Diffie-Hellman 알고리즘이 적용된 USN에서 타임스탬프를 이용한 악의적인 노드 검출

Detection of Malicious Node using Timestamp in USN Adapted Diffie-Hellman Algorithm

한승진[*], 최준혁[**]
경인여자대학 정보미디어학부[*], 김포대학 e-비즈니스과[**]

Seung-Jin Han(softman@kic.ac.kr)[*], Jun-Hyeog Choi(jhchoi@kimpo.ac.kr)[**]

요약

본 논문에서는 유비쿼터스 환경에서 OTP가 적용된 Diffie-Hellman 방식을 이용하여 노드간 키를 전달할 때 타임스탬프의 시간 차이를 이용하여 악의적인 노드를 검출할 수 있는 방법을 제안한다. 기존의 방식들은 정확한 시간 동기화나 방향성 안테나를 이용한 방법으로 악의의 노드 검출을 시도하였다. 본 논문에서는 방향성 안테나 추가 혹은 제 3 신뢰기관(TTP) 없이 타임스탬프를 이용한 OTP를 Diffie-Hellman 방식에 적용하여 중간의 악의노드 검출 방법을 제안하고 이에 대한 안전성을 검증한다. 본 논문에서 제안하는 방법은 유비쿼터스 환경에서도 쉽게 적용이 가능한 방법이다.

■ 중심어 : | USN(Ubiquitous Sensor Network) | MANET(Mobile Adhoc Networks) | Diffie-Hellman | 중간자공격(Man-in-the-middle Attack) | 재생공격 | 일회용 패스워드 | 키교환 | 제 3신뢰기관(TTP) |

Abstract

In this paper, we proposed scheme that we use a difference of timestamp in time in Ubiquitous environments as we use the Diffie-Hellman method that OTP was applied to when it deliver a key between nodes, and can detect a malicious node at these papers. Existing methods attempted the malicious node detection in the ways that used correct synchronization or directed antenna in time. We propose an intermediate malicious node detection way at these papers without an directed antenna addition or the Trusted Third Party (TTP) as we apply the OTP which used timestamp to a Diffie-Hellman method, and we verify safety regarding this. A way to propose at these papers is easily the way how application is possible in Ubiquitous environment.

■ keyword : | USN(Ubiquitous Sensor Network) | MANET(Mobile Adhoc Networks) | Diffie-Hellman | Man-in-the-Middle Attack | Replay Attack | One Time Password | Pair-Wise Key | Trusted Third Party |

접수번호 : #081002-003
접수일자 : 2008년 10월 02일
심사완료일 : 2008년 11월 11일
교신저자 : 한승진, e-mail : softman@kic.ac.kr

Ⅰ. 서 론

MANET(Mobile Ad hoc Networks)과 센서 네트워크는 고정된 기반 시설(Infrastructure) 없이 이동 혹은 이동성이 거의 없는 노드간 패킷을 주고 받는 무선 네트워크로서 유비쿼터스 센서 네트워크 (Ubiquitous Sensor Networks) 환경을 구축하기 위한 핵심 기술로서 많은 연구가 진행되고 있다[1][2]. 그러나 USN에서의 노드는 패킷을 송수신하는 노드의 역할 뿐만 아니라, 다른 노드에서 전송되어 온 패킷을 또 다른 노드로 전달해야 하는 라우터 기능까지 포함한다. 또한 소스 노드로부터 목적지 노드까지의 거리가 1 홉으로 이루어지는 경우도 있지만 대부분 n 홉(n ≥ 2)으로 이루어져 있기 때문에 소스 노드와 목적지 노드의 중간에 악의적인 목적을 가진 노드가 존재한다면 많은 문제가 발생한다.

USN은 중간의 노드를 경유하여 목적지까지 패킷을 전송하기 때문에 중간 노드가 악의적인 의도를 갖게 되는 경우 이 노드로부터 선의의 노드를 보호할 수 있는 장치 혹은 방법이 필요하다. 그러나 USN은 무선 환경에서 기반 시설 없이 오직 노드간 패킷을 주고 받는 방식이기 때문에 보안에 대한 많은 문제점이 드러나고 있다[3-5]. 소스 노드에서 목적지 노드까지 패킷을 안전하게 보내는 방법 중 하나로 소스 노드와 목적지 노드간 주고 받는 패킷을 암호화하는 방법이 있다. 그러나 USN의 노드들은 대부분이 배터리로 동작을 하기 때문에 유선망에서 사용하는 강력한 암호화 알고리즘을 사용하기에는 충분한 자원을 가지고 있지 않다.

본 논문에서는 OTP(One Time Password)가 적용된 Diffie-Hellman의 키 교환 알고리즘[6]을 USN 환경에 적용하여 노드간 키 전달 시 사용하는 타임스탬프 값을 이용하여 악의적인 노드를 검출하는 방법을 제안한다. 이와 같은 방법은 유비쿼터스 센서 네트워크(USN)에서 노드들이 타임 스탬프만을 이용하여 간단히 악의적인 노드를 검출할 수 있기 때문에 활용 가능성이 크다고 볼 수 있다.

Ⅱ. 관련연구

센서 네트워크에서 많은 보안에 대한 선행 연구들이 있다. 기존의 연구들은 크게 악의적인 목적을 갖는 노드의 침입 탐지(Intrusion Detection), 키 관리(Key Management), 안전한 라우팅을 위한 보안(Secure Routing), 이기적 노드(Selfish Node)의 탐지 및 관리 등이 있다. 2장에서는 본 논문과 관련있는 침입 탐지(Intrusion Detection)에 대한 기존 연구를 살펴본다.

[7]에서는 모든 노드가 이웃 노드의 전송을 overhear 할 수 있다는 가정하에 노드간 비대칭 암호방식을 이용하였다. 신고자와 혐의자 목록으로 구성되는 신고 메시지와 신고 테이블을 이용하여 데이터를 버리는 경우, 데이터를 변형시키는 경우, 다른 노드로 위장하여 거짓 신고하는 경우, 다른 노드를 임의로 거짓 신고하는 경우, 정상적인 노드를 거짓 신고하는 경우 등 5가지로 나누어 악의적인 노드를 검출할 수 있는 방법을 제안하였다.

[8]에서는 데이터 마이닝 기법을 이용하여 정상적인 노드의 라우팅 동작에 대한 패턴과 비정상적인 노드의 라우팅 동작과 비교하여 악의적인 노드를 검출하는 방법을 제안하였다.

[9]에서는 네트워크 동작의 데이터를 수집하고 정상적인 노드의 동작과 기준이 되는 정상적인 동작과의 차이점을 비정상적인 노드의 동작과 비교하는 LocalIDSs(Intrusion Detection Systems)를 MANET의 각 노드에 부착하여 악의적인 노드를 검출하였다. 비정상적인 노드의 리스트는 각 클러스터 헤드로 전송되고, 또한 이 리스트는 클러스터 헤드를 통해 네트워크 전체로 전송된다. 그러나 각 노드에 LocalIDSs라는 장치를 추가하고, 네트워크 전체 데이터(raw data)를 수집해야 된다는 것이 네트워크 입장에서는 상당한 부하이다.

무선 네트워크 환경에서 웜홀 검출에 대한 연구는 초창기 정확한 시간 동기화[11][12]나 방향성 안테나를 이용한 방법[13]으로 웜홀 검출을 시도하였으나 별도 하드웨어에 대한 부담이 발생한다.

본 논문에서는 OTP가 적용된 Diffie-Hellman을 이용한 키 교환 시스템이 적용된 센서 네트워크 환경에서 네트워크의 성능 저하 없이 노드간 메시지 암호화와 시간 동기화를 위해 전송되는 타임 스탬프를 이용하여 악의적인 노드를 검출할 수 있는 방법을 제안하고 이에 대한 안전성과 견고성에 대해 검증한다.

Ⅲ. 타임스탬프를 이용한 악의적인 노드검출

USN은 TTP를 설치하기가 상당히 어려운 구조 특성을 가지고 있다. 또한 모든 노드들은 배터리를 사용한 장치(예를 들어, 배터리를 이용한 노트북 혹은 태양 전지나 배터리를 이용한 센서노드)이기 때문에 유선 장치에 비해서 저전력 소모가 가능해야 하고, 대역폭이 작으며, 서비스 품질(QoS)이 현저하게 낮다.

Diffie-Hellman의 키 교환 알고리즘은 사전의 키 분배가 없고, TTP가 없이 두 노드간에 키 교환이 가능하다. 이는 MANET 혹은 USN과 같이 기반 구조가 없는 환경에서 키 교환 방식으로 적합하다.

본 논문의 키 교환 시스템은 Diffie-Hellman과 타임스탬프를 이용한 OTP를 이용하여 중간에 위치한 악의적인 노드가 잘못 된 OTP 값을 삽입하는 것을 검출할 수 있는 방법을 제안한다.

3.1 용어 정의

다음은 본 논문에서 사용하는 용어들의 정의이다[6].

표 1. 용어 정의

S, A, B, D : 사용자 노드
M : 악의적인(Malicious) 노드
x_0^A : 노드 A의 비밀 메시지(혹은 패스워드)
a : 노드 A의 개인 키
$g_a \leftarrow g^a \pmod p$: 노드 A의 공개 키
k_n : n 번째 세션 키
$H(\ )$: 단방향 해쉬 함수
$H(g_a)$: 노드 A의 공개 키 검사
$x_k^A \leftarrow H^k(x_0^A)$: $H(\ )$를 이용하여 x_0^A 를 k 번 해쉬한 결과 값
$c \leftarrow E_K\{d\}$: K 를 이용하여 평문 d 를 암호문 c 로 대칭적으로 암호화
$d \leftarrow D_K\{c\}$: K 를 이용하여 암호문 c 를 평문 d 로 대칭적으로 복호화
g : 곱셈 군(multiplicative group) Z_p^*의 생성자(generator), p의 원시근(primitive root)
p, q : 강한 소수(strong prime), p = 2 × q + 1
K' : 소스 노드와 목적지 노드가 이전 세션에서 사용하던 키
K_{AB} : 노드 A와 노드 B가 공유하는 비밀 키
N_B : 노드 B에서 전송한 난수
T_S : 소스 노드의 타임스탬프
T_K : K 번째 노드의 타임스탬프
T_D : 목적지 노드의 타임스탬프
$T_{N_{SD}}$: 소스 노드와 목적지 노드의 세션 종료 시간
$T_{threshold}$: 각 노드가 수용하는 타임스탬프 임계치

3.2 타임 스탬프를 이용한 악의적인 노드 검출

센서 네트워크의 구조 특성상 소스 노드에서 목적지 노드까지가 1-홉인 경우도 있지만 대부분 2-홉 이상으로 소스 노드와 목적지 노드 사이에 중간 노드들이 존재한다. 따라서 소스 노드와 목적지까지의 거리가 2-홉 이상인 경우는 중간 노드를 경유해야 패킷을 전달할 수 있다. 이때 중간에 위치하는 악의적인 노드는 소스 노드에게 자신이 목적지 노드라고 위장하고, 목적지 노드에게는 자신이 소스 노드라고 위장하면, 소스 노드와 목적지 노드만이 공유해야 하는 키를 악의적인 노드가 알 수 있다. Diffie-Hellman 키 교환 방식은 소스 노드와 목적지 노드가 상호 인증(Mutual Authentication)을 하지 않기 때문에 중간자 공격이 가능하다.

[6]에서는 이러한 중간자 공격으로부터 취약한 상호 인증 문제를 해결하기 위해 μTESLA[10]를 사용하였다. μTESLA는 TESLA를 센서 노드에 적합하도록 인증부분만 분리한 것으로 인증 키 체인 생성이나 브로드캐스트 데이터 생성 방식은 TESLA와 유사하다. μTESLA가 베이스 스테이션만이 센서 노드들에게 브로드캐스트할 수 있는 것을 본 논문에서는 모든 노드들이 가능하도록 수정한다. 그러나 μTESLA는 모든 노드들이 시간 동기화가 되어 있어야 하고 실제 네트워크 전송 지연이 있어 키 노출 지연 시간의 설정이 필요하다.

본 논문에서는 모든 노드들의 시간 동기화를 하지 않고, 패킷을 송수신하는 노드들끼리만 동기화를 한다. 시간 동기화를 위해서는 노드들 간의 타임스탬프를 이용하지만, 모든 노드들에 모든 시계가 모두 완전하게 일치되는 것을 기대할 수 없으므로 적당한 시간오차(Clock Skew)를 고려한다. 본 논문에서는 시간오차를 최소화하고, 다음에서 설명하는 재생 공격을 방지하기 위해 [6]에서 사용한 노드간 타임스탬프를 이용하여 악의적인 노드를 검출하는 방법을 제안한다.

노드들 간의 시간 동기화 문제는 [그림 2]처럼 소스 노드(S)가 목적지 노드로 패킷을 전송할 때 패킷에 타임스탬프를 포함해서 전송한다. 이를 수신한 노드 A는 암호화된 자신의 타임스탬프를 패킷에 추가하여 노드 B에게 전달한다. 노드 B는 노드 A와 같이 암호화된 타임스탬프를 추가하여 다음 노드로 전송하고, 최종적으로 목적

지 노드(D)는 패킷을 수신한다. 목적지 노드의 타임스탬프(T_D)는 소스 노드의 타임스탬프(T_S)와 중간 노드들의 타임스탬프(T_2, T_3, T_4)를 참조하여 소스 노드와의 시간을 동기화한다.

[11]에서는 노드들 간의 시간 오차(Clock Skew; 커버로스[14]에서는 5분을 시간오차로 두고 있다. 본 논문은 노드들이 분포되어 있는 지역의 넓이가 미국의 동부와 서부와 같이 지역 간에 발생하는 시간 오차가 발생할 정도로 넓지 않다고 가정한다.)를 극복하기 위해 소스 노드는 목적지 노드에게 타임스탬프를 전송한다. 소스 노드의 타임스탬프가 추가된 메시지를 수신한 중간 노드는 소스 노드에게서 자신까지 패킷이 전송되는데 소요된 시간 정보를 추가하여 다음 노드로 전송한다. 이를 수신한 다음 노드 역시 시간 정보를 추가하여 다음 노드로 전송한다. 이러한 과정을 거친 후 타임스탬프가 추가된 메시지가 목적지 노드에게 최종적으로 전달이 되면 목적지 노드는 소스 노드가 전달한 타임스탬프와 중간 노드들이 추가한 타임스탬프를 이용하여 자신의 시간 정보와 비교한다. 이때 특정 노드가 추가한 타임스탬프가 이전 노드보다 작거나 현저하게 크다면 목적지 노드는 이를 악의적인 노드로 간주하고 해당 타임스탬프를 삭제하고, 해당 노드는 악의적인 노드로 규정하고 이에 대한 정보를 이웃 노드로 브로드캐스트 한다. 이 정보를 수신한 이웃 노드들은 표 1과 같은 자신의 테이블에 특정 노드가 악의의 노드라는 사실을 등록하지 않고 특정 노드가 악의의 노드라는 것을 그 외에 다른 노드들이 추가로 알려온다면 해당 노드들 악의의 노드로 규정한다. 이는 특정 노드가 거짓으로 임의의 노드를 악의의 노드라고 알리는 것을 방지하기 위함이다.

수신한 노드에서 타임스탬프의 시간 차이가 (수식 1)의 조건을 만족하면 이를 무시하고, (수식 1)의 조건을 만족하지 못하면 시간 정보를 수정한다.

$$|Clock - T| < \Delta t_1 + \Delta t_2 \quad (1)$$

여기서, T는 송신 노드가 전송한 타임스탬프이고, Clock은 수신자의 로컬 타임스탬프이고, Δt_1은 소스 노드와 목적지 노드간의 시간 차이이다. Δt_2는 노드 간 패킷 전달 지연 시간이다.

여기서, Δt_1과 Δt_2를 사용하지 않고 타임스탬프를 이용하는 이유는 악의적인 목적을 갖는 노드가 이전 노드에게서 수신한 타임스탬프로부터 정확한 시간정보를 추가하지 않고, 큰 Δt_2 값을 추가함으로써 이전에 사용되었던 세션 키를 사용할 수 있도록 하는 것을 방지하기 위함이다.

목적지 노드는 소스 노드와 중간 노드들이 전송한 타임스탬프를 이용하여 자신의 시간을 동기화하고(상호 인증을 하기 위해 μTESLA를 사용하기 위한 방법도 포함) 소스 노드와의 시간 오차를 최소화하여 재생 공격으로부터 대비한다.

소스 노드는 자신의 인증 검사에 대한 무결성을 위해 μTESLA를 이용한다. 이때 μTESLA에서 필요한 노드들 간의 시간 동기화는 소스 노드와 목적지 노드간 주고받는 메시지내의 타임스탬프를 이용하여 동기화 한다. 여기서는 센서 노드에서 적용되는 μTESLA처럼 모든 노드들 간의 시간 동기화가 필요 없고, 오직 소스 노드와 목적지 노드간만이 동기화가 필요하다.

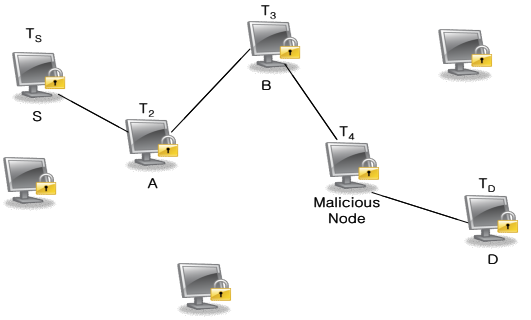

그림 1. 수정된 Diffie-Hellman을 적용한 센서네트워크

본 논문에서는 기존의 Diffie-Hellman 방법을 다음과 같이 수정한 키 교환 시스템을 이용한다.

소스 노드(S)는 $a \in_A [1, p-1]$를 선택하고, g^a를 계산한다.

$$\left\{S \rightarrow A(D): ID_A, E_{x^A_{n_A - i}}(g^a, T_S), x^A_{n_A - i + 1}, i\right\} \quad (2)$$
$$where, 1 \le (n_A, i) \le k$$

[그림 1]처럼 메시지 (2)를 수신한 노드 A는 자신이 암호화한 타임스탬프를 추가하여 다음 노드인 노드 B로 메시지를 전달한다. B와 C는 A의 방법과 동일한 방법으로 각각 C와 D에게 메시지를 전달한다.

$$\left\{A \rightarrow B(D) : [ID_A, E_{x^A_{n_A-i+1}}(g^a, T_S), x^A_{n_A-i+2}, i], T_2\right\} \quad (3)$$

$$\vdots$$

$$\left\{C \rightarrow D : [[[ID_A, E_{x^A_{n_A-i+(k-1)}}(g^a, T_S), x^A_{n_A-i+(k)}, i], T_2], \ldots, T_{k-1}\right\} \quad (4)$$

목적지 노드 D는 다음 (수식 5)와 (수식 6)을 체크한다.

$$H^k(x^A_{n_A-i+k}) = x^A_{n_A} \quad (5)$$

$$T_S < T_2 < \ldots < T_{k-1} < \ldots < T_D \quad (6)$$

소스 노드 S가 목적지 노드 D가 이전에 메시지를 송수신하여 [표 2]와 같은 테이블에 존재한다면 소스 노드 S의 타임스탬프 T_S는 $T_{N_{SD}}$보다 커야 한다. 즉, (수식 7)과 같은 조건을 만족해야 한다.

$$T_S > T_{N_{SD}} \quad (7)$$

여기서, $T_{N_{SD}}$는 소스 노드 S와 목적지 노드 D의 세션이 이전 단계에서 종료된 시간이다.

여기서 특정 노드(N)의 타임스탬프의 값이 다음 노드($N+1$)의 타임스탬프 값보다 작거나, 이전 노드들의 타임스탬프에 비해서 현저히 크다면 목적지 노드는 해당 노드를 공격자로 규정하고 해당 노드 ID를 이웃 노드로 브로드캐스트 한다.

이때, 소스 노드 S와 목적지 노드 D와의 세션 종료시간이 [표 2]에 존재하지 않는다면 목적지 노드 D는 소스 노드 S가 최초 전송으로 간주한다. 그러나 $T_{N_{SD}}$는 다음을 만족해야 한다.

$$T_{threshold} < T_{N_{SD}} \quad (8)$$

여기서, $T_{threshold}$는 각 노드가 정한 다른 노드들과의 세션 종료 시간에 대한 임계치이다.

공격자는 키의 freshness를 위해 가급적 최근에 종료된 세션 키를 이용하려 할 것이다. 따라서 $T_{N_{SD}}$가 임계치 $T_{threshold}$ 범위를 벗어난다면, 목적지 노드는 재생 공격으로 간주하고 해당 메시지를 폐기하고, [표 2]에 악의노드 여부에 표시 후 이웃 노드로 브로드캐스트 한다. 각 노드는 다음과 같은 테이블을 생성하고 관리한다.

표 2. 연결이 종료된 노드들과의 세션 종료 시간

노드	세션 종료 시간	악의노드 보고 횟수	악의노드 보고 노드
1	TT:MM:SS	0	
2	TT:MM:SS	1	1,3
3	TT:MM:SS	3	1,2,4
:	:	:	
N	TT:MM:SS	0	

여기서, TT:MM:SS는 각각 노드 1, 2, 3, ... , N 이 [표 2]를 관리하는 노드와 세션이 종료된 시간이다. 보고 횟수는 이웃의 노드들로부터 특정 노드에 대해 악의 노드라고 보고가 들어온 횟수이고, 악의 노드 보고 노드는 특정 노드가 악의 노드라고 이웃 노드에게 브로드캐스트한 노드이다. 위 두가지는 임의의 노드가 자신의 이웃노드를 악의 노드라고 거짓으로 보고하는 것을 방지하기 위함이다.

목적지 노드는 위의 조건이 만족한다면 $x^A_{n_A-i+(k-1)}\{g^a\}$과 $x^A_{n_A-i+k}$를 저장한다.

목적지 노드(D)는

$b \in {}_A[1, q-1]$를 선택하고, g^b 를 계산한다.

나머지는 소스 노드가 목적지 노드로 전송하는 (수식 2) ~ (수식 8)과 유사하다.

[그림 2]는 위에서 설명한 내용을 알고리즘 형태로 표현한 것이다.

각 노드는 다음에 경로 설정 혹은 중간 노드의 역할을 하는 경우 자신이 가지고 있는 [표 2]를 참조하여 경로 설정을 한다. 따라서, 이웃 노드들로부터 악의 노드로 의심받는 노드는 소스 노드로서 경로 설정 및 중간 노드로

서 경로 설정에 참여할 수 없게 된다.

Node S(소스 노드)
1. $a \in_A [1, p-1]$ 선택
2. g^a 계산
3. *next* *node*(A)로 $\left\{S \rightarrow A(D): ID_A, E_{x^A_{n_A-i}}(g^a, T_S), x^A_{n_A-i+1}, i\right\}$ 전송

Node A
1. 메시지에 암호화된 타임스탬프 추가
2. *next* *node*(B)로 $\left\{A \rightarrow B(D): [ID_A, E_{x^A_{n_A-i+1}}(g^a, T_S), x^A_{n_A-i+2}, i], T_2\right\}$ 전송
...
Node C
1. 메시지에 암호화된 타임스탬프 추가
2. *next* *node*(D)로 $\left\{C \rightarrow D: [[[ID_A, E_{x^A_{n_A-i+(k-1)}}(g^a, T_S), x^A_{n_A-i+(k)}, i], T_2], \ldots, T_{k-1}\right\}$ 전송
Node D(목적지 노드)
1. if $H^k(x^A_{n_A-i+k}) = x^A_{n_A}$ then
2. if ($T_S < T_2 < \ldots < T_{k-1} < \ldots < T_D$) AND ($T_S > T_{N_{SD}}$) AND ($T_{threshold} < T_{N_{SD}}$) then
3. $x^A_{n_A-i+(k-1)}\{g^a\}$, $x^A_{n_A-i+k}$ 저장
4. $b \in_A [1, q-1]$ 선택
5. g^b 계산
6. else
7. 수신된 메시지 폐기
8. 악의 노드 목록 저장
9. 해당 노드를 악의 노드로 이웃 노드에게 브로드캐스트
10. # 브로드캐스트 메시지를 수신한 이웃노드는 악의 노드 보고 노드 및 횟수를 저장
11. End if
12. End if

그림 2. 타임 스탬프를 이용한 악의적인 노드 검출 알고리즘

Ⅳ. 분석

본 논문에서 연구하는 시스템의 보안성을 평가하기 위해 다음과 같은 [7]에서 제안하는 알고리즘 중 각 경우에 대해 본 논문에서 제안하는 방법이 타당한지 분석한다.

- Case 1 : 데이터를 변형시키는 경우

악의의 노드가 이전 노드에게서 받은 데이터를 변형시키기 위해서는 목적지 노드의 개인키를 알아야만 한다. 따라서 데이터를 변형시키는 것은 불가능하다.

- Case 2 : 다른 노드로 위장하여 거짓 신고하는 경우

본 논문에서 제안하는 알고리즘은 비대칭 암호방식을 사용하기 때문에 다른 노드로 위장하여 거짓 신고하기 위해서는 해당 노드의 개인키를 알아야 하기 때문에 Case 1처럼 불가능하다.

- Case 3 : 다른(혹은 정상적인) 노드를 임의로 거짓 신고하는 경우

특정 노드가 주변 노드들을 거짓 신고하게 되면 신고를 받은 주변 노드들의 테이블에는 표 1의 악의 노드 보고 노드에 특정 노드만 추가될 것이다. 즉 주변의 모든 노드들의 테이블에 거짓 신고한 특정 노드만 추가될 것이다. 따라서 거짓 신고자임을 식별할 수 있게 된다.

- Case 4 : 타임스탬프 값을 올바르게 입력하지 않은 경우

특정 노드가 타임스탬프 값을 비 정상적으로 크게 입력하였다면 해당 값은 다음 노드로 전달될 것이며 다음의 정상 노드는 정상적인 타임스탬프를 입력하여 목적지 노드로 전송한다. 목적지 노드는 중간 노드들의 타임 스탬프를 보고 특정 노드가 악의의 노드라는 것을 판단하고 소스 노드와 이웃노드로 신고한다.

Ⅴ. 결론

본 논문에서는 유비쿼터스 환경으로 진입하기 위한 핵심기술인 USN 기술 중 센서 네트워크에서 소스 노드와 목적 노드간에 TTP와 기타 별도의 하드웨어 도움 없이 각 노드들이 관리하는 세션 종료 테이블을 이용하여 악의의 노드 검출이 가능함을 보였고, 분석을 통해 네트워크의 성능 저하없이 악의의 노드를 검출할 수 있음을 입증하였다. 향후 [6]에서 제안한 방법에 포함하여 RFID/USN에서 악의의 노드 검출이 가능한 암호 키 교환 메커니즘으로 응용이 가능하다.

참 고 문 헌

[1] C. E. Perkins, *Ad Hoc Networking, Addison-Wesley*, 2001.

[2] IETF MANET Working Group, http://www.ietf.org/html.charters/manet-charter.

html, 2004.

[3] G. C. Wang, G. H. Cho, and S. W. Bang, "A Pair-wise Key Establishment Scheme without Pre-distributing Key for Ad-hoc Networks," ICC'05, Vol..5, pp.3520-3524, pp.16-20, 2005(5).

[4] 서승현, 조태남, 이상호, "OTP-EKE : 원-타임-패스워드 기반의 키 교환 프로토콜", 정보과학회 논문지 : 시스템 및 이론, 제29권 제5호, 한국정보과학회, 2002(6).

[5] Wenbo Mao, *Modern Cryptography : Theory and Practice*, Prentice Hall, 2003(7).

[6] 한승진, 최준혁, "MANET에서 제 3 신뢰기관(TTP)과 사전 키 분배가 없는 강한 키 교환 방식", 한국컴퓨터정보학회 논문지, 제13권 제5호, 2008(9).

[7] 이강석, 최종오, 지종복, 송주석, "MANET에서 악의적인 노드의 안전하고 효율적인 검출 방안", 정보처리학회논문지 C, 제12-C권 제5호, 2005(10).

[8] H. Liu and R. Gupta, "Temporal Analysis of Routing Activity for Anomaly Detection in Ad hoc Networks," Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), pp.505-508, Vancouver, 2006(10).

[9] B. D. C. João, G. Carlos, and K. M. Raman, "Infrastructures and Algorithms for Distributed Anomaly-Based Intrusion Detection in Mobile Ad-Hoc Networks," In Proceedings of the IEEE Military Communications Conference (IEEE MILCOM 2005), Atlantic City, NJ, 2005(10).

[10] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," Proc., of the 7th ACM/IEEE International Conference on MobiCom, 2001.

[11] Y. C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in Proceedings of the 22nd INFOCOM, pp.1976-1986, 2003.

[12] S. Capkun, L. Buttyan, and J. P. Hubaux, "SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks," in Proceedings of the 1st ACM workshop on SASN'03, pp.21-32, 2003.

[13] L. Hu and D Evans, "Using Directional Antennas to Prevent Wormhole Attacks," in Proceedings of the 1st ACM workshop on NDSS'04, 2004(2).

[14] S. Miller, "Kerberos Authentication and Authorization System," Section E.2.1, Project Athena Technical Plan, MIT. Project Athena, Cambridge, MA. 1988(10).

저 자 소 개

한 승 진(Seung-Jin Han) 정회원

- 1990년 2월 : 인하대학교 전자계산학과(이학사)
- 1992년 2월 : 인하대학교 전자계산공학과(공학석사)
- 2002년 2월 : 인하대학교 전자계산공학과(공학박사)
- 1992년 ~ 1996년 대우통신 종합연구소
- 1996년 ~ 1996년 : 한국전산원 초고속사업단
- 1996년 ~ 1998년 : SK Telecom 디지털 사업본부
- 2002년 ~ 2004년 : 인하대학교 컴퓨터공학부 강의조교수
- 2004년 ~ 현재 : 경인여자대학 정보미디어학부 조교수
- 2007년 ~ 현재 : TTA PG103 표준화위원

<관심분야> : USN, MANET, Mobile Computing, Embedded System, Security

최 준 혁(Jun-Hyeog Choi) 정회원

- 1990년 2월 : 경기대학교 전자계산학과(이학사)
- 1995년 2월 : 인하대학교 전자계산공학과(공학석사)
- 2000년 2월 : 인하대학교 전자계산공학과(공학박사)
- 1997년 ~ 현재 : 김포대학 e-비즈니스과 부교수
- 2001년 ~ 2002년 : 한국전자통신연구원 컴퓨터소프트웨어연구소(초빙연구원)
- 2003년 ~ 현재 : 특허청 특허출원 심사자문위원
- 2003년 ~ 현재 : 김포발전연구소 소장

<관심분야> : 정보검색, 유전자 알고리즘, 신경망, Embedded System, 전자상거래 보안