Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.1, Feb. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.1.29

SDWSN 환경의 PUF 기반 그룹 키 분배 방법 개선*

오 정 민,[1†] 정 익 래,[1] 변 진 욱[2‡]
[1]고려대학교 정보보호대학원, [2]평택대학교

An Enhanced Scheme of PUF-Assisted Group Key Distribution in SDWSN*

Jeong Min Oh,[1†] Ik Rae Jeong,[1] Jin Wook Byun[2‡]
[1]Graduate School of Information Security, Korea University, [2]Pyeongtaek University

요 약

최근, IoT 무선 디바이스 등의 증가로 WSN(Wireless Sensor Network) 환경에서 네트워크 트래픽이 증가하면서 네트워크 자원을 안전하고 효율적으로 관리하는 SDN(Software-Defined Networking)을 WSN에 적용한 SDWSN(Software-Defined Wireless Sensor Networking)과 그에 대한 보안 기술에 대한 관심도가 증가하고 있다. 본 논문에서는 SDWSN 환경에서 PUF(Physical Unclonable Function) 기반 그룹 키 분배 방법을 안전하고 효율적으로 설계하는 방법을 서술한다. 최근에 Huang 등은 그룹 키 분배에 SDN의 장점과 PUF의 물리적 보안 기능을 이용하여 그룹 키 분배 방법을 설계하였다. 하지만, 본 논문에서는 Huang 등의 프로토콜이 보조 제어부 미인증과 불필요한 동기화 정보를 유지하는 취약점이 존재함을 발견하였다. 본 논문에서는 보조 제어부에 인증과정을 안전하게 설계하고, 불필요한 동기화 정보는 삭제하되 카운터 스트링과 랜덤 정보를 추가하여 Huang의 취약점을 개선하였다.

ABSTRACT

In recent years, as the network traffic in the WSN(Wireless Sensor Network) has been increased by the growing number of IoT wireless devices, SDWSN(Software-Defined Wireless Sensor Network) and its security that aims a secure SDN(Software-Defined Networking) for efficiently managing network resources in WSN have received much attention. In this paper, we study on how to efficiently and securely design a PUF(Physical Unclonable Function)-assisted group key distribution scheme for the SDWSN environment. Recently, Huang et al. have designed a group key distribution scheme using the strengths of SDN and the physical security features of PUF. However, we observe that Huang et al.'s scheme has weak points that it does not only lack of authentication for the auxiliary controller but also it maintains the redundant synchronization information. In this paper, we securely design an authentication process of the auxiliary controller and improve the vulnerabilities of Huang et al.'s scheme by adding counter strings and random information but deleting the redundant synchronization information.

Keywords: Wireless sensor networks, software-defined networking, key distribution, physical unclonable functions, Fuzzy extractor

Received(12. 20. 2018), Modified(02. 01. 2019), Accepted(02. 12. 2019)

* 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임. (No. 2017R1D1A1B03032424)

† 주저자, jeong-min@korea.ac.kr
‡ 교신저자, jwbyun@ptu.ac.kr(Corresponding author)

I. 서 론

다양한 센싱 기술의 발달과 이를 탑재한 IoT(Internet of Things) 무선 디바이스의 개수가 급격히 증가하면서 WSN(Wireless Sensor Network) 환경에서 네트워크 자원을 효율적으로 관리하는 문제가 큰 이슈로 부각되고 있다. 일반 유선 네트워크에서도 인터넷 사용자 증가, 온라인 게임, 스트리밍 서비스의 급격한 활성화로 인해 전세계 IP 트래픽이 제타바이트에 돌입하면서 네트워크 자원의 효율적인 관리 문제가 끊임없이 발생하고 있으며 이 문제는 앞으로 더욱 이슈화될 전망이다.

일반 유선 네트워크는 네트워크 자원 문제를 해결하기 위해 네트워크 트래픽에 따른 데이터의 신속한 경로 변경과 효율적인 전송을 통제할 수 있는 SDN (Software Defined Networking) 기술을 사용하였다. SDN은 크게 제어부(control plane)와 전송부(data plane)로 구성된다. 제어부는 통합 애플리케이션을 통해 네트워크 경로 설정 및 제어를 담당하고, 전송부는 데이터를 전송하는 부분이다. 기존의 네트워크 장비는 이러한 제어부와 전송부를 한 장비에 모두 포함해서 새로운 애플리케이션 출시에 따른 유연성, 이식성, 효율적인 네트워크 전송 제어 및 관리가 어려웠다. 이와 다르게, SDN에서는 제어부와 전송부를 분리하여 이질적인 네트워크 장비 및 자원에 대해 중앙 집중식으로 통합 관리할 수 있다. 이로 인해, 데이터의 효율적인 전송과 더불어, 클라우드 서비스와 같은 스토리지 리소스를 함께 사용할 수 있는 자원 관리의 효율성을 지닌다. 이러한 유선 네트워크 기반의 SDN의 장점을 무선센서네트워크 기반으로 확장한 것이 SDWSN(Software-Defined Wireless Sensor Networking) 기술이다. 무선은 유선에 비해 채널 자체가 지니는 근본적인 취약점이 (예:전파방해, 재밍, 브로드캐스팅) 많으므로, SDN에서 SDWSN으로 확장 시 다양한 무선 환경에서의 보안 위협을 중요하게 고려해야 한다. 이러한 SDWSN 환경에서의 보안위협을 해결하기 위한 논의는 기존 유선 네트워크 기반에서의 해결책 중심으로 진행되고 있으나 SDWSN의 최대 장점이 될 수 있는 성능 및 비용 개선이 제대로 향상되지 않는 경우가 많다. 또한, 오직 성능만 중시해서 SDWSN의 보안 취약점이 개선되지 않거나 처음부터 전혀 고려하지 않는 경우도 있다. 그래서 아직도 SDWSN 환경에서 뚜렷한 보안 가이드라인은 존재하지 않으며, 보안과 성능 사이를 조율하여 현실적인 방안을 찾기 위해 다양한 방법들이 현재 논의되는 실정이다.

SDWSN의 보안 연구는 대부분 SDN의 제어부에 대한 보안 연구가 주를 이룬다. SDN 구조는 중앙 집중식 구조로 네트워크 설정 및 제어 기능이 제어부에 집중되어 있다. 이 구조는 SDWSN이 전체 네트워크의 패킷이나 상태를 기반으로 비정상적인 트래픽 탐지하고 처리하는 보안기능, 무선 센서 노드의 기능 경량화 등을 제공한다. 하지만 기능이 많은 만큼 네트워크에서 다수의 무선 센서 노드 운영간 병목현상을 초래한다[1][2]. 또한, 제어부를 공격지점으로 삼아 데이터 도청, 데이터 위·변조, 제어부 점유. DoS 공격 등 다양한 공격을 할 수 있다[3][4][5][6]. 일반 유선 네트워크에서는 인증 및 권한 부여를 위해 보안 TCP 프로토콜을 사용하거나 TLS 기반의 보안 채널을 사용하지만 무선 세선 노드에서 똑같은 SSL/TLS 프로토콜을 구현하는 것은 노드에 부담이 된다[7][8]. 또한, 제어부가 점유되더라도 네트워크에 악영향을 주지 못하도록 애플리케이션 계층과 보안 커널을 정의하면 보안 응용 프로그램과 인터페이스에서 보안정책을 위반하지 할 수 없지만 제어부 구성의 복잡성이 증가하게 된다[7][8]. 네트워크 객체를 향한 DoS 공격은 정책 파일의 항목과 일치하지 않는 트래픽은 삭제해서 막을 수 있지만 지속적인 모니터링에 의해 네트워크 성능이 저하될 수 있다[10].

1.1 연구 동기

본 논문에서는 SDWSN 환경에서 그룹 키를 효율적으로 분배하는 방법을 제안하고자 한다. 무선 프로토콜의 장점은 그룹으로 쉽게 브로드캐스팅이 가능하다는 점으로 무선 센서 환경인 SDWSN에서도 그룹키 분배 방법에 대한 연구는 반드시 필요하다. 또한, 노드 노출에 강건하게 설계하기 위해 PUF(Physically Unclonable Function)를 적용하였다. 사실, 전통적인 센서 네트워크 환경에서 안전한 그룹 키 분배에 대한 연구는 이미 활발히 진행되었다. 하지만, PUF 기반의 SDWSN 환경에서의 그룹 키 분배 연구는 상대적으로 연구 초기 단계에 있다. PUF는 일반적으로 디바이스와 결합하여(Embedded) 구현되며, 하드웨어 제조 시 만들어진 하드웨어의 성질에 따라 입력에 대한 랜덤 출력 값을 생성하는 복제 불가능한 성질을 가지고 있다. 사용자의 개

인키 x가 노출되더라도 디바이스에 결합된 PUF를 복제하지 못하는 한 PUF의 출력값 y를 누구도 알 수 없으므로, 암호학적으로 키 분배 및 생성에 활발히 활용되고 있다.

아주 최근에, Huang은 그룹 키 분배에 PUF를 적용하여 SDN과 PUF의 장점을 동시에 고려한 프로토콜을 SDWSN 환경에서 제안하였다[11]. 제안된 프로토콜은 제어부를 주 제어부(Main controller)와 보조 제어부(Auxiliary controller)로 나누고, 전송부에 속한 네트워크 노드에 대한 그룹 키 분배는 주 제어부가 보조 제어부 없이 직접 분배하거나 보조 제어부를 통해 분배한다. 이러한 Huang의 프로토콜은 보조 제어부를 통해 그룹 키를 분배할 때, 보조 제어부와 센서 노드 사이의 인증을 수행하지 않는 취약점과 그룹 키 분배시마다 동기화 정보를 유지하고 업데이트하는 비효율적인 과정을 가지고 있다. 보조 제어부를 인증하지 않으면 주 제어부와 함께 다수의 보조 제어부, 노드로 구성된 SDWSN 환경에서 서비스 거부(Denial of Service) 공격에 취약하게 된다. 공격자는 보조 제어부 위치를 식별하거나 포획하기 쉬운 노드의 메시지 전송 기능을 획득하여 무차별적인 메시지를 보조 제어부를 향해 전송해서 주 제어부에게 서비스 거부 공격을 할 수 있다. 또한, 재생 공격을 해결하기 위한 동기화 정보를 유지하고 업데이트하는 과정은 지속적인 자원의 소비뿐만 아니라 공격자가 동기화를 강제적으로 막고 프로토콜을 실패시킨다면, 다음 인증 과정이 정상적으로 이루어지지 않는 결함이 있다.

본 논문은 이러한 Huang의 인증 환경 및 동기화 과정에 주목하여 안전성과 효율성 측면에서 개선된 프로토콜을 설계하였다. 먼저, 안전성 측면에서는 주 제어부와 보조 제어부, 노드 간의 각각의 상호 인증이 되도록 하되 그룹 키를 안전하게 노드에게 분배하는 삼자간 환경에서의 PUF 기반 인증 모델을 설계하였다. 또한, 효율적 측면에서는 동기화 정보를 유지하지 않고 카운터 스트링과 랜덤 정보를 통해 인증을 진행하도록 설계하였다.

본 논문의 구성은 2장에서 논문에서 사용하는 가정 및 기호를 정의하고 3장에서 Huang의 프로토콜의 키 분배 과정, 4장에서 제안하는 프로토콜의 키 분배 과정을 설명한다. 5, 6장에서 두 프로토콜의 안전성과 보안, 성능을 분석한 후 7장에서 그 결과에 대해 논의한다.

II. 가정 및 기호

이 장에서는 본 논문의 근간이 되는 PUF 및 퍼지 추출기(fuzzy extractor)에 대해 정의한다.

2.1 PUF

PUF는 하드웨어 제조 공정에서 발생하는 미세한 차이로 인해 같은 회로를 제조하더라도 동일한 회로가 제조될 수 없는 특성을 이용하는 기술로 각 PUF마다 랜덤하면서도 고유한 출력값을 생성할 수 있다. 이렇게 생성된 출력값은 홍채나 지문과 같은 생체정보처럼 고유하다는 특징을 가지고 있기 때문에 인증을 비롯한 다양한 분야에서 활용될 수 있다.

일반적으로 이상적인 PUF는 어떠한 환경에서도 입력값이 같으면 항상 동일한 출력값을 가지는 것으로 가정되나 실제 환경에서는 온도 등에 의해 발생한 미세한 잡음이 출력값에 영향을 주게 된다. 특히, 무선 노드는 특성상 유선 노드에 비해 출력값이 다양한 환경 변수에 노출되어 영향을 받을 가능성이 더 높다.

Huang의 프로토콜이 사용한 PUF는 기존 PUF 연구에서는 가장 많이 연구된 PUF 중 하나인 RO(Ring Oscillator)-PUF를 사용하였으나 J. Delvaux 등은 균일하게 RO-PUF의 출력값을 재생성하기 위해서는 보조 데이터(helper data)가 필요하며 그를 위한 방법으로 퍼지 추출기(fuzzy extractor) 사용을 권장하고 있다[12]. 본 논문에서 제안하는 프로토콜에서는 보조 데이터를 통해 기존보다 균일한 출력값이 나오도록 Dodis 등의 퍼지 추출기를 적용하여 설계하였다[13].

2.2 퍼지 추출기

퍼지 추출기는 Gen(Generate)과 Rep(Reproduce)라는 과정을 통해 기존 입력값과 유사한 입력값이 입력되었을 때 보조 데이터로 균일한 출력값으로 생성해주는 기술이다. 이 기술을 PUF에 적용하면 다양한 잡음 변수로 변형이 생긴 PUF 출력값을 균일하게 생성하여 인증이 정상적으로 진행될 수 있게 도와준다.

제안한 프로토콜에 적용된 퍼지 추출기를 적용한 PUF의 동작은 먼저, 다음 식처럼 입력값 x_i에 대한 PUF 출력값 y_i가 Gen을 통해 새로운 출력값

k_i와 잡음를 처리하는 보조 데이터 H_i를 출력한다 : $(k_i, H_i) = Gen(PUF(x_i))$.

그리고 입력값 x_i로 출력값 k_i를 재생성할 때에는 다음 식처럼 (x_i, H_i)를 PUF와 Rep에 입력하는데 입력값 x_i 대신 유사한 입력값인 x_i'가 입력되더라도 보조 데이터를 통해 k_i를 재생성할 수 있다 : $k_i = Rep(PUF(x_i), H_i)$.

이러한 퍼지 추출기 동작의 제한사항은 보조 데이터의 크기이다. 비록 Jung Hee Cheon 등이 설계한 퍼지 추출기가 보조 데이터 크기를 1024bit 기준으로 최소 1.59MB로 축소시켜서 경량화 가능성을 보여줬지만 여전히 프로토콜의 메시지로 전송하기에는 부담이 되는 데이터 크기이다[14]. 그래서 제안하는 프로토콜에서는 무선 센서 노드의 내부 저장공간에 퍼지 추출기의 보조 데이터를 저장하여 사용하도록 설계하였다. 추후 퍼지 추출기의 보조 데이터 크기가 인증 프로토콜 메시지 안에 포함될 정도로 경량화되면 보조 데이터의 저장 위치를 더 유연하게 설정한 인증 프로토콜 연구가 진행될 수 있다.

2.3 기호

이 논문에서 사용하는 주요 기호들은 [Table. 1]과 같이 정리하였다.

Table 1. Symbols in protocols

Symbol	Internal use
r_i	Challenge sequence number of the ith node
x_i^r	rth PUF's input of the ith node
y_i^r	rth PUF's output of the ith node
Γ_i	PUF of the ith node
$R_{x_i^r}$	Random string of rth PUF's input
PRNG	Pseudo Random Number Generator
e_i	Factor for Group key calculation of the ith node
l	Identify the use of Group key
N_i	Identity of the ith node
k_i	Fuzzy extractor's output of the ith node
H_i	Fuzzy extractor's Helper Data of the ith node
r_a	Random integer of AC
E_K	AES algorithm Encryption
$H(\bullet)$	Hash algorithm Encryption
MAC_K	MAC algorithm Encryption
$\|$	concatenation operator
$\oplus$	exclusive or operator

III. Huang의 프로토콜[11]

이 장에서는 Huang의 프로토콜에 대해 살펴본다. 이 프로토콜의 그룹 키 분배는 "예비(Preliminary) - 조정(Coordination) - 분배(Distribution)" 세 가지 과정을 8개 단계로 나누어서 아래와 같이 수행한다.

3.1 예비 과정(Preliminary phase)

1) 주 제어부는 내부 저장공간에 전송부의 모든 노드의 초기 PUF CRP인 (r_i, x_i^0, y_i^0) 을 저장하고, i번째 노드는 (r, x_i^0)을 저장한다. 그리고 주 제어부와 보조 제어부의 인증을 위한 key_c를 랜덤하게 생성하여 저장한다. x_i^0은 주 제어부에 의해 생성된 랜덤한 값이며, y_0^r는 x_i^0을 입력값으로 하여 각 노드의 PUF에서 생성한 PUF 출력값 $y_0^r = \Gamma_i(x_i^0)$ 이다.
2) 두 제어부와 무선 센서 노드에는 Hash와 AES 알고리즘을 설정해 놓는다.

3.2 조정 과정(Coordination phase)

3) 주 제어부는 그룹 키 key_m을 랜덤한 값으로 생성하고 내부 저장공간에 저장한다.
4) 주 제어부는 내부 저장공간에서 PUF CRP (r, x_i^r, y_i^r)을 읽고 노드 N_i를 선정한다. 그리고 그룹 키 계산을 위한 요소 $e_i = y_i^r \oplus key_m$ 와 메시지 msg_c를 다음 식처럼 생성한 후 노드와 가장 가까운 보조 제어부로 전송한다 : $msg_c = E_{key_c}(N_i \| H(N_i) \| E_{y_i^r}(e_i \| r \| l) \| H(e_i \| r \| l))$.

Main Controller storage : key_c, (r, x_i^r, y_i^r)		**Auxiliary Controller** storage : key_c		**Node** storage : (r, x_i^r)
$key_m \leftarrow \{0,1\}^{l_m}$ select N_i $e_i \leftarrow y_i^r \oplus key_m$ store key_m, l $msg_c = E_{key_c}(N_i \parallel H(N_i) \parallel E_{y_i^r}(e_i \parallel r \parallel l) \parallel H(e_i \parallel r \parallel l))$	msg_c →	verify msg_c using key_c		
		$msg_d = E_{y_i^r}(e_i \parallel r \parallel l) \parallel H(e_i \parallel r \parallel l)$	msg_d →	$R_{x_i^r} \leftarrow PRNG(x_i^r)$ $y_i^r \leftarrow \Gamma_i(R_{x_i^r})$ $(e_i \parallel r \parallel l) \leftarrow D_{y_i^r}(E_{y_i^r}(e_i \parallel r \parallel l))$ verify msg_d using y_i^r store key_m, l
verify msg_s using y_i^r $x_i^{r+1} \leftarrow H(x_i^r)$ update $(r+1, x_i^{r+1}, y_i^{r+1})$	msg_s ←	-	msg_s ←	$x_i^{r+1} \leftarrow H(x_i^r)$ $R_{x_i^{r+1}} \leftarrow PRNG(x_i^{r+1})$ $y_i^{r+1} \leftarrow \Gamma_i(R_{x_i^{r+1}})$ update $(r+1, x_i^{r+1})$ $msg_s \leftarrow E_{y_i^r}(y_i^{r+1} \parallel r+1) \parallel H(y_i^{r+1} \parallel r+1)$

* Verification failure, transfer error message(msg_e).

⇩

Main Controller storage : key_c, , key_m, $(r+1, x_i^{r+1}, y_i^{r+1})$, l		**Auxiliary Controller** storage : key_c		**Node** storage : $(r+1, x_i^{r+1})$, key_m, l

Fig. 1. Huang et al. Protocol

3.3 분배 과정(Distribution Phase)

5) 보조 제어부는 key_c로 msg_c를 복호화한 후 N_i를 $H(N_i)$를 통해 검증한다. 검증이 완료되면 메시지 msg_d를 다음 식처럼 생성하여 노드 N_i로 전송한다 : $msg_d = E_{y_i^r}(e_i \parallel r \parallel l) \parallel H(e_i \parallel r \parallel l)$.

6) 노드 N_i는 내부 저장공간에 저장되어 있던 x_i^r을 이용하여 $R_{x_i^r} = PRNG(x_i^r)$ 을 만든 후, PUF의 입력으로 넣어 PUF 출력 y_i^r을 생성한다. 생성된 y_i^r으로 msg_d를 복호화한 후 $e_i \parallel r \parallel l$와 $H(e_i \parallel r \parallel l)$을 비교하여 검증한다. 검증된다면 e_i, r, l을 이용하여 다음 단계로 넘어가지만, 검증이 안 되면 보조 제어부를 거쳐 주 제어부에게 메시지 msg_e을 전송하고 생성했던 y_i^r을 제거한다.

7) 노드 N_i는 $x_i^{r+1} = H(x_i^r)$ 을 생성하여 $R_{x_i^{r+1}} = PRNG(x_i^{r+1})$을 계산한다. $R_{x_i^{r+1}}$를 PUF에 입력하여 출력 y_i^{r+1}을 생성한 후 $(r+1, x_i^{r+1})$을 내부 저장공간에 업데이트 시킨다. 키 분배 성공 메시지 msg_s을 다음 식처럼 생성한 후 보조 제

Main Controller storage : key_c, $\widehat{key_c}$, k_i		**Auxiliary Controller** storage : key_c, $\widehat{key_c}$		**Node** storage : (x_i, H_i)
$key_m \leftarrow \{0,1\}^{l_m} + \{0,1\}^{l_r}$ $r_{an} \leftarrow \{0,1\}^{l_{an}}$ select N_i $c = E_{key_c}(N_i, r_{an}, E_{k_i}(key_m, r_{an}))$ $msg_c \leftarrow c \parallel MAC_{\widehat{key_c}}(c)$	msg_c $\rightarrow$	verify msg_c using $\widehat{key_c}$ $N_i, r_{an} \leftarrow D_{key_c}(c)$ $key_{an} = H(r_{an} \parallel 1)$ $\widehat{key_{an}} = H(r_{an} \parallel 2)$ store key_{an}, $\widehat{key_{an}}$		
		$r_a \leftarrow \{0,1\}^{l_a}$ $d = E_{key_{an}}(r_a) \parallel E_{k_i}(key_m, r_{an})$ $msg_d \leftarrow d \parallel MAC_{\widehat{key_{an}}}(d)$	msg_d $\rightarrow$	$k_i \leftarrow Rep(x_i, H_i)$ $key_m, r_{an} \leftarrow D_{k_i}(E_{k_i}(key_m, r_{an}))$ $key_{an} = H(r_{an} \parallel 1)$ $\widehat{key_{an}} = H(r_{an} \parallel 2)$ verify msg_d using $\widehat{key_{an}}$ store key_{an}, $\widehat{key_{an}}$, key_m
		verify msg_s using $\widehat{key_{an}}$ $r_a + 1 \leftarrow D_{key_{an}}(E_{key_{an}}(r_a + 1 \parallel E_{k_i}(key_m)))$	msg_s $\leftarrow$	$r_a \leftarrow D_{key_{an}}(E_{key_{an}}(r_a))$ $e = E_{key_{an}}(r_a + 1 \parallel E_{k_i}(key_m))$ $msg_s \leftarrow e \parallel MAC_{\widehat{key_{an}}}(e)$
verify msg_f using $\widehat{key_c}$ $E_{k_i}(key_m) \leftarrow D_{key_c}(E_{key_c}(E_{k_i}(key_m)))$ $key_m \leftarrow D_{k_i}(E_{k_i}(key_m))$	msg_f $\leftarrow$	$f = E_{key_c}(E_{k_i}(key_m))$ $msg_f \leftarrow f \parallel MAC_{\widehat{key_c}}(f)$		

* Verification failure, transfer error message(msg_e).

⇩

Main Controller		**Auxiliary Controller**		**Node**
storage : key_c, $\widehat{key_c}$, k_i, key_m		storage : key_c, $\widehat{key_c}$, key_{an}, $\widehat{key_{an}}$		storage : (x_i, H_i), key_{an}, $\widehat{key_{an}}$, key_m

Fig. 2. Proposed Protocol

어부에게 전송한다 : $msg_s = E_{y_i^r}(y_i^{r+1} \parallel r+1) \parallel H(y_i^{r+1} \parallel r+1)$.

8) msg_s는 보조 제어부를 거쳐 주 제어부로 전송되고 주 제어부는 저장되어 있던 y_i^r로 메시지를 복호화하여 $H(y_i^{r+1} \parallel r+1)$을 검증한다. 검증이 완료되면 $x_i^{r+1} = H(x_i^r)$ 을 생성하고 y_i^{r+1}, $r+1$으로 내부 저장공간에 저장되어 있던 PUF CRP를 $(r+1, x_i^{r+1}, y_i^{r+1})$으로 업데이트한다.

IV. 제안하는 프로토콜

이 장에서는 Huang의 프로토콜을 개선한 그룹 키 분배 방법을 제안한다. 주요 개선사항은 보조 제어부와 노드 사이의 인증키 key_{an} 추가, 동기화 정보 유지 및 업데이트 과정 삭제 후 카운터 스트링과 랜덤 정보 추가, Hash 알고리즘 대신 MAC 알고리즘 적용 등이다.

먼저, 인증키 추가는 주 제어부가 랜덤한 값 r_{an}를 생성하여 보조 제어부와 노드에게 배포한다. 보조 제어부에서는 주 제어부가 전송한 메시지를 주 제어

부와 공유한 키 key_c로 복호화하여 r_{an}을 획득하고, r_{an}로 key_{an}을 생성하여 저장한다. 노드는 주 제어부가 보조 제어부를 거쳐 전송한 메시지를 PUF로 생성한 출력값으로 복호화하여 r_{an}을 획득하고, r_{an}로 key_{an}을 생성하여 저장한다. 배포가 완료된 key_{an}은 노드가 보조 제어부를 거쳐 주 제어부로 메시지를 보낼 때 보조 제어부와 노드 사이의 인증을 진행하는데 사용한다.

두 번째로, Haung의 프로토콜에서 사용된 동기화 정보 r을 삭제하는 대신 랜덤 스트링으로만 구성되었던 그룹 키 key_m을 랜덤 스트링과 카운터 스트링을 결합하여 사용하였고 보조 제어부에서 랜덤한 값을 포함한 메시지를 노드에 전송하도록 설계하였다. 최초 그룹 키 분배 과정을 제외한 과정부터는 노드가 메시지의 key_m과 노드에 저장공간에 있는 key_m'의 카운터 스트링을 비교하여 정당한 key_m인 경우만 내부 저장공간에 갱신하여 저장한다. 그리고 Huang의 프로토콜처럼 PUF의 CRP를 계속 업데이트하려면 매 세션마다 PUF 출력값을 노드가 퍼지 추출기의 보조 데이터와 같이 생성하고 주 제어부와 동기화를 수행해야 하므로 항상 동일한 PUF의 입력값을 사용하도록 설계하였다.

또한, Hash 알고리즘 대신 MAC 알고리즘을 적용하기 위해 사전에 주 제어부와 보조 제어부가 공유하는 키 $\widehat{key_c}$를 배포하고, 주 제어부에서 생성한 r_{an}로 보조 제어부와 노드가 공유하는 키 $\widehat{key_{an}}$을 생성하여 사용하도록 설계하였다.

그룹 키 분배 과정은 Huang의 프로토콜과 동일하게 "예비(Preliminary) - 조정(Coordination) - 분배(Distribution)" 세 가지 과정을 8개 단계로 나누어서 아래와 같이 수행한다.

4.1 예비 과정(Preliminary Phase)

1) 주 제어부는 내부 저장공간에 모든 노드의 퍼지 추출기 출력값 k_i를 저장하고, i번째 노드는 (x_i, H_i)를 저장한다. 그리고 주 제어부와 보조 제어부 간의 인증을 위해 랜덤한 값 r_c으로 key_c, $\widehat{key_c}$을 다음 식처럼 생성하여 주 제어부와 보조 제어부의 내부 저장공간에 저장한다 : $key_c = H(r_c \| 1)$, $\widehat{key_c} = H(r_c \| 2)$.

2) 두 제어부와 무선 센서 노드에는 MAC과 AES 알고리즘을 설정해 놓는다.

4.2 조정 과정(Coordination Phase)

3) 주 제어부는 그룹 키 key_m과 r_{an}을 랜덤한 값으로 생성하고 key_m을 내부 저장공간에 저장한다. 단, key_m은 l_m 길이의 랜덤 스트링과 l_r 길이의 카운터 스트링으로 구성되며, 카운터 스트링은 매 세션마다 증가한다.

4) 주 제어부는 내부 저장공간에서 k_i를 읽고 노드 N_i를 선정한다. 그리고 메시지 msg_c를 다음 식처럼 생성한 후 노드와 가장 가까운 보조 제어부로 전송한다 : $msg_c \leftarrow c \| MAC_{\widehat{key_c}}(c)$, $c = E_{key_c}(N_i, r_{an}, E_{k_i}(key_m, r_{an}))$.

4.3 분배 과정(Distribution Phase)

5) 보조 제어부는 $\widehat{key_c}$를 이용하여 msg_c를 검증하고 key_c로 복호화한다. 복호화하여 얻은 r_{an}로 key_{an}, $\widehat{key_{an}}$을 생성한다. 보조 제어부는 랜덤한 값 r_a를 생성하고 그룹키 분배 메시지 msg_d를 다음 식처럼 생성하여 노드 N_i로 전송한다 : $msg_d \leftarrow d \| MAC_{\widehat{key_{an}}}(d)$, $d = E_{key_{an}}(r_a) \| E_{k_i}(key_m, r_{an})$.

6) 노드는 msg_d에 내부 저장공간에 저장되어 있던 (x_i, H_i)쌍과 퍼지 추출기의 Rep 생성자를 이용하여 k_i를 생성하고 $E_{k_i}(key_m, r_{an})$을 복호화한다. 복호화하여 얻은 r_{an}로 key_{an}, $\widehat{key_{an}}$을 생성하고 $\widehat{key_{an}}$으로 msg_d를 검증한다.

7) 노드는 key_{an}으로 $E_{key_{an}}(r_a)$을 복호화하여 r_a를 얻고 key_m, key_{an}과 함께 그룹 키 분배 성공 메시지 msg_s을 다음 식처럼 생성하여 보조 제어부로 전송한다 : $msg_s \leftarrow e \| MAC_{\widehat{key_{an}}}(e)$, $e = E_{key_{an}}(r_a + 1 \| E_{k_i}(key_m))$.

8) 보조 제어부는 $\widehat{key_{an}}$을 이용하여 msg_s룰 검증하고 key_{an}으로 $E_{key_{an}}(r_a + 1 \| E_{k_i}(key_m))$를 복호화한다. $r_a + 1$가 확인되면 r_a를 삭제하고 msg_f

를 다음 식처럼 생성하여 주 제어부로 전송한다. 주 제어부는 msg_f를 $\widehat{key_c}$로 검증한 후 k_i과 key_c로 복호화하여 key_m을 확인한다 : $msg_f \leftarrow f \parallel MAC_{\widehat{key_c}}(f)$, $f = E_{key_c}(E_{k_i}(key_m))$.

V. 안전성 및 보안 분석

이 장에서는 Huang의 프로토콜과 제안한 프로토콜의 안전성 및 보안을 분석한다. Huang이 가정했던 주 제어부가 가지고 있는 내부 저장공간은 안전하고, 암호화 알고리즘은 (예:AES, Hash, MAC) 아무런 정보 없이 공격자가 복호화할 수 없다는 가정을 동일하게 유지하여 분석하였다. 제안된 프로토콜의 안전성 분석은 안전성 모델에 기반한 증명 과정 없이 휴리스틱한 접근법을 취하였다. 그 이유는 아직 SDWSN 환경에서의 안전성 모델에 대한 연구가 키 분배 및 키 교환 관점에서 이루어지지 않았기 때문이다. SDWSN 환경에서 제어부, 보조 제어부라는 구성요소는 기존 안전성 모델과의 분명한 차이점을 제공하므로 이에 대한 추가적인 안전성 모델 정의에 대한 연구가 필요하다.

5.1 안전성 분석

먼저, Huang의 프로토콜은 두 가지의 그룹 키 분배 방식을 혼용해서 사용한다. 첫 번째 방식인 주 제어부에서 직접 그룹 키를 분배하는 방식은 대부분의 PUF를 활용한 모델처럼 제어부와 노드간의 양자간 모델이다. 두 번째 방식은 주 제어부와 노드 사이에 보조 제어부를 포함하여 그룹 키를 분배하는 방식으로 제어부와 전송부의 성능 향상 및 병렬처리 기능 등을 높이기 위해 보조 제어부를 활용한 삼자간 모델이다. 첫 번째 방식은 PUF의 출력값을 이용하여 주 제어부와 노드 인증 및 키 교환을 수행하여 전체 구성요소의 인증이 이루어진다. 이와 달리 두 번째 방식은 주 제어부와 노드 인증이 수행되지만 보조 제어부에서 구성되는 메시지에 대한 인증은 결여되어 있다. 그 이유는 전체적인 인증 프로토콜의 메시지가 첫 번째 방식을 이용한 주 제어부와 노드 사이의 PUF 인증을 기준으로 설계되어 있고, 보조 제어부는 주 제어부에서 설정한 노드 경로를 따라 메시지를 key_c로 복호화한 후 노드에 전달하는 역할만 가지도록 설계되었기 때문이다. 본 논문에서 제안하는 프로토콜은 보조 제어부와 노드 사이의 인증을 수행하는 방법을 전혀 고려하지 않은 Huang의 프로토콜과는 달리 주 제어부와 보조 제어부 간의 인증키 key_c를 분배하는 것과 더불어 보조 제어부와 노드 간의 인증키 key_{an}을 분배하여 보조 제어부의 인증을 수행하도록 설계하였다. 최종적으로 주 제어부와 보조 제어부, 노드 간의 각각의 상호 인증이 되도록 하되 그룹 키를 안전하게 노드에게 분배하는 삼자간 환경에서의 PUF 기반 인증 모델을 설계하였다.

또한, Huang의 프로토콜은 주 제어부와 노드가 인증을 위해 PUF의 입력과 출력에 대한 정보 r을 동기화시켜 유지하는 구조를 가진다. 만약 r이 맞지 않을 경우, 키 교환 과정이 정상적으로 작동하지 않게 된다. 즉, 공격자가 동기화를 강제적으로 막고 프로토콜을 실패시킨다면, 다음 인증 과정이 정상적으로 이루어지지 않는 결함이 있다. 그러므로 이러한 결점을 막기 위해서는 동기화 정보 없이 인증 프로토콜을 설계하는 것이 더욱 바람직하다. Huang이 이러한 동기화 정보를 유지하도록 설계한 가장 큰 이유는 메시지 재생 공격을 막기 위해서다. 본 논문에서는 동기화 정보를 유지하지 않되 메시지 재생 공격을 막을 수 있도록 주 제어부에서 생성되는 그룹 키에 카운터 스트링을 추가하고 각 메시지가 랜덤 정보를 유지하도록 설계하였다. 즉, 카운터 스트링을 통해 노드에서 그룹 키를 검증하고, 주 제어부, 보조 제어부, 노드에서 생성되는 모든 메시지에 랜덤 정보를 포함되어 재생 공격을 방지한다. 또한, 제안하는 프로토콜은 Huang의 프로토콜과 달리 PUF 출력값을 동기화하지 않고 항상 동일한 PUF의 입력값을 사용한다. 이렇게 설계한 이유는 PUF의 특성으로 인해 공격자가 PUF 입력값을 획득해도 노드로부터 메시지 암호화에 사용되는 PUF의 출력값을 얻을 수 없으며, PUF의 출력값이 단독으로 사용되지 않고 항상 랜덤하게 생성된 key_m, key_{an}과 함께 사용되므로 Huang의 프로토콜과 동일한 PUF와 암호 알고리즘의 안전성을 가진다. 또한, 그룹 키가 주 제어부에서 직접 분배되거나 보조 제어부를 활용하여 분배되어도 노드에게 안전하게 전달되도록 설계하였다.

5.2 보안 분석

이 절에서는 Bellare-Rogaway의 보안 모델처럼 공격자가 네트워크 내의 모든 통신을 통제하고 있

다고 가정하고 (즉, 공격자는 네트워크의 메시지를 읽고 전송하거나 메시지가 목적지에 도달하기 전에 수정하고 메시지를 지연하거나 재생할 수 있다.) 제안한 프로토콜에서 개선한 퍼지 추출기 적용, 보조 제어부 인증, 동기화 정보 제거의 관점에서 분석하였다[15].

5.2.1 퍼지 추출기 적용

Huang의 프로토콜과 제안한 프로토콜은 각각 PUF 출력값과 퍼지 추출기 출력값으로 주 제어부에서 그룹 키를 암호화하여 노드에게 전송한다. 노드에서 동작하는 퍼지 추출기의 입력값은 저장공간에 있는 보조 데이터와 PUF의 입력값이 PUF로 생성하는 출력값이다. 그래서 공격자가 노드의 저장공간에서 PUF의 입력값과 퍼지 추출기의 보조 데이터를 획득해도 정당한 사용자가 아닌 공격자는 노드의 PUF 출력값을 얻을 수 없으므로 메시지를 복호화하고 그룹 키를 획득할 수 없다.

또한, 특정 노드가 가지고 있는 PUF와 동일한 출력값을 생성하는 PUF의 복제 불가능한 특성 때문에 만들 수 없으며, 그룹 키는 매 세션마다 주 제어부에서 랜덤하게 생성되므로 노드의 저장공간에 있는 그룹 키로 새로운 그룹 키를 획득하거나 생성할 수 없다. 공격자에게 포획된 노드는 네트워크 모니터링을 통해 트래픽 밀도가 높거나 이상 노드는 식별하면 노드 그룹에서 탈퇴시킨 후 다른 노드들의 그룹 키를 변경하면 정상적으로 네트워크를 운영할 수 있다. 공격자가 노드를 포획하여 저장되어 있던 정보로 다른 정보를 얻으려고 하는 경우를 고려해보자.

- 〔Huang의 프로토콜〕 : 노드의 저장공간에 있는 (r,x_i^r)가 유출되어도 r은 단순 동기화 정보이고, PUF 입력값인 x_i^r로 출력값 y_i^r를 생성할 수 없으므로 y_i^r로 암호화된 msg_d와 msg_s를 복호화하여 key_m을 얻을 수 없다. 또한, 저장공간의 key_m은 주 제어부에서 랜덤하게 생성된 값으로 공격자가 생성할 수 없다. 그러므로 네트워크가 이상 노드를 식별하여 그룹키를 변경해버리면 획득했던 이전 세션의 그룹키는 사용할 수 없다.
- 〔제안한 프로토콜〕 : (x_i,H_i)가 유출되어도 PUF 입력값 x_i와 보조 데이터 H_i로 출력값 k_i를 생성할 수 없으므로 k_i로 암호화된 $E_{k_i}(key_m,r_{an})$을 복호화하여 r_{an}, key_m을 얻을 수 없다. 또한, 노드의 저장공간에 있는 key_m은 주 제어부에서 생성한 랜덤 값이고, key_{an}은 주 제어부에서 생성한 랜덤 값에 Hash를 진행하여 생성한 값으로 공격자가 생성할 수 없다. 그러므로 네트워크가 이상 노드를 식별하여 그룹키를 변경해버리면 획득했던 이전 세션의 그룹키는 사용할 수 없다.

5.2.2 보조 제어부 미인증

Huang의 프로토콜은 노드가 주 제어부로 메시지를 전송할 때 보조 제어부는 중간에서 전송만 하는 역할을 한다. 이 때문에 공격자는 주 제어부의 정확한 위치를 알 수 없어도 식별된 보조 제어부를 향해 메시지를 보내거나 임의의 노드에서 메시지를 전송하기만 해도 주 제어부에 대한 서비스 거부 공격을 더 쉽게 할 수 있다. 이와 달리 제안한 프로토콜은 보조 제어부에서 정당한 노드가 보낸 메시지인지 확인하기 때문에 식별되지 않은 주 제어부를 공격할 수 없다. 공격자가 보조 제어부를 식별하거나 임의의 노드에서 메시지를 전송할 수 있는 경우를 고려해보자.

- 〔Huang의 프로토콜〕 : 보조 제어부가 식별된 경우, 임의의 메시지를 보조 제어부로 전송하기만 하면 그 모든 메시지가 주 제어부로 전송되어 서비스 거부 공격을 할 수 있다. 또한, 임의의 노드에서 메시지 전송이 가능한 경우, 노드가 통신 범위 안에 위치한 모든 보조 제어부에 메시지를 보내기 때문에 같은 메시지 수를 보내더라도 주변 보조 제어부 수만큼 더 많은 메시지가 주 제어부로 전송되어 효과적인 서비스 거부 공격이 가능하다.
- 〔제안한 프로토콜〕 : 보조 제어부가 식별되거나 임의의 노드에서 메시지 전송이 가능한 경우, 임의의 메시지가 보조 제어부로 전송되더라도 보조 제어부가 정당한 메시지를 확인하기 때문에 주 제어부로 메시지가 전송되지 않아 주 제어부에 대한 서비스 거부 공격을 할 수 없다.

5.2.3 동기화 정보 제거

Huang의 프로토콜은 그룹 키를 분배할 때마다

메시지에 포함된 동기화 정보인 r이 증가하기 때문에 재생 공격에 대한 저항성을 가지고 있다. 제안한 프로토콜은 동기화 정보를 제거하였지만 카운터 스트링과 랜덤 정보로 메시지를 구성하여 재생 공격에 대한 저항성을 가지고 있다. 공격자가 각 객체 사이에서 재생 공격을 하는 경우를 생각해보자.

- 〔주 제어부 ~ 보조 제어부〕 : Huang의 프로토콜와 제안한 프로토콜 둘 다 보조 제어부의 저장공간에 있는 key_c로 메시지 안의 정보만 확인하기 때문에 이 구간에서 재생 공격이 가능하다. 하지만 Huang의 프로토콜은 노드에서 r을 확인하고, 제안한 프로토콜은 저장공간의 그룹 키와 새로 받은 그룹키에 포함된 카운터 스트링을 확인한다. 정당하지 않은 메시지이면 msg_e를 보조 제어부를 통해 주 제어부로 전송하기 때문에 최종적인 그룹 키 분배에는 영향을 안 준다.
- 〔보조 제어부 ~ 노드〕 : Huang의 프로토콜은 메시지에 포함된 r과 노드의 저장공간에 있는 r'가 다르므로 재생 공격이 불가능하다. 또한, 제안한 프로토콜에서는 보조 제어부에서 생성한 랜덤 정보 r_a가 이전 메시지의 r_a'와 다르고 key_m의 카운터 스트링이 정당하지 못하기 때문에 재생 공격이 불가능하다.

VI. 성능 분석

이 절에서는 Huang의 프로토콜과 제안한 프로토콜이 주 제어부 1개, 보조 제어부 m개, 노드 n개로 구성된 SDWSN 환경에서 동일한 PUF를 사용한다고 가정하여 성능을 분석하였다. 이렇게 한 이유는 SDWSN 환경에서 노드는 자신의 통신 범위 내에 위치한 주변 노드 및 제어부를 향해 메시지를 브로드캐스팅하고, 제안한 프로토콜에서 적용한 퍼지 추출기는 균일한 출력값을 얻기 위한 용도로 Huang의 프로토콜도 PUF의 성능을 높이기 위해서는 적용해야 하기 때문이다.

전송 비용과 저장공간 비용을 계산하기 위한 파라미터는 key_m과 e_i, y_i^r, k_i는 α로 설정하였다. y_i^r과 k_i는 PUF 출력값이고, key_m은 y_i^r과 XOR 연산을 진행하고 e_i는 연산 결과이므로 같은 크기로 설정하였다. 또한, 노드 주소는 γ로 설정하였으며, Huang의 프로토콜의 상수 r, l과 제안한 프로토콜의 랜덤 상수 r_a, r_{an}는 β로 설정하였다. 마지막으로 key_c와 각 프로토콜에 쓰인 Hash의 출력값과 MAC의 출력값은 δ으로 고정되도록 설정하였다.

연산처리 비용의 파라미터는 XOR과 PRNG 연산은 단순한 작업이므로 연산처리 비용에는 포함 안 시켰다. PUF, AES, Hash, MAC 알고리즘 연산을 각각 $\triangle_P$, $\triangle_A$, $\triangle_H$, $\triangle_M$로 설정하였으며, 동기화 정보를 업데이트시키는 연산을 $\triangle_U$로 설정하였다.

6.1.1 전송 비용

Huang의 프로토콜에서 msg_e는 특별한 정보를 가지고 있지 않으므로 생략하였으며, msg_e을 제외한 msg_c, msg_d, msg_s의 길이는 $L_c=\gamma+\alpha+2(\beta+\delta), L_d=\alpha+2\beta+\delta$, $L_s=\alpha+\beta+\delta$ 이며, 제안한 프로토콜에서의 msg_c', msg_d', msg_s', msg_f의 길이는 $L_c'=\gamma+\alpha+2\beta+\delta$, $L_d'=\alpha+2\beta+\delta, L_s'=\alpha+\beta+\delta$, $L_f=\alpha+\delta$ 이다. 제안한 프로토콜이 보조 제어부와 노드의 인증을 위해 랜덤 값 r_{an}을 추가로 전송하지만 주 제어부에서는 Hash 값을 전송하지 않고, 보조 제어부에서는 msg_s을 인증을 통해 Huang의 프로토콜이 msg_s를 인증하지 않아 보조 제어부 주변에 있는 노드 수 n만큼 msg_s를 전송하는 비용이 감소한다. 또한, 노드에서의 전송 비용이 동일하므로 전체적인 비용에서 제안한 프로토콜의 비용이 감소한다.

- 〔주 제어부의 전송 비용〕 : Huang의 프로토콜과 제안한 프로토콜은 보조 제어부로 msg_c만 전송한다. 두 프로토콜이 msg_c를 전송하는데 필요한 전송 비용의 차이는 제안한 프로토콜 기준으로 $-\delta$이다. 제안한 프로토콜이 동기화 정보 대신 랜덤한 값 r_{an}을 넣고 Hash 값을 삭제하여 Huang의 프로토콜보다 전송 비용이 감소한다.
- 〔보조 제어부의 전송 비용〕 : Huang의 프로토콜은 msd_d와 msg_s를 전송하고 제안한 프로토콜은 msg_d와 msg_f를 전송한다. Huang의 프로토콜의 전송 비용은 $\alpha+2\beta+\delta+(n\times(\alpha+\beta+\delta))$, 제안한 프로토콜의 전송 비용은 $2\alpha+2\beta+2\delta$로 두 전송 비용의 차이는 제안한 프로토콜

기준으로 $-((n-1)\alpha+(n-1)\delta)$ 이다. 보조 제어부에 연결되는 노드가 많아질수록 제안하는 프로토콜의 전송 비용이 Huang의 프로토콜보다 감소한다.

- 〔노드의 전송 비용〕 : Huang의 프로토콜과 제안한 프로토콜은 보조 제어부로 msg_s를 전송한다. 두 프로토콜이 msg_s를 전송하는데 필요한 전송 비용은 $\alpha+\beta+\delta$로 동일하다.

6.1.2 저장공간 비용

Huang의 프로토콜은 저장공간에 주 제어부는 key_c, $r+1$, x_i^{r+1}, y_i^{r+1}, key_m, l, 보조 제어부는 key_c, 노드는 $r+1$, x_i^{r+1}, key_m, l을 저장한다. 제안한 프로토콜은 동일한 PUF를 사용했다고 가정하면 노드가 H_i를 저장하지 않기 때문에 저장공간에 주 제어부는 key_c, $\widehat{key_c}$, k_i, key_m, 보조 제어부는 key_c, $\widehat{key_c}$, key_{an}, $\widehat{key_{an}}$, 노드는 x_i, key_{an}, $\widehat{key_{an}}$, key_m를 저장한다. Huang의 프로토콜은 주 제어부와 노드가 동기화 정보를 저장하고 있어 주 제어부에 연결된 노드 수 만큼 동기화 정보를 저장해야 하는 부담이 있다. 제안한 프로토콜에서 동기화 정보를 삭제해서 전체적으로 저장공간 비용이 감소하지만 보조 제어부와 노드는 상호 인증을 위한 key_{an}, MAC 연산을 위한 $\widehat{key_c}$, $\widehat{key_{an}}$이 추가되어 저장공간 비용이 증가한다. 두 프로토콜의 저장공간 비용은 전체적으로 보면 제안한 프로토콜이 감소하지만 각 객체로 보면 보조 제어부, 노드 비용은 Huang의 프로토콜이 유리하고 주 제어부 비용은 제안한 프로토콜이 유리하다.

- 〔주 제어부의 저장공간 비용〕 : Huang의 프로토콜은 $(n\times 2(\alpha+\beta))+\alpha+(m\times\delta)$, 제안한 프로토콜은 $(n\times\alpha)+\alpha+(m\times 2\delta)$ 의 저장공간 비용이 필요하다. 제안한 프로토콜에서 동기화 정보를 삭제하면서 주 제어부에 연결된 노드당 필요한 정보가 $\alpha+2\beta$만큼 감소하고, $\widehat{key_c}$를 추가하여 각 보조 제어부당 δ가 증가한다. 두 전송 비용의 차이는 제안한 프로토콜 기준으로 $-((n\times(\alpha+2\beta))-(m\times\delta))$로 제안한 프로토콜의 비용이 연결된 보조 제어부 수 만큼 증가하지만 주 제어부에 연결된 노드 수 만큼 감소하여 전체적인 비용은 감소한다.
- 〔보조 제어부의 저장공간 비용〕 : Huang의 프로토콜은 δ, 제안한 프로토콜은 4δ의 저장공간 비용이 필요하다. 제안한 프로토콜에서 보조 제어부 인증을 위해 key_{an}을 추가하고 MAC 연산을 위해 $\widehat{key_c}$, $\widehat{key_{an}}$을 추가한 만큼 3δ의 비용이 증가한다.
- 〔노드의 저장공간 비용〕 : Huang의 프로토콜은 $2\alpha+2\beta$, 제안한 프로토콜은 $2\alpha+2\delta$의 저장공간 비용이 필요하다. 제안한 프로토콜에서 동기화 정보를 삭제해서 2β만큼 저장공간이 감소하지만 key_{an}, $\widehat{key_{an}}$을 추가하여 2δ가 증가한다. 두 프로토콜에서 제안한 프로토콜 기준 저장공간 비용 차이는 $-(2\beta-2\delta)$로 Hash 값 δ가 랜덤한 상수 β보다 크기 때문에 제안한 프로토콜이 Huang의 프로토콜보다 저장공간 비용이 증가한다

6.1.3 연산처리 비용

Huang의 프로토콜에서 주 제어부는 msg_c 생성과 msg_s를 검증, 동기화 정보 업데이트를 수행하며, 보조 제어부는 msg_c 검증, 노드는 msg_d를 검증과 msg_s 생성, 동기화 정보 업데이트를 수행한다. 제안한 프로토콜에서는 주 제어부는 msg_c 생성과 msg_s를 검증을 수행하며, 보조 제어부는 msg_c, msg_s 검증과 msg_d, msg_f를 생성, 노드는 msg_d를 검증하고 msg_s 생성한다. Huang의 프로토콜은 주제어부와 노드가 그룹 키 분배가 될 때마다 동기화 정보를 업데이트한다. 이 때, 노드는 1번씩만 업데이트를 수행하면 되지만 주 제어부는 연결된 노드 수 만큼 업데이트를 수행해야 하는 부담이 있다. 또한, 보조 제어부에서 msg_s를 인증하지 않기 때문에 주 제어부에서 msg_s를 검증하는데 msg_s를 전송하는 보조 제어부 수 m만큼 AES와 Hash 연산이 증가한다. 제안한 프로토콜은 동기화 정보 업데이트 연산이 없어지면서 주 제어부와 노드의 연산이 감소한다. 하지만 노드에서 보조 제어부 인증을 위한 msg_s를 생성하기 위해 AES 연산이 증가하고, 보조 제어부에서 msg_s를 검증하기 위해 주변 노드 수 n만큼 MAC

Table 2. Performance Analysis

Communication Overhead			
Protocol	Main Controller	Auxiliary Controller	Node
Huang	$\gamma+\alpha+2(\beta+\delta)$	$\alpha+2\beta+\delta+(n\times(\alpha+\beta+\delta))$	$\alpha+\beta+\delta$
Proposed	$\gamma+\alpha+2\beta+\delta$	$2\alpha+2\beta+2\delta$	$\alpha+\beta+\delta$
Storage Overhead			
Protocol	Main Controller	Auxiliary Controller	Node
Huang	$(n\times2(\alpha+\beta))+\alpha+(m\times\delta)$	δ	$2\alpha+2\beta$
Proposed	$(n\times\alpha)+\alpha+(m\times2\delta)$	4δ	$2\alpha+2\delta$
Computational Overhead			
Protocol	Main Controller	Auxiliary Controller	Node
Huang	$((n\times m)+2)(\triangle_A+\triangle_H)+(n\times\triangle_U)$	$\triangle_A+\triangle_H$	$2\triangle_A+3\triangle_H+2\triangle_P+\triangle_U$
Proposed	$4\triangle_A+2\triangle_M$	$3\triangle_A+2\triangle_H+(n+3)\triangle_M$	$4\triangle_A+2\triangle_H+2\triangle_M+\triangle_P$

연산이 증가한다. 또한, 보조 제어부와 노드에서 r_{an}으로 key_{an}, $\widehat{key_{an}}$을 생성하기 위한 HASH 연산이 증가한다. 두 프로토콜의 연산처리 비용은 전체적으로 보면 제안한 프로토콜이 감소하지만 각 객체로 보면 주 제어부 비용은 제안한 프로토콜이 유리하고 보조 제어부와 노드의 비용은 Huang의 프로토콜이 유리하다.

- 〔주 제어부의 연산처리 비용〕 : Huang의 프로토콜에서 연산처리 비용은 msg_c를 생성하는 $2(\triangle_A+\triangle_H)$, msg_s를 검증하는 $n\times m(\triangle_A+\triangle_H)$, 동기화 정보를 업데이트하는 $n\times\triangle_U$가 필요하다. 제안한 프로토콜에서 연산 처리 비용은 msg_c를 생성하는 $2\triangle_A+\triangle_M$, msg_f를 검증하는 $2\triangle_A+\triangle_M$이 필요하다. 두 프로토콜의 연산처리 비용 차이는 제안한 프로토콜 기준으로 $-(((n\times m)-2)\triangle_A+(n\times m)\triangle_H+(n\times\triangle_U))$ 이므로 SDWSN 환경을 구성하는 보조 제어부와 노드가 많아질수록 제안한 프로토콜의 연산처리 비용이 감소한다.
- 〔보조 제어부의 연산처리 비용〕 : Huang의 프로토콜에서 연산처리 비용은 msg_c를 검증하는 $\triangle_A+\triangle_H$이 필요하다. 제안한 프로토콜에서 연산처리 비용은 msg_c를 검증하는 $\triangle_A+\triangle_M$, msg_d를 생성하는 $\triangle_A+\triangle_M$, msg_s를 검증하는 $n\times(\triangle_M)$, msg_f를 생성하는 $\triangle_A+\triangle_M$, key_{an}, $\widehat{key_{an}}$을 생성하는 $2\triangle_H$ 이 필요하다. 두 프로토콜의 연산처리 비용 차이는 제안한 프로토콜 기준 $2\triangle_A+2\triangle_H+(n+2)\triangle_M$로 보조 제어부로 msg_s를 보내는 노드가 많아질수록 제안한 프로토콜의 연산처리 비용이 증가한다.
- 〔노드의 연산처리 비용〕 : Huang의 프로토콜에서 연산처리 비용은 msg_d를 검증하는 $\triangle_A+\triangle_H+\triangle_P$, msg_s를 생성하는 $\triangle_A+2\triangle_H+$, $+\triangle_P$ 동기화 정보를 업데이트하는 $\triangle_U$가 필요하다. 제안한 프로토콜에서의 연산처리 비용은 msg_d를 검증하는 $\triangle_A+\triangle_M+\triangle_P$, msg_s를 생성하는 $3\triangle_A+\triangle_M$, key_{an}와 $\widehat{key_{an}}$을 생성하는 $2\triangle_H$이 필요하다. 두 프로토콜의 연산처리 비용 차이는 제안한 프로토콜 기준 $2(\triangle_A+\triangle_M)-(\triangle_H+\triangle_P+\triangle_U)$로 적용하는 PUF나 동기화 업데이트, 암호 알고리즘에 따라 변하겠지만 제안한 프로토콜의 연산처리 비용이 증가한다.

VII. 결론 및 향후 연구

본 논문에서는 Huang의 프로토콜의 보조 제어부 미인증과 동기화 정보 유지하는 취약점을 개선하면서 보안을 유지하고 성능을 향상시키는 방안에 대해 연구하였다[11]. 안전성 및 보안 면에서는 보조 제어부 미인증은 보조 제어부와 노드 간의 인증키 key_{an}을 추가적으로 분배하여 Huang의 프로토콜보다 서비스 거부 공격에 강건한 삼자간 환경에서의 PUF 기반 인증 모델을 설계하였다. 또한, 동기화 정보는 삭제 후 key_m에 카운터 스트링을 포함시키고 랜덤 정보를 활용하여 동기화 정보와 동일하게 메시지 재생 공격에 저항성을 가지도록 설계하였다. 또한, 퍼지 추출기를 사용하면서 SDWSN의 다양한 환경에서 균일한 출력값으로 그룹 키 분배과정을 수행하도록 설계하였다.

제안한 프로토콜은 Huang의 프로토콜보다 다양한 환경에서 사용할 수 있는 가용성을 제공할 수 있다. 보조 제어부 인증을 통한 서비스 거부 공격에 대한 저항성과 동기화 정보를 삭제하면서 실 환경에서 발생할 수 있는 무선 네크워크 끊김 현상에 대한 저항성을 가지고 있으며, SDWSN 환경을 구성하는 보조 제어부와 노드가 많을수록 주 제어부의 성능 측면에서 유리하다. 또한, 퍼지 추출기를 적용하여 환경적 변수로 인한 오차 발생률 등을 최소화한다.

성능 면에서는 동기화 정보 삭제, 보조 제어부 인증 추가로 주 제어부 및 전체적인 비용은 감소한다. 하지만 성능이 저하된 부분도 있는데 보조 제어부 인증과 MAC 알고리즘을 위해 추가적인 key_{an}, $\widehat{key_{an}}$, $\widehat{key_c}$ 저장 및 Hash 연산이 추가되어 보조 제어부와 노드의 저장공간과 연산처리 비용이 증가한다.

추후 주 제어부의 비용을 유지한 채로 보조 제어부와 노드의 저장공간과 연산처리 비용을 감소시키는 연구가 향후 과제로 남아 있다.

References

[1] S. Sezer, S. Scott-Hayward, P.-K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," IEEE Communications Magazine, vol. 51, no. 7, pp. 36-43, 2013.

[2] A. De Gante, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," In Communications (QBSC), pp. 71-75, 2014

[3] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, "Security in software-defined networking: Threats and countermeasures," Mobile Networks and Applications, vol. 21, no. 5, pp. 764-776, 2016.

[4] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for OpenFlow networks," In Proceedings of the first workshop on Hot topics in software defined networks, pp. 121-126, 2012

[5] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, "Towards a secure controller platform for OpenFlow applications," In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, pp. 171-172, 2013

[6] T. Luo, H.-P. Tan and T. Q. S. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," IEEE Communications letters, Vol. 16, No. 11, pp. 1896-1899, 2012

[7] M. P. Fernandez, "Comparing OpenFlow controller paradigms scalability: Reactive and proactive," In Advanced Information Networking and Applications, pp.1009-1016, 2013

[8] S. T. Ali, V. Sivaraman, A. Radford, and S. Jha, "A survey of securing networks using software defined networking," IEEE Transactions on Reliability, vol. 64, no. 3, pp. 1086 -

1097, 2015.

[9] S. Shin, P. Porras, V. Yegneswaran, and M. Fong, "FRESCO: Modular composable security services for software-defined networks," In 20th Annual Network & Distributed System Security Symposium. 2013

[10] J. R. Ballard, I. Rae, and A. Akella, "Extensible and scalable network monitoring using OpenSAFE," In Proceedings of USENIX Internet Network Management Workshop/Workshop on Research on Enterprise Networking, pp. 8, 2010.

[11] M. Huang, B. Yu, and S. Li, "Puf-assisted group key distribution scheme for software-defined wireless sensor networks," IEEE Communications Letters, Vol. 22, no. 2, pp. 404-407, 2018

[12] J. Delvaux and I. Verbauwhede, "Key-recovery attacks on various RO-PUF constructions via helper data manipulation," In Proceedings of the conference on Design, Automation & Test in Europe, pp. 72, 2014

[13] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," In International conference on the theory and applications of cryptographic techniques, pp. 523-540, 2004.

[14] Jung Hee Cheon, Jinhyuck Jeong, Dongwoo Kim, and Jongchan Lee, "A reusable fuzzy extractor with practical storage size: Modifying canetti et al.'s construction," In Australasian Conference on Information Security and Privacy, pp. 28-44, 2018.

[15] M. Bellare and P. Rogaway, "Entity authentication and key distribution," In Annual international cryptology conference, pp. 232-249, 1993

〈저자소개〉

오 정 민 (Jeong Min Oh) 학생회원
2011년 2월: 충북대학교 전자공학과 졸업
2017년 9월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 정보보호 프로토콜 설계 및 분석, 센서 네트워크 보안, 블록체인 기술

정 익 래 (Ik Rae Jeong) 종신회원
1998년 2월: 고려대학교 전산학과 졸업
2000년 2월: 고려대학교 정보보호학과 석사
2004년 8월: 고려대학교 정보보호학과 박사
2008년 3월~현재: 고려대학교 정보보호대학원 조교수, 부교수, 교수
〈관심분야〉 프라이버시 향상 기술, 데이터베이스 보안, 생체인증

변 진 욱 (Jin Wook Byun) 종신회원
2001년 2월: 고려대학교 전산학과 졸업
2003년 2월: 고려대학교 정보보호학과 석사
2006년 8월: 고려대학교 정보보호학과 박사
2006년 11월~2007년 12월: Royal Holloway University of London 박사후 연수
2008년 3월~현재: 평택대학교 정보통신학과 부교수
〈관심분야〉 사용자 인증, 암호 프로토콜, 데이터베이스 보안, 프라이버시 보호 기술