전력선 통신 시스템을 위한 인증 메커니즘

허준[1], 홍충선[1], 주성호[2], 임용훈[2], 이범석[2], 현덕화[2]
[1]경희대학교 컴퓨터공학과, [2]한국전력공사 전력연구원

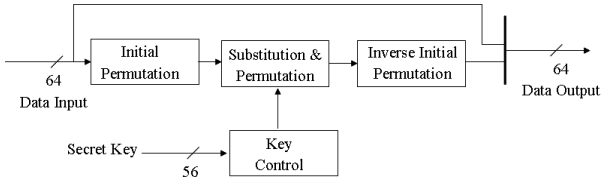
Authentication Mechanism for Power Line Communication System

Joon Heo[1], Choong Seon Hong[1], Sung Ho Ju[2], Yong Hun Lim[2], Bum Suk Lee[2], Duck Hwa Hyun[2]
[1]Department of Computer Engineering Kyung Hee University, [2]KEPRI KEPCO

Abstract - 지금까지 국내 PLC 기술의 경우 그룹식별자(GID)를 키로 사용하는 암호화/복호화를 통해 보안 기능을 제공하고 있다. 그러나, 이러한 방법의 경우 동일한 셀(Cell)에 존재하는 PLC장비들이 동일한 키를 사용하게 된다. 따라서, 공격자는 GID를 획득함으로써 동일한 셀에 존재하는 모든 장비들의 정보를 획득하거나 조작할 수 있다. 이러한 문제가 발생하는 가장 큰 원인은 두 가지로 정리할 수 있다. 첫째, PLC 장비간 인증절차를 거치지 않는 것이다. 둘째, 단순 식별자를 암호화를 위한 보안키로 사용한다는 것이다. 따라서 본 논문은 한국산업규격(KS X 4600-1)의 매체접근제어(MAC)계층에서 정의하고 있는 기술을 근거로 PLC 장비간 인증 및 보안키 생성 메커니즘을 제안하고, 그에 따른 프레임 형식 및 동작과정을 제안한다.

1. 서 론

현재 KS X 4600-1 규격[1]에서 PLC 네트워크에서의 데이터 통신에는 보안을 위하여 56-비트 DES[3][5] 방식의 암호화가 사용된다. 동일한 물리적 네트워크 상에 공존하는 셀들을 구분해 주는 역할을 하는 것은 46-비트 길이의 GID로서 이는 동일한 물리적 네트워크 내에 서로 다른 셀들이 공존할 수 있음을 의미하며, GID가 동일한 경우에만 스테이션들 간의 통신이 허용된다. 이와 같이 동일 셀에 속한 스테이션들은 암호화 키를 공유하며 56-비트 DES 방식의 암호화를 통해 데이터 통신에 대한 더욱 확고한 보안을 보장한다. 그러나, 이 방법을 사용하게 되면 새롭게 PLC망이 구성될 경우 관리자에 의한 수정이 모든 장비에 대하여 필요하게 되며, 같은 그룹에 존재하는 모든 장비들이 동일한 보안키를 사용한다면 외부로부터의 공격에 쉽게 노출될 수 있다.



<그림1> 64-비트 DES 알고리즘을 사용한 암호화

DES 알고리즘을 사용해 암호화 할 경우 그 절차는 <그림1>와 같이 설명할 수 있다. 그림에서 볼 수 있듯이 송신기의 가장 앞부분에 Encryption 모듈을 사용하여 데이터를 암호화한 후 전송하게 된다. DES 알고리즘의 경우 64비트의 값을 입력받아 실제 key를 사용하여 연산을 할 때는 56비트 기반으로 연산하고 있음을 알 수 있다. <그림1>과 같이 DES 알고리즘을 사용하여 암호화/복호화 과정을 수행하려고 하면 인가된 장비는 64비트의 보안키를 공유해야 한다.

또한, 장비간 인증과정을 통해 인가된 장비만이 네트워크를 구성하고 PLC 기술을 사용해 데이터를 전송할 수 있도록 해야 함에도 불구하고, 현재까지의 PLC 기술은 장비간 인증을 위한 어떠한 방법도 제시하지 않고 있다. 이러한 인증의 취약성은 국외 표준인 HomePlug[2], OPERA[7]등에서도 나타나고 있다[4]. 현재 국내에서도 UPLC 프로젝트[6]와 같은 사업을 통해 전력선 통신을 기존 인프라와 결합하여 자동화하는 노력이 활발히 진행되고 있으므로, 보안 취약점을 발견하고 이를 개선할 수 있는 기술을 개발하는 것이 매우 중요하다고 할 수 있다. 해결해야하는 보안 문제 중 장비간 인증은 가장 먼저 해결되어야 하는 부분이라고 할 수 있다.

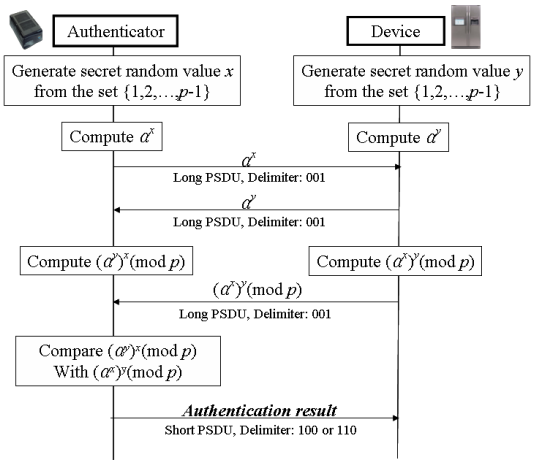
본 논문에서는 PLC 장비들이 능동적으로 인증절차를 수행하여 인가된 장비인지 아닌지를 확인할 수 있도록 한다. 또한, 이러한 인증절차를 위해 생성한 값들을 키 생성함수(KGF, Key Generation Function)의 입력 값으로 다시 사용하여 상호간의 보안키를 생성할 수 있도록 한다. 이렇게 생성된 보안키는 인증절차를 수행한 두 개의 장비간의 고유한 값으로써, 현재 GID값을 동일한 셀에서 공통적으로 사용하는 방법보다 매우 안전한 방법이라고 할 수 있다. 이를 위해 본 논문에서는 PLC 장비간 인증 메커니즘, 인증을 위한 프레임 구분자 및 형식의 정의, 기기인증 메커니즘의 예, 보안키 생성 메커니즘에 관하여 순차적으로 설명하고, 결론과 향후과제에 관하여 언급한다.

2. 제 안 사 항

2.1 PLC 장비간 인증 메커니즘

장비간 인증은 특정 기기에 한정되는 것이 아니지만 <그림2>에서는 인증장비(Authenticator)와 PLC 장비(Device)간을 예로 설명하고 있다. 설계된 메커니즘에서 인가된 모든 장비는 동일한 두 개의 요소 값(초기 값 a, modulo p)을 공유해야 한다. 장비간 인증 메커니즘을 절차는 다음과 같다.

가. Authenticator는 랜덤 값 x, Device는 랜덤 값 y를 정해진 범위 $\{1,2,...,p-1\}$안에서 선택한다.
나. 선택된 랜덤 값을 사용해 Authenticator는 a^x를, Device는 a^y를 계산하여 상대방에게 전송한다.
다. 자신이 생성한 값과 상대방으로부터 수신한 값을 통해 Authenticator는 $(a^y)^x$를, Device는 $(a^x)^y$를 생성한다.
라. Device가 $(a^x)^y$값을 Authenticator에게 전송하면, Authenticator는 자신이 생성한 $(a^y)^x$값과 비교하여 일치하면 '인증 성공' 메시지를 일치하지 않으면 '인증 실패' 메시지를 Device에 전송한다.



<그림2> 장비간 인증 메커니즘

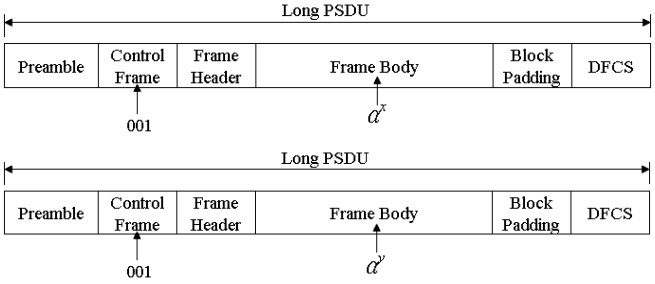
2.2 인증을 위한 프레임 구분자 및 형식의 정의

본 논문은 한국산업규격 KS X 4600-1[1]의 MAC 계층 규격을 기반으로 메커니즘을 제안하고 있다. 따라서, 전송되는 프레임이 인증을 위한 프레임인 것을 표시해야 하고, 또한 인증성공과 인증실패를 전달할 수 있는 프레임이 정의되어야 한다. 따라서, <표1> 같이 새로운 구분자를 정의한다. <표1>에서 음영처리된 3개의 구분자가 새롭게 정의한 프레임이며, 음영처리되지 않은 부분은 기존 규격에서 사용하고 있는 구분자이다. 새롭게 추가된 프레임의 구분자는 기존 규격에서 사용가능한 구분자로 기술되어 있는 부분을 사용한다.

<표1> 인증을 위한 구분자의 정의

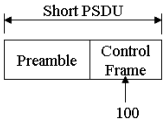
구분자 형태	정의	PSDU 길이
000	유니캐스트 데이터	Long PSDU
010	관리	Long PSDU
011	브로드캐스트 데이터	Long PSDU
101	응답	Short PSDU
001	인증 데이터	Long PSDU
100	인증 성공	Short PSDU
110	인증 실패	Short PSDU

<표1> 같이 구분자 '001' 은 장비간 인증을 위해 교환되는 프레임이다. 따라서, 인증 값이 전달되어야 하므로 PSDU는 Long PSDU가 되어야 한다. 이러한 방법으로 새롭게 정의된 프레임의 상세 내용은 <그림3>과 같다.

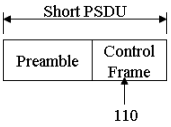


<그림3> 인증 데이터 전달을 위한 프레임의 사용

또한, <그림2>와 같은 인증절차에서 Authenticator는 Device에게 인증이 성공되었는지 아니면 인증이 실패되어 PLC 네트워크에 포함시킬 수 없는 지의 결과를 전송해 주어야 하며, 이를 위해 <그림4>와 같은 '인증 성공' 프레임 과 <그림5>와 같은 '인증 실패' 프레임을 정의하였다.



<그림4> 인증 성공 프레임



<그림5> 인증 실패 프레임

2.3 기기인증 메커니즘의 예

앞서 인가된 모든 장비는 두 개의 값(초기 값 a, modulo p)을 공유해야 함을 설명하였다. 예를 들어, 인가된 장비들이 modulo p를 공유하고 있다면, 초기 값 a에 따른 a^i(modulo p) (단, i=1,2,...,p−1) 값은 <표2>와 같은 결과를 나타내게 된다.

<표2> a^i(modulo p) (단, i=1,2,...,p-1)

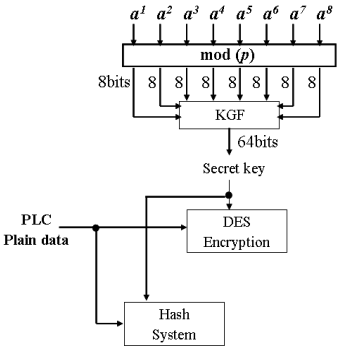
a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}
1	1	1	1	1	1	1	1	1	1
2	4	8	5	10	9	7	3	6	1
3	9	5	4	1	3	9	5	4	1
4	5	9	3	1	4	5	9	3	1
5	3	4	9	1	5	3	4	9	1
6	3	7	9	10	5	8	4	2	1
7	5	2	3	10	4	6	9	8	1
8	9	6	4	10	3	2	5	7	1
9	4	3	5	1	9	4	3	5	1
10	1	10	1	10	1	10	1	10	1

<그림2>에서 Authenticator와 Device가 초기값 a=2, p=11를 공유하고, 만약 Authenticator가 x=2, Device가 y=4를 랜덤하게 선택하여 a^x, a^y를 생성한 후 교환한다면, 최종적으로 Authenticator는 '3' 이라는 값을 비교해 Device를 인증하게 된다. 동일한 값을 통해 인증이 성공할 경우 인증 성공 프레임을 Device에게 전송하여 데이터 송수신이 가능함을 알려주게 되고, 인증에 실패할 경우 더 이상의 송수신을 하지 않는다.

$$(a^x)^y \pmod p = (2^2)^4 \pmod{11} = 2^8 \pmod{11}$$
$$2^1 \pmod{11} = 2$$
$$2^2 \pmod{11} = 4$$
$$2^3 \pmod{11} = 8$$
$$2^4 \pmod{11} = 5$$
$$2^5 \pmod{11} = 10$$
$$2^6 \pmod{11} = 9$$
$$2^7 \pmod{11} = 7$$
$$2^8 \pmod{11} = 3$$

2.4 보안키 생성 메커니즘

본 장에서는 앞서 설명한 인증메커니즘의 연산 값을 사용하여 보안키를 생성할 수 있는 메커니즘을 제안한다. 이 방법은 추가적인 절차 없이 보안키를 공유할 수 있으며 구체적인 메커니즘은 <그림6>과 나타낼 수 있다.



<그림6> 보안키 생성 메커니즘

앞서, 장비간 인증을 위해 두 장비는 랜덤 값을 생성하고 이를 통해 연산되고 교환된 결과를 통해 인증 절차를 수행하는 메커니즘을 설명하였다. 이 메커니즘에서 두 장비는 순차적인 계산 값을 가지게 된다. 따라서, 이러한 순차적 값들 중 아래와 같은 처음 8개의 값을 각 8비트의 값으로 변환하면 <그림6>과 같이 64비트의 보안키를 생성할 수 있게 된다.

a^1 (modulo p), a^2 (modulo p), a^3 (modulo p), a^4 (modulo p), a^5 (modulo p), a^6 (modulo p), a^7 (modulo p), a^8 (modulo p)

3. 결 론

전력선 통신기술은 광범위한 응용기술로 사용될 수 있는 장점을 가지고 있으나, 높은 에러율과 같은 물리적 한계와 보안상의 취약 부분을 해결해야 한다. 논문에서는 전력선 통신의 보안상 취약부분 중 장비간 인증을 위한 메커니즘을 제안하고, 이 값을 사용한 보안 키 생성 메커니즘에 관하여 설명하였다. 향후 과제로는 제안된 메커니즘을 실제 장비에 사용할 수 있도록 상세 기술을 설계하고 테스트하며, 국내 표준에 적합하도록 최적화해야 할 것이다.

[참 고 문 헌]

[1] KS X 4600-1 고속전력선통신 매체접근제어(MAC) 및 물리계층(PHY)

[2] HomePlug Specification Version 1.0, http://www.homeplug.org

[3] Man Young Rhee, "Internet Security", WILEY, 2003

[4] Joon Heo, Choong Seon Hong, Seong Ho Ju, Yong Hun Lim, Bum Suk Lee, Duck Hwa Hyun,"A Security Mechanism for Automation Control in PLC-based Networks",Proceedings of IEEE ISPLC2007, pp.466-470, Pisa, Italy, March 26-28 2007

[5] FIPS Publication 46-3, "Data Encryption standard (DES)," US DoC/NIST, 1999.

[6] UPLC(Ubiquitous Power Line Communication) project part of Korea Electric Power Corporation projects, http://www.kepri.re.kr/uplc

[7] Opera Alliance, "OPERA Specification: Technology," Jan. 2006.