(1) 키 생성

- 단계 1 : 베이스 스테이션은 t개의 l비트의 랜덤한 수로 구성된 비밀키(SK)를 생성한다.
  $SK= (S_0, S_1, \cdots\cdots, S_{t-1})$ $(S_i \in R \{0,1\}^l)$
- 단계 2 : 비밀키에 해쉬함수를 취하여 공개키(PK)를 생성한다. 공개키는 베이스 스테이션과 센서노드들이 사전에 공유하고 있는 키로 암호화 되어 전송 된다 ([그림 1]).
  $PK= (V_0, V_1, \cdots\cdots, V_{t-1})$ $(V_i=H(S_i))$

(2) 서명 생성



[그림 1] HORS의 키 생성

- 단계 1 : 해쉬함수 H(·)를 사용하여 인증하고자 하는 메시지(m)에 해쉬함수를 취한다.
  $h= H(m)$
- 단계 2 : h를 k개의 $\log_2 t$비트인 서브스트링으로 나눈다. 각 hi를 정수로 해석한다.
  $h= H(m)= h_0||h_1||...||h_{k-1}$ $(0 \le h_i \le t-1)$
- 단계 3 : 비밀키에서 $h_0, h_1, \cdots\cdots, h_{k-1}$ 번째 값을 선택하여 서명을 생성한다.
  $\sigma = (Sh_0, Sh_1, \cdots\cdots, Sh_{k-1})$

(3) 서명 검증

베이스 스테이션으로부터 받은 서명 $\sigma = (Sh_0, Sh_1, \cdots\cdots, Sh_{k-1})$을 각각 해쉬함수를 취해 가지고 있는 공개키와 비교하여 수락(Accpet) 혹은 거부(Reject)를 결정한다.

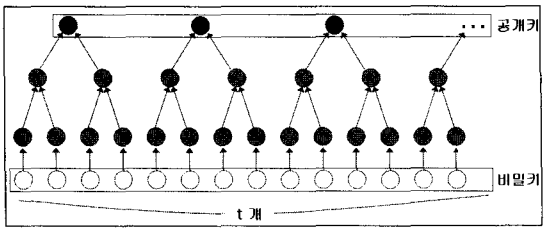
$f(Sh_i) \stackrel{?}{=} Vh_i$ $(0 \le i \le k-1)$

2.2 EBAS[5]

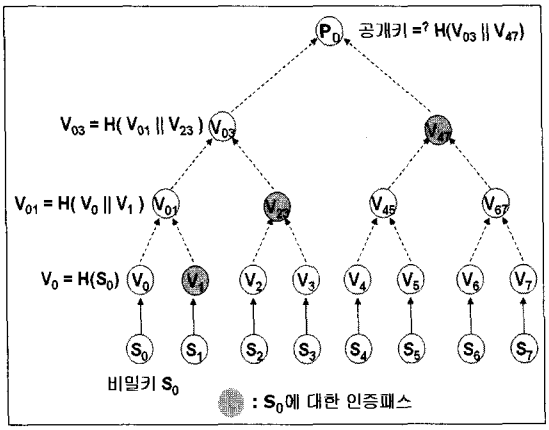
EBAS는 HORS의 각 노드들이 저장해야 할 공개키의 크기가 너무 크다는 문제점을 머클해쉬트리(Merkle Hash Tree)[2]를 사용하여 보완한 브로드캐스트 인증 기법을 제안했다. 따라서 EBAS는 서명 검증을 위해 추가적인 인증패스(Authentication path)를 필요로 한다. 이러한 인증 패스의 추가적인 전송을 줄이기 위해 d개의 서브 트리로 나누어 구성한다.

(1) 키 생성

- 단계 1 : HORS와 같은 방법으로 비밀키(SK) 생성한다.
  $SK = (S_0, S_1, \cdots\cdots, S_{t-1})$ $(S_i \in R \{0,1\}^l)$
- 단계 2 : 비밀키에 해쉬함수를 취하여 머클해쉬트리를 구성한다.
- 단계 3 : 공개키는 서브 머클트리(Merkle Tree)의



〔그림 2〕 EBAS의 키 생성



〔그림 3〕 인증 패스(Authentication path)

루트 값들의 집합으로 이루어진다(〔그림 2〕).

$PK = (P_0, P_1, \cdots\cdots, P_{d-1})$

(2) 서명 생성

서명 생성은 HORS와 같은 방법으로 생성한다. 단 EBAS에서는 검증 단계에서 공개키인 트리의 루트값을 생성하기 위해 인증패스가 필요하다. 따라서 서명은 $\sigma = (Sh_0, Sh_1, \cdots\cdots, Sh_{k-1})$와 각 Sh_i에 대하여 인증패스(Authentication path)(〔그림 3〕)로 이루어진다.

(3) 서명 검증

베이스 스테이션으로부터 받은 서명과 인증패스를 사용하여 생성한 루트 값을 공개키와 비교한 후 수락 혹은 거부를 결정한다.

Ⅲ. 제안하는 스킴

본 논문에서는 기존의 두 스킴에 대한 문제점들을 보완하는 개선된 스킴을 제안한다. 제안하는 스킴은 Bloom Filter[1]를 사용하여 HORS의 문제점인 공개키의 크기를 줄이고 EBAS의 문제점인 서명을 검증하기 위한 추가적인 인증 패스의 전송과 저장을 없앤다.

(1) 키 생성

- 단계 1 : 베이스 스테이션은 l비트의 랜덤한 수 S_0를 생성한 후 해쉬 체인을 생성하여 비밀키(SK)와 Commitment를 만든다.

  $SK = (S_0, S_1, \cdots\cdots, S_{t-1}), (S_i = H_i(S_0))$

  $Commitment = S_t$
- 단계 2 : 각 S_i에 대하여 s개의 Bloom Filter에 대응시키는 s개의 독립적인 해쉬함수($H_0, H_1, \cdots, H_{s-1}$)를 취하여 t개의 Bloom Filter를 생성한다〔그림 4〕. 공개키로 사용할 초기 Bloom Filter($BF_0, BF_1, \cdots\cdots, BF_{t-1}$)들과 Commitment(St)들을 각 센서 노드들을 배치하기 전에 사전 저장시킨다.

  $H_j(S_i), (0 \le i \le t-1,\ 0 \le j \le s-1)$

  $PK = (S_t), (BF_0, BF_1, \cdots\cdots BF_{t-1})$

(2) 서명 생성

- 단계 1 : 해쉬함수 $H(\cdot)$를 사용하여 인증하고자 하는 메시지(m)에 해쉬함수를 취한다.

  $h = H(m)$
- 단계 2 : h를 k개의 $\log_2 t$비트인 서브스트링으로 나눈다. 각 hi를 정수로 해석한다.

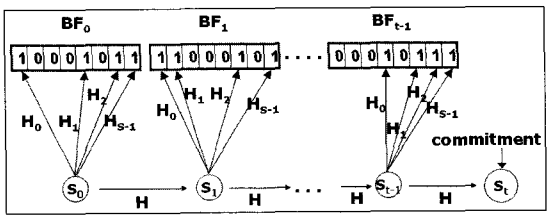
  $h = H(m) = h_0 || h_1 || \ldots || h_{k-1}\ (0 \le h_i \le t-1)$
- 단계 3 : 비밀키에서 $h_0, h_1, \cdots\cdots, h_{k-1}$ 번째 값을 선택하여 서명을 생성한다.

  $\sigma = (Sh_0, Sh_1, \cdots\cdots, Sh_{k-1})$

(3) 서명 검증

- 단계 1 : 받은 서명 Sh_0에 반복하여 해쉬함수를 취해 공개키의 Commitment와 일치하는지 확인한다.

  $H_{t-h0}(Sh_0) =^? S_t$
- 단계 2 : 해쉬함수 $H(\cdot)$를 사용하여 인증하고자 하



〔그림 4〕 제안하는 스킴의 키 생성

〔표 1〕 m/n, s 에 따른 false positive의 확률

m/n	s	s=1	s=2	s=3	s=4	s=5	s=6	s=7	s=8
2	1.39	0.393	0.400						
3	2.08	0.283	0.237	0.253					
4	2.77	0.221	0.155	0.147	0.160				
5	3.46	0.181	0.109	0.092	0.092	0.101			
6	4.16	0.154	0.080	0.060	0.056	0.057	0.063		
7	4.85	0.133	0.061	0.042	0.035	0.034	0.036		
8	5.55	0.118	0.048	0.030	0.024	0.021	0.021	0.022	
9	6.24	0.105	0.039	0.022	0.016	0.014	0.013	0.013	0.014
10	6.93	0.095	0.032	0.017	0.011	0.009	0.008	0.008	0.008

는 메시지(m)에 해쉬함수를 취한다.

h = H(m)

- 단계 3 : h를 k개의 $\log_2 t$비트인 서브스트링으로 나눈다. 각 h_i를 정수로 해석한다.

  $h = H(m) = h_0 \| h_1 \| ... \| h_{k-1}$ $(0 \le h_i \le t-1)$

- 단계 4 : 서명 Sh_i'에 대한 Bloom Filter를 생성한다.

  $\sigma' = (Sh_0', Sh_1', \cdots\cdots, Sh_{k-1}')$

  $H_j(Sh_i')$, $(0 \le i \le k-1,\ 0 \le j \le s-1)$

  $(BFh_0', BFh_1', \cdots\cdots, BFh_{k-1}')$

- 단계 5 : 4단계에서 생성한 Bloom Filter와 공개키(PK)가 일치하는지 확인함으로써 수락(Accpet) 혹은 거부(Reject)를 결정한다.

  $(BFh_0, BFh_1, \cdots\cdots, BFh_{k-1}) \stackrel{?}{=} (BFh_0', BFh_1', \cdots\cdots, BFh_{k-1}')$

(4) 키 재생성 방법 (Re-keying Mechanism)

키의 재생성은 서명 생성 시 인증하고자 하는 메시지에 다음 인증에 사용할 공개키를 붙여서 메시지와 다음의 공개키를 함께 인증함으로써 가능하다.

(m ‖ PK), H(m ‖ PK)

Ⅳ. 분석

5.1 안전성 분석 및 시스템 변수 설정

제안하는 스킴은 공개키의 크기를 줄이기 위해 Bloom Filter를 사용하므로 False Positive가 일어날 수 있다. 즉 올바르지 않은 Si에 대해서 $H_j(Sh_i)$가 Bloom Filter값에 정당하게 마킹(Marking)될 수 있다. 따라서 False Positive가 발생할 확률은 제안하는 스킴의 안전성에 영향을 준다. False Positive가 발생할 확률은 다음과 같다.

$$f = \left\{1 - \left(1 - \frac{1}{m}\right)^{sn}\right\}^s \approx \left(1 - e^{-sn/m}\right)^s$$

f:False Positive의 확률

s:해쉬함수의 개수

m:Bloom Filter의 크기

위의 식에서 Bloom Filter는 계산량(s)과 저장량(m)과 안전성(f) 사이에 세 가지 절충(Trade-Off)이 발생한다. 따라서 다음 식에서 세 가지 파라미터(Parameter) s, m, f의 최적의 값을 찾아야한다.

$$g = \ln(f) = s \cdot \ln\left(1 - e^{-sn/m}\right)$$

양변을 미분하면,

$$\frac{dg}{ds} = \ln\left(1 - e^{-sn/m}\right) + \frac{sn}{m} \cdot \frac{e^{-sn/m}}{1 - e^{-sn/m}}$$

critical point를 구하기 위해 $\frac{dg}{ds} = 0$을 만족하는 s를 구하면 다음과 같다.

$$s = (\ln 2) \cdot \frac{m}{n} \Leftrightarrow \left(\frac{1}{2}\right)^s = (0.6185)^{m/n}$$

〔표 1〕과 같이 m/n과 s에 따른 False Positive 확률을 구할 수 있다.

따라서 센서 노드의 저장량, 계산량, 안전성을 고려하여 최적의 파라미터 값을 m = 8n, s = 4, f = 0.024로 정한다면 n = 1, m = 8, s = 4일 때 False Positive의 확률이 약 2.4%가 된다.

공격자가 서명을 위조하기 위해서는 먼저 주어진 Commitment(S_t)에 대한 Pre-image들을 구해야한다. 안전한 해쉬함수는 H(x) = y를 만족하는 x를 찾기가 어려운 일방향성(One-wayness)과 H(x)=H(x')를 만족하는 충돌쌍(x, x')을 찾기 어려운 충돌방지성(Collision-resistance)의 성질을 가져야한다. 따라서 주어진 Commitment에 대해 Pre-image를 구하기는 계산적으로 불가능하다. 만약 충돌쌍을 구했다고 하더라도 위조한 서명 Sh_i에 대한 $Vh_i(0 \le i \le k-1)$가 Bloom Filter값을 모두 통과해야하므로 서명을 위조하는 것은 계산적으로 불가능하다.

5.2 효율성 분석

〔표 2〕 시스템 파라미터

파라미터	HORS	EBAS	제안하는 스킴
t (비밀키의 원소 개수)	256개	256개	32개
l (비밀키의 원소 크기)	80bit	80bit	64bit
H(해쉬함수 출력 크기)	160bit	160bit	64bit
k (서명의 원소 개수)	20개	20개	13개
d (서브머클해쉬트리의 개수)	-	16개	-
m (Bloom filter의 크기)	-	-	8bit
n (Bloom filter에 대응되는 원소의 개수)	-	-	4개
s (Bloom filter에 대응시키는 해쉬함수의 개수)	-	-	4개

기존의 스킴과 효율성을 비교, 분석하기 위해 HORS, EBAS와 제안하는 스킴에서 제시한 파라미터 값을 비교하였다〔표 2〕.

5.2.1 저장량

제시한 파라미터를 기준으로 기존 스킴과 제안하는 스킴을 비교해 보면 HORS에서 각 센서 노드는 베이스 스테이션의 다량의 공개키를 저장해야 한다. EBAS에서는 머클트리를 사용하여 공개키 크기를 줄였으므로 서브트리의 개수만큼의 공개키가 필요하다. 제안하는 스킴은 사이즈가 큰 공개키 대신에 Bloom Filter를 저장하면 되므로 센서 노드의 저장량은 40byte가 된다. 그러므로 저장량 측면에서 기존 스킴보다 효율적이다〔표 3〕.

5.2.2 통신량

세 스킴 모두 각 센서 노드들은 수신자 입장이 되기 때문에 통신량 분석에서는 노드들의 패킷 수신량만을 비교한다. HORS에서 노드들은 베이스 스테이션으로부터의 공개키와 메시지와 그에 따른 서명을 수신해야 한다. 메시지의 크기는 메시지에 따라 매번 달라지므로 고려하지 않는다. EBAS에서 서명에 각 서명을 구성하는 비밀키에 대해 인증 패스가 첨부되어서 전송된다. 제안하는 스킴에서는 최적화된 공개키와 서명만 수신하면 된다. 따라서 제안하는 스킴이 통신량 측면에서 효율적이다〔표 3〕.

5.2.3 계산량

제안하는 스킴의 센서 노드가 서명 검증에 필요한 연산은 기존의 스킴과 마찬가지로 해쉬함수 연산이다. 따라서 먼저 해쉬함수의 계산 시행 횟수를 비교해 보면 HORS는 21번 EBAS는 최대 101번 실행 해야한다. 그러나 모든 서명의 Sh_i 대해 항상 인증패스를 이용하여 루트값을 확인 하는 것보다 이미 인증된 해쉬트리의 노드들에 대해 저장을 함으로서 후에 중복되는 해쉬함수 계산을 줄일 수 있다. 따라서 EBAS에서는 저장량과 계산량과의 Trade-Off가 발생한다. 제안하는 스킴은 2번의 해쉬함수와 52번의 Bloom Filter에 대응시키는 해쉬함수의 계산량을 필요로 한다. 기존의 스킴보다 검증 오버헤드는 증가 했지만 해쉬함수 연산은 센서 노드에게 큰 부담이 되지 않을 뿐더러 센서노드의 총 자원의 소비 중에 통신량에 드는 자원 소비가 97%가 차지한다는 점에서 계산량의 증가보다는 통신량의 감소에 비중을 두어야한다〔표 3〕.

V. 결 론

본 논문에서는 Bloom filter를 사용하여 HORS의 문제점인 공개키의 크기를 줄이고 EBAS의 문제점인 추가적인 인증 패스의 전송과 저장을 없앤 무선센서네트워크 환경에 적합한 브로드캐스트 인증 스킴을 제안하였고 안전성과 효율성을 분석하였다. 제시된 인증 기법은 즉각적인 인증을 통해 인증 지연을 미연에 방지할 뿐 아니라 통신량, 저장량과 계산량 또한 효율적으로 개선하였다.

〔표 3〕 효율성 분석

	HORS	EBAS	제안하는 스킴
저장량(bytes)	5120	320	40
통신량(bytes)	5320	620	152
계산량(해쉬함수 연산회수)	21	51	85

참고문헌

[1] B. Bloom, "Space/time trade-offs in hash coding with allowable errors" *Commun. ACM*, vol. 13, no. 7, pp. 422-426, 1970.

[2] R. Merkle, "Protocols for public key cryptosystems," *in Proceedings of the IEEE Symposium on Research in Security and Privacy*, Apr 1980.

[3] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. Tygar. "SPINS: Security protocols for sensor networks," *Wireless Networks*.2002.

[4] L. Reyzin and N. Reyzin, "Better than BiBa: Short onetime signatures with fast signing and verifying," *In Seventh Australasian Conference on Information Security and Privacy* (ACISP 2002), July 2002.

[5] Shang-Ming, Chang Shiuhpyng Shieh, Warren W. Lin, and Chih-Ming Hsieh, "An Efficient Broadcast Authentication Scheme in Wireless Sensor Networks," *In Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. 2006.