Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.6, Dec. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.6.1243

IoT 환경의 MIPUF 기반 그룹키 관리 시스템 개선*

탁 금 지,[1†] 정 익 래,[1] 변 진 욱[2‡]
[1]고려대학교 정보보호대학원, [2]평택대학교

An Enhanced System of Group Key Management Based on MIPUF in IoT*

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

요 약

IoT 환경이 등장함에 따라 여러 스마트기기를 통해 소비자에게 편리함과 다양한 서비스를 제공하고 있다. 하지만 개인 정보 유출과 같은 보안 위협 사례가 보고되면서 IoT 환경상의 보안 문제의 중요성이 대두되고 있고 특히 키 관리의 안전성 문제에 대해서 거론되면서 PUF를 활용한 방안이 대응 방안으로 거론되고 있다. 키 관리 문제와 관련하여 그룹 키 관리 시스템의 안전성 문제에 대해서 MIPUF를 이용한 키 관리 프로토콜이 제안된 바가 있다. 해당 시스템은 경량 IoT 환경에 적용할 수 있고 PUF의 안전성으로 인해 전체 시스템의 안전성을 보장하지만, 일부 과정에서 안전성 및 연산의 효율성 측면에서 한계점을 보인다. 본 논문은 구성원에 대한 인증 추가, 데이터 간의 독립성 보장, 불필요한 연산을 축약, 그리고 데이터베이스 검색의 효율 증대함으로써 기존 프로토콜을 개선한다. 특정 공격에 대해 안전성 분석을 진행하고 연산량 비교를 통한 효율성 분석 결과를 제시한다. 이를 통해 본 논문은 데이터에 대한 신뢰도를 향상하고 본 논문이 제안한 방안이 기존 프로토콜보다 더 경량화 시켰음을 보인다.

ABSTRACT

With the emergence of the IoT environment, various smart devices provide consumers with the convenience and various services. However, as security threats such as invasion of privacy have been reported, the importance of security issues in the IoT environment has emerged, and in particular, the security problem of key management has been discussed, and the PUF has been discussed as a countermeasure. In relation to the key management problem, a protocol using MIPUF has been proposed for the security problem of the group key management system. The system can be applied to lightweight IoT environments and the safety of the PUF ensures the safety of the entire system. However, in some processes, it shows vulnerabilities in terms of safety and efficiency of operation. This paper improves the existing protocol by adding authentication for members, ensuring data independence, reducing unnecessary operations, and increasing the efficiency of database searches. Safety analysis is performed for a specific attack and efficiency analysis results are presented by comparing the computational quantities. Through this, this paper shows that the reliability of data can be improved and our proposed method is lighter than existing protocol.

Keywords: Internet of Things, physical unclonable function, multistage interconnection physical unclonable function, key management, key distribution

Received(09. 24. 2019), Modified(12. 04. 2019), Accepted(12. 04. 2019)

* 본 논문은 2019년도 한국정보보호학회 하계학술대회에 발표한 우수논문을 개선 및 확장한 것임.

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

† 주저자, gjtak57@korea.ac.kr

‡ 교신저자, jwbyun@ptu.ac.kr(Corresponding author)

I. 서 론

IoT(Internet of Things) 환경이 등장함에 따라 스마트 홈 가전제품, 스마트카드, 커넥티드 카(connected car) 등 생활에 편리함을 제공하는 다양한 기기가 등장하고 소비자에게 다양한 서비스를 창출하고 있다[1][2]. 하지만 IoT의 개인 정보 유출 및 프라이버시(privacy)의 침해 등 IoT 환경상의 해킹 공격 및 보안 위험 사례[3][4][5]가 꾸준히 보고되고 있으며 동시에 IoT 환경 내 개인정보보호 및 프라이버시 보존과 관련된 보안 문제가 중요하게 다루어지고 있다.

특히 안전성을 보장하기 힘든 환경에서 암호화키(Key) 관리 시스템의 안전한 키 관리 및 키 경량화 이에 대해 하드웨어 기반 보안 기술인 PUF(Physical Unclonable Function)가 소프트웨어 기반 보안 솔루션(solution)이 가진 구조적 문제를 해결함과 동시에 IoT 기기에 손쉽게 적용할 수 있다는 장점으로 하나의 해결방안으로 대두되고 있다.

IoT 환경상의 키 관리 시스템과 관련된 논제 중 하나인 그룹 키(group key) 관리 시스템의 안전성을 보장하는 문제에 대한 다양한 제안들에서도 PUF(Physical Unclonable Function)가 거론되고 있다. 그중 MIPUF(Multistage Interconnected PUF)를 이용하여 그룹 구성원 간에 안전하게 그룹 키를 분배, 저장, 그리고 구성원 변동에 따른 키 재분배하는 방안이 Hongxiang Gu에 의해 제시된 바가 있다[6]. 해당 제안은 경량화가 중요하게 여겨지는 IoT 환경에서 쉽게 적용할 수 있고 PUF와 다중 연결 네트워크(multistage interconnection network)의 안전성에 기반을 둔 MIPUF의 안전함으로 전체 시스템의 안전성을 보장한다고 주장하였다. 하지만 키 분배 과정상에서 암호화 연산의 효율성, 데이터 검색 효율성 및 데이터의 안전성, 그룹 구성원에 대한 인증, 키 재분배 과정에서 기존 그룹 키와 새로운 그룹 키 간의 연관성 등의 측면에서 안전성과 연산 효율성의 한계점이 존재한다.

본 논문은 MIPUF를 이용한 키 관리 시스템 프로토콜의 키 분배 과정에서의 인증의 부재, challenge의 독립성 및 CRP관리 문제, 키 재분배 과정에서 그룹 키 노출 위험성의 문제와 같은 안전성에 문제가 있음을 서술한다. 그리고 키 분배 과정에서 암호화 연산 문제 및 데이터베이스 검색과정의 문제 등 효율성 측면에서의 문제점도 같이 서술한다. 각 문제에 관한 개선방안을 차례로 제시한 뒤 두 프로토콜에 대해 특정 공격에 대한 안전성 분석 및 통신비용, 저장 공간, 에너지 소모량 등의 측면에서 효율성을 분석하고자 한다. 본 논문은 프로토콜 수행 과정에서 참여하는 구성원의 신원확인을 통한 정보에 대한 신뢰도 상승 및 연산의 축약으로 인한 프로토콜 경량화 및 에너지 절약에 대해 개선함을 보이는 데에 의의가 있다.

본 논문은 다음과 같은 구성을 갖춘다. 2장에서는 본 논문을 이해하는데 필요한 배경지식 및 프로토콜 서술에 사용될 기호에 대해 언급한다. 3장에서는 MIPUF를 이용한 Gu의 키 관리 프로토콜을 설명한다. 4장에서는 Gu 프로토콜의 문제점을 개선한 프로토콜을 제안한다. 5장에서는 기존 시스템의 프로토콜과 본 논문에서 제안하는 프로토콜의 안전성 및 효율성 비교를 진행한다. 6장에서는 본 논문의 결론을 제시한다.

II. 배경지식 및 기호 정의

이 장에서는 본 논문의 내용과 관련된 배경지식에 대해서 언급하고 앞으로 서술할 프로토콜에서 사용할 기호에 대해서 명시한다.

2.1 PUF(Physical Unclonable Function)

PUF는 반도체 제조 과정에서 발생하는 고유한 물리적 변형을 기반으로 생성되는 마이크로프로세서와 같은 반도체 장치의 고유 식별자이다. PUF는 물리적 미세 구조의 독창성에 달려 있고 이 미세 구조는 제조 과정에서 발생하는 임의의 물리적 요인에 따라 달라진다[7]. 이로 인해 최종 결과를 예측할 수 없고 특정 값으로 유도할 수 없어서 복제하는 것이 사실상 불가능하다. PUF에 대한 대표적인 특성은 아래의 Table 1.과 같다[8][9]. 이러한 PUF의 특성을 바탕으로 암호화, 전자서명 등 다양한 보안 기술에 이용되며 특히 인증 및 복제방지를 위한 장치로 활용되고 있다[10].

Table 1. Representative Properties of PUF.

Property	Description
Unclonability	The difference between expected responses of two different PUF instances to the same challenge should be enough large.
Unpredictability	It is infeasible to predict the response of a specific PUF instance to an unknown challenge, even if a certain number of previous challenge-response pairs of the PUF instance can be obtained.
Robustness	The difference between two distinct responses of a specific PUF instance to the same challenge should be small.
Tamper-evident	Any unauthorized attempt to access the PUF instance changes its challenge-response behavior.

2.2 MIPUF(Multistage Interconnected PUF)

MIPUF는 다중 연결 네트워크의 개념을 응용하여 특정 PUF 단위구조를 일렬로 연결한 것으로 차후 소개할 Gu의 프로토콜의 핵심 요소이다[6]. MIPUF를 구성하는 특정 PUF 단위구조는 크게 PE(Processing Elements)와 SE(Switching Elements)로 나눌 수 있다.

PE는 MIPUF의 노드(node)로 단일 혹은 복합으로 이루어진 PUF 구조물이다. Gu의 프로토콜에서는 구현의 용이성을 위해 k-bit의 challenge를 입력으로 m-bit의 response를 출력을 얻게하는 PUF를 병렬로 구현한다.

SE는 multiplexer의 집합으로 구성되어 있고 입력 비트값의 라우팅(routing)과 신호 교환(signal switching) 기능을 담당한다. MIPUF는 SE에 특정 상호연결 환경 설정값(the interconnection configuration)을 multiplexer의 입력으로 받아 SE의 기능을 수행하며, 하나의 SE는 PE의 이전 CRP(Challenge-Response Pair)의 response를 입력으로 받아 다음 CRP의 challenge를 출력한다.

Gu의 프로토콜에서 MIPUF는 2x2 SEs로 구성된 Omega network 형식으로 설계했다[11]. 두 개의 MIPUF 노드가 서로 N x N 연결이 되어 있다고 할 때, 2^N가지의 다른 switching 경로를 제공할 수 있고, 이에 대한 시간복잡도는 $O(N\log(N))$이다. 프로토콜에서 사용하는 MIPUF는 포트(port)의 재배치는 허용하지 않고 하나의 입력값에 대해 오로지 하나의 출력값이 대응될 수 있다. 여러 개의 PE와 SE로 구성된 MIPUF의 전체적인 구조도는 아래의 Fig. 1.과 같다.

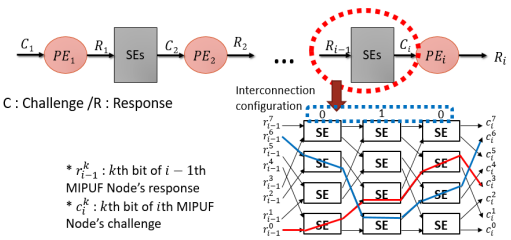


Fig. 1. The Architecture of MIPUF.

2.3 사용 기호 정의

이 문단에서는 앞으로 소개할 Gu의 프로토콜에 사용할 기호들의 목록에 대해 정의한다. 구체적인 기호 및 세부 설명은 아래의 Table 2.과 같다.

Table 2. Notations description.

Notations	Description
G	Group
N_i	Identity of ith node in group G
θ_i	Interconnection configuration of N_i
θ_i'	Interconnection configuration of CU
c_i^t	Challenge of N_i (t is the interconnection configuration)
r_i^t	Response of N_i corresponding to challenge c_i^t (t is the interconnection configuration)
F^t_i	MIPUF of N_i (t is the interconnection configuration)
K_G	Group key of group G

Notations	Description
p_i	Hint for K_G sent by CU to N_i
f_i	Hint for a interconnection configuration sent by CU to N_i
M_1	Encrypted message
msg_i^k	Encrypted message sent by CU to N_i
msg_i^u	Encrypted message sent by N_i to CU
l	Bit length of N_i
$E_k(\bullet)$	AES encryption with key k
$D_k(\bullet)$	AES decryption with key k
$H(\bullet)$	Hash function
$\oplus$	XOR function
$\|\|$	Concatenation function

III. Gu의 프로토콜 소개

본 장에서는 기존 Gu의 프로토콜[6]을 소개한 후 해당 프로토콜이 가진 문제점을 안전성 측면과 연산 효율성 측면에서 서술한다. 추후 언급한 문제들에 대한 개선방안에 대해서 4장에서 서술한다.

프로토콜을 수행하는 주체는 중앙 서버 역할을 하는 CU(Control Unit)와 그룹에 소속된 구성원 노드(Node)들이다. Gu의 프로토콜은 크게 키 분배, 키 저장, 키 재분배 3가지 과정으로 구성되어 있다.

3.1 키 분배(Key Distribution)

키 분배는 CU가 생성한 그룹 키를 그룹 구성원 노드들에 전송하는 일련의 과정이다. 키 분배 과정은 준비 단계, 키 전송 단계, CRP(Challenge-Response Pair) 갱신 단계, 총 3가지 단계를 차례로 수행하여 진행한다.

3.1.1 준비 단계(Preliminary Phase)

CU가 그룹 G에 속한 i번째 구성원 노드의 식별자 N_i를 임의로 생성하여 i번째 구성원에 전송한다. i번째 구성원은 받은 식별자 N_i를 해시함수의 입력값으로 사용해서 얻은 출력값을 상호연결 환경 설정값 θ_i를 얻는다. 그 후, challenge $c_i^{\theta_i}$를 선별하여 MIPUF를 통해 $c_i^{\theta_i}$에 대응하는 response $r_i^{\theta_i}$를 얻는다. i번째 구성원은 데이터베이스(database)에 $(c_i^{\theta_i}, \theta_i)$을 저장하고 CU에 $(\theta_i, c_i^{\theta_i}, r_i^{\theta_i})$을 전송한다. CU는 i번째 구성원으로부터 받은 값을 데이터베이스에 저장해둔다.

3.1.2 키 전송 단계(Key Delivery Phase)

CU는 새로운 상호연결 환경 설정값 θ_i'을 임의로 생성한다. i번째 구성원으로부터 받은 값 $(\theta_i, c_i^{\theta_i}, r_i^{\theta_i})$을 이용해 그룹 키 K_G의 힌트값 $p_i = r_i^{\theta_i} \oplus K_G$와 상호연결 환경 설정값의 힌트값 $f_i = r_i^{\theta_i} \oplus \theta_i'$를 계산한다. $r_i^{\theta_i}$를 AES 암호화 키로 이용하여 N_i, p_i, f_i를 암호화한 메시지 $M_1 = E_{r_i^{\theta_i}}(N_i || p_i || f_i)$을 얻는다. 그 후, N_i, $c_i^{\theta_i}$에 대한 해시값 $H(N_i || c_i^{\theta_i})$을 M_1와 이어 붙여 $msg_i^k = M_1 || H(N_i || c_i^{\theta_i})$를 생성하고 i번째 구성원에 전송한다.

3.1.3 CRP 갱신 단계(CRP update Phase)

i번째 구성원은 CU로부터 받은 msg_i^k에서 M_1과 $H(N_i || c_i^{\theta_i})$을 얻어내고 $r_i^{\theta_i}$를 AES 복호화 키로 이용해 M_1에 대한 복호화를 진행하여 N_i', p_i, f_i를 얻는다. N_i', $c_i^{\theta_i}$에 대한 해시값 $H(N_i' || c_i^{\theta_i})$과 $H(N_i || c_i^{\theta_i})$를 비교하여 서로 같은지 확인하고 만약 같지 않다면 프로토콜 진행을 멈추고 종료한다. 등식이 성립한다면 i번째 구성원은 $c_i^{\theta_i}$를 해시의 입력값으로 사용하여 새로운 challenge $c_i^{\theta_i'} = H(c_i^{\theta_i})$를 생성하고 MIPUF를 이용해 그에 대응하는 response $r_i^{\theta_i'}$를 얻는다. 그 후, $r_i^{\theta_i}$를 AES 암호화 키로 이용하여 N_i, $c_i^{\theta_i'}$, $r_i^{\theta_i'}$를 암호화한 메시지 $msg_i^u = E_{r_i^{\theta_i}}(N_i || c_i^{\theta_i'} || r_i^{\theta_i'})$를 생성한 후 CU에 전송한다.

CU는 $r_i^{\theta_i}$를 AES 복호화 키로 이용해 i번째 구성원으로부터 받은 msg_i^u에 대한 복호화를 진행하여 N_i, $c_i^{\theta_i'}$, $r_i^{\theta_i'}$를 얻어낸다. CU의 데이터베이스에 저장된 $c_i^{\theta_i}$의 해시값 $H(c_i^{\theta_i})$과 $c_i^{\theta_i'}$이 서로 같은지 확

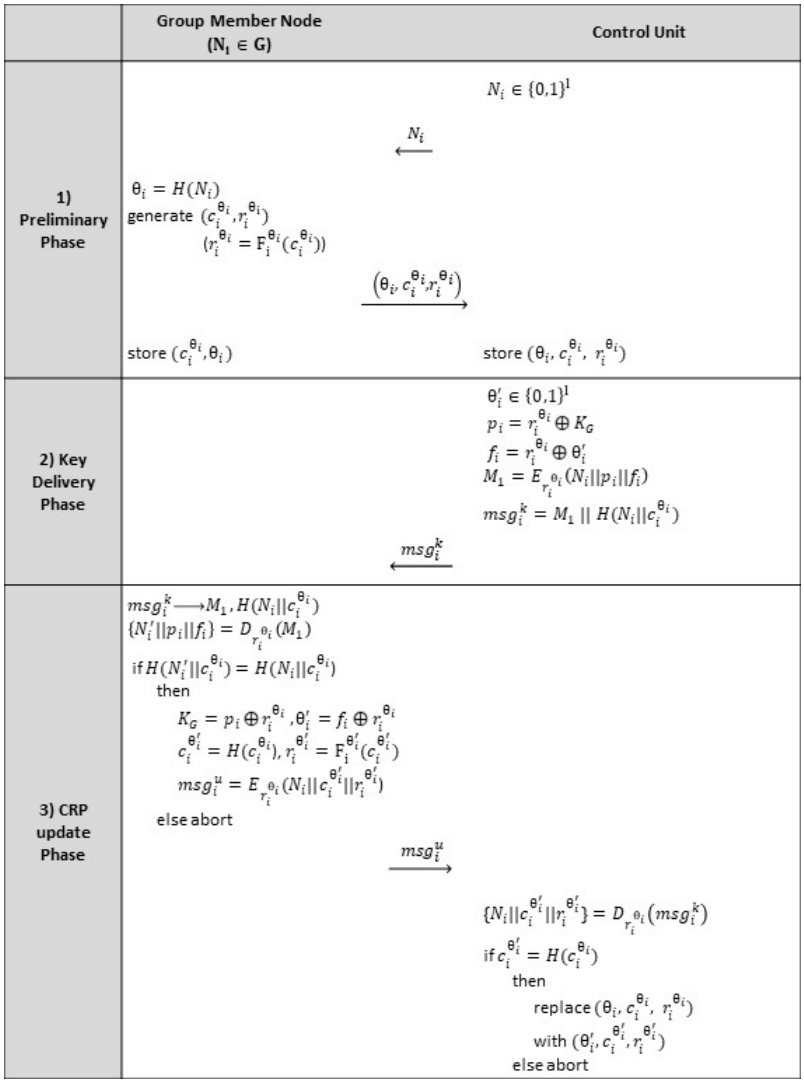


Fig. 2. The Key Distribution of Gu's protocol.

인하여 만약 같지 않다면 프로토콜 수행을 중단한다. 등식이 성립한다면 데이터베이스에 저장된 값 $(\theta_i, c^{\theta_i}_i, r^{\theta_i}_i)$을 새로운 값 $(\theta_i', c^{\theta_i'}_i, r^{\theta_i'}_i)$을 저장하고 프로토콜 수행을 종료한다.

3.2 키 저장(Key Storage)

3.1의 키 분배 과정을 마치면 그룹 구성원 노드는 CU로부터 받은 그룹 키 K_G를 메모리에 바로 저장하지 않고 그룹 키 힌트값 p_i를 계산한 후 저장한다.

3.3 키 재분배(Rekeying)

키 재분배는 그룹의 구성원 노드의 수에 변화가 있을 시, 기존에 사용했던 그룹 키를 폐기하고 새롭게 생성한 그룹 키를 구성원들에게 전달하는 과정이다. 이때 그룹에 새로운 구성원이 추가되는 경우와 기존의 구성원이 그룹을 탈퇴하는 경우 수행하는 과정에 차이가 있다.

3.3.1 새로운 구성원 추가

그룹 G에 새로운 구성원을 추가할 경우, CU는

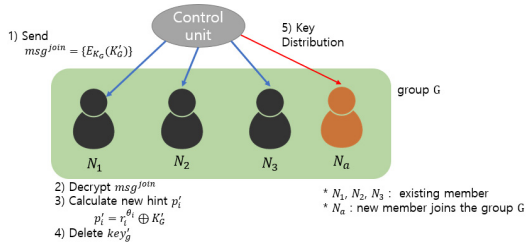


Fig. 3. Rekeying of Gu's protocol - New member joins the group.

새로운 그룹 키 K_G'를 생성하고 기존에 사용한 그룹 키 K_G를 AES 암호화 키로 이용하여 그룹에 속한 기존의 모든 구성원에게 메시지 $msg^{join} = E_{K_G}(K_G')$를 전송한다. 각 구성원은 기존 그룹 키 K_G를 AES 복호화 키로 이용하여 msg^{join}에서 새로운 그룹 키 K_G'를 얻어낸다. 그룹 키 K_G'를 이용하여 새로운 그룹 키 힌트 $p_i = r_i^{\theta_i} \oplus K_G'$를 계산하고 기존 그룹 키 K_G는 삭제한다. 그 후 CU와 그룹에 새로 들어오는 구성원은 4.1의 키 분배 과정을 통해 그룹 키 K_G'를 전달한다.

3.3.2 기존 구성원 탈퇴

그룹 G 에 있던 기존 구성원이 탈퇴할 경우, 먼저 그룹 G를 m개의 부분집합으로 나눈다. 이때, i번째 부분집합 $g_i, 1 \le i \le m$에 속한 구성원들은 서로 같은 상호연결 환경 설정값 θ_{g_i}를 공유한다.

CU는 새로운 그룹 키 K_G'를 계산하고 부분 그룹 g_i의 θ_{g_i}를 AES 암호화 키로 이용하여 메시지 $msg^{leave} = E_{\theta_i}(K_G') || H(\theta_{g_i})$를 생성한다. 그 후, 탈퇴할 구성원이 속한 부분 그룹 g_k를 제외한 나머지 부분 그룹들에 메시지 msg^{leave}를 전송한다.

부분 그룹 g_i에 속한 구성원들은 msg^{leave}로부터 $E_{\theta_i}(K_G')$, $H(\theta_{g_i})$를 얻고 θ_{g_i}를 AES 복호화 키로 이용하여 새로운 그룹 키 K_G'를 얻는다. 구성원이 알고 있는 θ_{g_i}을 해시함수의 입력값으로 사용하여 얻은 출력값과 메시지 msg^{leave}에서 얻은 $H(\theta_{g_i})$을 서로 비교하여 일치하면 그룹 키와 상호연결 환경 설정값을 갱신한다.

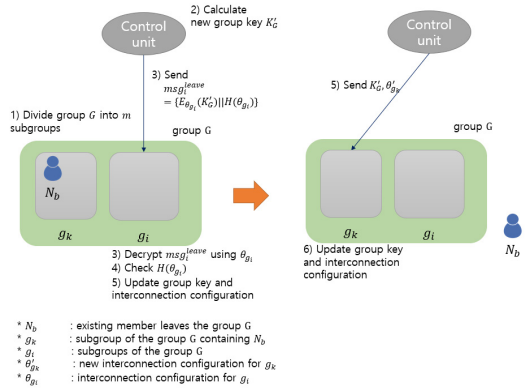


Fig. 4. Rekeying of Gu's protocol - Existing member leaves the group.

탈퇴할 구성원이 탈퇴절차를 마친 후, CU는 부분 그룹 g_k에게 새로 생성한 그룹 키 K_G'와 부분 그룹 g_k의 새로운 상호연결 환경 설정값 θ_{g_k}'을 전송한다.

3.4 Gu의 프로토콜의 문제점

본 문단은 앞서 3.1 ~ 3.3 문단에서 설명한 Gu의 프로토콜 3가지 과정 중, 키 분배 과정과 키 재분배 과정(새로운 구성원 추가)에서 어떤 문제가 있는지 서술한다. 본 논문에서 지적하고자 하는 문제는 불필요한 암호화 연산, 데이터베이스 검색과정의 비효율성, 구성원에 대한 인증 부재, 이전 challenge와 새로운 challenge 간의 독립성 약화 및 새로운 CRP쌍 갱신 과정 부재, 그리고 키 재분배 과정에서의 키 유출 가능성 등 5가지이다.

3.4.1 불필요한 암호화 연산

3.1의 키 분배 과정의 2번째 단계인 키 전송 단계에서 CU는 msg_i^k의 일부인 M_1을 계산할 때 그룹 키 힌트 p_i와 환경 설정값 힌트 f_i을 계산한다. 이때 p_i는 그룹 키 K_G을, f_i는 상호연결 환경 설정값 θ_i'을 외부에 노출되는 것을 방지하기 위해 XOR 연산을 이용한다. 하지만 M_1을 생성할 때 AES 암호화 연산을 이용하므로 XOR 연산으로 정보를 숨기는 효과가 중복되므로 p_i와 f_i을 계산하는 것은 불필요한 전력 낭비 등 불필요한 비용을 낭비할 위험이 있다고 판단된다.

3.4.2 데이터베이스 검색과정의 비효율성

3.1의 키 분배 과정의 3번째 단계인 CRP 갱신 단계에서 CU는 그룹 구성원이 보낸 메시지 msg_i^u를 복호화해서 N_i, $c_i^{\theta_i'}$, $r_i^{\theta_i'}$ 를 얻어야 한다. 이때 복호화에 사용하는 키는 $r_i^{\theta_i}$이므로 CU는 데이터베이스에서 i번째 구성원 노드에 대응되는 $(\theta_i, c_i^{\theta_i}, r_i^{\theta_i})$을 찾은 후 파악해야 한다. 하지만 Gu의 프로토콜에서 데이터베이스에 저장되는 데이터 형태로는 i번째 구성원 노드에 대응되는 데이터를 식별할 수 없다. 따라서 데이터베이스의 모든 레코드(record)를 하나씩 열람해서 얻은 response로 복호화를 수행해야 한다. 이는 만약 그룹에 포함된 구성원의 수가 많을수록 데이터베이스 검색에 걸리는 시간이 많아진다는 문제점이 있어 이에 관한 개선 사항이 필요하다.

3.4.3 구성원 노드에 대한 인증의 부재

3.1의 키 분배 과정의 3번째 단계인 CRP 갱신 단계에서 i번째 구성원 노드는 메시지 msg_i^u를 CU에게 전송한다. 여기서 msg_i^u를 생성할 때 암호화 연산만 수행하고 해당 메시지를 정말로 i번째 구성원 노드가 보낸 것이 맞는지 보장해주는 인증을 수행하지 않는다. 이는 만약 공격자가 중간에 msg_i^u를 가로챈 후, 마치 자신이 i번째 구성원 노드인 것처럼 가장하여 메시지를 조작한다면 CU는 i번째 구성원 노드가 보낸 메시지인지 확인할 방법이 없다. 따라서 공격자가 의도한 데이터로 데이터베이스의 데이터를 갱신하게 된다는 위험이 존재하기 때문에 이에 대해 개선할 필요가 있다.

3.4.4 이전 challenge와 새로운 challenge 간의 독립성 약화 및 새로운 CRP쌍 갱신 과정 부재

3.1의 키 분배 과정의 3번째 단계인 CRP 갱신 단계에서 i번째 그룹 구성원은 새로운 challenge $c_i^{\theta_i'}$을 얻기 위해 해시함수의 입력값으로 $c_i^{\theta_i}$을 이용하여 $c_i^{\theta_i'} = H(c_i^{\theta_i})$을 계산한다. 이 계산식에는 이전에 사용했던 challenge와 앞으로 사용할 challenge 사이에 해시함수의 입력과 출력이란 관계가 부여된다. 이로 인해 이전 challenge 값과 향후 생성될 값 사이의 독립성이 지켜지지 않는다. 만일 공격자가 프로토콜 상에서 이용되었던 challenge에 대한 정보를 획득하면 해시함수를 이용해 앞으로 사용될 challenge에 대해 예측할 수 있고, 이는 공격자가 메시지 형성에 영향을 줄 수 있는 요소로 작용할 수 있다.

새로운 challenge를 생성한 후 구성원 노드는 새로 생성한 CRP쌍을 저장하지 않는다. 이는 후에 프로토콜에서 키 분배 과정을 다시 시행할 때 과거의 challenge를 바탕으로 새로운 값을 생성하므로 중복된 CRP쌍이 생성될 수 있다는 문제점이 존재한다.

3.4.5 키 재분배 과정에서 키 노출의 위험

3.3의 키 재분배 과정에서 그룹에 새로운 구성원이 추가될 시 3.3.1에서 언급한 바와 같이 CU는 새로운 구성원에게 새로운 그룹 키 K_G'을 전달할 때, 이전 그룹 키 K_G를 암호화 키로 사용하여 암호 메시지를 생성하여 전달한다. 이는 만약 공격자가 새로운 그룹 키를 담은 메시지(3.3.1에서 언급된 메시지 msg^{join})를 탈취하고 이전에 사용했던 그룹 키 K_G를 알아낼 경우, 두 그룹 키 K_G와 K_G' 사이의 연관성을 파악할 가능성을 제공한다는 문제점이 있으며 이에 대해 개선해야 할 필요성이 있다.

IV. 제안하는 프로토콜

본 장에서는 Gu의 프로토콜의 키 분배 과정과 키 재분배 과정에 대해 개선한 프로토콜을 소개한다.

키 분배 과정의 주요 개선사항으로는 암호화 연산 축약, CU의 데이터베이스 검색 연산 효율 향상, 그룹 구성원 노드에 대한 인증 추가, 새로운 challenge의 독립성(independence) 향상, CRP 갱신 단계에서 구성원이 저장한 challenge와 상호 연결 환경 설정값 갱신 등이 있다.

키 재분배 과정의 주요 개선사항으로는 새로운 구성원이 추가되는 경우에 적용되며, 기존 그룹 키 K_G를 이용한 키 전송 과정 폐지가 있다.

4.1 키 분배 과정 개선방안

4.1.1 암호화 연산축약

3.4.1에서 언급한 바와 같이 기존 프로토콜에서 XOR 연산과 AES 암호화 연산으로 중복되는 암호화 효과를 줄일 필요가 있다. 이를 위해 3.1의 키 분배 과정의 2번째 단계인 키 전송 단계에서 CU가 msg_i^k의 일부인 M_1를 계산할 때 p_i 대신 K_G, f_i 대신 θ_i'를 넣어 계산하도록 한다. 즉, $M_1 = E_{r_i^{\theta_i}}(N_i||K_G||\theta_i')$로 계산한다.

4.1.2 CU의 데이터베이스 검색 연산 효율 향상

3.1의 키 분배 과정의 3번째 단계인 CRP 갱신 단계에서 그룹 구성원은 θ_i'를 이용해 $msg_i^u = E_{r_i^{\theta_i}}(N_i||c_i^{\theta_i'}||r_i^{\theta_i'}||\theta_i')$를 계산한 후, $r_i^{\theta_i}, \theta_i$와 함께 해시값 $M_2 = H(r_i^{\theta_i}||msg_i^u||\theta_i)$를 계산한다. 그 후 메시지 $msg_i^{auth} = msg_i^u||\theta_i||M_2$를 생성하여 CU에 전송하면 CU는 θ_i를 이용해 데이터베이스를 검색한다. 이로 인해 CU는 기존 프로토콜처럼 모든 레코드를 적용하지 않고도 특정 그룹 구성원에 대응

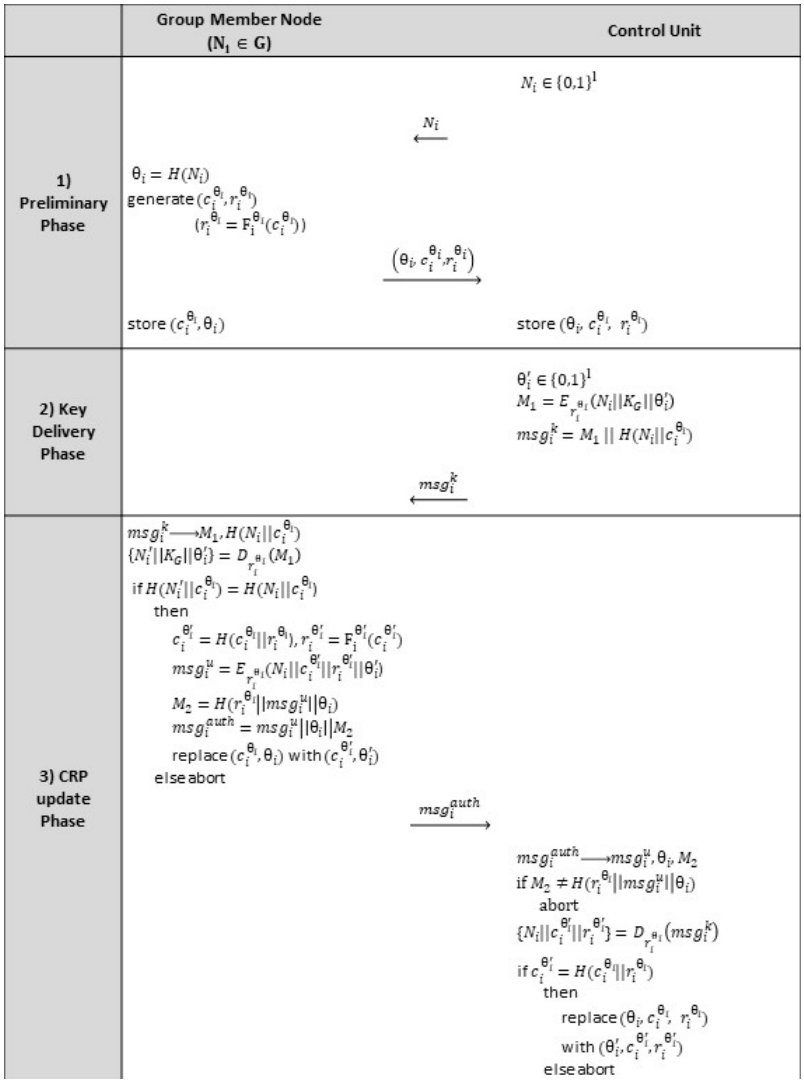


Fig. 5. The Key Distribution of proposed protocol.

되는 하나의 레코드를 찾음으로써 더 빠른 검색을 수행할 수 있다.

4.1.3 그룹 구성원 노드에 대한 인증 추가

3.1의 키 분배 과정의 3번째 단계인 CRP 갱신 단계에서 그룹 구성원은 4.1.2에서 언급한 인증 메시지 msg_i^{auth}를 생성하여 CU에 전송한다. CU는 msg_i^{auth}에서 M_2을 얻어낸 후, 데이터베이스 검색을 통해 알아낸 $r_i^{\theta_i}, \theta_i$과 msg_i^{auth}에 포함되어 있었던 msg_i^u를 이용해서 $H(r_i^{\theta_i}||msg_i^u||\theta_i)$를 계산하고 M_2와 비교하여 인증과정을 수행한다. $r_i^{\theta_i}$과 θ_i은 i번째 구성원 노드와 CU만이 아는 값이며, msg_i^u는 i번째 구성원 노드만이 만들 수 있고 해시함수의 안전성에 의해 인증을 수행할 수 있다.

4.1.4 이전의 challnege와 새로운 challenge 간의 독립성 향상

3.1의 키 분배 과정의 3번째 단계인 CRP 갱신 단계에서 그룹 구성원은 새로운 challenge $c_i^{\theta_i'}$를 계산할 때 해시함수의 입력값으로 $c_i^{\theta_i}, r_i^{\theta_i}$ 을 이용한다. 즉, $c_i^{\theta_i'} = H(c_i^{\theta_i}||r_i^{\theta_i})$로 계산한다. PUF의 특성으로 인해 구성원 노드 이외에는 $c_i^{\theta_i}$ 만으로 $r_i^{\theta_i}$ 을 유추하는 것은 어렵고 해시함수의 안전성에 의해 구성원 노드 이외의 제 3자가 새로운 challenge $c_i^{\theta_i'}$를 계산하는 것은 매우 어려우므로 이전의 challenge와 새로운 challenge 사이의 독립성을 향상할 수 있다.

4.1.5 그룹 구성원의 challenge와 상호연결 환경 설정값 갱신

기존 Gu의 프로토콜은 키 분배 과정의 3단계 CRP 갱신 단계에서 새로운 challenge를 생성한 후 따로 저장하지 않아 차후 새로운 challenge를 생성할 때 과거에 사용되었던 값이 생성될 수 있다.

이를 방지하기 위해 그룹 구성원은 3.1.3에서 언급한 msg_i^{auth}를 CU에 전송한 후 기존에 저장한 $(c_i^{\theta_i}, \theta_i)$을 $(c_i^{\theta_i'}, \theta_i')$로 교체하여 저장하도록 한다.

4.2 키 저장 과정 개선방안

3.1의 키 분배 개선방안의 4.1.1에 의해 p_i를 계산하지 않게 되므로 키 저장 과정을 전체 개선 프로토콜에서 제외한다.

4.3 키 갱신 과정 개선방안

앞서 3.4.5에서 언급했다시피, 그룹에 새로운 구성원을 추가할 때, 새로운 구성원 노드에만 이전에 사용했던 그룹 키 K_G를 암호키로 이용하여 새로운 그룹 키를 전송하는 것은 문제점이 있다. 따라서 그룹에 새로운 구성원을 추가할 시 CU는 기존의 그룹 키 K_G를 폐기한 후 새로운 그룹 키 K_G'를 생성한 후, 3.1의 개선된 키 분배 과정을 진행한다.

V. 안전성 및 효율성 분석

이 장에서는 Gu의 프로토콜과 본 논문이 제안한 프로토콜의 안전성과 효율성을 분석한다. 각각의 성질에 관하여 본 논문은 Gu 프로토콜의 가정을 이용하여 분석을 진행하였다.

5.1 안전성 분석

Gu의 프로토콜[6]에서 MIPUF는 물리적 보안 관점에서 안전하다고 가정한다. 다만 공격자는 MIPUF를 이용해 여러 개의 CRP 쌍을 획득할 수 있다고 가정한다. 키 분배 과정에서 예비단계를 제외한 모든 통신 과정에서 이용되는 무선 통신 채널(wireless channel)은 안전하지 않다고 간주한다. 그리고 각 구성원 노드마다 해시 연산 및 AES 암호화는 안전하게 수행되며 CU는 데이터베이스에 안전하게 키를 보관한다고 가정한다.

Gu의 프로토콜은 앞서 언급한 가정들을 바탕으로 도청 공격(eavesdropping attack), 중간자 공격(man-in-the-middle attack), 재생 공격(replay attack), 그리고 가장 공격(impersonation attack)에 대해 안전함을 보였다[6]. 본 논문은 Gu 프로토콜과 제안한 프로토콜에 대한 안전성 분석을 위해 Gu 프로토콜에서 사용한 가정들을 적용하여 Gu 프로토콜이 언급한 4가지 공격 중 중간자 공격과 가장 공격에 대해 안전한지 비교분석을 진행한다.

5.1.1 중간자 공격

Gu의 프로토콜의 키 분배 과정 3번째 단계에서 i번째 구성원은 새로운 challenge를 얻기 위해 기존 challenge $c_i^{\theta_i}$를 이용해 새로운 challenge $c_i^{\theta_i'} = H(c_i^{\theta_i})$를 계산한다. 그리고 키 분배 과정의 3번째 단계에서 i번째 구성원 노드는 기존에 저장한 데이터 $(c_i^{\theta_i}, \theta_i)$에 대한 갱신을 하지 않는다. 이를 이용하여 공격자는 i번째 구성원이 CU에게 보내는 메시지를 수집하면서 식 $c_i^{\theta_i'} = H(c_i^{\theta_i})$을 이용해 앞으로 사용될 challenge와 그에 대응하는 response를 얻는다. 처음 키 분배를 진행한 뒤 그 이후에 진행되는 키 분배 과정에서 i번째 구성원과 CU가 보내는 메시지를 중간에 가로채어 미리 얻은 CRP쌍을 적용하여 복호화를 시도한다. 복호화를 정상적으로 진행하게 하는 하나의 CRP쌍을 찾으면 이를 이용해 메시지를 조작하여 i번째 구성원이나 CU에게 전송함으로써 공격자가 의도하는 바대로 데이터를 조작할 수 있다. 이 공격이 가능한 이유는 향후 프로토콜을 수행해 얻은 challenge가 과거의 challenge와 연관성이 생기기 때문이다. 즉, 현재의 challenge를 해시함수에 적용하여 얻은 결과를 바탕으로 앞으로 사용될 challenge를 예측할 수 있기 때문이다.

한편 제안한 프로토콜은 Gu 프로토콜과 달리 새로운 challenge $c_i^{\theta_i'}$를 계산할 때 해시함수의 입력값으로 $c_i^{\theta_i}$, $r_i^{\theta_i}$을 이용한다. 이로 인해 challenge 사이에 연관성이 생기는 것을 방지하여 각 challenge의 독립성을 보장한다. 그리고 새로운 CRP를 계산한 후 과거 데이터 $(c_i^{\theta_i}, \theta_i)$를 새로운 데이터 $(c_i^{\theta_i'}, \theta_i')$로 갱신한다. 해시함수의 안전성과 MIPUF의 안전성에 의해 공격자는 과거의 데이터를 알 수 없다. 따라서 제안한 프로토콜은 중간자 공격에 안전하다고 볼 수 있다.

5.1.2 가장 공격

Gu의 프로토콜에서 공격자가 i번째 구성원 노드와 CU 사이에 전송되는 메시지를 가로채 가장 공격을 시도한다고 가정하자. 프로토콜의 키 분배 과정 3단계에서 공격자는 i번째 구성원이 전송한 메시지 msg_i^u를 가로채고 msg_i^u의 생성방법을 모방하여 공격자가 제작한 메시지 $msg_i^{u'}$을 제작한 뒤 CU에게 전송한다. CU는 공격자로부터 받은 메시지에 대해 데이터베이스의 레코드를 하나씩 열람해서 얻은 response를 복호화 키로 사용해 복호화를 진행한다. 그 후 $H(c_i^{\theta_i})$와 $c_i^{\theta_i'}$이 서로 같은지에 대한 검증 과정을 통과하여 공격자가 의도한 CRP쌍을 마치 i번째 구성원의 새로운 CRP쌍인 것처럼 조작할 수 있다. 이로 인해 공격자는 i번째 구성원인 것처럼 위장하여 CU에게 가장 공격을 수행할 수 있다. 이 공격이 가능한 이유는 CU가 받은 메시지가 정당한 i번째 구성원이 보낸 것이 맞는지 확인하는 인증을 수행하지 않기 때문이다.

한편 제안한 프로토콜은 Gu 프로토콜과 달리 i번째 구성원은 msg_i^{auth}를 생성하여 CU에게 전송하는데, 이때 msg_i^{auth}에는 $M_2 = H(r_i^{\theta_i} || msg_i^u || \theta_i)$가 포함되어 있다. 공격자가 Gu 프로토콜과 같은 방식으로 공격을 진행한다고 가정했을 때, 공격자는 i번째 구성원 노드의 $\theta_i, c_i^{\theta_i}, r_i^{\theta_i}$에 대한 정보를 알 수 없고 해시함수의 안전성에 의해 M_2를 조작하는 것은 어렵다. 따라서 CU가 진행하는 인증 단계를 통과하는 것은 어려우므로 제안한 프로토콜은 가장 공격에 대해 안전하다.

5.2 효율성 분석

앞서 3.4.1과 3.4.2에서 언급했듯이 기존 Gu의 프로토콜은 불필요하게 암호화 연산을 사용하고 데이터베이스 검색과정이 비효율적이라는 문제가 있다. 본 논문이 제안한 프로토콜은 불필요하게 적용한 암호화 연산 문제에 대해 4.1.1에서 언급한 바와 같이 키 분배 과정의 2번째 단계에서 CU는 힌트값 p_i, f_i 를 계산하지 않음으로써 연산량을 줄이도록 했다. 그리고 데이터베이스 검색의 효율성 문제에 대해서는 4.1.2에서 언급한 바와 같이 노드를 구별할 수 있도록 키 분배 과정의 3번째 단계에서 구성원 노드가 CU에 보내는 메시지 msg_i^{auth}안에 θ_i를 첨가하여 검색의 효율을 높이고자 하였다.

본 논문은 Gu 프로토콜에서 언급한 가정들을 바탕으로 제안한 프로토콜의 연산량을 계산하여 두 프로토콜의 효율성을 분석한다. Gu 프로토콜에서는 그룹 G은 N개의 구성원 노드가 있으며, 키 재분배 과정에서 그룹을 부분 그룹으로 나누면 그 부분 그룹의

개수는 M개라고 가정한다. Gu의 프로토콜에서 사용하는 데이터의 길이 기호 정의는 아래의 Table 3.과 같고, 사용하는 연산에 대한 에너지 소모량에 대한 기호 정의는 아래의 Table 4.과 같다.

Table 3. Notation of data length.

data	length(bit)
MIPUF's challenge	a
The interconnection configuration	b
MIPUF's response	$c(b \geq c)$
N_i(ith Node's identity)	l
input for hash function	a
group key	$c(b \geq c)$

Table 4. Notation of energy consumption.

Field	cost
MIPUF	E_P
Hash function	E_H
Random number generator	E_R
XOR operation	E_X
AES encryption	E_A

5.2.1 통신비용(Communication cost)

Gu 프로토콜에서 사용한 메시지 msg_i^k, msg_i^u, msg^{join}, msg_i^{leave}의 길이는 각각 $a+b+c+l$, $a+c+l$, c, $a+c$이다. 그리고 키 분배 단계에서는 $2N$, 키 재분배(구성원 추가) 단계에서는 $N+2$, 키 재분배(구성원 탈퇴) 단계에서는 $(\frac{N}{M}-1)+(M-1)$만큼 통신비용이 필요하다. 이때 키 재분배(구성원 탈퇴)의 경우 $M=\sqrt{N}$일 때, 통신비용은 최소가 된다.

제안한 프로토콜은 키 재분배(구성원 탈퇴)단계에는 개선된 사항이 없으므로 이 과정에 대한 분석은 생략한다. 제안한 프로토콜에서 사용한 메시지 msg_i^k, msg_i^{auth}의 길이는 각각 $a+b+c+l$, $2a+2b+c+l$이며, 키 재분배(구성원 추가) 단계는 모든 구성원이 키 재분배를 진행하므로 msg^{join}의 값을 따로 구하지 않는다. 이를 바탕으로 제안한 프로토콜에서 필요한 통신비용을 계산하면 키 분배 단계는 $2N$, 키 재분배(구성원 추가)단계는 $2(N+1)$이다.

제안한 프로토콜이 키 재분배(구성원 추가) 단계에서 비용이 Gu의 프로토콜보다 크다고 볼 수 있다. 하지만 이전 그룹 키와 향후 사용될 그룹 키의 연관성을 드러나지 않도록 하는 것이 시스템 운영에 있어 중요하므로 이를 위해 필요한 비용이라고 판단하였다.

5.2.2 저장 공간 비용(Storage overhead)

Gu 프로토콜에서 CU는 데이터베이스에 그룹 구성원 노드의 식별자, CRP, 상호연결 환경 설정값을 저장하므로 CU의 저장 공간 비용은 $N \bullet (a+b+2c+l)$이다. 각 구성원 노드는 노드의 식별자, challenge, 상호연결 환경 설정값, 그룹 키 힌트를 저장하므로 각 구성원 노드의 저장 공간 비용은 $a+b+c+l$이다.

제안한 프로토콜에서 CU는 Gu 프로토콜과 같은 값들을 저장하므로 CU의 저장 공간 비용은 $N \bullet (a+b+2c+l)$이다. 각 구성원 노드는 Gu 프로토콜과 다르게 그룹 키 힌트를 저장하지 않으므로 각 구성원 노드의 저장 공간 비용은 $a+b+l$이다.

제안한 프로토콜이 Gu의 프로토콜과 비교했을 때 구성원 노드의 저장 공간을 더 절약할 수 있고 기존보다 구성원 노드가 좀 더 경량화를 추구할 수 있다고 판단하였다.

5.2.3 에너지 소모량(Energy cost)

Gu 프로토콜의 키 분배 단계에서 CU는 한 개의 구성원 노드와 통신하는 동안 $2E_A+2E_H+2E_R+2E_X$만큼의 에너지를 소모한다. 각 구성원 노드는 $2E_A+2E_H+E_P+2E_R$만큼의 에너지를 소모한다. 키 재분배 단계(구성원 추가)에서 CU는 기존 구성원과 통신하는 과정에서 E_A+E_R, 새로운 구성원과 통신하는 과정에서 $2E_A+2E_H+2E_R+2E_X$만큼의 에너지를 소모한다. 기존 구성원은 통신하는 과정에서 $2E_A+2E_H+2E_R+2E_X$, 새로운 구성원은 E_A+E_X만큼의 에너지를 소모한다. 키 재분배 단계(구성원 탈퇴)에서 CU는 탈퇴하는 구성원이 없는 부분 그룹과 통신하는 과정에서 $(M-1) \bullet (E_A+E_H+E_R)$,

Table 5. Overhead Evaluation.

Communication cost(total number of messages)		
stage	protocol	
	Gu	proposed
key distribution	$2N$	$2N$
rekeying (join)	$N+2$	$2(N+1)$
rekeying (leave)	$(\frac{N}{M}-1)+(M-1)$ (minimum cost is achieved when $M=\sqrt{N}$)	$(\frac{N}{M}-1)+(M-1)$ (minimum cost is achieved when $M=\sqrt{N}$)

Storage overhead(bit)		
member	protocol	
	Gu	proposed
CU	$N \cdot (a+b+2c+l)$	$N \cdot (a+b+2c+l)$
ith node (N_i)	$a+b+c+l$	$a+b+l$

Energy cost			
stage	member	protocol	
		Gu	proposed
key distribution	CU	$2E_A+2E_H+2E_R+2E_X$	$2E_A+3E_H+2E_R$
	ith node (N_i)	$2E_A+2E_H+E_P+2E_R$	$2E_A+3E_H+E_P$
rekeying (join)	CU	E_A+E_R(existing node) $2E_A+2E_H+2E_R+2E_X$ (new node)	$2E_A+3E_H+2E_R$
	ith node (N_i)	$2E_A+2E_H+2E_R+2E_X$ (new node) E_A+E_X(existing node)	$2E_A+3E_H+E_P$
rekeying (leave)	CU	$(M-1) \cdot (E_A+E_H+E_R)$ (existing members in g_i) $(\frac{N}{M}-1) \cdot (2E_A+2E_H+2E_R+2E_X)$ (existing members in g_k)	$(M-1) \cdot (E_A+E_H+E_R)$ (existing members in g_i) $(\frac{N}{M}-1) \cdot (2E_A+2E_H+2E_R+2E_X)$ (existing members in g_k)
	ith node (N_i)	$(M-1) \cdot (E_A+E_H+E_R)$ (existing members in g_i) E_A+E_H (existing members in g_k)	$(M-1) \cdot (E_A+E_H+E_R)$ (existing members in g_i) E_A+E_H (existing members in g_k)

탈퇴하는 구성원이 포함된 부분 그룹과 통신하는 과정에서 $(\frac{N}{M}-1) \cdot (2E_A+2E_H+2E_R+2E_X)$만큼의 에너지를 소모한다. 탈퇴하는 구성원이 없는 부분 그룹에 속하는 모든 구성원은 통신 과정에서 $(M-1) \cdot (E_A+E_H+E_R)$, 탈퇴하는 노드가 포함된 부분 그룹에 속하는 모든 구성원(단, 탈퇴하는 구성원은 제외한다.)은 통신 과정에서 E_A+E_H만큼의 에너지를 소모한다.

제안한 프로토콜은 키 분배 단계에서 CU는 힌트 값을 계산하기 위한 XOR 계산을 하지 않고 노드의 인증을 위한 연산을 진행한다. 그리고 각 노드는 인

증 절차를 위해 해시 연산을 추가로 진행한다. 따라서 CU는 한 개의 구성원 노드와 통신하는 동안 $2E_A+3E_H+2E_R$만큼의 에너지를 소모한다. 그리고 각 구성원 노드는 $2E_A+3E_H+E_P$만큼의 에너지를 소모한다. 키 재분배(구성원 추가)단계에서 모든 구성원은 키 분배 과정을 진행하므로 CU와 구성원 노드들의 에너지 소모량은 키 분배 단계에서 소모하는 양과 같다. 키 재분배(구성원 탈퇴)단계는 Gu의 프로토콜과 같은 방법으로 진행되므로 Gu의 프로토콜과 같은 값의 에너지 소모량이 계산된다.

제안한 프로토콜은 Gu의 프로토콜과 비교했을 때, 키 분배 과정에서 CU는 $2E_X-E_H$, 구성원 노드는 $2E_R-E_H$ 만큼의 에너지를 절약함을 알 수 있다. 또한, 키 재분배(구성원 추가)단계에서 기존의 구성원 노드에 대해서 CU는 $E_A+4E_H-E_X$을 더 사용하고, 새로 추가되는 구성원은 $2E_R+2E_X-E_H$ 만큼의 에너지를 절약하며, 그리고 기존 구성원은 $E_A+3E_H+E_P-E_X$만큼이 에너지를 더 사용한다. 이를 통해 제안한 프로토콜은 Gu 프로토콜보다 키 분배 과정에서 에너지 소모가 적고 키 재분배(구성원 추가) 과정에서 비용이 Gu의 프로토콜보다 에너지 소모가 더 크다고 볼 수 있다. 하지만 5.2.1에서 언급했던 바와 같이 이전 그룹 키와 향후 사용될 그룹 키의 연관성을 유추하는 것을 방지하기 위한 비용이라고 판단하였다.

5.2.4 데이터베이스 검색 시간복잡도

3.4.2.에 언급된 문제점으로 인해 그룹 G의 N개의 구성원 노드에 대해서 Gu의 프로토콜의 경우 $O(N)$만큼의 시간복잡도가 요구된다. 한편 제안한 프로토콜의 경우 4.1.2에 언급된 개선방안과 같이 상호연결 환경 설정값을 이용해서 데이터베이스의 레코드를 검색하므로 $O(1)$만큼의 시간복잡도가 요구된다.

VI. 결 론

본 논문은 Gu의 MIPUF를 이용한 IoT 환경상의 그룹 키 관리 프로토콜의 문제점을 키 분배 과정, 새로운 구성원이 추가될 때의 키 재분배 과정에서 드러나는 안전성 문제점을 제기하였다. 이에 대한 프로토콜 개선방안으로 키 분배 과정에서는 암호화 연산 축약, CU의 데이터베이스 검색 연산 효율 향상, 그룹 구성원 노드에 대한 인증 추가, 새로운 challenge의 독립성 향상, 구성원의 challenge와 상호연결 환경 설정값 갱신을 제안하였다. 그리고 새로운 그룹 구성원이 추가되는 경우의 키 재분배 과정은 모든 노드가 키 분배 과정을 실행하도록 설계하였다.

Gu의 프로토콜과 본 논문이 제안한 프로토콜을 대상으로 중간자 공격과 가장 공격에 대해 안전성을 분석하여 Gu의 프로토콜은 2가지 공격에 대해 모두 취약하고 제안한 프로토콜은 안전함을 보였다.

효율성 측면에서 제안한 프로토콜은 키 분배 과정에서 에너지 소모량, 구성원 노드의 저장 공간 비용, 데이터베이스 검색 연산, 측면에서 Gu의 프로토콜보다 비용이 감소하였다. 반면, 키 재분배(구성원 추가) 과정에서 통신비용, 에너지 소모량 측면에서는 Gu의 프로토콜보다 증가했으나 그룹 키의 안전성을 보장하는 것이 요구되므로 전체적인 측면에서 감수할 수 있는 비용이라고 판단하였다.

향후 키 재분배(구성원 추가) 과정의 통신비용, 에너지 소모량을 줄이면서 그룹 키의 안전성을 보장하는 연구가 향후 진행해야 할 과제이다.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645-1660, Sep. 2013.

[2] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," Business Horizons, vol. 58, no. 4, pp. 431-440, Aug. 2015.

[3] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," in Computer, vol. 50, no. 7, pp. 80-84, Jul. 2017.

[4] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT):

Taxonomy of security attacks," 2016 3rd International Conference on Electronic Design (ICED), pp. 321-326, Aug. 2016.

[5] J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 32-37, Feb. 2017.

[6] H. Gu and M. Potkonjak, "Efficient and Secure Group Key Management in IoT using Multistage Interconnected PUF," Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '18), pp. 1-6, Aug. 2018.

[7] P. Tuyls, B. Skoric, and T. Kevenaar, Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting, 1st Ed., Springer-Verlag London, Jan. 2007.

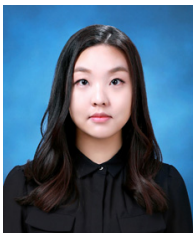
[8] R. Maes, Physically Unclonable Functions: Constructions, Properties and Applications, 1st Ed., Springer-Verlag Berlin Heidelberg, Nov. 2013.

[9] I. Verbauwhede and R. Maes, "Physically unclonable functions: Manufacturing variability as an unclonable device identifier," Proceedings of the 21st Edition of the Great Lakes Symposium on Great Lakes Symposium on VLSI, pp. 455-460, May. 2011.

[10] C. Herder, M. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," Proceedings of the IEEE, vol. 102, no. 8, pp. 1126-1141, Aug. 2014.

[11] D.H. Lawrie, "Access and Alignment of Data in an Array Processor," IEEE Transactions on Computers, vol. C-24, no. 12, pp. 1145-1155, Dec. 1975.

〈저자소개〉



탁 금 지 (Geum Ji Takg) 학생회원
2018년 2월: 이화여자대학교 컴퓨터공학과 졸업
2018년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 정보보호, 블록체인 보안, IoT 보안, 암호 프로토콜

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 박사 후 연수
2008년 3월~현재: 평택대학교 정보통신학과 조교수, 부교수, 교수
〈관심분야〉 사용자 인증, 암호 프로토콜, 데이터베이스 보안, 프라이버시 보호 기술