닷넷 리모팅 환경에서의 보안 방안 구현
Implementation of Security Plan from .NET Remoting Environment

김영서
Young-Sear Kim

요 약

최근의 웹환경은 닷넷과 자바로 양분되어 있다. 닷넷은 웹환경에서 원격의 객체를 호출하는 방법으로서 리모팅이라는 기술이 있다. 리모팅 기술 방식은 HTTP(Hypertext Transport Protocol)환경에서 데이터를 주고받는 원격호출의 한 형태로 SOAP(Simple Object Application Protocol)형태의 확장기술이다. 본 연구는 ERP(Enterprise Resource Planning)와 같은 복잡한 업무의 웹프로그램에서 많이 사용하는 닷넷 리모팅 기술에서 ERP에 최적화한 응용계층 보안 적용, 리모팅 모듈과 암호화 모듈에서 중복되어 발생하는 직렬화 작업제거, 닷넷에서 사용되는 데이터 객체에 불필요한 데이터 제거를 통해 성능을 향상시키는 보안모듈 적용 방법을 제시하였다. 제시한 방법은 구현 및 실험을 통해 일반적인 보안적용에서 보다 약 2배의 속도 향상이 있음을 확인하였다. 향후 이와 관련된 컴포넌트 기반의 프레임워크 연구를 통해 편리한 개발자 환경 제공이 필요하다.

Abstract

While .NET and J2EE bisects recent distributed environment, .NET displays "Remoting" as a technology to call remote object. Remoting is frequently used as a protocol in OLTP's WEB program development in form of RPC that exchange data in XML form under HTTP environment. Purpose of this research is to draw problems when applying security to .NET remoting technology that is recently used in web programming, and to find effective application plan by implementing. The main discussion is following. First, network layer security should be replaced to application layer security for better performance and flexibility. Second, the serialization procedure that is repeated in both remoting and encryption module should take place once. Lastly, implementation of "Surrogate" and "Compress" will be discussed that enables to eliminate unnecessary data(table relations, keys, etc) that is used in dataset object of .NET in order to reduce the size of data. It is possible to achieve improvement in speed by two times through immediate implementation in these cases. In order for easier use, component based framework should be supplied hereafter.

Keywords : dot net, remoting, End-to-End, Security, Network, Serialize, Surrogate object

I. 서 론

컴퓨팅 환경이 중앙 집중 방식에서 분산 환경으로 변화하면서 지금까지 많은 표준들이 제시 되었으나 현재는 마이크로소프트가 중심이 된 윈도우스 기반의 "닷넷" 이라는 표준 환경과 선, 오라클, IBM 등이 중심이 된 자바 기반의 "J2EE(Java 2 Enterprise Edition)"가 사실상의 표준으로 사용되고 있다.[1][2][3] 두 기술은 ASP(Active Server Page)와 JSP(Java Server Page)라는 서버측 기술을 보유하고 있어서 동적인 웹페이지를 생성 할 수 있으며, COM(Component Object Model)과 EJB(Enterprise Java Beans)라는 콤포넌트 기술을 보유함으로써 프로그래밍 언어와 관계없이 상호 연계가 가능하다. 또한 개발된 프로그램의 위치가 동일 호스트 또는 네트워크 외부의 호스트에 상관없이 호출이 가능하며, ADO(Active Data Object) / OLE DB와 JDBC(Java DataBase Connector)라는 데이터베이스를 접근하는 표준 모듈을 가지고 있어서 다양한 데이터베이스를 편리하게 호출할 수 있는 장점이 있다.[4] 이러한 두 표준기술이 오늘날 구현하고자 하는 핵심 내용은 여러 가지가 있으나 가장 중요한 핵심은 확장성과 재사용성이다.[4] 확장성은 사용자가 증가할 시 유연하게 서버의 확장이 가능한 구조이고, 재사용성은 컴포넌트 방식을 이용하여 라이브러리 형태로 다양하게 재사용할 수 있어서 개발 생산성을 향상 시킬 수 있는 방식이다. 이러한 확장성과 재사용성을 극대화하기 위해 두 표준은 최근 웹서비스(Web Service)라는 기술을 사용하고, 이 기술을 손쉽게 구현할 수 있는 제반 환경을 제공하고 있다.

웹서비스는 네트워크상의 호스트에 존재하는 다양한 컴포넌트를 발견하는 기술인 UDDI(Universal Description Discovery Integration)와 발견된 컴포넌트의 구조를 누구나 이해하는 기술인 WSDL(Web Service Description

*을지대학교
투고 일자 : 2009. 9. 20 수정완료일자 : 2009. 10. 23
게재확정일자 : 2009. 10. 29

Language), 그리고 발견된 컴포넌트를 호출할 수 있는 기술인 SOAP(Simple Object Access Protocol)으로 구성되어져 있다. 이는 자신이 필요로 하는 네트워크상의 호스트에 존재하는 컴포넌트를 개발 당사자 간의 상호 협의 없이 찾고, 이해하고, 호출할 수 있는 기술이다. 이중 SOAP 기술은 컴포넌트를 호출하는 기술로서 HTTP(Hypertext Transport Protocol)라는 프로토콜을 사용하여 방화벽의 문제를 해결하고, XML(eXtended Markup Language)이라는 메시지를 사용하여 주고받는 메시지를 손쉽게 이해하게 하는 기술이다. 이러한 SOAP기술과 유사하게 마이크로소프트사가 가지고 있는 기술이 닷넷에 적용된 리모팅(Remoting) 기술이다.[5][6]

최근 닷넷 프로젝트가 일반화 되고 있으며, 이중 리모팅 기술을 사용하는 경우가 증가하고 있다. 이에 본 연구에서는 ERP(Enterprise Resource Planning)와 같은 복잡한 업무의 웹프로그램에서 많이 사용하는 닷넷 리모팅 기술에서 ERP에 최적화한 응용계층 보안 적용, 리모팅 모듈과 암호화 모듈에서 중복되어 발생하는 직렬화 작업제거, 닷넷에서 사용되는 데이터 객체에 불필요한 데이터 제거를 통해 효율적으로 보안을 적용하는 방법을 제시하고 실험을 통하여 평가하였다.

II. 리모팅 환경 하에서의 보안 문제점

2.1 네트워크 보안의 문제점

리모팅 환경은 앞장에서도 언급한 바와 같이 HTTP를 사용한다. HTTP를 사용하는 환경에서 사용되는 보안은 SSL(Secure Socket Layer)이 대표적이다.[7][8] SSL은 네트워크상에서의 데이터 전달에만 초점을 맞춘 것으로서 이는 응용 프로그램들의 고유 보안에 대한 고려 없이 개발되거나 사용되고 있을 때에 한해서 보안성을 쉽게 향상 시킬 수 있는 장점을 가지고 있다. 그림 1에서와 같이 네트워크 장비와 장비간의 보안을 말한다. 이는 기존 응용 프로그램에서 보안을 제공하지 않고 통신 인프라 상에서의 보안에 치중함으로써 개별적인 응용 프로그램에서 원하는 보안을 제공하기 힘든 경우가 대부분이다. 즉, 보조적인 수단으로서 보안성을 향상 시키는 곳에만 사용될 뿐 최종 사용자가 원하는 수준의 보안성을 제공하는 것은 불가능 하다. 그러므로 네트워크 환경 하에서의 보안 문제점은 다음과 같다.

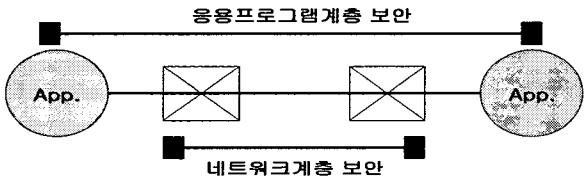

그림 1. 응용계층보안과 네트워크계층 보안

Fig. 1. Application layer security model vs Network layer security model.

첫째로, 성능상의 낭비가 크다. 네트워크 계층에서의 보안 서비스를 제공하는 시스템들은 응용 프로그램의 문맥(context)을 모르는 상태에서 일괄적으로 모든 데이터를 처리해야 하기 때문에 실제 비즈니스 시스템에서 과도하게 시스템 자원을 낭비한다. 실제로 이미지와 같이 파일 크기가 크면서도 보안성이 필요 없는 데이터도 보호하기 위해서 암호화 처리를 하는 것은 불필요한 작업이다.

둘째, 단대단(End-to-End) 보안을 제공하지 못하는 문제점이 있다. 즉, 링크 보안만을 제공함으로써 여러 통신 링크를 거쳐서 문서가 이동해야 하는 경우에는 한 통신 링크에서 다른 통신 링크로 넘어가는 접점에 보안상의 문제점이 발생한다. 그림 2에서 B의 경우 보안상의 문제점을 보여준다. 따라서 네트워크 계층상에서의 보안이 아무리 잘 되어 있어도 이 접점 부분이 많이 존재하거나 접점이 책임 있는 신뢰를 제공할 수 없는 공간에 존재하지 않으면 안전한 시스템 구축을 할 수 없게 된다. 대표적으로 모바일 환경에서의 WTLS (Wireless Transport Layer Security)계층이 가지는 제약과 지불 솔루션으로서 SSL이 가지는 제약이 있다. 이러한 통신 링크 사이의 보안 허점을 막기 위해서는 한 개의 링크로만 이루어진 시스템을 이용하거나 각 허점들을 시스템 소유자가 제어 할 수 있는 공간에 두어야만 한다. 그러나 두 가지 모두 비현실적인 경우가 많다. 하나의 링크로만 이루어진 시스템은 매우 단순한 시스템으로서 비즈니스에서 사용되는 시스템은 여러 링크들의 집합으로 이루어져 있다. 또한 필요한 모든 시스템들을 시스템 소유자 혹은 이용자의 공간에 모아 두는 것은 그 비용이 비현실적으로 증가하는 문제점이 발생한다.

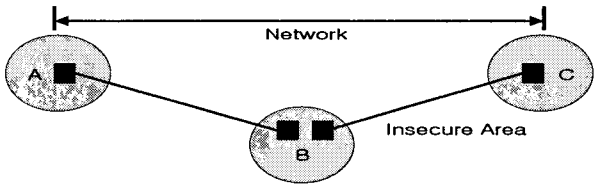

그림2. 네트워크계층에서의 링크 간 보안허점

Fig. 2. Network layer security hole.

셋째, 다양한 통신 및 컴퓨팅 환경의 융합 능력이 없다. 현재는 컴퓨터와 네트워크 기술이 빠르게 발전함으로써 유선은 물론 무선까지 포함한 다양한 통신 인프라에 접목해서 시스템이 구축된다. 같은 조직 및 시스템 내에서도 다양한 통신환경과 컴퓨팅 시스템이 사용되면서, 사용자 및 고객에게 다양한 접근 방법과 편리한 인터페이스를 제공해야 하는 환경이 필수적인 요소가 되었다. 이러한 상황에서 네트워크 프레임워크에서 제공하는 보안 특징을 이용하는 것은 편리하고 빠르게 보안 시스템을 구축하는 일일 수는 있지만 일관된 보안 정책을 적용할 수 없으며, 많은 경우 새로 채택되어 사용되는 최신 시스템에는 꽤 오랜 기간 보안이 적용되지 않기 때문에 기존의 적용된 보안 시스템조차 무용지물로 만들게 된다.

넷째, 처음부터 제공하지 못하는 보안 서비스들 또한 수없이 많다. 특히 전자 상거래에서의 계약 시스템은 SSL과 같은 네트워크 계층에서의 보안 솔루션으로는 구현이 불가능하다. 이는 계약서에 전자서명을 할 수 있는 응용 프로그램이 필요하기 때문이다. 그러므로 네트워크 계층에서 제공하는 보안서비스는 응용프로그램의 문맥에 밀접한 서비스가 불가능 하다.[10]

2.2 리모팅 보안의 문제점

닷넷의 리모팅(Remoting)과 SOAP은 서론에서 언급한 웹 서비스 기술의 하나로서 서로 다른 OS기종의 연계를 위해 사용되는 프로토콜이며, HTTP를 근간으로 하고 있다.

HTTP는 초기에 웹에서 간단한 데이터만을 송수신하기 위하여 설계되어서 많은 제약을 가지고 있다. 그 중에서도 HTTP는 객체기반으로 데이터를 전송 할 수가 없다. 오직 텍스트와 바이너리와 같은 스트림만 전송이 가능하다. 닷넷에서 리모팅을 사용할 경우는 표준으로 데이터셋(Data Set)이라고 하는 객체 형태를 데이터로서 사용하고 있으나, HTTP상에서 전송을 할 경우에는 데이터셋 그대로 전송이 불가능하여 XML기반으로 포맷팅(직렬화)을 하여 전송을 하게 된다.

XML형태로 포맷팅을 하는 이유는 일반적인 메시지 타입인 문자열(String)이나 배열(Array)이 메시지의 포맷정보인 스키마 정보가 없으므로 이해하기가 어렵기 때문이다. 결론적으로 객체처럼 유사하게 스키마를 가질 수 있는 표준 언어인 XML을 이용하는 것이다. 닷넷의 데이터셋 객체는 여러 가지 프로그래밍의 편의를 위하여 데이터베이스의 테이블 관계와 같은 정보를 가지고 있다. 이런 이유로 HTTP상에서 포맷팅을 XML형태로 할 경우 메시지의 크기가 상당히 증가 한다. XML은 모든 정보의 스키마를 태그 형태로 값의 앞과 뒤에 중복으로 표현하기 때문에 그 크기가 커지게 된다. 이런 점이 성능상의 낭비를 가져오는 원인으로 작용하게 된다.

그리고 데이터셋에 대해 암호화를 하려면 이 또한 직렬화(Serialize) 과정이 필요하게 되며 이런 부분이 리모팅에서 일어나는 직렬화 과정과 중복이 되어 속도 저하를 일으키게 된다. 물론, 이러한 성능 상의 단점을 보완하기 위한 해결책들이 제시되고 있다. SSL의 경우에는 성능 상의 문제점을 해결하기 위해 하드웨어 기반의 SSL 가속기를 사용하여 성능향상을 얻을 수 있으며, VPN(Virtual Private Network)도 하드웨어 VPN을 이용함으로써 해결할 수 있다.[11] 하지만 이를 구현하기 위해서는 추가적인 장비들을 별도로 구매해야 할 필요성이 제기되고 이는 추가비용을 야기 시킴으로써 효율적인 비즈니스 활동에 장애요인으로 작용하게 되며, 특히 필요없이 정보를 암호화 하고 인증하는 동작이 발생하기 때문에 기본적으로 낭비 요소로 작용하게 된다.

III. 리모팅 환경에서의 보안 적용방안

앞 장에서 언급했던 닷넷 리모팅 환경 하에서의 문제점들인 네트워크계층 보안의 문제점과 리모팅 환경에서 암호화 시 이중 직렬화 문제 그리고 닷넷 데이터셋의 크기가 큰 문제에 대한 해결방안은 다음과 같다.

3.1 응용계층 보안으로의 전환

네트워크계층의 보안적용에 대한 문제점들을 해결하기 위해서는 응용계층에서의 보안으로 전환 하여야 한다. 응용계층의 보안은 사용자가 직접 프로그래밍 하는 어플리케이션에 직접 보안 모듈을 삽입하여 적용할 수 있으므로 네트워크계층에서 문제점으로 지적된 여러 사항을 해결 할 수가 있다.

우선 성능 향상 및 세밀한 제어가 가능하다. 네트워크계층 보안은 모든 응용프로그램과 응용프로그램 내의 모든 모듈이 일괄적으로 보안을 적용하는 것에 반해 응용계층 보안은 보안을 적용하고 싶은 응용프로그램과 응용프로그램 내에서도 보안이 적용되어야만 하는 모듈에 국한해서 보안을 적용할 수 있다. 이는 불필요한 컴퓨팅 자원을 사용하지 않고 보다 정밀한 보안을 적용 할 수가 있으며 성능 향상을 가져올 수가 있다. 사실상 최근의 대부분의 응용프로그램이 프로그램 모듈별 세밀한 보안적용을 요구하고 있어서 응용계층의 보안이 대부분을 이루고 있다.

두 번째로는 단대단(End-to-End)간에 유연하면서도 허점이 없는 보안을 구현 할 수가 있다. 최근에 모바일과 같은 다양한 인프라가 구현됨으로 인해 네트워크계층 보안으로는 상호간에 표준이 상이함으로 인해 발생할 수밖에 없는 다양한 링크와 링크간의 보안상 문제점이 존재한다. 하지만 응용계층 보안은 그림 3에서와 같이 최종 사용자와 사용자 사이에서 보안이 이루어지므로 이러한 문제에 대해 자유로울 수 있다.

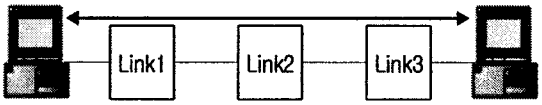

그림 3. 응용계층 보안과 네트워크 링크와 관계
Fig.3. Relationship of Application layer security and Network layer security.

위의 그림은 응용계층의 보안이 최종 사용자 PC에서 이루어지므로 중간에 존재하는 다양한 환경의 네트워크 링크와는 관계없이 종단 간에 보안을 적용할 수 있음을 보여 주고 있다.

3.2 중복된 사전작업의 제거

암호화를 하는 모듈은 데이터셋과 같은 객체를 직접적으로 처리할 수가 없다. 이는 문자열(String) 전환을 통해서만 암호화와 복호화 같은 작업을 할 수가 있다. 이런 암호화 모듈의 특징이 리모팅 환경에서 문제점으로 작용한다. 왜냐하면, 리모팅 역시 프로토콜로 HTTP를 사용하기 때문에 데이터셋과 같은 객체를 직렬화를 통하여 문자열로 전환 후 전송하므로 암호화시의 문자열 전환과 중복되는 불필요한 작업이 발생하게 되어 성능의 저하를 가져오게 된다.

그림 4에서와 같이 리모팅 환경에서 암호화 작업을 할 경우, 문자열 변환 및 리모팅 직렬화와 같은 중복된 직렬화가 일어나는 것을 알 수 있다. 이와 같은 문제점을 해결하기 위해서는 리모팅 모듈을 수정하여 리모팅 모듈 내부에서 문자열 변환 및 직렬화가 한번만 발생하도록 암호모듈과 복호모듈을 삽입하여야 한다. 그림 5에서와 같이 리모팅 모듈에 직접적으로 암호화 및 복호화 모듈을 내장함으로써 리모팅 내부에서 일어나는 스트링 변환과 직렬화를 암호화 및 복호화 모듈이 같이 사용하여 성능 향상을 할 수 있다. 이처럼 중복된 직렬화를 제거하기 위해서는 닷넷에서 기본으로 제공하는 리모팅 모듈을 수정하여야 한다. 마이크로소프트는 리모팅모듈을 수정 가능하게 개발하였고, 수정방법을 홈페이지를 통해 제시하고 있다.

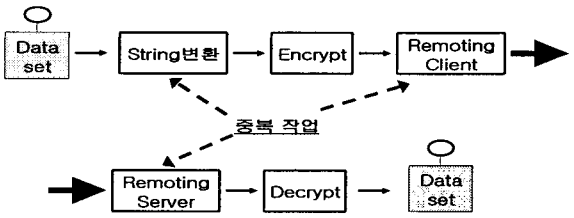

그림 4. 중복된 문자열 변환구간
Fig 4. Reiteration of string conversion

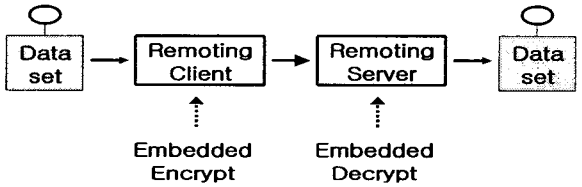

그림 5. 중복된 사전작업을 제거한 환경
Fig. 5. Remove of reiteration

3.3 최적화 방안을 통한 성능확보

최종적으로 성능향상을 위해 데이터의 크기를 최소화 하는 축약과 압축을 사용하여 데이터의 최적화하는 방안을 제시하였다. 첫째로, 축약객체를 만들어야 한다. 축약객체는 닷넷에서 사용하는 데이터셋 객체에서 불필요한 부분을 제거한 객체를 말한다. 닷넷에서 사용하는 데이터셋이라고 하는 객체는 기존의 자바의 레코드셋이라는 객체에 비해 크기가 상당히 크다.[5]

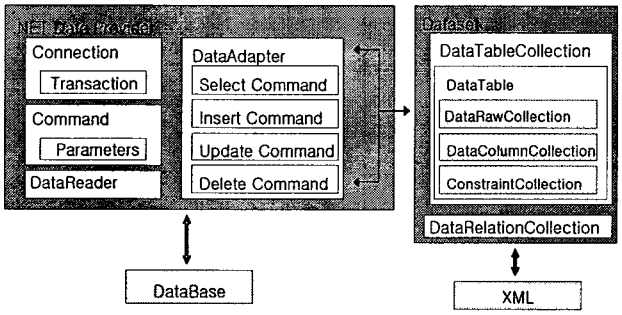

그림 6. 닷넷에서 사용하는 데이터셋 객체
Fig. 6. DataSet model of .NET

위의 그림 6에서와 같이 기존의 레코드셋은 SQL질의에 대한 결과 값 저장소 역할이 대부분이었으나, 데이터셋은 데이터베이스가 기본적으로 가지고 있는 테이블 및 관계 정보를 대부분 가지고 있는 저장소 객체이다. 그러므로 서버의 데이터베이스를 클라이언트에서 조회한 후 비연결적으로 처리를 함으로써 불필요한 네트워크 트래픽을 제거 할 수 있고 손쉽게 XML로 변환 할 수 있는 등의 장점이 있다. 그러나 그에 반해 객체의 크기가 상당히 커진다. 또한 이런 객체를 암호화하기 위해 문자열 변환 작업을 XML 형태로 구현을 할 경우 각종 정보가 XML 태그로 표현됨으로 인해 데이터의 크기가 더욱 더 증가하게 된다. 그러므로 데이터 이외에 테이블 정보 및 관계정보 등은 불필요한 정보로 이런 정보의 제거는 성능 향상에 반드시 필요하다.

(참조: msdn.microsoft.com)

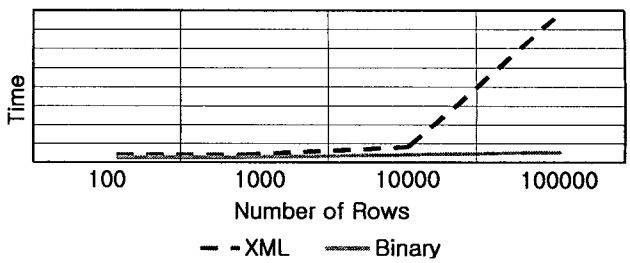

그림 7. XML과 Binary와의 End-to-End 속도 차
Fig. 7. Compare XML and Binary

그림 7에서와 같이 동일한 데이터 크기에 대해 Binary가 XML보다 상당히 응답속도가 빠른 것을 알 수 있다. 또한 데이터의 크기가 크면 클수록 속도의 차이는 더욱 커지는 것을 알 수가 있다.[9] 이와 같은 결과도 XML의 특징인 태그 정보 때문이다. 그러므로 데이터셋에서 불필요한 데이터를 제거하여 축약객체를 만들면 XML로 변환 시 데이터의 크기가 현격히 줄어든다.

둘째로는 데이터의 압축방법이다. 압축은 우리가 데이터의 전송 속도를 높이기 위해 일반적으로 가장 많이 사용하는 방법이기도 하다. 그러나 압축은 데이터의 크기를 줄여주지만 반면에 압축을 위한 추가적인 시간이 발생한다. 어떤 부분에 있어서는 데이터의 크기를 줄여 속도를 줄이는 효과에 비해 압축을 하고 해제하는데 소요되는 시간이 더 걸려 실질적인 속도의 향상을 가져오지 못하는 경우가 많다.

닷넷의 데이터셋의 경우도 직접적으로 압축을 실행하면 속도의 향상을 가져오기 어렵다. 그러나 데이터 사이즈가 작은 축약 객체로 만든 후 압축을 실행하면 속도의 향상을 이룰 수 있다. 이처럼 적절한 축약 객체와 압축을 사용하면 닷넷의 리모팅 환경에서의 성능 향상을 이룰 수 있다.

VI. 시스템 구현 및 분석

4.1 시스템 구현

실험을 위하여 클라이언트 PC에는 OS로 Windows XP를, 웹브라우저로 Internet Explorer를 사용하였고, Server는 OS로 Windows Server 2003을, 웹서버로 Internet Information Server를 사용하였다.

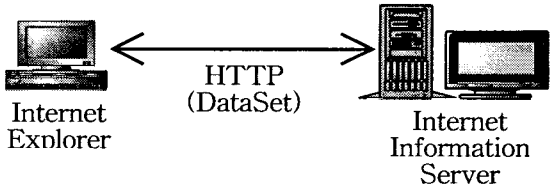

그림 8. 테스트 실험 환경
Fig. 8. Simulation Environment

이 경우 웹브라우저와 웹서버간에 HTTP 상에서 데이터셋을 주고 받을 때, 암호화를 삽입하면 암호화를 하기 위해 직렬화 하는 것이 다시 리모팅과 같은 원격호출모듈의 기본

적인 직렬화 기능과 함께 중복하여 사용되므로 속도저하가 발생한다. 그러므로 리모팅 모듈을 직접 수정하여 직렬화가 중복되어 사용되지 않게 하였다.

4.2 시퀀스 모델링 및 시나리오

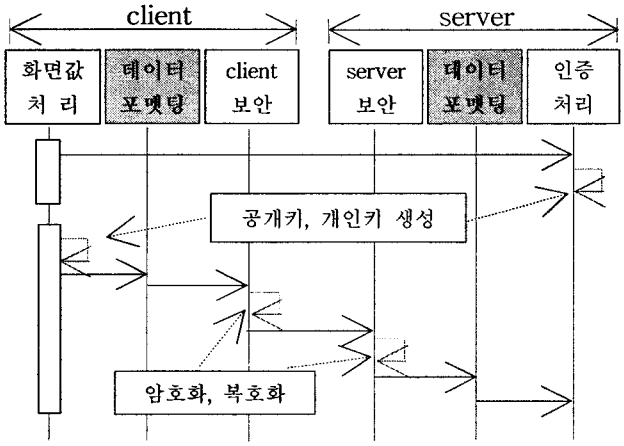

그림 9. 사용자 등록처리 시퀀스 모델링
Fig. 9. Sequence modeling of User Register

위의 그림 9에서는 최초의 사용자를 등록하는 부분을 모델링한 것으로 화면값 처리 부분은 서버로부터 받은 식별자(ID)와 임시키로 사용자를 등록하고 클라이언트의 개인키와 공개키 쌍을 생성한다. 데이터 포맷팅 부분은 데이터셋을 사용하는 리모팅 부분을 수정한 것으로 직렬화, 축약객체, 압축의 세 부분으로 구성되어 있다. 클라이언트 보안과 서버보안은 공개키 전달을 위한 암호화 및 복호화 모듈을 말하고, 인증모듈은 서버의 공개키와 개인키 쌍을 생성하고 최초에 사용자에 대한 식별자와 임시키를 메일로 발송하는 기능을 말한다.

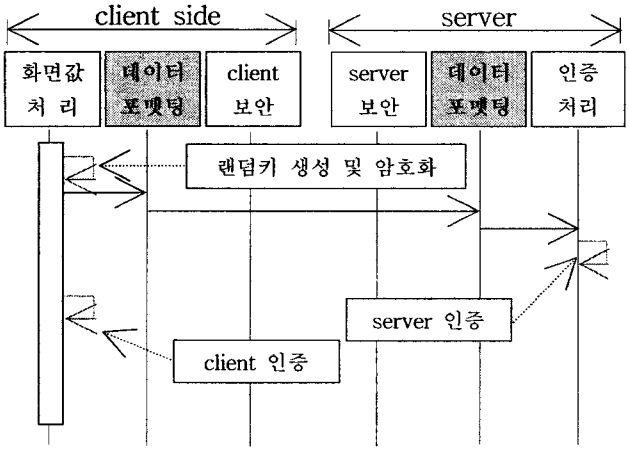

그림 10. 사용자 로그인 시퀀스 모델링
Fig. 10. Sequence modeling of login.

위의 그림 10에서는 사용자 등록 이후 로그인 시 상호 인증을 하는 부분을 모델링 한 것으로, 화면 값 처리는 클라이언트 랜덤 키를 생성 후 서버의 공개키로 암호화하여 클라이언트의 개인키로 서명하는 부분과, 서버에서 랜덤키 확인 및 클라이언트 랜덤키의 정당한 회신을 확인하고, 두 키를 Hash를 이용하여 조합 후 세션키를 생성한다. 인증모듈은 클라이언트 랜덤키를 확인하고 서버 랜덤키 생성 후 클라이언트 랜덤키와 함께 공개키로 암호화 하며 서버의 개인키로 서명한다. 이때 서버도 두 랜덤키를 Hash를 이용하여 세션키를 생성 후 보관한다.[2][3][6][7]

4.3 실험 결과 및 분석

위의 설계대로 구현된 프로그램을 실험 환경하에서 테스트하였으며, 테스트 시나리오는 우편번호 4만여 건을 암호화 하였으며, 실제로 테스트 시간은 암호화를 위한 사전 포맷팅 부분만을 측정하였다. 실험 결과 일반 방식으로 할 경우는 아무런 처리도 안한 경우로서 직렬화 이후의 데이터 크기가 16M 이며 처리시간은 평균 4.66초 이었다. 압축의 경우는 단순 압축을 한 경우로 데이터 크기는 많이 줄었으나 압축을 하는 시간이 많이 소요되어서 실질적인 성능 향상은 없었다.

표 1. 실험결과 요약
Table 1. Result of simulation

방 식	처리 크기	소요시간(sec)		
		1차	2차	3차
표준 (일반방식)	16M	4.66	4.65	4.66
압축실시	590K	4.75	4.75	4.74
축약실시	5.8M	4.70	4.72	4.72
축약 후 압축실시	409K	2.60	2.63	2.62

축약의 경우는 데이터셋에서 사용하지 않는 불필요한 데이터를 제거한 후 실험한 것으로서 이경우도 데이터 크기는 줄었으나 축약시간이 많이 걸려서 이득은 없었다. 마지막으로 축약을 실시한 후 압축을 한 경우는 평균 2.62초로 표준방식 보다 44%의 성능 개선을 확인하였다.

V. 결론 및 향후과제

본 논문에서는 최근의 분산 환경 표준의 하나인 닷넷 환경에서의 전사적자원관리(ERP)와 같은 OLTP성 웹프로그램에 주로 사용되는 리모팅 모듈의 효과적인 보안 적용 방안에 대해 제안하였다. 이번 연구를 통하여 얻어진 결과는 리모팅 환경에 있어서 보안을 적용 할 경우, 응용프로그램에 특화된 보안 적용을 하고 암호화 할 때와 리모팅 할 때 중복되어서 객체의 직렬화가 일어나는 점을 해소하기 위한 방안으로 리모팅 모듈에 직접 암호화 모듈을 삽입하여 직렬화를 한번만 일어나게 하였으며, 닷넷에서 사용되는 데이터 객체인 데이터셋이 데이터베이스의 모든 테이블 관계, 키 등의 속성을 XML 형태로 표현하고 있어서 그 크기가 크므로 인해 발생하는 속도 저하를 줄이기 위해 불필요한 데이터를 제거하는 축약(Surrogate)객체 생성하는 것과 데이터의 압축이 필요하다는 것을 확인하였다. 위와 같은 사실을 실험을 통해 약 44%의 처리속도 개선효과를 입증하여 보았으며, 이는 데이터가 증가하면 증가 할수록 더욱 속도의 개선 차이가 발생함을 확인하

였다. 향후 이 같은 리모팅의 보안적용 시 문제점을 극복하기 위하여 닷넷에서 제공하는 리모팅 모듈에 대해 좀 더 용이하게 수정 할 수 있도록 컴포넌트화가 필요하며, 닷넷뿐 아니라 자바환경에서도 적절한 보안적용을 위한 연구가 필요하다.

참고문헌

[1] 남도현, "웹기반의 ERP보안 프레임워크 구현, "중앙대, pp.19-21, 2004.
[2] 임성춘, 남도현, 이종태, "중소기업 ERP 유지보수의 효율적 지원절차에 관한 연구," 대한산업공학회/한국경영과학회 춘계공동학술대회논문집, pp.1309-1316, 2006.
[3] 박성진, 문봉교, "분산객체 컴퓨팅과 ERP서버," 정보과학회지, 제16권 11호, pp.29-37, 1998.
[4] 박종명, 강준석, "ASP.NET 웹파트 : 웹파트로 개인화 서비스구현," 마이크로소프트웨어, 통권303호, pp.174-179, 2009.
[5] 오재인, "ERP를 통한 통합정보시스템의 구현전략: A기업의 사례," 경영과학회지, 제15권 제2호, pp.83-90, 1998.
[6] 최무진, "국내 ERP연구에 대한 고찰과 과제," 경영정보학회 1999춘계학술대회논문집, pp.37-45, 1999.
[7] 이만영외, 인터넷보안기술, 생능출판사, pp.52-53, 2002.
[8] 조은애, 문창주, 백두권, "CBD에 기반한 SSL 컴포넌트의 설계 및 구현," 정보과학회논문지. 제12권 3호, pp.192-207, 2006.
[9] 차영욱, 김춘희, "XML과 웹 서비스 컴퓨팅 ,"안동대학교 산학협력단 출판사, pp.23-33, 2007
[10] 홍승필외, e-Business Security, 파워북출판사, pp.165-170, 2000.
[11] 신태삼, 김영범, "MPLS망을 기반으로 하는 VPN의 성능에 관한 연구," 신호처리·시스템 학회 논문집 V8, n.1 pp.51-57, 2007.

김 영 서(Young-Sear Kim)
正 會 員
1987년 2월 인하대 전자공학과(공학사)
1989년 2월 인하대 전자공학과(공학석사)
2009년 2월 인천대 정보통신공학과 (공학박사)
2002년 8월 ~ 2007년 2월 서울보건대학 의료공학과 조교수
2007년 3월 ~ 현재 을지대학교 의료공학과 조교수
※주관심분야 : 생체신호처리, 정보보호 및 보안