Internet을 이용한 DLC시스템의 보안

Security consideration of DLC on Internet

이 재 인, 배 두 현, 송 오 영, 박 세 현
School of EE Engineering, Chung-Ang University, Seoul, Korea
Jae-In Lee, Du-Hyun Bae, Oh-Young Song, Se-Hyun Park

요 약

본 논문에서는 인터넷을 이용한 DLC 시스템의 보안 방안에 대한 연구와 제안된 시스템의 성능평가 내용을 기술한다. 이 논문에서는 DLC 시스템만이 아닌 다양한 제어 시스템에서의 보안 시스템 구성이 가능한 방법을 제안하였다. DLC 시스템의 보안 요소는 데이터 기밀성과 무결성, 부인 방지, 제어장치와 서버간의 상호인증이다. 데이터 기밀성과 무결성은 SSL을 통해서 해결하고 제어장치와 서버간의 상호인증은 PKI 서명을 통해서 해결하였다. 또한, DLC 시스템과 같은 close system에서 사설 PKI의 사용을 통한 시스템 성능향상에 대해 살펴본다.

I. 서론

국가 산업의 원동력으로써 다른 산업에 미치는 영향을 고려할 때, 안정적인 전력 공급의 중요성이 크게 부각되고 있다. 전력수요 증가에 따른 예비전력의 확보를 위해 발전소를 새로 건설하는 것은 비용적 측면에서 부담이 될 뿐 아니라, 각종 환경문제와 NYMBY(Not In My Back Yard)현상 등의 문제가 발생하여 새로운 전원공급 설비 확보에 어려움이 있다. 따라서 효율적인 전력 제어 및 부하 관리를 통해서 안정적인 전력 공급을 실시하는 방안의 필요성이 대두되었고 이를 위해 직접부하제어(DLC - Direct Load Control)의 개념이 도입되었다.

직접부하제어는 전력계통에 발생하는 피크전력 억제를 위해 전력회사와 수용가가 부하제어를 위한 시간 및 용량 등에 대해 약정을 체결하고 피크전력 발생시간대에 전력회사가 직접 수용가의 전력 사용 설비를 제어함으로써 전력회사의 피크상승에 따른 발전소 추가 건설투자 부담을 완화시키고 전체 산업에 안정적인 전력 공급을 가능하게 한다.

제어대상부하는 일반적으로 부하를 낮추거나 전원을 잠시 차단하여도 근무여건 및 생산 공정에 차질이 없는 전력사용설비를 중점적 대상으로 한다. 예를 들면, 패키지 에어컨, 냉동기, 공조 설비, 조명설비, 전기로, 기타 FAN 및 Pump 종류를 들 수 있다. 이러한 부하에 대한 직접부하제어를 통해서 2010년에는 165,000MW(최대부하의 15%)의 최대전력 억제효과를 가져올 수 있다.[11]

중앙 부하제어 서버와 수용가에 설치된 부하 제어 장치간의 통신을 이용하여 중앙 집중적이고 체계적인 관리 및 부하 제어가 가능하다. 기존에 적용된 통신방식은 단방향 통신 및 실시간 통신의 어려움이 있어서 새로운 통신방식의 필요성이 대두되었다. 최근 인터넷의 급속한 확산으로 대부분의 수용가에 고속통신 인프라가 확보되었고, 기설치 된 수용가의 인터넷 망을 활용할 수 있다는 측면에서 인터넷을 이용한 통신방식이 각광받고 있다. 인터넷을 이용한 통신 방식은 초기 설치비 및 유지비 감소로 경제성이 탁월하며, 1:N 통신이 가능하기 때문에 전력 관리 센터 1개소에서 다수의 수용가에 대한 감시 및 동시 부하제어가 가능하다는 장점이 있다.

눈부신 성장을 거듭하고 있는 인터넷을 통해서 원격으로 에너지 시스템을 감시하고 제어하여 전력수급의 안정화를 꾀하는 것은 신속성과 정확성의 측면에서 매우 중요한 기술이다. 하지만, 인터넷을 이용한 통신방식은 제어신호가 외부에 노출되는 것을 방지할 수 없고 해킹 등 잠재적인 위협으로부터의 보호가 되지 못하여 오히려 역효과를

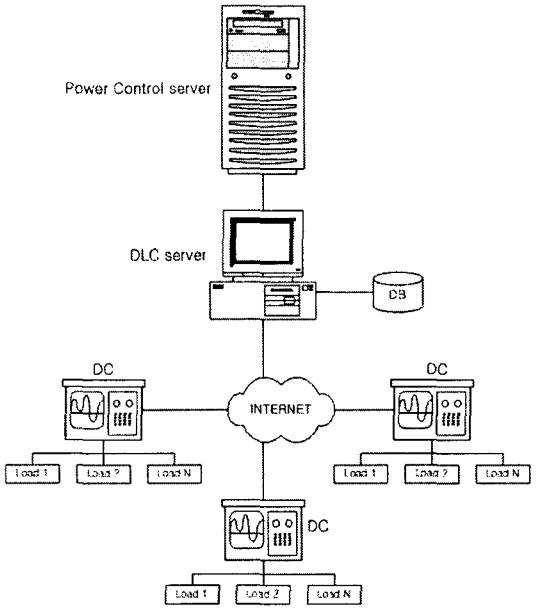
낳게 된다.

인터넷을 통한 에너지 제어 및 감시 서버와 수용가 제어 장치 간의 통신에서 야기될 수 있는 문제점은 기존의 인터넷의 보안 문제점과 같다. 즉 서버 측에서는 수용가 제어 장치에 대한 인증이 필요하고 수용가 제어 장치에서 보내오는 부하 정보에 대한 기밀성 및 무결성이 요구된다. 또한, 수용가 제어 장치는 에너지 제어 및 감시 서버에 대한 인증이 필요하고, 서버에서 오는 제어 신호에 대한 기밀성 및 무결성이 요구된다. PKI는 인터넷 상에서 발생할 수 있는 보안적 문제점에 대한 해법을 제시한다. PKI를 통해 인증, 부인방지, 기밀성, 무결성을 제공할 수 있다. 인증서를 통해서 사용자 및 장비에 대한 강력한 인증 메커니즘을 제공하고, server-side SSL(Secure Socket Layer)을 통해 서버와 장비 사이에 주고받는 메시지에 대한 기밀성 및 무결성을 제공할 수 있다[5][6][7].

본 논문에서는 기존 에너지 제어 및 감시 시스템을 수정하지 않고 인터넷을 이용한 직접부하제어에서의 보안적 문제점을 해결하기 위한 방안을 제시하였다. 기존 시스템을 수정하지 않는 방법을 제시함으로써, 현재 사용되고 있는 다른 시스템에서도 본 논문에서 제시한 방법과 같은 형태로 시스템을 구성하여 보안적 문제점을 해결하기 위해 활용할 수 있다. 또한 본 논문에서는 보안 요소에 따른 시스템 성능을 측정하였다. 성능 측정시 고려한 보안 요소는 다음과 같다. 첫째, 제어 서버와 수용가 제어 장치 사이에 발생하는 제어 신호 주기에 따른 전체 시스템 성능을 측정한다. 제어 신호 주기는 각 시스템간의 주고받는 메시지가 증가할 때마다 전체 시스템의 성능이 어떻게 변화하는가를 나타낸다. 둘째, SSL session 주기에 따른 전체 시스템 성능을 측정한다. SSL session은 시간이 지남에 따라 session의 보안성이 떨어지게 된다. 따라서 session에 사용되는 키를 일정시간이 지나면 교체를 해주어야 한다. SSL session을 다시 맺기 위해서는 상호 인증의 과정이 필요함으로 그에 따른 시스템 성능 저하를 야기한다. 셋째, 서버와 통신하는 클라이언트가 증가할 때마다의 성능을 측정한다.

이 후의 내용은 다음과 같다. 2절은 기존의 DLC 시스템에 대한 소개를 한다. 3절은 DLC 시스템의 보안적 취약점에 대해 살펴본다. 4절은 보안 서비스가 제공하기 위한 제안된 DLC 시스템에 대해서 소개한다. 5절에서는 구현 환경 및 테스트 환경에 대해 소개한다. 6절에서는 성능 측정 및 분석 결과를 기술한다. 마지막으로 7장에서 본 논문의 결론을 맺는다.

II. DLC 시스템



각 본 논문에서는 에너지 제어 및 감시 시스템 구성 요소에 대한 명칭을 다음과 같이 명명한다.

- DLC 시스템 : 에너지 제어 및 감시 시스템
- DLC server(DLCS) : 각 수용가에서 오는 전력 정보 메시지 모니터링 및 부하제어를 담당하는 서버
- Power Control Server(PCS) : DLC server에서 전송하는 부하제어 예측 자료를 바탕으로 필요한 절감전력용량과 이에 따른 절감비용을 DLC server에 전송
- Demand Controller(DC) : 수용가의 전력 정보를 수집하여 DLC server로 전송하고 DLC server로부터 오는 제어 신호에 따라 부하제어를 하는 장치

DLCS는 수용가에 대한 정보(수용가 위치, 부하용량, 제어 부하수, 약정 전력 등)를 DB에 저장한다. DLCS는 DC로부터 일정한 주기로 수용가 전력사용량 및 부하상태를 전송 받는다. DLCS는 제어 발생시 수용가별 절감전력, 절감비용을 산출하여 DB에 저장하고 필요에 따라 이 데이터를 PCS에 전송한다. PCS로부터 부하제어 명령을 받으면 제어할 전력량에 따라 제어 수용가를 선정하여 그에 따른 제어를 실시한다. 제어 발생시 부하제어

명령에 대한 DC에서의 'ACK' 신호를 바탕으로 부하제어 성공 여부를 판단하고 총 절감 전력, 절감 비용을 산출하여 데이터베이스에 저장하고 필요에 따라 이 데이터를 PCS에 전달한다. DLCS는 수시로 계산되는 일별, 시간별 수용가 부하제어 예측값을 데이터베이스에 저장하고 필요에 따라 PCS에 전송한다. DC의 주요 기능은 주기적으로 (초단위) 수용가 전력사용량 및 부하상태를 DLCS에 전송한다. DLCS에서 제어명령이 오면 그에 따른 부하제어를 실시하고 'ACK' 신호를 전송함으로써 부하제어가 성공하였음을 DLCS에게 알린다.

1) DLC 시스템 보안적 취약점

DLC 시스템은 DLCS와 DC간에 수용가 전력정보 및 부하제어명령을 인터넷을 통해서 전송하기 때문에 이러한 데이터들은 쉽게 공격당할 수 있다. 즉, 일반적인 TCP/IP network에 대한 공격이 DLC 시스템에도 적용될 수 있다. 공격자는 data packet에 대한 snooping을 통해 민감한 데이터에 대한 정보를 획득할 수 있고, 정보를 변경하거나 DLCS와 DC의 connection에 packet들을 끼워 넣을 수 있다. 공격자가 부하제어명령을 변경하여 실제적으로 부하제어가 일어나지 못하도록 한다면, 예비전력이 고갈되어 에너지 관리 시스템의 마비를 유발할 수 있는 위험성이 있다. 또한, DC에서 전송되는 수용가 전력사용량에 대한 변경은 전력회사와 수용가 사이의 분쟁을 일으키게 된다. 또한, 기존의 DLC 시스템에서는 DLCS와 DC사이의 인증 방법이 없다. DC 장비에 대한 인증을 하지 않게 되면 악의적으로 DC 장비에 대한 교체가 가능하여 부하제어명령을 실시하여도 데이터의 조작 등을 통해 실제적으로는 부하제어가 일어나지 않음에도 불구하고 DLCS에서는 부하제어가 실시되었다고 속일 수 있으므로 전체 시스템의 신뢰성이 떨어지게 된다. 또한, 패킷 재사용으로 인한 replay attack이 가능하고 따라서 DLCS에 부하를 주게 되어 DoS 공격을 당할 수 있다.

이러한 인터넷 상의 보안적 취약점을 해결하기 위해서, DLC 시스템은 데이터의 무결성과 기밀성, 디바이스 인증 및 DLCS 인증을 통한 상호인증이 지원되어야 하고 replay attack을 방지할 수 있는 방법이 모색되어야 한다. 특히, DC 장비에 대한 인증과 데이터 기밀성은 매우 중대한 보안 요소로서 취급되어야 한다.

III. DLC 시스템 보안

DLC system 보안은 데이터 무결성 및 기밀성과 DLCS와 DC간의 상호인증을 통해 가능하다. 데이터의 기밀성 및 무결성은 end-to-end security를 지원하는 방법이 선택되어야 하고, 상호인증은 인증 강도와 관리의 편의성을 고려하여 적절한 인증 방법이 선택되어야 한다.

1) DLC 시스템 인증

두 통신 당사자간에 인증 방법에는 ID/PW 방식과 MAC(Message Authentication Code) 인증, 그리고 공개키 서명을 통한 인증 방법이 있다.

ID/PW 방식은 패킷 스니핑(packet sniffing)을 통해서 ID/PW가 쉽게 노출될 수 있다. 이러한 문제점을 보안하기 위해 Secure Remote Password(SRP)와 Challenge Handshake Authentication Protocol(CHAP)와 같은 방법을 통해 패스워드가 노출되지 않게 할 수는 있으나, dictionary attack이나 brute-force attack등에 쉽게 깨질 수 있고 패스워드를 선정 및 관리하는 것도 쉽지 않다. 또한, ID/PW 방식은 인증만을 제공할 뿐 다른 보안적 요소들을 제공할 수 없다.. HMAC 인증 방식은 두 통신 당사자가 비밀키를 공유하고 있어서 message hash를 통해 인증을 실시한다. HMAC 인증 방식은 키 분배 및 관리에 어려움이 있고, 부인방지를 제공하지 못한다[5]. 공개키 서명을 통한 인증 방법은 서명자의 개인키를 이용하여 전자서명을 실시하므로 수신자가 받는 메시지의 변조나 위조를 방지할 수 있고 메시지의 송신자가 나중에 부인할 수 없도록 하는 기능을 제공한다. 그러나, PKI를 이용한 인증은 키의 길이가 길어짐에 따라 전자서명 및 전자서명 검증에 더 많은 시간이 필요하다[6].

DLC system에서는 데이터의 기밀성 및 무결성 그리고 상호 인증 및 부인방지 등의 보안적 요소가 필요하므로 PKI를 사용한 상호인증 방식을 채택하게 되었다. DLCS와 각각의 DC들은 자신만의 인증서를 가지고 있어서 인증에 필요한 메시지에 자신의 개인키로 서명을 하고 상대방이 서명 검증 및 인증서에 대한 검증을 하여 상호 인증을 수행한다.

2) 데이터 기밀성 및 무결성

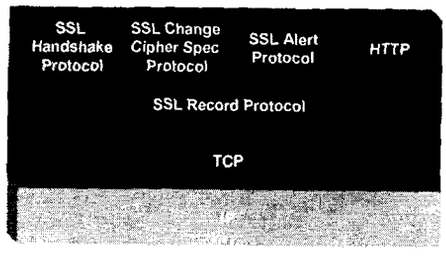
Privilege Verifier는데이터의 기밀성 및 무결성은 전송계층 보안을 통해 이루어지는 것과 데이터 자체에 대한 암호화 및 서명을 통하여 제공할 수 있다. 전송계층 보안은 IPSec(Internet Protocol Security)이나 SSL(Secure Socket Layer)을 통해서 제공될 수 있다[11][8][9].

IPSec은 시스템이 필요한 보안 프로토콜을 선택하게 하고, 서비스에 필요한 알고리즘을 결정하고, 요구된 서비스에 필요한 암호화 키를 제공함으로서 IP 계층에서 보안 서비스를 제공한다. 보안 서비스를 제공하기 위해 2가지 프로토콜이 제공되는데, Authentication Header(AH)와 Encapsulating Security Payload(ESP)가 있다[12][13].

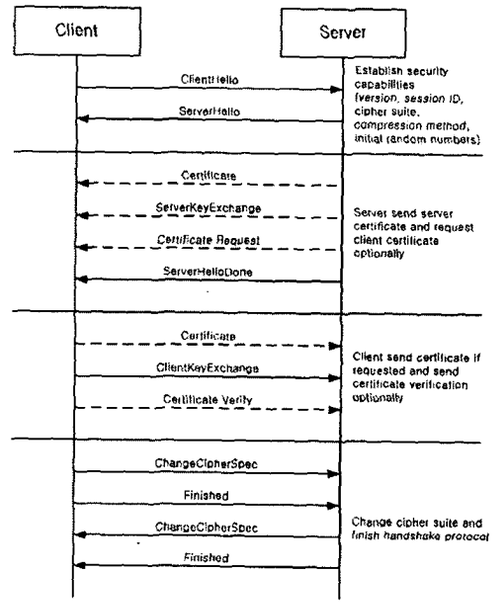
AH는 IP 패킷의 데이터 무결성과 인증을 제공한다. 데이터 무결성은 전송되는 패킷 내용이 불법적으로 변경되는 것을 방지한다. 인증은 end system 또는 네트워크 장치가 사용자나 응용 프로그램을 인증할 수 있도록 하고, 필요에 따라서 트래픽을 필터링할 수 있게 한다. 또한, address spoofing 공격이나 replay attack도 막을 수 있다[12]. ESP는 패킷의 기밀성을 제공한다. 데이터 기밀성은 전송되는 패킷 내용이 외부로 노출되는 것을 방지한다. 또한, AH에서 제공하는 보안 서비스를 제공할 수 있다[13].

SSL은 TCP를 이용하여 신뢰할 수 있는 end-to-end 보안 서비스를 제공하기 위해 설계되었다. SSL은 웹상에서 클라이언트와 서버사이의 인증 및 보안 채널을 형성하기 위해 가장 많이 사용되고 있는 프로토콜이다. SSL 3.0

SSL은 단일 프로토콜이 아니라 2개의 계층으로 된 프로토콜이다.



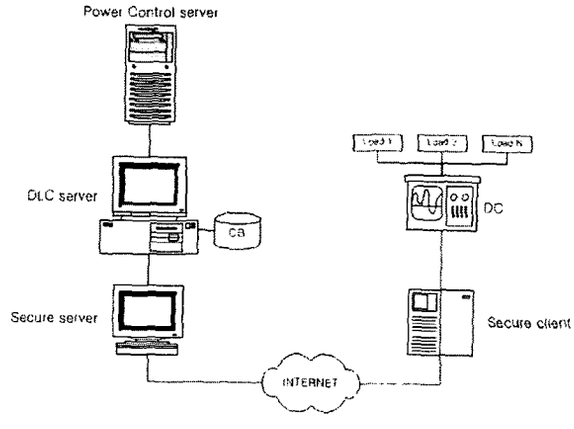
SSL Record Protocol은 다양한 상위 계층 프로토콜에 기본적인 보안 서비스를 제공한다. SSL Record Protocol은 SSL 연결을 위해 기밀성 및 메시지 무결성 서비스를 제공한다. SSL Alert Protocol은 하나의 메시지로 구성되어 있으며, 값 1을 갖는 한 바이트로 되어 있다. 이 메시지의 목적은 연결상에서 쓰이도록 암호화 단위를 갱신하여 미정인 상태를 현 상태로 복사하게 하는 것이다. SSL Alert Protocol은 대등한 개체에게 SSL 관련 경고를 전달하기 위해 상요된다. SSL을 사용하는 다른 응용 프로그램과 마찬가지로 현재 상태에 의해서 지정된 바와 같이 경고 메시지는 압축되고, 암호화된다. SSL Handshake Protocol은 서버와 클라이언트 간에 인증을 제공하고, 암호와 MAC 알고리즘을 교환한다. SSL Handshake Protocol은 클라이언트와 서버에 의해 교환된 메시지의 연속으로 이루어진다[8][5][6][7].



본 논문에서 데이터의 기밀성과 무결성을 제공하기 위해서 SSL을 사용하였다.

IV. DLC 보안시스템 제안

본 이미 활용되고 있는 DLC 시스템에 대한 변경 없이 보안 서비스를 확립하기 위해 본 논문에서는 다음과 같은 시스템을 제안하였다.



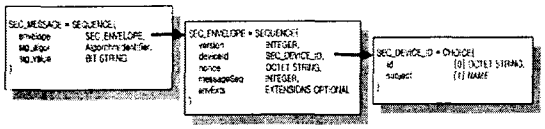
Secure client(이하 SC)의 기능은 다음과 같다.

- Secure server 인증
- Secure server와 SSL connection을 통해 data 송수신
- DC에서 DLCS로 보내는 데이터를 SSL 연결을 통해 Secure server로 송신

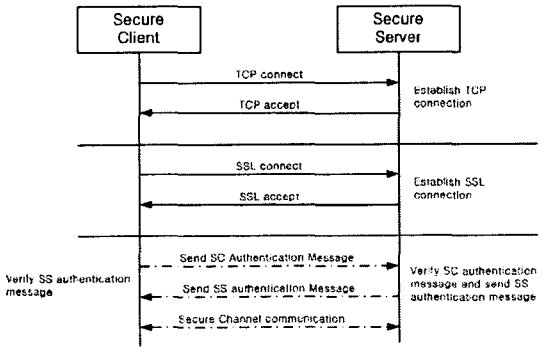
Secure server(이하 SS)의 기능은 다음과 같다.

- Secure client 인증
- SSL session 관리
- Secure client와 SSL connection을 통해 data 송수신
- DC에서 Secure client를 통해 전송된 데이터를 DLCS로 전송

SC는 전원이 켜지면 시스템 초기화 후 SS로 TCP 접속을 시도한다. TCP 접속이 완료되면 SS와 SSL connection을 시도한다. SSL connection이 완료되면 SC는 인증 메시지를 생성하여 서버에게 보낸다. 인증 메시지의 포맷은 그림과 같다. nonce와 메시지 번호(messageSeq)를 통해서 replay attack을 방지한다.



메시지를 생성하여 ASN.1 encoding을 하여 SSL session을 통해 전송하고 SS에서는 인증 메시지를 ASN.1 decoding 후에 인증 메시지 서명 검증을 실시한다. 서명이 유효하면, SS는 SC의 인증서에 대한 검증을 실시한다. SC의 인증서 검증이 유효하면 SS는 자신의 인증 메시지를 생성하여 SC에게 보낸다. SC에서는 SS에서와 같이 인증 메시지 서명 검증 및 SS의 인증서 검증을 실시한다. 상호 인증이 완료되면 DC와 DLCS 사이의 전송되는 데이터는 SS와 SC사이의 미리 맺어 있는 SSL session을 통해 전송되어 진다.

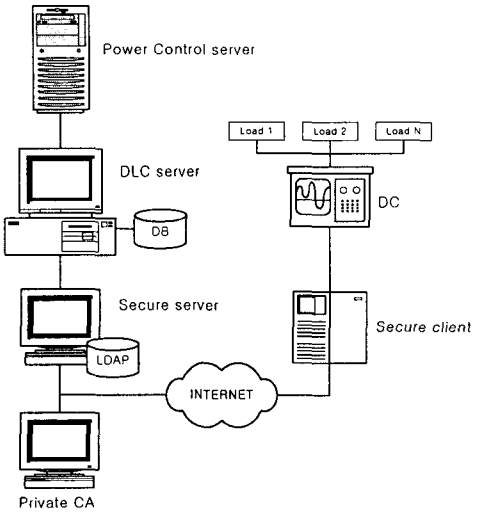


인증을 수행하기 위해서는 메시지에 대한 서명 검증과 서명에 사용된 인증서에 대한 검증을 해야 한다. 메시지에 대한 서명 검증은 서명자의 인증서에 있는 공개키를 추출하여 메시지 원문을 hash한 후 동봉된 서명된 값을 공개키 값으로 복호화하여 hash된 값을 비교해 봄으로써 서명자가 공개키와 쌍을 이루는 개인키를 가지고 있다고 판단할 수 있다. 그러나, 서명에 대한 검증만으로는 인증을 할 수 없다. 서명에 사용된 인증서를 검증해야 하는데, 인증서 검증에는 인증서의 상태 검증 및 인증서 chain에 대한 검증, 인증서 정책맵핑 및 정책검증이 있고 이 3단계의 검증이 모두 유효해야 비로소 인증서가 유효하다고 판단할 수 있다.

인증서에 대한 검증은 SS에게 많은 컴퓨팅 연산을 사용하게 하여 부하를 가중시킨다. 특히, 인증서에 대한 chain이 많아지게 되면 전체 인증서 chain에 대한 서명 검증을 실시하게 되고 또한 CRL 검증 횟수도 많아지게 되어 서명 검증에 대한 부담이 더욱 증가하게 된다. 인증서에 대한 검증은 OCSP[2]나 SCVP[3]를 통해 검증을 수행할 수도 있으나, DLC system은 폐쇄된(closed) 시스템이므로 공인 인증서를 사용하여 SS에 대한 부담을 가중시킬 필요는 없다.

사설 CA를 운영하면 인증서 chain의 수도 감소하고 SS에서 SC의 인증서의 상태에 대한 검증을 할 필요도 없게 된다. 사설 CA는 DC의 인증서를 발급하고 그 인증서를 LDAP[4] 서버에 저장한다. DC를 수용가에 설치할 때 발급된 DC의 인증서 및 개인키를 설정하고 DLCS의 인증서를 설정한다. SC는 SS의 인증서 hash 값을 가지고 있어서 SS의 인증 메시지가 도착하였을 때, 인증 메시지에 포함된 SS의 인증서를 추출하고 hash하여 가지고 있는 hash값과 비교함으로써 인증서에 대한 검증을 실시한다. SS는 SC의 인증 메시지가 도착하면 DC의 deviceID를 통해서 LDAP 서버로부터 인증서를 추출하여 서명 검증을 실시한다. 따라서, DLCS에서는 인증서 chain에 대한 검증 및 인증서 상태검증을 하지 않고 단지 인증 메시지의 서명을 검사함으로서 DC에 대한 인증을 할 수 있다.

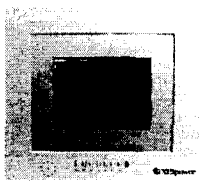
본 논문에서 사용된 DLC 시스템에서는 SS와 DC가 분리되어 있어 DC에 대한 실제적인 device 인증을 수행할 수는 없다. SS에서는 SC에 대한 인증을 수행할 뿐이지 DC에 대한 인증은 수행하지 않기 때문이다. 따라서, 실제적인 device 인증을 위해서는 DC와 SS가 하나의 장비로 구성되어야 한다. 그러나, SC가 구현된 환경이 DC가 구현된 환경과 일치하기 때문에 쉽게 DC에 SC를 포팅할 수가 있어서 실제 제품으로 통합되어 구현된다면 DC에 대한 인증이 가능하다.

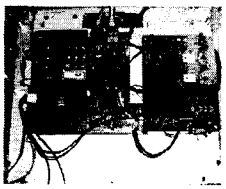


V. 테스트 환경

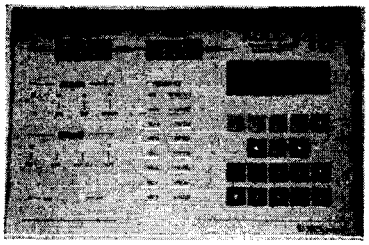
1) 테스트 환경

SS는 Solaris 5.6/440MHz CPU/128Mb memory에 구현되었다. DLCS와 Private CA는 Window 2000 server/1GHz CPU/512Mb memory에 구현되었다. SC는 Linux kernel 2.4.7/Intel StrongARM SA-1110, 206MHZ CPU에 구현되었다.





그림은 SC가 동작하는 장치이다. 첫 번째 그림은 SC의 전면사진이고 두 번째 그림은 SC의 내부 구조이다.



그림은 테스트에 사용된 DC이다. DC의 spec은 SC의 spec과 같이 Linux kernel 2.4.7/Intel StrongARM SA-1110, 206MHZ CPU에 구현되었다.

2) 보안 환경

SSL과 암호화 알고리듬을 지원하기 위해서 OpenSSL 0.9.6b library를 사용하였고, LDAP을 지원하기 위해 OpenLDAP 2.0.12 library를 사용하였다. 인증 메시지 서명 알고리즘은 SC와 SS 모두 RSA-SHA1을 사용하였다. SHA1을 이용해서 서명될 원문을 hash하고 그 hash값을 RSA 알고리즘을 사용하여 서명한다. RSA key size는 SC와 SS 모두 1024bit를 사용하였다. SSL은 SS의 인증서를 이용한 server-side SSL을 사용하였고, DES-CBC3-MD5 scheme을 사용하여 암호화 통신을 하였다.

VI. 성능평가

본 논문에서는 3가지 파라미터로 성능을 평가한다.

첫째, SC수에 따른 SS의 CPU 사용량을 측정한다. SC수가 증가함에 따라서 SS에서의 CPU 사용량 증가 정도를 측정함으로써, SS가 관리할 수 있는 SC의 수를 예측할 수 있다.

둘째, SSL session 주기에 따른 SS의 CPU 사용량을 측정한다. SSL session 주기에 따른 CPU 사용량을 측정하여 보안적 시스템적으로 알맞은 SSL session 주기를 예측할 수 있다. 본 논문에서는 SSL session 주기를 SSL 통신을 통해서 전송받는 데이터 횟수로 간주한다. SSL session 주기가 100이라는 것은 SC와 SS사이의 데이터를 주고받은 횟수가 100이 되면 현재 사용되는 SSL session을 끊고 새로운 SSL session을 맺는 것을 의미한다.

셋째, SC와 SS 사이의 data 주기에 따른 SS의 CPU utilization을 측정한다. data 주기에 따른 CPU 사용량을 측정함으로써, 본 논문에서 제시하는 제어 시스템의 부하 제어 주기를 결정할 수 있다. 일반적인 DLC 시스템에서는 1초 간격으로

DC에서 수용가 전력 사용 및 상태 데이터를 DLCS에게 전송한다. DLC 시스템이 아닌 다른 제어 시스템에서는 1초 정도의 제어 주기를 가지고는 부족할 수가 있다. 또한, 1초 이상의 제어 주기로도 충분할 수 있다. 제어주기에 따른 CPU 성능 측정은 다양한 제어 시스템에서 본 논문에서 제시하는 보안 시스템을 도입하였을 때의 성능을 가늠해 볼 수 있는 척도를 제시할 수 있다.

3가지 파라미터는 제어 시스템과 보안이 결합된 시스템에서는 주요 요소이다. 각 제어 시스템에 따라 3가지 파라미터를 적절히 선택함으로써, 보안 서비스와 제어 시스템 성능간의 trade-off를 통한 최적의 성능을 낼 수 있는 시스템을 구축할 수 있다.

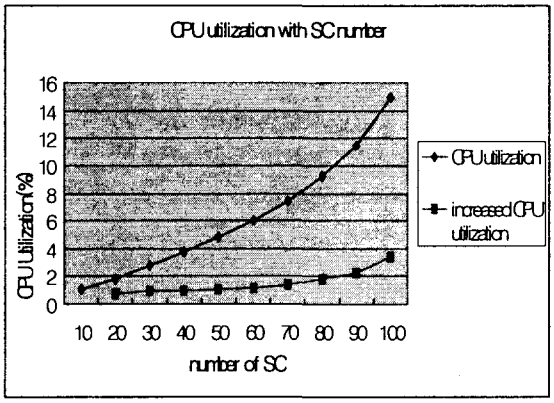
CPU 성능 측정은 Solaris가 제공하는 'top' 명령을 사용하여 SS 프로그램의 CPU 점유율을 측정하였다.

1) SC 수에 따른 SS의 CPU utilization

SC 수에 따른 SS의 CPU 사용량 측정은 SC의 수가 증가함에 따라서 SS에서의 CPU 사용량의 증가 정도를 측정한다. SC가 SS와 통신하기 위해서는 TCP connection과 SSL connection을 맺어야 한다. SS는 SC 마다 thread를 생성하여 하나의 thread가 하나의 SC에 대한 처리하므로 SC의 수가 증가할수록 thread의 수가 증가하게 되고 각 thread의 context switching등에 시간이 소요되므로 CPU 사용량은 증가한다.

SC 수에 따른 SS의 성능 측정 환경은 SSL session 주기를 100으로 하고, SC와 SS사이에 전송되는 제어패킷 주기는 1초로 설정하였다. DLCS에서 전송되는 부하제어 신호는 없다고 가정하였다. 각 SC에서 전송되는 데이터의 크기는 '75'로 간주한다. SC에서 처리해야하는 부하의 수가 증가할수록 데이터 사이즈는 증가한다. 본 논문에서는 모든 SC에서 처리해야 하는 부하의 수가 같다고 가정한다.

SC의 수	CPU 사용률	증가된 사용률
10	1.08%	-
20	1.78%	0.70
30	2.79%	1.01
40	3.74%	0.95
50	4.84%	1.10
60	6.02%	1.18
70	7.44%	1.42
80	9.23%	1.79
90	11.49%	2.26
100	14.92%	3.43



표는 SC의 수가 10씩 증가할 때마다 CPU의 사용률을 나타내고 있다. SC의 수가 증가함에 따라 필요한 SSL session의 수가 증가하고 처리해야 하는 암호화된 데이터의 양도 증가한다. 또한, 복호화된 데이터를 DLCS에게 전송하기 위한 session의 수도 증가한다. 즉, 하나의 thread는 각 DC에 대한 SSL session과 DLCS에 대한 TCP session을 필요로 한다. CPU 사용률의 증가량을 분석하면 SC의 수가 50이상이 되면 증가된 사용률이 일정하지 않고 커지는 것을 확인할 수 있다. 이로서, SC의 수가 증가함에 따라서 CPU의 사용률은 일정하게 증가하지 않고 지수적으로 증가함을 알 수 있다. 이는 thread사이의 context switching을 통

한 CPU 사용량의 증가와 통신 에러를 통한 새로운 sessoin을 맺는 과정이 증가함에 기인한다.

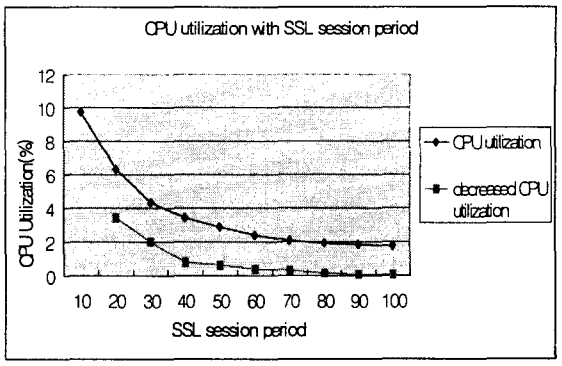
2) SSL session 주기에 따른 CPU utilization

SC가 SS와 데이터를 주고 받기 위해서는 TCP connection 후 SSL connection을 맺는다. 그 후에 서로 인증 메시지를 전송한다. SS에서는 LDAP에 접속하여 SC의 인증서를 획득한 후, 인증 메시지 검증을 실시한다. SC에서는 자신이 가지고 있는 SS의 인증서로 SS에서 보낸 인증 메시지 검증을 실시한다. SC와 SS간의 상호인증 후에 전력 정보 데이터 및 부하 제어 데이터를 전송한다.

SSL session의 주기가 짧아질수록 SSL session을 끊고 다시 맺기 위한 절차가 빈번히 발생하게 된다. 이는, 인증 메시지 서명 검증에서의 공개키 서명 검증 및 LDAP에서 인증서 획득하는데 많은 CPU 사용률이 발생한다. 또한, SSL session이 끝나면 해당 thread도 종료하게 되어 새로운 SSL session을 위해서 다시 thread를 생성해야 한다.

SSL session 주기에 따른 CPU utilization 성능 측정 환경은 SC와 SS간의 packet 주기가 1초이고 SC의 수가 20개로 설정하였다.

SSL session 주기	CPU 사용률	감소된 사용률
10	9.77%	-
20	6.32%	3.45
30	4.29%	2.02
40	3.45%	0.84
50	2.85%	0.61
60	2.44%	0.41
70	2.10%	0.34
80	1.92%	0.17
90	1.85%	0.07
100	1.78%	0.08



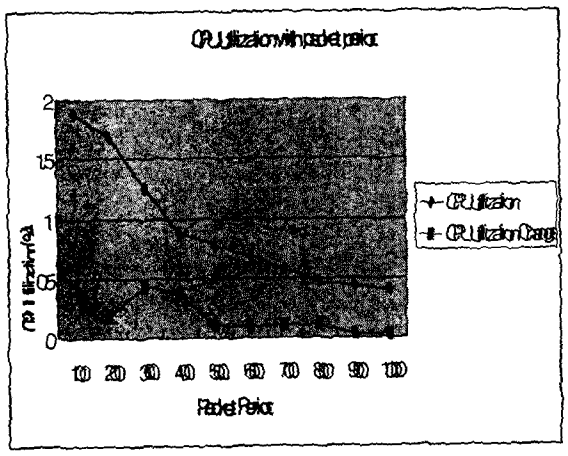
SSL session 주기가 10인 경우 CPU의 사용률이 9.77로 매우 높은 것을 확인할 수 있다. 그러나, 주기가 100인 경우 1.78로 현저히 낮아진다. 주기가 10인 경우는 LDAP search로 인한 overhead 및 공개키 서명 검증으로 인한 overhead가 매우 많기 때문이다. 감소된 사용률을 관찰해보면, SSL session 주기가 길어질수록 감소율은 급격히 떨어지다가 일정한 주기가 되면 일정하게 감소하는 것을 확인할 수 있다.

3) SC와 SS사이의 data 주기에 따른 CPU utilization

Packet 주기는 SC와 SS 사이에 시간당 얼마나 많은 데이터가 전송되는가를 나타낸다. 또한, packet 주기가 짧아진다는 것은 DLCS에서 짧은 간격으로 전체 시스템을 제어할 수 있다는 뜻이다. 즉, 제어 시스템의 순간 대응력이 향상된다는 의미이다. 제어 시스템의 특성에 따라서 제어에 필요한 데이터의 전송 주기를 정해야 하며 중앙 집중적인 제어 시스템에서 데이터를 처리할 수 있는지의 여부도 고려되어야 한다.

SC와 SS사이의 data 주기에 따른 CPU 성능측정 환경은 SC의 수를 20으로 하고, SSL session의 주기는 무한대로 하였다. SSL session의 주기를 무한대로 한 것은 SSL session 주기로 인한 재인증이나 thread 생성 등에 필요한 CPU 사용으로 나타나는 영향을 없애기 위해서이다.

packet 주기	CPU 사용률	감소된 사용률
100	1.88	-
200	1.69	0.18
300	1.26	0.43
400	0.89	0.37
500	0.78	0.11
600	0.68	0.10
700	0.57	0.11
800	0.46	0.11
900	0.43	0.03
1000	0.40	0.03



표를 관찰해보면 packet 주기가 길어질수록 CPU의 사용률은 감소한다. packet의 주기가 길어질수록 처리해야 하는 대칭키 암호화 데이터의 양이 감소하기 때문이다. 그러나, packet의 주기가 짧아져도 CPU의 사용률의 변화가 크지 않다. 이것은 대칭키 암호화는 CPU 사용에 별다른 영향을 미치지 않는 것으로 판단할 수 있다.

CPU 사용률 (SSL session 주기 : 100)	CPU 사용률 (SSL session 주기 : ∞)
1.78	0.4

VII.결론

인터넷DLC를 이용한 전력제어는 현재 많은 국가에서 실시되고 있다. 수용가의 전력제어를 위해 수용가에 있는 전력제어 장비와 중앙 전력제어 서버와의 통신을 위해서 여러 가지 방법이 있으나, 인터넷을 이용하는 방법이 비용적 측면에서 볼 때 가장 효율적이라 할 수 있다. 그러나, 인터넷을 통한 민감한 데이터의 송수신은 보안적 취약점을 가지고 있다. 특히, 전력산업은 국가 기반산업으로 다른 산업에 미치는 영향력을 감안한다면 인터넷을 통한 전력제어에서의 보안적 취약점은 오히려 역효과를 가져올 수 있다. 따라서, 보안 메카니즘은 인터넷을 통한 제어시스템에서는 상당히 중요하고, 성공적인 보안 메카니즘을 제시함으로써 타 시스템에서의 인터넷을 통한 제어 시스템 구축에 동기를 부여할 수 있다.

본 논문에서는 이러한 DLC 시스템에서의 보안적 문제점을 해결하였다. 또한 본 논문에서 제시한 해결 방법은 다른 제어 시스템에서도 사용될 수 있다. 즉, legacy 시스템에 SC와 SS 모듈을 탑재하여 보안적 요소들을 제공할 수 있다. DLC 시스템에서의 보안적 요소는 상호 인증 및 데이터 무결성과 기밀성, 부인방지를 들 수 있다. 데이터의 무결성과 기밀성은 SSL을 통해 해결하였으며, 상호 인증과 부인방지는 공개키 서명을 사용하여 해결하였다. 본 논문에서는 각 보안 parameter에 따른 성능을 측정하여 제시한 시스템의 성능을 분석하였고, 이러한 성능 측정 데이터는 다른 제어 시스템 설계 시 참고가 될 수 있다.

참고문헌

[1] R.Housley, W.Polk, W.Ford, D.solo, Internet X.509 Public Key Infrastructure Certificate and CRL Profile, IETF RFC 3280, 2002

[2] M.Myers, R.Ankney, A.Malpani, S.Galperin, C.Adams, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, IETF RFC 2560, 1999

[3] Internet Draft "Simple Certificate Validation Protocol" A.Malpani, R.Housley, T.Freeman, http://www.ietf.org/internet-drafts/draft-ietf-pkix-scvp-11.txt

December 2002
[4] S.Boeyen, T.Howes, P.Richard, Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2, IETF RFC 2559, 1999
[5] William Stallings, "Cryptography and Network Security - third edition" Prentice-Hall 2003
[6] Andrew Nash, William Duane, Celia Joseph Derek Brink, "PKI: Implementing and Managing E-Security" McGraw-Hill 2001
[7] Russ Housley, Tim Pork, "Planning for PKI" WILEY 2001
[8] T.Dierks, C.Allen, The TLS Protocol version 1.0, IETF RFC 2246, 1999
[9] Shuang-Yi Tang, Ying-Ping Lu, Du, D.H.C "Performance study of software-based iSCSI security" Security in Storage Workshop, 2002. Proceedings. First International IEEE, 2002 pp. 70-79
[10] http://www.keyinsystem.com/index_en.html
[11] S.Kent, R. Atkinson, Security Architecture for the Internet Protocol, IETF RFC 2401, 1998
[12] S.Kent, R. Atkinson, IP Authentication Header, IETF RFC 2402, 1998
[13] S.Kent, R. Atkinson, IP Encapsulating Security, IETF RFC 2406, 1998