The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 15, No. 5, pp.9-17, Oct. 31, 2015. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2015.15.5.9
JIIBC 2015-5-2

M2M 통신 환경에서 트랩도어 충돌 해쉬를 이용한 그룹키 생성 및 교환 기법

Group Key Generation and Exchange Scheme using a Trapdoor Collision Hash in M2M Communications Environment

김성수*, 전문석*, 최도현**

Sung-Soo Kim*, Moon-Seog Jun*, Do-Hyeon Choi**

요 약 무선 통신 기술의 발전과 ICT 시장의 변화에 따라 M2M 서비스 활성화 및 기술은 지속적인 발전을 거듭하고 있다. 사람의 제어나 직접적인 개입 없이 사물과 사물간의 통신환경을 구축하는 M2M 환경이 주목받고 있다. 무선 통신 환경의 특성으로 데이터 노출, 위조, 변조, 삭제, 프라이버시 등의 문제에서 다양한 보안 위협에 노출 될 가능성과 안전한 통신 보안 기술이 중요 요구사항으로 이슈화되고 있다. 본 논문은 트랩도어 충돌 해쉬의 요구사항을 분석하고, 트랩도어의 특수성을 이용하여 M2M 환경에서 그룹간의 키를 생성하고, 이를 세션키로 교환하는 기법을 제안한다. 그리고 그룹키 생성에 이은 디바이스와 게이트웨이의 인증을 확인하는 기법을 제안한다. 제안하는 기법은 충돌 메시지와 충돌 해쉬의 특수성을 이용하여 그룹 통신 구간의 위장 공격, 중간자 공격, 재전송 공격 등의 공격 저항성을 가지는 안전한 기법임을 확인하였다.

Abstract The development of wireless communication technology and change in the ICT market has led to the development of the M2M service and technology. Under these circumstances, the M2M environment has been the focus of communication environment construction between machines without control or direct intervention of human being. With characteristics of wireless communication environment, the possibility of being exposed to numerous security threats and safe communication security technology have becoming an issue an important requirements for problems such as data exposure, forgery, modulation, deletion, and privacy. This research analyzes requirements of trapdoor collision hash, generates keys between groups under the M2M environment by using the specificity of trapdoor, and suggests technology to exchange keys with session keys. Further, it also suggests techniques to confirm authentication of device and gateway in accordance with group key generation. The techniques herein suggested are confirmed as safe methods in that they have attack resistance such as Masquerade Attack, Man-in-the-Middle Attack, and Replay Attack in the group communication block by using the speciality of collision message and collision hash.

Key Words : M2M, IoT, Trapdoor Collision Hash, Group Key

*정회원, 숭실대학교 일반대학원 컴퓨터학과
**정회원, 숭실대학교 일반대학원 컴퓨터학과(교신저자)
접수일자 : 2015년 8월 26일, 수정완료 2015년 9월 26일
게재확정일자 : 2015년 10월 9일

Received: 26 August, 2015 / Revised: 26 September, 2015 / Accepted: 9 October, 2015
**Corresponding Author: cdhgod0@ssu.ac.kr
Dept. of Computer Science and Engineering, Soongsil University, Korea

Ⅰ. 서 론

M2M(Machine-to-Machine) 기술은 사물과 사물 또는 사물과 사람 간에 정보를 수집하고 이를 처리하는 지능형 인프라를 의미한다. 유무선 통신의 ICT 기술을 결합하여 사물, 차량, 사람의 상태정보 등 다양한 정보를 수집하고 이를 활용하고 있다. 이는 센서 네트워크 등으로부터 데이터를 수집하여 가공하고 분배하는 서비스와 유사한 형태로, 사람의 개입 없이 디바이스 간에 스스로 서로 통신하며 정보를 교환하는 점에서 과거 센서 네트워크 서비스에 비해 발전된 서비스를 제공한다[1].

LTE, TISPAN, WCDMA, HSDPA, GSM, W-LAN 등 위성 통신과 이동통신 및 무선 인터넷, Zigbee, Bluetooth 등 저출력 통신기술과 융합하여 보다 넓은 영역과 서비스 범위를 확대하고 있다[2]. M2M 통신은 사용자에게 편리한 통신 환경과 다양한 서비스를 제공한다[3]. 하지만 현재 M2M 표준 디바이스는 보안 기능이 취약한 경우가 많아 중간자공격, 재전송공격 등 해킹 공격에 대해 뚜렷한 대응책이 없다[4]. 또한 M2M 서비스 환경에 따라 다수 디바이스 지원함에 따라 다양한 제조사가 연계되어, 이들 간에 신뢰성 입증 또한 문제가 되고 있다[5].

본 논문에서는 oneM2M 환경의 안전성을 위해 다수의 M2M 디바이스와 소수의 게이트웨이 간에 그룹키를 생성하고, 보안성을 제공하는 세션키 전송 프로토콜을 제안한다. 본 논문의 구성은 다음과 같이 2장은 관련연구, 3장은 제안 기법 및 프로토콜, 4장은 제안하는 기법과 프로토콜에 대해 분석하고, 마지막 5장에서 결론을 맺는다.

Ⅱ. 관련연구

본 장에서는 oneM2M 구조와 ITU-T의 M2M 보안 요구사항에 대해 다루고, 제안하는 그룹키 생성과 효율적으로 안정적인 통신 세션을 설정하기 위한 트랩도어 충돌 해쉬에 대해 설명한다.

1. oneM2M 아키텍처와 보안 요구사항

그림 1은 oneM2M 아키텍처 구성요소를 나타낸다[6]. 각 M2M 디바이스의 통신 노드는 최소 하나 이상의 AE(Application Entity)와 CSE(Common Service Entity)를 포함한다. CSE는 M2M 노드 중 실행되는 논리적인 엔티티의 CSFs(Common Service Functions)를 공통 M2M 서비스 기능으로 포함하고, oneM2M 아키텍처에서 정의하는 각 노드들은 등록, 연결에 대한 프라이버시 및 보안 서비스 제공, 관리 기능 등 구성요소에 대한 CSF를 정의하고 있다. 또한 CSF에서 정의하는 M2M 디바이스의 구성 및 위치, AE와 CSE의 유무에 따라 통신의 형태는 표 1과 같다.

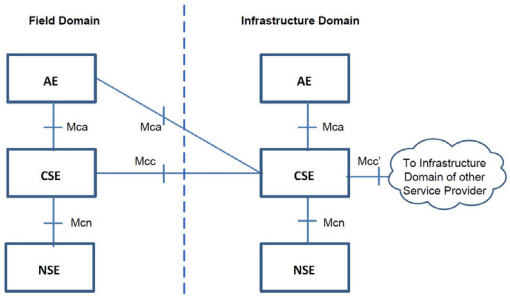

그림 1. oneM2M 아키텍쳐
Fig. 1. oneM2M Functional Architecture

- ADN(Application Dedicated Node) : M2M 어플리케이션을 포함하고, 제한적으로 서비스 로직만 포함
- ASN(Application Service Node) : M2M 어플리케이션과 공통 서비스 기능 포함
- MN(Middle Node): M2M 디바이스 노드와 네트워크를 연결 기능을 담당하는 일종의 게이트웨이
- IN(Infrastructure Node) : M2M 서비스 제공자

표 1. oneM2M 통신의 형태
Table 1. Types of oneM2M Communication

Type 1	ASN - MN(Middle) - IN
Type 2	ADN - MN(Middle) - IN
Type 3	ASN - Mcc - IN
Type 4	ADN - Mca - IN

통신 형태에 따라 다양한 M2M 디바이스가 존재 가능하며, 각 노드 사이는 저출력 통신 기술인 Zigbee, Bluetooth와 근거리 무선 통신 기술인 WLAN, WiFi 등의 D2D 기술, 그리고 셀룰러 통신 기술인 LTE, WCDMA 등의 무선 통신 기술을 사용한다[7]. M2M 서비스 계층의 기능적 요구사항은 ITU-T와 ETSI가 M2M

서비스와 어플리케이션 계층을 구분하여 표준화를 진행하고 있는 상태이다[8]. 표 2는 ITU-T의 서비스 계층 보안 요구사항을 나타낸다[9].

표 2. ITU-T 보안 요구사항
Table 2. ITU-T Security Requirement

Authentication	Service layer is required to provide authentication mechanisms for applications and devices and prevent unauthorized use of the devices.
Privacy	Service layer is required to support privacy protection capabilities, such as anonymity of identity and location, according to regulation and laws.
Confidentiality	Service layer is required to support data transfer confidentiality.
Integrity	Service layer is required to support data integrity protection.
Support of security for service scenarios involving multiple actors	Service layer is required to support security capabilities, such as supporting user access control of protected data, for M2M service scenarios involving multiple actors inside a single administrative domain and across different administrative domains (e.g., countries, operators).

보안 요구사항은 비인가 디바이스 및 어플리케이션의 인증, 신원확인, 위치정보 등과 같은 개인정보가 익명으로 보호되어야 하고, 데이터의 기밀성 유지, 무결성 보장, 다수 사용자에 대한 서비스 시나리오의 보안성을 제공해야 한다[9].

2. 트랩도어 충돌 해쉬

일반적으로 보안 프로토콜에서 사용하는 해쉬 함수는 메시지의 오류 및 변조 탐지용으로 무결성, 전자 서명과 함께 사용된다[10]. 트랩도어 충돌 해쉬 함수는 메시지에 대한 서명이 유효한지 검증하는 용도로 사용되고, 각 다른 서명에 대해서는 증명할 수 없다[11].

하지만 트랩도어 충돌 해쉬 함수는 송신자와 수신자가 충돌을 찾을 수 있는 특징을 가진 트랩도어 함수이므로 충돌 값을 알지 못하는 다른 사용자는 같은 트랩도어 충돌 해쉬 값을 계산 할 수 없는 특징이 있다.

표 3은 트랩도어 충돌 해쉬 함수의 보안 요구사항[12]을 나타내고, 충돌 저항성, 해쉬 결과 값에 대한 역원, 충돌 검증에 사용하는 비밀키 숨김, 비밀키 노출에 영향을 받지 않는 4가지 보안 요구사항을 충족해야 한다.

표 3. 트랩도어 충돌 해쉬 보안 요구사항
Table 3. Trapdoor Collision Hash Security Requirement

Collision-resistance	There is no efficient algorithm that given only PK, L, m and r, (but not the secret key SK) can find a second pair m, r.
Semantic Security	The chameleon hash value C does not reveal anything about the possible message m that was hashed.
Message Hiding	Assume the recipient has computed a collision using the universal forgery algorithm. By showing the second pair(m', r') without the need to open the original message it may correspond to invalid request.
Key Exposure Freeness	If a recipient with public key PK has never computed a collision under label L, then given C=Hash(PK, L, m, r) there is no efficient algorithm that can find a collision. (a second pair m, r mapping to the same digest C).

각 송수신자는 사전에 동의한 큰 소수 p, q, g를 공유하고 있고, 개인키와 공개키 쌍(SK, PK), 랜덤 메시지 m, 보조 랜덤값 r은 Z_q^*상의 값을 선택하여 이용할 수도 있다.

트랩도어 충돌 해쉬 함수의 다음과 같은 연산과정을 통해 충돌 값을 찾아 동일한 해쉬 값을 계산한다. 큰 소수 g, p를 생성하고 비밀키 x를 이용하여 공개키 y를 계산한다[13].

$$y = g^x \bmod p \quad (\text{단, } x \in Z_p^*) \tag{1}$$

초기 임의의 랜덤 메시지 m과 보조 랜덤값 r값을 선택 또는 생성하여, 식 (1)에서 계산한 공개키 y를 이용하여 식 (2)와 같이 트랩도어 충돌 해쉬값(C_y)을 계산한다.

$$C_y(m,r) = g^m y^r \bmod p \tag{2}$$

이후 $C_y(m,r)$와 동일한 충돌 해쉬 값을 계산하기 위한 새로운 충돌 메시지 m'은 새로운 랜덤값 r'과 계산되는 기존의 충돌 비밀키 $x \in Z_p^*$를 알고 있는 측에서만 계산이 가능하다. 동일한 충돌 해쉬 값을 계산하기 위한 충돌 메시지 m'은 식 (3)과 같이 계산할 수 있고, 식 (4)는 충돌 메시지의 특징을 나타낸다.

$$m' = m + x(r - r') \quad (3)$$
$$m + xr = m' + xr' \quad (4)$$

새로운 충돌 해쉬값 $C_y(m', r')$는 (m', r')에 대한 충돌 해쉬 값은 첫 임의의 랜덤 값 (m, r)쌍에서 $m' \neq m$일 때, 두 쌍의 메시지와 랜덤 값의 충돌 해쉬 값은 동일한 값임을 다음 식 (5)에서 확인 가능하다.

$$\begin{aligned} C_y(m', r') &= g^{m'} y^{r'} \bmod p \\ &= g^{m + x(r - r')} g^{xr'} \bmod p \\ &= g^m g^{xr} \bmod p \\ &= C_y(m, r) \end{aligned} \quad (5)$$

트랩도어 충돌 해쉬 함수는 위와 같은 특성으로 송수신간에 충돌 메시지로 동일한 충돌 해쉬 값을 계산하지 못한다면 이를 위조하거나 속인 것으로 간주할 수 있다. 또한 기존의 RSA(Rivest-Shamir-Adleman)나 DSA(Digital Signature Algorithm) 기법과 같은 전통적 서명 기법을 사용하기 때문에 보안 프로토콜에 적용하기 쉬운 이점이 있으므로, 본 논문에서는 송수신자 간의 동일한 충돌 해쉬 값을 이용하여 그룹의 세션키를 설립한다.

Ⅲ. 제안 기법

본 논문에서 제안하는 그룹키 생성과 키교환 기법은 사전에 동의한 큰 소수 g, p를 이용하여 초기 임의의 랜덤 메시지 m과 보조 랜덤값 r을 모든 디바이스로 분배하여 충돌 메시지를 생성하고, 초기 디바이스를 인증한다. 이후 트랩도어 충돌 해쉬를 계산하여 정상적인 디바이스임을 인증한다. 또한 앞 단계에서 인증을 완료한 디바이스들은 이를 그룹키로 정의하고, 보안 세션을 설립하는 기법을 제안한다. 본 장에서 제안하는 최종 목표는 그림 2와 같다.

1. 가정 사항

서비스제공자(IN)를 포함한 모든 게이트웨이(MN)와 디바이스(ASN 및 ADN)는 서로 통신이 가능한 위치에 있고, 신뢰성의 컴퓨팅 환경으로 물리적인 공격에 안전하며, IN과 MN은 안전한 통신 회선을 보장한다.

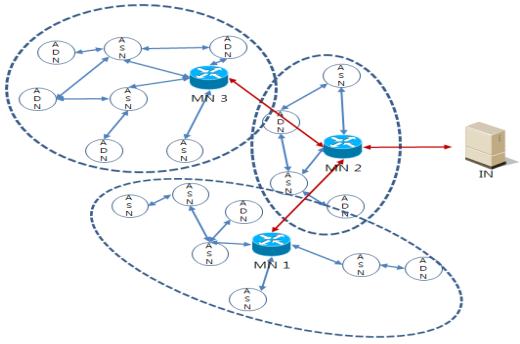

그림 2. 제안하는 기법의 최종 목표
Fig. 2. The Final Goal of The Proposed Method

2. 제안 기법의 파라미터

제안 기법에서 표기하는 디바이스의 요약어는 표 4와 같고, 안전한 그룹키 생성 및 키교환에 사용하는 파라미터 설명은 표 5와 같다.

표 4. 디바이스 요약어
Table 4. Device Abbreviation

Name	Description
IN	서비스 제공자
MN	게이트웨이(미들 노드)
ASN, ADN	M2M 디바이스

표 5. 트랩도어 충돌 해쉬 파라미터
Table 5. Trapdoor Collision Hash Parameter

Parameter	Description
g, p	큰 소수
x	비밀키
r_i	보조 랜덤 값
m_i	충돌 메시지
y_i	충돌 해쉬 공개키
S	전자 서명
ID_i	식별 정보
C_{y_i}	트랩도어 충돌 해쉬
SK_i	그룹 세션키

3. 사전 단계

그림 3과 같이 M2M 환경에서 정상적인 장치인 IN, MN, ASN, ADN은 기존 무선망의 EAP-TLS 등과 같은 인증방식으로 식별 정보와 인증, 그룹키에 사용하는 값을 사전에 동의하여 IN으로부터 MN과 ASN 및 ADN으로 공유한다.

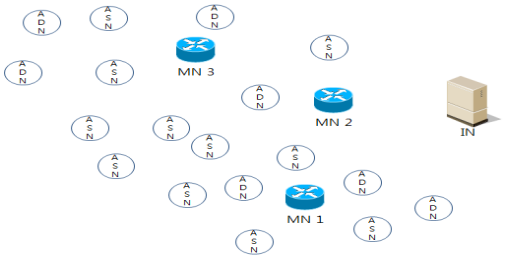

그림 3. 디바이스 인증 및 그룹키 교환대기 상태
Fig. 3. Device Authentication and Group Key Exchange Standby Status

4. 그룹키 생성 및 키 교환

IN으로부터 다수의 게이트웨이 MN$_i$은 그룹간의 키로 사용할 최초 랜덤 메시지 m_i와 보조 랜덤값 r_i를 수신한다. m_i와 r_i을 수신한 MN$_i$은 자신의 비밀키 $x_{i_{mn_i}}$를 선택하고, 보조 랜덤 메시지 $r_{mn_{i+1}}$을 생성하여 충돌 메시지 $m_{mn_{i+1}}$를 식 (6)과 같이 계산한다.

$$m_{mn_{i+1}} = m_i + x_{i_{mn_i}}(r_i - r_{mn_{i+1}}) \quad (6)$$

생성한 보조 랜덤 메시지 $r_{mn_{i+1}}$과 충돌 메시지 $m_{mn_{i+1}}$, 공개키 y_{mn_i}을 이용하여, 트랩도어 충돌 해쉬값을 식 (7)과 같이 미리 계산한다. 이는 각 ASN, ADN 노드들이 생성한 충돌 해쉬 값과 동일한 값인지 비교하기 위해 계산한다.

$$C_{y_{mn_i}}(m_{mn_{i+1}}, r_{mn_{i+1}}) = g^{m_{mn_{i+1}}} y^{r_{mn_{i+1}}} \bmod p \quad (7)$$

MN$_i$의 수신 가능한 주변 노드들에게 $x_{i_{mn_i}} r_{mn_{i+1}}$ 값과 이 값으로 새롭게 계산하여 생성한 충돌 메시지 $m_{mn_{i+1}}$을 전송한다. $x_{i_{mn_i}} r_{mn_{i+1}}$과 $m_{mn_{i+1}}$ 값을 수신한 노드들은 자신의 비밀키 $x_{i_{node_i}}$를 선택하고, 자신이 충돌 메시지 값을 계산하기 위한 보조 랜덤 메시지 $r_{node_{i+1}}$ 값을 생성한다. 선택한 비밀키와 생성한 보조 랜덤 메시지로 노드에서 동일한 충돌 해쉬값을 계산할 수 있는 충돌 메시지 $m_{node_{i+1}}$은 식 (8)과 같다.

$$m_{node_{i+1}} = m_{mn_{i+1}} + (x_{i_{mn_i}} r_{mn_{i+1}} - x_{i_{node_i}} r_{node_{i+1}}) \quad (8)$$

노드들은 자신의 ID_{node_i}와 $m_{node_{i+1}}$, $r_{node_{i+1}}$ 값의 서명으로 S_{node_i}를 생성하고, 이를 MN$_i$으로 전송한다. 노드들로부터 전송받은 서명 S_{node_i}를 포함한 ID_{node_i}와 $m_{node_{i+1}}$, $r_{node_{i+1}}$를 MN$_i$는 저장한다.

노드들과 MN$_i$ 사이 서명에 대한 변조나 위조 확인을 위해 MN$_i$ 자신의 식별정보 ID_{mn_i} 와 서명S_{mn_i}, 공개키 y_{mn_i} 그리고 노드로부터 받은 서명 S_{node_i}값을 해당 노드들에게 전송한다.

MN$_i$로부터 전송받은 ID_{mn_i}, S_{mn_i}, y_{mn_i}를 저장하고, 노드 자신이 전송하고 재전송 받은 서명 S_{node_i}가 변조나 위조에 노출 확인을 위해 자신의 공개키 y_{node_i}로 확인한다. MN$_i$와 그룹키로 사용하기 위한 $C_{y_{node_i}}$를 자신의 $m_{node_{i+1}}$, $r_{node_{i+1}}$로 트랩도어 충돌 해쉬 값을 식 (9)와 같이 계산한다.

$$C_{y_{mn_i}}(m_{mn_{i+1}}, r_{mn_{i+1}}) = g^{m_{mn_{i+1}}} y^{r_{mn_{i+1}}} \bmod p \quad (9)$$

MN$_i$의 서명 또한 변조나 위조에 대한 확인을 하기 위해 노드는 MN$_i$의 서명 S_{mn_i}와 노드들 자신의 공개키 y_{node_i}를 MN$_i$에게 전송한다. 노드로부터 수신한 노드의 각 공개키 y_{node_i}로 저장한 노드의 서명 S_{node_i}를 확인한다.

정상적인 노드로부터 수신된 값을 확인하고, MN$_i$는 자신이 계산한 트랩도어 충돌 해쉬값 $C_{y_{mn_i}}$와 노드로부터 수신한 $m_{node_{i+1}}$, $r_{node_{i+1}}$, y_{node_i} 값으로 $C_{y_{node_i}}$를 계산하고 이를 동일한 값인지 식 (10)과 같이 비교한다.

$$C_{y_{mn_i}}(m_{mn_{i+1}}, r_{mn_{i+1}}) = C_{y_{node_i}}(m_{node_{i+1}}, r_{node_{i+1}}) \quad (10)$$

정상적인 노드에서 생성하고 계산한 값들은 MN$_i$과 동일한 트랩도어 충돌 해쉬 값을 계산할 수 있다. 또한 동일한 트랩도어 충돌 해쉬 값을 확인하고, 식 (11)과 같이 각 노드들과 MN$_i$과의 그룹키로 설립하고, 이를 그룹 세션키로 설정한다.

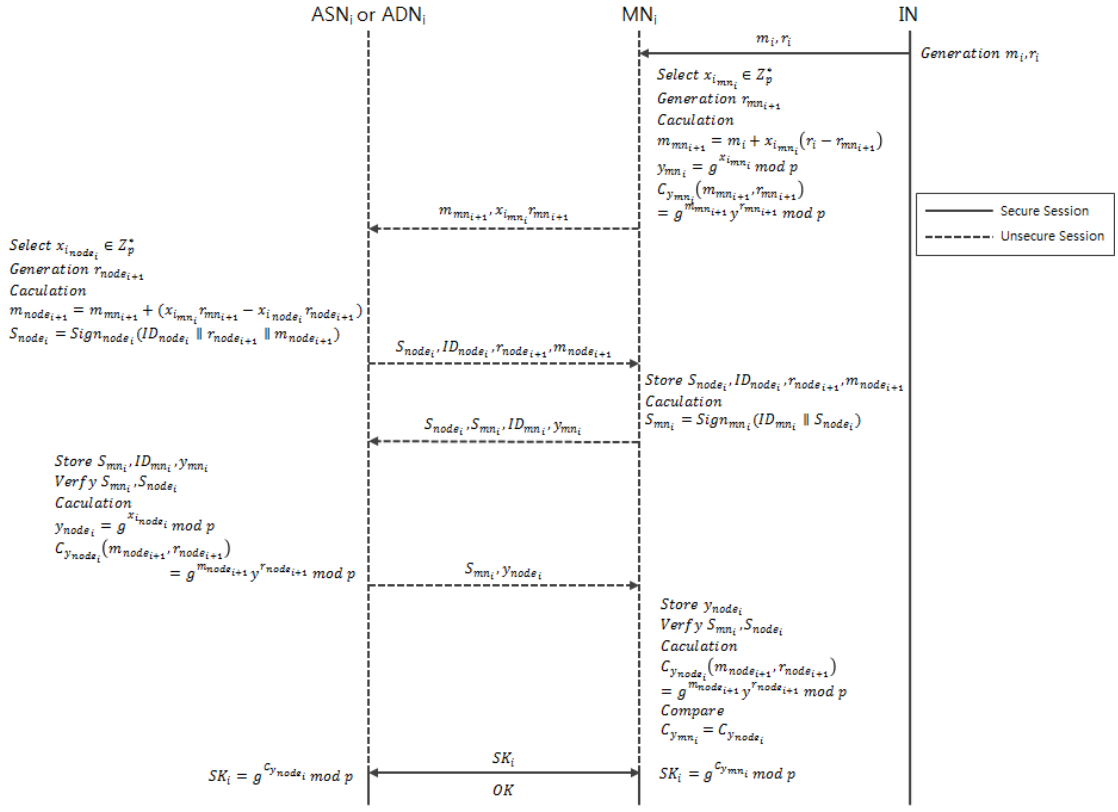

그림 4. 제안하는 그룹키 생성 및 키 교환 기법
Fig. 4. The Proposed of Group Key Generation and Exchange Method

$$SK_i = g^{C_{y_{mn_i}}} \bmod p = g^{C_{y_{node_i}}} \bmod p \quad (11)$$

세션키 SK_i는 다수의 노드들과 하나 또는 다수의 MNi과 그룹내의 정보를 주고받는 중요한 보안 세션키로 설정한다.

Ⅳ. 성능 분석

본 논문에서 제안한 그룹키 생성 및 키교환 기법은 각 그룹의 게이트웨이인 MNi로부터 수신한 $x_{i_{mn_i}} r_{mn_{i+1}}$과 $m_{mn_{i+1}}$ 값을 이용하여 충돌 메시지를 생성하고 동일한 트랩도어 충돌 해쉬 값을 계산하였다. 또한 이를 서명함으로써 위조나 변조에 대한 공격을 확인하며 수행하였다.

정상적인 노드임을 확인한 후 동일한 트랩도어 충돌 해쉬 값이 한 그룹의 그룹키로 설정되고, 정상적인 노드들과 사전에 공유한 큰 소수 g, p를 이용하여 그룹 세션키를 설정하였다.

1. 제안 기법 검증

본 논문에 제안한 기법은 표 6과 같이 파라미터에 대한 값으로 1개의 게이트웨이와 5개의 노드들로 검증하였다.

노드들은 MNi의 비밀키 x_{mn}와 연산한 $x_{mn}r$, $m_{mn_{i+1}}$ 값의 수신을 시작으로 노드들은 자신의 x_{nodei}를 사용하여 새로운 충돌 메시지m'_{nodei}를 계산하여 생성하고, 노드들이 생성한 r'_{nodei}과 자신의 공개키 y_{nodei}로 트랩도어 충돌 해쉬 값 $C_{y_{nodei}}$ 을 계산하였다. 표 6의 5개 노드들과 MNi는 모두 동일한 트랩도어 충돌 해쉬값을 계산하여 그룹키로 확정지으며, 그룹 세션키 또한 동일한 값을 나타낸다.

2. 안전성 분석

본 절에서는 제안하는 기법의 안정성에 대해 분석하고, MNi와 노드들 사이의 가능한 공격들에 대해 분석하였다.

표 6. 트랩도어 충돌 해쉬 파라미터 검증
Table 6. Trapdoor Collision Hash Parameter Verification

g	227		p	487	
m	70		r	150	
MN1					
x_{mn}	89	$x_{mn}r$	12549	m'_{mn}	871
y_{mn}	246	$C_{y_{mn}}$	137	SK	6
Node1					
x_{node1}	71	r'_{node1}	179	m'_{node1}	711
y_{node1}	384	$C_{y_{node1}}$	137	SK	6
Node2					
x_{node2}	83	r'_{node2}	158	m'_{node2}	306
y_{node2}	246	$C_{y_{node2}}$	137	SK	6
Node3					
x_{node3}	43	r'_{node3}	298	m'_{node3}	606
y_{node3}	190	$C_{y_{node3}}$	137	SK	6
Node4					
x_{node4}	61	r'_{node4}	217	m'_{node4}	183
y_{node4}	10	$C_{y_{node4}}$	137	SK	6
Node5					
x_{node5}	19	r'_{node5}	691	m'_{node5}	291
y_{node5}	217	$C_{y_{node5}}$	137	SK	6

• 트랩도어 충돌 해쉬 비밀키 공격

기존의 트랩도어 충돌 해쉬 기법은 동일한 비밀키를 사용하여 충돌 메시지 생성하여 계산하기 때문에 노드간의 전송하는 키쌍 (m', r')과 (m'', r'')을 수집한다면, 다음 식 (12)와 같이 비밀키 x에 대한 유추가 가능하다. 제안하는 기법은 초기 $x_{i_{mn_i}} r_{mn_{i+1}}$과 $m_{mn_{i+1}}$을 MNi에서 전송하고, 노드들에서 MNi으로 전송하는 $m_{node_{i+1}}$와 $r_{node_{i+1}}$은 식 (12)와 같은 문제를 갖지 않는다.

$$\begin{aligned} C_{y''}(m'', r'') &= C_{y'}(m', r') \\ g^{m''} g^{xr''} \bmod p &= g^{m'} g^{xr'} \bmod p \\ m'' + xr'' &= m' + xr' \\ x &= \frac{m' - m''}{r' - r''} \end{aligned} \tag{12}$$

트랩도어 충돌 해쉬 값을 노출하지 않고, 노드들에서 새로운 $x_{i_{node_i}} \in Z_p^*$를 선택하여 새로운 충돌 메시지를 계산하고 생성하기 때문에 충돌 메시지에 사용한 비밀키 $x_{i_{node_i}}$에 대한 공격은 식 (13)과 같이 어렵다.

$$x_{i_{node_i}} \neq \frac{m_{mn_{i+1}} - m_{node_{i+1}}}{x_{i_{mn_i}} r_{mn_{i+1}} - r_{node_{i+1}}} \tag{13}$$

또한 $x_{i_{mn_i}}$와 $x_{i_{node_i}}$에 대한 공격은 이원일차연립방정식의 문제가 되고, 서로가 전송하는 공개키 y_{mn_i}와 y_{node_i}는 이산 멱승 연산으로 생성하기 때문에 비밀키 공격은 불가능하다.

• 트랩도어 충돌 메시지 공격

초기 MNi에서 전송한 $x_{i_{mn_i}} r_{mn_{i+1}}$과 $m_{mn_{i+1}}$으로 공격자는 임의의 보조 랜덤값을 생성하여 충돌 메시지를 생성하지만, 이는 정상적인 노드와 MNi에서 계산한 올바른 충돌 메시지를 생성하기 어렵다. 공격자는 자신이 임의로 선택한 비밀키 x에 대해서 충돌 메시지를 생성하고 이를 식 (12)와 같이 대입하여 MNi의 비밀키 $x_{i_{mn_i}}$ 대한 공격을 시도할 수 있지만, 트랩도어 충돌 해쉬 비밀키 공격 저항성에 의해 불가능하다.

• 위장 공격

공격자는 그룹 세션키 SK_i에 대해 공격을 시도할 수 있지만, MNi와 노드들 사이에서 전송한 $x_{i_{mn_i}} r_{mn_{i+1}}$, $m_{mn_{i+1}}$과 $m_{node_{i+1}}$, $r_{node_{i+1}}$를 이용하여 그룹키인 트랩도어 충돌 해쉬값 C_y을 생성할 수 없다.

이는 식 (7)과 식 (9)와 같이 사전에 공유한 큰 소수 g, p를 알아내는 공격이 어렵고, 충돌 메시지와 랜덤값으로 트랩도어 충돌 해쉬 값을 계산하기 어려우며, 초기에 전송한 $x_{i_{mn_i}} r_{mn_{i+1}}$, $m_{mn_{i+1}}$을 시작으로 정상적인 노드들에 대해서만 동일한 C_y를 생성할 수 있으므로 정상적인 노드임을 가정한 위장공격은 불가능하다.

• 중간자 공격

노드들과 MNi 사이에서 공개키 y_{mn_i}와 y_{node_i}를 저장

하고, 수집하여 정상적인 노드에 대한 인증을 시도할 수 있으나, 그룹 세션키 SK_i에 대한 생성이 불가능하고, 이산 멱승 연산의 문제를 가지므로 불가능 하다.

- **재전송 공격**

공격자는 노드들과 MN_i 사이에서 공개키 y_{mn_i}와 y_{node_i}를 포함한 $x_{i_{mn_i}} r_{mn_{i+1}}$, $m_{mn_{i+1}}$, $m_{node_{i+1}}$, $r_{node_{i+1}}$를 수집하여 이를 재사용하여 MN_i으로부터 그룹 세션키의 사용을 인증할 수 있으나, 그룹 세션키 SK_i의 동일한 트랩도어 충돌 해쉬 C_y를 생성할 수 없을 뿐만 아니라 SK_i 또한 생성 할 수 없으므로 공격은 어렵다.

V. 결 론

본 논문에서는 트랩도어 충돌 해쉬 기법을 이용하여 M2M 환경에 정상적인 노드끼리 사용할 수 있는 그룹키를 생성하고, 세션 키를 설정함으로써 보안성을 제공하는 그룹 통신 기법을 제안하였다. 초기 랜덤 메시지와 보조 랜덤값을 서비스 제공자로부터 수신하고, 동일한 트랩도어 충돌 해쉬 값을 생성하여 정상적인 노드와 게이트웨이 임을 확인하였다. 또한 노드들과 게이트웨이 간의 전송하는 정보들은 악의적인 공격에도 비밀키 공격, 충돌 메시지 공격, 위장 공격, 중간자 공격, 재전송 공격에 저항성을 가지는 안전한 기법임을 확인하였다. 또한 충돌 메시지를 공격자가 생성하여도 동일한 트랩도어 충돌 해쉬 값은 생성하지 못하는 트랩도어의 특수성을 가지고 있다.

본 논문에서 제안한 기법은 앞으로 M2M 환경에서 다수 무선 통신구간의 키 관리를 위한 필수 보안 요구사항으로 보안성과 안전성을 제공할 수 있을 것으로 기대한다.

References

[1] Wen Quan JIN, Do Hyeun Kim, “Implemen tation and Experiment of CoAP Protocol Based on IoT for Verification of Interoperability,” The Journal of The Institute of Internet Broadcasting and Communication(JIIBC), Vol 14, No 4, pp 7-12, Aug 2014.

[2] JungOh Park, Sangkun Kim, “Mutual Authentica tion and Key Establishment Mechanism for Secure Data Sharing in M2M Environment,” The Journal of The Institute of Internet Broadcasting and Communication(JIIBC), Vol 15, No 4, pp 33-41, Aug 2015.

[3] G. Lawton, “Machine-to-Machine technology ge ars up for growth,” IEEE Computer Society, Vol 37, No 9, pp 12-15, Sep 2004.

[4] KISA, “Internet Threat Trend things”, Korea Internet & Security Agency, 2014.

[5] Jeongin Kim, Namhi Kan, “Secure Configuration Scheme of Pre-shared Key for Lightweight Devices in Internet of Things,” The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol 15, No 3, pp 1-6, Jun 2015.

[6] oneM2M-TS-0001, “oneM2M Functional Archi tecture Technical Specification” v0.2.1, 2013.

[7] Kim Zongheon, Kim Jaeu, Yoo Seok, Lee Jaeyong, “Wireless technology for M2M/IoT services,” Korea Institute of Communications and Information Sciences(KICS), Vol 30, No 8, pp 11-19, 2013.

[8] Lee Junseop, “A Study on the M2M service layer in ITU-T and oneM2M,” Conf. of Korea Information and Communications Society, pp 1381-1382, Jan 2015.

[9] oneM2M-TR-0008, “Analysis of security solution s for oneM2M system”, v0.2.1, 2013.

[10] Yoo Heekyung, Sung Kyung, “Analysis and implementation of Digital Signature Algorithm using Hash function,” Journal of The Korea Knowledge Information Technology Society (KKITS), Vol 6, No 3, pp 129-142, Jun 2011.

[11] KISA, “The Trend of Project related to Tech nology for Personal Information protection”, Korea Internet and Security Agency, 2006.

[12] Ateniese Giuseppe, De Medeiros Breno, “Iden tity-based chameleon hash and applications,”

Financial Cryptography. Springer Berlin Heidelberg, pp 164-180, 2004.

[13] Krawczyk H., Rabin T., "Chameleon signatures," Proceedings of NDSS, pp 143-154, 2000.

저자 소개

김 성 수(정회원)

- 2008년 2월 : 영동대학교 컴퓨터공학과 공학사
- 2010년 2월 : 숭실대학교 컴퓨터학과 석사
- 2010년 3월 ~ 현재 : 숭실대학교 컴퓨터학과 박사과정
- 2014년 4월 ~ 현재 : 안양대학교 교양대학 겸임교수

<주관심분야 : 네트워크 보안, 인증 이론, 암호학>

전 문 석(정회원)

- 1981년 2월 : 숭실대학교 전자계산학과 졸업
- 1986년 2월 : University of Maryland Computer Science 석사
- 1989년 2월 : University of Maryland Computer Science 박사
- 1991년 3월 ~ 현재 : 숭실대학교 컴퓨터학과 정교수

<주관심분야 : 정보보호, 네트워크 보안, 암호학>

최 도 현(정회원)

- 2008년 2월 : 동서울대학 컴퓨터소프트웨어 공학사
- 2010년 8월 : 숭실대학교 컴퓨터학과 석사
- 2010년 9월 ~ 현재 : 숭실대학교 컴퓨터학과 박사과정

<주관심분야 : 모바일 보안, PKI, Secure Coding>