EKCS: 무선환경에서의 암호화 키 관리를 위한 시스템 설계 및 구현

이현창°, 국윤규, 김운용, 최영근
광운대학교 컴퓨터과학과 분산객체 컴퓨팅 연구실

EKCS: Design and Implementation for Encryption Key Control System on wireless Environment

Hyun-Chang Lee°, Youn-Gyou Kook, Woon-Yong Kim, Young-Keun Choi
DOCOM Lab. Computer Science Dept. Kwangwoon University

요약

정보 인프라가 제공됨에 있어서 정보보호를 위한 암호화의 중요성이 강조되고 있다. 기존의 유선 인터넷환경에서 EDI 의 전자서명과, 기밀성 서비스는 RSA 방식을 통해 이루어지고 있다. 이러한 유선환경의 방식을 무선환경에 적용시키기 위해서는 여러가지 제약조건으로 인하여 속도, 데이터 교환 및 암호화 알고리즘의 적용에 있어서 문제점이 야기된다. 본 논문에서는 무선 환경에서의 효율적인 암호화 키 관리를 위한 EKCS(Encryption Key Control System) 시스템을 제시한다. EKCS 시스템은 적은 메모리와 제한된 환경에서의 처리능력을 가진 무선 환경에 적합한 ECC 암호화 알고리즘과 무선환경에서의 제약성을 극복하고자 데이터의 폭넓은 활용성을 가진 XML 문서를 사용한다. 본 논문은 무선환경에서 컴포넌트 기반의 다중 계층 구조를 갖는 암호화 키 관리 시스템을 설계 및 구현한다.

1. 서 론

개방적 특성을 가지고 있는 인터넷 상에서 기업 뿐 만 아니라 개인의 정보 활용은 보안이라는 문제점을 야기한다[11]. 특히 무선 환경에서의 보안은 단순히 정지된 시스템이나 이동이 제한적인 데이터에 국한된 것이 아닌, 장소나 시간에 구애 받지 않고 어떤 휴대 단말기에서도 데이터 활용이 가능하도록 지원해야 한다. 이중에서 데이터를 전송하거나 저장된 데이터를 비 인가된 변조로부터 보호하기 위한 데이터의 무결성을 보장하여야 한다. 따라서 이동성에 대한 보장 방식과 데이터의 활용 기술이 달라져야 한다[6].

첫째 일반적인 암호화 알고리즘에 있어서 RSA 방식 키 값을 1,024 비트로 제공하며[9], 무선환경에서 증시하는 데이터 사이즈와 대역폭의 한계가 있다. 하지만 타원곡선 알고리즘에서는 160 비트로 구현되므로 두 가지 문제점을 극복을 한다. 따라서 무선환경에서의 데이터 이동에 있어서 안전하고 빠른 속도로 이동을 할 수 있다.

둘째 기존의 전자 상거래 문서 방식인 EDI(Electronic Data Interchange)는 특정 시스템에서 전자문서의 논리적 정보와 물리적 정보의 혼재를 가져왔다. 기존의 전자 상거래 문서 방식은 문서 전체를 인증을 받아서 암호화 할 뿐이다. 하지만 XML 문서처럼 구조화된 문서는 주변장치, 시스템 언어, 응용 프로그램, 네트워크 등과 독립적으로 암호화할 수 있으며 또한 효율적인 문서를 저장 및 검색 한다.

본 논문에서는 XML 문서를 이용한 무선환경에서의 데이터 교환에 있어 메시지에 대한 안정성을 한층 더 강화하기 위한 목적으로 암호화 및 복호화 키 값은 별도의 키 관리인증서버와 암호화 저장서버로 관리하므로 이중의 잠금 장치를 할 수 있다. 따라서 무선의 제한적인 환경을 극복하고 효율적인 키 관리 및 안전한 데이터의 무결성 및 교환의 효율성을 가진다.

본 논문은 무선환경에서의 정보보호를 위한 암호화의 문제점을 보안하기 위한 시스템으로써, 2 장에서는 관련 연구를 살펴보고 3 장에서는 EKCS 암호화 시스템을 설계한다. 4 장에서는 실제적인 구동 방식 및 구현을 기술하고 마지막으로 5 장에서는 결론 및 향후 과제를 기술한다.

2. 관련연구

현재 XML 암호화 방법은 여러 가지가 있다. **Element-Wise Encryption** 은 안전한 XML 문서를 생산하기 위하여 XML 문서의 세션에 대한 암호화 알고리즘을 적용하기 위한 기술이고[3], **XKMS**(XML Key Management

Specification) XML 전자서명 및 XML Encryption 처리시 필요한 공개 키 및 정보 관리를 위한 기술이다[4]. XKMS 는 X-KISS 와 X-KRSS 의 두 부분으로 구성된다[3]. 또한 XML 전자서명과 연동시 기존 PKI 시스템에 대한 복잡성을 클라이언트에게 숨겨 키 관리 부담을 트러스트 서비스에 위임해 구현을 용이하게 하는 것이다.

무선환경에서의 RSA 알고리즘 방식은 무선환경에서 중시하는 데이터 사이즈와 대역폭의 한계가 있다. 따라서 본 논문에서는 160 비트로 구현되는 타원곡선 알고리즘을 사용한다. 주어진 소수에 대하여 유한체의 부분 군을 이용하는 경우는 그 후보가 곱셈 군 밖에 없는 반면 타원곡선 암호시스템의 경우는 주어진 유한체 상에서 정의된 다양한 타원곡선을 선택할 수 있어 풍부한 타원곡선 군을 활용할 수 있다. 특별한 유형의 타원곡선을 제외하고는 유한체의 곱셈 군에서 이산대수문제를 푸는 지수계산 알고리즘을 적용할 수 없어 타원곡선 이산대수문제를 푸는 준 지수시간 알고리즘(subexponential time algorithm)이 존재하지 않는다. 이러한 안정성을 보장받을 수 있는 안전한 암호화 시스템을 설계한다. 다른 암호시스템에 비해 더 짧은 Key 사이즈로 대등한 안전성을 주고 있다는 것이다. 예를 들어 RSA 1024 비트 Key 와 ECC 160 비트 Key 를 갖는 암호시스템은 대등한 안전도를 가진다. [표 1]을 보면 자세히 알 수 있다 [1][2][5].

163	512	1.5	10^{4}
132	768	1.6	10^{8}
160	1,024	1.7	10^{11}
211	2,048	1.10	10^{20}
400	21,000	1.21	10^{11}

[표 1. key Size Comparison]

3. EKCS 시스템 설계

EKCS 시스템 구성은 각각의 모듈로 정의되어 계층적으로 관리를 한다. 각각 분리된 모듈의 내부는 컴포넌트로 구성된다. 구성된 컴포넌트들은 인터페이스로 연결되어 있다. [그림 1]을 보면 실제적인 암호화 키 시스템의 인터페이스 부분은 EKCS 시스템 계층이다. 마지막은 DB 서버 부분으로 암호화 키 및 복호화 키 부분을 저장 관리한다. 즉 키 쌍을 저장 관리하기 위함은 데이터의 값 즉 키 값의 무결성을 보장해 주는 위한 계층이다. 전체적인 시스템 아키텍처는 [그림 1]과 같다.

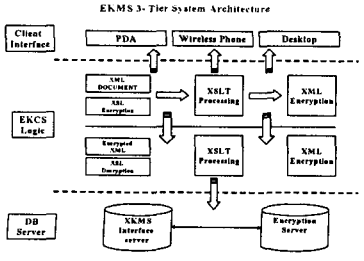

그림 1. EKCS 시스템 구조

3.1 Client Interface 계층

클라이언트 계층인 프리젠테이션 계층은 사용자가 시스템에 접속하는 선택에 따라 서비스를 요청을 할 수 있다. 클라이언트의 타입에 따라 PDA,무선핸드폰, 데스크 탑 등의 다양한 인터페이스 제공한다.

3.2 EKCS(Encryption Key Control System)계층

EKCS 는 전체 시스템 상에서 데이터를 암호화 및 복호화 하는 부분이다. 여기의 암호화 및 복호화 된 데이터 값을 DB 서버(인증서버 및 암호화 DB 서버)로 보내는 역할을 한다. [그림 2]을 보면 각각의 번호는 시스템의 흐름도 이다.

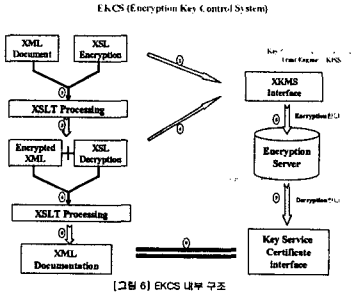

그림 2. EKCS 시스템 계층 및 흐름도

① XML 문서와 XSL 문서를 암호화하는 과정이다. XSL 문서의 엘리먼트 값 안에는 암호화 코드와 복호화 코드 값을 가진다.

② XSL 암호화 문서를 XKMS 인터페이스로 서버 보낸다.

③ XSLT Processing 을 이용해 암호화된 XML 문서로 만든다.

④ 복호화 XSL 문서는 전송된 XKMS 인터페이스 서버에 보내어진다. 신뢰성 엔진인 KISS 를 통해 원본 XML 문서의 안전한 키 값인지를 확인을 한다.

⑤ 암호화 코드 값과 복호화 코드 두 코드 값은 하나의 필드 값 안에 한 쌍의 코드 값을 가진다. 한 쌍의 코드 값은 또 한번의 암호화 처리를 한다. 암호화된 코드 값은 암호화 서버로 관리된다.

⑥ 복호화 하기 위해 XSL 에 복호화 코드를 XSLT Processing 을 이용하여 원본의 데이터인 XML 문서를 만든다.

⑦ 암호화 서버는 클라이언트에게 서비스를 하기 위해서는 복호화 작업을 해야한다. 그 작업은 키 서비스 인증 인터페이스를 통해 작업을 한다.

⑧ 원본의 XML 문서를 만드는 과정이다.

3.3 Encryption DB Server 계층

[그림 2]의 Encryption DB Server 와 Key Service Certificate Interface Server 는 클라이언트에게 안전한 키 서비스를 해 주기 위한 자바 보안 구조 중에 JSSE(Java Secure Soket Extension)을 사용한다. JSSE 는 SSL 을 사용하기 위해 API 로 제공된다. SSL 은 소켓 프로토콜이기 때문에 API 는 자바의 표준 소켓 규칙을 따르는 클래스를 제공한다.

4. EKCS 시스템 구현

4.1 EKCS 시스템 구현

EKCS시스템은 기존의 암호화 시스템에 비해 이중 암호화 및 인증 작업을 한다. 이렇게 함은 기존의 암호화 시스템을 보완하기 위해 키 코드 값 관리의 편리함과 함께 안전한 키 코드 값을 보완 하고자 하는 시스템이다. [그림 3]은 XML 문서의 암호화 부분을 체크하고 암호화 키 코드 값으로 암호화 하고자 한다. XML문서를 암호화 하는 것은 기존의 일반 문서를 암호화 하는 것과는 지양하는 바가 조금 다르다. XML은 특정 부분을 암호화 하고 이에 적용한 알고리즘과 암호화 키에 대한 정보를 새로운 XSL엘리먼트로 정의하여 암호화한 특정부분을 대체 할 수 있다. [그림 3]은 엘리먼트의 전체 부분을 암호화 할 수 있지만, 일부분만을 암호화 할 수 있도록 했다.

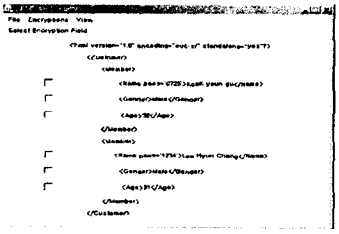

그림 3. XML 문서 부분 암호화 과정

암호화 하려는 XML 문서를 개별적인 키 값을 넣어 줌으로 해서 자신만이 아는 키 값을 가진다. 정보를 가진 XSL문서는 암호화 키 값을 가지는 과정 [그림 4]에서 보여 진다.

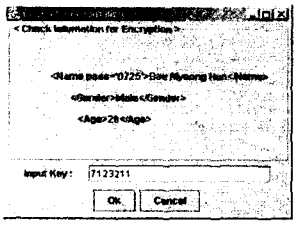

그림 4. 암호화 키 입력 부분

암호화 한 값을 다시 암호화 과정을 통해 Encryption DB server로 관지한다. Encryption DB server는 Key Certificaties Interface부분으로 키 암호화를 복호화 시켜 보내진다. [그림 5]는 DB 계층을 관리하는 것을 보여준다.

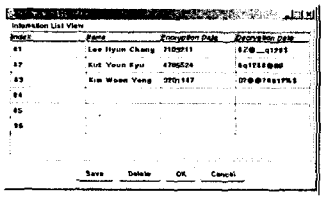

그림 5. EKCS DB 시스템 계층

5. 결 론

본 논문은 무선환경에서의 제한된 환경을 극복하고자 한다. 데이터 교환에 있어서의 편의성을 제공하는 XML 문서를 사용하여, XSLT processor 작업 과정을 통하여 암호화 및 복호화 문서변환 작업을 한다. 적은 메모리와 제한된 처리능력을 가진 무선 환경에 적합한 ECC 암호화 알고리즘과 무선환경에서의 적은 량의 Key 값을 사용하기 위하여 ECC 알고리즘으로 암호화를 한다. 암호화 하는 부분은 XML 안에 있는 KeyInfo 값을 추출하여 XKMS Interface로 Key 값을 관리하며 Key 값을 또 한번의 암호화를 과정을 가진다. Key 관리를 편하게 하기 위해 Key DB Server에 키 값을 저장 및 관리한다. 따라서 본 논문은 안전한 암호화 Key 관리 목적과 동시에 빠른 데이터의 이동성을 보장하고자 하는 것이 목적이다.

향후 본 논문의 시스템은 클라이언트가 증가함에 따라 DB 서버 계층에서 데이터의 일괄처리를 보완 해야 한다. 또한 웹 서비스 및 모바일 에이전트상에서의 실질적인 암호화 시스템을 적용 하고자 한다.

참고 문헌

[1] http://www.isg.rhul.ac.uk/~sdg/ecc.html
[2] http://www.newcom.com.au/products/
[3] Blake Dournaee, "XML Security", 2002 RSA Press
[4] H.M. Deitel, P.J. Deitel, T.R. Nieto, T.M. Lin, P. Sadhu, "XML How To Program", Prentice Hall, 2001
[5] Lawrence C.Washington " Elliptic Curve :Number Theory and Cryptography" June 2003
[6] Robert Bauchle, Fred Hazen, http://www.searchsecurity.com Dec 26, 2000
[7] R.G.Bartlett and M.W.Cook XML Security Using XSLT 2002 IEEE
[8] Sun Microsystems, "JSSE™ 1.4 Reference Guide", http://java.sun.com/jsse/
[9] Takeshi Imamura,Blair Dillaway, Ed Simon http://www.W3.org/TR/2002/PR-xmlenc-core-20021003/
[10] Yu Feng, Jun Zhu, "Wireless Java Programming with J2ME", SAMS, June. 2001.
[11] 이옥연, " ETRI 최신기술 이전 동향" http://www.etri.re.kr/news/01-08/etri05.htm 2001.8