JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 19, No. 11, pp. 2173-2182, Nov. 2018

그룹내 메시지 교환에 대한 비밀분산을 이용한 익명 서비스 제공 프로토콜

권 유 진[1] · 김 정 운[1] · 남 기 원[2] · 한 승 철[1]
[1]명지대학교 일반대학원 보안경영공학과
[2]중앙대학교 사범대학

A New Anonymity Service Providing Protocol Using Secret Sharing Scheme for Group Communication

Yu Jin Kwon[1] · Jung Woon Kim[1] · Ki Won Nam[2] · Seung Chul Han[1]
[1]Interdisciplinary Program of Security and Management Engineering, Myongji University, Korea
[2]College of Education, Chung-ang University, Korea

[요 약]

인터넷의 발전으로 공통 관심사를 가지는 다수의 사용자들이 SNS(Social Network Service)를 이용하여 가상의 대화 공간을 형성하고, 특정 주제에 대해 의견을 교환하는데 있어서 익명 서비스를 요구하는 상황이 증가하고 있다. 그룹내 메시지 교환에서 익명 서비스란, 다른 사용자들에게 작성자의 식별 정보를 노출하지 않고 자신의 메시지를 전달할 수 있도록 하는 기능을 의미하며 안전장치로서, 추후 익명 메시지의 실명 공개가 필요할 경우, 특정 조건이 만족되면 식별 정보를 밝힐 수 있도록 하는 것을 의미한다. 본 논문에서는 비밀 분산기법을 이용하여 SNS(채팅앱, 메신저, 게시판 등) 그룹내 메시지 교환에 있어서 익명 서비스를 제공하는 효율적인 프로토콜을 제안한다.

[Abstract]

As the use of various kind of social network services is rapidly increasing, the situation of exchanging messages in virtual space is also growing. The anonymous service means a function that allows users to transmit their own message without revealing their identity of the author. If disclosure of the anonymous message is required, the identification information should be disclosed if certain conditions are met. In this paper, we present a new anonymity and security service providing protocol based on secret sharing method for group communication.

색인어 : 그룹 커뮤니케이션, 익명성, 비밀분산, 보안, 프로토콜

Key word : Group communication, Anonymity, Secret Sharing, Security, Protocol

http://dx.doi.org/10.9728/dcs.2018.19.11.2173

Received 10 October 2018; **Revised** 07 November 2018
Accepted 20 November 2018

***Corresponding Author; Seung Chul Han**

Tel: +82-31-330-6434
E-mail: bongbong@mju.ac.kr

Ⅰ. 서 론

인터넷의 발전으로 다수의 사용자들은 언제 어디서나 다양한 단말기를 통한 의사소통이 가능해 졌다. 이에 따라 소셜 네트워크 서비스(SNS), 메신저, 인터넷 게시판 등을 이용하여 공통 관심사를 가지는 이용자들이 모여 가상의 대화 공간을 형성하고, 특정 주제에 대해 메시지를 교환하는 상황이 증가하고 있다[1]. 또한 의견교환의 과정에 있어서 자신의 신분 또는 사회적 지위에 관계없이 자유로운 의사소통을 위한 익명성의 필요성이 대두되고 있다. 대표적인 어플리케이션으로는 전자화폐, 전자투표, 오픈 채팅, 익명 게시판 등을 말할 수 있다. 예를 들어, '국민신문고'를 통해 국민이라면 누구든지 익명게시판을 이용하여 각종 민원이나 사건에 대해 제보할 수 있다[2]-[4]. 이러한 익명 게시판은 언론이 다양한 정보를 수집하는데 긍정적 영향을 주며, 제도적 문제를 개선하는데 도움을 줄 수 있다. 반면, 이러한 익명 게시판은 타인에 대한 비방 또는 사칭으로 피해자가 발생하는 악용 사례도 발생할 수 있다. 이와 같이 여러 분야에서 익명성을 필요로 하는 사례가 증가하고 있음에도 불구하고 대부분 인터넷 공간에서는 익명성을 제공함으로써 발생하는 문제에 대해서는 깊게 고려하지 않고 있다.

그룹내 메시지 교환에서 익명 서비스란, 그룹내 다른 사용자들에게 작성자의 식별 정보를 노출하지 않고 자신의 메시지를 전달할 수 있도록 하는 기능을 의미한다. 이러한 익명 서비스는 자유롭게 의견을 제시할 수 있는 환경을 제공하므로 조직 또는 그룹 내부의 의사소통 증가로 이어 질 수 있으며, 갈등을 줄이고 문제를 객관적으로 해결할 수 있도록 하는 장점이 있다. 하지만, 익명 서비스를 제공함으로써 발생하는 부정적인 현상도 존재한다. 대표적으로 익명성을 이용해 다른 사람을 사칭한다거나 제 3자를 비방하여도 누구인지 밝혀낼 수 없다는 문제가 존재한다. 이로 인해 개인의 이익 추구를 위한 사회적 혼란을 야기할 수도 있다. 이러한 문제를 해결하기 위해 익명성을 보장하면서도 필요시 작성자의 신원을 밝히기 위한 안전장치가 필요하다.

본 논문에서는 SNS(채팅앱, 메신저, 게시판 등) 그룹내 메시지 교환에 있어서 익명 서비스를 제공하는 효율적인 프로토콜을 제안하고, 이에 대한 보안성 논의를 통해 안전성을 분석한다. 또한 익명 서비스 제공과 더불어 안전장치로서, 추후 익명 메시지의 실명 공개가 필요할 경우, 특정 조건이 만족되면 비밀정보를 밝힐 수 있도록 비밀분산 기법을 이용하여 메시지 작성자를 공개하고 검증할 수 있는 기능을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 소개한다. 3장에서는 그룹 통신에서 보안과 익명 서비스 제공, 비밀분산을 이용하여 그룹원 요청에 의해 실명 공개가 가능한 프로토콜과 시스템을 제안한다. 4장에서는 다양한 보안 위협으로 부터의 안전성을 분석하고 , 5장에서 결론을 맺는다.

Ⅱ. 관련 연구

2-1 익명성에 관한 연구

익명성이란 어떤 행위를 한 사람이 누구인지 드러나지 않게 하는 것으로 자신을 숨기는 것을 의미한다. 2000년대 초에 인터넷의 발달과 함께 온라인의 대표적인 특성인 익명성에 대한 많은 연구가 진행되었다[5]-[10]. 온라인상의 익명성은 어떠한 제약 없이 자유롭게 의견을 제시할 수 있는 환경을 제공 해준다[11]. 이러한 환경은 조직 또는 그룹 내부의 의사소통 증가로 이어 질 수 있으며, 갈등을 줄이고 문제를 객관적으로 해결할 수 있도록 하는 장점이 존재한다[12].

하지만 익명성으로 인해 사람들의 감정을 자극하는 거짓된 내용, 특정한 개인이나 집단에 위협을 가하는 행동 등의 다양한 역기능이 존재한다[13]. 그러므로 본 논문에서는 익명성으로 인해 발생하는 문제점은 비밀분산을 통해 최소화 시키고 익명성의 장점을 극대화 시킬 수 있는 방법에 대해 연구하였다.

2-2 비밀분산에 관한 연구

비밀분산은 하나의 비밀정보를 여러 개의 비밀조각으로 분할하여 다수에게 공유시킴으로써 비밀정보에 대한 관리와 복구가 용이하다. 이와 관련하여 다양한 연구가 존재하며, 대표적인 연구로는 다항 보관법을 이용한 Shamir의 (t, n)-임계치 비밀분산법과 G.J.Simmons의 다중레벨 비밀분산법이 있다[14]-[17].

Shamir의 (t, n)-임계치 비밀분산법은 다항 보관법을 이용한 방식으로 비밀정보를 n개의 비밀조각으로 분할하고, 그 중 임의의 t개 이상의 비밀조각이 모이면 원래의 비밀정보를 복원할 수 있다. t-1개의 비밀조각이 모였을 때는 복원이 불가능하며, 일부 비밀정보가 분실 또는 파괴되어도 비밀정보에 대한 복원이 가능하다. 또한, 허가되지 않은 참가자는 비밀조각을 가지지 않기 때문에 비밀정보에 대한 아무런 정보도 얻을 수 없다.

다중레벨 비밀분산법은 계층구조를 반영한 것으로 참가자 집합을 계층에 따라 레벨을 나누고, 비밀정보에 대한 복원 권한을 다르게 부여하는 방식이다. 레벨 별로 그룹화 되어 있으며, 각 레벨별 그룹에 최소 모여야 되는 비밀조각의 수가 서로 다르게 적용된다. 낮은 레벨일수록 더 많은 비밀조각 수를 요구하며, 서로 다른 레벨의 비밀정보를 가진 참여자가 합의하는 경우 가장 하위 레벨의 조건에 맞춘다. 본 논문에서 제안하는 익명 메시지 작성 환경에는 모든 사용자가 수평적인 관계에 있으므로 권한을 위임하는 다중레벨 비밀분산법이 아닌 Shamir의 비밀분산법이 적합하다.

본 논문에서 제안하는 SNS 메시지 작성 환경에의 익명성에 대한 연구는 보완할 사항이 있으나, 현재 본 논문의 내용으로 특허출원 실사를 통과하였다[1].

1) 출원번호 10-2016-0161050, 10-2018-0025771,대한민국

Ⅲ. 보안과 익명 서비스 제공 시스템

3-1 시스템 컴포넌트

본 논문에서 제안하는 비밀분산을 이용한 익명 서비스 제공 프로토콜은 메시지 서버, 검증 서버, 익명 메시지를 작성하는 사용자(이하 '작성자'로 지칭함), 익명의 메시지에 대해 공개를 요청하는 동일 그룹에 속한 사용자(이하 '요청자'로 지칭함), 익명 메시지를 받는 동일 그룹에 속한 사용자(이하 '수신자'로 지칭함)로 구성된다.

○ 메시지 서버(Message Server, 'Server_M'로 지칭)
메시지 서버는 메시지 브로드캐스트, 전송되는 메시지에 대한 보관, 암/복호화를 위한 키 관리, 비밀분산의 KEY조각 재조립, 실명 검증 요청, 실명 공개 등의 역할을 수행한다.

○ 검증 서버(Verification Server, 'Server_V'로 지칭)
검증 서버는 메시지 아이디 및 메시지 아이디 발급 시간에 대한 생성 및 관리, 암/복호화를 위한 키 관리, 메시지 서버가 요청하는 실명 공개에 대한 검증, 비밀분산 KEY조각 생성 및 분배 등의 역할을 수행한다.

○ 작성자(Writer, 'Sender'로 지칭)
작성자는 일반메시지 혹은 익명 메시지를 작성한다.

○ 요청자(Requestor, 'Requestor'로 지칭)
요청자는 익명의 메시지에 대해 동일한 그룹에 속하는 모든 사용자들에게 공개를 요청하는 역할을 수행한다.

○ 수신자(All User, 'Receiver'로 지칭)
수신자는 작성자와 동일한 그룹에 속하는 모든 사용자들이며 작성자가 작성한 메시지를 수신하고 공개요청에 대한 의사결정을 통해 비밀분산 KEY조각의 생성 및 전달 여부를 결정한다. 또한, 언제든지 '요청자'처럼 익명 메시지에 대한 공개를 요청 할 수 있다.

시스템 컴포넌트들은 다음 전제를 가지고 있다.

- Server_M, Server_V, 작성자, 요청자, 수신자는 모두 각각의 공개키와 개인키를 가지고 있다.
- 같은 그룹내의 Server_M, Server_V, 작성자, 요청자, 수신자는 동일한 그룹키를 공유하며, 그룹내 전달되는 메시지의 암/복호화에 사용한다.
- 작성자, 요청자, 수신자는 Server_V와 상호간의 대칭키를 나누어 가지고 있다.
- 작성자, 요청자, 수신자는 Server_M와 상호간의 대칭키를 나누어 가지고 있다.
- Server_M와 Server_V는 상호간의 대칭키를 나누어 가지고 있다.

3-2 프로토콜

프로토콜은 익명 메시지에 대한 작성자의 익명성 보장과 추후 익명 메시지를 작성한 작성자에 대한 공개요청 발의 시 일정조건이 만족하면 익명 메시지 작성자를 공개한다. 또한 그룹내의 참여자만이 메시지 수신 및 작성이 가능하다.

본 기술에서 제안하는 프로토콜은 크게 7단계의 과정으로 구분된다. 프로토콜 기술에 사용되는 기호는 [표 1]과 같다.

표 1. 기호에 대한 설명
Table 1. Summary of Symbols

Symbol	Definition
Server_M	Message Server
Server_V	Verification Server
Sender	Writer
Receiver	All User
Requestor	Requestor
SKey[V-Sen]	Symmetric key between Verification Server and Writer
SKey[V-User]	Symmetric key between Verification Server and All User
SKey[M-V]	Symmetric key between Verification Server and Message Server
SKey[M-Sen]	Symmetric key between Message Server and Writer
SKey[M-Req]	Symmetric key between Verification Server and Requestor
AKey[M]+	Public key of Message Server
SKey[M-User]	Symmetric key between Message Server and All User
SID	Writer Identification
MID	Message Identification
TS[MID]	Timestamp of Generated MID
Msg	Text
Num	Message Number
TS[Msg]	Timestamp of Written Message
Sen_Table	DB Table of Sender
V_Table	DB Table of Verification Server
M_Table	DB Table of Message Server
GKey	Group Key
H[*]	Hash Function
KEY	Encryption/Decryption Key
KEY(SID)	Encrypted SID value using KEY
Seg[i](KEY)	An expression that generates fragments of a KEY through Secret Sharing
Segment[i]	Fragment value that can restore the KEY value generated through secret sharing

작성자, Server_V, Server_M는 메시지들에 대한 정보를 다음 [표 2]과 같은 DB 테이블에 저장된다.

표 2. Sender, Server_ V, Server_M의 DB 테이블
Table 2. DB Tables of Sender, Server_V, Server_M

Table Name	Fields
Sen_Table	SID
	MID
	$TS_{[MID]}$
	Msg
	$TS_{[Msg]}$
	Num
	$Seg_{[i]}(KEY)$
V_Table	MID
	SID
	$TS_{[MID]}$
	H[MID:Msg]
	$H[SKey_{[V\text{-}Sen]}, MID, TS_{[MID]}]$
	KEY
	KEY(SID)
	$Seg_{[1]}(KEY),\cdots,Seg_{[n]}(KEY)$
M_Table	Num
	Msg
	$TS_{[Msg]}$
	MID
	H[MID:Msg]
	$H[SKey_{[V\text{-}Sen]}, MID, TS_{[MID]}]$
	KEY(SID)
	$Segment_{[1\_n]}$
	KEY
	SID

본 프로토콜의 시나리오는 작성자, 요청자, 수신자로 이루어진 그룹내에서 작성자가 익명 메시지를 작성하고, 추후 요청자와 수신자가 익명 메시지에 대한 작성자에 대한 실명 공개를 요청하고, 공개조건은 과반수이상이라고 가정한다.

1) 익명 메시지 아이디 발급 과정

첫 번째 과정은 작성자가 익명 메시지 작성을 위해 검증 서버에 메시지 아이디 발급을 요청한다. 작성자는 메시지 내용을 작성하고 추후 실명 공개를 위해서 추가 정보를 생성, 저장한다. 모든 과정은 작성자와 검증 서버가 공유하는 대칭키($SKey_{[V\text{-}Sen]}$)로 암/복호화된다.

Step 1. 익명의 메시지를 작성하려는 작성자가 검증 서버에 익명 메시지 작성에 필요한 메시지 아이디 발급을 요청한다. 작성자는 자신의 SID와 Timestamp(TS)를 작성자와 검증 서버가 공유하는 대칭키($SKey_{[V\text{-}Sen]}$)로 암호화하여 검증 서버로 전송한다[18]-[20].

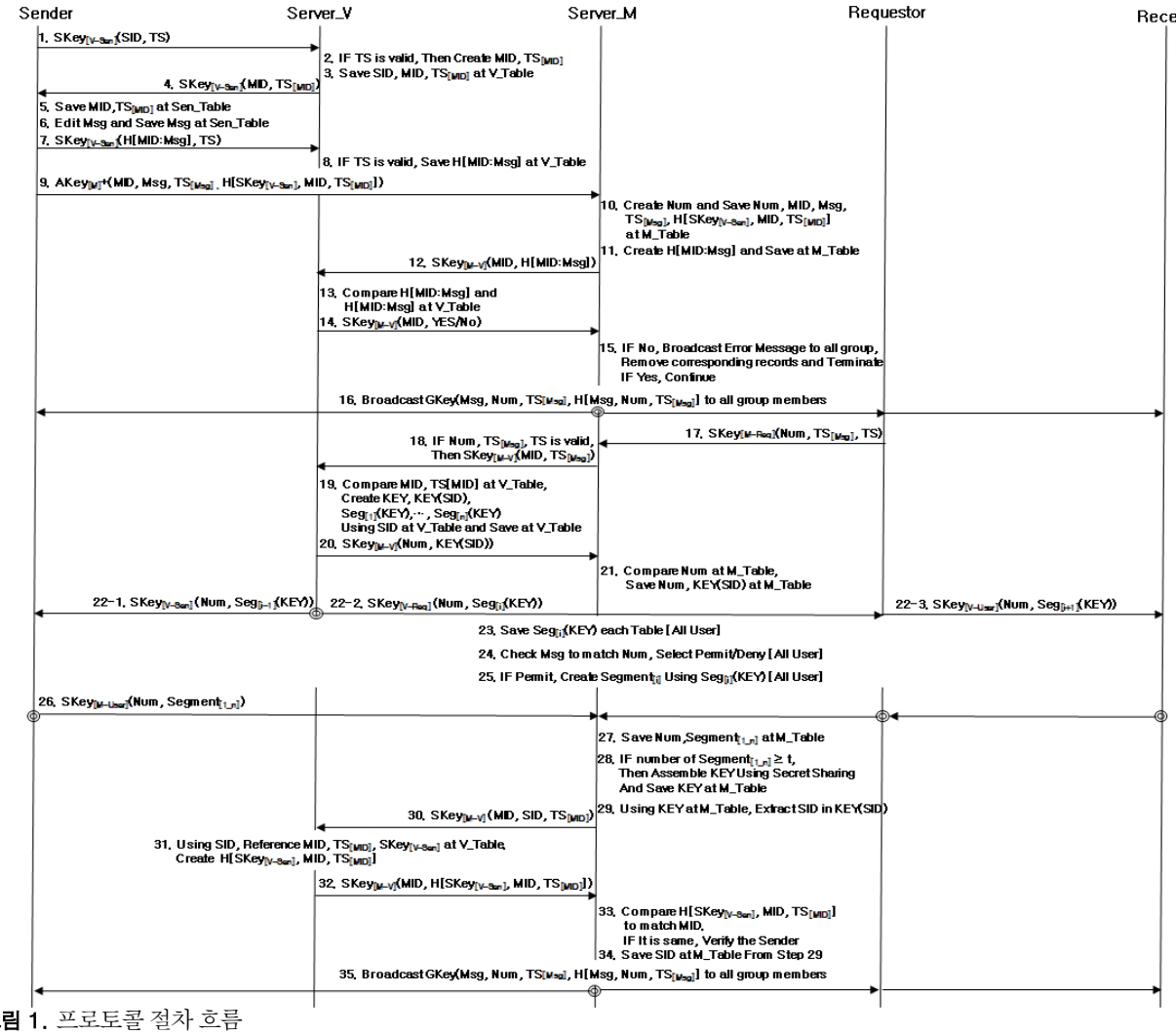

그림 1. 프로토콜 절차 흐름

Fig. 1. Protocol Process Flow

Step 2. 검증 서버는 TS로 유효성을 체크하고(재전송공격 여부 등), 유효한 경우, 고유의 메시지 ID(MID)와 MID 생성시간인 Timestamp($TS_{[MID]}$)를 생성한다.

Step 3. 검증 서버는 SID, MID, $TS_{[MID]}$를 자신의 DB 테이블(V_Table)에 저장한다.

Step 4. 검증 서버는 MID, $TS_{[MID]}$를 작성자와 공유하는 대칭키($SKey_{[V\text{-}Sen]}$)로 암호화하여 작성자에게 전송한다.

Step 5. 작성자는 수신한 MID, $TS_{[MID]}$를 자신의 DB 테이블(Sen_Table)에 저장한다.

Step 6. 작성자는 메시지 내용(Msg)을 작성하고 자신의 DB 테이블(Sen_Table)에 저장한다.

Step 7. 작성자는 MID와 Msg의 해시값(H[MID:Msg)]와 Timestamp(TS)를 대칭키($SKey_{[V\text{-}Sen]}$)로 암호화하여 검증 서버에 전송한다.

Step 8. 검증 서버는 TS로 유효성을 체크하고(재전송공격 여부 등), 유효한 경우, H[MID:Msg]를 자신의 DB 테이블(V_Table)에 저장한다.

2) 익명 메시지 전송 요청 과정

두 번째 과정은 작성자가 익명 메시지를 그룹내 배포를 위해, 메시지 서버에게 전달하는 과정이다.

Step 9. 작성자는 MID, Msg, 메시지 작성시간인 Timestamp($TS_{[Msg]}$), 그리고 작성자와 검증 서버간의 대칭키($SKey_{[V\text{-}Sen]}$), MID, $TS_{[MID]}$의 해시값(H[$SKey_{[V\text{-}Sen]}$, MID, $TS_{[MID]}$])을 메시지 서버의 공개키($AKey_{[M]+}$)로 암호화하여 메시지 서버에게 전송한다. 이 때, 메시지 서버는 누구로부터 메시지를 전달받았는지 파악할 수 없다.

Step 10. 메시지 서버는 고유의 메시지 번호(Num)를 생성하고, Num, MID, Msg, $TS_{[Msg]}$, H[$SKey_{[V\text{-}Sen]}$, MID, $TS_{[MID]}$]를 자신의 DB 테이블(M_Table)에 저장한다.

Step 11. 메시지 서버는 MID와 Msg의 해시값(H[MID:Msg)]을 생성하고 자신의 DB 테이블(M_Table)에 저장한다.

3) 익명 메시지 배포 과정

세 번째 과정은 메시지 서버가 전달받은 익명 메시지가 유효한 메시지인지 검증 서버를 통해 검증하고 그룹내 사용자들(작성자, 수신자, 요청자)에게 배포하는 과정이다.

Step 12. 메시지 서버는 MID, H[MID:Msg]를 메시지 서버와 검증 서버가 공유하는 대칭키($SKey_{[M\text{-}V]}$)로 암호화하여 검증 서버로 전송한다.

Step 13. 검증 서버는 수신한 H[MID:Msg]를 자신의 DB 테이블(V_Table)에 저장된 H[MID:Msg]값과 비교하여 동일한지 확인한다.

Step 14. 검증 서버는 비교 결과와 MID를 대칭키($SKey_{[M\text{-}V]}$)로 암호화하여 메시지 서버로 전송한다.

Step 15. 메시지 서버는 검증 서버로부터 전달받은 익명의 메시지 검증 결과를 확인하고 유효하지 않은 메시지 일 경우, 모든 그룹원에게 오류 결과를 전달하고 M_Table의 해당 메시지 ID의 레코드를 삭제 후 종료하며, 유효한 메시지 일 경우 다음 단계로 진행한다.

Step 16. 메시지 서버는 그룹내 사용자들이 공유하는 그룹키(GKey)로 메시지 내용(Msg), 메시지 번호(Num), 메시지 전송시간($TS_{[Msg]}$), 메시지 내용과 메시지 번호 그리고 메시지 전송시간을 이용해 생성한 해시값(H[Msg, Num, $TS_{[Msg]}$])을 암호화하여 그룹내 사용자들(작성자, 수신자, 요청자)에게 배포한다. 해시값은 무결성 검증을 위해 사용된다.

4) 익명 메시지 실명 공개 과정

네 번째 과정은 익명 메시지를 작성한 작성자가 아닌 그룹내 사용자들 중 요청자가 익명 메시지의 작성자를 공개하고자 할 때 메시지 서버와 검증 서버를 통해 수행되는 실명 공개 요청 과정이다.

Step 17. 요청자는 익명의 메시지 작성자에 대한 실명 공개 요청을 위해 Num, $TS_{[Msg]}$, TS를 요청자와 메시지 서버가 공유하는 대칭키($SKey_{[M\text{-}Req]}$)로 암호화하여 메시지 서버로 전송한다.

Step 18. 메시지 서버는 Num, $TS_{[Msg]}$, TS로 유효성을 체크하고(재전송공격 여부 등), 유효한 경우, 자신의 DB 테이블(M_Table)을 참조하여 해당 MID, $TS_{[MID]}$와 함께 자신과 검증 서버가 공유하는 대칭키($SKey_{[M\text{-}V]}$)로 암호화하여 검증 서버로 전송한다.

5) 익명 메시지 실명 공개를 위한 비밀정보 조각화 과정

다섯 번째 과정은 익명 메시지 실명 공개 요청 처리를 위해 검증 서버가 메시지 서버와 그룹내 모든 사용자들(작성자, 수신자, 요청자)에게 비밀분산(t, n) 처리를 위해 조각화하고 배포하는 과정이다.

Step 19. 검증 서버는 암/복호화 키 값(KEY)을 임의로 생성하고 메시지 서버로부터 전달받은 MID와 $TS_{[MID]}$에 해당하는 자신의 DB 테이블(V_Table)의 레코드에 KEY값을 저장한다. 해당 레코드의 SID를 암/복호화 키로 암호화한 KEY(SID)를 저장한다. 비밀분산을 통해 KEY값의 조각을 생성하는 연산식들($Seg_{[1]}$(KEY),…,$Seg_{[n]}$(KEY))을 생성하고 저장한다.

Step 20. 검증 서버는 Num, KEY(SID)를 메시지 서버가 공유하는 대칭키($SKey_{[M\text{-}V]}$)로 암호화하여 메시지 서버로 전송한다.

Step 21. 메시지 서버는 전달받은 Num, KEY(SID)를 자신의 DB 테이블(M_Table)의 해당 레코드에 저장한다.

Step 22. 검증 서버는 그룹내 모든 사용자$_{[1,\cdots,n]}$에게 Num와 해당하는 서로 다른 $Seg_{[1\_n]}$(KEY)를 자신과 각각 사용자간의 대칭키($SKey_{[V\text{-}User]}$)로 암호화하여 그룹내 모든 사용자에게 전송한다.

Step 23. 그룹내 모든 사용자(작성자, 수신자, 요청자)는 전달 받은 서로 다른 $Seg_{[i]}$(KEY)를 자신의 DB 테이블에 저장한다.

6) 익명 메시지 실명 공개 요청에 대한 비밀정보 재조립 과정

여섯 번째 과정은 익명 메시지 실명 공개 요청 처리를 위해 그룹내 모든 사용자(작성자, 수신자, 요청자)가 공개 여부에 대해 선택하고 메시지 서버가 그룹내 모든 사용자로부터 조각을 전달받고 재조립하는 과정이다.

Step 24. 그룹내 모든 사용자(작성자, 수신자, 요청자)는 Num에 해당하는 Msg를 확인 후 공개 여부를 선택한다.

Step 25. 그룹내 모든 사용자(작성자, 수신자, 요청자)중 공개를 원하는 사용자는 자신의 $Seg_{[i]}$(KEY)를 이용하여 KEY값을 복원할 수 있는 조각 값($Segment_{[i]}$)을 생성한다.

Step 26. $Segment_{[i]}$를 생성한 사용자는 Num와 $Segment_{[i]}$를 자신과 메시지 서버간의 대칭키($SKey_{[V\text{-}User]}$)로 암호화하여 메시지 서버에게 전송한다.

Step 27. 메시지 서버는 자신의 DB 테이블(M_Table)에 해당 Num의 레코드에 $Segment_{[1_n]}$들을 저장한다.

Step 28. 메시지 서버는 전달받은 $Segment_{[1_n]}$가 t개 이상일 경우 비밀분산을 이용하여 KEY를 계산하고 저장한다.

Step 29. 메시지 서버는 자신의 DB 테이블(M_Table)에 KEY와 KEY(SID)를 이용하여 SID를 추출한다.

7) 익명 메시지 실명 공개 과정

일곱 번째 과정은 그룹내 t명 이상의 사용자가 익명의 메시지 작성자에 대해 공개하고자 할 때 수행되는 과정이다.

Step 30. 메시지 서버는 추출된 SID를 이용하여, 자신의 DB 테이블을 참조하여 해당 MID, SID, $TS_{[MID]}$와 함께 자신과 검증 서버가 공유하는 대칭키($SKey_{[M\text{-}V]}$)로 암호화하여 검증 서버로 전송한다.

Step 31. 검증 서버는 자신의 DB 테이블에서 $SKey_{[V\text{-}Sen]}$, MID, $TS_{[MID]}$를 참고하여 해시값($H[SKey_{[V\text{-}Sen]}, MID, TS_{[MID]}]$)을 생성한다.

Step 32. 검증 서버는 MID, $H[SKey_{[V\text{-}Sen]}, MID, TS_{[MID]}]$를 자신과 메시지 서버가 공유하는 대칭키($SKey_{[M\text{-}V]}$)로 암호화하여 메시지 서버로 전송한다.

Step 33. 메시지 서버는 전달받은 MID에 해당하는 레코드의 $H[SKey_{[V\text{-}Sen]}, MID, TS_{[MID]}]$과 비교하여 일치하면 해당 SID를 사용하는 사용자가 해당 메시지를 작성했음을 확인한다. 불일치하면 종료한다.

Step 34. 메시지 서버는 전달받은 MID에 해당하는 자신의 DB 테이블(M_Table) 레코드에 SID를 저장한다.

Step 35. 메시지 서버는 메시지 내용(Msg), 메시지 번호(Num), 작성시간($TS_{[Msg]}$), SID, $H[Msg, Num, TS_{[Msg]}]$를 그룹내 사용자들이 공유하는 그룹키(GKey)로 암호화하여 전체 발송한다. 익명의 메시지와 사용자의 아이디를 그룹내 모든 사용자(작성자, 수신자, 요청자)에게 공개함으로써 해당 익명 메시지에 대한 작성자를 공개한다.

Ⅳ. 안전성 분석

4-1 보안성 요소

1) 기밀성(Confidentiality)

기밀성은 허가된 사용자만이 데이터에 접근하고 확인할 수 있는 것을 의미한다.

표 3. 암/복호화에 사용되는 KEY에 대한 정보
Table 3. Encryption/Decryption KEY

Key	Description
$SKey_{[V\text{-}Sen]}$	Symmetric key between Verification Server and Writer
$SKey_{[V\text{-}User]}$	Symmetric key between Verification Server and All User
$SKey_{[M\text{-}V]}$	Symmetric key between Verification Server and Message Server
$SKey_{[M\text{-}Sen]}$	Symmetric key between Message Server and Writer
$SKey_{[M\text{-}Req]}$	Symmetric key between Verification Server and Requestor
$AKey_{[M]+}$	Public key of Message Server
$SKey_{[M\text{-}User]}$	Symmetric key between Message Server and All User
GKey	Group Key

본 논문에서 제안하는 프로토콜은 모든 과정에서 발생하는 컴포넌트 간에 데이터 교환 시 모든 데이터에 대하여 위 [표 3]와 같이 상황에 적합한 대칭키(SKey), 공개키($AKey_+$), 그룹키(GKey)로 암/복호화 하는 과정을 거치게 된다. 결과적으로 본 프로토콜에서 발생하는 모든 트래픽에 대하여 암호화하여 전송하기 때문에 공격자가 중간에 암호화된 트래픽을 가로채더라도 원문의 데이터를 복호화 할 수 없으므로 데이터에 대한 기밀성이 보장된다.

2) 익명성(Anonymity)

익명성이란 익명 메시지의 작성자가 누구인지 그룹원은 식별할 수 없어야함을 의미한다. 본 논문에서는 메시지에 대한 익명성을 제공하기 위해 메시지 서버(Server_M)와 검증 서버(Server_V)를 분리하여 구성하였다. 메시지 서버와 검증 서버간의 저장 내용을 분리 해 놓음으로써 각 컴포넌트가 그 메시지에 대하여 작성자가 누구인지 알 수 없다. 즉, 검증 서버를 통해 메시지 ID(MID)를 발급 받기 때문에 검증 서버는 메시지 ID(MID)와 작성자 식별 정보(SID)를 알고 있고, 메시지 서버는 익명 메시지 작성자로부터 메시지 ID(MID)와 메시지 내용(Msg)만을 메시지 서버의 공개키($AKey_+$)로 암호화하여 전달받기 때문에 익명 메시지 작성자 식별 정보(SID)를 알 수 없다.

표 4. 각 컴포넌트가 알고 있는 정보
Table 4. Component Information

	Server_M	Server_V	Sender	Receiver	Requestor
MID	O	O	O	X	X
SID	X	O	O	X	X
Msg	O	X	O	O	O
Num	O	X	O	O	O
$TS_{[MID]}$	X	O	O	X	X
$TS_{[Msg]}$	O	X	O	O	O
H[MID:Msg]	O	O	O	X	X
$H[SKey_{[V\text{-}Sen]}, MID, TS_{[MID]}]$	O	X	X	X	X

[표 4]는 그룹원이 실명 공개 요청을 하기 전(Step 1-Step 16)의 각 컴포넌트가 가지는 값을 나타낸다. 검증 서버(Server_V)에는 작성자 ID(SID)를 저장하고, 메시지 서버(Server_M)에는 익명 메시지 내용(Msg)을 저장하여 익명 메시지에 대한 식별 정보를 두개의 서버에 분산하여 저장하고 있다. 실명 공개 요청 시, 두 서버가 공통으로 가지는 메시지 ID(MID)를 이용하여 검증 서버(Server_V)와 메시지 서버(Server_M)간에 식별 정보를 주고 받음으로써 메시지 내용(Msg)에 대응하는 작성자 ID(SID)를 식별할 수 있다. 이처럼, 검증 서버(Server_V) 또는 메시지 서버(Server_M) 중 하나의 서버가 가지고 있는 정보만으로는 메시지 내용(Msg)을 작성한 작성자 ID(SID)에 대한 정보를 얻을 수 없다. 또한, 수신자인 그룹원(Receiver)의 경우 작성자 ID(SID), 메시지 ID(MID)에 대한 정보를 가지지 않기 때문에 익명 메시지 작성자(Sender)에 대한 정보를 얻을 수 없다.

즉, 메시지 서버(Server_M)와 검증 서버(Server_V)가 가지는 DB 테이블에 저장되어 있는 익명 메시지 작성자(Sender)에 대한 식별 정보(SID)와 메시지 정보(Msg)를 분리하여 저장함으로써 메시지 작성자의 익명성을 보장한다.

3) 무결성(Integrity)

무결성은 데이터를 허가되지 않은 사람 또는 시스템이 임의적으로 변경하지 못하게 하는 것을 의미하다.

[표 5]과 같이 해시함수는 평문(Plain Text)의 한글자만 변경이 되어도 해시함수(H[*])를 통해 출력되는 암호문(Cipher Text)의 결과는 크게 차이가 있다는 특징을 가지고 있다. 따라서 본 논문에서는 이러한 해시함수의 특징을 이용하여 모든 컴포넌트 간에 주고받는 데이터의 위변조를 방지함으로써 무결성을 보장한다.

표 5. 해시함수를 통한 값 비교
Table 5. Comparison of values through hash function

No.	Plain Text	Hash	Cipher Text
1	Hi	MD5	C1A5298F939E87E8F962A5EDFC206918
2	HI		DFCC685CFAAB38461D6D11B662A9C4E4
3	Ho		403CEFC528EDF4BACE1E2D3533DAD8FE
4	Hi	SHA-2	3639EFCD08ABB273B1619E82E78C29A7D F02C1051B1820E99FC395DCAA3326B8
5	HI		CD6F6854353F68F47C9C93217C5084BC66 EA1AF918AE1518A2D715A1885E1FCB
6	Ho		C758089F0833B093BBB4B6E796B9EF3738 08DDCCBCF0C36D0687BF3FF38E53F6

4-2 공격 및 위협에 대한 분석

1) 재전송 공격(Replay Attack)

재전송 공격은 네트워크 통신상에서 전송되는 데이터를 복사한 후 나중에 재전송함으로써 정당한 사용자로 가장하는 공격을 말한다.

본 논문에서는 전송하는 메시지에 대한 재전송 공격을 방지하기 위해 모든 컴포넌트 간에 주고받는 데이터에 Timestamp(TS)와 해시함수(H[*])를 사용하여 재전송 공격을 방지하고 있다. 메시지 작성할 때와 메시지 ID(MID) 생성 시 Timestamp(TS)를 생성하여 시스템상의 시간을 기록하여 같이 전송한다. 이를 통해 정당하지 않은 사용자가 사전에 복사한 메시지를 재사용하여 정당한 사용자로 가장하지 못하게 한다. 따라서 본 논문에서 제안하는 프로토콜은 재전송 공격으로 부터 안전하다.

2) 메시지 아이디(MID) 오용 위협

메시지 아이디 오용 위협은 작성자(Sender)가 메시지 ID(MID)를 악의적으로 수정하여 트래픽을 보내 메시지 ID(MID)를 오용함으로써 발생하는 위협을 말한다.

본 논문에서는 사전에 메시지 아이디 발급 과정(Step 1-Step 8)을 통해 작성자는 검증 서버(Server_V)로부터 메시지 ID(MID)를 발급받는다. 익명 메시지 작성자(Sender)는 익명 메시지를 작성하고 발급받은 메시지 ID(MID)와 메시지 내용(Msg)를 해시한 값(H[MID:Msg])을 검증 서버(Server_V)에 보낸다. 그리고 그룹내 메시지를 전송 요청을 위해 메시지 ID(MID), 메시지 내용(Msg), 메시지 작성시간($TS_{[Msg]}$), 작성자와 검증 서버간의 대칭키($SKey_{[V\text{-}User]}$)와 메시지 ID(MID), $TS_{[MID]}$의 해시값($H[SKey_{[V\text{-}Sen]}, MID, TS_{[MID]}]$)을 메시지 서버의 공개키($AKey_{[M]+}$)로 묶어서 보낸다. 메시지 서버(Server_M)은 익명 메시지를 배포하기 전 작성자로부터 받은 익명 메시지가 유효한 메시지인지 확인하기 위해 H[MID:Msg]값을 메시지 서버와 검증 서버 간의 대칭키($SKey_{[M\text{-}V]}$)로 암호화하여 보내 검증 서버에 저장된 값(H[MID:Msg])과 비교를 통해 검증한다. 이처럼 작성자가 메시지 전송 요청 시 작성자(Sender)가 전송

한 메시지 ID(MID)에 대하여 검증 서버(Server_V)를 통해 유효한지 검증함으로써 메시지 ID(MID)에 대한 오용 위협으로부터 안전성을 제공한다.

4-3 위험 분산

1) 권한 분산

본 논문에서는 비밀정보로 사용되는 KEY(SID)를 n개의 비밀조각으로 조각화한 것으로, t개의 비밀정보가 모이면 비밀조각 재조립이 가능한 Shamir의 (t, n)-임계치 방식을 이용하여 그룹원에게 익명 메시지 작성자를 공개한다. 비밀분산은 비밀정보를 분산하여 보관하는 기법으로, 비밀정보에 대한 복구를 원할 때 일부 정보가 손실 되어도 특정 조건이 만족 된다면 비밀정보에 대해 복구가 가능한 특성을 갖고 있다.

표 6. 분산된 비밀조각 정보
Table 6. Distributed Secret Information

Server_V	Segment Information having group members	
KEY(SID)→Segment[1-n]	Sender	Segment[i-1]
	Receiver	Segment[1_n] excluding i-1, i+1
	Requestor	Segment[i+1]

본 논문에서는 익명 메시지에 대한 비밀분산 조각화 과정(Step 19-Step 23)을 통해 작성자 공개를 위한 권한을 분산한다. 검증 서버(Server_V)는 작성자 ID를 암호화한 값(KEY(SID))의 암/복호화 키(KEY)를 조각화하기 위한 연산식($Seg_{[i]}(KEY)$)을 생성한 후 작성자(Sender)를 포함한 그룹원(Receiver)에게 메시지 번호(Num), 그룹원에 대응하는 서로 다른 연산식($Seg_{[1\_n]}(KEY)$)을 전달한다. 전달받은 비밀 조각의 정보는 [표 6]과 같다. 익명 메시지 작성자에 대한 실명 공개 요청을 하고자 할 때 그룹원(Receiver)은 자신의 DB에 저장되어 있던 연산식($Seg_{[1\_n]}(KEY)$)을 통해 비밀조각($Segment_{[i]}$)을 생성하여 메시지 서버(Server_M)에 전송하고, 메시지 서버(Server_M)는 t개(과반수이상)의 조각이 모였을 경우 비밀조각($Segment_{[i]}$)을 KEY로 재조립한다. 재조립된 KEY를 통해 KEY(SID)로부터 실명 검증에 필요한 정보인 작성자 ID(SID)를 복원할 수 있다. 즉, 그룹원 한명의 비밀정보($Segment_{[i]}$)로는 KEY를 재조립하여 작성자 ID(SID)를 추출할 수 없다. 이처럼 그룹원 한명이 익명 메시지에 대한 검증을 자신의 이익을 위해 남용하는 것을 방지하기 위해서 그룹원 전체에게 KEY를 조각화하여 나누어 저장함으로써 권한을 분리하였다.

2) 키 분실 위험 분산

키 분실에 대한 위험 분산은 비밀정보에 대한 키를 그룹원에게 분산하여 저장함으로써 키에 대한 분실/도난이 발생하여도 비밀정보를 복구할 수 있도록 위험을 그룹원에게 분산시키는 것을 말한다.

표 7. 비밀분산 사용에 따른 키 복구 가능 여부 비교
Table 7. Availability of Key restore

Use of Secret Share	Available of Restore
X	Impossible
O	Possible

본 논문에서는 익명 메시지에 대한 비밀정보의 조각화 과정(Step 19-Step 23)을 통해 사전에 검증 서버(Server_V)로부터 그룹원은 키(KEY)를 조각화하기 위한 연산식($Seg_{[i]}(KEY)$)을 나누어 가진다. 그룹원(Receiver)이 익명 메시지에 대한 작성자(Sender) 공개 요청 시, 사전에 나눠가진 연산식($Seg_{[i]}(KEY)$)을 이용해 비밀조각($Segment_{[i]}$)을 생성한 후 메시지 서버(Server_M)에 전송함으로써 과반수이상의 동의가 있으면 비밀정보인 작성자 ID(SID)를 알아낼 수 있다.

이 과정에 있어 작성자(Sender)를 제외한 그룹원(Receiver)의 비밀정보에 대한 분실, 도난이 발생할 수 있다. 정보를 한사람이 가진 경우, KEY와 같은 특정 값 또는 비밀조각을 분실하게 되었을 경우 비밀정보에 대하여 복원이 불가능 하다. 하지만 제안하는 프로토콜은 Shamir의 비밀분산을 사용하여 비밀조각($Segment_{[i]}$)이 t개(특정조건)보다 적은 비밀조각이 분실되었을 경우, 분실되지 않은 다른 그룹원의 연산식($Seg_{[i]}(KEY)$)을 통해 비밀 조각(Segment[i])을 생성하여 t개가 모이면 비밀정보인 작성자 ID(SID)를 복원할 수 있기 때문에 데이터에 대한 복원 확률이 높다. [표 7]과 같이 비밀분산을 사용하지 않은 방식과 비교했을 때 키 분실로 인한 데이터 복구를 못하는 상황에 대한 위험을 줄일 수 있다.

3) 서버 정보 유출 위험 분산

서버 정보 유출 위험은 악의적인 공격을 통해 서버의 DB 테이블 정보가 외부로 유출 되는 위험을 말한다.

표 8. 검증 서버와 메시지 서버 DB정보가 가지는 식별 정보
Table 8. Identified Information of Server_V, Server_M

Identified Information	Server_V	Server_M
Different information	SID	Msg
Common information	MID	

본 논문에의 프로토콜은 메시지 아이디 발급 과정(Step 1-Step 8)을 통해 검증 서버(Server_V)는 메시지 작성자 ID(SID)와 메시지 ID(MID)에 대한 정보를 가지고, 메시지 서버(Server_M)는 메시지 내용(Msg)과 메시지 ID(MID)에 대한 값을 가지도록 분산 저장하고 있다. 이처럼 메시지에 대한 익명성을 유지시키기 위한 중요 정보를 분산하여 저장함으로써 하나의 DB 테이블 정보가 유출되어도 익명 메시지 내용(Msg) 또는 작성자 ID(SID) 정보를 알 수 없게 설계함으로써 단일 서버의 정보 유출에 대한 위험을 분산시킬 수 있다.

Ⅴ.결 론

본 논문에서는 SNS (채팅앱, 메신저, 게시판 등) 그룹내 메시지 교환에 활용할 수 있는 비밀분산을 이용한 익명 서비스 제공을 제공하는 효율적인 프로토콜을 제안하고 이에 대한 보안성 분석을 통해 안전성을 분석하였다. 본 프로토콜은 익명 서비스 제공과 더불어, 추후 익명 메시지의 실명 공개가 필요할 경우, 특정 조건이 만족되면 비밀정보를 밝힐 수 있도록 비밀분산 기법을 이용하여 메시지 작성자를 공개하고 검증할 수 있는 기능을 제시하여, 익명성의 역기능을 방지한다. 또한 재전송 공격, 메시지 아이디 오용 위협, 서버의 DB정보 유출 위협 등 다양한 공격으로부터 안전함을 도출하였다.

향후 전자화폐, 전자투표, 오픈 채팅, 전자상거래, 사물인터넷, 스마트 그리드 등의 익명성이 요구되는 사례의 특성에 맞게 최적화하는 연구를 진행한다면, 본 논문에서 제안한 프로토콜이 다양한 분야에서 활용될 수 있을 것으로 기대된다.

감사의 글

본 논문은 한국연구재단 이공분야 기초연구사업 중견연구자 지원사업 (2017R1A2B1005285)의 지원으로 연구됨

참고문헌

[1] Y. H. Kim, Social Network Service (SNS) Usage Trend and Usage Behavior Analysis, *KISDI STAT Report*, Vol. 18-11 pp. 1-7, 2018.

[2] T. T. Wang, A Comparative Analysis on Using Pattern of Mobile Messenger between Korea and China, Master dissertation, Tongmyong University, Nam-gu, 2014.

[3] Anti·Corruption and Civil Rights Commission. Project to encourages people to participate in government to improve policies. (SINMOONGO) [Internet]. Available: https://www.epeople.go.kr/jsp/user/UserMain.jsp

[4] H. J. Lee, A Study on Factors Affecting the Investment Intention of Information Security, *Journal of Digital Contents Society,* Vol.19(no.8), 2018.

[5] I. W. Park, M. H. Kim, "The Effects of Anonymity on Arguments and Flaming in Discussion through a Synchronous Computer Mediated Communication", *The Journal of Educational Technology,* Vol.16, No. 4, pp. 91-106, 2000.

[6] J. S. Park, The Effect of Anonymity of Computer-Mediated Communication on The Organizational Communication, Master dissertation, DongGuk University, Jung-gu, 2001.

[7] H. K. Oh, A Study on the Fair Electronic Cash System with Anonymity Control, Master dissertation, SoonChunHang University, Sinchang-myeon, 1999.

[8] C. D. Kim, Anonymous Fair Exchange Scheme for E-Commerce Protocol, Master dissertation, HanYang University, Seongdong-gu, 2003.

[9] J. S. Moon, Authentication and Key Agreement Protocol Preserving Anonymity in the Wireless Internet, Master dissertation, Dong-A University, Saha-gu, 2002.

[10] J. Y. Lee, A Study on Authentication System using Secret Sharing Scheme, Ph.D. dissertation, DaeJeon University, Daejeon-si, 2006.

[11] J. Y. Choi, Study on the different anonymity perceptions impacting on posting malicious message according to types of online communities, Master dissertation, HanYang University, Seongdong-gu, 2016.

[12] K. A. Na, A Study on role of anonymity which influence the communication contents on the discussion using internet bulletin board, Master dissertation, YeungNam University, Gyeongsan-si, 2004.

[13] S. A. Lee, A Study on Ethics Consciousness among Internet Users with a Focus on Anonymity, Master dissertation, HongIk University, Mapo-gu, 2003.

[14] A. Shamir, How to Share a Secret, *Communications of the ACM*, vol.22, pp. 612-613, 1979.

[15] Y. W. Song, Design of a Secret Sharing Scheme in a Tree-structured Hierarchy, Master dissertation, Ewha Womans University, Seodaemun-gu, 2003.

[16] S. M. Yang, Design of a Reusable Secret Sharing Scheme in a Hierarchical Group, Master dissertation, Ewha Womans University, Seodaemun-gu, 2003.

[17] K. B. Kim, A Threshold Secret Sharing Scheme Practicable Secret Key Combination According to Authority, Master dissertation, HanYang University, Seongdong-gu, 2008.

[18] Advanced Encryption Standard(AES) : NIST Federal Information Processing Standards Publication 197, 2001.

[19] D. W. Kim, Cryptographic Algorithm and Key Length Use Guide, KISA, Songpa-gu, pp. 1-15, 2013.

[20] D. W. Kim, Cryptographic Use Guide, KISA, Songpa-gu, Vol.2010-23, pp. 1-108, 2010.

권유진(Yu-Jin Kwon)

2017년 : 명지대학교 컴퓨터공학과 (학사)

2017년~현 재 : 명지대학교 보안경영공학과 (석사과정)
※관심분야 : 정보보호(Personal Information), 암호학, IoT, 네트워크 보안

김정운(Jung-Woon Kim)

2012년 : 청강문화산업대학 사이버정보보안학 (전문학사)
2015년 : 국가평생교육원 정보보호학 (학사)
2018년 : 명지대학교 대학원 보안경영공학과 (석사)

※관심분야 : 정보보호(Personal Information), ISMS, IoT, 네트워크 보안

남기원(Ki-Won Nam)

1999년 : 중앙대학교 (학사)
2001년 : 중앙대학교 대학원 (석사)
2013년 : 중앙대학교 대학원 (박사)

2014년~현 재 : 중앙대학교 교수
※관심분야 : 컴퓨팅사고, 소프트웨어교육, 로봇활용 교육프로그램, 피지컬컴퓨팅

한승철(Seung-Chul Han)

1995년 : 서강대학교 (학사)
2003년 : 퍼듀대학교 대학원 (석사)
2007년 : 플로리다대학교 대학원 (박사)

2008년~현 재 : 명지대학교 컴퓨터공학과 교수
※관심분야 : 정보보호(Personal Information), 모바일, 컴퓨터 보안