웹 서비스 성능 개선 방안

허의남*, 이필우**

An Efficient Scheme for Web Services Performance Improvement

Eui-Nam, Huh and Pil-Woo Lee

Abstract

This paper proposes a novel Web Services enhancement method, which enhances XML based e-commerce and Web Services securely and efficiently. Recently, the ratio of XML Denial of Services (XDOS) is increased significantly. However, XML data is visible only at the target application site or the application layer. Thus, an efficient and secure mechanism handling XML data on the network layer to provide fast performance on the server and protect XDOS at network level is strongly required basic component. This paper presents many issues and gives useful results from the various experiments to support an implemented XAN platform.

Key Words: Web Service, 보안, 암호, 전자서명, 전자상거래, XDOS, XML

* 경희대학교 전자정보대학 교수

** 한국과학기술정보연구원(KISTI) 팀장

1. 서론

전자상거래의 확대는 디지털화 된 유통 정보가 보다 커다란 가치를 갖게 됨을 의미하므로, 전자상거래에 있어 보안성이 결여된다면, 거래에 있어서의 각종 정보가 불법 노출, 부당 거래, 자원의 불법적인 접근, 서비스 거부 등의 보안 침해 사고가 빈번히 발생할 수 있다. 디지털 정보에 대한 보안 및 인증은 비대면 전자거래에서 상호 신뢰를 기반으로 거래를 수행하기 위해 최우선적으로 고려되어야 할 측면이다.

최근 글로벌 전자상거래 표준으로서 ebXML, RosettaNet, Web Services 등이 등장하였으며, 이중 전 세계적으로 주목받고 있는 표준은 ebXML 과 Web Services이다. ebXML에서의 보안 및 인증은 전자거래에 대한 안전성을 보장하기 위해 꼭 필요한 기술이며 XML 정보보호 기술이 그 근간을 이루고 있다.

Web Services에서의 보안도 중요한 문제로 대두되고 있으며 현재 Web Services 보안의 기반이 되는 WS-Security (Web Services Security)가 표준화 완료 단계에 있으며, WS-SecurityPolicy, WS-SecureConversation, WS-Trust, WS-Federation 등이 표준화 단계를 거치고 있다. ebXML 보안 기술과 Web Services 보안 기술의 기초가 되는 것은 XML 정보보호 기술로, XML 전자서명 (XML Digital Signature), XML 암호 (XML Encryption), SAML (Security Assertion Markup Language), XKMS (XML Key Management Specification) XACML(Access Control Markup Language) 등이 있다. 향후 전자상거래는 ebXML 및 Web Services 기반 환경 하에서 이루어지리라 예상되며, 이를 위해서는 이들에 대한 안전성을 보장해 줄 수 있는 기반을 제공하는 XML 정보보호 기술 및 웹 서비스 보안 기술의 적용이 반드시 필요하다.

이에 따른 IT기술의 큰 흐름은 기술과 서비스의 통합(Convergence), 표준화(Standardization), 고속 네트워크화(Hi-Performance Net-work)로 특징지을 수 있다. 웹 서비스는 분산 자원과 Web 기술의 통합을 목표로 하는 표준화된 기술로 빠른 속도로 전자통신 업계의 '공통어'로 자리 잡고 있다. 하지만 인터넷을 통해 교환되는 XML 문서의 비약적인 증가로 인하여, XDOS (XML Denial-of-Service), 무한 루프의 잘못된 스키마 등의 공격성 XML 문서가 또한 증가되고 있다. 이러한 트래픽을 네트워킹 레벨에서 해결하지 못해 결국 응용계층에서 이를 처리해야만 하므로 시스템상의 치명적인 보안 결함이 생길 수 있는 것이다. 이러한 문제점을 해결하기 위한 방안 중의 하나인 XML-Aware Network (XAN) -암호화, 서명, 변환, 필터링 등의 처리를 네트워크 레벨에서 처리 - 에 대한 기술의 필요성이 대두되어 본 연구에서는 보다 안전하고 효율적인 전자상거래를 지원하기 위한 요소들을 구현하여 하나의 Simulator 환경을 구성하도록 한다. 또한 보다 나은 성능을 발휘하기 위한 모델링을 제시하고 그 성능을 개선하고자 한다. 본 연구의 결과물을 통해 전자거래 실제 상황에서 발생하는 각종 공격의 유형도 알아내는 중요한 플랫폼으로 그 역할을 수행하리라 본다.

본 논문의 구성은 2장에서 기존 XML 기반한 전자 상거래의 문제점 및 SOAP을 이용한 전자 거래의 문제점을 알아보고 3장에서는 2장의 문제점에 대한 기존의 관련 연구에 대해서 알아보았다. 그리고 4장에서 안전한 XAN 장비의 플랫폼 모델을 설계하고 5장에서 실험결과를 보여주고 6장에서는 5장의 실험 결과 데이터를 통해서 XAN의 최적의 성능 개선 기법을 제안하고 7장에서 결론 및 향후 연구계획을 논한다.

2. 웹 서비스 주요 문제점

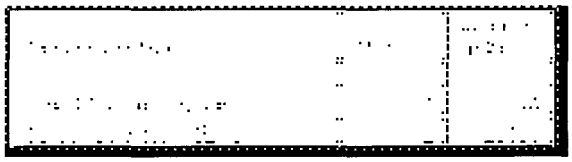
2.1 기존 XML의 문제점

웹 컨텐트를 가속화하는 솔루션에는 HTTP

로드 밸런서, 컨텐트 캐시 장비, SSL 가속기 등이 있는데, 이들은 XML 트래픽을 가속화하거나 처리하는 능력은 없다. 또한 지금까지 XSLT (eXtensible Stylesheet Language Trans -formation) [1]를 이용한 XML 변환이 실제 활용 면에서 너무 느리다는 문제가 지적돼 왔다. 전형적인 XSLT 변환은 초당 수백 킬로바이트로 매우 느리게 진행되는데 이 정도 속도로는 작업 진행이 사실상 불가능하다. 아래 <그림 1>은 서버가 처리하는 XML 문서 변환에 대한 TPS(transaction per second)를 보여준다.

Benchmark Run	TPS	Latency (mSec)
Direct XML to browser	2060	23
JSP, no XML	1250	67
Server Convert, XALAN	107	463

(a) 소형의 XML-to-HTML 변환



(b) Data intensive XML-to-HTML 변환

<그림 1> XML 문서 변환 벤치마킹

오늘날 XML 기반 어플리케이션 이용 증가로 각종 암호화, 인코딩/디코딩, 전자서명, 변환 등의 프로세싱 자원을 충당하기 위해서는 서버를 보충해야 할 필요성이 대두 했다. 그러나 서버를 새로 장만하기에는 초기 비용이나 유지비용이 많이 들게 될 뿐 아니라 실시간 사업이나 웹 어플리케이션의 응답 시간이 너무 오래 걸리는 것에 대한 근본적인 문제 해결 방안이 아니다. 따라서 기업 사용자들은 안전한 전자거래를 위해서는 XML 성능 문제를 네트워크 단계에서 직접 해결해야 할 필요성을 갖게 되었다. 여기에는 XML 트래픽을 다루는데 목적을 둔 밑바닥부터 설계된 기술이 필요하다. 차세대 XAN 장비는 현재 미국의 'Data Power' 회사의 XA-35 과 XS-40 [3] 의해 부상중이다. 이러한 솔루션들은 XML 데이터 스트림 관리, 처리, 보호, 가속 등을 가능케 한다.

2.2 XML 기반의 전자거래의 문제점

본 절에서는 XML 기반의 전자 상거래 상에서 다양한 형태의 문제점을 아래와 같이 분석해 보았다.

-검증 및 서명 시 Spoofing

Service A에서 Service B로 인증을 요구하는 경우 SOAP을 통한 연동으로 Service A에서 Service B에 있는 특정 서비스(예:actor속성이 있는 경우)를 이용시 해커들의 Spoofing으로 ID 도용이 항상 존재한다.

- XML DOS 공격 Malformd DTD (infinite loop)

잘못된 DTD나 XSD로 계속해서 요구하는 경우 서버에 엄청난 부하를 주어 결국 서비스를 불가하게 한다. 그러나 이는 기존의 네트워킹 장비(firewall)에서 찾아 낼 수가 없다는 것이다.

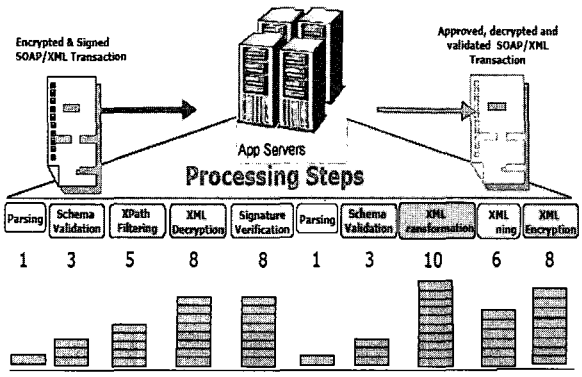
- Application의 위험

부적절한 entity가 포함되거나 자원의 reference (참조)가 잘못 지적된 경우 및 메시지의 검증 레벨에 있어서 응용 서버가 알아야 할 많은 위험이 존재한다. 또한 SOAP 및 WSDL과 같은 프로토콜의 유효성이 검증되지 않은 경우에 공격을 받을 수 있다.

- Resource Intensive

웹 서비스에 보안 처리가 된 경우 이를 정상 처리 하는 데는 엄청난 서버의 자원이 낭비 된다. 아래 그림 2는 (참조 [2]) 안전한 거래가 이뤄지기 위한 XML 처리 과정과 발생

되는 부하를 보여준다. Parsing 과정의 부하를 하나의 기본 단위 "1"로 했을 경우 각 수행 단계별 부하 범위가 최대 "10"까지 발생함을 볼 수 있다.



<그림 2> XML 문서의 보안처리시 오버헤드

3. 관련연구

대부분의 웹 서비스 방식은 Transport 수준의 보안에만 주력을 하고 있다. (예, SOAP 메시지는 Transport 계층에서 중개자 없이 직접 전송이 되는 형태이다.) 현재 웹 서비스 개발자들은 Point-to-Point 보안 방법, IPSec을 이용한 암호화 기법, IP 인증, Privacy, 데이터 무결성 등의 구현에 주력을 하고 있다. 오늘날 XML 기반의 웹 서비스 비즈니스가 중요한 의미를 내포하고 있지만 비즈니스 환경에서 계속적으로 변화해 가는 새로운 표준들과 불확실성을 쉽게 수용 할 수 있는 웹 서비스의 보안의 구체적인 활용이 요구된다. 따라서 XML 보안에서는 새로운 기술들을 흡수해야 할 뿐만 아니라 서로 다른 기업들의 목표에 맞는 performance와 protection이 모두 가능해야 한다[4].

현재 전자서명과 암호화를 수행하기 위해서 IBM의 AlphaWorks, Baltimore의 X/Secure등 몇 가지의 XML 보안 제품이 개발 되어 있다. IBM의 Alpha Works는 XML 전자서명과 XML 암호를 위한 기능을 제공하고 있으며 상용화 된 제품이 아니라 XML 전자서명의 예제 구현을 위하여 개발되었다. Alpha Works의 전자서명 모듈은 XML 전자서명 표준 초안에 따라 개발 되었으며, XML 암호 모듈의 경우에는 자체적으로 정의한 규격에 따라 정의한 규격에 따라 구현하였다.[5] 최근 Apache 에서도 Apache-XML-Security라는 XML 전자서명 구현 패키지를 개발 하였으며 MS에서도 자사의 .NET 프레임워크에 XML 전자서명 기능을 통합해 넣었다. 이외에도 DSTC, HP, Entrust, NEC, Verisign에서도 각각 자사의 XML 전자서명 패키지를 개발하였다. Baltimore의 X/Secure[6][7] 또한 XML 전자 서명과 암호화를 제공한다. X/Secure도 XML 전자서명 표준 문서의 초안에 따라 구현하였지만, 현재 발간된 초안에 명시된 기능들 중 많은 부분을 제공 하지 못하고 있다.[5]

XML 전자서명은 W3C의 XML-Signature 워킹그룹이 IETF와 공동으로 XML 전자서명에 대한 표준화 작업을 진행 중이며, 표준화의 핵심인 XML 전자서명 구문 및 처리절차 (XML-Signature Syntax and Processing)가 [8]올해 2월 W3C의 권고안(recommendation)으로 채택되었다. XML 암호화는 W3C의 XML Encryption 워킹그룹에서 표준화 작업을 수행중이며, 2002년 8월에 후보 권고안 (candidate recommendation)이 제안되었다[9].

4. 웹 서비스 성능 개선을 위한 구조

4.1 XAN의 기능과 역할

앞 절에서 살펴보았듯이 XML 웹 서비스 보안 문제는 앞으로 더욱 중요성이 대두되기 때문에 시스템 설계시 다음 10 가지 사항들이 권고 되어 지고 있으며 이를 토대로 본 논문에서는 XAN S/W 플랫폼을 구현한다.

-전송계층의 보안

전송계층이 안전하여야 한다는 것이다. 즉 IP 기반의 HTTP 프로토콜이 대중화 되어 있기 때문인 것이다. SSL VPN 이 가장 널리 사용되는 방안인데 여기에 사용자의 Certificate 등도 같이 처리되어져야 하는 것이 바람직하다. 이러기 위해서는 하드웨어에서 이 기능을 처리함이 바람직하다고 볼 수 있다.

-XML 필터링

XML을 활용한 비즈니스 룰은 매우 복잡해질 것이 당연시되고 있기에 enterprise에 처리되기 전에 문서의 내용이 필터링 되어야만 위협이나 DoS 로부터 안전할 수 있다.

-내부 자원의 마스킹

현재의 NAT(Network Address Translation) 기술은 직접적인 TCP연결을 막을 수 있는 중요한 자원 보안 기법 중의 하나임에 틀림없다. 이러한 기술이 웹 서비스에도 존재하여야 자원을 보호할 수 있다. XML 전자 거래에서는 XML Proxy(rewrite URL)가 이와 같은 역할을 한다.

-XDOS 공격으로 부터의 방어

XDOS 의 공격은 네트워크의 Syn flooding 만큼 많지는 않을 것으로 예상되나 그 피해는 훨씬 크다. 따라서 XML filtering 기술을 이용한 Gateway 혹 Proxy 가 있어야 한다.

-모든 메시지의 유효성 검사

스키마와 일치하지 않는 엘리먼트의 검사, 문서의 그 조가 올바르게 작성되었는지에 대한 검사가 필요하다.

-메시지 변환 기능

XML 문서는 각기 다른 사용자의 인터페이스에 맞도록 변환되어 져야 한다. 즉, XSLT를 이용한 변환이 필요하다.

-문서의 전자서명

모든 문서에 대한 소유자를 명시할 수 있는 기능이 반드시 필요하다. 소유자의 identity를 문서에 첨가하고 이를 분석해 내는 과정은 반드시 안전한 전자거래를 위한 XAN의 기본적 기능이라 하겠다.

-모든 메시지의 Time-Stamp화

기업의 엔터프라이즈 업무를 처리하는 각종 트랜젝션 관리에 있어 모든 메시지의 시간 마킹 기능은 반드시 이루어져야 한다.

-요소별 암호화

XML은 내용과 구조를 가지고 있어 그 의미를 파악하기가 쉬우므로 (예: <주민번호> 123333-1222124 </주민번호>) 값과 엘리먼트의 내용은 반드시 암호화되어야 한다.

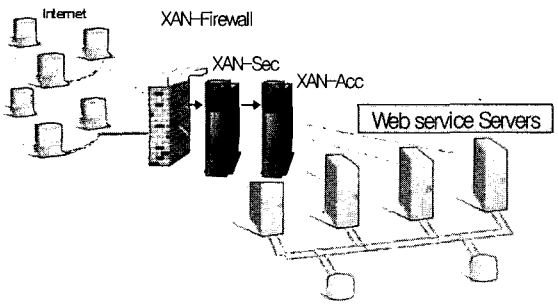
-안전한 감시기능

주로 log를 분석하여 시스템의 상황을 점검하는데 전자거래에서는 XML signature에도 시간이 명시되고 메시지에도 time-stamp가 찍히므로 이를 활용한 새로운 기업 엔터프라이즈용 감시 시스템이 구현되어 져야 한다.

4.2 XAN의 구조 및 Flow 설계

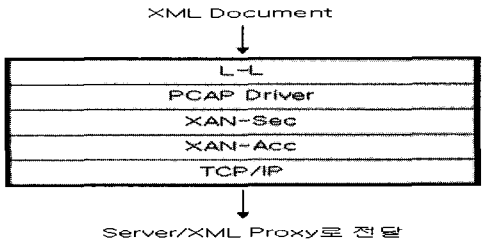
본 절에서는 4.1절에서 권고되는 각종 기능들을 분석하여 효율적인 XAN S/W 플랫폼이 되도록 설계하였다. XAN S/W 플랫폼은 XML Signature 표준(안) [8], [10] 과 XML Encryption 표준(안)[9], [11]에 기반 하여 각 모듈을 설계 하였으며, 실제 응용 프로그램에 적용될 경우, 개발자의 편의성을 향상 시킬 수 있도록 각 API가 설계 되었다. 이를 위해서 보안 표준(안)에서 정의한 전자 서명의 구조를 단순히 생성 하는 것 뿐 아니라 몇 개의 API들 만을 호출함으로서 복잡한 처리 절차를 수행 할 수 있도록, 많은 부분을 캡슐화 하여 API를 설계 하였다. 아래 <그림 3>은 각 필

요 장비의 구성도를 보여 주고 있다. 여기서 XAN-Sec은 보안을 담당하는 Security 서버 부분이며 XAN-Acc는 성능을 가속화하기 위한 Accelerator 서버 역할을 수행한다.



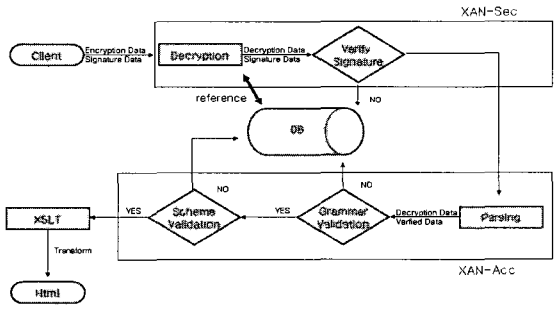
<그림 3> XAN 시스템 구성도

본 논문에서는 부가적으로 S/W 플랫폼인 XAN을 구현하기 위해서 PCAP (Packet Capturing Library)을 활용하여 서버 측 게이트웨이에서 전자상거래 타겟 포트로 들어오는 패킷을 검사한다. <그림 4>는 PCAP이 포함된 S/W 플랫폼 구성도이다. PCAP에서 필터링된 해당 패킷들은 XAN-Sec와 XAN-Acc단계로 순서대로 처리 되며 결과가 이상 없이 수행 되면 해당 패킷을 Server/XML proxy로 전송하게 된다.



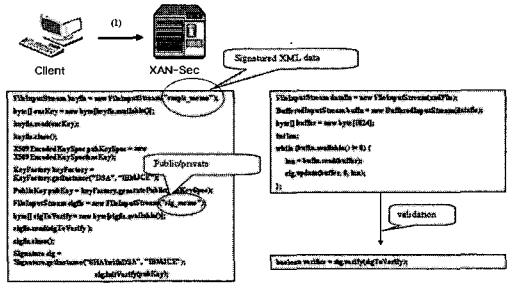
<그림 4> XAN S/W 플랫폼 구성도

<그림 5>는 전체 문서의 처리 과정을 보여준다. Client에서 XML 문서를 보낼 경우 XAN-Sec에서 복호화 (Decryption), 전자서명 검증(Signature verification)을 담당하고 XAN-Acc에서 파싱, 문법 유효성 점검 (Grammar validation), 스키마와의 일치성 (Schema validation)을 담당한다. 만약 전자서명에서 검증 실패하거나 문법의 유효성 부분이나 스키마 검증을 통해 유해 사이트 혹은 보안에 영향을 미친다고 판단되면 해당 XML 데이터를 보낸 Client IP, XML 문서명과 함께 통과하지 못한 검증 모듈 이름과 count를 DB에 저장 관리 하게 된다. 만약 또 다시 동일한 Client에서 보안에 위배 되는 XML 데이터가 올 경우 미리 저장된 DB를 참조하여 해당 검증 단계부터 검증을 실시한다. 이렇게 함으로써 반복되는 공격성 패킷을 조기에 처리하여 부하를 줄인다. 또한 동일 문서가 몇회 이상 (구현 시 5번으로 셋팅) 수신될 경우 해당 사이트를 유해 사이트로 분류하여 더 이상 검증 단계들을 진행하지 않게 된다.



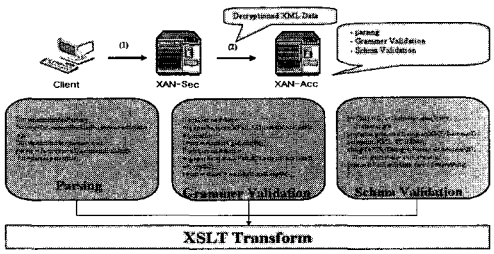
<그림 5> XAN 문서 처리 과정

<그림 6>은 XAN-Sec가 Decryption된 XML 데이터를 받아서 Encryption하면서 생성된 Public/Private key와 비교하여 전자 서명하는 과정을 보여준다.



<그림 6> XML Decryption 및 전자서명 검증

<그림 7>은 XAN-Sec에서 전자서명 검증된 XML 데이터를 파싱하여 Grammar Validation, Schema Validation을 진행하는 과정을 보여 준다. 모든 단계가 정상적으로 수행 되었을 경우 원하는 Server 혹은 XML Proxy로 보내지기 위해 XML Transform이 진행된다.



<그림 7> XAN-Acc 전처리 과정

5. XAN S/W 플랫폼 성능 분석 및 실험

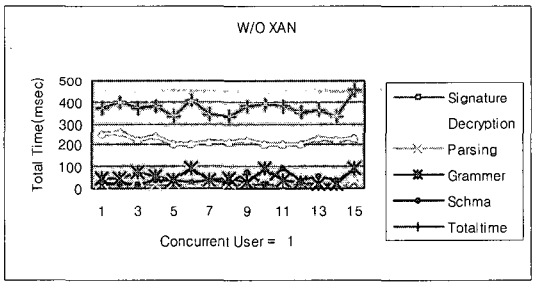
본 실험은 <표 1>과 같이 2대의 PC에서 각종 실험을 수행 하였다.

<표 1> 실험환경

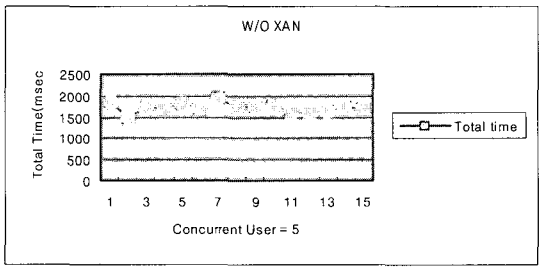
	서버1	서버2
OS	Window2000 Service Pack 4	Window XP Service Pack1
CPU	P4 - 2.4G	PM - 1.4G
Memory	512M	256M
Resource	CPU/메모리 10/20%사용중	CPU/메모리 10/35%사용중

XAN이 없이 서비스 제공자 서버(추후 W/O XAN으로 표기함)일 경우 서버1을 사용하였고 XAN을 사용한 경우는 서버1은 XAN-Sec를 내장하고 서버2는 XAN-Acc를 내장하여 실험을 진행 한다.

W/O XAN의 실험을 위해 한명의 사용자가 동일한 XML 데이터를 한번씩 15회에 걸쳐서 요청해 보고 또한 동시에 5명의 사용자가 15회에 걸쳐서 요청해 본다. 여기서 $S.P_{st}$를 서비스 제공자 싱글 서버의 Signature time, $S.P_{dt}$를 Decryption time, $S.P_{pt}$를 Parsing time, $S.P_{gt}$를 Grammer validation time, $S.P_{mt}$를 Schema validation time, $S.P_{tt}$를 Processing total time로 표시 하여고 그림 8의 (a)와 같이 한명의 사용자가 요청 하는 경우 XAN-Sec인 $S.P_{st} + S.P_{dt}$ 의 평균 시간은 약 240msec 이고 XAN-Acc인 $S.P_{pt} + S.P_{gt} + S.P_{mt}$ 의 평균 시간은 약 120msec 이다. 그리고 $S.P_{tt}$는 약 360 msec 로 측정되었다. 또한 <그림 8>의 (b)에서 나타나 있듯이 5명의 동시 유저가 동시에 XML 데이터를 15회에 걸쳐서 계속적으로 보내본 결과 $S.P_{tt}$는 약 1800 msec의 시간이 측정됨을 알 수 있었다.



(a) 1 사용자 접속시 W/O XAN 성능

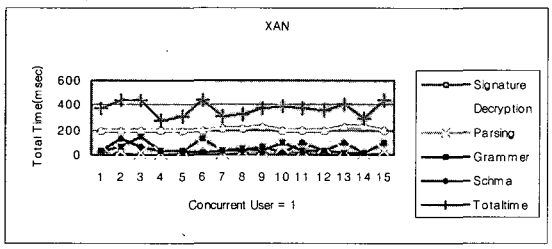


(b) 동시 5 사용자 접속시 W/O XAN 성능

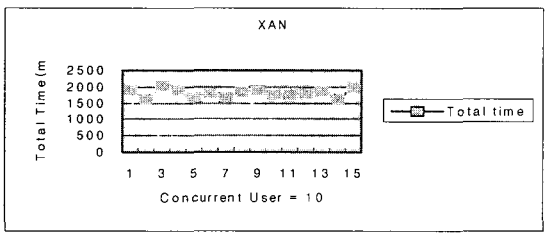
<그림 8> W/O XAN 성능

또한 XAN을 두어 성능 실험을 위해 한명의 사용자가 XML 데이터를 한번씩 15회에 걸쳐서 요청해 보고 또한 동시에 여기서 $D.P_{st}$를 Signature time, $D.P_{dt}$를 Decryption time, $D.P_{pt}$를 Parsing time, $D.P_{gt}$를 Grammer

validation time, $D.P_{mt}$를 Schema validation time, $D.P_{tt}$를 Processing total time 로 표시하면 <그림 9>의 (a)에서와 같이 한명의 사용자가 요청하는 경우 XAN-Sec인 $D.P_{st}$ + $D.P_{dt}$ 의 평균 시간은 약 240msec 이고 XAN-Acc 인 $D.P_{pt}$ + $D.P_{gt}$ + $D.P_{mt}$ 의 평균 시간은 약 120msec 이다. 여기에 over time이 20msec정도 나타났다. 그리고 $D.P_{tt}$는 약 380 msec 이다. 또한 <그림 9>의 (b)와 같이 10 명의 동시 사용자가 접속할 경우에는 $D.P_{tt}$ 약 1800msec이다.



(a) 1 사용자 접속시 XAN 성능



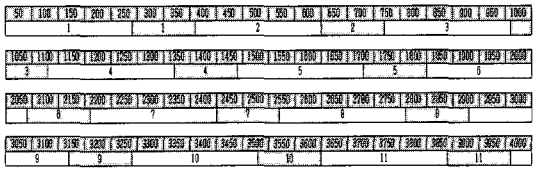
(b) 10 사용자 동시 접속시 XAN 성능

<그림 9> XAN 성능

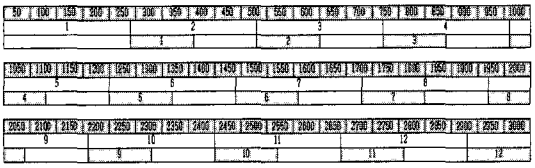
본 실험을 통하여 1 사용자 인 경우는 네트워크 처리 부분이 추가로 첨가되어 XAN 의 효율성을 찾을 수는 없지만 사용자 증가될수록 XAN의 역할은 큰 의미를 지닌다는 것을 알수 있었다.

6. 성능 최적화 모델링

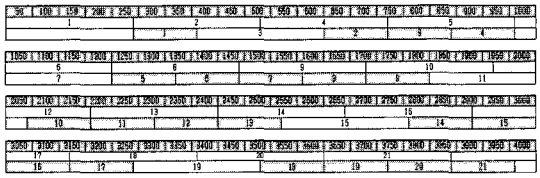
본 연구는 5장의 성능 분석을 통하여 최상의 성능을 내기 위하여 XAN-Sec와 XAN-Acc의 평균 시간으로 모델링을 진행 하였다. 5장에서 확인 한 바와 같이 해당 프로세스 진행시의 평균 시간은 XAN-Sec는 240msec 이고 XAN-Acc는 120msec이다. <그림 10>은 이를 바탕으로 한 검증 모듈별 수행 시간에 바탕을 둔 모델을 보여준다. 이 그림에서는 구현시 사용한 Multi-thread 개념을 반영하지 않았기 때문에 실제 프로세스와는 다소 차이가 있을 수 있지만 최적의 성능을 내는 환경을 찾는데 그 목적이 있다. <그림 10>에서 ▭ 은 XAN-Sec를 나타내고 ▭은 XAN-Acc를 나타낸다. 안에 숫자는 처리하는 동시 사용자 수를 나타낸다. (a), (b)는 앞 절에서 성능 실험이 실제로 동작 하는 모습을 보여 주고 있다. (b)에서 XAN의 경우 XAN-Sec는 바쁘게 활동하고 있지만 XAN-Acc는 상당히 많은 타임 슬럿이 비어 있음을 알 수 있다. 즉 로드 밸런서 (Load balance)를 고려하기 위해 XAN-Sec는 이전과 동일하게 동작 하게 하고 XAN-Acc 측에 남은 시간 슬럿(time slot) 을 활용하기 위해 best fit 방법을 적용 XAN -Sec 기능을 추가 하여 일정 단위로 실행을 시켜 남은 time slot을 활용하는 방법을 (c)에서 보여 주고 있다. (d)는 두 대의 PC 모두 XAN-Sec와 XAN-Acc 기능을 내장 시켜 한 번씩 번갈아 request를 던져 주는 Round robbin 방식이다. Round Robin 방식을 사용할 경우는 많은 수의 동시 사용자가 접속 할 경우 Waiting이 많이 일어나기 때문에 실제 결과 데이터는 모델링과 다소 차이가 나는 것을 알 수 있다. 현재 분산 환경에서 Round Robbin 방식은 많이 사용되지 않고 여유 시간 (slack)이 있는 자원에 동적으로 필요한 양의 자원을 할당하는 best fit 방식이 널리 사용되고 있어 본 논문에서도 이를 적용하였다.



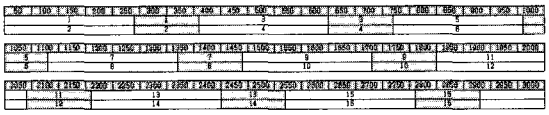
(a) W/O XAN 성능



(b) 로드 발란서가 적용되지 않은 XAN 성능



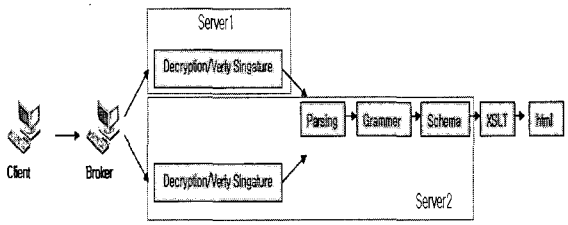
(c) Best fit 방법을 이용한 XAN 성능

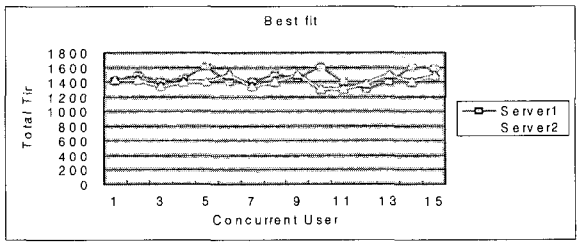


(d) Round robbin 방법을 이용한 XAN 성능

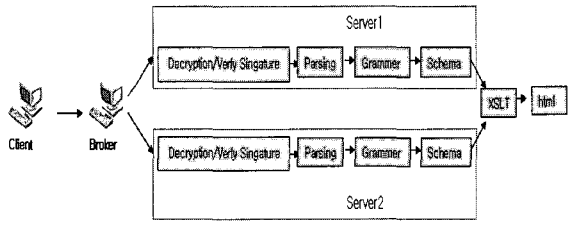
<그림 10> 성능 분석

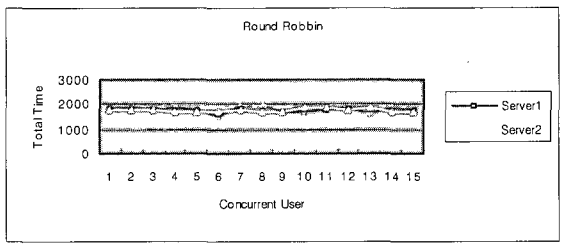
<그림 11> (a)는 Best fit을 적용한 XAN의 실험 결과를 보여 주고 있다. <그림 10>의 (c)모델링 값과 같이 3, 7, 9번째 사용자가 들어 올 경우에는 해당 XML 데이터를 서버2에서 Decryption 단계를 처리 하도록 구성 하였다. 그 결과 기존의 XAN의 성능보다 약 2배의 성능을 개선 된 것을 볼 수 있다. 그림 11의 (b)는 Round Robin 방식을 적용한 그래프이다. Round Robin을 적용 했을 경우에는 앞 단계인 XAN-Sec에서 시간 소요가 발생하기에 Waiting 현상이 나타나 성능의 향상에 걸림돌이 된다. <그림 10>의 (d) 와 같은 예상은 할 수 가 없다.





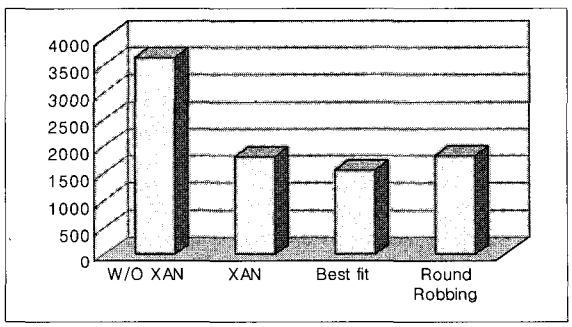
(a) Best Fit을 이용한 XAN 성능





(b) Round Robbin을 이용한 XAN 성능

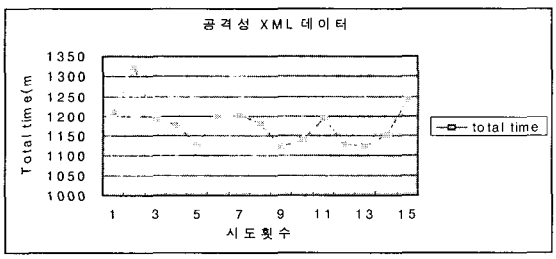
<그림 11> XAN 성능 최적화 모델링



<그림 12> 각 모델별 성능 분석

<그림 12>에서와 같이 다양한 로드 발란서 기법을 적용하여 수행한 결과 W/O XAN일 때 보다 XAN일 경우 약 2배의 성능이 개선되었고 남은 시간 슬랏을 모두 활용한 Best fit 방식을 이용 할 경우는 약 2.4배의 성능 개선을 시 킬 수 있다.

또한 본 논문에서는 전자서명에서 검증이 실패하거나 문법 검증, 스키마 검증에서 걸러지는(detected) 오류 혹은 공격성 XML 문서 등에 대한 성능 및 보안 측면에서 휴리스틱한 방안을 고려를 하였다. 앞 절에서 설명한 것처럼 사용자의 IP, XML 문서명, 및 검증 단계를 DB 에 저장해 둬서 동일한 유저가 동일한 XML 문서를 보낼 경우 해당 XAN 장비에서 Decryption 후에 DB를 검사 하고 검증 실패 되었던 단계를 재 수행하여 문서를 테스트 하므로 나머지 단계의 수행을 줄여 더 좋은 성능 개선을 확인하도록 구현하였다. 그림 13은 10명의 사용자가 동시에 검증에 실패한 문서를 15회에 걸쳐서 연속적으로 요청한 경우의 성능을 보여 준다. 즉, 본 논문에서 제안한 XAN 플랫폼 상에서의 결과는 150건의 문서를 1초내에 처리하는 성능 개선 효과를 실험을 통해 알 수 있었다.



<그림 13> 공격성 XML 요청시 XAN 성능

7. 결론

본 논문에서 웹 서비스 지원을 위한 XML에 대한 연구와 웹 서비스 성능향상을 위한 기법을 제시하였다. 인터넷 및 분산 컴퓨팅 환경의 혁신으로 나온 XML과 웹 서비스가 활성화되기 위해서 해결해야 할 주요 난제로 지목되고 있는 성능 및 보안 부문을 해결하기 위해 여러 기능들이 표준화 되어 왔으며 이를 기반으로 산업계에서 웹 서비스 보안을 위한 표준화 작업에 적극적으로 참여하고 있다. 기업의 서비스 기반의 전자 거래는 보안기능과 응용 업무 기능으로 분리되어야 하고 보안기능은 Network상에서 XAN을 이용하여 적은 비용으로 웹 서비스가 더 빠르고, 안전하게 운영되도록 해야 할 것이다. 본 연구의 결과는 잠재적으로 매우 다양한 분야에 적용 될 수 있다. 주식 거래 정보, 개발 중인 신제품이나 회사에 대한 중요한 기밀 정보, 입찰, 주문, 결제 내역서 등의 인터넷을 통하여 전송되는 경우 XAN 장비를 사용하며 네트워크를 통해 교환되는 모든 종류의 XML 형태의 데이터에 적용 될 수 있고, 그 적용 결과로 XML 형태의 전자 서명과 암호문을 생성하기 때문에 기존에 개발 되어 사용 중인 XML 응용 프로그램과 쉽게 연동 할 수 있다. XML이 전자 상거래에서 사용되는 문서의 표준 형식으로 채택됨에 따라 본 논문에서 소개된 XAN 플랫폼은 그 활용의 범위가 더욱 넓어지고 다양해질 것이다. 향후 XML 기반의 통신 프로토콜 보안, XML 기반 키 관리, XML 기반 접근 제어, XML 기반 보안 정보 교환기술에 대한 연구가 활발해 질것으로 보이며 그에 대한 연구도 계속해서 나갈 것이다.

참고문헌

[1] The World Wide Web Consortium(W3C) 's XML web page; http://www.w3.org/XML/

[2] Rick McGuir,. XML Acceleration Appliances. Emerging Internet Technologies IBM Software Group. November 5, 2003

[3] DataPower web page; http://www.datapower.com

[4]XML Security Gateway web page:http://

www.datapower.com/products/xs40.html
[5] IBM AlphaWorks Homepage, http://www. alphaworks.ibm.com/tech/xmlsecuritysuite
[6] Baltimore, "X/Secure White Paper," http://www.baltimoreinc.com/library/whitepapers/xsecure.html
[7] Baltimore, "X/Secure Developer's Guide," 1999.
[8] IETF/W3C, XML-SignatureRequirements (WorkingDraft)," Oct. 1999, http://www.w3.org/TR/1999/WD-xmldsig-requirements-19991014.html
[9] W3C XML Encryption WG, "XML Encryption Charter," http://www.w3.org, 2001.
[10]IETF/W3C, XML-Signature Syntax and Processing(Working Draft), Oct. 2000, http://www.w3.org/TR/2000/WD-xmldsig-core-20001012/
[11]xml-encryption@w3.org Mail Archives, http://lists.w3.org/Archives/Public/xmlencryption/

주 작 성 자 : 허 의 남
논 문 투 고 일 : 2005. 10. 12
논 문 심 사 일 : 2005. 10. 22(1차), 2005.10. 27(2차) 2005. 11. 22(3차)
심 사 판 정 일 : 2005. 11. 22

저자소개

허의남
1990 부산대학교 전산통계 학사
1995 텍사스대학교 전산학과 석사
2002 오하이오대학교 전산학과 박사
2002~2003 삼육대학교 컴퓨터과학과 조교수
2003~2005.8 서울여자대학교 정보통신공학부 조교수
2005.9~현재 경희대학교 전자정보대학 조교수
관심분야: 네트워킹, 그리드, 모바일 컴퓨팅, 유비쿼트스, 정보보호

이필우
1993 Univ. of Tsukuba 석사
1997 Univ. of Tsukuba 박사
1997~ 2000 우송대학교 초빙교수
2000.~현재 한국과학기술정보연구원 그리드 연구실 팀장
관심분야: QoS, 네트워킹, 그리드 네트워크, 시뮬레이션