디지털산업정보학회 논문지 제16권 제4호-2020년 12월
http://dx.doi.org/10.17662/ksdim.2020.16.4.029

원격 감시 제어시스템에서 키 관리 방안 연구

이 건 직*

Research on key management for supervisory control and data acquisition system

Lee Keonjik

〈Abstract〉

SCADA (Supervisory Control and Data Acquisition) systems for remote monitoring, data acquisition and control are applied to major industrial infrastructures including power, water and railroad. Recently, there are many researches on key management scheme for secure communication due to change to the open network environment. These systems are located at far distances and are connected to the main control center through various types of communication methods. Due to the nature of these systems, they are becoming the significant targets of cyber attack. We propose an efficient key management scheme which is established on ID-based cryptosystem without an expensive computation on MTU (Master Terminal Unit), Sub-MTU, and RTU (Remote Terminal Unit). The proposed method is secure and effective in key management among multiple legitimate devices.

Key Words : Key Management, Critical Infrastructure, ID-based Cryptosystem

Ⅰ. 서론

SCADA는 감시 제어 및 데이터 취득 시스템으로서 산업 제어 시스템으로 분류되며, 정보통신기술이 발전하고 네트워크 범위가 광역화되면서 전기, 석유, 철도, 상하수도 등 주요 공공 및 산업 기반 시설에 적용되고 있다. SCADA 시스템이 폐쇄형 네트워크 환경에서 벗어나 외부의 네트워크와 상호 연결되어 기존에는 고려할 필요가 없었던 보안 문제들이 도출됨에 따라 안전한 데이터 통신을 위한 키 합의 기법들이 연구되고 있다. 기존의 SCADA에서 키 관리 프로토콜은 사전에 장기키(LTK: long term key)가 각 장치에 발급되고 그 장기키를 이용하여 두 통신 장치간에 비밀 세션키를 공유하는 형태이다[1, 2]. 장기키는 비밀 공유된다고 가정하는데, 장기키가 노출되는 경우 시스템 안전을 위해 장기키의 합의 및 교체가 필요하며, 각 장치들의 개인키(private key)가 노출될 경우에 키 복구가 요구된다.

SCADA 환경에서 효율적인 키 관리 연구는 시스템의 말단에 연결된 원격 장치들의 제한된 컴퓨팅 자

* 대구대학교 자유전공학부 교수

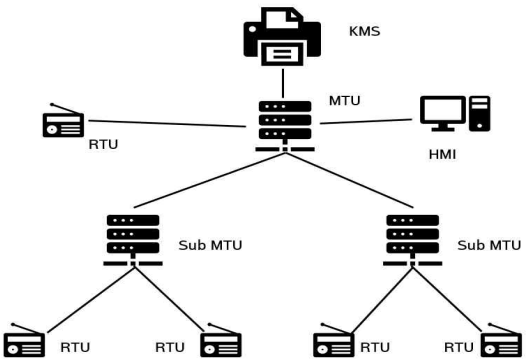
원으로 인해 공개키(PKC: public key cryptosystem)와 비밀키 암호시스템(SKC: secret key cryptosystem)이 혼합된 형태의 키 합의 프로토콜이 연구되고 있다[3-5]. 특히 복잡한 공개키 관리의 부담이 없는 IBC (ID-based cryptosystem) 암호가 주목을 받고 있는데 IBC를 이용하여 통신 개체간의 세션키를 효율적으로 합의하기 위해서 다양한 연구가 진행되고 있다[6-9].

본 논문에서는 IBC 기반의 SCADA 시스템 환경에서 장치 간의 통신에 필요한 비밀 세션키 설정을 위하여 장기키를 효율적으로 합의하는 프로토콜을 제안한다. 그리고 제안된 방법의 성능이 개선됨을 살펴보고, BAN 로직을 통해 안전한 통신 채널이 연결됨을 입증한다[10, 11]. 본 논문의 구성은 2장에서는 기존의 SCADA 시스템 개요와 SCADA 키 관리 프로토콜의 관련 연구를 설명한다. 3장에서는 제안한 키 합의 프로토콜과 키 복구 방안에 관해서 설명하고, 4장에서는 제안 기법의 안전성과 신뢰성을 분석하고 마지막으로 5장에서는 결론을 맺는다.

II. 관련 연구

2.1 SCADA 시스템 구조

SCADA 시스템은 일반적으로 MTU, RTU 등 여러 장비가 계층적으로 구성된다. 여기서 계층은 실제 시스템의 운영환경이나 장비 성능, 거리 등 여건에 따라 여러 계층으로 모델링 될 수 있다. SCADA 시스템 구조가 복잡할수록 키 관리 및 분배가 어려워지는데 본 논문에서는 시스템 구조를 <그림 1>처럼 단순화하기로 한다. SCADA 시스템에는 여러 MTU가 존재하며 최상위에 위치하는 MTU는 하위에 있는 Sub-MTU나 RTU들과 연결된다. 관리자는 전체 시스템의 키 관리를 위해 KMS(Key Management System)를 운영하며 KMS는 MTU와 연결된다. HMI(Human Machine Interface)는 운영자가 SCADA 시스템과 상호 작용을 하기 위한 콘솔 장치로서 제어화면, 경보화면, 상태화면, 리포트 등을 위한 GUI 환경을 통해서 데이터 출력, 제어, 연산 등의 기능을 제공한다. SCADA 시스템의 계층 구조에서 RTU는 물리 환경과 상호 작용을 수행하는 센서와 이를 제어하는 마이크로 프로세서로 구성된다. RTU는 일반적으로 불안전한 원격지에 위치하므로 MTU 또는 Sub-MTU와 RTU 사이의 기밀성 유지 및 키 관리가 매우 중요하다. RTU는 원격지의 장치 관리를 담당하며 수집된 데이터와 상태 정보를 취합하여 상위 Sub-MTU 또는 MTU에 안전하게 전송하고 반대로 MTU들은 제어 신호를 RTU들에게 안전하게 전송하여 RTU들을 제어한다[12, 13].



<그림 1> SCADA 시스템 구조

MTU는 RTU로부터의 데이터를 수집 및 저장하고 RTU로의 송신을 관리하며, 관리자와의 인터페이스를 통해 현장 감시 및 제어 기능을 제공하며, 각종 경보 및 사건 기록 등을 수행한다. Sub-MTU는 장치의 설치 거리나 환경을 고려하여 MTU의 역할을 분담하면서 하위 RTU 장치들을 제어한다. SCADA 시스템에서 안전한 데이터 통신을 보장하기 위해 두 개체간의 단기 세션키를 설립하려면 PKC를 이용한 두 개체간

의 장기키 합의가 필요하다.

2.2 SCADA 키 관리 프로토콜

PKC는 송신자가 메시지를 인증기관에서 받은 수신자의 공개키로 암호화하여 전송하고, 수신자는 자신의 개인키로 암호문을 복호화하는 기법이다. 문제점은 거대한 공개키 관리를 위하여 인증서 기반의 PKI (public key infrastructure)를 사용하므로 복잡한 인증서 관리 문제가 부가적으로 발생한다. 또한, 인증서 폐기 목록(Certificate Revocation List)을 통해 인증서의 유효성을 확인하는 작업이 필요하며 그리고 송수신 장치 간에 고비용의 암호/복호화 연산으로 인한 네트워크상의 오버헤드가 높다.

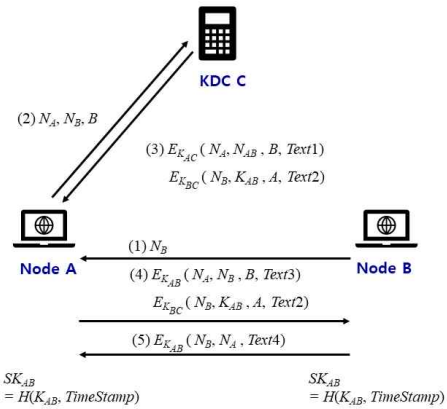
PKC의 본질적인 문제점들을 해결하기 위해 IBC가 제안되었으며, 송신자는 인증 기관의 도움을 받지 않고 이메일이나 IP주소처럼 타인과 구별되는 수신자의 공개된 식별 정보로부터 공개키를 획득한다. 겹선형 페어링(bilinear pairing)을 사용한 IBC가 구현되었는데 문제점은 포함된 연산 중에서 페어링 계산과 MTP(Map-to-Point) 해시 함수의 계산량이 너무 많다는 것이다[14, 15].

2.2.1 SKE(Sandia Key Management)

SCADA에서 처음으로 제안된 키 관리 기법으로서 MTU (Sub-MTU)와 RTU처럼 계층 구조에서는 제어장치-종속장치 방식을 이용하며 SKC 암호방식을 이용한다. 그리고 Sub-MTU들 간의 통신은 Peer-to-Peer 방식을 취하며 PKC 암호방식을 이용한다[1]. 제어장치 MTU와 종속장치 RTU 간에 세션키 설정은 사전에 LTK, FLAG, ID, LEN, TVP(Time Varying Parameter)값들을 비밀리에 공유한다고 가정한다. 제어 장치는 GSK(General Seed Key)와 난수를 입력으로 해시 함수 H를 실행하여 GK(General Key)를 생성하고 난 후, SKC(비밀키로 장기키 사용)로 GK를 암호화하여 종속장치로 보낸다. 종속 장치는 받은 암호문에서 장기키를 사용하여 GK를 복원하고 쌍방은 SK = H(GK, FLAG, ID, LEN, TVP)을 각각 계산하여 세션키 SK를 공유한다.

2.2.2 SKMA(SCADA Key Management Architecture)

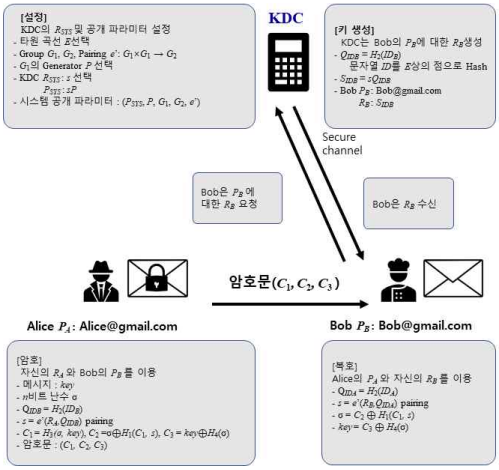
SKE에서는 PKC와 SKC 암호방식을 혼용하였는데 SKMA는 SKC방식으로만 구성된다[2]. <그림 2>는 SKMA에서 노드 B가 추가될 때 두 노드 A와 B간에 세션키 SK_{AB}의 생성을 나타낸다. B와 KDC (Key Distribution Center)간에는 장기키 K_{BC}가 사전에 공유된다고 가정한다. 그리고 서로 통신할 두 노드 A와 B간에 SKC를 사용하여 노드키 K_{AB}가 교환된다. 마지막으로 두 노드간 암호화 통신에 사용될 세션키 SK_{AB}는 $H(K_{AB}, TimeStamp)$ 계산을 통해서 생성된다. 문제점은 노드키를 생성하려면 매번 KDC를 통해서 분배받아야 하는 번거로움이 있다. 따라서 노드 간 비밀통신을 위해 고속의 SKC를 사용하지만 통신량이 많고 KDC오버헤드가 많다는 단점이 있다.



<그림 2> SKMA에서 세션키 SKAB 설정

2.2.3 ASKMA(Advanced SKMA)

ASKMA는 논리 키 계층(Logical Key Hierarchy) 구조를 이용한 키 관리 기법이다[16]. 계층 구조에서 상위의 MTU와 중간 Sub-MTU들의 관계는 이진 트리로 구성되며, MTU 또는 Sub-MTU와 하위 RTU들 간에는 *n*-ary 트리로 구성된다. 특징은 브로드 캐스팅을 지원하는 그룹 키 관리를 제공하며, 새로운 RTU가 SCADA 시스템에 가입하거나 기존의 RTU가 탈퇴할 때 그룹 키를 업데이트한다. 노드의 추가/삭제의 경우에 해당 노드의 상위 레벨의 모든 키를 업데이트하고 이를 이용하는 노드들이 가진 키도 모두 업데이트해야 하므로 비용이 높다.



<그림 3> IBCKM에서 LTK 설정

2.2.4 IBCKM(IBC Key Management)

암호시에 송신자의 개인키와 수신자의 공개키를 이용하고, 복호시에는 송신자의 공개키와 수신자의 개인키를 이용하여 IBC상에서 송신자 인증이 가능한 키 관리 기법이다[5, 15]. <그림 3>에서 설정은 KDC의 시스템 개인키, 시스템 공개키, 그리고 시스템 공개 파라미터를 구성한다. 키 생성은 KDC가 요구한 사용자의 공개키에 대응하는 개인키를 계산하며, KDC는 개인키를 보안 채널로 전송한다. 암호는 송신자의 개인키와 수신자의 공개키를 이용하여 메시지를 암호하며, 암호문 (C_1, C_2, C_3)가 수신자에게 전송된다. 복호 단계는 수신자의 개인키와 송신자의 공개키를 이용하여 암호문에서 메시지를 복원한다. 문제점은 키 생성, 암호, 그리고 복호 과정에서 고비용 페어링 계산과 고비용 MTP 해시 함수가 사용된다.

Ⅲ. 제안한 기법

본 논문에서 제안하는 IBC기반의 키 합의 프로토콜은 고비용의 페어링과 MTP 해시 함수를 사용하지 않으며 공개 채널을 통해서 쌍방이 안전하게 키를 공유할 수 있다. 제안하는 프로토콜은 셋업 단계, 키 발급 단계, 그리고 인증 및 키 합의 단계로 구성된다. 프로토콜의 기술을 위해 사용되는 표기법은 다음과 같다.

- ID_i: 개체 i의 identity와 유효기간
- R_{SYS}: 시스템의 개인키
- P_{SYS}: 시스템의 공개키
- R_A (R_B): 개체 A (B)의 개인키
- P_A (P_B): 개체 A (B)의 공개키
- a (b): 개체 A (B)가 선택한 난수
- K_{AB}: 개체 A와 B의 합의된 비밀 공유키

3.1 셋업 단계

KDC는 k 비트의 소수 p를 선택하고 시스템 파라미터로 유한체 F_P상에서 타원 곡선 포인트 $E_P(a, b)$들의 집합 E/F_P, 타원 곡선 덧셈 그룹 $G = \{(x, y): x \text{ and } y \in F_P \text{ and } E_P(a, b)\} \cup \{O\}$, 그리고 그룹 G의 생성자 P를 설정한다. 여기서 k는 보안 비트 크기이며 O는 타원 곡선상에서 무한대 포인터이다. KDC의 비밀 개인키 $R_{SYS} \in Z_p^*$는 난수로 선택하고, 대응되

는 공개키는 다음 수식을 계산해서 구한다.

$$P_{SYS} = R_{SYS}P$$

해시 함수 H_1, H_2, H_3는 다음과 같다.

$$H_1 = \{0, 1\}^* \times G \rightarrow Z_p^*$$
$$H_2 = \{0, 1\}^* \times G \times G \rightarrow Z_p^*$$
$$H_3 = \{0, 1\}^* \times \{0, 1\}^* \times G \times G \times G \rightarrow \{0, 1\}^k$$

시스템 파라미터 $\{F_p, E/F_p, G, P, P_{SYS}, H_1, H_2, H_3\}$는 공개하고, R_{SYS}는 비밀리에 안전하게 보관한다.

3.2 키 발급 단계

시스템에 참여하는 모든 개체는 자신의 ID에 대한 개인키를 KDC로부터 발급받는다. 식별자 ID_A를 가진 개체 A의 경우에 KDC는 다음의 과정을 수행한다.

먼저 난수 $n_A \in_R Z_p^*$를 선택하고, $N_A = n_AP$를 계산한다. A의 개인키 R_A를 계산하면 다음과 같다.

$$R_A = n_A + H_1(ID_A, N_A)R_{sys}$$

그리고 KDC는 R_A와 N_A를 안전한 경로를 통해 A에게 발급한다. 여기서 N_A는 키 합의 단계에서 ID로부터 공개키 계산과 서명 검정에 이용된다.

3.3 인증 및 키 합의 단계

개체 A는 난수 $a \in Z_p^*$를 선택하고, 다음을 계산한다.

$$N_A' = (a \oplus R_A)P$$
$$S_A = (a \oplus R_A) + H_2(ID_A, N_A, N_A')R_A$$

그리고 메시지 (ID_A, N_A, N_A', S_A)를 비보안 채널을 통해 B에게 보낸다. 개체 B는 난수 $b \in Z_p^*$를 선택 후, 다음의 두 값을 계산한다.

$$N_B' = (b \oplus R_B)P$$
$$S_B = (b \oplus R_B) + H_2(ID_B, N_B, N_B')R_B$$

그리고 메시지 (ID_B, N_B, N_B', S_B)를 비보안 채널을 통해 A에게 전송한다. 다음으로 A는 B로부터 받은 메시지 (ID_B, N_B, N_B', S_B)를 이용하여 B의 공개키를 계산한다.

$$P_B = N_B + H_1(ID_B, N_B)P_{sys}$$

R_B와 P_B의 수학적 관계는 다음 수식에서 확인할 수 있다.

$$\begin{aligned} P_B &= R_BP \\ &= (n_B + H_1(ID_B, N_B)R_{sys})P \\ &= n_BP + H_1(ID_B, N_B)R_{sys}P \\ &= N_B + H_1(ID_B, N_B)P_{sys} \end{aligned}$$

그리고 A는 다음 수식을 계산해서 서명을 검증한다.

$$N_B' + H_2(ID_B, N_B, N_B')P_B = S_BP$$

일치하면 메시지 (ID_B, N_B, N_B', S_B)가 중간에 변조가 없음이 확인되고, 개체 B로부터 발송됨도 확인된다. 마지막으로 개체 A는 개체 B와 비밀리에 공유하는 장기키를 다음과 같이 계산한다.

$$K_{AB} = H_3(ID_A, ID_B, N_A', N_B', (a \oplus R_A)N_B')$$

해시 함수안에 포함된 $(a \oplus R_A)N_B'$값은 A와 B가 협력하여 각자의 비밀 데이터를 결합해서 생성한 값으로

서 개별적으로는 생성 불가능하다.

다음으로 개체 A의 키 계산 과정과 유사하게 개체 B는 A로부터 받은 메시지 (ID_A, N_A, N'_A, S_A)로부터 A의 공개키를 계산한다.

$$P_A = N_A + H_1(ID_A, N_A)P_{sys}$$

그리고, 다음의 수식을 계산한다.

$$N'_A + H_2(ID_A, N_A, N'_A)P_A = S_AP$$

좌우변이 일치하면 수신 메시지의 무결성이 확인되고 발신자가 A임이 보장된다. 최종적으로 B도 A와 동일한 비밀 대칭키 K_{AB}를 공유하게 된다.

$$K_{AB} = H_3(ID_A, ID_B, N'_A, N'_B, (b \oplus R_B)N'_A)$$

여기서 B가 계산한 K_{AB}에 포함된 $(b \oplus R_B)N'_A$는 B가 독자적으로 계산을 못하며, 이 값과 A가 생성한 K_{AB}에 있는 $(a \oplus R_A)N'_B$는 같은 결과값을 가진다.

3.4 키 복구

각 장치들은 SCADA시스템 가입시에 KDC와 공유하는 비상키(emergency key)를 제안된 프로토콜을 통해 발급받아 시스템의 비상 복구시에만 사용하도록 물리적으로 안전하게 보관한다[5]. 어떤 장치의 개인키가 노출되면 KDC는 새로운 개인키를 생성한 후 각 장치들과 사전에 공유된 비상키를 이용하여 대칭키 암호화를 수행한 후 그 결과를 전송해서 개인키를 복구한다.

시스템 개인키가 노출된 경우에 KDC는 시스템 개인키를 변경 후 각 장치의 개인키 복구와 동일하게 비상키들을 사용하여 생성한 새로운 개인키들을 모든 장치에게 발급한다. 그리고 각 MTU와 Sub-MTU는 제안된 프로토콜을 실행하여 RTU와의 장기키를 재합의한다.

IBC에서는 모든 개체들의 개인키를 KDC가 알기 때문에 일반 사용자 네트워크에는 부적합하지만 SCADA처럼 관리자의 제어권이 필요한 신뢰성 기반의 폐쇄 그룹망에서는 PKC보다 IBC가 더 효과적으로 사용될 수 있다. PKC를 SCADA에 적용하면 개인키가 노출될 경우에 제한된 자원을 가진 원격 RTU에서 장치마다 개인키/공개키를 새로 만들고 공개키 인증을 받아야 하므로 갱신 과정과 공개키 인증 관리가 복잡하다.

IV. 비교 분석

프로토콜의 설계시에 의도치 않은 오류로 인해 치명적인 결함이 생길 수 있다. 프로토콜 설계시에 준수해야할 보안 특성들을 검토하고 다음으로 논리적 안전성 분석 도구인 BAN 로직을 이용하여 제안된 프로토콜의 안전도를 검증하고 신뢰성을 분석한다[10, 11].

4.1 취약성 분석

제안된 프로토콜이 기본적인 보안 특성들을 준수하는지를 간단히 분석하며, 이러한 공격들에 대한 안전도는 타원 곡선상에서 이산 대수 문제의 어려움에 기반한다.

- Known key security (KKS)

비밀 세션키 한 개가 공격자에게 노출된 경우에 다른 세션키에 영향을 미쳐서는 안 된다. 제안된 프로토콜에서는 매 세션마다 새로운 세션키가 생성된다. 단기 비밀 값인 난수(N'_A계산시에 난수 a와 개인키 R_A 사용)가 세션키 생성에 이용되는데 이산대수 문제의

어려움 때문에 난수와 개인키의 계산이 어려워서 공격은 불가능하다.

● Basic impersonation resilience (BIS)

개체 A의 개인키를 알지 못하는 공격자가 개체 A로 위장하는 것은 불가능하다. 프로토콜에서 N_A'와 S_A를 계산시에 A의 개인키 R_A가 사용되는데 공격자가 A의 공개키로부터 개인키의 계산은 어려우며, 또한 개인키를 몰라서 S_A값도 계산할 수 없으므로 이 공격은 불가능하다.

● Perfect forward security (PFS)

개체 A의 개인키가 노출된 경우에 과거의 세션키는 노출되지 않아야 한다. 제안된 프로토콜에서는 공격자가 A의 개인키를 획득하더라도 세션키의 구성요소인 $(a \oplus R_A)N_B'$의 계산이 어려우므로 이 특성은 보장된다. 왜냐하면 N_A'로부터 a값을 계산하기가 불가능하기 때문이다.

● KDC forward security (KFS)

KDC의 시스템 개인키가 노출된 경우에도 과거의 세션키는 노출되지 않아야 한다. 공격자가 노출된 시스템 개인키를 이용하더라도 Perfect forward security와 같이 개체의 난수 a값의 계산이 어려워서 공격은 불가능하다.

● No key control (NKC)

개체 A 또는 B가 세션키가 미리 선택된 값을 갖도록 강제할 수 없다. 제안된 프로토콜은 두 개체 A와 B가 함께 프로토콜을 수행하여 키 K_{AB} (구성 요소 $(a \oplus R_A)N_B'$와 $(b \oplus R_B)N_A'$)를 합의하기 때문에 어느 한 개체가 키를 결정할 수 없다.

● Key-compromise impersonation attack (KIA)

공격자가 키 합의 중인 A의 개인키를 알고 A에게 B인 것처럼 위장하는 공격이다. 세션키를 빼내기 위해 B로 위장하지만, 공격자는 B의 개인키를 모르기 때문에 세션키 구성 요소인 서명 S_B의 계산이 불가능하다.

● Unknown key-share attack (UKSA)

A와 B가 세션키를 공유한다고 믿는데, B는 실제로 공격자 E와 세션키를 공유한 형태의 공격이다. 세션키를 생성할 때 A와 B의 식별자 정보가 교환되는 메시지 안에 포함되어 있으므로 공격은 불가능하다.

● Ephemeral secrets reveal resistance (ESRR)

개체의 단기 비밀값이 노출되더라도 세션키는 노출되지 않아야 한다. 프로토콜에서 개체 A의 단기 세션 비밀값인 난수 a가 노출된 경우에 공격자는 세션키 K_{AB}를 계산하려고 시도한다. 그러나, 공개키 P_A로부터 개인키 R_A를 계산하는 것이 어려우므로 공격자는 K_{AB}의 마지막 구성 요소인 $(a \oplus R_A)N_B'$를 계산할 수 없다.

4.2 신뢰성 분석

프로토콜의 설계시에 일반적 명세는 교환되는 메시지의 내용을 기술함에 치중한다. 이러한 명세는 논리적 조작이 힘들며 메시지의 명확한 의미를 이해하기가 어렵다. 따라서 보안 정형 명세를 통해 그 의미를 명확히 표현할 필요가 있다. BAN Logic은 각종 보안 프로토콜의 정형화된 분석을 위해 널리 사용되는 도구로서 비정형화된 확률적 방법이 아닌 논리적 추론 규칙을 적용한 정형화된 안전도 분석을 통하여 프로토콜의 목표 성취를 추론한다[10, 11]. 본 논문에서 제안된 프로토콜의 메커니즘을 정형 명세하고 검증을 위해 BAN Logic을 이용한다. BAN 로직을 이용하여 프로토콜을 검증하려면 먼저 프로

토콜 명세로부터 BAN 표기법을 이용하여 이상화된 프로토콜을 명세한다. 다음으로 초기 상태에 관한 가정들을 기술하고 프로토콜의 문장에 논리적인 의미를 덧붙인다. 그리고 BAN 규칙들을 적용하여 제안한 프로토콜의 보안성을 검증한다. 먼저 BAN 로직에서 사용되는 표기법은 다음과 같다.

- P, Q : 프로토콜에 참여하는 두 개체.
- K : 암호키.
- $P \mid\equiv X$: P는 X를 신뢰.
- $P \lhd X$: P는 X를 수신.
- $P \mid\sim X$: P는 X를 송신.
- $P \mid\mid\sim X$: P는 현재 세션에서 X를 송신.
- $P \mid\Rightarrow X$: P는 X에 대한 권한 소유.
- $\#(X)$: X가 새롭게 생성.
- $\overset{K}{\rightharpoonup} P$: P는 공개키 K 소유.
- $\{X\}_K$: 메시지 X는 키 K로 암호.
- $P \overset{K}{\leftrightharpoons} Q$: P와 Q가 K를 비밀 대칭키로 공유.
- $\frac{P}{Q}$: P가 참이면 Q도 참.

프로토콜을 검증하기 위해서 BAN 표기법을 이용하여 다음의 추론 규칙들을 적용한다.

R1. 대칭키(Symmetric key) 규칙

$$\frac{P \mid\equiv Q \overset{K}{\leftrightharpoons} P,\ P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}$$

: P가 Q와 P의 키 K를 신뢰하고 K로 만든 암호문을 수신하면, P는 Q가 X를 보냈음을 신뢰한다.

R2. 공개키(Public key) 규칙

$$\frac{P \mid\equiv \overset{K}{\rightharpoonup} Q,\ P \lhd \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X}$$

: P가 Q의 공개키 K를 믿고 Q의 개인키로 만든 암호문을 받았다면 Q가 평문X를 보냈음을 믿는다.

R3. 난스 검증(Nonce verification) 규칙

$$\frac{P \mid\equiv \#(X),\ P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

: P가 새로운 X를 신뢰하고, Q가 X를 보냈다는 것을 신뢰한다면, P는 Q가 X를 믿는다는 것을 신뢰한다.

R4. 관할(Jurisdiction) 규칙

$$\frac{P \mid\equiv Q \mid\Rightarrow X,\ P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$$

: P는 Q가 X를 제어하는 것을 신뢰하고, Q가 X를 믿는 것을 신뢰한다면, P는 X를 신뢰한다.

R5. 믿음(Belief) 규칙

$$\frac{P \mid\equiv (X,\ Y)}{P \mid\equiv X}$$

: P가 $(X,\ Y)$식을 신뢰하면, 식의 부분 X도 신뢰한다.

R6. 세션키(Session key) 규칙

$$\frac{P \mid\equiv \#(K),\ P \mid\equiv Q \mid\equiv X}{P \mid\equiv P \overset{K}{\leftrightharpoons} Q}$$

: P가 새로운 키 K를 신뢰하고 Q가 X를 믿는다는 것을 신뢰한다면 P는 키 K가 P와 Q사이의 비밀 대칭키라는 것을 신뢰한다. 여기서 X는 키 K를 생성할때 핵심 요소이다.

R7. Freshness 규칙

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,\ Y)}$$

: X가 새로우면, X를 포함한 식도 새롭다고 신뢰한다.

R8. 해시(Hash) 규칙

$$\frac{P \mid\equiv Q \mid\sim H(X),\ P \triangleleft X}{P \mid\equiv Q \mid\sim X}$$

: Q가 $H(X)$를 보낸 것을 신뢰하고, X를 받았다면 P는 Q가 X를 보냈다는 것을 신뢰한다.

R9. 합성(Synthetic) 규칙

$$P \mid\equiv Q \mid\sim X \mapsto P \mid\equiv \#(X)$$

: Q가 현재 세션에서 메시지 X를 보낸것을 신뢰할 려면 P는 X가 새로움을 당연히 신뢰해야한다.

다음은 BAN 로직을 이용한 프로토콜 검증 절차이며, 그 결과로서 두 개체에게 새로운 비밀키가 합의되었음을 확신하게 된다.

4.2.1 초기상태가정(Initial assumptions)

제안된 프로토콜의 초기 상태를 기술한다.

I_1: $A \mid\equiv \xrightarrow{P_B} B$, I_2: $B \mid\equiv \xrightarrow{P_A} A$,

I_3: $A \mid\equiv \xrightarrow{P_A} A$, I_4: $B \mid\equiv \xrightarrow{P_B} B$,

I_5: $A \mid\equiv \#(N_A)$, I_6: $B \mid\equiv \#(N_B)$,

I_7: $A \mid\equiv \#(N'_A)$, I_8: $B \mid\equiv \#(N'_B)$,

I_9: $A \mid\equiv B \mid\Rightarrow P_B$, I_{10}: $B \mid\equiv A \mid\Rightarrow P_A$,

I_{11}: $A \mid\equiv B \mid\Rightarrow N_B$, I_{12}: $B \mid\equiv A \mid\Rightarrow N_A$,

I_{13}: $A \mid\equiv B \mid\Rightarrow N'_B$, I_{14}: $B \mid\equiv A \mid\Rightarrow N'_A$.

초기 상태는 개체 A와 B는 각자의 공개키를 신뢰하며, 공개키, N, 그리고 N'에 대한 권한도 신뢰한다.

4.2.2 이상화된 형태 (Idealized form)

BAN 로직 표기법을 이용해서 제안된 프로토콜을 이상화된 형태로 바꾸면 다음과 같이 표현된다.

F_1: $A \rightarrow B$: $ID_A, N_A, N'_A, \{N_A, N'_A\}_{R_A}$

F_2: $B \rightarrow A$: $ID_B, N_B, N'_B, \{N_B, N'_B\}_{R_B}$

4.2.3 프로토콜 목표 (Protocol goals)

프로토콜의 궁극적 목표인 A와 B사이에 신뢰를 만들기 위하여 다음의 목표에 도달해야 한다. 첫 번째 목표는 G_1과 G_2의 성취이며, 키 K_{AB}가 A와 B간에 안전한 통신을 위한 비밀키임을 서로 믿게 하는 것이다. 두 번째 목표는 G_3와 G_4이며, A와 B간에 K_{AB}를 신뢰한다는 사실을 서로 믿게 하는 것이다.

G_1: $A \mid\equiv A \xleftrightarrow{K_{AB}} B$, G_2: $B \mid\equiv A \xleftrightarrow{K_{AB}} B$

G_3: $A \mid\equiv B \mid\equiv A \xleftrightarrow{K_{AB}} B$, G_4: $B \mid\equiv A \mid\equiv A \xleftrightarrow{K_{AB}} B$

4.2.4 프로토콜 검증 (Protocol verification)

BAN 로직을 사용하여 프로토콜의 이상화된 형식을 분석하여 유효성을 검증한다. BAN 로직의 추론 규칙들을 적용함으로써 형식화된 분석이 진행된다. 먼저 (F_1)에서 (V_1), (V_2), 그리고 (V_3)를 구한다.

V_1: $A \mid\equiv (N_A, N'_A)$

V_2: $B \triangleleft (N_A, N'_A)$

V_3: $B \triangleleft \{N_A, N'_A\}_{R_A}$

A는 (N_A, N'_A)를 신뢰하며, B는 (N_A, N'_A)를 수신받고 또한 A의 개인키로 암호화된 $\{N_A, N'_A\}_{R_A}$, 즉 (N_A, N'_A)의 서명도 전송받았다는 의미이다. 먼저 수신자 B의 측면에서 분석하면 (I_2)와 (V_3)에 (R2)규칙을 적용하면 (V_4)을 유도할 수 있다.

V_4: $B \mid\equiv A \mid\sim (N_A, N'_A)$

(V_4)는 B가 A의 공개키가 P_A임을 신뢰하고 A의 개인키 R_A로 암호화된 (N_A, N'_A)를 받았다면 B는 A가 (N_A, N'_A)를 보냈음을 신뢰한다는 의미이다. 다음으로 (I_5)와 (I_7)에 (R7)을 적용하면 (V_5)을 유도할 수 있다.

$$V_5:\ A\ |\equiv\ \#(N_A,\ N'_A)$$

A는 N_A가 현재 사용을 위해 새로 생성되었음을 신뢰하고 또한 N'_A를 신뢰하면 A는 메시지 (N_A, N'_A)도 새롭다고 신뢰할 수 있다는 의미이다. 그리고, (V_4)과 (V_5)에서 (V_6)를 얻을 수 있다.

$$V_6:\ B\ |\equiv\ A\ |\!|\sim\ (N_A,\ N'_A)$$

B는 현재 실행 중인 세션에서 A가 (N_A, N'_A)를 전송했음을 신뢰한다는 의미이다. (V_6)에 (R9)을 적용하면

$$V_7:\ B\ |\equiv\ \#(N_A,\ N'_A)$$

(V_7)은 B가 현재 세션에서 A가 (N_A, N'_A)를 전송했음이 참값이 되려면 B는 반드시 (N_A, N'_A)가 새로운 메시지임을 신뢰해야 한다는 의미이다. 다음으로 (V_4)과 (V_7)에 (R3)을 적용하면 (V_8)을 유도할 수 있다.

$$V_8:\ B\ |\equiv\ A\ |\equiv\ (N_A,\ N'_A)$$

B는 (N_A, N'_A)의 새로움을 신뢰하고, A가 (N_A, N'_A)를 보냈음을 신뢰하면 B는 A가 (N_A, N'_A)를 신뢰한다는 사실을 믿을 수 있다. (V_8)에 (R5)를 적용하면

$$V_9:\ B\ |\equiv\ A\ |\equiv\ N_A$$
$$V_{10}:\ B\ |\equiv\ A\ |\equiv\ N'_A$$

B가 메시지 (N_A, N'_A)를 신뢰하면 B는 부분 메시지 N_A 또는 N'_A를 각각 신뢰할 수 있다는 의미이다. 다음으로 (I_{12})와 (V_9)에 (R4)를 적용하면 (V_{11})를 얻는다.

$$V_{11}:\ B\ |\equiv\ N_A$$

B가 개체 A가 N_A를 제어함을 믿고, A가 N_A를 신뢰함을 믿는다면, B는 N_A를 신뢰한다는 의미이다. 마찬가지로 (I_{14})와 (V_{10})에 (R4)를 적용하면 (V_{12})을 얻는다.

$$V_{12}:\ B\ |\equiv\ N'_A$$

즉, B가 개체 A가 N'_A를 관할하는 것을 믿고, A가 N'_A를 신뢰하는 것을 믿는다면, B는 N'_A를 신뢰할 수 있다는 의미이다. 다음으로 B는 공개키 P_A와 양쪽의 비밀 값을 포함한 $(b \oplus R_B)N'_A$를 계산하고 마지막으로 공유 비밀키 K_{AB}를 생성한다. 먼저 $(b \oplus R_B)N'_A$와 (V_7)에 (R7)을 적용하여 (V_{13})을 유도한다.

$$V_{13}:\ B\ |\equiv\ \#((b\oplus R_B)N'_A)$$

(N_A, N'_A)가 새롭다면, (N_A, N'_A)를 서명 검증에 이용해서 결국 계산된 값도 새롭다는 의미이다. 유사하게 (V_{13})으로부터 (V_{14})을 유도해 낼 수 있다.

$$V_{14}:\ B\ |\equiv\ \#(K_{AB})$$

즉 $(b \oplus R_B)N'_A$가 새롭다면, 이를 포함하는 K_{AB}도 새롭다는 의미이다. (V_8)과 (V_{14})에 (R6)을 적용하면

$$V_{15}:\ B\ |\equiv\ A \overset{K_{AB}}{\longleftrightarrow} B$$

B가 세션키 K_{AB}를 새롭다고 믿고, B가 개체 A가

(N_A, N_A')를 신뢰함을 믿는다면, B는 키 K_{AB}가 A와 B사이에 공유키임을 믿는다. 여기서 (N_A, N_A')는 키 K_{AB}를 만들때 핵심 원소이다.

그리고, 프로토콜의 대칭성 때문에 개체 B는 개체 A도 K_{AB}에 대해 동일한 신뢰를 가진다고 믿음을 가지게 된다. 따라서, 마지막으로 (V_{16})를 유도해 낼 수 있다.

$$V_{16}: \quad B \mid\equiv A \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

다음으로 송신자 A의 측면에서 유사하게 프로토콜 유효성 분석을 전개하면 다음과 같다. 앞의 (F_2)에서 다음의 (V_{17}), (V_{18}), 그리고 (V_{19})를 유도할 수 있다.

$$V_{17}: B \mid\equiv (N_B, N_B')$$
$$V_{18}: A \triangleleft (N_B, N_B')$$
$$V_{19}: A \triangleleft \{N_B, N_B'\}_{R_B}$$

위 식들의 의미는 B는 (N_B, N_B')를 믿으며, A는 (N_B, N_B')를 전송받고 또한 B의 개인키로 암호화된 $(N_B, N_B')_{R_B}$도 전송받았다는 의미이다.

프로토콜에 참여하는 두 개체 A와 B는 상호 대칭적 역할을 동일하게 수행하므로 개체 A의 프로토콜 유효성 분석 전개 과정의 상세한 기술은 생략하며 마지막 두 과정만 기술하면 다음과 같다.

$$V_{20}: A \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

$$V_{21}: A \mid\equiv B \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

(V_{20})과 (V_{21})의 의미는 A는 대칭키 K_{AB}가 둘 사이에 공유키임을 신뢰하고 B도 K_{AB}에 대해 같은 믿음을 가진다고 신뢰한다는 의미이다.

4.2.5 프로토콜 결과 (Protocol results)

위의 전개 과정들을 정리하면 결론적으로 다음과 같은 신뢰에 도달함을 알 수 있다.

$$V_{20}(= G_1): A \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

$$V_{15}(= G_2): B \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

$$V_{21}(= G_3): A \mid\equiv B \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

$$V_{16}(= G_4): B \mid\equiv A \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

위 결과는 제안한 프로토콜이 앞에서 언급했던 궁극적인 목표에 도달했음을 나타내고 있다. 따라서 제안된 프로토콜은 통신 개체 A와 B 사이에 재사용 없는 비밀 공유키를 성공적으로 생성했다는 것을 보장하며 안전한 키 합의 프로토콜임을 알 수 있다.

4.3 키 관리 비교

키 합의 프로토콜을 수행하기 위해서는 페어링, MTP해시, 일반 해시, 유한체 덧셈과 곱셈, 타원 곡선 상에서 포인터 덧셈과 스칼라 포인터 곱셈(scalar point multiplication, SPM) 연산과 같은 공개키 기반의 연산과정이 필요하다. 이 중에서 가장 많은 수행시간이 필요한 연산은 SPM 연산이다. 모듈러 지수승 연산이 RSA 암호시스템의 성능을 좌우하듯 타원곡선 암호 기반의 시스템 성능은 SPM 연산에 의하여 좌우된다. SPM 연산은 임의의 랜덤수 k와 타원 곡선 위의 한 점 P의 곱셈 연산으로 정의되며, 타원곡선 위의 점 P의 k번 덧셈 연산으로 계산된다. 다른 연산들은 SPM에 비해 무시할 수 있는 연산량이므로 논문에서는 주로 SPM 연산을 비교한다. 특히, 세션 키 합의는

두 개체가 메시지를 교환하면서 진행하는 대화형 통신이므로 이 부분에서의 계산 효율성은 시스템 구현에 있어서 매우 중요한 요소이다. 고비용 연산인 SPM과 페어링 연산의 계산 비용을 비교해보면 1번의 SPM에는 약 2 msec의 연산시간이 필요하고, 페어링 연산 1번에는 약 18 msec의 비용이 소요된다[17]. 일반적인 해시 함수는 계산 시간이 고려되지 않지만 MTP 해시 함수는 SPM 1번 계산량 정도의 고비용 연산에 속한다. <표 1>은 제안하는 SCADA 키 관리 방법과 기존 방식들과의 비교를 보여주고 있다.

<표 1> SCADA 키 관리 방안 비교

구분	SKE[1]	SKMA[2]	IBCKM[5]	Proposed
key management	SKC/PKC	SKC	SKC/IBC	SKC/IBC
LTK agreement	No	No	No	Yes
session key replacement	LTK	LTK	LTK	LTK
LTK update	No	No	sender update	IBC update
high-cost computation	paring:No MTP Hash:No	paring:No MTP Hash:No	paring:Yes MTP Hash:Yes	paring:No MTP Hash:No
key recovery	No	No	Yes	Yes
computation cost(# SPM)	-	-	10	4

LTK(장기키)의 갱신은 갱신 주기가 길고 키 갱신에 필요한 상호 인증 과정이 필요하므로 인증 기능이 없는 SKC보다는 송수신자 인증이 가능한 IBC 암호 방식을 사용하는 것이 효과적이다. 본 논문에서 제안한 방식에서는 고비용의 페어링 연산이 필요 없는 IBC 공개키 암호 알고리즘을 사용하므로 상대방의 공개키 인증 과정이나 폐기 관리가 필요 없게 된다. 그리고, 매 세션마다 새로운 공유 세션키가 필요하므로 합의된 장기키를 이용해서 고속 SKC 암호 기법으로 세션키를 교환하는게 저속 PKC 암호 기법을 사용하는 것보다 효율적이다. IBCKM의 문제점은 암호/복호시에 고비용의 페어링 연산의 사용과 키 발급 단계와 암호/복호 단계에서 사용자 ID로부터 공개키를 생성하기 위해 MTP 해시함수의 사용이다. <표 1>의 computation cost 항목은 고비용 연산인 페어링, MTP 해시, 그리고 SPM만을 고려해서 비교한 것으로서 제안한 방법이 IBCKM보다 약 61 % 정도 연산 시간을 향상시킬수 있음을 알 수 있다. IBCKM의 다른 문제점은 키 교환 프로토콜을 제안한게 아니고 기존의 IBC 암호/복호 알고리즘을 그대로 키 교환에 사용하였다. 따라서, 송신자가 장기키를 독자적으로 생성한 후 암호화해서 수신자에게 전송하므로 쌍방간에 합의해서 공유키를 생성하지는 못 한다.

다음 <표 2>는 고비용 SPM 연산을 기준으로 제안된 키 합의 프로토콜과 관련된 ID기반의 인증 키 합의 프로토콜의 계산 성능 비교를 보여준다. <표 2>에서 기존 방법들과 비교해볼 때 제안된 프로토콜은 ESRR을 포함해서 기본적으로 준수해야 할 보안 속성들을 모두 준수하고 있으며, 계산 효율성 면에서 [6-9]의 방법들에 비해 각각 약 20%, 66.7%, 33.3%, 42.8% 이상 연산 시간을 향상시킬 수 있다.

<표 2> IBC상에서 인증 키 합의 프로토콜 비교

protocol	essential security properties	computation cost(# SPM)	improvement
Xie-Wang[6]	KKS,BIS,PFS,KFS, NKC,KIA,UKSA	5	20%
Vivek et al.[7]	KKS,BIS,PFS,KFS, NKC,KIA,UKSA	12	66.7%
Sun et al.[8]	KKS,BIS,PFS,KFS, NKC,KIA,UKSA, ESRR	6	33.3%
Ni et al.[9]	KKS,BIS,PFS,KFS, NKC,KIA,UKSA, ESRR	7	42.8%
Proposed	KKS,BIS,PFS,KFS, NKC,KIA,UKSA, ESRR	4	-

Ⅴ. 결 론

국가 기반 시설에 대한 실시간 제어를 위한 SCADA 시스템은 특성상 강력한 관리 기능이 있는 폐쇄적인 그룹망에서 운영된다. 본 논문에서는 SCADA 시스템의 특성을 고려하여 각 장치에서 암호 통신에 이용되는 세션키의 공유에 필요한 장기키를 IBC를 통해서 생성 및 교환하는 안전하고 효율적인 키 관리 방안을 제안하였다. 제안된 방법은 고비용의 페어링 연산과 MTP 해시 함수를 포함하지 않는 IBC를 기반으로 기존의 방법들보다 저비용으로 키 합의가 가능하므로 컴퓨팅 파워가 부족한 원격 단말에 효율적으로 적용 가능하다. 산업 제어 시스템에 대한 보안 위협에 대응하기 위해 시스템의 운영 소프트웨어나 하드웨어 개선 연구, 네트워크 개선 연구 등이 있지만 확실한 해결 방안은 아니다. 기술적 측면의 보안과 더불어 관리적 보안 정책을 적절하게 수립한다면 저비용으로 시스템의 주요 자산에 대한 보안 취약성을 제거하여 안전한 보안 환경을 구축할 수 있다.

참고문헌

[1] B. Cheryl, G. Donald, N. William and T. Mark, “Key management for SCADA,” Sandia National Laboratory, Mar. 2002.

[2] R. Dawson, C. Boyd, E. Dawson, and J. Nieto, “SKMA: A Key Management Architecture for SCADA Systems,” 4th Australasian Information Secur- ity Workshop, 2006.

[3] A. Rezai, P. Keshavarzi, and Z. Moravej, “Key management issue in SCADA networks: a review,” Eng. Sci. Technol., Int. J., 20(1), 2017, pp.354-363.

[4] L. Martirano, M. Kermani, F. Manzo, A. Bayatma- koo, and U. Graselli, “Implementation of SCADA Systems for a Real Microgrid Lab Testbed,” In Proceedings of 2019 IEEE Milan PowerTech, Italy, 2019, pp.1–6.

[5] 오두환 · 최두식 · 나은성 · 김상철 · 하재철, “ID기반 암호 기법을 이용한 SCADA 시스템에서 비밀키 관리 및 복구 방안,” 정보보호학회논문지, 제2권, 제3호, 2012, pp.427-437.

[6] M. Xie and L. Wang, “One-round identity-based key exchange with perfect forward security,” Inf. Process. Lett. 2012, 112(14), pp.587–591.

[7] S.S. Vivek, S.S.D. Selvi, L.R. Venkatesan, and C.P. Rangan, Proceedings of the ProvSec, in: LNCS, vol. 8209, Springer-Verlag, 2013, pp.38–58

[8] H. Sun, Q. Wen, H. Zhang, and Z. Jin, “A strongly secure identity-based authenticated key agree- ment protocol without pairings under the GDH assumption,” Secur Comm. Netw., 2015, 8(17), pp.3167-3179.

[9] L. Ni, G. Chen, J. Li, and Y. Hao, “Strongly secure identity-based authenticated key agree-ment protocols without bilinear pairings,” Inform Sciences., 2016, Vol.367, pp.176-193.

[10] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” ACM Trans. Comput. Syst., 1990, Vol.8, No.1, pp.18-36.

[11] 오중타잉 · 최태영, “A Robust Three-Factor User Authentication Scheme based on Elliptic Curve Cryptography and Fuzzy Extractor,” 정보과학회논문지, 제46권, 제6호, 2019, pp.587-597.

[12] J. Gao, J. Liu, and B. Rajan, “SCADA communication and security issues,” Security and Communication Networks, 2014, 7(1),

pp.175-194.

[13] A. Rezai, P. Keshvarzi, and Z. Moravej, "Secure SCADA communication by using a modified key management scheme," ISA Trans., 2013, 52(4), pp.517-524.

[14] D. Boneh and M. Franklin, "Identity-Based encryption from the Weil pairing," SIAM Journal of Computing, 2003, Vol.32, No.3, pp.586-615.

[15] B. Lynn, "Authenticated Identity-Based Encryption," available at http://eprint.iacr.org/2002/72, 2002.

[16] D.H. Choi, H.M. Kim, D.H. Won, and S.J. Kim, "Advanced Key Management Architecture for Secure SCADA Communications," IEEE Trans. Power Deliv., 2009, 24(3), pp.1154-1163

[17] L. Dang, J. Xu, X. Cao, H. Li, J. Chen, Y. Zhang, and X. Fu, "Efficient identity-based authenticated key agreement protocol with provable security for vehicular ad hoc networks," Int. J. Distrib. Sensor Netw., 2018, 14(4), pp.1-17.

■ 저자소개 ■

이 건 직
(Lee Keonjik)

2020년 현재
대구대학교 자유전공학부 교수
2001년 8월 경북대학교 컴퓨터공학과(공학박사)

관심분야 : Computer Arithmetic Algorithm, Parallel Processing, Information Security
E-mail : othiin@naver.com

논문접수일 : 2020년 11월 22일
수 정 일 : 2020년 12월 9일
게재확정일 : 2020년 12월 15일