논문 17-42-12-19 The Journal of Korean Institute of Communications and Information Sciences '17-12 Vol.42 No.12
https://doi.org/10.7840/kics.2017.42.12.2326

스마트미터의 안전한 소프트웨어 업데이트 기술개발

명 노 길◆, 박 병 석*, 은 창 수°

Development of Secure Software Update Technology for Smart Meters

Nogil Myoung◆, Byeongseok Park*, Chungsoo Eun°

요 약

AMI 시스템이 대규모로 구축되고 운영됨에 따라 법정 계량기인 전력량계의 소프트웨어 오류 해결과 기능 확장에 필요한 원격 소프트웨어 업데이트 기술개발과 제도 도입이 시급한 상황이다. 이를 위해 본 논문에서는 법정 계량기의 소프트웨어 업데이트 요구사항과 IEC 국제표준 프로토콜인 DLMS에서 제공하는 소프트웨어 업데이트 프레임과 기능 요구사항을 우선적으로 분석하고, 자바카드에 구현한 전자봉인 애플릿 이용하여 안전한 소프트웨어 업데이트 절차를 제안 하였다. 제안한 방식의 기능검증과 현장 적용성을 확인하기 위해서 고압고객 50호 규모의 테스트베드를 구축하고 성능시험을 수행한 결과를 고찰한다.

Key Words : Advanced metering infrastructure, Smart meter, Device language message specification

ABSTRACT

Smart meters deployed in the fields often require updates for feature extension, bug-fixing and other operations. While remote software update is considered a more cost-effective alternative to on-site interventions, technologies and regulations are not ready in Korea. Therefore, technologies and legal framework for remote software update in AMI have become pressing issues in the Korean electric metering ecosystem. In this paper we analyze remote software update requirements for legally certified smart meters and important features of remote software update based on IEC metering standards known as DLMS. We also propose secure software update architecture and procedures using an e-Sea implemented in java card. Furthermore, operational validity of the proposed architecture and procedures was evaluated on a test-bed built for 50 households. Related results are presented and analyzed.

Ⅰ. 서 론

전력회사는 풍력, 태양광과 같은 신재생에너지원 사용 확대를 통한 CO_2 저감과 설비 운영효율화 및 전기에너지 사용 최적화를 위해 기존 전력망에 ICT(Information Communication Technology) 기술을 융합하는 스마트그리드를 적극 추진 중에 있다. 스마트그리드는 발전, 송·변전, 배전 및 판매 등의 분야별로 다수의 응용시스템이 존재하나, 그 중에서 AMI(Advanced Metering Infrastructure)는 스마트미터 또는 전자식 전력량계(이하 전력량계)를 통한 고객과의 접점을 제공하여 DR(Demand Response) 서비스를 가능하게 하므로, 핵심 어플리케이션으로 각광을 받고 있다. 이에 주요 선진국은 스마트그리드 구축을

◆ First Author : Korea Electric Power Research Institute, ng-myoung@kepco.co.kr, 정회원
° Corresponding Author : Chungnam National University, eun@cnu.ac.kr, 종신회원
* Korea Electric Power Research Institute, blue-grid@kepco.co.kr, 정회원
논문번호 : KICS2017-09-239, Received September 7, 2017; Revised December 3, 2017; Accepted December 6, 2017

위해 우선적으로 AMI를 설치 중에 있다. 국내의 경우 2022년까지 1.7조원을 투자하여 전국 2,250만호를 대상으로 AMI를 구축할 계획이었으나, 누진제 축소 정책과 전력 프로슈머 등장 활성화 등 에너지 신사업 출현 가속화를 위해 2020년까지 조기 구축 중에 있으며, 2017년 기준으로 445만호를 구축하여 운영 중에 있다[1].

AMI 핵심 기기는 전력량계로 기존 기계식 전력량계에 비해 다양한 계측·계량 항목, 세분화된 ToU (Time of Use), 다양한 유·무선 통신방식 지원, 개인정보 보호를 위한 보안기능, 신재생 분산전원 수용, 선불요금제 및 도전감시와 같은 각종 부가기능 때문에 소프트웨어(펌웨어)의 복잡성이 계속 증가하고 있다. 현장에 부설된 전력량계에서 비정상 동작이 발생할 때마다, 제조회사와의 시시비비는 물론 하자소송과 같은 첨예한 법적 다툼까지 발생하고 있다. 이러한 상황은 향후 전국에 2,250만개 이상의 전력량계가 전부 보급될 경우 더욱 더 가중될 것으로 예상된다.

현시점에서 소프트웨어 알고리즘 설계 또는 코딩 실수로 인한 하자가 자명하더라도 제조회사가 순순히 인정하기 어려운 점은 전국에 산재되어 부설된 전력량계로 출동하여 기존 전력량계를 철거하고, 형식승인을 신규로 획득한 전력량계로 재 부설하는 것은 전력량계 가격뿐만 아니라 교통비, 인건비 및 시공비 등에서 많은 추가비용이 발생하기 때문이다. 따라서 스마트폰처럼 신규 기능 추가 또는 기능 오류를 손쉽게 해결할 수 있는 법정 전력량계 대상의 원격 소프트웨어 업데이트 제도 도입이 시급한 시점이다. 유럽과 북미 일부 전력회사는 기본 계량기능을 제외한 나머지 응용기능에 대해서는 해당 법규와 절차를 준수하면서 원격 소프트웨어 업데이트를 시행중에 있다[2,3,13].

국내의 경우 전력량계는 계량법에 의해 관리되는 법정 계량장치로 계량법상 원격 소프트웨어 업데이트 기능을 불허하고 있지만, 국제표준 동향과 해외사례를 반영하여 국가기술표준원과 전력량계 형식승인기관을 중심으로 원격 소프트웨어 업데이트 제도 도입을 위한 절차, 통신 장애 시 복구방안 및 수신된 소프트웨어의 진위성과 무결성 검증방안에 대한 법·제도와 세부 적용 기준을 제정 중에 있다[4].

본 논문에서는 법정 전력량계를 대상으로 원격 소프트웨어 업데이트를 수행할 때 만족해야 하는 요구사항에 대해서 우선적으로 기술하고, 전력량계에서 사용하고 있는 IEC 62056 시리즈인 DLMS (Device Language Message Specification) 표준에서 대용량 데이터 전송 목적으로 정의한 이미지 분할 전송방식과 수신한 이미지의 무결성 검증에 필요한 이미지 전송 IC(Interface Class)에 대해서 설명한다. 사업화 모델을 발굴하기 위해서 이미지 분할 전송방식과 전자봉인을 이용한 소프트웨어 업데이트 방식을 제안하고, 현장 적용 가능성을 확인하기 위해 수행한 테스트베드 구축과 실증결과를 소개하며, 결론을 맺고자 한다.

Ⅱ. 기술개요

2.1 법정 계량기 소프트웨어 업데이트 요구사항

전력사용량 과금을 위해 고객에 부설하는 전력량계는 오차정밀도 등의 시험항목들을 형식승인기관으로부터 합격해야 사용할 수 있는 법정 계량장치로 분류된다. 법정 계량기는 국제법정계량기구(OIML, International Organization of Legal Metrology)에서 제정한 OIML D 31 규격을 만족해야 한다.

일반 요구사항으로는 소프트웨어의 식별기능, 알고리즘과 기능의 정확성, 소프트웨어 보호기능(조작방지) 및 하드웨어 특성 지원(불법기능 감지 및 내구성 보호 기능) 등이 있다. 식별기능은 식별자 또는 토큰을 이용하여 명확하게 구별해야 하고, 식별자와 식별방법을 형식승인 성적서에 명기해야 하며, 소프트웨어 자체의 무결성은 책크섬과 같은 알고리즘을 사용해야 한다.

특정 구성에 대한 요구사항으로는 전자식 장치와 하위부품의 분리 및 소프트웨어의 기능 분리를 요구하고 있다. 특히 소프트웨어는 형식승인기관이 소프트웨어의 기능 분리 여부를 판단할 수 있도록 법정기능과 비 법정기능으로 분리되어야 하며, 불분명할 경우에는 법정기능으로 간주한다. 여기서 법정기능은 계량·계측 알고리즘에 관한 소프트웨어 모듈로 프로그램, 서브루틴 및 객체 등을 포함하며 비 법정기능은 계량·계측 알고리즘 이외의 통신 및 부가서비스 등을 위한 소프트웨어 모듈을 의미한다[5].

소프트웨어 업데이트 방식은 그림 1과 같이 추적되는 업데이트(traced update) 방식과 검증된 업데이트(verified update) 방식이 있다. 전자는 전력량계가 설치되어 있는 현장에서 검정관이 입회하여 직접 전력량계의 물리적 봉인을 해제하고 소프트웨어를 업데이트하는 방식이다. 이에 반해 후자는 통신을 이용한 원격 업데이트를 의미하며, 수신한 신규 소프트웨어의 진위성과 무결성을 검증 후 소프트웨어 업데이트를 수행해야 한다. 이 때 소프트웨어 자체의 진위성과 무결성을 확인하는 방법으로 전자봉인을 사용할 수 있으며, 전자봉인은 비인가 된 신원의 접근 제한, 사용자 권한에 따른 접근제어, 해시 또는 전자서명을 이용

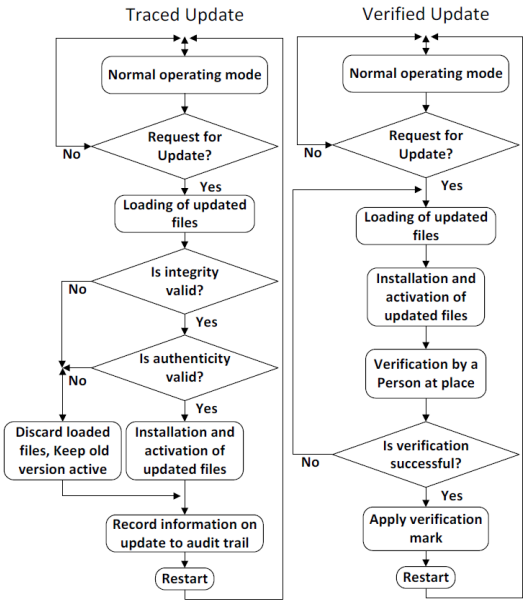

그림 1. 소프트웨어 업데이트 방식(좌 : 추적 업데이트, 우 : 검증된 업데이트)
Fig. 1. Methods of SW update(left : traced update, right : verified update)

한 소프트웨어의 무결성 검증과 위·변조 발생을 대비하여 분석용 감사 로그를 생성하는 기능을 지원해야 한다. 감사 로그에는 과정별 소프트웨어 업데이트 성공 및 실패 정보, 설치 전후의 소프트웨어 식별정보, 형식승인기관 정보 및 전력량계 형식승인 정보를 포함해야 한다.

2.2 소프트웨어 업데이트를 위한 DLMS 주요기능

2.2.1 DLMS 개요

DLMS는 IEC 62056 시리즈로 북미를 제외한 대부분의 국가에서 채택하여 사용하고 있는 전기, 가스, 수도 및 열량 등의 모든 에너지에 대한 데이터 모델링 방법과 모델링된 데이터를 전송하기 위한 통신구조와 메시징 프로토콜로 정의할 수 있다[7]. 기본 동작 구조는 server-client로 동작하고, 상호호환성 및 확장성을 위해서 계량 데이터를 포함한 모든 데이터는 사전에 정의된 IC를 이용하여 객체 모델링을 수행하며, 사용할 수 있는 하위 통신방식에 대한 제약은 없다.

계량기의 계량 데이터 및 파라미터를 포함한 모델링된 모든 데이터는 유일한 식별자인 OBIS(Object Identification System) 코드를 부여받는다[8]. 따라서 DLMS 클라이언트는(이하 클라이언트, DCU(Data Concentration Unit), AMI HE(Head-End) 또는 SMMS (Smart Meter Management System) 등) OBIS 코드를 이용하여 DLMS 서버로(이하 서버, 계량기 또는 전력량계) 동작하는 계량기의 계량데이터를 포함한 모든 데이터를 유일무이하게 식별하여 수집하거나, 파라미터를 변경할 수 있다.

데이터를 수집하거나 파라미터를 변경하기 위해서 클라이언트와 서버는 우선적으로 AA(Application Association)를 체결해야 한다. AA는 NS(No Security), LLS (Low Level Security) 또는 HLS와 (High Level Security) 같은 인증 방식과 APDU (Application Protocol Data Unit) 암호화 및 전자서명 적용 유무와 GET/SET/ACTION 등과 같은 xDLMS (Extended DLMS) 서비스 범위 등을 클라이언트와 서버 간에 협상하는 과정이다[7].

범용 DLMS 통신 프로파일로는 시리얼 통신 기반인 HDLC(High-level Data Link Control) 프로파일과 패킷 통신 기반인 UDP/IP 프로파일이 있으며, 특정 PHY/MAC 표준에 특화된 프로파일이 존재하는데, 대표적인 예로는 KS(Korean Industrial Standard) PLC(Power Line Communication) 프로파일, PRIME (Power-line Intelligent Metering Evolution) PLC 프로파일, G3 PLC 프로파일, RF mesh 프로파일 및 유·무선 M-bus 프로파일 등이 있다[7].

전 세계적으로 DLMS 전력량계가 AMI에 구축되고 운영됨에 따라 DLMS 표준은 전력회사의 다양한 현장요구 사항을 반영하기 위해 계속 진화하고 있다. 특히 AMI의 NAN(Neighborhood Area Network) 통신으로 많이 사용하고 있는 PLC 및 WiSUN(Wireless Smart Utility Network)과 같은 best effort 통신에서 발생하는 빈번한 통신장애와 낮은 유효속도를 고려하기 위해 데이터 압축 및 중복패킷 제거와 같은 패킷 사이즈 최소화 방법과 GET/SET/ACTION 서비스를 통합하여 한 번에 수행하는 ACCESS 서비스와 같은 단순화 기법들이 도입되었다[7,8]. 또한 사회적으로 요구하고 있는 개인정보 보호를 위해서 PKI(Public Key Infrastructure)를 도입하여 보안성을 대폭 강화 하였다.

2.2.2 HLS 기반의 AA

계량 데이터 수집, 파라미터 설정 및 소프트웨어 업데이트와 같은 다양한 서비스를 수행하기 위해서는 우선적으로 AA를 체결해야 한다. AA는 클라이언트와 서버 간의 인증과정과 xDLMS 서비스를 협상하는 과정으로 이분화 할 수 있다. NS로 불리는 첫 번째 인증 방식은 클라이언트와 서버의 SAP(Service Access Point)으로만 인증하는 방식이며, 두 번째 방식인 LLS

표 1. HLS 인증 메커니즘
Table 1. HLS authentication mechanisms

Authentication Mechanism	Pass1 Client→Server	Pass2 Client→Server	Pass3 Client→Server f(StoC)	Pass4 Server→Client f(CtoS)
	AARQ(Application Association Request)	AARE(Application Association Response)	COSEM-OPEN.request Reply to HLS Authentication	COSEM-OPEN.response Reply to HLS Authentication
Mechanism_ID(5) HLS GMAC	CtoS (Random String 8~64 Octets) System-title-C in calling-AP-title	StoC (Random String 8~64 Octets) System-title-S in responding-AP-title	SC ‖ IC ‖ GMAC (SC ‖ AK ‖ StoC)	SC ‖ IC ‖ GMAC (SC ‖ AK ‖ CtoS)
Mechanism_ID(6) HLS SHA 256			SHA-256(HLS-secret ‖ System-title-C ‖ System -title-S ‖ StoC ‖ CtoS)	SHA-256(HLS-secret ‖ System-title-S ‖ System -title-C ‖ CtoS ‖ StoC)
Mechanism_ID(7) HLS ECDSA	CtoS (Random String 32~64 Octets) System-title-C in calling-AP-title Cert-Sign-C in calling - AE-qualifier	StoC (Random String 32~64 Octets) System-title-S in responding-AP-title Cert-Sign-S in responding - AE-qualifier	ECDSA(System-title-C ‖ System-title-S ‖ StoC ‖ CtoS)	ECDSA(System-title-S ‖ System-title-C ‖ CtoS ‖ StoC)

는 SAP 이외에 클라이언트와 서버 간 사전에 공유된 비밀번호를 이용하여 인증하는 방식이다. 마지막 방식은 challenge(StoC, CtoS)를 이용하여 양방향 상호인증을 수행하는 HLS 방식이 있다.

그림 2는 보안성이 가장 우수한 HLS 기반의 AA 과정을 도시하였으며, 이후 계량데이터 수집 및 파라미터 설정과 같은 메시지 교환과정과 서비스를 종료할 때 필요한 AA 해지 과정을 함께 도시하였다. 표 1은 DLMS 표준에서 제공하는 HLS 인증 메커니즘 중 사용을 장려하고 있는 GMAC(Galois Message Authentication Code), SHA(Secure Hash Algorithm)-256 및 ECDSA(Elliptic Curve Digital Signature Algorithm) 방식을 이용한 AA 과정을 pass1 ~ pass4로 구분 및 요약하여 기술하였다[7]. 인증서 기반의 보안성이 가장 우수한 HLS ECDSA 방식의 세부 구현 내용은 3장에서 기술하기로 한다.

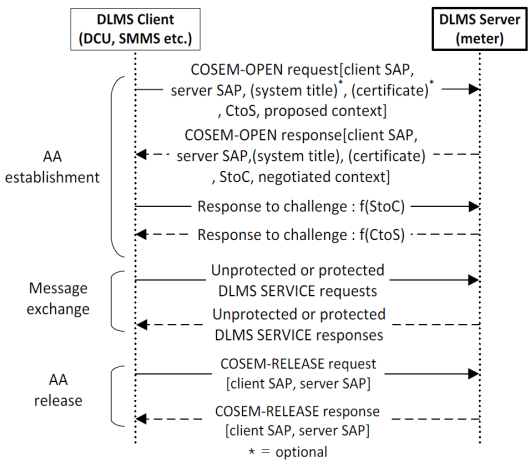

그림 2. DLMS 서비스 절차(HLS 상호인증 방식)
Fig. 2. DLMS service procedures(HLS mutual authentication)

2.2.3 Image transfer IC

초기 DLMS 표준에서는 계량기가 생성한 데이터를 효율적으로 수집하는 기능, 즉 검침 기능에만 중점을 두었으나, 최근에는 계량기의 다양한 기능을 최대한 활용하기 위해서 각종 파라미터 설정·제어 및 소프트웨어 업데이트와 같은 원격관리 기능 개선에 집중하고 있다. 단일 계량데이터(4bytes~12bytes)에 비해 상대적으로 크기가 큰 ToU(10Kbytes~30Kbytes) 및 소프트웨어(100Kbytes~500Kbytes) 데이터를 best - effort AMI 통신망을 이용하면서도 전송 신뢰성과 무결성을 제공하기 위해 표 2와 같은 image transfer IC를 정의했다[8]. Image transfer IC 각 항목에 대한 설명과 이를 이용한 소프트웨어 업데이트에 대한 세부 구현 내용은 3장에서 기술하기로 한다.

2.3 전자봉인 장치(e-Seal)

전자봉인 장치는 계량기로부터의 독립성을 보장받으며 업데이트 하고자 하는 소프트웨어의 진위성과 무결성을 검증하는 역할을 수행한다[9,14]. 전자봉인 장치로 안전한 메모리 영역제공, 물리적인 위변조 방어 기능 및 각종 암호 알고리즘 제공과 더불어 표준화된 하드웨어 인터페이스 및 프로토콜을 제공할 수 있는 스마트카드가 각광을 받고 있다. 스마트카드에서 진일보한 자바카드 플랫폼은 스마트카드와 같은 마이크로

표 2. 이미지 전송 인터페이스 클래스
Table 2. Image transfer IC

Image transfer	0...n	class_id=18, version=0			
Attributes	Data type	Min	Max	Def	Short name
1.logical_name(static)	octet-string				x
2.image_block_size(static)	double-long-unsigned				x+0x08
3.image_transferred_blocks_status(dynamic)	bit-string				x+0x10
4.image_first_not_transferred_block_number(dynamic)	double-long-unsigned				x+0x18
5.image_transfer_enabled(static)	boolean				x+0x20
6.image_transfer_status(dynamic)	enum				x+0x28
7.image_to_activate_information(dynamic)	array				x+0x30
Specific methods	Mandatory(m)/optional(o)				
1. image_transfer_initiate	m				x+0x40
2. image_block_transfer	m				x+0x48
3. image_verify	m				x+0x50
4. image_activate	m				x+0x58

컨트롤러, RAM(Random Access Memory), EEPROM(Electrically Erasable Programmable Read Only Memory), crypto 모듈 및 입출력 장치 등의 하드웨어에 chip OS(Operation System)로 구동된다. 부가적으로 JVM(Java Virtual Machine)과 자바 API (Application Programming Interface)를 통해 개방형 플랫폼을 지원하므로 응용 서비스 프로그램인 애플릿을 손쉽게 구현하여 사용할 수 있는 장점이 있다[10].

키 주입, 인증서 서명검증과 소프트웨어 진위성 및 무결성 검증에 필요한 기능을 지원하는 전자봉인 애플릿을 자바카드 플랫폼에 구현했으며, 자바카드는 전

표 3. 자바 카드 주요기능
Table 3. Main functions of a java card

Classification		Functions
HW		32bit CPU, 0.5Mbytes flash, 24K RAM
SW		Linux 2.6/applet for eSeal
Electrical/com' protocols		ISO/IEC 7816 series
UART serial speed		9.6kbps~230.4kbps
Security	Symmetric key algorithms	AES-GCM, AIRA-GCM(implemented as SW)
	Publick key algorithms	ECC-256(NIST P-256)
	Secure memory	Provided
	HASH/Random number gen'	SHA 256/CTR-DRBG

력량계로부터 전력을 공급받으며 전력량계와는 ISO/IEC 7816-3의 T=0 프로토콜로 동작하며, 패킷 송·수신을 위해서는 ISO/IEC 7816-4의 APDU 명령어를 정의하여 사용하였다. 표 3에는 본 연구에서 사용한 자바카드의 주요 사양과 자바카드에서 제공하는 암호 알고리즘을 명시했다.

Ⅲ. 소프트웨어 업데이트 기능구현 및 현장 성능시험

3.1 제안한 소프트웨어 업데이트 개요

DLMS 표준에서 정의된 이미지 분할 전송 기능과 자바카드에 구현한 전자봉인 애플릿을 활용하여 그림 3과 같은 소프트웨어 업데이트 절차를 제안했다. 기능 개선 또는 오류 해결 목적으로 소프트웨어 업데이트가 필요한 경우 신규 소프트웨어 코드에 대한 유효성 검증, 전체 이미지에 대한 해시값 계산과 상기 정보를 포함한 디지털 인증서를 생성 및 발급할 수 있는 PKI를 전제로 한다. 신규 소프트웨어 코드 자체의 유효성 검증과 이에 대한 디지털 인증서의 발급은 전력회사 또는 전력량계 제조회사에서도 수행할 수 있지만, 업무적 공정성과 독립성을 보장할 수 있는 전력량계 형식승인기관에서 수행하는 것으로 비즈니스 모델을 정의 했다.

전력회사는 SMMS를 이용해서 신규 소프트웨어 이미지를 DLMS 표준에서 정의한 image transfer IC를 이용하여 해당 전력량계로 분할 전송한다. 분할 전

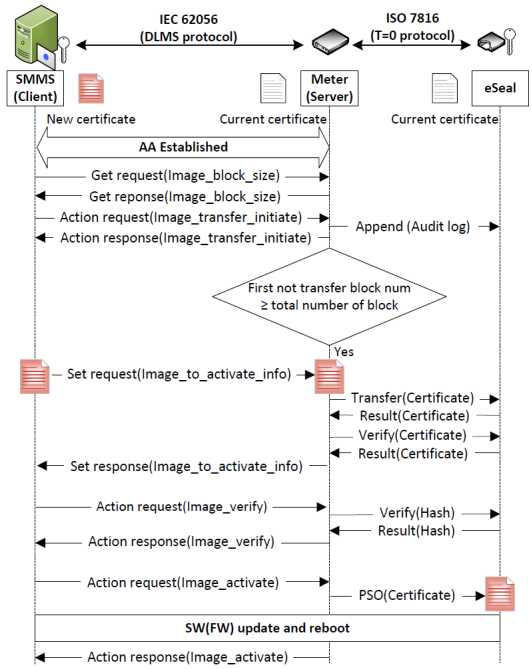

그림 3. 제안한 소프트웨어 업데이트 절차
Fig. 3. Proposed software update procedures

송 시점부터 감사로그를 생성하며 모든 블록이 정상적으로 전송된 후 추가적으로 전력량계 형식승인기관에서 발급한 신규 디지털 인증서를 해당 전력량계에 전송한다. 전력량계는 수신한 신규 디지털 인증서를 검증하기 위해서 자바카드의 전자봉인 애플릿에 전송하며, 전자봉인 애플릿은 수신한 신규 디지털 인증서의 서명검증을 통해 진위성 검증을 수행한 후 암호화된 해시값을 복호화 하여 메모리에 저장한다.

전력량계는 전자봉인 애플릿과 별도로 수신한 신규 소프트웨어 이미지에 대한 해시값을 자체적으로 계산 후 전자봉인 애플릿에 전송하여 검증을 요청한다. 검증의 핵심은 전력량계가 자체 계산한 해시값과 디지털 인증서로부터 복호화 한 해시값을 비교하여 신규 소프트웨어의 무결성을 확인 하는데 있다.

3.2 HLS ECDSDA기반의 AA 구현

DLMS 표준에서는 표 1과 같이 다양한 HLS 인증 방식을 제공하는데, 본 연구에서는 디지털 인증서를 사용하여 보안 강도가 가장 우수한 방식인 mechanism_ID(7) HLS ECDSA를 구현하였다. Pass1 단계에서 클라이언트는 서버에 AA 요청 패킷인 AARQ(Application Association Request) 패킷을 전송하고 pass2 단계에서는 이에 대한 응답으로 서버는 AARE(Application Association Response) 패킷을 클라이언트에 전송한다. 표 4는 DLMS 표준에서 정의하고 있는 AARQ/AARE 패킷구조와 실제 구현에 사용한 필드를 명시했다[7]. 구현에 사용한 선택사항 필드들은 HLS ECDSA 동작에 꼭 필요한 주요정보, 즉 전자서명 생성 및 검증과 관련한 필드로만 국한하여 적용하였다.

필수 필드인 DLMS 프로토콜 버전은 기본 설정 값인 1.0, application context name 필드는 LNR(Logical Name Referencing)과 ciphering을 지원하는 context_ID(3)인 logical_name_referencing_with_ciphering을 적용했다. 선택사항인 calling AP(Application Process) title 및 responding AP title 필드는 각각 클라이언트와 서버의 system title을 전송하는데 사용한다. 8bytes 크기를 갖는 system title은 3bytes 크기의 제조회사 ID와 5bytes 크기의 일련번호로 구성되며, 서버, 클라이언트 또는 third party와 같은 DLMS 엔터티를 유일무이하게 식별 하는 기능을 한다. Calling/responding AE(Application Entity) qualifier 필드는 각각 클라이언트와 서버의 디지털 인증서를 전달하는데 사용하며, calling/responding AE invocation ID 필드는 각각 세션 일련번호와 사용자 ID를 전송하는데 사용한다. 여기서 사용자 ID는 각각 클라이언트와 서버의 SAP을 의미한다.

Sender/responder ACSE(Association Control Service Element) requirements 필드는 표 1에 명시한 mechanism ID를 사용할 수 있게 하는 authentication functional unit(0x80)을 지정하여 사용했다. 사용한 mechanism name은 ECDSA 알고리즘을 사용하는 mechanism_ID (7)을 사용했으며, calling/responding authentication value 필드는 각각 32bytes 크기로 정의한 CtoS/StoC 값을 전송하는데 사용한다. 마지막 필드인 user information은 클라이언트와 서버 간 협상을 통해서 xDLMS 서비스 범위를 결정하는데 필요한 context parameter를 전송하는데 사용했다. AARE 패킷에서 필수 필드인 result와 result source diagnostic 값은 각각 정상적으로 동작하는 경우인 accepted(0×00)와 HLS success(0×0E)일 경우에만 HLS의 pass3 단계로 진입한다.

Pass3 단계는 클라이언트가 개인키로 ECDSA (System-title-C ‖ System-title-S ‖ StoC ‖ CtoS)를 연산하여 64bytes 크기의 서명값을 생성하고, ACTION.request 명령으로 association LN(Logical Name) IC의 첫 번째 메서드인 reply_to_HLS_authentication을 실행하여 생성한 서명값을 서버로 전

표 4. AARQ/AARE 패킷 구조
Table 4. AARQ/AARE packet structure

AARQ			AARE		
Name	Man.(M) Opt.(O)	value	Name	Man.(M) Opt.(O)	value
Protocol ver.	M	1.0	Protocol version	M	1.0
Application context name	M	Context_ID(3)	Application context name	M	Context_ID(3)
Called AP title	O	Not used	Result	M	0x00(accepted)
Called AE qualifier	O	Not used	Result source diagnostic	M	0x0E (HLS success)
Called AP invocation ID	O	Not used	Responding AP title	O	Server's system title
Called AE invocation ID	O	Not used	Responding AE qualifier	O	Sever's digital certificate
Calling AP title	O	Client's system title	Responding AP invocation ID	O	Session sequence number
Calling AE qualifier	O	Client's digital certificate	Responding AE invocation ID	O	Server user ID
Calling AP invocation ID	O	Session sequence number	Responder ACSE requirements	O	0x80
Calling AE invocation ID	O	Client user ID	Mechanism name	O	Mechanism_ID(7)
Sender ACSE requirements	O	0x80	Responding authentication value	O	32 bytes StoC
Mechanism name	O	Mechanism_ID(7)	Implementation information	O	Not used
Calling authentication value	O	32 bytes CtoS	User information	O	Association information
Implementation information	O	Not used	-	-	-
User information	O	Association information	-	-	-

송한다. 서버는 클라이언트의 공개키로 서명검증을 성공적으로 수행하면, pass4 단계로 진입한다.

Pass4 단계는 pass3 단계와 유사하게 서버는 개인키로 ECDSA(System-title-S ‖ System-title-C ‖ CtoS ‖ StoC)를 연산하여 64bytes 크기의 서명값을 생성하고, ACTION.response 명령을 이용하여 association LN IC의 첫 번째 메서드인 reply_to_HLS_authentication을 실행하여 생성한 서명값을 클라이언트로 전송한다. 클라이언트는 서버의 공개키로 서명검증을 성공적으로 완료할 경우 HLS ECDSA 상호인증은 완료된다. HLS ECDSA 상호인증과 더불어 마지막 필드인 user information을 사용하여 xDLMS 서비스 수준을 결정하는 xDLMS context parameter 협상도 병행되어 최종적으로 AA를 완료한다.

3.3 Image transfer IC를 이용한 소프트웨어 분할 전송 기능 구현

DLMS 표준에서 소프트웨어 또는 펌웨어와 같은 이미지의 블록 전송을 수행하기 위해서는 전송 주최가 되는 클라이언트의 이미지 전송기능이 활성화가 되어 있어야 한다. GET.request 명령으로 표 2의 image transfer IC의 다섯 번째 속성인 image_transfer_enabled 값을 읽어서 확인한다. 이미지 전송이 가능한 경우는 true(enabled)인 경우이며, 만약 false(disabled)일 경우에는 전력량계가 소프트웨어 업데이트를 진행 중에 있거나, AA 체결 전 또는 정전 등에 의한 전원공급이 없는 경우 등 이다.

다음으로 클라이언트는 GET.request (image_block_size) 명령으로 서버 자신이 최대로 받을 수 있

는 이미지 블록 크기를 요청하고, 그에 대한 응답을 받는다. 이는 클라이언트보다 상대적으로 프로세스 성능이 떨어지는 서버의 환경을 우선적으로 고려하기 위함이며, image transfer IC의 두 번째 속성인 image_block_size 값을 읽어서 확인한다. 이후 클라이언트는 브로드캐스트를 이용하여 각 서버를 초기화 시킨다. 초기화의 의미는 서버가 수신할 예정인 소프트웨어 이미지의 image_identifier와 총 image_size를 사전에 파악하고 상기 이미지를 수용할 수 있는 충분한 메모리를 확보하는 과정이며, ACTION.request (image_transfer_initiate) 명령으로 image transfer IC의 첫 번째 메서드인 image_transfer_initiate를 실행함으로써 수행한다. 현 시점부터 소프트웨어 업데이트가 종료될 때까지 서버와 전자봉인 장치 간에 발생하는 모든 이벤트는 분석 자료로 활용하기 위해서 텍스트 파일형태의 감사로그로 저장된다.

서버 초기화 후 클라이언트는 서버가 수용 할 수 있는 크기에 맞게 이미지를 블록으로 분할 생성 후 유니캐스트 또는 브로드캐스트로 전송한다. 즉 클라이언트는 ACTION.request 명령으로 image transfer IC의 두 번째 메서드인 image_block_transfer를 실행하여 image_block_number와 image_block_value를 서버에 전송한다. 서버는 개별 이미지 블록을 성공적으로 수신할 때 마다, image transfer IC의 두 번째 속성인 image_transferred_block_status의 값을 1 (transferred)로 변경함과 더불어 네 번째 속성인 image_first_not_transferred_block_number의 값을 최신 값으로 업데이트 한다.

만약 전송에 실패한 블록이 발생할 경우 해당 블록들만 서버로 재전송 할 수 있는 두 가지 알고리즘을 제공하는데, 첫 번째 방식은 GET.request 명령을 이용하여 image transfer IC의 세 번째 속성인 image_transferred_block_status의 값을 확인하고 0(not transferred)인 값을 갖고 있는 블록들 즉 실패한 블록들만 재전송 하는 방식이다. 두 번째 방식은 GET.request 명령을 이용하여 image transfer IC의 네 번째 속성인 image_first_not_transferred_block_status의 값을 확인하고, 상기 값이 전체 블록 수보다 적으면 전송에 실패한 블록이 있음을 인지하고 실패한 블록만 추출하여 재전송 하는 방식이다. 상기 두 가지 알고리즘은 전송에 실패한 모든 블록들이 성공할 때까지 반복적으로 수행한다. 본 연구에서 두 번째 방식인 image_first_not_transferred_block_status 방식을 사용하여 구현하였다.

클라이언트는 모든 블록이 성공적으로 전송되었음을 확인 후 소프트웨어 이미지의 무결성 검증에 필요한 정보를 담고 있는 신규 디지털 인증서를 서버로 전송한다. 전송 과정은 SET.request (image_to_activate_info) 명령을 이용하여 image transfer IC의 일곱 번째 속성인 image_to_activate_info를 업데이트 한다. Image_to_activate_info는 DLMS 표준에서 정의한 구조체 형식을 그대로 적용했으며 세부 구성 항목으로는 소프트웨어 이미지 크기, 디지털 인증서 바디 및 서명값으로 구성하여 아래와 같이 적용하였다[8].

```
Array   image_to_activate_info_element
Image_to_activate_info_element::=structure
{Image_to_activate_size:double-long-unsigned,
Image_to_activate_identification:octet-string,
Image_to_activate_signature:octet-string}
```

그림 3에서와 같이 신규 디지털 인증서를 수신한 서버는 TRANSFER(certificate) 명령으로 자바카드의 전자봉인 애플릿에 전달 후 VERIFY(certificate) 명령으로 신규 디지털 인증서의 서명 검증 등을 요청한다. 전자봉인 애플릿은 우선적으로 신규 디지털 인증서의 형식과 유효기간을 확인하고 공개키를 이용하여 신규 디지털 인증서의 서명을 검증한 후, 인증서 바디에 암호화되어 삽입된 해시값을 비밀키로 복호화 한다. 이후 수행결과를 RESULT(certificate) 명령으로 서버에 응답한다.

여기서 비밀키는 전력량계 형식승인을 발급할 때 자바카드의 안전한 메모리에 주입하는 것으로 가정하였다. 서명검증으로 진위성이 확인된 신규 디지털 인증서는 자바카드의 안전한 메모리에 임시로 저장되며, 신규 소프트웨어 이미지의 무결성 검증 후 소프트웨어 업데이트 이전에 수행하는 PSO(Perform Security Operation) 명령을 통해서 기존 디지털 인증서를 대체한다. 서버는 SET.response(image_to_activate_info) 명령을 이용하여 전자봉인 애플릿으로부터 수신한 RESULT(certificate)를 클라이언트에 전송하여 응답한다.

신규 디지털 인증서의 진위성을 정상적으로 확인한 경우, 클라이언트는 전송이 완료된 소프트웨어 이미지의 무결성을 검증하기 위해서 ACTION .request(image_verify) 명령으로 image transfer IC의 세 번째 메서드인 image_verify를 실행한다. 무결성 검증을 위해 서버는 클라이언트와 사전에 합의된 해시 알고리즘을 사용한다. 본 연구에서는 SHA256 알

고리즘을 사용했으며, 서버는 수신한 소프트웨어 이미지 전체에 대한 해시값을 계산하고, 계산한 해시값을 VERIFY(hash) 명령으로 전자봉인 애플릿에 신규 소프트웨어 무결성 검증을 요청한다. 전자봉인 애플릿은 서버가 계산하여 전송한 해시값과 디지털 인증서로부터 추출하여 복호화 된 해시값을 비교하여 무결성 이상 유무를 판단 후 RESULT(hash) 명령으로 서버에 응답한다. 서버는 수신한 RESULT(hash) 응답을 ACTION.response(image_verify) 명령을 이용하여 클라이언트로 전송한다.

현 단계까지 정상적으로 수행한 경우 소프트웨어 업데이트를 수행하기 위해 클라이언트는 ACTION. request(image_activate) 명령을 서버에 전송하는데 이는 image transfer IC의 네 번째 메서드인 image_activate를 실행함으로써 수행한다. 서버는 소프트웨어 업데이트를 최종적으로 시행하기 위해서 전자봉인 애플릿에 PSO(certificate) 명령을 전송하면, 전자봉인 애플릿은 신규 디지털 인증서의 서명검정을 재 수행하여 진위성을 다시 한 번 확인 후 기존 디지털 인증서를 신규 디지털 인증서로 교체한다. 서버는 기 수신한 소프트웨어 이미지를 이용하여 업데이트를 수행하고 재부팅 한다. 재부팅 후 서버는 재부팅 결과를 ACTION. response(image_activate) 명령으로 클라이언트에 응답함으로써 전자봉인을 이용한 안전한 소프트웨어 업데이트가 종료된다.

3.4 자바카드를 이용한 전자봉인(eSeal) 기능 구현

보안성과 확장성이 우수한 자바카드 플랫폼에 전자봉인 애플릿을 구현하였으며, 전자봉인 애플릿은 전력량계와 독립적으로 동작하며 신규 디지털 인증서의 서명검증과 업데이트하고자 하는 신규 소프트웨어의 무결성을 검증하는 역할을 수행한다. 전력량계와 자바카드 간은 ISO/IEC 7816에서 지원하는 master-slave 구조의 반 이중 통신 방식으로 동작한다. 그림 4는 자바카드를 전력량계에 삽입했을 때의 동작 절차를 보여준다. 자바카드 삽입을 감지한 전력량계는 reset 명령으로 자바카드 초기화를 수행한다.

이후 자바카드는 ATR(Answer to Reset) 명령을 전력량계에 전송하는데, ATR 명령에는 자바카드 및 데이터 전송 프로토콜에 관한 파라미터를 포함하고 있다. 전력량계는 PPS(Protocol Parameter Selection) 명령을 통해 상기 파라미터의 값을 변경할 수 있다. 본 연구에서는 백홀 통신인 LTE를 고려하여 ISO/IEC 7816 인터페이스의 통신 속도를 초기설정 속도 9.6kbps에서 230.4kbps로 변경하여 사용하였다. 전력량계는 PPS 응답명령을 수신 후 사용하고자 하는 애플릿을 선택하기 위해서 Select an eSeal 명령을 자바카드에 전송한다. 전자봉인 애플릿 선택 이후부터는 ISO/IEC 7816-3/4 표준의 T=0 프로토콜과 그림 5에 명시한 C-APDU(Command Application Protocol Data Unit) 패킷과 이에 대한 R-APDU(Response

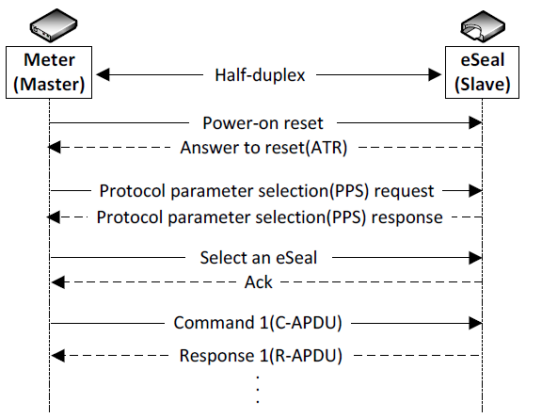

그림 4. 전력량계와 전자봉인 간 동작 절차
Fig. 4. Operational procedures between meter and eSeal

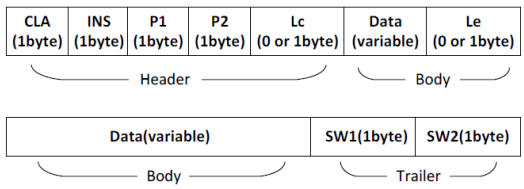

그림 5. C-APDU(상) 및 R-APDU(하) 패킷 구조
Fig. 5. C-APDU(upper) 및 R-APDU(lower) packet structure

표 5. C-APDU 및 R-APDU 필드 의미 및 크기
Table 5. Field meaning and size of C-APDU and R-APDU

Code	Definition	Length (byte)
CLA	Class of instruction	1
INS	Instruction code	1
P1,P2	Instruction parameter 1 or 2	1
Lc	Length of data in C-APDU	0 or 1
Data	Command or response data	0 or variable
Le	Length of expected data in R-APDU	0 or 1
SW1,SW2	Status word 1 or 2	1

Application Protocol Data Unit) 패킷으로 동작한다[11]. 본 연구에서 정의한 세부 필드 구성과 크기는 표 5에 정리하였다.

전자봉인 애플릿은 AID(Application ID)로 식별되며 AID는 5bytes 크기의 RID(Registered Application Identifier)와 최대 11bytes 크기를 갖는 PIX (Proprietary Application Identifier)로 구성할 수 있는데, 본 연구에서는 9bytes 크기의 AID (0xD4107600002000001)를 정의하여 사용하였다. 그림 3에 명시한 전력량계와 전자봉인 애플릿 간에 사용하는 APPEND/TRANSFER/VERIFY/PSO 명령 등에 대응되는 C-APDU 명령 필드를 ISO/IEC 7816 표준에 맞게 표 6과 같이 정의하여 구현 하였다.

표 6. C-APDU 필드 정의
Table 6. Field definition of C-APDU

Command	C-APDU field					
	CLA	INS	P1	P2	Lc	Data size(bytes)
SELECT	00	A4	00	00	09	AID(9)
APPEND	00	E2	01	1D	50	Log(80)
TRANSFER	00	DA	01	1A	E8	Certificate(398)
VERIFY	00	2A	00	00	E8	Certificate(398)
VERIFY	00	20	00	01	2A	Identifier(4)+ date/time(6)+ digest(32)
PSO	00	2A	80	00	E8	Certificate(398)
PUT1	00	24	02	91	20	ECDSA private key(32)
PUT2	00	24	02	92	41	ECDSA public key(64)

3.5 General ciphering 및 signing 구현

그림 3에서 명시한 클라이언트와 서버 구간 송수신하는 모든 패킷에 대해서는 DLMS 표준에 명시된 방법 중에 하나인 general ciphering과 general signing을 사용하여 기밀성과 무결성을 확보하였다. 그림 6 및 그림 7은 각각의 패킷 구성 필드를 보여 준다. General ciphering 필드는 트랜잭션을 구분하는 32bytes 크기의 트랜잭션 ID, 송·수신을 각각 구분하는 8bytes 크기의 시스템 타이틀, key info에는 대칭키 생성에 필요한 32bytes 크기의 share secret을 만들기 위한 static unified model(0e, 2s, ECC CDH)을 지정하여 사용했다[7].

Security suite는 AES(Advanced Encryption Standard)-GCM 128을 사용하지 않고, 국내 지능형

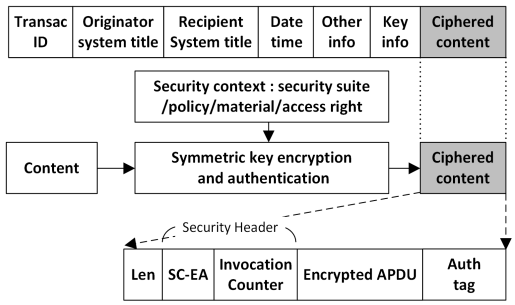

그림 6. General ciphering APDU 패킷 구조
Fig. 6. Structure of general ciphering APDU

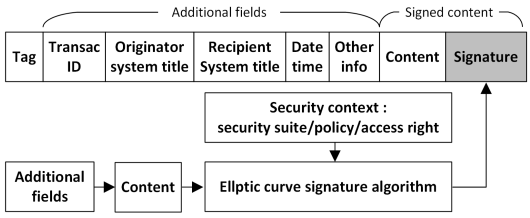

그림 7. General signing APDU 패킷 구조
Fig. 7. Structure of general signing APDU

전력망 보안에서 요구하는 KS 암·복호화 알고리즘인 ARIA(Academy Research Institute Agency)-GCM 128을 사용했으며, 보안헤더(SH)의 Security Control(SC)-EA(Encryption/Authentication)는 인증암호화 적용 유무를 의미하며, 4bytes 크기의 invocation counter는 구별자 역할을 한다. 16bytes 크기의 대칭키는 shared secret와 부가정보를 KDF(Key Derivation Function) 함수의 입력으로 사용하여 생성한다[15]. 그림 7과 같은 general signing을 사용할 경우 content에 대한 암호 또는 인증암호는 적용할 수 없고 무결성 검증만 제공하기 때문에, 기밀성을 확보하기 위해서는 추가적으로 general ciphering을 적용하는 것이 필요하다.

3.6 성능시험을 위한 테스트베드 구축

제안한 전자봉인 기반의 소프트웨어 업데이트 방법의 현장 적용성을 검증하고자 그림 8과 같은 구성도를 갖는 50호 규모의 테스트베드를 구축하였다. 전력량계는 자바카드와 ISO-7816 인터페이스 및 프로토콜로 통신하고, 외장형 LTE 모뎀 간에는 RS-422 인터페이스와 DLMS의 HDLC 프로파일로 통신한다. ISO-7816과 RS-422 인터페이스의 통신 속도는 모두 230.4Kbps를 사용했으며, LTE 모뎀은 전력량계로부터 수신한 시리얼 통신 기반의 HDLC 프레임을 IP 패

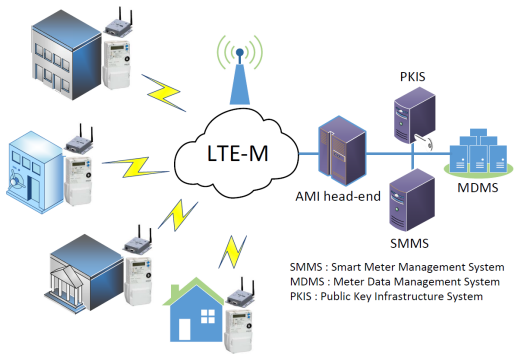

그림 8. 테스트베드 구성도
Fig. 8. Test-bed configuration

킷화 하는 캡슐화 또는 그 반대의 역할만 수행한다.

현장 성능시험에 사용한 전력량계 시작품의 MCU(Micro Controller Unit)는 OIML 규격을 만족하기 위해서 계량·계측용 연산 기능과 응용 서비스용 기능을 독립적으로 구현할 수 있는 Atmel사의 듀얼 코어 미터링 플랫폼인 SAM 4C32E 모델을 사용했다. 전자봉인 애플릿은 32bit MCU를 갖는 Infineon사의 SLM97 스마트카드 플랫폼에 구현했다. 전력량계 시작품의 소프트웨어 전체 이미지 크기는 198Kbytes이며 DLMS 표준에서 명시한 수신 가능한 최대 단일 패킷 크기는 2Kbytes 까지 사용할 수 있지만, TCP(Transmission Control Protocol) 패킷 크기를 고려하여 1,536bytes로 정의하여 사용하였다.

디지털 인증서는 X.509 ver.3을 준용하고 선택 필드를 최소화하여 398bytes 크기를 갖는다. 5분 주기의

그림 9. 미터(자바카드) 및 모뎀 현장 설치 사진
Fig. 9. Meter(with java card) and modem installation picture

LP(Load Profile)와 각종 이벤트가 발생할 경우 실시간 push 형태로 AMI HE로 전송하며, 필요시 on-demand 방식으로 추가 검침정보를 획득 및 파라미터를 설정할 수 있도록 AMI 시스템을 설계하고 구현하였다.

3.7 성능시험 결과 및 분석

구축한 테스트베드에서 그림 3에서 제안한 방식의 유효성을 검증하기 위해서 표 7과 같이 검증할 수 있는 최소 단위 기능으로 세분화 하였다. 각각의 기능 검증을 위해서 수집한 패킷의 형태는 그림 6에서 명시한 것과 같은 구조의 암호화된 패킷이며, 전력량계 시작품과 LTE 모뎀 구간에서 캡처했다. 표 7의 여러 항목 중 소프트웨어 이미지를 전송 후 검증하고 실행하는 중요 항목에 대해서만 암호화된 패킷을 복호화

표 7. 제안한 소프트웨어 업데이트 방식의 동작기능 검증
Table 7. Verification of proposed SW update

Seq	Names	Notes
1	SNRM	Data link layer connection
2	AARQ	Application layer connection
3	AARE	
4	f(StoC)	
5	f(CtoS)	
6	Get.request (image_transfer_enabled)	Image(SW/FW) transfer, verify and activate
7	Get.request (image_block_size)	
8	Action.request (image_transfer_initiate)	
9	Get.request (image_transfer_status)	
10	Action.request (image_block_transfer)	
11	Get.request (image_first_not_ trans_block_number)	
12	Set.request (image_to_activate_info)	
13	Action.request (image_verify)	
14	Get.request (image_transfer_status)	
15	Action.request (image_activate)	
16	Get.request (image_transfer_status)	

표 8. 테스트베드에서의 소프트웨어 업데이트 시간 측정 결과
Table 8. Measurement results of SW update time at the test-bed

Items	Measured time	Notes
AARQ	0.168s	pass1
AARE	0.253s	pass2
StoC	0.176s	pass3
CtS	0.225s	pass4
SW update1	116s	no ciphering/signing
SW update2	118s	ciphering only
SW update3	125s	ciphering/signing

후 xDLMS 서비스에 맞게 해석한 결과를 표 9에 요약정리 하였다. 그림 3에서 제안한 절차와 방식이 정상적으로 동작함을 세부 기능별로 검증하였다.

테스트베드 구축 후 안정적인 통신이 가능한 48호 고객을 대상으로 전력량계 시작품과 상위 검침·운영 시스템 간 AA 과정을 포함한 소프트웨어 업데이트 소요시간을 측정 후 평균값을 표 8에 명시하였다. 소프트웨어 업데이트 소요시간에는 계량법에서 요구하는 5초미만을 만족하는 평균 3.12초의 재부팅(계량·계측 중단) 시간을 포함하고 있다[16]. 소프트웨어 업데이트 과정은 암호화와 전자서명 적용 유무에 따른 3가지로 구분하였으며, 소프트웨어 업데이트 방식1은 암호화와 전자서명을 모두 적용하지 않은 방식이고, 소프트웨어 업데이트 방식2는 암호화만 적용했으며, 소프트웨어 업데이트 방식3은 암호화와 전자서명을 동시에 적용한 경우이다. 암호화와 전자서명을 동시에 적용할 경우 general signing을 우선 적용 후 general ciphering을 추가적으로 적용하였다.

측정된 응답시간은 ISO-7816과 RS-422 인터페이스의 통신 속도 및 전력량계 MCU의 프로세싱 능력에 따라 가변적일 것으로 예상되지만, 송·변전 및 배전 설비의 실시간 원격제어 감시와 성격이 다른 AMI 요구사항과 전력회사의 운영관점에서 문제가 없을 것으로 판단된다[9]. 제안한 전자봉인 기반의 소프트웨어 업데이트의 동작 유효성과 적용성을 현장실증을 통해서 검증하였으며, 향후 본격 시범사업을 통해 전사 확대할 수 있는 기반을 마련하였다.

Ⅳ. 결 론

본 논문에서는 IEC 62056 표준인 DLMS에서 정의한 이미지 분할 전송방식과 자바카드 플랫폼에 구현한 전자봉인 애플릿을 이용하여 법정 전력량계에 대한 소프트웨어 업데이트 방식과 절차를 제안하였다. 현장에서의 기능 검증과 운용성을 확인하기 위해서 LTE 통신네트워크 구축, 전력량계 시작품 제작 및 AMI HE 등의 상위 응용시스템을 개발하고 50호 규모의 테스트베드를 구축하였다.

테스트베드에서 실시간 원격검침 뿐만 아니라 제안한 전자봉인 기반의 소프트웨어 업데이트 동작 절차의 유효성과 사업화 적용 가능성을 검증하였다. 소프트웨어 업데이트 시간은 general ciphering과 general signing 등 보안 강도에 따라 다양하게 측정하여 비교하였다. 제안한 전자봉인 기반의 소프트웨어 업데이트 절차는 논의 중인 관련 법규와 제도가 제정된다면, 전력 현장에 적용하여 전력량계의 기능오류 해결 및 신규 기능 확장 등의 필요할 경우 유용하게 사용될 것으로 기대된다.

References

[1] N. Myoung, et al., “A study on AMI system of KEPCO,” *J. KICS*, vol. 35, no 8, pp. 1251-1258, 2010.

[2] J. Simmins, et al., *Remote meter FW update*(2010), Retrieved Jun. 25, 2017, from http://www.smartgrid.epri.com/UseCases/Remote%20Meter%20Update_ph2add.pdf

[3] Ron Amundson, et al., *Smart meter SDG&E key decision & lessons learned*, Retrieved May 20, 2017, from http://www1.itron.com/newsAndEvents/Documents/uc09/itr_017552.pdf

[4] I. S. Yang, *Smart metering technology trends*, Retrieved Oct. 25, 2017, from http://www.procon.cokr/pdf /2017%203 /11-1.pdf

[5] *General requirements for software controlled measuring instruments* (2008), Retrieved Jun. 1, 2017, from http://www.workgroups.oiml.org/tcsc/tc-05-sc-02/archives/D031-e08.pdf.

[6] O. Hersent, et al., *The Internet of things : key application and protocols*, eBook Ed., Wiley Telecom, 2012.

[7] DLMS UA, *Green book*(technical report), 8.0 Ed., Retrieved Oct., 20, 2017, from http://dlms.com/documents/Excerpt_GB8.pdf

[8] DLMS UA, *Blue book*(technical report), 12.0 Ed., Retrieved Oct., 20, 2017, from http://dlms.com/documents/Excerpt_BB12.pdf

표 9. 기능 검증을 위한 패킷 분석
Table 9. Packet analysis for verifications

Seq			Captured packets
13 (image_ verify)	Ciphered (Original)	Tx	7E A0 67 02 FF 29 DC 6D F1 E6 E6 00 DD 20 B4 7B E7 F5 67 A3 EB 6C 3B A8 60 73 1F C5 F1 C3 11 C6 A8 E5 70 0A E6 B6 4C C7 02 E2 40 FD 9E A1 08 4E 4A 43 00 00 00 00 01 08 4E 4A 43 12 A1 53 44 01 00 00 01 02 01 02 00 1E 3F 00 00 00 B6 B2 47 3D 82 ED 51 83 8B 03 11 34 C3 64 06 3C 56 E0 D3 FE 37 B3 97 75 32 62 C3 58 7E
		Rx	7E A0 5F 29 02 23 FC E3 9F E6 E7 00 DD 20 DD EB BD EB BB BA 81 BE 79 66 4A 18 85 99 1B 4B BE 75 C8 21 59 45 12 F6 3B DE 27 08 11 92 47 CD 08 4E 4A 43 12 A1 53 44 01 08 4E 4A 43 00 00 00 00 01 00 00 01 02 01 02 00 16 3F 00 00 00 B6 3C B5 8B D7 97 F6 E2 DA CB 29 41 A0 00 DD B2 A0 15 73 02 7E
	Plain (Parsed)	Tx	C3 01 81[action request normal] 00 12 00 00 2C 00 00 FF[class ID=18 image transfer, ver=0] 03 00[image verify]
		Rx	C7 01 81[action response normal] 00[success] 00[no return parameter]
14 (image_ transfer_ status)	Ciphered (Original)	Tx	7E A0 67 02 FF 29 FE 7D F3 E6 E6 00 DD 20 DD EB BD EB BB BA 81 BE 79 66 4A 18 85 99 1B 4B BE 75 C8 21 59 45 12 F6 3B DE 27 08 11 92 47 CD 08 4E 4A 43 00 00 00 00 01 08 4E 4A 43 12 A1 53 44 01 00 00 01 02 01 02 00 1E 3F 00 00 00 B7 D0 7E 2C 59 EC E2 3B 60 61 F4 2D 56 04 A8 70 4B 29 A4 46 D8 D9 A2 E5 C0 FF D8 4E 7E
		Rx	7E A0 60 29 02 23 1E D2 E5 E6 E7 00 DD 20 48 1C 28 2F F6 03 12 DB 3E B8 F0 91 47 C5 90 A0 AA 3A B0 FC AF 62 0E FA 18 78 CF 9C C1 09 CB 1E 08 4E 4A 43 12 A1 53 44 01 08 4E 4A 43 00 00 00 00 01 00 00 01 02 01 02 00 17 3F 00 00 00 B7 FD C2 21 CD 59 98 30 E3 7E 72 40 D6 86 DC 9F B9 D4 E8 E2 41 7E
	Plain (Parsed)	Tx	C0 01 81[get request normal] 00 12 00 00 2C 00 00 FF[class ID=18 image transfer, ver=0] 06 00[image transfer status]
			C4 01 81[get response normal] 00 16 03[image verification success]
15 (image_ activate)	Ciphered (Original)	Tx	7E A0 67 02 FF 29 10 0D FD E6 E6 00 DD 20 48 1C 28 2F F6 03 12 DB 3E B8 F0 91 47 C5 90 A0 AA 3A B0 FC AF 62 0E FA 18 78 CF 9C C1 09 CB 1E 08 4E 4A 43 00 00 00 00 01 08 4E 4A 43 12 A1 53 44 01 00 00 01 02 01 02 00 1E 3F 00 00 00 B8 97 D9 3B 1F 67 F5 60 B6 19 C5 C8 B6 E1 19 93 3E A1 4F F5 90 0F 28 B5 97 59 D6 62 7E
		Rx	7E A0 5F 29 02 23 30 83 93 E6 E7 00 DD 20 6B 66 59 73 DA 50 EF 73 7A 6D C2 DE 35 FF 8B F0 D6 B4 10 C9 6D 94 96 22 3C 23 17 B6 59 9D E6 25 08 4E 4A 43 12 A1 53 44 01 08 4E 4A 43 00 00 00 00 01 00 00 01 02 01 02 00 16 3F 00 00 00 B8 AE 44 CD 7E 6F 8F C3 7F 88 E9 9E 4B 4A C4 E4 49 04 13 2E 7E
	Plain (Parsed)	Tx	C3 01 81[action request normal] 00 12 00 00 2C 00 00 FF[class ID=18 image transfer, ver=0] 04[image activate] 00[no return parameter]
		Rx	C7 01 81[action response normal] 00[success] 00[no return parameter]
16 (image_ transfer_ status)	Ciphered (Original)	Tx	7E A0 67 02 FF 29 DC 6D F1 E6 E6 00 DD 20 9C B3 D5 9D 08 A5 93 9E 23 B4 42 1A C8 58 E9 66 3A 6C 2D 8A D9 F2 EB 44 D0 74 23 64 1F 60 90 E2 08 4E 4A 43 00 00 00 00 01 08 4E 4A 43 12 A1 53 44 01 00 00 01 02 01 02 00 1E 3F 00 00 00 06 4A 6E 45 62 5D BB ED 00 A9 7A 9B B5 17 CD 83 3F 02 36 B2 F1 FA 27 C1 0B 49 06 E8 7E
		Rx	7E A0 60 29 02 23 FC CE 21 E6 E7 00 DD 20 87 92 51 95 60 3F F3 7E A7 E4 7C FC 1D 33 FD 8A A1 24 5E 3F BD 6F EC B4 04 C4 36 0B 54 68 C7 E0 08 4E 4A 43 12 A1 53 44 01 08 4E 4A 43 00 00 00 00 01 00 00 01 02 01 02 00 17 3F 00 00 00 06 E0 FD 7A 94 58 FB 4F 52 05 B6 76 D8 0B F9 0B DA 1E 8F 10 42 7E
	Plain (Parsed)	Tx	C0 01 81[get request normal] 00 12 00 00 2C 00 00 FF[class ID=18 image transfer, ver=0] 06 00[image transfer status]
		Rx	C4 01 81[get response normal] 00 16 06[image activation success]

[9] *Expansion of crypto module usage for lubricator and LPG meter*, Retrieved Nov., 08, 2017, from http://www.motie.go.kr /common/download.do?fid=bbs&bbs_cd_n=6&bbs_seq_n=64480&file_seq_n=3.

[10] G. Selimis, et al., "Software and hardware issues in smart card technology," *IEEE Commun. Survey & Tuts.*, vol. 11, no. 3, pp. 143-152, 2009.

[11] W. Rankl, et al., *Smart card hand book*, 3rd Ed., John Wiley&Sons, 2003.

[12] N. Sulianovic, et al., "Requirements for communication infrastructure in smart grids," in *Proc. ENERGYCON 2014*, pp. 1492-1499, Dubrovnik, Croatia, May 2014.

[13] *S-EG-05-Specification for the approval of software controlled electricity and gas metering devices*, Retrieved Oct., 29, 2017, from http://www.ic.gc.ca/eic/site/mc-mc.nsf/eng/lm04533.html

[14] I. Yang, et al., "A study on smart card-based security mechanism of upgrades smart meter SW for secure deployment in SmartGrid," *JICS*, vol. 15, no. 2, pp. 129-142, 2014.

[15] *NIST SP 800-56A Rev3*, Retrieved Nov. 1, 2017, from http://csrc.nist.gov /csrc/media/ publications/sp /800-56a/rev-3/draft/documents/sp800-56ar3-draft.pdf.

[16] *E-meter technical regulation*, Retrieved Nov. 5, 2017, from http:// www.motie.go.kr/motie/ms/nt/gosi/bbs/bbsView.do?bbs_seq_n=62388&bbs_cd_n=5.

명 노 길 (Nogil Myoung)

2003년 : 충북대 전기전자공학부 졸업(학사)
2003년 : 한국무선관리사업단 근무
2006년 : KAIST 전기전자공학부 졸업(석사)
2014년~현재 : 충남대학교 전파정보통신공학과 박사과정
2006년~현재 : 한전전력연구원 근무(선임연구원)
<관심분야> 전력IoT, AMI/통합검침, 스마트미터링, 스마트그리드, 전력네트워크 시각동기화

박 병 석 (Byungseok Park)

1993년 : 한남대 전자공학과 졸업(학사)
1995년 : 한남대 전자공학과 졸업(석사)
2012년 : 한남대 전자정보통신공학과 졸업(박사)
1995년~현재 : 한전전력연구원 근무(책임연구원)
<관심분야> 스마트그리드, AMI, PLC, 딥러닝, 영상 인식

은 창 수 (Changsoo Eun)

1985년 : 서울대학교 전자공학과 졸업 (학사)
1987년 : 서울대학교 대학원 전자공학과 졸업 (석사)
1995년 : 텍사스 오스틴 주립대 대학원 전기 및 컴퓨터 공학과 졸업 (박사)
1987~1995년 : ㈜대우전자 중앙연구원 근무 (선임연구원)
1997년~현재 : 충남대학교 전파정보통신공학과 교수
<관심분야> 신호처리, 아날로그 회로설계, IoT, 무선 통신 기술