情報保護學會論文誌
第 22 卷 第 3 號, 2012. 6

ID 기반 암호 기법을 이용한 SCADA 시스템에서 비밀 키 관리 및 복구 방안

오 두 환,[1†] 최 두 식,[1] 나 은 성,[2] 김 상 철,[2] 하 재 철[1‡]
[1]호서대학교 정보보호학과, [2]에스지에이(주)

Key Management and Recovery Scheme over SCADA System Using ID-based Cryptosystem

Doo-Hwan Oh,[1†] Doo-Sik Choi,[1] Eun-Sung Na,[2] Sang-Chul Kim,[2] Jae-Cheol Ha[1‡]
[1]Dept. of Information Security, Hoseo University, [2]SGA Co., Ltd.

요 약

SCADA(Supervisory Control and Data Acquisition) 시스템은 국가의 중요한 기반 망인 전력, 가스, 상하수도 등을 제어하기 위해 사용된다. 이전의 폐쇄적인 통신 환경과 달리 최근 개방 통신 환경을 사용함에 따라 안전한 통신을 위한 키 관리 기술이 연구되고 있다. 본 논문에서는 SCADA 시스템을 구성하는 MTU(Master Terminal Unit), Sub-MTU, RTU(Remote Terminal Unit)에 각각 페어링을 이용한 ID 기반의 키 관리 방안을 제시한다. 또한, 예상치 못한 사고나 악의적인 공격들로 인하여 장치들의 개인 키가 노출되거나 또는 KMS(Key Management System)의 마스터 키가 노출되었을 경우에도 SCADA의 키 관리 시스템을 복구할 수 있는 방법을 제시한다.

ABSTRACT

The SCADA(Supervisory Control and Data Acquisition) systems are used to control some critical national infrastructures such as electricity, gas, and water distribution systems. Recently, there are many researches on key management scheme for secure communication due to change to the open network environment. We propose a new key management method which is established on ID-based cryptosystem using pairing on MTU(Master Terminal Unit), Sub-MTU, and RTU(Remote Terminal Unit). Furthermore, we present a redistribution protocol of private key of each device and a system recovery protocol as a countermeasure of exposure of KMS(Key Management System) master key which is occurred by some unexpected accidents or malicious attacks.

Keywords: SCADA System, ID-based Cryptosystem, Key Management

I. 서 론

SCADA 시스템은 국가의 주요 기반망인 전력, 가스, 상하수도 등을 제어하기 위해 사용된다. 원격지에 위치하고 있는 RTU는 다양한 정보를 수집하여 상위 MTU 또는 Sub-MTU에 전송하고 MTU들은 하위 RTU들을 제어한다[1]. 최근, SCADA 시스템이 이

접수일(2011년 8월 16일), 수정일(2011년 12월 12일), 게재확정일(2011년 12월 12일)
† 주저자, odhwan@naver.com
‡ 교신저자, jcha@hoseo.edu

전v의 폐쇄적인 통신 환경과 달리 개방 통신 환경으로 전환됨에 따라 안전한 통신을 위한 키 관리 기술들이 연구되어 왔다. 기존의 SCADA 시스템에서의 키 관리 기술들은 각 RTU나 MTU, Sub-MTU에 미리 장기간 사용하는 키(Long Term Key, LTK)가 주어지고 공개 키 혹은 대칭 키 암호화 방식을 이용하여 세션 키를 공유하기 위한 프로토콜이었다[1, 2]. 이외에도 RTU가 원격에 위치하여 SCADA 시스템으로 가입되거나 탈퇴했을 때 키의 갱신성(freshness), 그룹 키 안전성, 전방향 안전성, 후방향 안전성을 만족시키면서 세션 키 혹은 그룹 키를 생성하는 프로토콜이 있었다[3, 4]. 이러한 프로토콜은 SCADA 시스템을 운영하는 환경이나 장비의 위치에 따라 각기 다른 장·단점을 가지고 있어 절대적인 안전성 및 효율성 분석은 시스템을 보는 관점마다 달라질 수 있다. 그리고 기존의 프로토콜들은 각 장치들간에 공유된 LTK에 대한 분배 및 업데이트에 대한 내용을 다루지 않고 있다. 원칙적으로 LTK의 사용 기간은 장치의 수명과 일치하는 영구적 목적의 키는 아니므로 일정 기간이 지나면 갱신이 이루어져야 한다. 또한, SCADA 시스템은 일반 공중망과 달리 국가 주요 기반 시설 관리를 담당하기 때문에 예기치 못한 사고나 여러 악의적인 공격 등 최악의 상황까지 고려한 대응책이 필요하다.

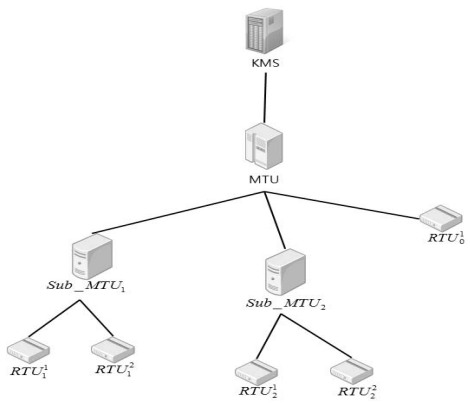
본 논문에서는 SCADA 시스템을 구성하는 MTU, Sub-MTU, RTU에 각각 페어링을 이용한 ID 기반의 키 관리 방안을 제시한다. 또한, 데이터 암호용 세션 키를 공유하기 위해 LTK를 분배하는 프로토콜을 제안한다. 그리고 예상치 못한 사고나 악의적인 공격들로 인하여 장치들의 개인 키가 노출되거나 또는 KMS(Key Management System)의 마스터 키가 노출되었을 경우에도 SCADA의 키 관리 시스템을 복구할 수 있는 방법을 제시한다. 이전의 SCADA 시스템에서 RTU는 낮은 연산 속도와 작은 저장 공간을 가지고 있었지만, 컴퓨팅 속도 및 저장 장치의 발달과 함께 페어링을 이용한 ID 기반 암호화 시스템에서 연산 속도를 줄이는 다양한 방법들이 제안되었기 때문에 RTU에서도 충분히 구현될 수 있다[5, 6].

본 논문의 2장에서는 관련 연구로서 SCADA 시스템과 이전에 제안된 SCADA 시스템에서의 키 관리 프로토콜을 설명한다. 3장에서는 인증 기능을 가진 ID 기반 암호화 시스템을 소개하고, 4장에서는 제안하는 ID 기반 키 분배 및 복구 프로토콜을 설명한다. 5장에서는 제안 방식을 비교 및 분석을 수행하고, 6장에서 결론을 맺는다.

II. SCADA 시스템과 키 관리 기술 관련 연구

2.1 SCADA 시스템의 구조

SCADA 시스템은 일반적으로 MTU, RTU 등 여러 개의 장비가 계층적인 구조로 이루어져 있다. 여기서 계층은 시스템의 운용 환경이나 거리, 장비 성능 등의 여건에 따라 여러 층으로 모델링 될 수 있다. 시스템 구조가 복잡할수록 키 관리 및 분배 혹은 폐기가 어려워지는데 본 논문에서는 SCADA 시스템의 구조를 [그림 1]과 같이 단순화하였다. 즉, SCADA 시스템에는 여러 개의 MTU가 존재하며 상위에 위치하는 MTU는 하부의 Sub-MTU 또는 RTU들과 연결되어 있다. 관리자는 HMI(Human-Machine Interface)를 통하여 MTU와 연결되어 데이터 통신을 수행한다. 특별히, 관리자는 키 분배 및 관리를 위해 KMS을 운영하며 KMS는 각 MTU와 직접 연결되어 있다. [그림 1]에서 계층구조상 KMS는 최상위에 위치하며 그 밑에 MTU가 위치한다. 각 MTU는 하부에 위치한 RTU 또는 Sub-MTU들을 관리하기 위한 노드로 하부 장치들을 제어한다. 또한, Sub-MTU는 장치의 설치 거리나 환경을 고려하여 MTU의 역할을 분담하면서 하부 RTU 장치들을 제어한다. RTU는 센서를 제어하는 마이크로프로세서와 물리 환경과 상호작용을 수행하는 장치로 구성되는데 일반적으로 안전하지 않은 원격에 위치한다고 가정한다. 따라서 RTU와 MTU(혹은 Sub-MTU)와의 기밀성 유지 및 키 관리 방안이 매우 중요한 요소가 된다. HMI는 관리자가 SCADA 시스템과 상호작용을 수행하기 위



[그림 1] SCADA 시스템의 구조

한 장치로서 MTU에 연결되어 전체 SCADA 시스템을 관리하는데 PDA, 웹 브라우저, 데스크톱 PC로 구현될 수 있다. KMS는 시스템 전체의 키를 생성하고 관리하는 시스템으로서 SCADA 시스템의 보안과 관련한 핵심적인 역할을 하게 된다.

논문 [2]에서는 SCADA 시스템을 구성할 경우의 여러 가지 제약 사항을 제시하고 있는데 그 중에서 RTU는 물리적으로 불안전한 곳에 위치하지만 MTU 혹은 Sub-MTU와 시간 동기화가 이루어진다고 가정하고 있다. 또한 SCADA 시스템은 다음과 같은 보안 요구 사항을 만족하도록 설계되어야 함을 전제로 하고 있다.

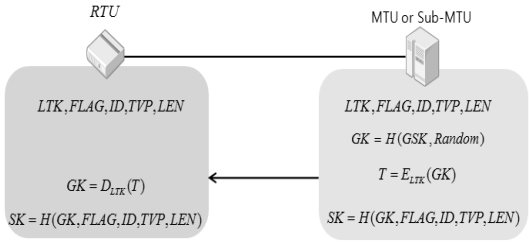
- 기밀성 (Confidentiality) : 정당하지 않은 사용자로부터 정보 및 자원에 대한 접근을 제한한다.
- 무결성 (Integrity) : 데이터가 변경되지 않았음을 보장한다.
- 가용성 (Availability) : 정보 또는 자원 사용 요구에 대하여 지속적인 서비스를 제공해야 한다.

기밀성은 주로 데이터의 암호화를 통해 제공할 수 있으며 무결성은 비밀 키를 이용한 해쉬(Hash) 알고리듬을 이용하여 제공할 수 있다. 가용성은 DoS (Denial of Service)와 같은 공격에 대응하는 기능을 말한다. 이 중에서 기밀성과 무결성을 제공하면서 데이터를 전송하기 위해서는 두 통신자간의 비밀 키가 필요하며 이 키는 매 통신 시 마다 달라야 하므로 비밀 세션 키가 필요하다. 결국, RTU는 MTU(혹은 Sub-MTU)와의 안전한 통신을 위해서는 세션 키가 필요하며 이를 효과적으로 공유하는 것이 SCADA 시스템 보안의 핵심 기능이 된다.

2.2 SCADA 시스템의 키 관리 프로토콜 연구

2.2.1 SKE

SKE(Sandia Key Management)는 Beaver 등이 제안한 SCADA 시스템에서의 키 관리 프로토콜로서 사용되는 통신 형태를 2가지로 분류하고 있다 [1]. 첫 번째는, MTU와 RTU 또는 Sub-MTU와 RTU 같이 계층적인 구조의 통신은 C-S(Controller to Subordinate) 방식을 이용하며, 두 번째로 Sub-MTU들 간의 통신은 P2P(Peer to Peer) 방식을 이용한다. [그림 2]는 MTU 또는



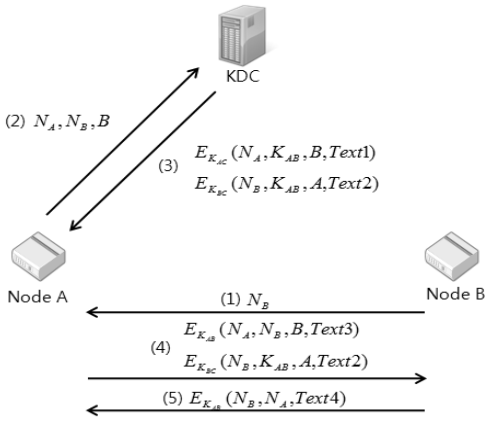
[그림 2] SKE에서 세션 키 설정 과정

Sub-MTU와 RTU 간의 세션 키를 설정하는 과정을 나타낸다. 이 때, MTU 또는 Sub-MTU와 RTU 간에는 미리 LTK, FLAG, ID, TVP(Time Varying Parameter), LEN(Length of data)을 공유하고 있어야 한다. 제어 장치와 종속 장치는 LTK를 수동으로 공유하고 있으며 GSK(General Seed Key)와 랜덤한 비트 열을 해쉬하여 GK(General Key)를 생성하며 이렇게 생성된 GK는 LTK로 암호화되어 종속장치로 전송된다. 만약 GK가 사용되는 기간은 보안 정책에 따라 다르지만 특별한 상황이 발생하게 되면 갱신되어야 한다. C-S 방식을 이용할 경우에는 대칭 키를 이용하는 반면 Sub-MTU들 간의 통신은 P2P(Peer to Pee) 방식을 사용하는데 이 경우는 공개 키 암호 알고리듬을 사용하여 CK(Common Key)를 만들어 사용하게 된다. 이 CK가 C-S 방식에서는 GK와 같은 역할을 하게 된다.

2.2.2 SKMA

SKMA(Key Management Architecture for SCADA)는 2006년에 R. Dawson 등에 의해 제안되었는데 SCADA 시스템의 제약사항 및 보안 요구사항들의 분석과 함께 새로운 RTU가 SCADA 시스템에 등록되었을 경우, RTU 간의 세션 키 설립을 통해 안전한 통신을 제공한다[2]. SKMA는 대칭 키 암호 알고리듬으로만 이루어져 있으며 사용되는 키의 형태는 다음과 같다. 여기서 노드는 MTU, Sub-MTU, RTU 모두가 될 수 있다.

- K_{AC}(long term node-KDC key) : 노드 A와 KDC C 사이에 사전에 공유된 키
- K_{AB}(long term node-node key) : 노드 A와 노드 B 사이에 공유된 키로서 노드 추가 시 생성함
- 세션 키 : 메시지를 암호화하기 위해 사용되는 키



〔그림 3〕 SKMA에서 세션 키 설정 과정

〔그림 3〕은 SKMA에서 노드와 노드 간에 키를 설립하는 과정을 나타낸 것이다. 두 노드간의 공유 키 K_{AB}를 생성하기 위해서는 항상 KDC와 통신을 통해 분배받아야 하는 번거로움이 있다. 따라서 노드 간 통신을 위해 고속의 대칭 키 암호 알고리듬을 사용하지만 통신량이 많다는 것과 KDC의 부하가 많이 걸린다는 단점을 가지고 있다.

2.2.3 ASKMA

ASKMA(Advanced Key Management Architecture for Secure SCADA Communications)는 Choi 등이 제안한 SCADA 시스템에서의 키 관리 기법으로 LKH(Logical Key Hierarchy) 구조를 이용한 방식이다〔3〕. ASKMA 키 관리 구조에서 상위의 MTU와 Sub-MTU들 간의 관계는 이진트리 형식으로 구성되며, MTU 또는 Sub-MTU와 RTU들 간에는 n-ary 트리 형식으로 구성된다. ASKMA에서는 브로트 캐스팅을 지원하는 그룹 키 관리 기법을 제공하며, 이를 위하여 새로운 RTU가 SCADA 시스템에 가입하거나 기존의 RTU가 SCADA 시스템으로부터 탈퇴할 겨우, 그룹 키의 안정성을 보장한다. 또한, Choi 등은 이 프로토콜을 개선한 ASKMA+를 제안하기도 하였다〔4〕. 그러나 SCADA 시스템에서 그룹 키를 사용한 브로드 캐스팅 기능이 꼭 필요한 것인지는 여러 의견이 상존하고 있다.

III. ID 기반 암호화 시스템

3.1 ID 기반 암호화 시스템 개요

ID 기반 암호화 시스템에 대한 개념은 1985년 Shamir에 의해 처음 제안되었으나 구현상의 어려움과 시스템을 폐쇄적으로 운영해야 한다는 성질 때문에 많이 연구되지는 못하였다〔7〕. 그러나 2000년, Ohgishi, Sakai, Kasahara는 처음으로 타원곡선상의 페어링에 기반한 새로운 ID 기반 암호화 알고리듬을 제안하면서 주목을 받기 시작하였다〔8〕. 그 후 2001년, Boneh와 Franklin은 Weil 페어링을 이용하여 실현 가능한 ID 기반 B-F 암호화 알고리듬을 제안한 후에 2003년에 이 알고리듬을 확장하였다〔9〕. 또한, Lynn은 암호문 수신자 측에서 송신자를 인증할 수 있는 인증 가능한 B-F ID 기반 암호화 시스템(Authenticated ID-based Encryption)을 제안하였다〔10〕. 현재 많이 사용 중인 RSA와 같은 공개 키 기반 암호화 시스템은 공개 키 기반 구조(Public Key Infrastructure, PKI)에 기반하므로 공개 키의 유효성 검증 및 폐기 그리고 관리 기능을 수행하여야 한다. 반면, ID 기반 암호화 시스템은 공개 키로 E-mail이나 전화번호, 식별 정보 등의 이미 알려진 정보들을 사용하기 때문에 공개 키에 대한 유효성 검증이 필요없다는 큰 장점을 가지고 있다. 또한, ID 기반 암호 시스템에서는 키 관리 센터가 각 사용자의 키 생성단계를 수행하므로 비밀 키를 모두 센터가 알고 있어 일반 공중망에서 일반 사용자를 대상으로 하는 암호 시스템으로는 부적합하지만 강력한 신뢰성을 기반으로 하는 폐쇄적인 그룹망에 유용하게 사용될 수 있다. 이런 의미에서 SCADA 시스템과 같이 관리자의 강력한 통제력을 필요로 하는 일정한 규모의 폐쇄 그룹에서는 공중망이 아니므로 PKI 기반 구조 보다는 ID 기반 암호 시스템이 효율성이나 보안성 측면에서 보다 효과적으로 사용될 수 있다.

PKI 기반 암호 시스템은 기본적으로 개인의 정보 보호를 중요시하므로 비밀 키를 개인이 먼저 생성하고 그에 대응하는 공개 키를 만든 후 이 공개 키를 공인 인증 기관에 의뢰하여 인증 받게 된다. 따라서 관리자라도 개인의 비밀 키를 알 수 없으며 사용자의 공개 키만 인증하게 된다. 이 경우 개인 키가 노출되거나 하는 위급 상황 시에는 키를 폐기하고 다시 신분 인증 과정을 통해 새로이 발급 받아야 한다. 만약 PKI 기반 암호 시스템을 SCADA 시스템에 적용시킨다면 개인 키 및 공개 키 생성을 각 장치에서 수행 할 수 있는 능력이 있어야 한다. 그리고 관리자가 공개 키 인증서를 발급하고 암호 시 마다 공개 키를 검증해야 한다. 또한, 장치의 개인 키가 노출되는 상황이나 시스템 전

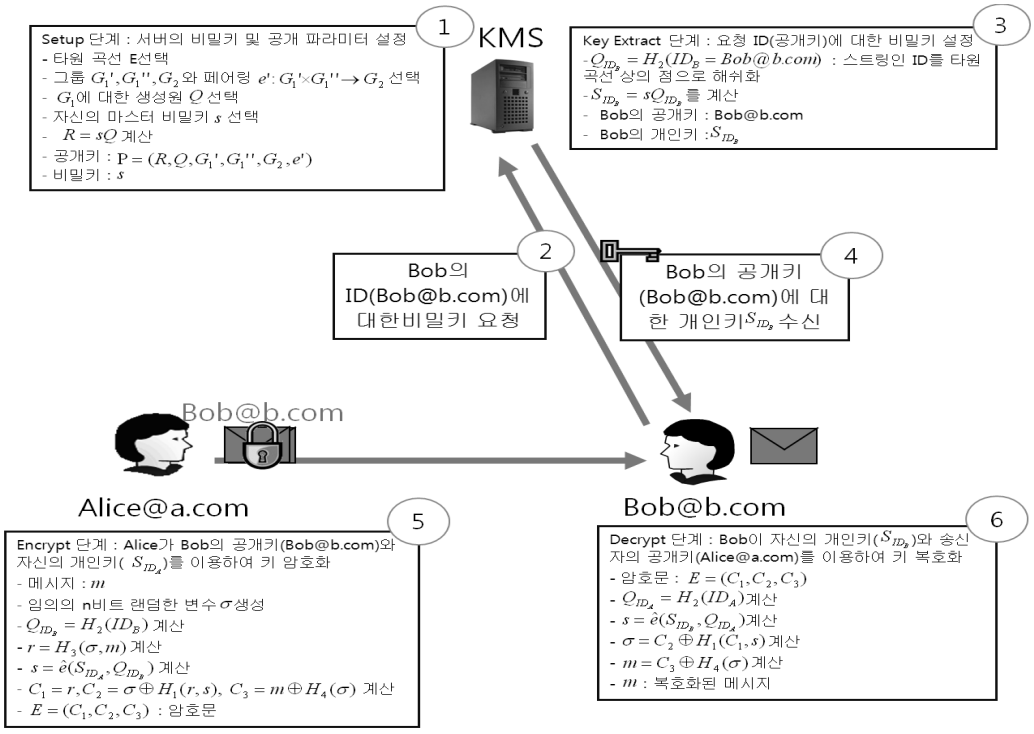
체를 바꾸어야 할 상황이 발생하게 되면 PKI 암호 시스템은 이를 복구하거나 인증하는데 어려움이 많다. 즉, 모든 장치마다 개인 키 및 공개 키를 생성하여 공개 키 인증을 받아야 하지만 원격지에 떨어진 장치들을 하나씩 확인하고 인증해야 한다.

이에 반해 ID 기반 암호 시스템은 관리자가 각 장치의 비밀 개인 키를 모두 알 수는 있지만 암호 통신 시 공개 키 인증과정이 필요없으며, 특히 응급 상황 발생에 따른 개인 키 갱신이나 시스템 복구 시 별도로 원격지의 장치를 인증할 필요가 없다. 단지, 키 관리 센터에서 새로운 개인 키를 생성하거나 시스템 파라미터를 비밀로 전송하면 쉽게 시스템 복구가 가능하다. 이에 관한 구체적인 내용은 다음 장에서 설명한다.

3.2 인증 기능이 있는 B-F ID 기반 암호화 시스템

ID에 기반한 암호 알고리듬 중 B-F 암호 알고리듬은 송신자 인증을 제공하지 않는다[9]. 따라서, 어느 사용자든지 수신자의 ID를 알고 있으면 암호문을 생성할 수 있기 때문에 악의적인 사용자가 정당한 송신자를 가장하여 수신자의 ID를 이용해 암호화를 수행할 수 있다. 이러한 단점을 극복하기 위해 암호화 시 송신자의 개인 키를 포함시키고 복호화 시 송신자의 ID, 즉 공개 키를 이용함으로써 송신자 인증을 수행할 수 있다. Lynn에 의해 제안된 인증 기능이 있는 B-F 암호화 알고리듬은 Setup, Key Extract, Encrypt, Decrypt 단계로 이루어져 있다[10]. Setup 단계는 KMS에서 자신의 마스터 비밀 키와 다양한 공개 파라미터를 설정하는 단계이며, Key Extract 단계는 요청된 사용자의 ID에 대한 개인 키를 생성하는 단계이다. Encrypt 단계는 송신자가 수신자의 ID와 송신자의 개인 키를 이용하여 암호화를 수행하며, Decrypt 단계는 송신자의 ID와 수신자의 개인 키를 이용하여 복호화를 수행하는 단계이다. [그림 4]는 인증 기능이 있는 B-F 암호화 알고리듬을 나타낸 것이고 세부적인 절차는 다음과 같다.

① Setup 단계 : KMS에서 수행되며 마스터 비밀 키 s는 개인 키로서 오직 KMS만이 생성하여 알고 있어야 한다. Q는 타원곡선 $E(F_p)$ 상에서의 임의의 점이며 KMS는 공개 파라미터인 $R=sQ$를 계산한다.



[그림 4] 인증 기능이 있는 B-F 암호 알고리듬

② Key Extract 단계 : KMS에서 수행되며 사용자는 자신의 E-Mail이나 MAC 주소 같은 ID를 KMS에 전송하면 KMS는 스트링 형태인 ID를 타원곡선 상의 점 Q_{ID}로 변환하며, ID 기반 암호화 시스템에서 Q_{ID}는 공개 키이고 $S_{ID}=sQ_{ID}$는 개인 키이다. KMS는 안전한 채널을 이용하여 생성된 S_{ID}를 개인 키 요청 사용자에게 전송한다.

③ Encrypt 단계 : 송신자에서 수행되며 수신자의 공개 키인 ID_B와 송신자의 개인 키인 S_{ID_A}를 이용하여 메시지 m을 암호화하기 위한 단계로서, 송신자는 랜덤한 σ를 선택하고 $Q_{ID_B}=H_2(ID_B)$, $r=H_3(\sigma,m)$, $s=\hat{e}(S_{ID_A},Q_{ID_B})$를 계산한다. 그 다음 송신자는 $C_1=r$, $C_2=\sigma\oplus H_1(r,s)$, $C_3=m\oplus H_4(\sigma)$을 계산하고 암호문 (C_1, C_2, C_3)을 수신자에게 전송한다.

④ Decrypt 단계 : 수신자는 자신의 개인 키인 S_{ID_B}와 송신자의 공개 키인 ID_A를 이용하여 복호화를 수행한다. 수신자는 자신의 개인 키인 S_{ID_B}와 송신자의 공개 키인 ID_A를 이용하여 $s=\hat{e}(Q_{ID_A},S_{ID_B})$를 계산할 수 있고 $\sigma=C_2\oplus H_1(C_1,s)$와 $m=C_3\oplus H_4(\sigma)$을 계산하여 m을 복호화할 수 있다.

IV. 제안하는 ID 기반 키 분배 방식

4.1 키 관리 구조

본 논문에서는 SCADA 시스템에서 사용할 수 있는 ID에 기반한 키 분배 및 관리 시스템을 제안하고자 한다. 제안하는 방식에서 사용되는 키는 다음과 같다.

① ID(공개 키)/ 개인 키 : ID를 공개 키로 하는 MTU, Sub-MTU, RTU들의 공개/개인 키 쌍으로서 ID가 변하지 않는 이상 원칙적으로 장치에 영구적으로 사용된다. 이 키를 이용하여 LTK 분배를 수행한다.

② LTK : MTU와 Sub-MTU간 혹은 MTU(혹은 Sub-MTU)와 RTU 간에 공유되는 대칭 키로서 Sub-MTU는 MTU 및 RTU 통신을 해야 하므로 2개를 관리한다. 갱신 주기는 관리자의 키 관리 정책에 의해 결정되며 보안도를 고려하여 장기간 사용된다. LTK는 SCADA 계층 구조에서 하위 계층 장치의 요구에 의해 상위 계층의 장치가 생성하며 상대방의 공개 키로 암호화한 후 전송된다.

③ 세션 키 : MTU와 Sub-MTU간 혹은 MTU(혹은 Sub-MTU)와 RTU간 데이터 암호 통신에 사용되는 대칭 키로서 LTK로 암호화 되어 분배된다. 매 세션 통신시 마다 새로운 키로 공유하여 사용한다. 세션 키는 SCADA 계층 구조의 상위 하위 구조와 관계없이 송신자의 필요에 의해 생성하여 데이터를 암호화한 후 전송한다.

④ EK(Emergency Key) : 각 MTU, Sub-MTU, RTU에게 발급되는 대칭 키로서 개인 비밀 키 노출이나 시스템 복구 시와 같은 긴급 상황 시에만 사용된다. 이 키는 각 장치에서 물리적으로 안전한(temper-resistant) 공간에 저장되며 관리자가 관리하며 각 장치와 통신할 수 있는 유일한 키이다.

본 논문에서는 SCADA 시스템 구조를 〔그림 1〕과 동일하게 가정하였으며 제안하는 키 분배 및 관리를 위한 각 장치들이 수행하는 역할을 요약하면 다음과 같다.

① KMS : 최상위에 위치하며 MTU, Sub-MTU, RTU의 개인/공개 키 생성 및 업데이트를 담당한다. 각 장치들과는 비상시에 사용할 수 있는 EK를 생성 및 분배한다. 또한 특정 장치의 키가 노출되거나 시스템이 파괴되었더라도 EK를 가진 장치에서 공개/개인 키 쌍을 복구하는 역할을 수행한다. KMS는 각 장치들 간의 통신에는 관여하지 않는다.

② MTU : MTU와 Sub-MTU 또는 RTU들 간의 LTK를 생성하고 공유하게 된다. 이 경우 LTK를 Sub-MTU의 공개 키로 인증 암호화하여 전송한다. 또한, 초기 설정 시 Sub-MTU 및 하부 RTU에게 EK를 전달하는 역할을 하며 응급 복구 시 EK로 암호화된 데이터를 전달하게 된다.

③ Sub-MTU : Sub-MTU와 RTU 간의 LTK를 생성 및 공유하게 된다. 이 경우 LTK를 RTU의 공개 키로 인증 암호화하여 전송한다. 또한, 초기 설정 시 RTU에게 EK를 전달하는

역할을 하며 응급 복구 시 EK로 암호화된 데이터를 전달하게 된다.

④ RTU : MTU 혹은 Sub-MTU와 LTK를 공유하게 되며 LTK를 통해 세션 키를 공유할 수 있게 된다. 실제 데이터를 세션 키로 암호화하여 상위 계층으로 송신하게 된다. 제안하는 키 분배 및 관리 프로토콜을 설명하기 위해 사용된 표기법을 정리하면 〔표 1〕과 같다. 논문에서는 설명의 편의를 위해 MTU는 하나(MTU_0)이며 그 하부에 Sub-MTU가 m개 있다고 가정한다. 따라서 RTU는 MTU나 Sub-MTU 밑에 위치하게 된다.

4.2 ID 기반 LTK 분배 프로토콜

SCADA 시스템에서 안전한 암호 통신을 위해서는 LTK 분배가 핵심이 된다. 이 키를 이용하여 세션 키를 분배하기 때문에 주기적인 LTK 분배를 위해서는 다음과 같은 초기 환경이 갖추어져 있어야 한다. 각 MTU, Sub-MTU, RTU들에게는 공개 키 역할을 수행하는 자신의 유일한 ID와 이에 대응하는 개인 키를 KMS로부터 발급받아 저장하고 있어야 한다. 또한, 각 장치의 ID에 날짜 정보를 포함하여 ID에 대한 유효 기간을 설정할 수 있어 각 장치의 ID와 이에 대응하는 개인 키 또한 쉽게 업데이트 할 수 있다. 이 외에도 MTU, Sub-MTU, RTU들은 초기 장비 설정 시 KMS와 사전에 공유하고 있는 EK를 발급받아 시스템을 복구할 때 사용한다. 제안하는 ID에 기반한 키 분배 프로토콜은 MTU와 Sub-MTU, MTU 또는 Sub-MTU와 RTU들 간의 LTK 분배 프로토콜로 구별된다.

4.2.1 MTU와 Sub-MTU 간의 LTK 분배

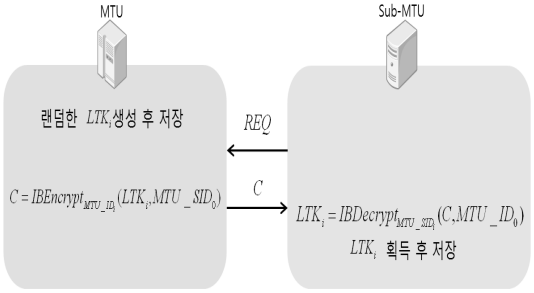
MTU와 Sub-MTU 간의 LTK 분배 프로토콜은 〔그림 5〕와 같이 나타낼 수 있으며, 다음과 같이 수행된다. MTU는 자신의 하부에 위치한 각각의 Sub-MTU들과 LTK를 공유하기 위하여 랜덤한 LTK_i를 생성한 후에 안전하게 저장한다. 그 후 Sub-MTU는 LTK_i 요청 메시지인 REQ를 MTU에게 전송하게 된다. MTU는 Sub-MTU의 ID인 MTU_ID_i를 이용하여 식 (1)을 수행함으로써 LTK_i를 암호화한다. 이 때, MTU는 자신의 개인 키인 MTU_SID_0을 포함시켜 인증 기능을 지닌 B-F 암호 알고리듬으로 암호화하게 된다. 따라서 수신자가 발신처를 인증함으로써 다른 장치가 자신에게 임의의 메시지를 전송하는 것을 방지하게 된다. MTU의 개인키인 MTU_SID_0는 실제로 전송되는 것이 아니라 인증 기능이 있는 ID 기반 암호화의 특징으로 인해 암호화 과정에서 사용될 뿐이다.

$$C = IBEncrypt_{MTU\_ID_i}(LTK_i, MTU\_SID_0) \quad (1)$$

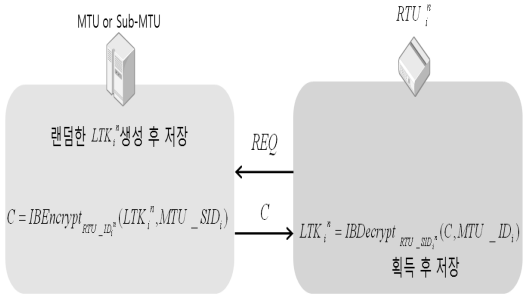
암호문 C를 수신한 Sub-MTU는 자신의 개인 키

〔표 1〕 표기법

표기법	내 용
MTU_i	i번에 위치한 MTU 및 Sub-MTU (MTU_0 : MTU, MTU_i, $1 \le i \le m$: Sub-MTU)
RTU_i^n	MTU_i 하위의 n번째 위치한 RTU
KMS_ID	KMS의 ID(공개 키)
KMS_SID	KMS의 개인 키
MTU_ID_i	i에 위치한 MTU 및 Sub-MTU의 ID(공개 키) (MTU_ID_0 : MTU의 ID, MTU_ID_i, $1 \le i \le m$: Sub-MTU의 ID)
MTU_SID_i	i에 위치한 MTU 및 Sub-MTU의 개인 키 (MTU_SID_0 : MTU의 개인 키, MTU_SID_i, $1 \le i \le m$: Sub-MTU의 개인 키)
$RTU\_ID_i^n$	MTU_i 하위의 n번째 위치한 RTU의 ID(공개 키)
$RTU\_SID_i^n$	MTU_i 하위의 n번째 위치한 RTU의 개인 키
LTK_i	MTU와 Sub-MTU$_i$ 간의 LTK
LTK_i^n	MTU_i와 RTU_i^n 간의 LTK
EK_i	KMS와 MTU 또는 Sub-MTU$_i$ 간의 복구 키
EK_i^n	KMS와 RTU_i^n 간의 복구 키



〔그림 5〕 MTU와 Sub-MTU 간의 LTK 분배



〔그림 6〕 MTU 또는 Sub-MTU와 RTU 간의 LTK 분배

인 MTU_SID_i와 MTU의 ID인 MTU_ID_0를 이용하여 식 (2)를 수행함으로써 암호문 C를 복호화하여 LTK_i를 얻은 후 안전한 곳으로 저장한다. Sub-MTU는 정당한 MTU의 ID를 복호화 시 이용하기 때문에 암호문 C가 정당한 MTU로부터 전송되었다는 것을 검증한다.

$$LTK_i = IBDecrypt_{MTU\_SID_i}(C, MTU\_ID_0) \quad (2)$$

4.2.2 MTU(혹은 Sub-MTU)와 RTU 간 LTK 분배

MTU(혹은 Sub-MTU)와 RTU 간의 LTK 분배 프로토콜은 〔그림 6〕과 같이 나타낼 수 있으며, 다음과 같이 수행된다. MTU(혹은 Sub-MTU)는 자신의 하부에 위치한 각각의 RTU들과 LTK를 공유하기 위해 랜덤한 LTK_i^n을 생성한 후에 안전하게 저장한다. 그 후 RTU는 LTK_i^n 요청 메시지인 REQ를 MTU(혹은 Sub-MTU)에게 전송하고 MTU(혹은 Sub-MTU)는 RTU의 ID인 $RTU_ID_i^n$을 이용하여 식 (3)을 수행함으로써 LTK_i^n를 암호화한다. 이 때, MTU(혹은 Sub-MTU)는 자신의 유일한 ID에 대한 개인 키인 MTU_SID_i을 포함시켜 인증 기능을 지닌 B-F 암호 알고리듬으로 암호화하게 된다.

$$C = IBEncrypt_{RTU\_ID_i^n}(LTK_i^n, MTU\_SID_i) \quad (3)$$

암호문 C를 수신한 RTU는 자신의 개인 키인 $RTU_SID_i^n$과 MTU 또는 Sub-MTU의 ID인 MTU_ID_i를 이용하여 식 (4)를 수행함으로써 암호문 C를 복호화하여 LTK_i^n를 얻은 후 안전한 곳으로 저장한다. RTU는 자신의 상부에 위치한 MTU(혹은 Sub-MTU)의 ID를 복호화 시 이용하기 때문에 암호문 C가 정당한 MTU(혹은 Sub-MTU)로부터 전송되었다는 것을 검증한다.

$$LTK_i^n = IBDecrypt_{RTU\_SID_i^n}(C, MTU\_ID_i) \quad (4)$$

4.3 개인 키 및 시스템 복구 프로토콜

SCADA 시스템은 국가의 중요한 망을 관리하기 위해 사용되기 때문에 예기치 못한 사고나 악의적인 공격에 대하여 키가 노출되었을 때 복구할 수 있는 방법이 요구된다. ID에 기반한 암호 시스템이 PKI 기반구조에 비해 수월한 점은 바로 복구 프로토콜에서 찾을 수 있다. PKI 기반 암호 시스템의 경우 이러한 응급 상황이 발생하게 된다면 개인 키를 각 장치에서 생성해야 하고 인증 과정이 필수적이므로 이를 복구하기 위한 절차가 복잡하게 된다. 제안하는 키 복구 프로토콜은 2가지 경우를 가정하여 설명한다. 첫 번째는 MTU, Sub-MTU, RTU와 같이 SCADA 시스템을 이루는 각 장치들의 개인 키가 노출되었을 경우 각 장치의 개인 키를 복구할 수 있는 프로토콜이 있고, 두 번째는 KMS의 마스터 비밀 키가 노출되었을 경우 SCADA 시스템을 이루는 모든 장치들의 개인 키를 복구할 수 있는 프로토콜이 있다.

EK는 각 장치들이 처음 설치되자마자 KMS로부터 생성되어 분배된다. 먼저 KMS는 MTU, Sub-MTU, RTU들에 대한 EK들을 생성한 후에 안전하게 저장한다. 그 후 각 장치들은 EK 요청 메시지인 REQ를 KMS에게 전송하면 KMS는 각 장치들의 ID를 이용하여 식 (5)을 수행함으로써 EK를 암호화한다. 이 때, KMS는 자신의 유일한 ID에 대한 개인 키인 KMS_SID을 암호화 시 포함하여 다른 장치가 자신으로 가장하는 것을 방지한다.

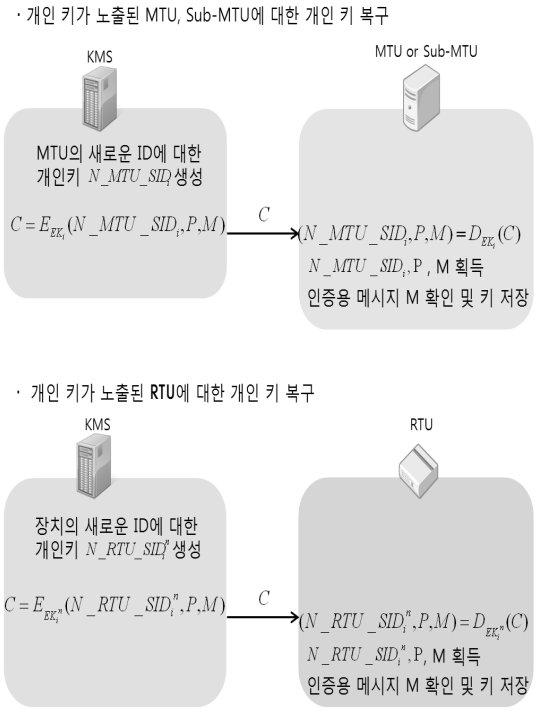
$$KMS-MTU:\\ C=IBEncrypt_{MTU\_ID_i}(EK_i, KMS\_SID)\\ KMS-RTU:\\ C=IBEncrypt_{RTU\_ID_i^n}(EK_i^n, KMS\_SID) \quad (5)$$

암호문 C를 수신한 각 장치들은 자신의 개인 키와 KMS의 ID인 KMS_ID를 이용하여 식 (6)을 수행함으로써 암호문 C를 복호화하여 EK를 얻은 후 안전한 곳으로 저장한다. 각 장치들은 KMS의 ID를 복호화 시 이용하기 때문에 암호문 C가 정당한 KMS로부터 전송되었다는 것을 검증한다.

$$KMS-MTU:\\ EK_i = IBDecrypt_{MTU\_SID_i}(C; KMS\_ID)\\ KMS-RTU:\\ EK_i^n = IBDecrypt_{RTU\_SID_i^n}(C; KMS\_ID) \quad (6)$$

4.3.1 각 장치의 개인 키 복구 프로토콜

예기치 못한 사고나 악의적인 공격들에 의하여 SCADA 시스템을 구성하는 각 장치들의 개인 키가 노출되었을 경우 KMS는 각 장치들과 미리 공유된 EK를 이용한 대칭 키 암호화 방식을 통해 ID 기반의 새로운 개인 키 및 공개 파라미터를 장치에 암호화하여 전송함으로써 개인 키를 복구한다. 〔그림 7〕은 각 장치의 개인 키 복구 과정을 나타낸다. 각 장치의 개인 키가 노출되면 KMS는 장치의 새로운 ID에 대한 개인 키를 생성한 후 EK를 이용하여 대칭 키 암호를 수행한 후 보내게 된다. 이때 각 장치들이 암호문의 송신자가 KMS라는 사실을 확인하기 위해서는 이미 알려진 메시지 M을 인증용으로 사용한다. 즉, 각 장치들은 응급 복구 메시지와 암호문이 오면 이를 EK로 복호한 후 인증용 메시지 M이 있는지 확인함으로써 이 메시지를 보낼 수 있는 유일한 송신자가 KMS임을 확인하고 자신의 개인 키를 복구한다. 여기서 KMS가 사용하는 새로운 ID는 고유 ID에 유효기간 등 알려진 정보를 갱신한 것이다. 즉, 새로운 ID=(고유한 ID || 2015.12)와 같이 갱신되는데 이때 각 장치와 연결된 다른 장치들은 새로운 ID정보로 갱신을 시켜주어야 한다.



〔그림 7〕 각 장치들의 개인 키 복구 과정

4.3.2 전체 시스템 키 복구 프로토콜

페어링을 이용한 ID 기반 암호화 시스템의 특징으로 인하여 SCADA 시스템을 구성하는 각 창치들의 개인 키를 발급하는 KMS의 마스터 비밀 키가 공격자로 인하여 노출된다면 공격자는 모든 장치들의 개인 키를 복구할 수 있다. 따라서 KMS의 마스터 비밀 키가 노출되었을 경우 KMS는 공개 시스템 파라미터와 미스터 비밀 키를 변경한 후에 각 장치들의 ID에 대응하는 개인 키를 다시 생성하게 된다. 그 다음 KMS는 각 장치들과 미리 공유된 EK들을 이용하여 새로운 개인 키 및 공개 파라미터를 암호화하여 모든 장치에 전송한다. 이 과정은 〔그림 7〕과 동일한 절차를 통해 초기화 된다. 모든 개인 키들이 분배된 후에는 각 MTU와 Sub-MTU는 자신의 하부에 위치한 RTU에게 LTK를 다시 분배한다. 공개 키 인증서를 사용하는 PKI 기반 시스템에 비하여 이 복구 시스템의 장점은 각 장치가 비밀 키와 공개 키를 생성하거나 공개 키 인증을 필요로 하지 않으므로 노출된 원격 장비들을 다시 인증할 필요없이 해당 공개 키에 따른 개인 키를 다시 발급할 수 있다는 것이다.

V. 제안 프로토콜의 비교 분석 및 구현 결과

5.1 키 분배 및 관리 방법 비교

기존의 키 분배 및 관리 방안과 비교하여 논문에서

〔표 2〕 키 분배 및 관리 방안 비교

구 분	SKE〔1〕	SKMA〔2〕	제안 방식
RTU간 통신	불허	허용	불허
키 관리 방식	통신 방식에 따라 대칭 키/PKI 공개 키 방식	대칭 키 방식	대칭 키/ID 기반 공개 키 방식 혼용
LTK 갱신	수동 갱신	KDC를 통하여 갱신	ID기반 공개 키 방식으로 자동 갱신
세션 키 설정	LTK 이용	LTK 이용	LTK이용
키 복구 방식	없음	없음	KMS에서 복구
장단점	통신 방식의 이원화 (C-S 통신, P2P 통신)	LTK 갱신시 KDC에 많은 통신 부하	알고리듬 구현이 다소 복잡

제안하는 방법의 가장 큰 차이는 ID에 기반한 공개 키 암호 시스템과 대칭 키 암호 시스템의 혼용이다. RTU에서 실제 데이터를 암호화하는 경우에는 고속의 대칭 키 암호방식을 이용한다. 또한 이 암호를 위한 세션 키 분배도 통신 시마다 매우 빈번하게 일어나야 하므로 대칭 키 방식을 사용하는 것이 효과적이다. 그러나 LTK와 같은 경우는 갱신 주기가 매우 길어 공개 키 방식을 사용하는 것이 효과적이다. 이 경우에는 키 갱신을 위한 인증이 필요하므로 인증 기능능 가진 공개 키 암호 알고리듬을 사용하였다. 제안 방식에서는 ID에 기반한 암호 알고리듬을 사용하므로 공개 키 인증 과정이나 확인 그리고 폐기과정이 필요 없게 된다. 특히, 장치의 개인 키 등을 복구하거나 갱신할 경우에도 ID에 따른 개인 키를 다시 만들어 낼 수 있어 별도의 인증 과정없이 바로 복구할 수 있는 장점이 있다. 제안하는 키 분배 및 관리를 비교 분석한 것이 〔표 2〕이다. 제안 기법은 대칭 키 암호 방식과 ID 기반 암호 방식만 정상적으로 구현할 수 있다면 시스템에 바로 적용이 가능하다. 논문에서는 제안 방식을 컴퓨터 상에서 실제 구현하여 그 타당성을 검증하였다.

5.2 구현 성능 분석

본 논문에서 제안하는 프로토콜의 성능 및 을 검증하기 위하여 3장에서 소개된 인증 기능이 있는 B-F ID 기반 암호화 시스템을 구현하여 속도를 측정하였다. 성능 테스트 환경은 다음과 같다.

- CPU : Intel Duo CPU 3GHz
- RAM : 3GB
- 운영체제 : Linux(Fedora 12)
- 개발 언어 : C언어(gcc 컴파일러)
- 라이브러리 : OpenSSL-1.0.0d〔11〕

〔표 3〕 성능 분석 결과

구 분	암호화 속도	복호화 속도
위수 q : 160비트 유한체 p : 192비트	22.94 ms	22.01 ms
위수 q : 160비트 유한체 p : 256비트	31.98 ms	31.42 ms
위수 q : 160비트 유한체 p : 384비트	64.39 ms	63.75 ms
위수 q : 160비트 유한체 p : 512비트	110.23 ms	109.18 ms

테스트는 암호화와 복호화로 나누어서 수행하였으며 위수의 크기는 160 비트로 고정하였고 모듈러 N의 길이는 192, 256, 384, 512 비트로 나누었다. 각 테스트 당 100번을 수행하여 평균값을 산출하였다. 구현 방법은 표준 문서 RFC 5091 Identity-Based Cryptography Standard (IBCS) #1〔12〕을 참고하였으며, 〔표 3〕은 구현 속도를 나타낸 것이다. 실제로 MTU나 RTU 장비에 장착하여 구현하면 표의 시간보다 많이 걸릴 것이 예상되지만 LTK 속성상 갱신 기간이 길므로 전체적인 SCADA 시스템 운영에는 별 어려움이 없을 것으로 예상된다.

VI. 결 론

국가의 주요 시설의 감시나 제어를 위한 SCADA 시스템은 특성상 강력한 관리 기능이 있는 폐쇄적인 그룹에서 많이 운영된다. 본 논문에서는 이러한 점을 고려하여 SCADA 시스템에서 운영할 수 있는 안전한 키 분배 및 관리 방법을 제안하였다. 각 장치에서 사용하는 세션 키를 만드는 LTK를 ID 기반 암호 방식을 통해 분배하도록 설계하였다. ID 기반 암호화 시스템을 이용할 경우 ID 자체가 공개 키인 점을 고려

하여 공개 키 인증서 검증 및 폐기와 관련된 복잡한 절차를 생략할 수 있다. 또한 SCADA 시스템의 일부 혹은 전체적인 비밀 키 노출 시에도 각 장치에 대한 별도의 인증 없이 쉽게 재복구할 수 있는 기능을 가진 복구 방법도 제안하였다. 다만, 긴급 상황 시 사용할 수 있는 EK는 물리적으로 안전한 곳에 저장이 되어야 하며 절대 노출이 되지 않도록 해야 한다.

참고문헌

〔1〕 B. Cheryl, G. Donald, N. William and T. Mark, "Key management for SCADA," Sandia National Laboratory, Mar. 2002.

〔2〕 D. Robert, B. Colin, D. Ed and G. N. Juan, "SKMA a key management architecture for SCADA systems," Australasian Information Security Workshop, vol. 54, pp. 138-192, 2006.

〔3〕 D. Choi, H. Kim, D. Won and S. Kim, "Advanced key management architecture for secure SCADA communications," IEEE Trans. Power Delivery, vol. 24, no. 3, pp. 1154-1163, July. 2009.

〔4〕 D. Choi, S. Lee, D. Won and S. Kim, "Efficient Secure Group Communications for SCADA," IEEE Trans. Power Delivery, vol. 24, pp. 714-722, 2010.

〔5〕 E. Okamoto and T. Okamoto, "Cryptosystems Based on Elliptic Curve Pairing," MDAI 2005, LNAI 3558, pp. 13-23, 2005.

〔6〕 G. M. Bertoni, L. Chen, P. Fragneto, K. A. Harrison and G. Pelosi, "Computing tate pairing on smartcards," available at http:// www.st.com/stonline/products/families/smartcard/ches2005_v4.pdf, 2005.

〔7〕 A. Shamir, "Identity-based crypto- systems and signature schemes," CRYPTO 1984, LNCS 196, pp. 47-53, 1985.

〔8〕 S. Ohgishi, R. Sakai and M. Kasahara, "Cryptosystems Based on Pairing," In The 2000 Symposium on Cryptography and Information Security - SCIS 2000, 2000.

〔9〕 D. Boneh and M. Franklin, "Identity-Based encryption from the Weil pairing," SIAM Journal of Computing, vol. 32, no. 3, pp. 586-615, 2003.

〔10〕 B. Lynn, "Authenticated Identity- Based Encryption," available at http://eprint.iacr.org/2002/072, 2002

〔11〕 OpenSSL Project, "OpenSSL-1.0.0d," available at http://openssl.org/source/.

〔12〕 X. Boyen, L. Martin, "Identity-Based Cryptography Standard (IBCS) #1," available at http://www.ietf.org/rfc/rfc5091.txt, Request for Comments (RFC) 5091, 2007.

〈著者紹介〉

오 두 환 (Doo-Hwan Oh) 정회원
2010년 2월: 호서대학교 정보보호학과 졸업
2012년 2월: 호서대학교 정보보호학과 석사
2012년 3월~현재: ㈜ 윈스테크넷
〈관심분야〉 네트워크 보안, ID-기반 암호화 시스템, 오류주입 공격

최 두 식 (Doo-Sik Choi) 정회원
2010년 2월: 호서대학교 정보보호학과 졸업
2012년 2월: 호서대학교 정보보호학과 석사
2012년 6월~현재: ㈜ 소프트포럼
〈관심분야〉 네트워크 보안, 부채널 공격

나 은 성 (Eun-Sung Na) 정회원
2003년 2월: 한세대학교 컴퓨터공학과
2002년 10월~2009년 5월: (주)레드게이트 - 서버보안 개발팀장
2009년 06월~2011년 9월: (주)SGA - 개발팀장
2011년 10월~현재: (주)SGEN I&C - 개발팀장
〈관심분야〉 정보보호, 서버보안, 네트워크 보안, 임베디드보안

김 상 철 (Sang-Chul Kim) 정회원
1992년 2월: 경북대학교 전자공학과 졸업
1994년 2월: 경북대학교 전자공학과 석사
2002년 7월~2009년 5월: (주)레드게이트 연구소장
2009년 6월~현재: 에스지에이(주) 개발부문 총괄사장
〈관심분야〉 정보보호, 서버보안, 악성코드, 임베디드보안

하 재 철 (Jae-Cheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 졸업
1993년 2월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월~2007년 2월: 나사렛대학교 정보통신학과 부교수
2007년 3월~현재: 호서대학교 정보보호학과 부교수
〈관심분야〉 정보보호, 네트워크 보안, 부채널 공격