DOI: 10.3745/KIPSTC.2009.16-C.1.37

안전한 SCADA 통신을 위한 트리 기반의 효율적인 키 관리 구조

최 동 현[†] · 이 성 진[††] · 정 한 재[†] · 강 동 주[†††] · 김 학 만[††††] · 김 경 신[†††††] · 원 동 호[††††††] · 김 승 주[†††††††]

요 약

SCADA(Supervisory Control and Data Acquisition) 시스템은 국가 기반시설에서 주로 사용되는 제어 시스템이다. 과거 SCADA 시스템은 폐쇄 망에서 운영되어진다는 이유로 보안에 대한 고려 없이 설계되었다. 하지만 기술의 발달로 SCADA 시스템과 공용망과의 연계가 추진되면서 보안에 대한 문제점이 대두 되고 있다. 본 논문에서는 SCADA 시스템의 제약사항과 보안요구사항을 살펴보고, 안전한 SCADA 시스템을 위한 키 관리 구조를 제안한다. 기존에 제안되어있는 SCADA 시스템을 위한 키 관리 방식이 메시지 브로드캐스팅을 지원하지 못하는 반면, 제안하는 방식은 메시지 브로드캐스팅을 지원한다. 또한 제안하는 방식은 성능상의 제약을 가지고 있는 RTU의 계산량을 최소화하기 위해, 상위 노드(SUB-MTU 또는 MTU)에 계산량을 분배하여 RTU의 잠재적인 성능 병목을 해결하였다.

키워드 : SCADA, 키 관리, 보안, LKH

Advanced Key Management Architecture Based on Tree Structure for Secure SCADA Communications

Donghyun Choi[†] · Sungjin Lee[††] · Hanjae Jeong[†] · Dongjoo Kang[†††] · Hakman Kim[††††] · Kyungsin Kim[†††††] · Dongho Won[††††††] · Seungjoo Kim[†††††††]

ABSTRACT

The SCADA(Supervisory Control And Data Acquisition) system is a control system for infrastructure of nation. In the past, the SCADA system was designed without security function because of its closed operating environment. However, the security of the SCADA system has become an issue with connection to the open network caused by improved technology. In this paper we review the constraints and security requirements for SCADA system and propose advanced key management architecture for secure SCADA communications. The contributions of the present work are that our scheme support both message broadcasting and secure communications, while the existing key management schemes for SCADA system don't support message broadcasting. Moreover, by evenly spreading much of the total amount of computation across high power nodes (MTU or SUB-MTU), our protocol avoids any potential performance bottleneck of the system while keeping the burden on low power (RTU) nodes at minimal.

Keywords :SCADA, Key Management, Security, LKH

1. 서 론

국가기관들은 국가 기반시설이라고 할 수 있는 전기, 수도, 가스 수송망 등 국가 기능 수행에 필요한 기본적이고 필수적인 시스템의 관리를 위해 SCADA(Supervisory Control and Data Acquisition)라는 자동화 시스템을 사용하고 있다[1]. 과거에 SCADA 시스템은 폐쇄 망에서 운영되어진다는 이유로 보안에 대한 고려 없이 설계되었다. 하지만 기술의 발달로 SCADA 시스템과 공용망과의 연계가 추진되고 DNP3와 같은 표준화된 프로토콜의 사용이 증가되면서[2] SCADA 시스템의 보안에 대한 문제점이 대두되고 있다. 더군다나 SCADA 시스템의 경우 국가 기반시설을 관리하는 시스템이기 때문에 해당 시스템이 공격에 노출될 경우 그 피해의 규모가 사회 전반에 걸쳐 매우 크기 때문에[3] 많은 국가기관

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학IT연구센터 지원 사업의 연구결과로 수행되었음 (IITA-2009-(C1090-0902-0016))
† 준 회 원 : 성균관대학교 휴대폰학과 박사과정
†† 준 회 원 : 성균관대학교 휴대폰학과 석사과정
††† 정 회 원 : 한국전기연구원 연구원
†††† 정 회 원 : 인천시립대학 전기과 교수
††††† 정 회 원 : 인덕대학 인터넷TV방송과 교수
†††††† 종신회원 : 성균관대학교 정보통신공학부 교수
††††††† 종신회원 : 성균관대학교 정보통신공학부 교수(교신저자)
논문접수 : 2008년 5월 6일
수 정 일 : 1차 2008년 8월 14일, 2차 2008년 11월 1일, 3차 2008년 12월 22일
심사완료 : 2008년 12월 24일

에서 관심을 가지고 있다.

보다 안전한 SCADA 시스템을 위해 미국을 비롯한 여러 국가의 기관 및 연구소에서는 SCADA 시스템의 안전성 및 보안성 향상과 관련된 많은 연구를 진행하고 있다[12]. 그러한 결과로 SCADA 시스템 보안과 관련한 여러 표준문서와 보고서들이 발행되고 있다.

그 첫 번째로 ISA-SP99 위원회는 SCADA 시스템의 보안과 관련한 두 개의 기술 문서를 발행했다. 첫 번째 보고서인 ANSI/ISA-TR99.00.01-2007[14]에서는 안전한 SCADA 시스템을 위한 보안 기술에 대해 다루고 있다. 두 번째 보고서인 ANSI/ISA-TR99.00.01-2007[15] 보고서는 보안 구성요소의 통합에 관한 문제를 다루고 있다.

두 번째로 2004년 4월, NIST(National Institute of Standards and technology)는 산업 통제 시스템에 대한 시스템 보호프로파일(SPP)[16]을 발표했다. NIST 문서는 CC(Common Criteria)에 정의된 보호프로파일 양식을 따르고 있다. 시스템 보호프로파일은 기능 요구사항으로 TOE 무결성, 역할 기반 접근 제어, 데이터 인증 등과 보증 요구사항으로 생명주기지원, 설명서, 시험서, 취약성 평가 등을 기술하고 있다.

세 번째로 API(American Petroleum Institute)는 API-1164 Pipeline SCADA 보안 표준[17]을 2004년 9월에 공개하였다. 이 표준은 시스템의 무결성과 보안성을 위한 지침, 운영자 체크리스트, 보안 계획 템플릿에 관한 내용을 포함하고 있다.

네 번째로 2001년 9월 11일, AGA(American Gas Association)는 현재까지 두 개의 문서를 발표했다. 첫 번째 문서인 AGA-12 Part 1[10]은 SCADA 시스템을 보호하기위한 정책, 평가, 감사에 대해 다루고 있다. 또한 문서는 암호 시스템 요구사항과 보안 장치를 위한 테스트 계획을 기술하고 있으며, 추가적으로 보안장치가 NIST FIPS 140-2[19](암호 모듈에 대한 보안요구사항)에 부합할 것을 요구한다. 두 번째 문서인 AGA-12 Part 2[18]는 시리얼 통신과 암호화된 시리얼 통신 채널의 갱신에 관해서 기술하고 있다. 구체적으로는 대칭키를 이용한 인증기능을 제공하는 세션 프로토콜에 대해서 서술하고 있다.

마지막으로 2006년 10월, NIST는 SCADA와 산업 통제 시스템 보안을 위한 지침의 첫 번째 공식 초안을 발표 했다(NIST SP 800-82[20]). NIST 문서는 산업 통제 시스템의 위협 및 취약성에 대해 기술하고, 이러한 위협을 감소시킬 수 있는 보안 대응 방법들을 제안하고 있다.

본 논문에서는 이러한 표준화 문서 및 보고서를 바탕으로 SCADA 시스템에 필요한 보안요구사항을 도출하였다. 이렇게 도출된 보안요구사항들 중 본 논문에서는 통신채널을 안전하게 하는 기술에 초점을 두었다. 통신 채널을 안전하게 하는 기술은 SCADA 시스템을 안전하게 하는 중요 기술 중 하나이다[10]. 안전한 통신 채널을 만들기 위해서는 안전한 키 관리 기술이 필수적이다. 이러한 키 관리 기술에 관한 연구로 SANDIA에서 연구한 키 확립 프로토콜인 SKE[5]와 Queensland University of Technology에서 연구한 SKMA[6]가 있다. 기존의 SKE는 공개키 프로토콜을 사용하고 메시지를 브로드캐스팅 하지 못하는 단점이 있고, SKMA는 공개키 프로토콜을 사용하지는 않지만 메시지를 브로드캐스팅 하지 못하는 단점이 있다.

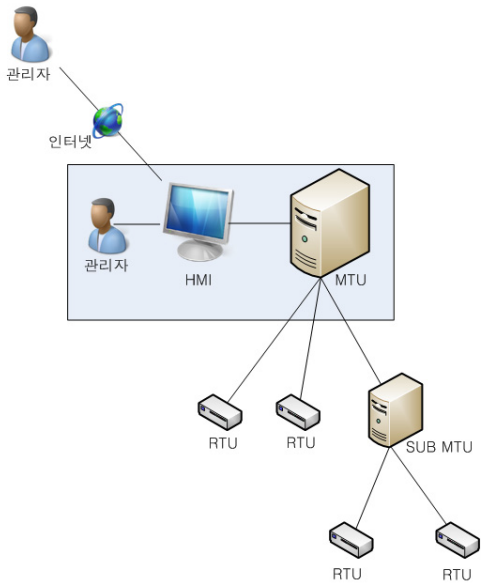
본 논문에서는 SCADA 시스템을 위한 키 관리 구조를 제안한다. 기존에 제안되어있는 SCADA 시스템을 위한 키 관리 방식이 메시지 브로드캐스팅을 지원하지 못하는 반면, 제안하는 방식은 메시지 브로드캐스팅을 지원한다. 또한 제안하는 방식은 성능상의 제약을 가지고 있는 RTU의 계산량을 최소화하기 위해, 상위 노드(SUB-MTU 또는 MTU)에 계산량을 분배하여 RTU의 잠재적인 성능 병목을 해결하였다.

본 논문의 구성은 다음과 같다. 2장에서는 SCADA 구조에 대해 설명하고, 3장에서는 SCADA 시스템의 보안요구사항에 대해 설명한다. 4장에서는 관련연구를 설명하며 5장에서는 본 논문에서 제안하는 키 관리 방식에 대해 설명한다. 6장에서는 기존의 방식과 비교한 후, 마지막으로 7장에서 결론을 맺는다.

2. SCADA 구조

SCADA 시스템은 원격장치의 상태 정보 데이터를 원격 단말 장치로 수집, 수신, 기록, 표시하여 중앙제어 시스템이 원격 장치를 감시하는 제어 시스템을 말한다. SCADA 시스템은 서로 통신하는 여러 장치로 이루어져 있다. 장치는 물리적인 환경에서 데이터를 수집하는 RTU(Remote Telemetry Unit)에서부터 운영자와 상호 작용을 하는 HMI(Human Machine Interface)까지 각각의 목적을 가지며 그에 맞게 설계되어 있다.

SCADA 시스템은 구성하고 있는 각각의 장치는 (그림 1)에 잘 나타나 있다. 상자 내부로 표시된 부분은 해당 시스



(그림 1) SCADA 시스템 구조

템이 물리적으로 안전하게 보호되어야 한다는 것을 말한다. SCADA시스템은 다수의 RTU 및 SUB-MTU로 구성된다. 시스템을 구성하는 각 장치에 대한 자세한 설명은 다음과 같다.

2.1 RTU(Remote Telemetry Unit)

마이크로프로세서로 구성된 RTU는 원격지에 설치되어 필요한 정보를 수집하고 이를 MTU로 전송하는 역할을 한다. RTU는 MTU의 제어 명령에 따라 다른 장치들을 제어하기도 한다. 이러한 RTU는 MTU와 RTU사이의 통신을 담당하는 컴포넌트, MTU로부터 오는 명령을 수행하는 컴포넌트, 데이터를 수집하는 센서로 구성된다.

대부분의 경우 RTU는 물리적으로 떨어진 원격지에 위치한다. 이러한 위치에 있는 RTU에 대해서 물리적인 안전을 보장하는 것은 힘들다. 예를 들어, 오수 펌프는 주거지역에 위치해야할 필요가 있는데 현실적으로 그렇게 광범위한 지역에 물리적인 보안을 실행하기는 어렵다.

2.2 MTU(Master Terminal Unit)

MTU는 RTU들을 감독하고 제어하는 기능을 하는 장치로 SCADA 시스템 구조상 최상단에 위치한다[9]. MTU는 RTU로부터 전송되어오는 정보들을 모아서 관리자에게 보여주며 관리자의 명령을 RTU에게 전달하여 행하도록 한다.

SCADA 시스템 구조는 일반적으로 하나의 MTU를 가지고 있으며, MTU 아래로 여러 개의 SUB-MTU나 RTU들이 포함되어 있다. MTU나 SUB-MTU의 경우 최소 현재 데스크톱 PC수준의 하드웨어 성능을 가진다. 이러한 장치는 일반적으로 SCADA 시스템에서만 사용가능한 하드웨어와 운영체제에서 동작한다.

2.3 HMI(Human Machine Interface)

HMI는 MTU와 동일한 위치에 존재한다. HMI는 운영자가 SCADA 시스템을 제어하기위해서 정보를 획득하고 어떤 제어 명령을 내릴 수 있는 장치이다. SCADA시스템을 위한 HMI는 PDA, 웹브라우저, 데스크톱 PC 등 넓은 범위에서 활용가능하게 개발되어 있다.

2.4 통신 채널

SCADA 시스템의 네트워크 토폴로지는 구조화되어 있다. 두 노드간의 가능한 통신 경로는 사전에 이미 알려져 있으며 노드들 간의 ad hoc 통신을 지원할 필요가 없다. 노드의 추가와 탈퇴는 관리되는 범위 안에서만 이루어지며 자유로운 가입 및 탈퇴는 지원하지 않는다. SCADA 시스템에서의 통신 경로에 대한 설명은 아래에서 좀 더 자세히 설명한다.

2.4.1 MTU-RTU 통신

MTU와 RTU사이에는 인터넷, 위성통신, 케이블, WiFi, 표준 모뎀/이더넷 등과 같은 다양한 통신 채널이 사용될 수 있다. 이러한 MTU-RTU 통신채널 상에는 공격자에 의해 악의적인 메시지의 삽입, 전송되는 메시지의 변경 또는 삭제와 같은 위협이 발생할 수 있다. 이에 대응하기 위해 암호 메커니즘과 같은 기술적 솔루션이 필요하다. 이러한 기술적 솔루션에는 암호 키를 관리하는 시스템도 포함된다.

2.4.2 RTU-RTU 통신

SCADA 시스템에서 RTU와 RTU사이의 통신은 가능하지만, 제한된 방법으로 발생된다. 이러한 통신에서 RTU중 하나는 다른 RTU를 제어할 수도 있고, 그 경우 RTU는 MTU와 같은 역할을 하게 된다. RTU-RTU간의 통신은 일반적으로 사전에 미리 계획된다. 보안 솔루션은 이러한 RTU들 간의 통신도 지원해야한다.

2.4.3 HMI-MTU 통신

HMI-MTU간의 통신은 일반적으로 TCP/IP 기본 프로토콜을 사용하며 클라이언트-서버 구조를 가진다. 이런 특성 때문에 HMI-MTU는 웹과 같은 여러 가지 서비스에 쉽게 적용 가능하다. 일반적으로 HMI와 MTU는 동일한 장소에 위치하기 때문에 통신 속도 등 여러 가지 측면에서 제약사항이 없다.

2.5 SCADA 시스템의 제약사항

<표 1>에서는 SCADA 시스템의 보안 구조를 설계할 때 고려해야할 제약사항들을 나열하였다. 이러한 제약사항들에 대해서 아래에서 설명한다.

SCADA 시스템의 구조는 일반적인 컴퓨터 네트워크들과

〈표 1〉 SCADA 시스템의 제약사항[6]

제약사항	설명
RTU의 자원 제약	RTU는 일반적으로 낮은 프로세스 처리능력과 메모리를 가지고 있다.
높은 가용성	SCADA 시스템은 어떤 문제가 발생하더라도 중단 없이 작동되도록 설계되어있다.
낮은 대역폭	대부분의 시스템의 경우 대역폭은 9600 baud로 제한된다.
긴 노드의 수명	노드들은 일반적으로 25년 정도의 수명을 가진다. 이는 일반적인 컴퓨터 하드웨어의 수명 보다 훨씬 길다
실시간	SCADA 시스템은 물리적인 처리과정을 제어하기 때문에 실시간 통신을 필요로 한다.
구조화된 네트워크	네트워크의 구조와 통신 채널이 이미 정의되어 있으며 ad hoc 통신은 필요로 하지 않는다.
단계적 적용	보안 통신의 단계적인 적용이 필요하다. 이는 기존의 장비 중 보안을 지원할 수 없는 하드웨어를 업그레이드하는데 수년이 소요되기 때문이다.
RTU는 물리적으로 불안전	RTU는 원격지에 설치되기 때문에 RTU에 항상 물리적 보안을 제공할 수는 없다.

같지 않다. 이러한 차이점이 기존의 컴퓨터 네트워크에 적용하는 여러 보안 기술을 그대로 적용하기 힘들게 한다. 다만 HMI-MTU 구조는 표준 서버-클라이언트 모델을 사용하기 때문에 이 부분에는 일반적인 보안 솔루션의 적용이 가능하다.

<표 1>에서 나열한 제약사항들은 특히 성능과 관계가 많다. 일반적으로 RTU의 경우 낮은 처리 능력을 가지고 있다. 또한, 다수의 통신 메커니즘이 낮은 대역폭을 가지고 있다. 게다가 SCADA 시스템에 의해서 제어되는 모든 장비나 프로세스는 실시간으로 감독되고 제어되어야 한다. 이러한 점은 시스템에 자원을 많이 소모하는 보안 프로토콜의 적용을 어렵게 한다. 또한 대부분의 SCADA시스템은 항상 켜져 있다. 이러한 점은 해당 시스템의 업데이트를 어렵게 한다.

RTU는 긴 생명주기를 가지고 있으며 지난 10여 년 동안 설계되어 왔다. RTU가 긴 생명주기를 가지고 있기 때문에 SCADA 시스템 내에는 여러 버전의 RTU들이 혼재되어 있다. 이러한 RTU 중 보안 프로토콜을 지원하지 못하는 장치가 존재하며 이러한 장치들을 모두 교체하는 데에는 많은 시간이 소요된다.

3. 보안요구사항

SCADA 시스템은 이기종의 유무선 네트워크와 다양한 프로토콜의 혼재로 기존 인터넷 등에서 발생되는 보안요구사항 외에도 추가적으로 고려해야할 보안요구사항들이 존재한다. 우리는 서론에서 간략히 언급한 ANSI/ISA-TR99.00.01-2007[14], ANSI/ISA-TR99.00.01-2007[15], 산업 통제 시스템에 대한 시스템 보호프로파일(SPP)[16], API-1164 Pipeline SCADA 보안 표준[17], AGA-12 Part 1[10], AGA-12 Part 2[18], NIST SP 800-82[20]을 바탕으로 보안요구 사항을 도출하였다. 이번 장에서는 SCADA 시스템에 필요한 보안요구사항에 대해 서술한다.

3.1. 기밀성

SCADA 시스템을 안전하게 하기 위해서는 MTU와 관리자, MTU와 RTU, RTU와 RTU사이에 안전한 데이터 통신이 가능해야한다. 이를 위해서는 각 노드 간에 전송되는 데이터의 암호화가 필요하다. 전화 또는 네트워크 회선을 통한 암호화되지 않은 데이터 통신은 전화도청이나 네트워크 스니핑의 여지가 있다. 이를 해결하기 위해 데이터는 암호화되어 전송되어야 한다. 이러한 데이터 암호화에는 키관리가 필수적이다. SCADA 시스템에서 키 관리설계는 다음과 같은 요구사항을 지원해야한다.

3.1.1 브로드 캐스팅

대부분의 SCADA 시스템은 메시지 브로드캐스팅 기능을 사용한다. 이러한 브로드캐스팅을 통해서 “긴급 정지”와 같은 중요한 메시지가 전송될 수 있기 때문에 이를 안전하게 전송하는 것은 매우 중요하다. 모든 RTU는 MTU로부터 전송받은 메시지를 복호화할 수 있어야 하며 이를 위해서는 그 메시지를 복호화할 수 있는 키를 가지고 있어야한다. 그러므로 키 관리 시스템을 설계할 때에는 브로드캐스팅을 지원하도록 하는 것이 매우 중요하다[10].

3.1.2 Key freshness

대부분의 SCADA 통신은 반복적인 특성을 가진다. 즉 일반적인 상황에서 SCADA 시스템이 요청한 상태정보의 결과는 동일하거나 매우 유사한 데이터이다. 그렇기 때문에 공격자는 전송되는 정보의 트래킹을 통해 부분적인 정보를 취득할 수 있다. 이를 막기 위해서는 각 세션마다 다른 세션키를 사용해야 한다[10].

3.1.3 프로토콜의 계산량 최소화

SCADA 시스템에서 전송되는 데이터는 실시간으로 송수신되어야한다. 안전성을 높이기 위해 데이터를 암호화 하더라도 그에 따른 데이터 전송의 지연은 최소화 되어야한다[10]. 하지만, 앞서 살펴본 바와 같이 SCADA 시스템에서 RTU들은 일반적으로 낮은 프로세스 처리 능력을 가진다. 그러므로 높은 프로세스 처리 능력을 요구하는 공개키 알고리즘의 사용을 최대한 배제해야한다. 암호화된 데이터의 전송을 위해 공개키 암호 알고리즘을 사용할 수 있지만 자원 소모가 많고 너무 느리다는 것이 문제가 되므로 가능한 공개키 암호 알고리즘의 사용을 배제해야 한다.

3.2 무결성

SCADA 시스템에서 전송되는 데이터는 무결성을 보장해야한다. 만일 전송되는 데이터가 악의적인 공격자에 의해서 변경 된다면 장비의 오작동을 유발할 수 있고 이는 많은 보안 사고로 연결될 수 있다. 메시지의 무결성을 유지하기 위해서는 암호학적인 메커니즘을 사용해야한다.

3.3 가용성

SCADA 시스템에서는 일반적인 경우와 다르게 가용성이 기밀성 보다 더 중요하다. SCADA 시스템은 지속적인 서비스를 제공해야하며 만일 어떤 공격에 의해서 작동이 멈추게 된다면 그 즉시 피해가 발생하며, 다시 원상태로 복구하기위해서는 많은 비용과 시간이 소요된다. 그러므로 암호화와 같은 보안 대책은 가용성이 허용되는 범위 내에서만 가능하다.

3.4 접근 제어

SCADA 시스템은 인증되지 않은 사용자의 접근을 차단해야한다. 이를위해 SCADA 시스템은 다 계층 접근 제어, 장치 접근 제어, 물리적 접근 제어를 지원해야 한다.

3.5 네트워크 보안

SCADA에서 사용되는 네트워크 토폴로지로 성형 네트워크 토폴로지가 권고된다. 성형 네트워크 토폴로지는 점대점 회선을 사용하며 모뎀연결이나 공중 네트워크 게이트웨이를

사용하지 않아 가장 안전하고 신뢰된다. 또한 SCADA 시스템의 경우 외부 네트워크와의 직접적인 연결을 피해야한다. 이를 위해 SCADA 시스템은 외부 네트워크와 내부 네트워크 사이에 DMZ를 가져야 한다. 또한 SCADA 시스템은 불필요한 접근을 차단하고 비정상 트래픽을 참지하기위해 침입차단시스템과 침입탐지시스템을 사용해야 한다.

3.6 보안정책

SCADA 시스템의 관리자는 패스워드 보안 정책, 보안 계획, 위협 분석, 복구 계획, 감사 정책 등과 같은 보안 정책을 개발하고 수행해야한다.

4. 관련 연구

본장에서는 SCADA 시스템에서 사용하는 키 관리 기술, LKH 프로토콜, Iolus Framework에 대해서 살펴본다.

4.1 SKE(Sandia Key Management)

SKE[5]는 SANDIA에서 연구한 키 관리 프로토콜이다. SKE에서 사용되는 통신 방식은 (그림 2)에서 보는 것처럼 SUB-MTU와 RTU 또는 MTU와 RTU사이의 통신에서 사용되는 C-S(Controller to Subordinate)와 SUB-MTU사이의 통신에 사용되는 P2P 통신으로 구분한다. SKE에서 제안된 주요한 통신 전략은 C-S통신 즉 제어장치와 종속장치간의 통신이다. 만약 SUB-MTU와 RTU사이의 통신이라면 여기서 SUB-MTU가 제어장치가 되고 RTU는 종속장치가 된다. SKE에서 제어장치와 종속장치 사이에 사용되는 키는 다음과 같다.

- LTK(Long Term Key) : 각각의 제어장치와 종속장치 사이에 수동적으로 배포된 키로 GK를 생성하는데 사용된다.
- GSK(General Seed Key) : KDC에 의해서 생성된 랜덤한 비트열로 GK를 생성할 때 사용된다. 여기서 KDC(Key Distribution Center)는 MTU와 HMI와 같이 물리적으로 안전한 곳에 설치되며 SCADA 시스템에서 사용할 키를 관리하는 역할을 한다.
- GK(General Key) : 제어장치와 종속장치 사이에 공유된다. 이것은 제어장치에 의해서 생성되며 GSK와 LTK를 사용한다. 이것은 제어장치에서 생성되어 LTK로 암호화되어서 종속장치로 전송된다.
- SK(Session Key) : 실제로 전송하려고 하는 데이터를 암호화하는데 사용되는 키 이다.

여기서 KDC는 시스템에서 각각의 장치들이 사용하는 키를 관리하는 기능을 한다. 제어장치와 종속장치는 LTK를 공유하고 있다. 제어장치는 KDC로부터 받은 GSK와 자신이 생성한 랜덤한 비트 열 B_r을 해쉬 하여 GK를 생성한다. 이렇게 생성된 GK는 LTK로 암호화하여 종속장치로 전송한다.

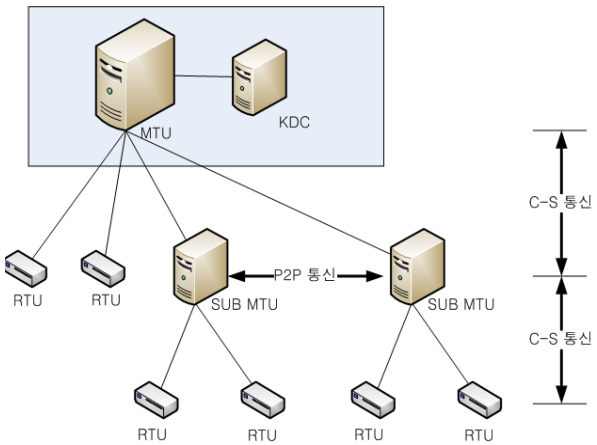
이렇게 해서 제어장치와 종속장치가 공유하고 있는 GK가 생성되고 각 세션마다 해당 GK로 세션키를 생성한다. 세션키 SK는 공유하고 있는 GK, FLAG, ID, 시간변수 TVP, 그리고 메시지길이 LEN을 해쉬 하여 생성한다. GK가 공격자에 의해서 공격 받게 된다면 제어장치가 다시 GK를 생성하여 업데이트 한다.

SUB-MTU간의 통신은 P2P통신으로 키 교환을 위해 공개키 암호 알고리즘을 사용한다. KDC는 각각의 SUB-MTU에게 공개키와 개인키 쌍을 할당한다. SUB-MTU는 이렇게 받은 키쌍을 이용하여 통신하고자하는 SUB-MTU와 키 교환 알고리즘을 이용하여 CK(Common Key)를 공유한다. CK는 C-S에서 GK와 같은 역할을 하게 된다.

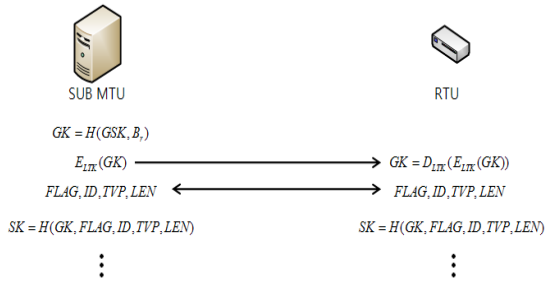
4.2 SKMA[6]

Queensland University of Technology에서 연구해서 발표한 SKMA는 SCADA 시스템을 위한 키 관리 방식이다. SKE가 공개키 암호 알고리즘을 사용하는 반면 SKMA는 대칭키 암호 알고리즘으로만 이루어져 있다는 장점이 있다. 여기서 노드는 RTU, SUB-MTU, MTU 모두 될 수 있다. SKMA에서 사용되어지는 키는 다음과 같다.

- Long term node-KDC key: 이 키는 node와 KDC사이에 공유되는 키로 통신을 위해 키를 설정할 때 사용된다.



(그림 2) SKE에서 구간에 따른 통신 방법



(그림 3) SKE에서 세션키 생성 과정

- Long term node-node key : 노드와 노드사이에 공유되는 키 이다.
- Session Key : 메시지를 암호화 하는데 사용되는 키 이다.
- Node-KDC Key : 이 키는 수동적으로 설치되며, node와 node사이의 키를 생성할 때 사용된다.

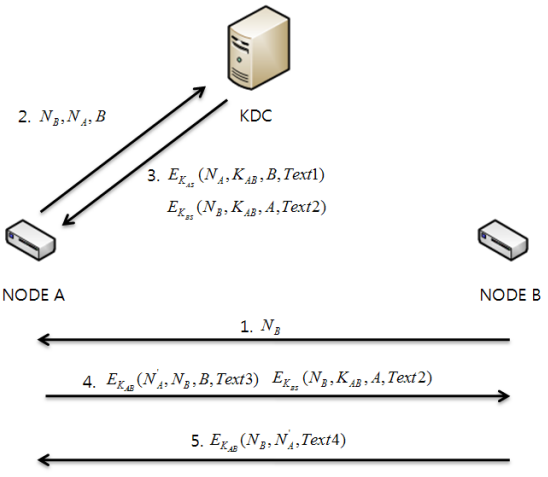
Node-KDC 키는 노드가 시스템에 배포되기 전에 설치된다. 새로운 노드가 추가될 때 node-node 키가 (그림 4)와 같은 과정을 통해서 교환된다.

위와 같이 SKMA에서는 서버와 노드 A, 노드 B간의 3자간 키 확립 프로토콜을 사용하는데 이는 ISO 11770-2[11] 메커니즘 9를 기본으로 한다. 통신에서 데이터를 암호화하는데 사용되는 세션키는 3자간 키 확립 프로토콜로 얻은 node-node 키와, 타임스탬프(세션의 유지기간에 기반을 둔)값의 해쉬하여 생성한다.

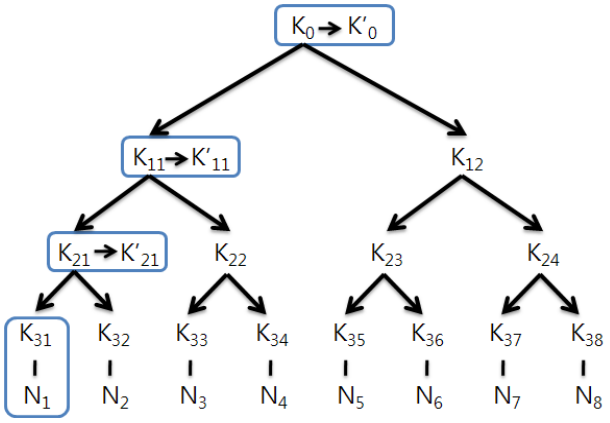
키 철회 메커니즘은 KDC에 의해서 수행되어진다. 키가 공격당했다는 사실을 자동적으로 알려주는 알고리즘은 현재 없다. 다만 시스템을 모니터링하다 이상행동을 발견하면 해당 노드와 관련된 키를 철회 한다. KDC 해당 노드의 키가 철회 되었다는 사실을 다른 모든 노드에게 알려야한다. 이때 사용되는 메시지는 각각의 노드와 KDC간의키로 암호화해서 전송되어진다.

4.3 LKH 프로토콜

그룹키 관리 구조에서는 멤버가 가입하거나 탈퇴할 때마다 키를 갱신해야한다. 효율적인 키 갱신을 위해 여러 가지 기법들이 제안되었다. 가장 널리 사용되는 방법 중 하나가 LKH[21] 이다. LKH는 멤버의 가입이나 탈퇴 시에 갱신되는 키의 수나 암·복호화 수를 키 트리의 높이인 O(log n)으로 줄인 것이다. 각각의 노드는 배포되기 전에 자신이 분배되어질 노드에서부터 root가 있는 곳까지의 경로 상에 있는 키를 저장한다. 만약 최초 배포단계가 끝나고 새로운 노드



(그림 4) SKMA 키 관리 방식



(그림 5) 새로운 노드의 가입

가 추가되게 된다면 키의 업데이트가 필요하다. 이때 키 업데이트는 추가되는 노드에서 root 까지의 경로에 있는 모든 노드의 키를 변경해야 한다. 예를 들어 (그림 5)에서처럼 N_1 노드가 추가되면 KDC는 기존의 키 K_0,K_{11},K_{21}을 업데이트하여 새로운 키 K'_0,K'_{11},K'_{21}을 생성한다. KDC는 식 (1)과 같이 키 업데이트 메시지를 생성하여 노드 $N_i(5 \le i \le 8)$에게 전송한다. 노드 $N_i(3 \le i \le 4)$에게는 식 (2)와 같이 키 업데이트 메시지를 생성하여 전송한다. 또한 노드 N_2에게는 식 (3)과 같이 키 업데이트 메시지를 생성하여 전송한다. 마지막으로 식(4)와 같은 메시지를 생성하여 새로운 노드 N_1 에게 전송한다. 이렇게 함으로써 각 노드는 새로운 노드 N_1이 가입하기 전에 사용하던 키를 모두 새로운 키로 업데이트 할 수 있다.

$$E_{K_0}(K_0') \quad (1)$$

$$E_{K_{11}}(K_{11}'), E_{K_0}(K_0') \quad (2)$$

$$E_{K_{32}}(K_{21}'), E_{K_{11}}(K_{11}'), E_{K_0}(K_0') \quad (3)$$

$$E_{K_{31}}(K_{21}', K_{11}', K_0') \quad (4)$$

탈퇴의 경우도 이와 유사한 방법으로 진행된다. 만약 노드 N_3이 탈퇴한다고 한다면 노드 N_3이 알고 있었던 K_0, K_{11}, K_{22} 키가 업데이트 되어야 한다. 이를 위해 KDC는 식 (5)과 같이 키 업데이트 메시지를 생성하여 노드 $N_i(5 \le i \le 8)$에게 전송한다. 노드 $N_i(1 \le i \le 2)$에게는 식 (6)와 같이 키 업데이트 메시지를 생성하여 전송한다. 마지막으로 노드 N_4에게는 식 (7)과 같이 키 업데이트 메시지를 생성하여 전송한다. 이를 통해 그룹에 속한 모든 멤버는 탈퇴한 노드 N_3이 모르게 키 업데이트를 완료하여 보안상의 문제점을 해결할 수 있다.

$$E_{K_{12}}(K_0') \quad (5)$$

$$E_{K_{21}}(K_{11}', K_0') \quad (6)$$

$$E_{K_{31}}(K_{22}', K_{11}', K_0') \quad (7)$$

LKH+[7][8] 프로토콜은 LKH프로토콜 중 새로운 노드가 가입하였을 때 업데이트 된 키를 암호화하여 전송하는 방식에서 기존의 키를 해쉬하는 방법으로 변경하였다. 이를 통해 기존의 업데이트 방식에서 필요한 추가적인 암복호화 및 메시지의 전송을 주려 보다 효율적으로 동작하게 하였다.

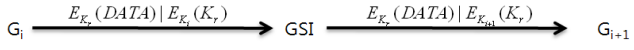
LKH++[13]의 경우는 기존의 프로토콜 중에서 노드가 탈퇴할 때 해당 노드의 키를 업데이트 하는 방법을 변경하여 보다 효율적으로 동작하게 한 방식이다. 이 방식은 탈퇴한 노드의 반대편 트리에서 탈퇴한 노드가 알지 못하는 최상위 키를 해쉬와 같은 알고리즘을 이용한 키 생성기를 사용하여 새로운 키를 생성해 내는 방식이다. 예를 들어 N_3이 탈퇴하면 $N_i(5\leq i\leq 8)$은 자신이 가지고 있는 키 K_0와 K_{12}를 이용하여 새로운 키 K'_0를 생성하여 키를 업데이트 하는 것이다. $N_i(1\leq i\leq 2)$의 경우는 자신이 가지고 있는 키 K_{21}과 K_{11}을 이용하여 새로운 키 K'_{11}을 생성하고, 키 K'_0의 경우는 KDC가 이를 K_{21}로 암호화하여 전송해준다. N_4의 경우는 루트노드가 업데이트 된 키 모두를 K_{34}키로 암호화여 전송해서 키를 업데이트 하는 방식이다. 이를 통해 LKH++는 탈퇴시에도 키 업데이트 방식에 해쉬 알고리즘을 사용하여 이전의 방식에서보다 효율적으로 키 업데이트가 가능하게 하였다.

4.4 Iolus Framework [4]

Iolus Framework는 하나의 큰 그룹을 여러 개의 서브 그룹으로 나누어서 관리하는 방식이다. 각각의 서브 그룹에는 최상위에 그룹 보안 에이전트(GSA)가 있어서 해당 그룹을 관리한다. 이 방식에서 멤버의 가입이나 탈퇴는 각각의 서브 그룹에서 개별적으로 이루어지게 된다.

본 방식에서 제일 중요한 역할을 하는 것은 그룹 보안 중계 장치(GSI) 이다. GSI는 GSA 사이에 위치하며 암호화된 데이터나 키를 중계하는 역할을 한다. GSI는 상위 서브 그룹에서 전송되어오는 데이터를 복호화 하고 하위 서브 그룹의 키로 해당 데이터를 다시 암호화여 하위 서브 그룹으로 전송한다. 여기서 전송되는 데이터의 재 암호화로 인한 오버헤드를 줄이기 위해 (그림 6)에서 보는 것처럼 랜덤한 키를 사용하게 된다. 데이터를 전송하고자하는 GSA G_i는 랜덤한 키 K_r을 생성하여 데이터를 암호화하고 생성한 랜덤한 키 K_r을 GSI와 공유하는 키 K_i로 암호화해서 GSI에게 전송한다. 이를 수신한 GSI는 랜덤한 키 K_r을 복호화 해서 얻어낸 후 이를 다시 데이터를 전송받는 GSA G_{i+1}와 공유하고 있는 키인 K_{i+1}키로 암호화하여 전송하게 된다.

이러한 방식은 큰 규모의 그룹을 작은 서브 그룹으로 나누어 보다 쉽게 관리가 가능하게 하는 장점을 가지고 있지만, 각각의 서브 그룹이 GSA라는 장치를 유지하여야 하기 때문에 비용증가의 문제가 발생하게 된다. 또한 GSA가 키를 번역해야하기 때문에 메시지 딜레이가 발생하게 된다. 이는 데이터가 실시간으로 전송되어야 할 경우 문제가 된다.



(그림 6) GSI의 키 중계

5. 제안하는 키 관리 방식

본 장에서는 LKH++[13] 프로토콜을 개선한 SCADA 시스템에 적합한 키 관리 방식을 제안한다. 대부분의 SCADA 시스템은 메시지 브로드캐스팅을 필요로 한다. 제안하는 키 관리 방식은 노드와 노드간의 안전한 통신은 물론, 메시지 브로드캐스팅에도 맞추어 설계하였다. 또한 성능제한을 가지고 RTU가 보관해야할 키의 개수를 고려하여 제안하였다. 이러한 키 관리 방식은 SCADA 시스템에서 사용하는 기존의 키 관리 방식 보다 효율적이고 안전한 키 관리를 가능하게 한다.

5.1 설계목적 및 용어정의

키 관리 방식의 설계 목적은 앞서 살펴본 SCADA 시스템의 제약사항 및 보안요구사항에 맞는 키 관리 방식을 설계 하는 것이다.

관련연구에서 살펴보았던 Iolus Framework의 경우는 암호화된 데이터의 전송을 할 때 중간단계에서 수차례의 중계가 필요하다는 단점을 가지고 있다. 이는 실시간으로 데이터를 수집하며 제어 명령을 내려야하는 SCADA 시스템에는 치명적인 단점이 될 수 있다. 또한 기존의 SKE와 SKMA 둘 모두는 메시지 브로드캐스팅을 지원하지 못하는 단점을 가지고 있다. 각각의 노드에 동일한 명령을 자주 전송하는 SCADA 시스템에서 이러한 조건은 큰 단점이 된다.

SCADA 시스템에서 RTU는 성능제한을 가지고 있다. 그렇기 때문에 공개키 암호 알고리즘과 같은 자원 소모가 많은 알고리즘의 사용을 RTU간의 통신에는 최대한 배제해야한다. 또한 악의적인 사용자로부터 공격이 발생하였을 때 해당 되는 노드를 제외한 키 업데이트가 가능하여야 한다. 이러한 키 업데이트 역시 자원소모가 많은 방법은 지양해야한다. 마지막으로 SCADA 시스템의 경우 메시지 브로드캐스팅이 필요하므로 이러한 기능을 제공해야 한다. <표 2>는 이러한 키 관리 요구사항을 나타내고 있다.

이 장에서 사용할 용어는 다음과 같다.

〈표 2〉 키 관리 요구사항

요구사항	설명
RTU-RTU간 통신	RTU와 RTU간의 안전한 통신이 가능해야함
메시지 브로드캐스팅	"긴급 정지"와 같은 중요한 메시지의 브로드캐스팅이 안전하게 이루어져야 함
키 갱신	각 세션마다 사용되어지는 키는 새롭게 생성되어야 함
실시간 통신	실시간 데이터 전송이 가능해야함
공개키 알고리즘 사용 배제	데이터 전송의 지연을 최소화하기 위해 공개키 알고리즘의 사용은 배제해야함
RTU가 저장하는 키의 수	효과적인 키 업데이트를 위해 RTU가 저장하는 키의 수가 적어야 함

- $E_k(D)$: 키 k로 대칭키 암호 알고리즘을 이용하여 입력값 D를 암호화
- $H(D)$: 입력값 D를 해쉬함
- $C_{i,j}$: 노드 'i'와 노드 'j'사이에서 사용되는 카운터 값
- C_i : 각 키 K_i에 대한 카운터 값
- $S_{i,j}$: i노드와 j노드 사이에 사용되는 세션키

KDC는 시스템에서 각각의 노드들이 사용하는 키를 관리하는 기능을 한다. KDC는 시스템 구조에 관한 정보를 알고 있으며, 키 설정 요구에 대해서 허락과 거부를 책임진다. 이러한 역할을 수행함으로써 키 분배를 용이하게 하고, 노드들 간의 신뢰된 관계를 설정 가능하게 한다. 게다가 키 철회 메시지는 KDC에 의해서 이루어진다.

KDC는 SCADA시스템의 MTU와 함께 위치한다. 이는 KDC와 MTU간의 메시지 교환 방법이 효과적임을 말한다. 또한 MTU와 KDC가 위치하는 곳은 일반적으로 RTU가 위치하는 원격지보다 물리적인 보안이 높다.

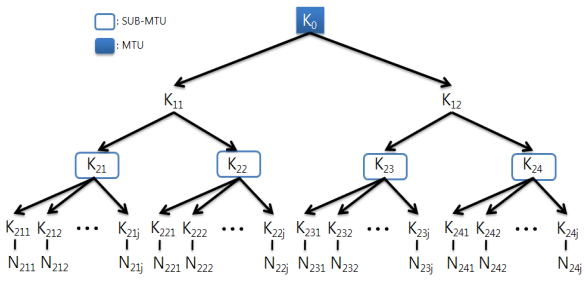
그럼에도 불구하고 보다 높은 보안을 위해서 가능하다면 KDC는 보안 하드웨어 장치에 개발되어야 한다.

5.2 키 관리 방법

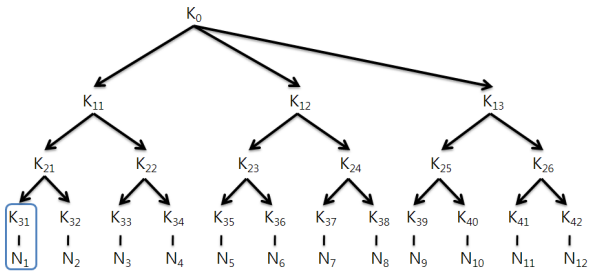
5.2.1 키 관리 구조 및 최초 분배

키 관리 구조는 이진트리와 n-ary 트리가 공존하는 방식으로 구성의 예는 (그림 7)과 같다. 이는 SUB-MTU의 수가 4개 이고 각 SUB-MTU에 속하는 RTU의 개수가 j개인 경우를 가정하였다. 최상위 노드에 MTU가 있고 그 밑으로 SUB-MTU가 이진트리의 마지막 리프노드에 위치한다. SUB-MTU의 구성을 이진트리의 마지막에 놓음으로써 특정 RTU의 탈퇴 시 키 업데이트가 효율적으로 진행할 수가 있게 된다. 만약 여기서 이진트리로 구성하지 않고 n-ary 트리로 구성하게 되면 (그림 8)에서 보는 것처럼 노드 N_1이 탈퇴한 후 키 업데이트를 할 때 N_5~N_8의 경우 키를 K_0과 K_{12}를 이용해서 업데이트하게 되는데 N_9~N_{12}의 경우는 K_{12}를 알고 있지 않기 때문에 이러한 방식의 업데이트가 불가능하다.

RTU는 SUB-MTU에 N-ary tree로 연결된다. 이렇게 구성할 경우 탈퇴하는 RTU가 있는 SUB-MTU의 경우 속해 있는 모든 RTU에게 키 업데이트 정보를 개별적으로 전송



(그림 7) 제안하는 키 관리 구조



(그림 8) 3-ary 트리에서의 키 업데이트

하여야 하는데 이는 일반 데스크톱 PC이상의 성능을 가지는 SUB-MTU가 감당할 수 있는 수준이다. 이처럼 SUB-MTU에 계산량을 보다 집중하여 RTU의 잠재적인 성능 병목을 개선하였다.

이렇게 키 관리 시스템을 구성할 경우 SUB-MTU에 속한 RTU가 저장하고 있어야하는 키의 수는 SUB-MTU가 총 n개, SUB-MTU당 RTU의 수가 최대 m개라고 가정할 때, $\log_2 n+2$개가 된다. 이는 RTU를 이진트리로 구성하는 경우 저장해야할 키의 개수 인 $\log_2 nm+1$보다 적어서 메모리 공간이 부족한 RTU에게 보다 효율적이다.

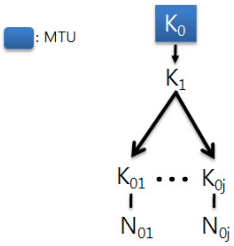
SCADA 시스템의 경우 MTU와 RTU가 직접적으로 연결이 되기도 한다. 이때 연결되는 RTU들은 (그림 9)처럼 키 K_1을 공유하게 된다. 이를 통해 K_0이 업데이트 될 경우 K_1을 이용해서 새로운 키 K'_0을 암호화해서 전송하도록 한다. 만약 키 K_1이 존재하지 않는다면 키 업데이트 때마다 MTU는 직접 연결된 각 RTU의 키로 업데이트된 키 K'_0를 전송해야하기 때문에 효율적이지 못하다.

SCADA 시스템의 경우 각각의 노드가 배포되기 전에 키 트리의 생성이 가능하다. 그러므로 LKH++에서처럼 배포 후 키 트리를 생성하는 것 보다는 사전에 키 트리를 생성하여 배포하는 것이 보다 효율적이다. 각각의 노드는 배포되기 전에 자신이 분배되어질 노드에서부터 최종 MTU가 있는 곳까지의 경로 상에 있는 키를 저장한다. (그림 7)의 노드 N_{211}의 경우는 해당 경로 상에 있는 키 K_{211}, K_{21}, K_{11}, K_0을 저장하게 된다. 또한 노드는 세션키 생성을 위한 카운터 값 C_i를 저장한다. 예를 든 노드 N_{211}의 경우 저장하고 있는 키에 해당되는 카운터 값 C_0, C_{11}, C_{21}, C_{211}을 저장한다. 이러한 카운터 값은 세션이 생성될 때마다 값이 하나씩 증가 된다.

이러한 카운터 값의 비동기화가 발생하면 각 노드는 카운터 값의 전후 값을 확인하여 카운터 값을 다시 동기화 한다. 만약 이러한 동기화 과정을 거친 후에도 카운터 값이 동기화 되지 않으면 KDC는 각 노드가 가지고 있는 키를 이용하여 카운터 값을 새롭게 생성하여 분배한다.

5.2.2 데이터의 암호화

최초 키 분배가 끝나고 각각의 노드는 발전소나 변전소와 같은 원격지에 설치된다. 설치가 끝나고 root 노드인 MTU와 해당 노드 N_{211}이 통신을 한다면 다음과 같이 세션키를 생성해서 데이터를 암호화하게 된다.



(그림 9) MTU와 직접 연결된 RTU 키 구조

$$S_{0,211} = H(K_{211}, C_{211}) \quad (1)$$

우선 식(1)에서처럼 노드 N_1의 키인 K_{211}과 카운터 값인 C_{211}을 해쉬하여 세션 값을 생성한다. 이러한 과정은 root 노드인 MTU와 노드 N_{211}에서 동일하게 진행되어 동일한 세션키를 생성하게 된다. 이렇게 세션키가 생성되면 해당 세션에서 오고가는 데이터는 식(2)에서처럼 세션키 $S_{0,211}$에 의해서 암호화 되어 전송되어진다. 세션이 종료되면 root 노드인 MTU와 노드 N_{211}은 식(3)에서처럼 가지고 있는 카운터 값을 1증가시킨다.

$$E_{S_{0,211}}(D) \quad (2)$$

$$C_{211} = C_{211} + 1 \quad (3)$$

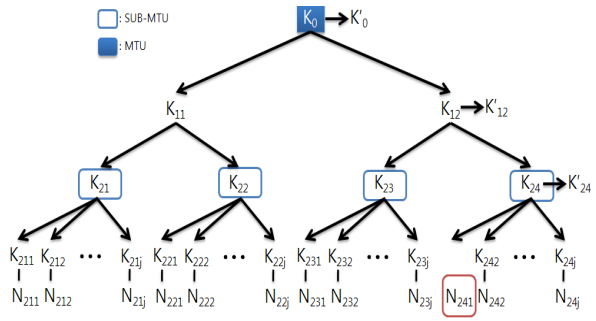
root 노드인 MTU가 어떤 정보를 각 노드에게 브로드캐스팅하려고 한다면, root 노드는 키 K_0과 카운터 값 C_0을 이용하여 브로드캐스팅에 사용할 키를 다음과 같이 생성한다.

$$S_{b,0} = H(K_0, C_0) \quad (4)$$

이렇게 생성된 키로 해당 되는 데이터를 암호화여 전송한다. 이를 수신한 각각의 노드는 키 K_0과 카운터 값C_0을 이용하여 동일한 세션키를 생성하고 데이터를 복호화 한다. 세션이 종료되면 루트 노드와 각 노드는 C_0값을 하나 증가시킨다.

노드와 노드 간의 통신은 해당 노드 간에 공유되는 키를 이용해서 암호화 하게 된다. 만약 N_{211}와 N_{221}노드가 통신을 한다고 가장하자. 노드 N_{211}와 N_{221}이 처음으로 통신을 한다면 카운터 값 $C_{211,221}$을 KDC로부터 전송 받는다. 이때 카운터 값 $C_{211,221}$은 수신자의 키로 암호화 되어 전송된다. 각 노드는 키 K_{11}를 수신한 카운터 값과 해쉬하여 세션키를 생성한다. 이렇게 생성된 키로 해당 되는 데이터를 암호화여 전송한다. 해당 세션이 종료되면 노드 N_{211}와 N_{221}은 카운터 값 $C_{211,221}$을 하나 증가 시키고 저장한다. 만약 노드 N_{211}와 N_{221}이 처음 통신하는 것이 아니라면 저장하고 있는 카운터 값 $C_{211,221}$을 이용하여 세션키를 생성한다.

$$S_{211,221} = H(K_{11}, C_{211,221}) \quad (5)$$



(그림 10) 기존 노드의 탈퇴

일반적인 경우 root 노드와 각 노드들 사이의 통신이 일반적이며, 각각의 리프 노드 대 노드간의 통신은 빈번히 발생하지 않는다.

5.2.3 새로운 노드의 가입

만약 최초 배포단계가 끝나고 새로운 노드가 추가 된다면 키의 업데이트가 필요하다. 추가되는 노드에서 root 까지의 경로에 있는 모든 노드의 키를 변경해야 한다. LKH++의 경우 KDC에서 의사난수를 생성하여 해당 난수를 각 노드에 브로드캐스팅한 후 난수와 업데이트해야 할 키를 XOR한다. 이는 키 업데이트를 위해 추가적인 메시지 브로드캐스팅이 필요하다. 통신량을 최소화하기 위해 우리는 기존의 키를 해쉬하는 방식으로 키 업데이트를 진행하였다. 예를 들어 N_{211}노드가 추가되면 우선 식(6)과 같이 기존의 키 K_0,K_{11},K_{21}을 해쉬하여 새로운 키 K'_0,K'_{11},K'_{21}을 생성한다. MTU는 N_{211}노드의 키인 K_{211}을 이용해서 식(7)처럼 키를 암호화해서 N_{211}에게 전송해준다. 이렇게 함으로써 N_{211}노드는 자신의 경로 상에 있는 모든 키를 저장 할 수 있다.

$$K'_{21} = H(K_{21}), K'_{11} = H(K_{11}), K'_0 = H(K_0) \quad (6)$$

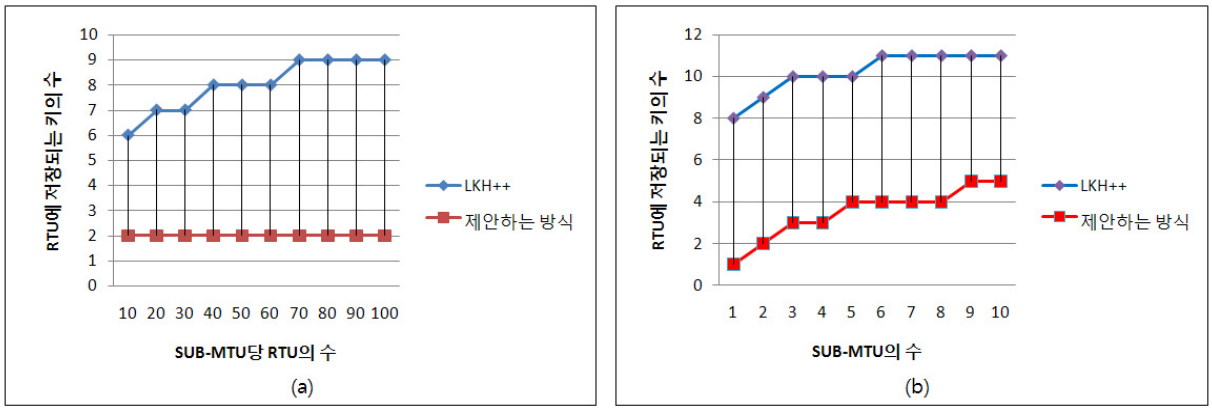
$$E_{K_{211}}(K'_{21}), E_{K_{211}}(K'_{11}), E_{K_{211}}(K'_0) \quad (7)$$

노드 N_{212}~N_{21j}의 경우는 자신이 가지고 있던 키 K_{21}, K_{11}, K_0을 해쉬하여 새로운 키 K'_{21}, K'_{11}, K'_0을 생성한다. 노드 N_{221}~N_{22j}의 경우는 키 K_{11}, K_0을 해쉬하여 새로운 키 K'_{11},K'_0을 생성하고, N_{231}~N_{23j}와N_{241}~N_{24j} 경우는 키 K_0을 해쉬하여 새로운 K'_0을 생성해서 키 업데이트를 완료한다.

5.2.4 기존 노드의 탈퇴

기존에 존재하던 노드가 탈퇴하거나 악의적인 공격자에 의해서 공격을 받았다고 판단되는 경우 해당되는 노드의 키를 업데이트 해야만 한다. KDC는 제거되는 사용자 노드에서부터 root 노드까지의 모든 경로에 있는 키를 변경해야 한다.

그림에서처럼 만약 노드 N_{241}이 탈퇴한다고 한다면 K_0, K_{12}, K_{24} 키가 업데이트 되어야 한다. 이를 위해 우선 K_0의 경우 다음과 같이 업데이트한다.



(그림 11) (a) SUB-MTU당 RTU의 수 에 따른 RTU에 저장되는 키의 수, (b) SUB-MTU의 수 에 따른 RTU에 저장되는 키의 수

$$K_0^{'} = H(K_0, K_{11}) \quad (9)$$

이때 기존의 키 K_0을 같이 해쉬하는 이유는 앞서 살펴본 새로운 노드가 가입될 때 키 업데이트를 기존의 키를 해쉬하여 진행하기 때문이다. 만약 K_0의 키를 K_{11}을 해쉬하여 K'_0을 생성할 경우 그 후에 노드가 새로 가입하게 되어 키를 업데이트 하게 될 때 K_{11}의 키를 해쉬하여 생성하게 되는데 이때 생성되는 키 K'_{11}이 K'_0과 동일하게 되기 때문이다.

위와 같이 키를 업데이트 할 경우 루트 노드를 기준으로 왼쪽의 경우 추가적인 통신 없이 자신이 가지고 있는 값의 해쉬를 통해서 키 업데이트가 가능하다는 장점이 있다. K_{12}의 경우 다음과 같이 업데이트한다.

$$K_{12}^{'} = H(K_{12}, K_{23}) \quad (10)$$

이 경우 K_{23}을 가지고 있는 SUB-MTU아래의 RTU들은 별도의 통신 없이 키의 업데이트가 가능하다. K_{24}의 경우는 MTU에 의해서 랜덤한 값으로 새로운 키 K'_{24}가 생성된다. 이렇게 생성된 새로운 키 K'_{24}, K'_{12}, K'_0을 SUB-MTU가 MTU로부터 안전하게 전송받게 된다. SUB-MTU는 이렇게 전송받은 키를 자신에게 속한 각 RTU의 키로 암호화하여 전송 하게 된다.

이를 통해 어떤 노드가 탈퇴하게 되면 해당 노드가 가지고 있어서 문제가 되는 모든 키를 업데이트함으로써 보안상의 문제점을 막을 수가 있다.

6. 비교 및 보안 분석

6.1 비교

<표 3>에서 보이는 것처럼 기존의 SKE는 RTU와 RTU 사이의 직접적인 통신을 지원하지 못한다. 하지만 SKMA와 본 논문에서 제안하는 방식은 RTU간의 직접적인 통신을 지원한다. 또한 SKE와 SKMA는 모두 메시지 브로드캐스팅을 지원하지 않는다. 이는 최상위 MTU에서 모든 RTU로 메시지를 전송해야 할 경우 각각의 키로 메시지를 암호화해서 전송을 하여야 함으로 비효율적이다. 하지만 제안하는 방식은 이러한 메시지 브로드캐스팅을 지원한다.

(그림 11)에서 (a)는 SUB-MTU의 수가 2개 이고 각각의 SUB-MTU에 속하는 RTU의 수에 따른 LKH++방식과 제안하는 방식에서 RTU가 저장해야하는 키의 수 비교 그래프이다. (그림 11)에서 (b)는 SUB-MTU에 속하는 RTU의 수를 100으로 고정하고 SUB-MTU의 수에 따른 LKH++방식과 제안하는 방식에서 RTU가 저장해야하는 키의 수 비교 그래프이다. 그래프에서 보는 것처럼 본 논문에서는 제안하는 방식은 LKH++를 그대로 적용하는 것에 비해서 RTU가 보관해야하는 키의 수가 적다.

6.2 보안 분석

그룹키는 정당한 멤버들에 의해서만 공유되어야 한다. 만약 멤버 중에 하나가 더 이상 정당한 멤버가 아니거나, 아직 그룹에 가입되지 않은 멤버라면 그룹 데이터로의 접근이 차단되어야 한다. 이러한 목적을 달성하기 위해서는 다음과 같은 요구사항을 만족해야한다.

- GKS(Group Key Secrecy): 그룹키를 얻고자하는 공격자의 시도가 계산상 불가능하게 하는 것을 보장하는 것을 말한다.
- FS(Forward Secrecy): 예전 그룹키를 알고 있는 수동적인 공격자가 현재의 그룹키를 얻을 수 없음을 보증함
- BS(Backward Secrecy): 그룹키의 일정 부분을 알고 있는 수동적인 공격자가 앞선 그룹키를 찾아낼 수 없음을 보장함

제안하는 시스템에서 사용하는 함수 $E_k(D)$,H(D)는 다음과 같은 특성을 가진다.

Property 1. 암호 함수 $E_k(D)$는 Ciphertext-only attack에 안전하다. $c_i=E_k(p_i)$가 공격자가 모르는 키 k로 공격자가 모르는 평문 p_i를 암호화한 암호문이라고 할 때, 암호문 집합 $c_0, c_1,...,c_m$이 공격자에게 주어질 때 공격자가 무시할 수 없는 확률로 평문 p_i와 관련된 어떠한 정보라도 찾아내는 것은 계산상 불가능하다.

Property 2. 함수 H(D)는 일방향 함수 이다. 주어진 g(x) 값에서 무시할 수 없는 확률로 x값을 찾는 것은 계산상 불가능하다.

6.2.1 공격자 모델

공격자 모델은 크게 수동적 공격자(passiva attacker)와 능동적 공격자(active attacker)로 나눌 수 있으며 각각의 특징은 다음과 같다.

- 수동적 공격자 : 프로토콜의 참가자와 실제로 통신에 참여하지 않고 두 참가자 사이의 통신 내용을 도청함으로써 공격을 수행하는 공격자
- 능동적 공격자 : 단순히 참가자들의 통신 내용을 도청하는 것뿐만 아니라 전송되는 메시지를 위변조하거나 새로운 메시지를 삽입하거나 하는 등 실제 통신에 참여하는 보다 강력한 공격자

본 논문에서 고려하는 키 관리 프로토콜에 대한 공격은 다음과 같다.

- GKS에 대한 공격 : 공격자는 해당 그룹에서 사용하고 있는 그룹키를 계산하내는 것을 목적으로 한다. 이를 위해 공격자는 그룹에 속한 멤버들이 전송하는 메시지들을 도청한다. 또한 공격자는 그룹에 속한 멤버로 가장하여 자신이 도청한 메시지를 위변조 하고 이를 다른 멤버에게 전송하여 얻은 정보를 활용하여 그룹키 계산을 시도한다..
- FS에 대한 공격 : 이전에 그룹에 속해있던 공격자가 현재 그룹에서 사용하고 있는 그룹키를 계산해내는 것을 목적으로 한다. 공격자는 그룹에 속해 있었기 때문에 키 업데이트 이전의 비밀 정보를 가지고 있다. 공격자는 해당 정보를 이용하여 그룹키 계산을 시도한다.
- BS에 대한 공격 : 새롭게 그룹에 가입된 공격자가 과거 그룹에서 사용했던 그룹키를 계산해내는 것을 목적으로 한다. 공격자는 현재 그룹에 속해 있으므로 해당 그룹의 현재 비밀정보를 알고 있다. 공격자는 이를 바탕으로 이전의 그룹키 계산을 시도한다.
- Known Key passive 공격 : 공격자는 두 사용자간 사용하고 있는 세션키를 계산해 내는 것을 목적으로 한다. 과거의 세션키와 전송 정보 및 현재 세션의 전송 정보를 이용하여 현재의 세션키 계산을 시도한다.
- Known Key Impersonation 공격 : 공격자는 두 사용자간 사용하고 있는 세션키를 계산해 내는 것을 목적으로 한다. 세션에 직접 참여한 후, 과거의 세션키와 전송 정보, 현재 세션의 전송 정보를 이용하여 다른 사용자 X에게 Y로 위장하여 세션키를 설정하여 세션키 계산을 시도한다.

6.2.2 Group key Secrecy

Group Key Secrecy는 능동적 공격자가 암호문이나, 키의 배포나, 업데이트에 사용되는 메시지를 가지고서 그룹키를 얻으려는 시도가 계산상 불가능해야 함을 말한다. 능동적 공격자는 그룹 통신에서 전송되는 메시지를 도청할 수 있으며, 해당 메시지를 변경, 삭제, 삽입역시 가능하다. 제안하는 키 관리 방식에서는 그룹키의 생성과 전송에 Property 1을 만족하는 암호화 함수 $E_k(D)$와 Property 2를 만족하는 일방향 해쉬 함수 H(D)를 사용한다. 그러므로 능동적인 공격자는 $\Omega(2^b)$(b는 키의 길이) 확률을 가지는 전수 조사 공격보다 더 좋은 가능성을 가질 수 없다.

6.2.3 Forward Secrecy

어떤 사용자가 그룹에서 탈퇴하면 자신이 그룹에서 사용하고 있던 키 경로의 키들을 전부 알고 있다. 이는 공격자

〈표 3〉 기존의 방법들과 비교

기능적 요구사항	SKE	SKMA	LKH++	Iolus Framework	제안하는 방식
RTU-RTU 직접 통신	불가능	가능	가능	-	가능
메시지 브로드캐스팅	불가능	불가능	가능	불가능	가능
키 갱신	가능	가능	가능	가능	가능
실시간 통신	강함	강함	강함	약함	강함
공개키 알고리즘 사용 여부	사용	사용안함	사용안함	사용안함	사용안함
RTU가 저장하는 키의 수	4	2	$\log_2 nm+1$	-	$\log_2 n+2$

에게 많은 정보를 제공한다. 이때 사용자가 이러한 정보를 가지고 현재의 그룹키를 도출해 낼 수 없다면 Forward Secrecy를 만족하게 된다. 제안하는 방식에서는 사용자가 그룹을 탈퇴하면 새로운 그룹키를 5장 2.3에서 언급한 방식으로 업데이트하기 때문에 시스템에서 암호학적으로 안전한 해쉬 알고리즘을 사용한다고 가정한다면 공격자는 전수 조사 공격을 통해 키를 얻을 수 있는 $\Omega(2^b)$(b는 키의 길이) 확률보다 더 좋은 확률로 키를 얻을 수 없다. 그러므로 제안하는 키 관리 방식은 Forward Secrecy에 안전하다.

6.2.4 Backward Secrecy

여기서는 Backward Secrecy에 대해서 살펴보겠다. 어떤 사용자가 그룹에 새로 가입하게 되면 그는 키 경로에 해당되는 모든 키를 받게 된다. 이때 사용자가 이러한 정보를 가지고 이전의 그룹키를 도출해 낼 수 없다면 Backward Secrecy에 안전하다고 할 수 있다. 이때 사용자는 이전의 키 업데이트 메시지와 이전의 그룹키로 암호화된 메시지를 가지고 있다. 제안하는 방식에서는 새로운 사용자가 가입될 때 기존의 그룹키를 해쉬 알고리즘을 이용하여 키를 업데이트 한다. 시스템에서 암호학적으로 안전한 해쉬 알고리즘을 사용한다고 가정한다면 공격자는 전수 조사 공격을 통해 키를 얻을 수 있는 $\Omega(2^b)$(b는 키의 길이) 확률보다 더 좋은 확률로 키를 얻을 수 없다. 그러므로 제안하는 키 관리 방식은 Backward Secrecy에 안전하다.

6.2.5 Key Freshness

Key Freshness는 세션키가 매번 다르게 생성되어야 생성 목적과 부합한다. 즉, 프로토콜에 참여하는 주체가 자신이 직접 생성하거나 전송받은 세션키에 대해, 해당 세션키는 이전에 한 번도 사용한 적이 없는 가장 최근에 생성된 키임을 확인할 수 있음을 말한다. 이를 제공하는 방법으로 타임스탬프 기법, nonce 기법, 카운터 기법이 있다. 제안하는 키 관리 방식에서는 카운터 기법을 활용하여 Key Freshness를 만족한다.

6.2.6 Known Key Security.

Known Key Security는 두 사용자 A, B 사이의 과거 세션키가 노출되더라도 현재의 세션키의 안전성에는 아무런 영향을 미치지 않아야 함을 말한다. 이를 만족하지 않을 경우, 과거의 세션키와 전송 정보 및 현재 세션의 전송 정보를 이용하여 현재의 세션키를 획득하는 Known Key Passive Attack과, 세션에 직접 참여한 후, 과거의 세션키와 전송 정보, 현재 세션의 전송 정보를 이용하여 다른 사용자 X에게 Y로 위장하여 세션키를 설정하는 Known Key Impersonation Attack이 가능하다.

제안하는 방식에서는 RTU나 SUB-MTU가 배포되기 전에 랜덤하게 생성된 카운터값을 세션키 생성을 위해 나누어 가진다. 여기서 사용하는 의사난수 생성기는 암호학적으로 완벽히 안전한 생성기를 사용한다. 또한 세션키를 생성할 때 암호학적으로 안전한 해쉬 함수를 사용하기 때문에 공격자는 과거의 세션키로부터 현재의 세션키에 관련한 정보를 제공받지 못한다. 따라서 Known Key Security를 만족한다.

7. 결 론

국가기관들은 국가 기반시설인 전기, 수도 가스 수송망 등 국가 기능 수행에 필요한 기본적이고 필수적인 시스템에 SCADA라는 자동화 시스템에 의존하고 있다. 만약 이러한 자동화 시스템이 악의적인 의도를 가진 공격자로부터 공격을 받게 된다면 그 피해는 국가 전반에 걸친 심각한 수준이 될 것이다. 하지만 기존의 SCADA 시스템은 폐쇄된 망에서의 운영만을 고려하여 보안 기능이 매우 취약하거나 부재해 있는 실정이다. 게다가 최근에 와서는 폐쇄된 망에서 운영되던 SCADA와 공용망과의 연계가 추진되고 있어, 보안 취약성에 따른 공격 가능성이 커지고 있다. 그러므로 이러한 피해를 미리 예방하고 공격에 대비하기위해서는 SCADA 시스템에 적합한 보안시스템에 대한 연구가 필요하다.

최근 SCADA 시스템을 보다 안전하게 하기위해 여러 방면에서 연구가 진행되고 있다. 그중 SCADA 시스템에서 안전한 통신을 지원하기 위한 키 관리에 관한연구로 SANDIA에서 연구한 키 확립 프로토콜인 SKE와 Queensland University of Technology에서 연구한 SKMA가 있다. 기존의 SKE는 공개키 프로토콜을 사용하는 단점이 있고, SKMA는 메시지를 브로드캐스팅하지 못하는 단점이 있다.

본 논문에서는 안전한 SCADA 통신을 위한 효율적인 키 관리 구조를 제안하였다. 이러한 키 관리 방식을 사용할 경우 기존의 방법들이 제공하지 못하는 안전한 메시지 브로드캐스팅이 가능하다는 장점이 있다. 또한 RTU가 저장해야하는 키의 수가 기존의 LKH++방식을 그대로 적용했던 것에 비해 적은 장점이 있다. 마지막으로 제안하는 방식은 성능상의 제약을 가지고 있는 RTU의 계산량을 최소화하기 위해, 상위 노드(SUB-MTU 또는 MTU)에 계산량을 분배하여 RTU의 잠재적인 성능 병목을 해결하였다.

참 고 문 헌

[1] 김인중, 정윤정, 고재영, 원동호, “중요핵심시설(SCADA)에 대한 보안 관리 연구”, *한국통신학회논문지* Vol.30 No.8C, pp.838-848, 2005.

[2] Curts, K. “A DNP3 protocol primer,” Technical report, *DNP User Group*, 2005.

[3] GAO, “Critical Infrastructure Protection : Challenge and Efforts to Secure Control Systems,” http://www.gao.gov, Mar., 2004.

[4] S. Mittra, “Iolus: A Framework for Scalable Secure Multicasting,” Proc. ACM SIGCOMM'97, pp.277-88, 1997.

[5] Cheryl Beaver, Donald Gallup, Willian Neumann, Mark Torgerson, “Key Management for SCADA,” *Sandia*,

http://www.sandia.org/scada/documnets/013252.pdf:, Mar. 2002.

[6] Robert Dawson, Colin Boyd, Ed Dawson, Juan Manuel Gonzalez Nieto, "SKMA - A Key Management Architecture for SCADA Systems," *In Proc. Fourth Australasian Information Security Workshop*, Vol. 54, pp.138-192, 2006.

[7] H. Harney, E. Harder, "Logical Key Hierachy Protocol," Internet Draft(work in progress), draft-harney-spartr-lkhp-sec-00.txt, *Internet Engeneering Task Force*, Mar. 1999.

[8] Marcel Waldvogel, "The VersaKey Framework: Versatile Group Key Management," *IEEE JSAC,* Vol.17, No.9, Sept., 1999.

[9] IEEE Standards Board, "IEEE standard definition, specification, and analysis of systems used for supervisory control, data acquisition, and automatic control", Technical report, IEEE. http://ieeexplore.ieee.org/iel1/3389/10055/00478424.pdf, March 1994.

[10] American Gas Association, "Cryptographic protection of SCADA communications Part 1: Background, Policies and Test Plan," Technical Report 12-1 Draft 5 revision 3, American Gas Assocation. http://www.gtiservice.org/security/; 2005.

[11] Information Technology - Security Techniques - Key Management - Part 2: Mechani는 Using Symmetric Techniques ISO/IEC 11770-2 International Standard, 1996.

[12] Vinay M. Igure, Sean A. Laughter, Ronald D. Williams, "Security issues in SCADA networks," Computers & Security 25, pp.498-506, 2006.

[13] Roberto Di Pietro, Luigi V. Mancini, Sushil Jajodia, "Efficient and Secure Keys Management for Wireless Mobile Communications," Proceedings of the second ACM international workshop on Principles of mobile computing, pp.66-73, 2002.

[14] Instrumentation systems and Automation Society, "Security Technologies for Industrial Automation and Control Systems," ANSI/ISA-TR99.00.01-2007, Research Triangle Park, North Carolina, 2007.

[15] Instrumentation systems and Automation Society, "Integrating Electronic Security into the manufacturing and Control Systems Environment," ANSI/ISA-TR99.00.02-2004, Research Triangle Park, North Carolina, 2004.

[16] National Institute of Standards and Technology, "System Protection Profile - Industrial Control Systems v1.0," Gaithersburg, Maryland, 2004.

[17] American Petroleum Institute, "API 1164: Pipeline SCADA Security," Washington, DC, 2004.

[18] American Gas Association, "Cryptographic Protection of SCADA Communications; Part2: Retrofit Link Encryption for Asynchronous Serial Communications," AGA Report No. 12 (Part 2), Draft, 2005.

[19] Information Technology Laboratory, National Institute of Standards and Technology "Security Requirements for Cryptographic Modules," FIPS PUB 140-1, 1994.

[20] K. Stouffer, J. Falco and K. Kent, Guide to Supervisory Control and Initial Public Draft, National Institute of Standards and Technology, Gaithersburg, Maryland

[21] Chung Kei Wong, Hohamed Gouda, Simon S. Lam, "Secure Group Communications Using Key Graphs," In Proceedings of the ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp.68-79, 1998.

부록 A. 논문에서 사용된 영문 약어 풀이

〈표 3〉 약어 풀이

약어	약어 풀이
AGA	American Gas Association
API	American Petroleum Institute
BS	Backward Secrecy
CC	Common Criteria
CK	Common Key
C-S	Controller to Subordinate
FS	Forward Secrecy
GK	General Key
GKS	Group Key Secrecy
GSK	General Seed Key
HMI	Human Machine Interface
KDC	Key Distribution Center
LTK	Long Term Key
MTU	Master Terminal Unit
NIST	National Institute of Standards and Technology
P2P	peer-to-peer
RTU	Remote Telemetry Unit
SCADA	Supervisory Control And Data Acquisition
SK	Session Key
SKE	Sandia Key Management
SKMA	SCADA Key Management Architecture

최 동 현

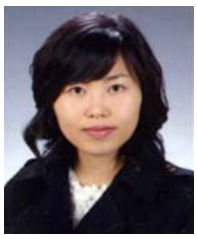
e-mail : dhchoi@security.re.kr

2005년 성균관대학교 정보통신공학부(학사)

2007년 성균관대학교 대학원 전자전기컴퓨터 공학과(공학석사)

2007년~현 재 성균관대학교 휴대폰학과 박사과정

관심분야 : 암호이론, 모바일 보안, 보안성 평가, DRM, SCADA



이 성 진

e-mail : sjlee@security.re.kr
2007년 성균관대학교 정보통신공학부(학사)
2008년~현 재 성균관대학교 휴대폰학과 석사과정
관심분야 : 암호이론, 정보보호, 네트워크 보안, SCADA, 보안성 평가

정 한 재

e-mail : hjjeong@security.re.kr
2006년 성균관대학교 정보통신공학부(학사)
2008년 성균관대학교 대학원 전자전기컴퓨터공학과(공학석사)
2008년~현 재 성균관대학교 휴대폰학과 박사과정
관심분야 : 정보보호, 보안성평가, 무선네트워크

강 동 주

e-mail : djkang@keri.re.kr
1999년 홍익대학교 전자전기제어공학과(학사)
2001년 홍익대학교 대학원 전기정보제어공학과(공학석사)
2001년-현 재 한국전기연구원 연구원
관심분야 : 전력 및 에너지 시스템 SCADA 보안

김 학 만

e-mail : hmkim7@icc.ac.kr
1991년 성균관대학교 전기공학과(학사)
1993년 성균관대학교 대학원 전기공학과(석사)
1998년 성균관대학교 대학원 전기공학과(박사)
1996년~2008년 한국전기연구원 선임연구원
2008년~현 재 인천시립대학 전기과 교수
2000년~현 재 대한전기학회 전력부문학회 편집위원
관심분야 : 전력시스템 해석 및 제어, 전력 IT Application 및 네트워크 보안

김 경 신

e-mail : kskim@mail.induk.ac.kr
1983년 성균관대학교 전자공학과 학사
1985년 성균관대학교 대학원 전자공학과(석사)
1997년 성균관대학교 대학원 정보공학과(박사)
1984년~1991년 삼성전자(주) 컴퓨터부문 선임연구원
1995년~현 재 인덕대학 인터넷TV방송과 교수
2001년~2002년 California State University, Northridge 객원연구원
관심분야 : 정보보호, 암호이론

원 동 호

e-mail : dhwon@security.re.kr
1976년~1988년 성균관대학교 전자공학과(학사, 석사, 박사)
1978년~1980년 한국전자통신연구원 전임연구원
1985년~1986년 일본 동경공업대 객원연구원
1988년~2003년 성균관대학교 교학처장, 전기전자 및 컴퓨터공학부장, 정보통신대학원장, 정보통신기술연구소장, 연구처장
1996년~1998년 국무총리실 정보화추진위원회 자문위원
2002년~2003년 한국정보보호학회장
2002년~현 재 대검찰청 컴퓨터범죄수사 자문위원, 감사원 IT 감사 자문위원
2007년~현 재 성균관대학교 정보통신공학부 교수, 한국정보보호학회 명예회장, 정보통신부지정 정보보호인증기술연구센터 센터장
관심분야 : 암호이론, 정보이론, 정보보호

김 승 주

e-mail : skim@security.re.kr
1994년~1999년 성균관대학교 정보공학과(학사, 석사, 박사)
1998년~2004년 한국정보보호진흥원(KISA) 팀장
2004년~현 재 성균관대학교 정보통신공학부 교수
2001년~현 재 한국정보보호학회, 한국인터넷정보학회, 한국정보과학회, 한국정보처리학회 논문지 및 학회지 편집위원
2002년~현 재 한국정보통신기술협회(TTA) IT 국제표준화 전문가
2005년~현 재 교육인적자원부 유해정보차단 자문위원, 디지털콘텐츠유통협의체 보호기술워킹그룹 그룹장
2007년~현 재 대검찰청 디지털수사 자문위원, KISA VoIP 보안기술 자문위원, 기술보증기금 외부 자문위원, 전자정부서비스보안위원회 사이버침해사고대응 실무위원회 위원
관심분야 : 암호이론, 정보보호표준, 정보보호제품 및 스마트카드 보안성 평가, PET