2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

안전한 XML 문서 전송을 위한 인증 프로토콜

임승채*, 임동조*, 김태연*, 이기준*, 정채영**

*조선대학교 대학원 전산통계학과
**조선대학교 자연과학대학 전산통계학과
e-mail: tmdco@weppy.com

Authentication Protocol for Secure XML Document Transfer

Sung-Chea Lim*, Dong-Jo Lim*, Tae-Yeun Kim*, Kee-Jun Lee*, Chai-Yeoung Jung**

*Dept. of Computer Science and Statistics, Graduate School, Chosun Univ.
**Dept, of computer Science and Statistics, College of Natural Sciences, Chosun Univ

요약

본 연구에서는 인터넷상의 자료 전송에 쓰이는 XML문서가 평문의 형태로 전송되는 한계를 극복하는 방안과 OSI 참조 모델의 상위 계층에서 안전한 암호화와 인증 기법을 이용한 프로토콜을 제시한다. 제안된 인증 프로토콜은 디지털 서명을 사용함으로써 XML문서의 내용 변경 여부를 확인하고, 암호화로 XML문서의 내용 유출을 막음으로서, 웹 어플리케이션 개발에 사용할 수 있는 가능성을 제시한다.

1 서론

인터넷의 급속한 성장으로 여러 종류의 네트워크 서비스가 WWW를 통하여 일반 사용자들에게 제공되고 있다. 그 한 가운데에서 점점 더 각광을 받아 가는 표준이 바로 XML이다. XML은 HTML의 한계를 극복함과 동시에 서로 다른 어플리케이션에서 가공되고 사용되는 데이터들의 표현이 가능하다[1][2]. 따라서 이질형 데이터 포맷 형태의 자료도 XML 문서로 변환하여 일반 네트워크를 통해 전송이 가능하다. 하지만 XML문서를 전송할 경우 평문 형태 그대로 전송이 되기 때문에 이에 대한 암호화 인증의 필요성이 대두되었다.

이에 따른 기존에 제안된 방식으로는 넷스케이프에서 개발된 SSL[3][4]과 EIT에서 HTTP에 보안을 접목한 SHTTP[5]가 있다. 하지만, SSL은 OSI 계층에서 중간 계층에 해당하는 방식인 하위 프로토콜의 형태를 띄고 있고. 또한 SHTTP는 인증과 전자 서명, 부인 방지 기능을 가진 프로토콜이긴 하지만 전용 브라우저를 사용해야 하는 단점을 지니고 있다.

본 논문에서는 웹 프로그래밍이나 웹 어플리케이션에서 사용이 가능한 범용적인 XML 암호화 방식을 제안하고자 한다. 본 논문의 구성은 2장에서 기존의 보안 방식에 대하여 알아보고, 3장에서는 제안하고자 하는 전송 프로토콜의 암호화 전송 방식 및 구성에 대해 알아보고, 4장에서 제안방식의 안정성 및 효율성을 고찰하고 5장에서 결론을 맺는다.

2. 기존의 보안 및 인증 방식

2.1 SSL

SSL은 넷스케이프사에서 개발된 프로토콜로 특정응용을 위한 보안 프로토콜이 아닌 일반적인 인터넷 보안 프로토콜로 사용되는 방식이다. 즉 인터넷 응용과 TCP/IP 통신 계층 사이에 존재하는 프로토콜로서 웹을 위한

HTTP뿐만 아니라 Telnet, FTP등 다른 응용분야에서도 사용이 가능하다.

2.1.1 SSL의 구조

SSL의 구조는 Handshake Layer 와 Record Layer로 구분되어지며 Handshake Layer는 암호화 방법이나 열쇠의 결정 및 협상을 담당하며 Record Layer는 전송을 위한 규약의 정의와 데이터 전송을 담당한다.

① Handshake Layer

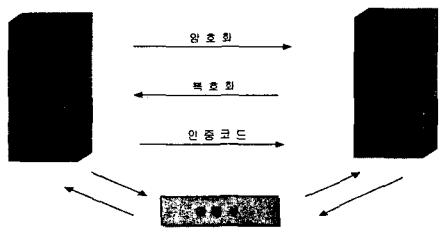
Handshake Layer는 Change Cipher Spec Protocol, Alert Protocol, Handshake Protocol로 구성되어 있다.

Handshake Protocol의 Handshake Message는 Client가 Server에 접속하는 즉시 서로간에 교환이 일어나게 되며 이때 Protocol version, Cryptography algorithm, key등을 교환하게 되며, 상호 인증도 이 과정에서 처리해 준다. 만일 Resume Session인 경우에는 이전의 Session에서 교환된 정보들을 그대로 사용하게 되기 때문에 보다 간단한 Handshake 과정만을 거친다.

Change Cipher Spec Protocol는 Handshake Protocol에서 협상한 암호화 방법, 암호화 키, 암호화 알고리즘등 암호화 전송방식을 알리는 역할을 수행한다.

Alert Protocol은 Clint와 Server사이에 주고받는 Data 중에서 정상적인 처리가 이루어지지 않을 경우 경고 메시지를 보내게 된다.

[그림 1]은 핸드쉐이크 과정을 나타낸다.

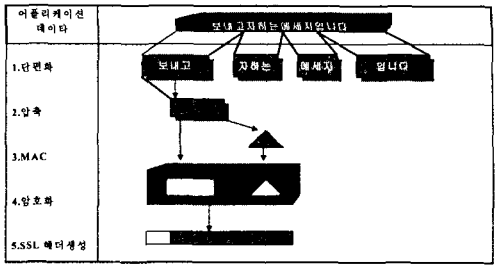


[그림 1] 핸드쉐이크 과정

② Record Layer

Record Layer는 Change Cipher Spec, Alert, Handshake, Application data의 네가지 메시지를 가진다.

Change Cipher spec message는 앞으로의 Data 전송에 있어서 서로간에 협의된 Cipher spec을 따르겠다는 것을 상대편에게 알려주는 역할을 수행하고 Alert message는 Warning과 Fatal, Close notify, Handshake failure등을 비롯한 여러 메시지가 존재하게 되는데 만약 Level이 Fatal인 경우에는 즉시 연결을 종료하게 된다. [그림 2]는 SSL의 Recode Layer를 나타낸다.



[그림 2] SSL 레코드 계층

2.1.2 SSL의 인증과 무결성[3][4]

① SSL의 인증

SSL의 인증은 신뢰할 수 있는 제 3의 기관이 발행한 디지털 파일을 이용한 서버측의 증명과 클라이언트측 증명으로 구성된다.

서버의 인증방식은 클라이언트가 SSL을 사용하는 서버에 접속 요청을 하였을때 서버는 자신의 인증에 서명만을 하고 암호화는 하지 않은 채 클라이언트에게 보낸다. 클라이언트는 서버로부터 전달받은 서명이 신뢰할 수 있는 인증기관에서 발행했는지 여부를 확인하고 만일 그렇지 않다면 클라이언트는 사용자에게 이 인증이 알려지지 않은 인증기관에서 발행되었다고 알려준다. 이를 위하여 클라이언트는 인증의 정보와 서버로부터 받은 정보의 비교가 필요하다.

클라이언트의 증명방식은 먼저 SSL을 사용하는 서버에 자신의 서명된 인증서를 전달한다. 서버는 인증서내에 존재에하는 클라이언트의 공용키를 사용하여 본인여부를 확인한다. 이때 서버는 인증기관의 신뢰성여부를 확인하고 만일 클라이언트의 정보와 인증서의 정보가 일치한다면 서버는 요청된 클라이언트를 받아 들인다.

② SSL의 무결성

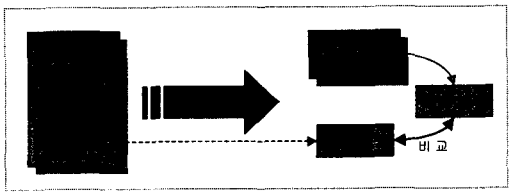
SSL의 무결성은 서버나 클라이언트가 자신이 수신한 메시지가 송신자가 보낼때의 메시지와 똑같은 것인지를 확인하는 방법이다. 따라서 메시지의 무결성을 위하여 키가 있는 암호학적 해쉬함수를 이용한다. 송신자는 메시지를 해쉬함수에 입력하여 그 결과인 MAC를 보내려는 메시지와 함께 보낸다. 그러면 수신자는 수신된 메시지를 키를 이용하여 해쉬함수에 입력하고 그 결과가 자신이 수신한 MAC와 비교하여 메시지의 무결성을 검사할 수가 있다. 하지만 SSL의 무결성방법은 부인봉쇄를 제공하고 있지는 않고 있다.

2.2 SHTTP[5]

SHTTP는 1994년 EIT(Enterprise Integration Technologies)에서 개발한 HTTP의 개선된 보안 프로토콜 버전으로 안전한 통신을 보장하기 위하여 응용 레벨에서 메시지 암호화를 수행한다. SHTTP는 SHTTP를 수용하는 웹 브라우저를 통하여 서비스되므로 SHTTP 전용 브라우저와 함께 SHTTP를 수용하는 서버를 필요로 하는 문제점을 지니고 있다.

2.3 XML 문서의 무결성 검증(MD5)

XML문서를 MD5로 디지털 서명하여 그 값을 XML 문서내에 삽입하여 전송하는 방식으로, XML문서의 변조 여부를 확인할 수 있다. 하지만 XML 문서 자체의 보안 인증 방식으로 보기에는 약간의 무리가 있다.



[그림 3] MD5를 통한 전송

3. 제안된 XML 인증 프로토콜

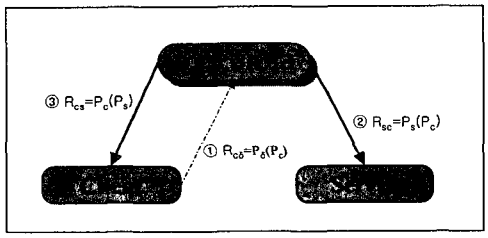
본 논문에서는 대칭키·공개키 암호화와 디지털 서명, 상호 인증을 통하여 XML 문서 전송을 위한 인증 프로토콜을 제안한다.

[표 1] XML 인증 프로토콜을 위한 기초항목들

항 목	내 용
P_δ	인증기관의 RSA 공개키
P_s	서버의 공개키
P_c	클라이언트의 공개키
$R_{\delta s}= P_\delta(P_s)$	서버에서 인증 기관으로의 공개키 전송
$R_{\delta c}= P_\delta(P_c)$	클라이언트에서 인증기관으로 공개키전송
서버키 인증서	인증기관에서 클라이언트로의 공개키전송
k_1	난수로 발생된 DES 대칭키
$R_{sk}= P_s(k_1, P_c)$	대칭키와 클라이언트의 공개키를 서버의 공개키로 암호화한 값
C_c	DES 키 k_1로 암호화한 XML 문서
$\tilde{m}'$	XML문서를 MD5로 만든 디지털 서명값
$\tilde{m}''$	암호문 C_c을 MD5로 만든 디지털 서명값

3.1 전송 준비 단계

클라이언트와 서버는 공인된 공개키 인증기관의 인증서를 통하여 각각의 공개키를 상대방에 전송한다. 클라이언트와 서버는 인증기관의 인증서를 통하여 인증기관 공개키 (P_δ)를 알고 있으므로 먼저 서버 측에서 공개키 P_s를 인증기관의 공개키 P_δ로 암호화한 값 $R_{\delta s}= P_\delta(P_s)$을 생성하여 인증기관으로 전송하여 저장하고, 클라이언트에서는 서버의 공개키를 받기 위하여 자신의 공개키 P_c를 인증기관의 공개키 P_δ로 암호화한 값 $R_{\delta c}= P_\delta(P_c)$을 생성하여 인증기관으로 전송한다. 인증기관에서 전송받은 클라이언트 암호값을 해독하여 클라이언트의 공개키로 서버측에서 보내온 공개키 (P_s)를 암호화한 $R_{cs}= P_c(P_s)$를 클라이언트에 전송한다.



[그림 4] 공개키 교환 인증

3.2 Client에서 Server로의 Data 전송

제안된 인증 방식은 클라이언트에서 서버로의 데이터 전송이나 질의 응답 시, XML문서 형태로 데이터를 생성하고, 이를 암호화 과정을 거쳐 전송하게 된다. 이 과정을 위하여 먼저 위에서 기술한 전송 준비 단계의 선결조건을 만족하여야 한다. 전송을 위한 질의어나 데이터는 XML 문서생성기를 통과하고, 작성된 XML문서와 DTD문서는 다음의 순서에 따라 전송된다.

① 전송할 XML문서를 MD5로 디지털 서명하여 그 값 $\tilde{m}'$ 를 XML문서 내에 삽입.

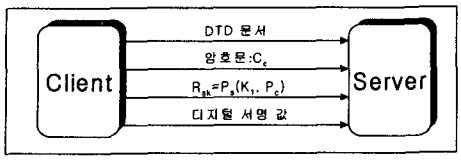
② XML문서를 난수로 발생시킨 DES 대칭키 k_1로 암호화하여 암호문 C_c 생성

③ 암호문 C_c을 한번 더 디지털 서명하여 $\tilde{m}''$ 을 생성.

④ 대칭키 k_1와 클라이언트의 RSA 공개키 P_c 을 서버의 RSA 공개키 P_s로 암호화한 암호문 $R_{sk}= P_s(k_1, P_c)$를 생성.

⑤ 생성된 값 C_c, R_{sk}, $\tilde{m}''$ 와 DTD 문서를 서버로

전송([그림 5])



[그림 5] 데이터 전송

⑥ 서버의 RSA 개인키로 암호문 R_{sk}을 복호화하여 대칭키 k_1와 클라이언트 RSA 공개키 P_c 구함.

⑦ 서버가 가지고 있는 클라이언트의 공개키와 복호화된 공개키 P_c을 비교.

⑧ 암호문 C_c을 디지털 서명 값 $\tilde{m}''$로 이상 유무 확인.

⑨ DES 대칭키 k_1로 복호화 함.

⑩ 복호화 된 XML문서 내의 디지털 서명 값 $\tilde{m}'$ 을 추출하여 문서의 변경 여부 확인.

⑪ 전송된 DTD문서와 XML문서 실행.

위 전송과정 중 ①과 ②의 과정을 통하여 서명 암호화하여 생성된 문서 C_c를 과정 ③에서 다시 서명한 이유는 전자의 XML문서에서의 디지털 서명은 XML 문서의 내용 변경 여부를 확인하기 위함이고, 후자의 경우 암호문이 네트워크를 통해 전송될 때 제 3자의 침입에 의한 정보의 누출이나 네크워크 오류에 의한 문서의 변경을 막기 위해서이다.

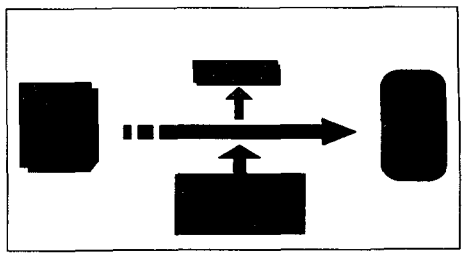
3.3 Server에서 Client로의 Data 전송

서버측에서 클라이언트로의 데이터 전송 방식은 위에서 기술한 방식을 그대로 적용하되 DES 대칭키를 난수로 생성하고, 클라이언트의 RSA 공개키 P_c을 사용하여 암호화하여 결과 값 전달한다.

4. 비교 및 고찰

제안된 XML 문서의 인증 방식은 기존의 MD5에 비하여 문서의 전송 시 정보의 누출에 대한 안정성을 지니고 있다. 이를 위하여 제안된 방법을 기존의 서명방법과 비교하여 본다.

MD5로 디지털 서명방식을 이용한 경우 불순한 의도를 지닌 제 3자는 서버와 클라이언트 사이에서 교환되는 XML문서를 중도에 가로채어 전송되는 XML문서를 자신의 의도대로 변경하고 변경된 XML문서에 대한 MD5 디지털 서명을 수행한 후 수신지로 다시 전송할 수 있다. 이때 문서를 전달받은 측에서는 문서의 변경여부를 확인할 수 없고 또한 문서의 내용이 암호화되어있지 않기 때문에 정보의 유출가능성은 여전히 남아있다.



[그림 6] 내용 변경 및 유출 과정

그러나 제안된 전송방식은 XML 문서의 내용 변경 여부를 확인하기 위한 디지털 서명작업과 함께 디지털 서명된 암호문에 대한 이중적 디지털 서명작업이 수행되었기 때문에 이 네트워크를 통해 전송시 제 3자의 침입에 의한 정보의 누출과 변경, 네크워크 오류에 의한 문서의 변경을 효과적으로 대처할 수 있다.

5. 결론

본 논문에서는 인터넷상의 안전한 XML문서의 전송을 위한 방안을 연구하였다. 제안한 프로토콜은 암호화와 디지털 서명, 인증 알고리즘을 사용하여 프로토콜을 구성함으로서 XML 문서의 안전한 전송이 가능하였고, 웹 어플리케이션 개발에 사용할 수 있는 가능성을 제시하였다. 이후 연구 과제로는 구현한 알고리즘을 실제 웹 환경에 적용 및 구현 방안과 함께 빠른 암호화 처리가 가능하도록 하는 연구가 수행되어야 하리라 사료된다.

참고 문헌

[1] Cnarles F. Goldfarb, Paul Prescod, "The XML Handbook," Prentice-Hall 1998.

[2] Frank Boumphery외 12명, "XML Application," Wrox Press, 1998.

[3] David Wagner. Bruce Schneier "Analysis of the SSL Protocol" The Second USENIX Workshop on Electronic Commerce Proceedings USENIXO Press; 1996

[4] Stephen A. Thomas "SSL & TLS Essentials: Securing the Web" Wiley Press

[5] 조은경외 "S-HTTP 기반의 안전한 웹 시스템 개발" 정보처리학회논문지 제4권 9호 pp2354~2365, 1997.