情報保護學會論文誌
第16卷 第1號, 2006. 2

무선 센서 네트워크를 이용한 무인 경비 시스템에서의 OMAC-SNEP 기술에 관한 연구

이 성 재,[1†‡] 김 학 범,[2] 염 흥 열[2]
[1]KT링커스, [2]순천향대학교

Study on the OMAC-SNEP for Unattended Security System Using Wireless Sensor Networks

Seong-Jae Lee,[1†‡] Hak-Beom Kim,[2] Heung-Youl Youm[2]
[1]KT Linkus, [2]SoonChunHyang University

요 약

USN은 모든 사물에 컴퓨팅과 통신기능 및 센싱 기능을 부여하여 언제, 어디서나, 통신이 가능한 환경을 구축하는 네트워크로서, 향후에는 다양한 센싱 기능이 추가되어 이들 간의 네트워크가 구축되는 형태로 발전할 것이다. 따라서 본 논문에서는 무선 센서 네트워크를 무인 경비 시스템에 적용할 경우 나타날 수 있는 정보 보안상의 취약점을 도출하고, 현재 제안된 보안 프로토콜 중에서 SNEP(Secure Network Encryption Protocol)을 사용하여 안전한 무선 센서 네트워크를 만들고자 한다. 그러나 SNEP에 사용한 CBC-MAC은 메시지의 길이가 가변일 경우 안전하지 못하기 때문에, 오직 한 개의키를 갖으면서도 임의의 길이 메시지도 안전하게 취급할 수 있는 OMAC(One-Key CBC-MAC)를 SNEP에 적용한 새로운 기법인 OMAC-SNEP을 제안하고, 구현하였다. 따라서 본 OMAC-SNEP 기법은 무인 경비 시스템은 물론 기타의 무선 센서 네트워크에서도 널리 활용이 가능할 것이라 생각된다.

ABSTRACT

Ubiquitous Sensor Network consists of a number of sensor nodes with a limited computation power and limited communication capabilities, and a sensor node is able to communicate with each other at anytime and in any place. Due to the rapid research and development in sensor networks, it will rapidly grow into environments where human beings can interact in an intuitive way with sensing objects which can be PDAs, sensors, or even clothes in the future. We are aiming at realizing an Unattended Secure Security System to apply it to Ubiquitous Sensor Network. In this paper, the vulnerabilities in the Unattended security system are identified, and a new protocol called OMAC-SNEP is proposed for the Unattended Secure Security System. Because the CBC-MAC in SNEP is not secure unless the message length is fixed, the CBC-MAC in SNEP was replaced with OMAC in SNEP. We have shown that the proposed protocol is secure for any bit length of messages and is almost as efficient as the CBC-MAC with only one key.

OMAC-SNEP can be used not only in Unattended Security System, but also any other Sensor Networks.

Keywords : *USN, Sensor Network, SNEP, OMAC*

접수일 : 2005년 11월 7일 ; 채택일 : 2006년 1월 2일
† 주저자, ‡ 교신저자 : seongjlee@paran.com

Ⅰ. 서 론

유비쿼터스 컴퓨팅(Ubiquitous Computing)은 새로운 개념의 IT 패러다임으로, 1988년 미국 제록스 팰로앨토 연구소의 마크 와이저(Mark Weiser)[1]에 의해 제안된 개념이다.

USN(Ubiquitous Sensor Network)이란 "필요한 모든 사물에 전자식별 태그를 부착하고 이를 통하여 사물의 인식 정보를 기본으로 주변의 환경정보(온도, 습도, 압력, 오염, 균열 등)까지를 탐지하여 이를 실시간으로 네트워크에 연결하고, 관련 정보를 관리하는 것"을 말한다.

센서 네트워크는 텔레메틱분야, 홈네트워크분야, 비상 대응분야, 의료분야 그리고 재고관리분야 등 다양한 분야에서 활용이 가능할 것으로 전망된다.

본 논문에서는 무선 센서 네트워크를 이용하여 무인 경비 시스템을 구축할 경우, 네트워크 구성방법을 제안하고 보안 취약성을 도출하며 정보보호 프로토콜 중에서 SNEP(Secure Network Encryption Protocol)을 사용하여 안전한 무선 센서 네트워크를 만들고자 한다. 그러나 SNEP에서 사용된 CBC-MAC은 메시지의 길이가 가변일 경우 안전하지 못하기 때문에, 오직 한 개의 키를 가지면서도 가변 길이의 메시지도 안전하게 취급할 수 있는 OMAC을 SNEP에 적용한 새로운 기법인 OMAC-SNEP을 제안하고, 실제 구현하고자 한다.

먼저 Ⅱ장에서는 무선 센서를 이용한 무인 경비 시스템 구성방안과 SNEP프로토콜을 기술하고, Ⅲ장에서는 CBC-MAC 관련연구를 살펴보며, Ⅳ장에서는 OMAC을 SNEP에 적용한 새로운 기법인 OMAC-SNEP을 기술하고, Ⅴ장에서는 OMAC-SNEP의 성능을 비교 분석하며, 끝으로 Ⅵ장에서는 결론을 기술하겠다.

Ⅱ. 무선 센서를 이용한 무인경비시스템

기존의 무인 경비 시스템은 보안이 취약한 지역에 여러 개의 센서를 설치하고, 센서에서 수집된 경보정보를 사용자단말기에서 취합한 후, 전기적 신호로 변환한 뒤, 원격지에 떨어져 있는 관제실로 전달하고, 관제실에서는 경비요원에게 출동을 지시하여 침입자를 검거 또는 퇴치하는 형식이다.

그러나 센서와 사용자 단말기사이에 유선을 이용하기 때문에 케이블 포설이 어렵고 미관상 좋지 않아

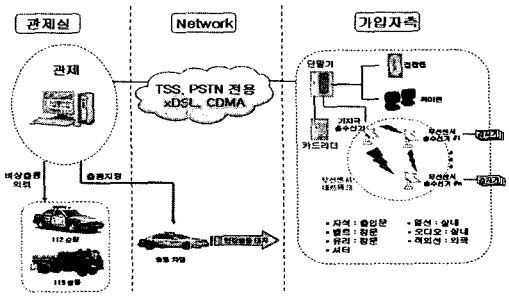

그림 1. 무인 경비 서비스 개략도

이를 무선센서 네트워크로 대체 할 필요가 있으며, 무선으로 대치할 경우의 배치도는 그림 1 과 같다.

본 논문에서는 무선 센서 네트워크에 한해서 제안하기로 한다.

분산된 센서 네트워크는 대개 계산 능력이 강력한 기지국(Base Station)과, 센서라고 불리는 많은 수의 낮은 용량의 노드(Node)들로 구성 된다.

감지기(센서노드)로부터 기지국으로 송신된 경보메시지가 정당한 송신자로부터 송신되었는지, 또 그 메시지가 전송 중에 변경되지 않았는지를 확인하는 인증, 무결성 그리고 기밀성 등의 보안서비스를 제공할 수 있는 능력이 센서노드에 있어야 한다.

1. 무선센서 네트워크에서의 보안취약성

무인 경비 시스템에서의 무선 센서 네트워크는 여러 개의 센서노드들로 구성되며, 통신 인프라와 연결하기 위한 하나의 기지국(Base Station)이 존재한다. 통신과 센싱 기능을 갖는 센서를 이용하므로, 기본적으로 전파식별에서 발생하는 모든 위협에 더하여 다음과 같은 위협들이 추가로 발생하게 된다.

- 센서 노드 장치의 도난 및 분실
- 노드간의 통신 정보 노출
- 센서노드에 의한 위조 정보 전송

이외에도 불법노드 설치, 서비스 거부 공격(Denial of Service : DoS), 배터리 소진 공격, IP스푸핑(Spoofing), 트로이 목마, 그리고 웜바이러스 등 많은 공격으로 인하여 무선 센서 네트워크를 이용한 무인 경비 시스템이 정상적인 서비스를 제공할 수 없게 만들 수 있다.

이러한 위협을 효율적으로 막기 위해 요구되는 보안 서비스는 기밀성, 인증 및 무결성, 그리고 신선성(Freshness) 서비스 등이 있다.

2. 센서 네트워크 보안프로토콜(SNEP: Secure Network Encryption Protocol)

센서 네트워크의 보안 요구사항을 만족하는 보안 프로토콜로 SNEP[4]이 있다. 이 프로토콜은 대칭형 암호 방식만을 사용하며, 무선 센서 네트워크와 같은 자원이 제한된 환경에서 보안 프로토콜을 제공할 때 자원의 오버헤드를 최소화하는 것을 목표로 한다.

센서 노드에서 기지국으로의 통신과 같은 유니캐스트 통신에서는 송수신자가 비교적 명확하기 때문에 두 통신 당사자가 서로 비밀을 공유하고, 각 패킷에 공유된 키로 계산한 메시지 인증 코드(Message Authentication Code : MAC)를 덧붙임으로써 비교적 충분한 보안을 제공할 수 있는데, 이와 같은 목적으로 사용되는 프로토콜이 SNEP이다. SNEP은 일반적으로 키 설정 단계, 암호화단계, MAC 생성단계 등으로 구성된다.

2.1 SNEP의 키 설정, 암호화 및 MAC 생성

2.1.1 키 설정 단계

SNEP은 일반적으로 RC5[10] 대칭 키 암호화 알고리즘을 이용하여 다양한 목적의 키 값을 유도한다. 마스터키(Master Key)는 기지국과 센서 노드 사이에서 사전에 나누어 가지며, 마스터키를 기반으로 암호화를 위한 암호화 키, MAC 값 생성을 위한 키 값, 랜덤 키 값 등을 생성한다.

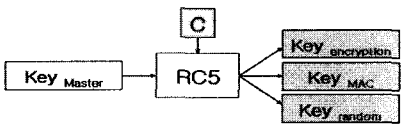

그림 2. 키 생성 메커니즘

2.1.2 암, 복호화 단계

암호화는 전 단계에서 생성한 암호키를 카운터 모드(Counter Mode)로 암호화하여 Chain으로 연결하는 구조이다.

E{(En_Key, Counter) P} = C를 이용해 암호화 루틴을 반복 수행하여 암호화 및 복호화를 수행한다.

2.1.3 MAC 생성단계

그리고 메시지인증코드(MAC) 생성키를 이용하여 그림 3과 같이 CBC 모드로 암호화된 메시지에 대한 메시지인증코드를 생성한다.

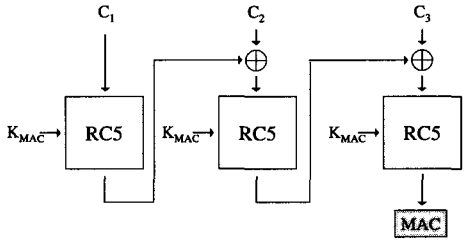

그림 3. SNEP의 MAC 생성메커니즘

2.2 SNEP 프로토콜의 보안서비스

SNEP 프로토콜은 다음과 같은 보안 서비스를 제공한다.

- 데이터 기밀성 : 데이터 교환 시 암호화를 통하여 데이터 기밀성을 제공한다.
- 확고한 보안성 : 공격자가 같은 키로 암호화된 평문-암호문 쌍을 알고 있다 하더라도 암호화된 메시지의 평문을 추출해 낼 수 없도록 한다.
- 양단간 데이터인증과 무결성 : MAC(메시지 인증코드)을 사용하여 양단간 데이터 인증과 무결성을 제공한다.
- 재사용 방지 : MAC에 카운터(Counter) 값을 포함하여 공격자에 의한 재사용 공격을 방지한다.
- 데이터 신선성 : 해당 데이터가 가장 최근의 버전임을 검증하기 위한 기능을 제공한다.
- 낮은 통신 부하 : 각 블록이 끝난 후 카운트를 하나씩 증가시켜서, 카운터를 메시지에 포함하지 않고 송신하므로, 낮은 통신 부하 및 기밀성을 제공한다.

Ⅲ. CBC-MAC

1. 개 요

CBC-MAC[2,3]은 블록암호 E를 사용해서 MAC을 만드는 알고리즘 중에서도 가장 잘 알려지고, 간단한 알고리즘이다.

메시지 인증은 수신된 메시지가 그 메시지를 송신했다고 주장하는 상대방으로부터 송신되었다는 것을, 비밀키를 나누어 갖은 통신 상대방이 검증할 수 있게 해 주는 기술이다. 메시지 인증은 가장 중요하면서도 폭 넓게 사용되는 암호 도구중의 하나로, "메

시지 인증 코드" 또는 MAC을 사용하여서 흔히 달성 된다.

인증되어야할 메시지 x를 비밀키 a로 계산한 것을 약어로 $MAC_a(x)$라 한다.

송신자는$(x,MAC_a(x))$를 송신하고,(x',σ')를 수신한 수신자는 $\sigma'=MAC_a(x')$라는 것을 증명하면 인증이 완료된다.

가장 흔히 사용하는 MAC은 블록암호의 기초로 "암호 블록 체이닝 : Cipher Block Chaining"을 사용한다. 그러나 CBC-MAC은 메시지의 길이가 가변일 경우 안전하지 못한 문제점이 있다[5]. 따라서 임의 길이의 메시지 또는 가변 길이의 입력도 처리할 수 있는 여러 가지 기법들이 제안되었다.

2. 관련연구

CBC-MAC의 문제점을 보완하기 위해 제안된 기법들로는 EMAC, XCBC, TMAC 그리고 OMAC 등이 있는데, 그 중에서 EMAC는 새로운 키 K_2를 갖는 E로 CBC-MAC(CBC_{K1}(M))을 다시 한 번 더 암호화하는 방식이다[6]. 즉, $EMAC_{K1,K2}(M)=E_{K2}(CBC_{K1}(M))$이다. 이때, M은 메시지, K_1은 CBC-MAC의 키 그리고 $CBC_{K1}(M)$은 M의 CBC-MAC값이다.

Petrank와 Rackoff는 메시지의 길이가 블록길이 n의 배수일 경우, EMAC가 안전하다는 것을 증명하였으나, EMAC는 블록암호 E에 2개의 키를 사용한다는 비효율성 문제점이 있다. 단, n은 블록암호 E[7]의 블록길이이다.

임의 길이의 메시지인 경우에는, 메시지의 길이가 n의 배수가 되게 하기 위하여, 마지막 암호화 단계의 메시지 M뒤에 1을 더하고, 뒤이어 i개의 0을 추가로 삽입하여(10^i를 Concatenate한다), 메시지의 길이가 블록길이 n이 되도록 하여야한다.

그 다음에 Black과 Rogaway는 위의 문제점[11]을 해결하기 위하여 XCBC를 제안하였는데, 이 XCBC는 3개의 키를 갖는다 : 즉, 한 개의 블록 암호 키 K_1과 두 개의 n비트 키 K_2와 K_3를 갖는다.

XCBC에서는 메시지의 사이즈가 n의 배수이면 10^i 를 첨가하지 않지만, n의 배수가 아닌 때에는, 최소한의 10^i을 첨가하고, 이들을 구별하기 위하여, 마지막 블록을 암호화하기 전에 K_2 또는 K_3로 XOR을 한다.

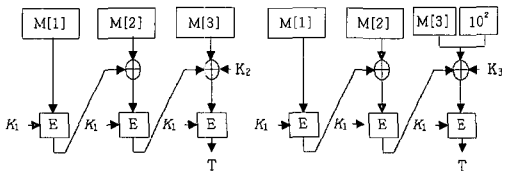

그림 4. XCBC의 도해

그러나 XCBC도 총 길이가 (k+2n)비트인 3개의 키가 사용된다는 비효율성 문제점이 있다.

XCBC는 그림 4와 같다.

Kurosawa와 Iwata는 2개의 키 K_1과 K_2를 요구하는 Two-key CBC-MAC(TMAC)[9]을 제안하였는데, TMAC는 총 길이가 (k+n)비트인 2개의 키를 사용한다. : 즉, 블록암호 키 K_1과 한 개의 n비트 키인 K_2를 사용하는데, TMAC는 XCBC의 (K_2, K_3)를 ($K_2 \cdot u$, K_2)로 대치하여 사용한다. 이때, u는 non-zero상수이고, "$\cdot$"는 $GF(2^n)$상에서의 곱셈을 나타낸다.

끝으로 Kurosawa와 Iwata는 블록암호가 오직 한 개인 키 K만을 요구하는 OMAC를 제안하였다.[13]

OMAC는 OMAC1과 OMAC2의 일반적인 이름으로, 임의의 상수(n개의 0비트)를 키 K로 암호화한 것을 L이라 가정하면 ($L=E_K(0^n)$), OMAC1은 XCBC의 (K_2, K_3)를 ($L \cdot u$, $L \cdot u^2$)로 대치하여 사용한다. 이와 비슷하게, OMAC2는 XCBC의 (K_2, K_3)를 ($L \cdot u$, $L \cdot u^{-1}$)로 대치하여 사용한다.

다음 표 1은 길이를 비교한 것이다. 단, K는 E의 키 길이를 말한다.

표 1. 키 길이 비교표

	XCBC[11]	TMAC[9]	OMAC[13]
키 길이	(k+2n)비트	(k+n)비트	k 비트

Ⅳ. 제안 기법(OMAC-SNEP : One-key CBC-MAC for Secure Network Encryption Protocol)

SNEP은 CBC-MAC을 적용하였기 때문에, 메시지의 길이가 가변일 경우 안전하지 못한 문제점이 있다. 따라서 본 논문에서는 SNEP에 OMAC를 적용시킨 새로운 기법인 OMAC-SNEP(One-key CBC-MAC for Secure Network Encryption Protocol)을 무선 센서 네트워크에 적용할 것을 제안하

고자 한다.

OMAC-SNEP(One-key CBC MAC)은 오직 한 개의 키를 가지면서도, CBC-MAC에서 문제가 되는 가변 길이의 어떠한 메시지에 대해서도 안전하게 취급할 수 있으며, 효율성과 보안성측면에서도 대단히 훌륭한 기법이다. NIST에서도 2005. 5. 18일에 권고사항인 800-38B[8] (블록암호운용 모드의 "인증을 위한 CMAC(Cipher Based MAC) 모드")를 최종적으로 특별 발간하였는데, 이 권고사항에 있는 CMAC는 OMAC(OMAC1)를 기초로 하여 작성한 것이므로, 머지 않아 ITU 등에서도 이를 기초로 한 권고사항이 발표 되리라 생각된다.[13]

1. OMAC

OMAC는 OMAC1과 OMAC2의 일반적인 이름으로, 오직 한 개의 키(K)만을 갖는데, 이 키는 블록 암호 E의 키 K와 동일한 것이다. 그 외에도 OMAC1은 XCBC의 키(K_2,K_3)를 ($L \cdot u$, $L \cdot u^2$)로 대치해서 사용한다. 이때, u는 GF(2^n)내에 있는 non-zero 상수이며, L은 $L=E_K(0^n)$이다. 그리고 OMAC2는 ($L \cdot u$, $L \cdot u^{-1}$)을 사용한다. $L \cdot u$, $L \cdot u^{-1}$와 $L \cdot u^2=(L \cdot u) \cdot u$를 만드는 방법은 L을 한번 변위(Shift)시키고, 조건부 XOR시켜서 $L \cdot u$를 만들며, 또한 L과 $L \cdot u$를 각각 한번 변위 시키고, 조건부 XOR시켜서 $L \cdot u^{-1}$과 $L \cdot u^2=(L \cdot u) \cdot u$를 쉽게 계산 할 수 있다.

OMAC는 만약, $|M|=mn$ 이라면, 마지막 블록을 암호하기 전에 $L \cdot u$를 XOR하고, $|M|=mn$ 이 아닌 경우에는, M에 10^i 패딩($i=n-1-|M| \bmod n$)이 첨가된다. 그리고 마지막 블록을 암호하기 전에 $L \cdot u^2$ (OMAC2인 경우, $L \cdot u^{-1}$)를 XOR하는 것을 제외하고는 CBC-MAC과 똑 같이 계산한다. 그림 5는 OMAC1을 나타낸 것이다.

TMAC에서는 K_2가 키의 일부분이지만, OMAC에서는 L이 키의 일부분이 아니며, K로부터 생산된다는 것에 주목해야 한다.

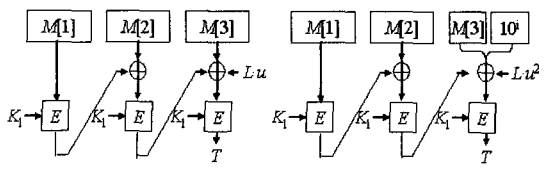

그림 5. OMAC1의 도해. 단, $L=E_k(0^n)$임.

그럼에도 불구하고, OMAC은 XCBC만큼 안전하다는 것이 입증되었다. 그 밖에도 OMAC는 XCBC(그리고 TMAC)가 갖고 있는 훌륭한 특성들을 모두 갖는다. 즉, OMAC의 영역은 $\{0,1\}^*$ 인데, OMAC는 블록암호용으로 1개의 키와 최대 $\{1, \lceil |M|/n \rceil\}$ 개의 블록암호만을 시행한다.

2. OMAC-SNEP

OMAC-SNEP은 SNEP과 유사하다. 다만 CBC-MAC을 OMAC으로 대치한 것이므로, OMAC-SNEP을 이용한 프로토콜은 다음과 같다.

2.1 OMAC-SNEP의 키설정, 암호화 및 MAC생성

2.1.1 키설정, 암복호화 단계

키 설정 및 암복호화 단계는 II장의 SNEP생성 과정과 같으므로, 중복 기재를 생략한다. 단, 키 생성 메커니즘은 그림 2를 참조하기 바란다.

2.1.2 OMAC-SNEP의 MAC생성단계

OMAC메시지 인증코드(OMAC-MAC) 생성키를 이용하여 그림 6과 같이 OMAC모드로 메시지 인증코드를 생성한다.

2.2 데이터 인증만 보장하는 경우

노드 A에서 기지국 B로 데이터 D와 메시지인증코드인 $MAC_{OMAC}(K_{OMAC}, C|D)$를 송신한다.

$A \rightarrow B : (D), MAC_{OMAC}(K_{OMAC}, C|D)$

(단, D는 데이터, C는 카운터이며, K_{OMAC}키는 마스터키에서 추출한다)

2.3 데이터 인증및 기밀성보장의 경우

데이터 인증과 기밀성을 보장하는 경우에는 노드 A에서 기지국 B로 암호화된 데이터 $E=\{D\}(K_{encr},c)$와 이에 대한 메시지인증코드로 $MAC_{OMAC}(K_{OMAC}, C|E)$을 송신하여 데이터 인증 및 기밀성을 제공한다 (단, D는 데이터, K_{encr}는 암호키, C는 카운터로 초

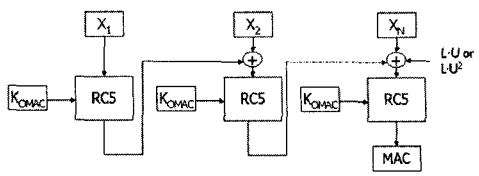

그림 6. OMAC-SNEP의 MAC 생성메커니즘

기 백터 값(Initial Vector), K_{encr} 키와 K_{OMAC}키는 마스터 비밀키 K로부터 추출된다).

따라서 노드 A가 기지국 B로 보내는 완전한 메시지는 다음과 같다.

A → B

:{D}(K_{encr},c),MAC_{OMAC}(K_{OMAC},C|{D}(K_{encr},c))

2.4 Nonce를 사용한 강한 신선성

노드 A가 난수와 함께 질문 한 것에 대하여, 노드 B가 응답하여, 강한 신선성을 제공하는 SNEP의 전체 프로토콜은 다음과 같다.

A → B : Na, Ra

B → A : {R_b}(K_{encr},c),MAC_{OMAC} (K_{OMAC}, N_b|C|{R_b} (K_{encr},c))

단, N_a는 A의 난수로 랜덤하게 생성되며, R_a는 A의 질문, R_b는 B의 응답이다.

만약 MAC이 정확하게 검증된다면 노드 A는 자기가 질문을 송신하고 난 후에, 노드 B가 이에 대한 응답을 생산한 것이라고 인정하여, 강한 신선성이 보장된다.

V. 성능 분석

이 장에서는 CBC-MAC의 문제점을 개선한 기법(EMAC, XCBC, TMAC 그리고 OMAC)들 간의 성능을 보안성 및 효율성 측면에서 비교 검토한다. CBC-MAC은 가변입력에 대한 문제점으로 인해 비교의 대상이 되지 않으나, 참고를 위해 기재하였다.

1. OMAC의 보안성 비교

보안성 비교에서는 다음과 같은 기호를 사용한다.

- $Adv_F^{mac}(t,q,nm)$은 최대위조확률을 말한다. 최대위조확률이란 적 A가 MAC키를 구별해 낸 후, 그 키로 새로운 메시지(M,F_k(M))을 출력(위조)해 낼 수 있는 최대 확률을 말하고,
- $Adv_E^{prp}(t', q', nm)$은 블록암호 E와 산발적으로 선택된 순열간을 구별하는 최대변별확률이다. 즉, 블록암호E의 암호키를 알아낼 수 있는 확률을 말한다. 단, F는 XCBC, TMAC, OMAC이고, prp는 순열, E는 블록암호, 그리고 최대치(mac)란 기껏해야 t'시간동안 동작하고, 겨우 q' 개의 질문을 하는 모든 적을 말한다.
- m은 태그의 길이를 나타낸다.

표 2는 기법별 보안 한계치를 나타낸 것인데, EMAC, XCBC, TMAC 그리고 OMAC사이에 큰 차이가 없는 것을 볼 수 있다. [12]에서도 XCBC (그리고 TMAC와 OMAC포함)가 EMAC만큼 안전하다는 것을 증명하고 있으므로, 이들 네 개의 기법들은 보안성 측면에서 모두 안전한 것으로 볼 수 있다.

표 2. 보안성 비교표

Frame	Security Bound
CBC-MAC	$Adv_{CBC^m-F}^{mac}(t,q) \le \frac{2q^2m^2+1}{2^n} + Adv_E^{prp}(t',q')$, where, $t' = t+O(mqn)$ and $q' = mq$.
EMAC	$Adv_{EMAC}^{mac}(a,\sigma) \le \frac{3q^2+1}{2^n} + Adv_E^{prp}(a) + Adv_E^{prp}(a)$, where, σ is the total block length of all queries.
XCBC	$Adv_{XCBC}^{mac}(t,q,nm) \le \frac{(4m^2+1)q^2+1}{2^n} + 3Adv_E^{prp}(t',q')$, where, $t' = t+O(mq)$ and $q' = mq$.
TMAC	$Adv_{TMAC}^{mac}(t,q,nm) \le \frac{(3m^2+1)q^2+1}{2^n} + Adv_E^{prp}(t',q')$, where, $t' = t+O(mq)$ and $q' = mq$.
OMAC	$Adv_{OMAC}^{mac}(t,q,nm) \le \frac{(5m^2+1)q^2+1}{2^n} + Adv_E^{prp}(t',q')$, where, $t' = t+O(mq)$ and $q' = mq+1$

2. OMAC의 효율성 비교

효율성비교에서는 다음과 같은 기호를 사용한다.

- "Domain"은 입력영역을 나타내고,
- "K len."은 키 길이를 나타내며,
- "#K sche."은 블록암호키 스케줄링의 수
- "#M" 은 송신자가 MAC을 한 메시지의 수를 나타낸다.
- "#E invo."는 메시지 M의 태그를 생산하기 위한 블록암호실행(invocation) 수를 나타내며, 단, |M|>0이다.
- "#E pre."는 사전 처리 시간동안에 시행한 블록암호 실행 수를 나타낸다.
- "+kst" 는 키 분리 기술이 사용된다는 뜻이다. OMAC에서는 키 분리 기술이 필요 없는데, 그 이유는 OMAC의 키 길이가 적합하기 때문이다.

표 3. 효율성 비교표

종류	Domain	*K* len.	*#K* sche.	*#E* invo.	*#E* pre.
CBC MAC	$(\{0,1\}^n)^m$	k	1	$\lceil M \rceil / n$	0
EMAC	$(\{0,1\}^n)^+$	$2k$	2	$1+\lceil M \rceil / n$	0
RMAC	$\{0,1\}^+$	$2k$	$1 + \#M$	$1+\lceil (\lceil M \rceil+1)/n \rceil$	0
XCBC	$\{0,1\}^+$	$k+2n$	1	$\lceil \lceil M \rceil / n \rceil$	0
TMAC	$\{0,1\}^+$	$k+n$	1	$\lceil \lceil M \rceil / n \rceil$	0
XCBC+kst	$\{0,1\}^+$	k	2	$\lceil \lceil M \rceil / n \rceil$	3 or 4
TMAC+kst	$\{0,1\}^+$	k	2	$\lceil \lceil M \rceil / n \rceil$	2 or 3
OMAC	$\{0,1\}^+$	k	1	$\lceil \lceil M \rceil / n \rceil$	1

표 3에 의하면 OMAC가 키 길이, 키 스케줄링 수 그리고 블록암호 실행 수 등의 효율성 비교에서, 제안된 기법(EMAC, XCBC, TMAC OMAC) 중에서는 가장 우수한 성능인 것을 알 수 있다[14]. 세부 내역을 살펴보면,

- **키 길이** : OMAC는 키 길이가 적당하면서도, EMAC, XCBC 그리고 TMAC와 같은 정도의 안전성을 유지한다. RMAC, EMAC, XCBC 그리고 TMAC는 키 분리 기술을 사용하면, 키 길이를 줄일 수 있으나, 사전 처리과정에서 키 스케줄링 수 와 블록암호 실행 수가 늘어나는 문제가 발생한다.
- **키 스케줄링 수** : XCBC, TMAC 그리고 OMAC는 가장 좋은 성능을 유지하는 반면에, EMAC는 두 개, 그리고 RMAC는 태그를 생산 할 때마다 한 개의 블록암호 키 스케줄링을 추가로 요구하여 더 많은 비용이 소요되는 문제점이 있다.
- **블록암호 실행 수** : RMAC와 EMAC는 XCBC, TMAC 그리고 OMAC와 비교해서 한 개 또는 두 개의 추가적인 블록암호 실행을 요구하고 있어, 짧은 메시지를 자주 암호화하는 실제 환경에서는 비효율적이다.

OMAC는 사전 처리과정에서 한 개의 블록암호를 실행하여야 하는 문제점이 있다. 그러나 블록암호 실행은 사전 쉬는 시간에 이루어지며, MAC생산에 비해서 비 주기적으로 실행되므로 그렇게 중요한 사항은 아닌 반면에, OMAC은 키 분리 기술을 사용하지 않아도 되는 장점이 있다. 이 키 분리 기술은 실제 환경에서 가장 에러가 많이 발생하는 처리과정 이므로, XCBC, TMAC와 같이 키 분리 기술을 사용하는 다른 제안된 기법들은 OMAC에 비해서 오히려 더 큰 키 설정 비용을 지불해야 하는 문제가 발생한다.

결과적으로 OMAC는 한 개의 키를 가지면서도 CBC-MAC에서 문제가 되는 임의 길이의 입력 또는 가변 길이의 입력에도 잘 적응하며, EMAC, XCBC 그리고 TMAC등과 같은 타 기법에 비해서도 효율성이나, 보안성 측면에서도 결코 뒤지지 않는 동등한 성능을 발휘하는 매우 훌륭한 기법인 것을 알 수 있다.

3. OMAC-SNEP성능비교 (측정결과)

OMAC-SNEP을 실제 구현하고, 측정한 결과(성능)는 다음과 같다.

○ 테스트 환경

기 종	Pentium Ⅳ
CPU	노스우드 2.8GHz H.T지원
메모리	2GByte
운영체제	Windows XP Pro SP2

○ 라인수

OpenSSL 보안 API를 이용한 RC5 암호 알고리즘을 제외한 소스코드의 라인수를 나타낸 것이다.

모듈 이름	라인수
main.c	874 = 874
CryptoUtil.c + H	97 + 42 = 139
OMAC + H	74 + 20 = 94
SNEP + H	105 + 28 = 133
OMAC-SNEP	1240

○ 수행시간

알고리즘	적용 암호 알고리즘	수행시간(1bit)
CBC-MAC	RC5	0.131㎲
CBC-MAC	AES	0.078㎲
OMAC	RC5	0.154㎲
OMAC	AES	0.089㎲

시뮬레이션에서 얻은 결과를 분석하면, SNEP(CBC-MAC)보다 OMAC-SNEP의 수행 속도가 약간 저하되는 현상을 볼 수 있다. 이는 OMAC의 마지막 라운드에서 CBC-MAC의 문제점을 보완하기 위해서 $L \cdot u$또는 $L \cdot u^2$으로 XOR을 적용하기 때문인 것으로 판단된다. 그러나 가변 길이에서의 안전성을 감안한다면, 현재 사용하고 있는 고속 시스템에서의 적용은 무리가 없음을 알 수 있다. 또한 적용 알고리즘에서 AES가 RC5보다 처리 속도측면에서는 0.053㎲(약42%)정도가 빠르게 수행되는 것을 볼 수 있다. 만일 적용 시스템의 서버와 클라이언트가 일반 PC를 이용한다면, AES를 이용하면 훨씬 좋은 성능을 얻을 수 있을 것으로 생각된다. 그러나 본 논문에서는 센서 네트워크 시스템에 적용하여 제안한 방식이므로 메모리 점유율, 처리 연산 수 그리고 이를 감안한 시스템 부하를 생각한다면 RC5 암호 알고리즘의 적용이 알맞을 것으로 판단된다.

· 암호화 단계 및 블록길이 변화에 따른 비교

다음은 SNEP(CBC-MAC)과 OMAC-SNEP

의 암호화 단계 수를 변화시키고, 또한 암호 블록의 길이를 변화 시키면서, 1 비트를 기준으로 속도를 비교 측정하였다. 아래 표에서 보는바와 같이 암호화 단계수와 블록 길이가 커지면 커질수록 두 방법간의 시간 차이가 줄어드는 것을 볼 수 있다. 이것 또한 마지막 라운드에서 한번의 XOR을 하기 때문인 것으로 판단된다.

표 4. SNEP(CBC-MAC)과 OMAC-SNEP간의 속도비교

알고리즘	암호화 단계 수 (1비트)			블록 길이 (8bit × 블록수 × 12R)		
	12R	24R	36R	1B (8bit)	1KB (8Kbits)	1MB (8Mbits)
SNEP	0.131 μs	0.236 μs	0.358 μs	0.131μs	9.695μs	9.354ms
OMAC -SNEP	0.154 μs	0.274 μs	0.406 μs	0.154μs	11.396μs	10.346ms

· 해쉬함수와 레지스터사용시의 $L \cdot u$ 및 $L \cdot u^2$ 생성 속도비교

$L \cdot u$ 및 $L \cdot u^2$을 생성하기 위해서 사용하는 보편적 해쉬 함수와 쉬프트 레지스터를 이용한 키 생성과정에서는 단연 쉬프트 레지스터의 속도가 현저히 빠른 것으로 측정 되었다.

표 5. 해쉬 알고리즘과 쉬프트 레지스터의 속도비교

해쉬 알고리즘	처리 비트 수	처리 시간	처리 속도
MD5	$760b \times 10^7$	24 s	316.7Mb/s
SHA-1	$760b \times 10^7$	36.9 s	180.3Mb/s
쉬프트 레지스터	$760b \times 10^7$	16 s	475Mb/s

결론적으로 OMAC-SNEP이 SNEP (CBC-MAC)의 동작속도에 비해 평균 15%정도 시간이 더 소요되는 것으로 나타났으나, 라운드와 블록의 수가 증가함에 따라 그 차이가 줄어드는 것을 알 수 있다. 그러나 이러한 속도에서의 단점에도 불구하고, OMAC-SNEP의 목적은 가변 길이의 입력에서도 안전한 동작을 보증하는 것이므로 상기와 같은 단점은 충분히 극복 될 수 있을 것이라 판단된다.

Ⅵ. 결 론

이 논문에서는 무선 센서 네트워크를 이용한 무인 경비 시스템에서의 정보 보안상 취약점을 도출하고, 도출된 취약점을 해결하기 위한 방법으로 현재까지 발표된 센서 네트워크 보안 프로토콜 중에서, 데이터 기밀성, 양단간 데이터 인증, 무결성 그리고 신선성을 제공하는SNEP을 적용 할 것을 제안하였다.

SNEP은 빠른 대칭형 암호 알고리즘을 기초로 하고, 비대칭형 알고리즘은 사용하지 않기 때문에, 폭 넓고도 다양한 장비에서 적용할 수 있는 빌딩블록이다. 그러나 SNEP은 CBC-MAC을 적용하기 때문에, 메시지의 길이가 고정되지 않으면 보안상 안전하지 못한 문제점이 있다. 따라서 본 논문에서는 오직 한 개의 키 만을 사용하면서도, 임의의 길이 메시지도 안전하게 취급할 수 있는 OMAC를 SNEP에 적용시킨 OMAC-SNEP을 새롭게 제안하고, 구현함으로서, 효율적이면서도 보안상 안전한 무선 센서 네트워크를 구축하고자 하였다.

구현된 OMAC-SNEP은 시뮬레이션에서 얻은 결과와 같이, OMAC-SNEP이 SNEP(CBC-MAC)에서 보다 약0.023μs(약15%)정도의 속도 저하 현상이 있는 것을 볼 수 있는데, 이는 OMAC-SNEP의 마지막 라운드에서 SNEP(CBC-MAC)의 문제점을 보완하기 위해서 $L \cdot u$또는 $L \cdot u^2$으로 XOR을 적용하기 때문인 것으로 판단된다. 그러나 이러한 단점은 가변 길이에서의 안전성을 감안한다면, 현재 사용하고 있는 고속 시스템에서의 적용은 별 무리가 없을 것으로 판단된다. 결국 OMAC-SNEP은 한 개의 키를 사용하면서도 SNEP(CBC-MAC)의 문제점을 완전히 해결한 것은 물론이고, EMAC, XCBC 그리고 TMAC등과 같은 타 기법에 비해서도 효율성이나, 보안성 측면에서 결코 뒤지지 않는, 동등한 성능을 발휘하는 매우 훌륭한 기법임을 알 수 있을 것이다.

따라서, 이제까지 유선 센서에 의존을 했거나, 정보보호를 고려하지 않은 무선 센서 네트워크를 이용한 무인 경비 시스템뿐만 아니라, 기타의 무선 센서 네트워크에도 OMAC-SNEP을 사용 할 수 있어 보다 안전하고, 효율적인 시스템을 구축하는데 많은 기여를 하리라 생각한다.

그러나 아직 전력 소모로 인해 배터리를 갈아 주어야하는 번거로움이 있기 때문에 좀 더 전력을 적게 소모하면서도, 속도가 빠른 USN(Ubiquitous Sensor Network)환경에 적합한 프로토콜이 더 개발되어야 할 것으로 생각한다.

참 고 문 헌

[1] Mark Weiser. The Computer for the 21 Centry. Scientific American. Vol. 256. No.3. pp. 94-104. Sep. 1991

[2] FTPS 113. Computer data authentication. Federal Information Processing Standards Publication 113, U.S〉 Department of Commerce/National Bureau of Standards, National Technical Information Service, Springfield, Virginia, 1994.

[3] ISO/IEC 9797-1. Information Technology-security techniques-data integrity mechanism using a cryptographic check function employing a block cipher algorithm. International Organization for Standards, Geneva, Switzerland. 1999. Second edition.

[4] A. Perrig, R. Szewczyk, J.D.Tygar, Victorwen D. E. Culler : SPINS : "Security Protocols for Sensor Networks, Wireless Networks" 8, 521. 534, 2002

[5] M. Bellare, J. Killian and P. Rogaway. The security of the cipher block chaining message authentication code. JCSS, Vol. 61, no. 3, 2000. Earlier version in Advances in Cryptology-CRYPTO '94. LNCS 839, pp.341-358. Springer-Verlag, 1994.

[6] A. Berendschot, B.den Boer, J.P.Boly, A.Bosselaers, J.Brandt, D.Chaum, L.Damgard, M.Dichtl, W.Fumy, M. van der Ham, C.J.A.Jansen, P.Landrock, B.Preneel, G.Roelofsen, P.de Rooij and J.Vandewalle. Final Report of RACE Integrity Primitives. LNCS 1007, Springer-Verlag, 1995.

[7] E. Petrank and C.Rackoff. CBC-MAC for real-time data sources. J.Cryptology, Vol.13, no.3, pp. 315-338, Springer-Verlag, 2000

[8] M.Dworkin. Recommendation for block cipher modes of operation : The CMAC mode for authentication. NIST special publication 800-38B, Available at http://csrc.nist.gov/publications/nist pubs/800-38B/SP-800-38B.pdf.

[9] K. Kurosawa and T.Iwata. TMAC : Two-key CBC-MAC. Topics in Cryptology-CT-RSA 2003, LNCS 2612, pp.33-49, Springer-Verlag, 2003

[10] R.L. Rivest. The RC-5 Encryption algorithm. Proc. 1st Workshop on Fast Software Encryption, Pages 86-96,1995

[11] J. Black and P.Rogaway. CBC-MAC's for arbitrary-length messages : The three key constructions. Advances in Cryptology-CRYPTO 2000, LNCS 1880, pp.197-215, Springer-Verlag, 2000.

[12] T.Iwata and K.Kurosawa. Stronger security bounds for OMAC, TMAC and XCBC. Manuscript. Available at Cryptology ePrint Archive, Report 2003/082, http://eprint.iacr.org/

[13] T.Iwata and K.Kurosawa. OMAC : One-key CBC-MAC. Pre-proceedings of Fast Software Encryption, FSE 2003, pp.137-161, 2003. To appear in LNCS, Springer-Verlag.

[14] T.Iwata and K.Kurosawa. OMAC : One-keyCBC-MAC.Available at http : //crypt.cis.ibaraki.ac.jp/omac.html

〈著者紹介〉

이 성 재 (Seong-Jae Lee)
1989년 8월 한양대학교 산업대학원 전자통신과(석사)
2006년 2월 순천향대학교 대학원 정보보호학과 졸업(박사)
2000년 1월~2004년 3월 (주)KT 임원
2004년 3월~현재 (주)KT링커스 감사
2006년 1월~현재 한국정보보호학회 부회장
관심분야 : 네트워크보안, USN보안, 공개키 기반구조, 이동통신보안
E-mail : seongjlee@paran.com

김 학 범 (Hak-Beom Kim)
1990년 8월 : 중앙대학교 대학원 컴퓨터공학과 졸업(석사)
2001년 2월 : 아주대학교 대학원 컴퓨터공학과 졸업(박사)
1991년 10월~1996년 6월 : 한국전산원 주임연구원
1996년 7월~2001년 8월 : 한국정보보호진흥원 기술표준팀장
2001년 9월~2003년 1월 (주)드림시큐리티 상무이사
2003년 2월~2005년 3월 (주)장미디어인터렉티브 상무이사
2005년 4월~현재 정보보호연구소 부소장
2001년 3월~현재 순천향대학교 공과대학 정보보호 학과 겸임교수
관심분야 : 네트워크보안, 공개키 기반구조, 컴퓨터 보안, 정보보호 표준화
E-mail : khb0305@paran.com

염 흥 열 (Heung-Youl Youm)
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 교수
1997년 3월~2000년 3월 : 순천향대학교 산업기술연구소 소장
2000년 4월~현재 : 순천향대학교 산학연컨소시엄센터 소장
1997년 3월~현재 : 한국정보보호학회 총무이사, 학술이사, 교육이사
2004년 1월~현재 : 한국인터넷정보학회 이사, 논문지 편집위원
2003년 9월~2004년 3월 ITU-T SG17/Q10, Associate Rapporteur
2004년 3월~현재 : ITU-T SG17/Q9 Rapporteur
관심분야 : 네트워크보안, 전자상거래보안, 공개키 기반구조, 부호이론, 이동통신보안
E-mail : hyyoum@sch.ac.kr