한국산학기술학회논문지
Vol. 10, No. 2, pp. 377-387, 2009

전자 서명과 시점 확인 서비스의 결합

장혜진[1*]

Integration of Timestamp Service into Digital Signatures

Hai Jin Chang[1*]

요 약 전자 서명은 서명 대상 데이터의 내용에 대한 완전성 보장과 서명자의 신원 확인 기능을 제공한다. 하지만 서명자의 서명만을 포함하는 기본적인 구조의 전자 서명은 서명 시점에 대한 정확성을 보장하지 못하며 서명에 대한 장기간의 신뢰를 제공하지 못한다. 본 논문은 전자 서명 메커니즘에 시점 확인 서비스(timestamp service)를 결합하여 서명 시점에 대한 정확성을 보장하며 수명이 긴 전자 서명의 생성과 검증 서비스를 제공하는 시스템을 제안한다. 제안된 시스템은 전자 상거래 계약, 문서 보존 서비스, 송장(invoice) 응용 등과 같이 기존의 전자 서명보다 장기간의 신뢰를 보장하는 서명을 요구하는 다양한 응용 분야에서 사용될 수 있다. 제안된 시스템은 한국표준과학연구원에서 운영되는 시점 확인 서비스 시스템과 연동하여 테스트되었다.

Abstract Digital signatures not only provide a way of guaranteeing the integrity of data but also establish the identity of the signer. However, basic digital signature format which contains only the signature of the signer does not guarantee the correctness of its creation time, and it can not remain valid over long periods. This paper proposes a system which integrates timestamp service into digital signatures. The system provides online services for the creation and verification of long term digital signatures which can give the guarantee of the correctness of their creation times and can be proved to be valid over long periods. The proposed system can be used in the various areas such as e-commerce contracts, document archival services, and invoice applications, which requires long term digital signatures. The proposed system is tested with the KRISS timestamp service system.

Key Words : digital signature, timestamp service, long-term digital signature, ES-T, ES-C, RFC 3369, RFC 3126

1. 서론

전자 서명은 온라인 뱅킹(banking), 온라인 전자 상거래, 그룹웨어 결재 시스템 등의 다양한 분야에서 사용된다. 전자 서명은 서명 내용에 대한 완전성(integrity) 보장과 서명자의 신원 확인(authentication)의 기능을 제공하기 때문이다.

전자 서명에는 서명자가 그 전자 서명을 생각한 시점에 대한 정보가 포함된다. 전자 서명의 기본 규격을 규정하는 RFC 3369[1]에 따르면, 전자 서명은 서명 시점을 저장하기 위해 signing-time이라는 속성을 가져야 하며, 그 속성은 단 하나의 시점 값을 가져야만 한다.

하지만 서명자 본인의 서명만을 포함하는 기본적인 형태의 전자 서명은 그 자체만으로는 서명 시점에 대한 신뢰를 제공할 수 있는 객관적 증거를 제공하지 못하며 서명에 대한 장기간의 유효성(long term validity)을 제공하지도 못한다는 제약을 갖는다. 예를 들어, 시각 t'에 서명자의 전자 서명용 인증서가 유효하다면, 서명자는 현재 시각 t에 자신의 컴퓨터의 시각을 t'로 조작한 후 그 컴퓨터에서 서명자 본인의 서명만을 포함하는 기본적인 형태의 전자 서명을 생성할 수 있다. 즉 어떤 추가적인 조건이나 증거가 없는 한 기본적인 형태의 전자 서명은 그것이 포함하고 있는 서명 시각에 그것이 생성되었다는 증거를 제공하지 못한다.

또한, 서명자가 서명 시점을 속이려는 의도를 갖지 않으며 서명에 사용된 인증서의 유효 기간이 만료되지 않

본 논문은 상명대학교 교내 선발 과제로 연구되었음.
[1]상명대학교 컴퓨터소프트웨어공학과
*교신저자: 장혜진(hjchang@smu.ac.kr)
접수일 09년 01월 12일 수정일 09년 01월 30일 게재확정일 09년 02월 18일

았으며 철회되거나 효력이 정지되지 않은 경우라 하더라도, 그 서명이 서명자 본인의 서명만을 포함하는 기본적인 형태의 전자 서명이라면, 상당한 시간이 흘러 서명자의 공개키 인증서의 유효기간이 지나버린 후에는 그 전자 서명을 유효하다고 할 수 없다는 문제가 생긴다. 왜냐하면 서명자 본인의 서명만을 포함하는 기본적인 형태의 전자 서명으로는 서명자가 그 서명을 자신의 공개키 인증서의 유효 기간 내에 생성한 것인가에 대한 객관적인 증거가 없기 때문이다.

서명자 본인의 서명만을 포함하는 기본적인 전자 서명을 사용하는 경우에도 서명 시점에 대한 추가적인 조건이나 근거가 존재한다면 서명 시점에 대한 확인을 할 수 있는 경우가 존재한다. 예를 들어, 온라인 뱅킹 등에서 사용자의 신원을 확인하기 위하여 서버가 새로운 논스(nonce)[2]를 생성하여 사용자에게 보내고 사용자가 그 논스에 대한 자신의 서명을 즉시 생성하여 서버로 돌려보내는 방식의 로그인 프로세스를 가정하자. 그 프로세스가 서명자 본인의 서명만을 포함하는 기본적인 형태의 전자 서명을 사용하더라도 서버가 사용자에게 논스를 생성하여 보내고 몇 초 정도의 제한된 시간 내에 그 논스에 대한 올바른 전자 서명을 사용자가 돌려보내는 경우에만 그 사용자의 로그인을 허락한다면 그 프로세스는 문제가 없다. 왜냐하면 로그인 시도마다 새롭게 생성되는 논스에 대한 전자 서명을 논스 생성 이후 몇 초 이내에 돌려받았다는 사실은 사용자의 서명이 논스 생성 이후로부터 몇 초 이내의 시점에 생성되었다는 간접적인 증거가 되기 때문이다.

하지만, 일반적으로 사용자의 전자 서명만을 포함하는 기본적인 전자 서명만으로는 서명 시점에 대한 신뢰와 서명의 장기간 신뢰를 제공하기 어렵다. 전자 상거래 계약, 디지털 문서 보존 서비스, 송장(invoice) 응용과 같은 분야들은 서명 시점에 대한 정확성 및 그 서명이 장기간에 걸쳐 유효함을 보장받아야 한다.

시점 확인 서비스(timestamp service)란 임의의 디지털 데이터에 대하여 그것이 어떤 특정한 시점에 존재하였음을 증명해주는 서비스를 의미한다[2]. 어떤 문서나 데이터가 어떤 시점에 존재하였는가에 대한 엄밀한 판정이 필요한 경우가 많다. 예를 들어 저작권 분쟁이나 특허 분쟁의 경우, 그 분쟁 대상 내용에 대하여 누가 먼저 독창성을 가지고 있었는가를 증명하는 것이 분쟁의 중요한 승패 판정 기준이 된다. 국내에서는 한국표준과학연구원에서 RFC 3161[3]을 준수하는 웹 기반의 시점 확인 서비스 시스템[4]이 운영되고 있다. 한국표준과학연구원은 공식적으로 국가가 공인하는 한국 표준 시각을 제공하고 있다. 디지털 콘텐츠나 문서에 대한 존재 시점의 확인 검증을 요구하는 다양한 잠재 응용 분야들이 존재한다. 시점 확인 서비스는 전자 공증 시스템(digital notary system)[5]과 같은 다양한 응용 분야에 활용될 수 있다.

본 논문은 전자 서명에 시점 확인 서비스를 결합하여 서명 시점에 대한 정확성을 보장하고 장기간 유효한 전자 서명을 생성하고 검증하는 서비스를 제공하는 시스템을 제안한다. 제안된 시스템은 사용자가 서명자 본인만의 서명을 포함하는 기본적인 구조의 전자 서명을 생성한 후, 그 전자 서명에 공신력 있는 기관인 한국표준과학연구원에서 운영하는 시점 확인 서비스 시스템[4]이 발행하는 타임스탬프 토큰(timestamp token)을 결합하고, 서명의 검증에 필요한 모든 인증서들과 CRL(certificate revocation list)에 대한 정보를 추가적으로 결합하여 서명 시점에 대한 정확성을 보장이 가능하며 장기간 유효한 전자 서명을 생성한다. 생성된 전자 서명에 결합된 타임스탬프는 사용자의 서명 시점에 대한 객관적인 근거가 될 수 있으며, 서명 이후 오랜 시간이 흘러 서명 검증 시점에 사용자의 인증서가 유효 기간이 지났더라도 서명 시점 당시에 사용자의 서명이 유효하였다는 사실을 증명할 수 있다면 그 사용자의 서명을 유효하다고 판정할 수 있도록 한다.

시점 확인 정보가 결합된 장기간 유효한 전자 서명의 생성과 검증에 필요한 시스템을 기관이나 전자 서명 응용 시스템마다 개별적으로 제작하여 사용할 수도 있다. 하지만 시점 확인 정보가 결합된 장기간 유효한 전자 서명의 생성과 검증 시스템은 상당한 복잡성을 가지며 높은 신뢰도를 확보해야 하므로 기관이나 응용별로 그런 시스템을 개별적으로 제작하여 사용하는 것은 많은 비용과 노력을 중복해서 요구한다고 할 수 있다. 본 논문이 제안하는 시스템은 수명이 긴 전자 서명 및 서명 검증 서비스를 제공하여 그런 비용과 노력을 줄여줄 수 있다.

본 논문에서 서명자의 서명만이 포함된 기본적인 형태의 전자 서명이란 RFC 3126[3]이 규정하는 기본 전자 서명 규격인 ES 포맷(electronic signature format)을 의미한다. 본 논문이 제안하는 시스템이 생성하는 타임스탬프가 결합된 긴 수명의 전자 서명이란 RFC 3126이 규정하는 ES-C 포맷(electronic signature with complete validation data references)을 의미한다.

본 논문의 제 2장은 관련 기술들에 대한 것이다. 제 3장은 본 논문이 제안하는 전자 서명과 시점 확인 서비스의 결합 시스템인 KES-T(Kriss extended signature with timestamp) 시스템과 그것의 구현 및 테스트를 위한 KES-T 라이브러리와 KES-T 데모 시스템에 대하여 기술한다. 제 4장은 결론이다.

2. 관련 기술들

전자 서명의 신뢰도를 강화하기 위한 방법에는 RFC 3126[6]과 같이 전자 서명의 규격 자체를 보안적으로 확장하고 강화하는 방법과 Verisign 사의 ACS(authenticated content signing)[7]과 같이 전자 서명을 배포하고 사용하는 비즈니스 로직의 강화를 기초로 하는 방법이 있다.

2.1 전자 서명 국제 표준 규격들

본 논문이 제안하는 서비스 시스템의 전자 서명 형태는 국제 표준을 준수한다. 전자 서명에 관련된 대표적인 국제 표준 규격으로는 공개키 기반구조(public key Infrastructure), 인증서 및 CRL(certificate revocation list)에 대하여 규정하는 RFC 2459[8], 전자 서명 및 암호화된 자료 구조 등에 대한 표준 기본 규격을 규정하는 RFC 3369[1], RFC 3369를 수용하면서 동시에 장기간 유효성을 보장하기 위한 강화된 서명 규격들에 대하여 규정하는 RFC 3126[6], 인증서의 상태를 온라인으로 검증하기 위한 프로토콜인 OCSP(online certificate status protocol)에 대하여 규정하는 RFC 2560[9] 등이 있다.

시점 확인 서비스에 대한 표준화 작업은 현재 국제 표준화 기구인 ISO/IEC JTC 1/SC 27과 IETF(www.ietf.org) 산하 PKIX에서 진행되고 있다. IETF의 표준안은 시점 확인 서비스의 기본 사항인 시점 확인 요청 및 시점 확인 응답에 대한 자료 구조와 프로토콜을 규정하고 있다. 시점 확인 시스템과 직접 관련된 IETF 표준안들은 RFC 3161[3] 등이 있으며, 정확한 시각 정보를 네트워크상에서 통신하기 위한 관련 표준안에는 RFC 1305[10], RFC 2030[11] 등이 있다.

전자 서명에 관련된 국제 표준들은 전자 서명의 포맷, 생성, 그리고 검증에 대한 최소한의 것들을 규정하지만 그 표준을 이용하는 서비스 모델들의 종류 및 서비스 모델을 구현하는 서버나 클라이언트의 규격이나 역할, 그리고 비즈니스 로직들과 같은 부분들을 규정하지 않는다.

RFC 3369[1]는 서명의 신뢰성을 높이는 방법의 하나로 대항 서명을 지원한다. RFC 3369가 규정하는 전자 서명 구조인 signed-data 구조의 content 구조의 signerInfos 구조의 unsignedAttrs 속성은 대항 서명 속성(counter signature attribute)을 원소로 포함될 수 있다. 대항 서명 속성에는 signed-data 내의 SignerInfo 값의 signatureValue 필드의 DER 인코딩(encoding) 바이트들에 대한하나 이상의 서명들을 규정한다. 즉, 대항 서명(countersignature) 방식이란 개념적으로 서명 자체를 다시 서명 대상 데이터에 포함시켜 직렬적으로 서명하는 것을 의미한다. 서명자 A의 서명에 대하여 서명자 B의 대항 서명이 이루어지는 경우 A의 서명 시각을 포함한 내용에 대하여 B가 대항 서명한다면 A는 B와 공모하지 않으면 서명 대상의 내용 및 서명 시점을 고칠 수 없다. 대항 서명은 그룹웨어의 결재 모듈 등에서 사용된다. 대항 서명 방식에서는 자신보다 나중에 서명한 서명자들과 공모가 없다면 서명 대상 내용이나 시각을 변경할 수 없지만 대항 서명에 참여한 서명자들에 대한 신뢰를 확인하기 어렵다면 대항 서명을 신뢰하기 어렵다.

전자 서명과 시점 확인 서비스를 결합하는 본 논문의 방법에서는 전자 서명 구조인 signed-data 구조의 content 구조의 signerInfos 구조의 unsignedAttrs 속성의 원소의 하나로 타임스탬프 속성(signature timestamp attribute)을 포함하며, 그 타임스탬프 속성은 한국표준과학연구원에서 운영하는 시점 확인 서비스 시스템[4]의 타임스탬프 토큰을 담는다. 즉, 본 논문의 시스템은 한국표준과학연구원에서 운영하는 시점 확인 서비스 시스템의 타임스탬프 토큰을 포함하는 서명을 생성한다. 전자 서명에 타임스탬프 토큰을 포함하는 방식은 시점 확인 서비스의 운영이 공신력을 가진 인가된 기관에 의해서 이루어진다는 점에서 일반적인 대항 서명 방식보다 더 높은 신뢰도를 갖는 서명을 생성하는 방식이라 할 수 있다.

2.2 Verisign의 ACS (Authenticated Content Signing)

Verisign 사(www.verisign.com)는 널리 알려져 있는 공개키 인증서(public key certificate) 관련 솔루션 판매 및 서비스 회사이다. Verisign사의 전자 서명은 웹서버 인증, 데이터 전자 서명, 서명된 코드의 생성 및 검증 등의 다양한 분야에서 널리 사용되고 있다. Verisign 사는 기존의 전자 서명 방식이 가진 문제들을 해결하기 위해 제 3세대 서명 방식(authenticated content signing)[7]이라 명명된 온라인 서명 및 서명 검증 서비스를 제공한다. Verisign 사의 제 3세대 서명 방식은 전자 서명의 형태 자체를 보안적으로 강화하기 위한 RFC 3126[6]과 같은 표준 규격들과 달리, 인증서 발행 및 서명에 대한 비즈니스 로직의 강화에 의해 전자 서명의 서명 시점 등에 대한 신뢰도를 강화한다. 그 비즈니스 로직을 요약하면 다음과 같다.

(단계 1) 사용자는 자신의 공개키와 그에 대응되는 개인키의 쌍 $<P_u, P_r>$를 생성한 후, Verisign 사에게 공개키 P_u와 사용자의 신상 정보를 보내 공개키 인증서 Cp_u를 발급받는다.

(단계 2) 사용자는 개인키 P_r로 콘텐츠(또는 코드)에 서명하고, 전자 서명 결과를 Verisign 사에

게 온라인으로 보낸다.

(단계 3) Verisign 사는 사용자가 보낸 서명된 콘텐츠(또는 코드)의 서명을 검증하여 그것이 Verisign 사에서 발행된 유효한 공개키 인증서에 대응되는 개인키로 서명된 것인가를 확인한다.

(단계 4) 단계 3의 확인이 성공하면 Verisign은 단계 4.1부터 단계 4.4까지를 수행한다.

(단계 4.1) 새로운 공개키와 개인키의 쌍 <P_u', P_r'>를 생성한다.

(단계 4.2) P_u'에 대한 공개키 인증서 Cp_u'를 생성한다. 여기서 Cp_u'의 소유자(owner)는 Versign 사이다.

(단계 4.3) Verisign 사는 P_r'로 콘텐츠(또는 코드)에 서명하고 P_r'를 폐기한다. 서명 결과에는 인증서 Cp_u'가 포함된다.

(단계 4.4) Verisign은 P_r'로 서명된 콘텐츠(또는 코드)를 서명자에게 송신한다.

위 절차가 성공적으로 종료되면 사용자는 Verisign 사가 직접 서명한 서명 결과를 갖게 된다. 여기서 단계 4.3의 인증서 Cp_u'는 VerSign 사의 자가 서명 루트 인증서(self-signed root certificate)중 하나와 인증서 경로(certificate path)를 형성하는 인증서이다.

Verisign 사의 제 3세대 서명 방식은 서명된 콘텐츠(또는 코드)의 식별과 서명자(즉 사용자)의 식별을 구분한다는 개념을 사용하고 있다. 위 절차에 따르는 서명 결과는 사용자의 공개키 인증서가 아니라 Verisign 사에서 발행한 인증서인 Cp_u'를 이용하여 검증될 수 있다. Verisign 사의 루트 인증서들(root certificates)은 널리 알려져 있는 인증서들이므로 Verisign 사의 전자 서명은 개인 서명자의 전자 서명보다 검증이 편리하다. Verisign 사의 루트 인증서는 Microsoft 사의 운영 체계들에 기본으로 포함되어 있다. 즉, Verisign 사의 3세대 전자 서명 방식은 결과적으로 서명자가 아니라 Verisign 사가 서명한 결과를 생성하므로 검증이 쉽다는 장점을 갖는다. Verisign 상의 제 3세대 서명 방식은 또한 서명 시점의 불법적인 조작이 어렵다는 장점도 갖는다. 왜냐하면 서명 시점의 조작은 Verisign 사와 서명자가 공모해야만 가능하기 때문이다. Verisign 상의 서명 방법은 서명할 때마다 새로운 키 쌍과 인증서가 생성되어 사용되므로, 서명된 결과에 문제가 있는 경우에도 해당 인증서만을 철회하면 된다는 장점도 갖는다. Verisign 사의 3세대 서명 방식은 기존 서명 방식의 여러 가지 문제점들을 해결하고 있지만, Verisign 사의 3세대 서명 방식은 Verisign 사의 루트 인증서를 사용하며 회사의 수익 모델과 연관된 방법이라는 문제를 갖는다.

본 논문이 제안하는 방식은 시점 확인 서버 운영 주체인 TSA(timestamp authority)로 국가 기관이 보장하는 공신력을 가진 기관을 사용한다. 또한 본 논문의 방식은 전자 서명 및 시점 확인 서비스에 대한 공개된 국제 표준 규격들을 준수하며 특정 기관이나 회사의 전자 인증서 관련 비즈니스 로직에 종속되지 않는 방식이다.

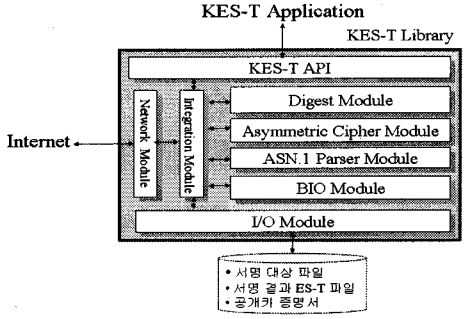
3. KES-T 시스템

KES-T 라이브러리는 본 논문이 제안하는 KES-T(Kriss extended signature with timestamp) 시스템에게 긴 수명을 가진 전자 서명의 생성과 검증을 위한 API(application programming interface)를 제공하기 위한 목적으로 MS Windows XP 상에서 ANSI C/C++ 언어로 개발되었다. KES-T 데모 시스템은 KES-T 시스템의 테스트 및 시연을 위한 시스템이다.

3.1 KES-T 라이브러리

KES-T 라이브러리는 MS Windows XP 뿐 아니라 Linux나 Unix에서도 사용이 가능하도록 GUI(graphic user interface) 관련 기능을 갖지 않으며, MFC(microsoft foundation class) 라이브러리와 같은 운영 체계 종속적인 기능을 사용하지 않도록 설계되었다.

KES-T 라이브러리는 그림 1과 같은 구조를 갖는다. 그림 1에서 입출력 모듈(i/o module)은 파일 시스템에 대한 입출력을 담당한다. 네트워크 모듈(network module)은 인터넷상의 다른 컴퓨터들과 메시지를 송수신하는 기능을 담당한다. 예를 들어, TSA(timestamp authority) 서버에게 타임스탬프 토큰을 요청하고 시점 확인 응답을 수신하는 등의 네트워크 통신 기능을 담당한다. TSA가 제공하는 시점 확인 응답에는 성공적으로 타임스탬프 토큰이 발행된 경우에는 그 타임스탬프 토큰이 포함되며 그렇지 않은 경우에는 오류 상태 정보가 포함된다.

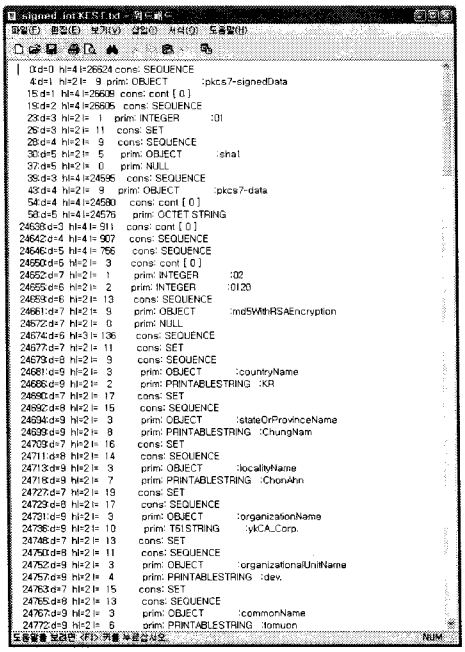


[그림 1] KES-T Library의 구조

그림 1의 비대칭 암호 모듈(asymmetric cipher module)은 축약 모듈 및 통신 모듈을 이용하여 시점 확인 정보가 결합된 서명 결과를 실제로 생성하고 검증하는 역할을 한다. 비대칭 암호 모듈은 RSA[12] 등의 알고리즘을 제공한다. 축약 모듈(digest module)은 SHA1[13], MD5[14] 등의 다양한 축약 알고리즘들을 제공한다. BIO 모듈은 보안 알고리즘들 간의 입력과 출력을 연결하는 기능을 담당한다. 통합 모듈(integration module)은 목표 시스템의 주 로직(main logic)을 담당하는 모듈이다. 서명/서명 검증 모듈, 축약 모듈, 비대칭 암호 모듈 등에서 사용되는 보안 알고리즘들은 OpenSSL[15]을 이용하여 구현되었다.

KES-T API 모듈은 축약 값을 계산하는 축약 모듈, 비대칭 암호 모듈 등의 기능을 결합하여 통합된 기능을 제공하는 통합 모듈(integration module)을 이용하여 응용 프로그램에게 시점 정보가 포함된 전자 서명의 생성 및 검증 등의 응용 프로그래밍 인터페이스를 제공한다.

ASN.1 파서 모듈(ASN.1 parser module)은 서명의 중간 결과 또는 최종 결과를 ASN.1[16,17] 형태로 파싱하는 기능을 제공한다. 전자 서명 표준 규격들은 ASN.1 규격으로 기술되며 ASN.1 파서를 이용하면 전자 서명의 결과의 구조에 대한 검증을 쉽게 할 수 있다. 다음의 그림 2는 KES-T 라이브러리에 의해 생성된 서명 샘플에 대한 ASN.1 파서의 파싱 결과를 출력한 화면이다.

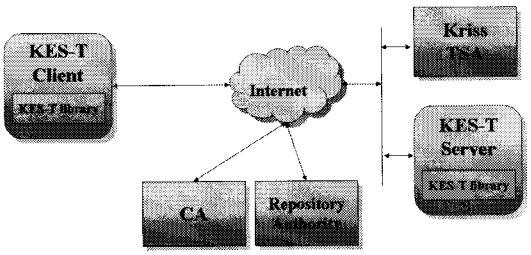


[그림 2] 서명 샘플의 ASN.1 파싱 결과

3.2 KES-T 시스템

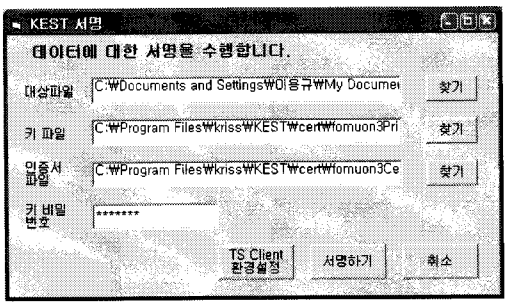
다음 그림 3은 KES-T 시스템의 구성도이다. KES-T 시스템은 KES-T 클라이언트, KES-T 서버, 그리고 한국표준과학연구원에서 운영하는 시점 확인 시스템인 Kriss TSA(timestamp service authority)[4], 인증서들의 발급 및 조회 서비스를 제공하는 CA(certificate authority), 그리고 RA(repository authority)등과 연동하여 동작한다. RA는 CA들이 발행한 CRL(certificate revocation list)들에 대한 검색 서비스를 제공한다. RA는 CA의 루트 인증서들의 철회 정보를 담아 공표하는 자료 구조인 CARL(certificate authority revocation list)들에 대한 검색 서비스도 제공한다. 개념적으로 CARL은 CRL의 일종으로 볼 수 있다.

그림 3에서 KES-T 클라이언트는 사용자의 전자 서명 생성 요청 또는 검증 요청을 KES-T 서버에게 전달하고, KES-T 서버는 Kriss TSA로부터 타임스탬프 토큰을 발행받아 타임스탬프 토큰이 결합된 긴 수명의 전자 서명을 생성한다.



[그림 3] KES-T 시스템의 구성

다음 그림 4는 KES-T 데모 시스템의 KES-T 클라이언트의 서명 생성 요청 처리 화면이다. 사용자가 서명 생성 요청 처리 화면에 나타난 인자들을 입력하고 서명하기 버튼을 클릭하면 다음 3.2.1절의 전자 서명의 생성 절차가 시작된다.



[그림 4] KES-T 클라이언트에서의 전자 서명 생성 요청 화면

3.2.1 KES-T 시스템의 전자 서명 생성 절차

전자 서명 생성에 필요한 인자는 서명 대상 파일, 사용자의 개인키 파일, 사용자의 인증서 파일, 사용자 개인키의 비밀번호, 그리고 전자 서명 정책이다. KES-T 데모 시스템의 KES-T 클라이언트에서는 타임스탬프 클라이언트의 환경 설정 인자들 중 타임스탬프 응답의 생성에 걸리는 시간 제약(timeout)값의 설정에 따라 전자 서명의 정책이 결정된다. 제 3.2.3절에 전자 서명 정책에 대한 자세한 기술이 나온다. KES-T 시스템이 시점 확인 정보가 포함된 전자 서명구조를 성공적으로 생성하는 절차는 다음 그림 5의 순차 다이어그램(sequence diagram)과 같다.

그림 5에서 최종적으로 생성되는 전자 서명은 RFC 3126[6]이 규정하는 ES-C(electronic signature with complete validation data references) 규격을 준수한다. KES-T 시스템의 전자 서명 생성 절차는 KES-T 클라이언트가 RFC 3126[6]이 규정하는 ES 구조의 전자 서명 즉 서명자의 서명만을 포함하는 기본적인 전자 서명을 생성하여 KES-T 서버에게 보내는 것에서 시작된다. KES-T 클라이언트가 ES 구조를 생성하는 세부 절차는 다음과 같다.

(단계 1) signed-data 구조 생성 - 사용자가 입력한 인자들을 이용하여 RFC 3369가 규정하는 signed-data 구조를 생성한다.

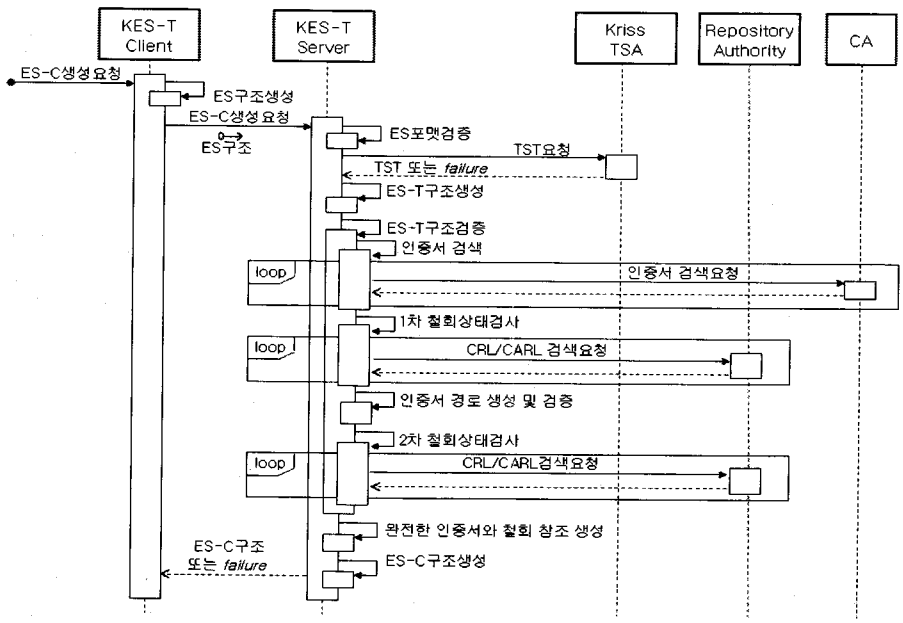
(단계 2) 속성 SignaturePolicyId와 속성 SigningCertificate 추가 - 서명 정책 식별자를 표현하는 SignaturePolicyId 속성과 서명자의 공개키 증명서를 담는 속성인 SigningCertificate 속성을 생성하여 생성된 signerInfo 구조의 signedAttrs 구조에 추가한다.

(단계 3) 서명 속성들의 축약값 계산 - 사용자의 서명 대상 데이터와 SignerInfo 구조의 서명 속성들의 모든 요소들에 대한 축약값을 구한다.

(단계 4) 서명값 생성 - 단계 2에서 구한 축약을 서명자의 개인키(private key)로 암호화하여 그 결과로 생성된 옥텟 문자열(octet string)을 SignerInfo 구조의 signature 필드에 넣는다.

KES-T 클라이언트가 생성한 ES 구조를 전달받은 KES-T 서버는 KES-T 클라이언트가 보낸 ES 구조의 포맷 즉 문법 구조를 검증하고, TSA에게 타임스탬프 토큰을 요청하여 ES 구조에 타임스탬프 토큰이 결합된 ES-T(electronic signature with time) 구조[6]를 생성한다. 타임스탬프 토큰은 전자 서명을 나타내는 구조인 signed-data 구조의 content 구조의 signerInfos 구조의 unsignedAttrs 구조의 한 요소 속성으로 결합된다[1, 6]. 개념적으로 타임스탬프 토큰은 사용자의 전자 서명에 대한 시점 확인 서비스 시스템의 서명이라고 할 수 있다.

ES-T 구조는 장기간 신뢰를 보장할 수 있는 전자 서명을 위한 핵심 구조이다. 서명자가 선택한 서명 정책에 따



[그림 5] KES-T 시스템의 전자 서명 생성 절차

라 타임스탬프 요청으로부터 타임스탬프 발행까지의 간격에 대한 시간 제약(time limit) 조건이 달라질 수 있다. 예를 들어 서명자가 선택한 서명 정책이 1초 이내의 시간 제약을 요구한다면 1초 이내에 타임스탬프 토큰이 발행되는 경우에만 다음 단계로 진행되며 그렇지 않은 경우 KES-T 클라이언트에게 실패(failure) 메시지가 전달되게 된다. ES로부터 ES-T 구조를 생성하는 시간 간격은 짧을수록 바람직하다[6]. 그림 5의 순차 다이어그램에는 다이어그램의 지나친 복잡성을 줄이기 위하여 서명 생성의 매 단계마다 발생하는 실패 메시지의 전달이 생략되어 있다.

ES-T 구조를 생성한 이후에는 다음과 같은 세부 단계들로 구성된 ES-T 구조 검증 절차가 진행된다.

(단계 1) 인증서 검색 - 서명자의 인증서로부터 그 인증서를 발행한 CA의 루트 인증서까지의 인증서 경로상의 인증서들을 CA(들)로부터 검색해 가져온다. 서명자의 인증서는 ES 구조 내에 포함되어 있다. 검색해 가져온 인증서들의 유효 기간은 서명자의 서명 시점을 포함하고 있어야 한다.

(단계 2) 1차 철회 상태 검사 - CRL 또는 CARL의 검색을 통해 서명자의 서명 시점에 서명자의 인증서와 단계 1에서 검색해온 인증서들이 철회되거나 효력 정지 상태에 있지 않음을 확인한다.

(단계 3) 인증서 경로 생성 및 검증 - 단계 1에서 검색한 인증서들을 이용하여 서명자의 인증서로부터 CA의 루트 인증서까지의 인증서 경로(certification path)가 단절되지 않고 형성됨을 검증한다.

(단계 4) 2차 철회 상태 검사 - 다시 CRL 또는 CARL의 검색을 통해 서명자의 서명 시점에 서명자의 인증서와 단계 1에서 검색해온 인증서들이 철회되거나 효력 정지 상태에 있지 않음을 확인한다.

ES-T 구조의 검증 절차가 성공적으로 완료되면, 서명에 관련된 완전한 인증서와 철회 참조들(complete certificate and revocation references)을 생성하여 ES-T 구조에 결합하면 ES-C 구조가 완성된다. 인증서 참조(certification reference)및 CRL 참조(CRL reference)에 대한 규격은 RFC 3126 [6]이 규정하고 있다.

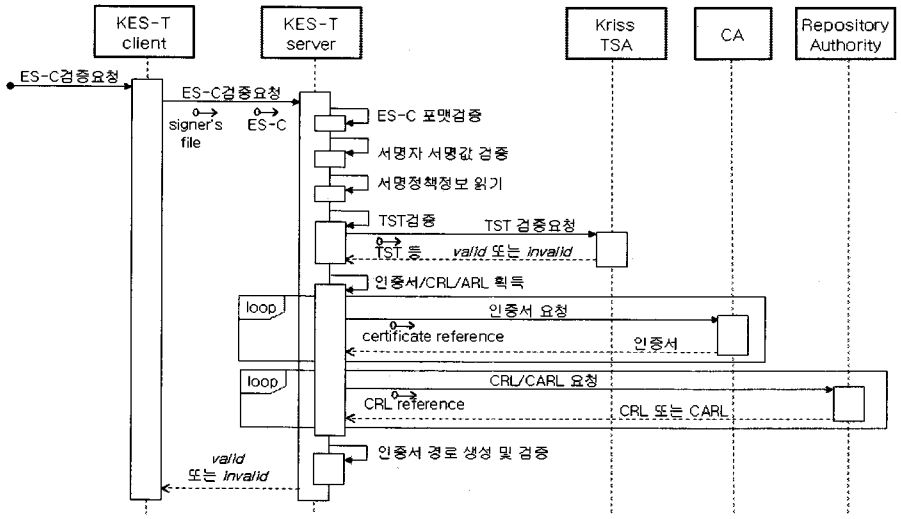
위 ES-T 구조의 검증 절차에서 인증서 철회 상태 검사가 1차와 2차로 두 번 반복하여 수행되는 이유는 CRL의 전파에 관련된 지연 시간 문제 때문이다. CRL의 전파에 관련된 지연 시간 문제란, CRL은 주기적으로 전파되므로 서명에 관련된 모든 인증서들과 그들의 철회 및 효력 정지에 대한 CRL 정보를 수집하여 검사하여 인증서들의 유효성이 일단 검증되었다고 해도 아직 철회나 효력 정지된 인증서 정보가 CRL에 반영되어 있지 못한 경우가 생길 수 있다는 문제를 의미한다. 즉, 1차 철회 상태 검사 때 인증서가 철회되거나 및 효력 정지 상태에 있지 않음을 확인하였다 하더라도, 사실은 그 인증서가 철회되거나 효력 정지 상태가 된 직후이어서 아직 CRL에 반영되어 있지 않을 수도 있다. 따라서 CRL을 사용하는 인증서 철회 상태 검사는 인증서의 철회 상태를 정확히 확인하기 위하여 CRL 전파 주기를 고려한 시간 간격을 두고 두 번 수행되어야 한다. ES-T 구조의 생성에서 2차 인증서 철회 상태 검사 완료까지의 시간 간격은 CRL 전파 주기를 고려하면서 동시에 짧을수록 바람직하다. KES-T 시스템의 인증서 철회 상태 검사는 CRL을 사용하고 있지만 OCSP(online certificate status protocol)[9]를 지원하는 온라인 인증서 상태 서버(online certificate status server)를 사용하도록 확장될 수 있다.

본 논문의 KES-T 시스템이 제공하는 ES-C 구조는 보다 긴 수명의 전자 서명을 지원하는 구조들로 쉽게 확장될 수 있다. 예를 들어, 완전한 인증서와 철회 참조 구조에 대응하는 실제 인증서들과 CRL들을 ES-C 구조에 결합하면 RFC 3126[6]이 규정하는 ES-X-Long 구조가 되며, ES-C 구조에 ES-C 구조에 대한 타임스탬프를 다시 결합하면 ES-X Type1 구조가 된다[6]. ES-X-Long 구조는 서명 검증 시에 인증서들과 CRL들을 검색할 필요가 없다는 장점이 있지만 실제 인증서들과 실제 CRL이 아닌 그들에 대한 참조들을 사용하는 ES-C 구조보다 크기가 크다는 단점을 갖는다.

3.2.2 KES-T 시스템의 전자 서명의 검증 절차

KES-T 클라이언트가 전자 서명의 검증을 요청하면 KES-T 서버는 다음 절차에 따라 전자 서명의 검증을 수행하여 그 결과를 KES-T 클라이언트에게 반환한다. KES-T 클라이언트를 통해 서명 대상 파일(signer's file)과 ES-C 구조가 KES-T 서버로 전달하면 KES-T 서버는 전자 서명의 검증 절차를 시작한다.

먼저, KES-T 서버는 ES-C 구조의 기본 포맷을 검증하고, 서명자의 서명값을 검증한다. 서명자의 서명값 검증은 서명 결과인 signed-data 구조의 서명 대상 데이터와 SignerInfo 구조의 서명 속성들(signed attributes)의 결합의 축약을 구하고 그 축약과 signature 필드의 값을 서명자의 개인키와 쌍이 되는 공개키 인증서의 공개키로 해독한 결과값이 같은가를 판정하는 것이다. 두 값이 같으면 서명자의 서명은 검증된다.



[그림 6] KES-T 시스템의 전자 서명 검증 절차

다음 그림 6은 KES-T 시스템의 전자 서명 검증 절차를 나타내는 순차 다이어그램이다.

전자 서명 검증 절차는 서명자의 서명값을 검증한 다음에 ES-C 구조에 포함된 서명 정책 즉 SignaturePolicyId 속성을 읽어 서명 정책 id를 확인한다. 서명 정책 id는 객체 식별자(object identifier)의 형태로 표현된다. 서명에 관련된 비즈니스 모델에 따라 다양한 서명 정책이 가능하다. 제 3.2.3 절에 KES-T 데모 시스템이 사용하는 정책에 대한 기술이 있다.

그 다음으로 ES-C 구조에 포함된 타임스탬프 토큰을 TSA에게 보내 TST 검증 절차를 수행한다. 만일 타임스탬프 토큰 검증 요청에 대하여 TSA가 *invaild*를 반환하면 서명 검증 절차는 KES-T 클라이언트에게 즉시 *invalid*를 반환한다. 그림 5와 마찬가지로 그림 6의 순차 다이어그램에도 다이어그램의 지나친 복잡성을 줄이기 위하여 서명 생성의 매 단계마다 발생할 수 있는 *invalid* 메시지의 전달이 생략되어 있다.

TST 검증이 성공적으로 완료되면 인증서 참조들과 철회 참조들로부터 인증서들과 CRL들, 그리고 CARL(certificate authority revocation list)들을 구하는 절차들이 수행된다. 인증서들, CRL들, 그리고 CARL들이 구해지면 인증서 경로 생성 및 검증 절차가 수행된다. 인증서 경로 생성 및 검증 절차는 서명자의 인증서로부터 서명자의 인증서를 발행한 CA의 루트 인증서까지의 인증서들의 유효 기간이 사용자 서명 시점을 포함하며, 서명자의 인증서로부터 CA의 루트 인증서까지의 인증서 경로가 끊어지지 않고 형성되며 CA의 루트 인증서가 철회되거나 효력정지 되지 않은 상태임을 확인하기 위한 절차이다.

ES-T 구조의 생성 절차에서 사용자의 ES 구조 생성 시각과 타임스탬프 생성 시각간의 간격이 충분히 작다고 할 때, 전자 서명에 결합된 타임스탬프가 유효하다는 것은 그 타임스탬프가 지정하는 시각과 가까운 사용자의 서명 시각에 사용자의 서명이 존재하였다는 증거가 된다. 따라서 서명 검증 시점에 서명에 사용된 사용자의 인증서의 유효 기간이 만료되었다 하더라도 타임스탬프가 지정하는 시각에 가까운 사용자의 서명 시각에 사용자의 인증서가 유효했었다는 것을 밝힐 수 있다면 사용자의 서명은 유효하다고 판정할 수 있다는 것이 수명이 긴 인증서의 근본 원리이다. 그림 5의 KES-T 시스템의 전자 서명 생성 절차에서 ES로부터 ES-T를 생성할 때, ES 포맷 검증 절차가 ES의 문법 구조를 검증하는 비교적 간단한 절차이며, 타임스탬프 요청으로부터 타임스탬프 발행까지의 간격에 대한 시간 제약(time limit)이 존재하므로 사용자의 서명 시각과 타임스탬프가 지정하는 시각의 간격은 충분히 작다고 할 수 있다. 일반적으로 개인의 인증서는 1년 정도의 짧은 유효 기간을 갖지만 TSA와 같이 신뢰되는 기관의 인증서의 인증서 및 CA의 루트 인증서는 유효 기간이 훨씬 길다.

3.2.3 전자 서명 정책

ES-T 구조 이상의 전자 서명에 반드시 포함되어야만 하는 속성의 하나인 전자 서명 정책은 RFC 3369[1]가 규정하는 signed-data 구조의 signerInfos 구조의 signedAttrs 구조의 signingCertificate 속성으로 표현된다. 전자 서명 정책은 비즈니스 요구를 만족시키기 위하여 서명의 생성

과 검증에 요구하는 기술적인 조건들을 의미하며 비즈니스 요구에 따라 다양한 전자 서명 정책들이 존재할 수 있다[6].

전자 서명 정책은 서명 생성 과정 및 결과가 만족해야 하는 조건 및 서명 검증에 성공하기 위해 만족해야 하는 조건을 규정한다. 응용 분야에 따라 전자 서명 정책 발행자(signature policy issuer)는 다양한 전자 서명 정책들을 세우고, 정책에 대한 유일한 객체 식별자(object identifier)와 유일한 서명 정책 규격(signature policy specification)의 쌍을 RA(repository authority)에 등록할 수 있다. 전자 서명 정책 속성을 가진 전자 서명의 경우 사용하는 전자 서명 정책에 따르는 조건들을 만족하도록 서명 생성 및 검증이 이루어져야 한다. KES-T 시스템에서 KES-T 서버 또는 그 서버를 운영하는 기관이 전자 서명 정책 발행자(signature policy issuer)이다. 전자 서명 정책에는 해당 서명 정책의 사용 허가 기간, 정책 발행자 이름 등의 정보가 포함되며 예를 들어, 다음과 같은 조건들에 대한 규정이 포함될 수 있다[6].

- 서명 대상 데이터를 서명의 내부에 포함할 것인가 외부에 둘 것인가에 대한 조건
- 서명에 사용된 사용자의 인증서 및 서명에 포함된 타임스탬프 토큰을 발행하는 데 사용된 TSA의 인증서의 검증 조건. 예를 들어, 사용자의 서명 시점으로부터 서명에 포함된 타임스탬프 토큰이 발행된 시점까지의 최대 허용 시간 간격 등의 조건.

KES-T 데모 시스템은 임의의 2개의 전자 서명 정책 객체 식별자들을 이용하여 서명 시점에서 타임스탬프 토큰 발행 시점간의 간격에 관련된 두 가지 서명 정책만을 사용하였다. 첫 번째 정책은 사용자의 전자 서명 생성 이후 3초 이내에 타임스탬프 토큰이 발행되어야만 한다는 조건이고, 두 번째 정책은 사용자의 전자 서명 생성 이후 15초 이내에 타임스탬프 토큰이 발행되어야만 한다는 정책이다. KES-T 데모 시스템은 단 두 가지의 간단한 서명 정책들만을 구현하고 있지만 KES-T 시스템은 비즈니스 요구에 따르는 다양한 전자 서명 정책을 수용하도록 쉽게 확장할 수 있다.

3.2.4 시점확인서비스 시스템과의 연동 검증 및 남은 문제점

시점 확인 서비스는 HTTP(hypertext transfer protocol), FTP(file transfer protocol), 전자 우편 등의 다양한 방법으로 서비스될 수 있다[3]. 본 논문의 KES-T 데모 시스템에서 KES-T 서버는 HTTP를 사용하여 한국표준과학연구원의 시점확인서비스 시스템과 연동한다.

KES-T 시스템이 시점확인서비스 시스템과 연동하여 올바르게 동작하는가에 대한 검증은 주로 시스템이 중간 결과 또는 최종 결과로 생성하는 자료 구조들에 대하여 이루어졌다. KES-T 시스템은 3.2.1 절에 기술한 바와 같이 ES-C 구조를 생성하며 그 중간 단계로 사용자의 전자 서명에 시점확인 시스템이 제공하는 타임스탬프 토큰이 결합된 ES-T 구조를 생성한다. 그림 2의 ASN.1 파서 모듈을 사용하여 KES-T 데모 시스템이 생성하는 ES-T 구조 등의 전자 서명 구조들을 검증하였으며, 검증 결과로 KES-T 데모 시스템 및 KES-T 라이브러리가 RFC 3369[1], RFC 3126[6] 등의 국제 표준 규격에 부합하는 올바른 전자 서명 구조를 생성함을 확인할 수 있었다. ASN.1 파서를 사용하면 ASN.1 표기법으로 규정된 자료 구조의 구조 및 내용을 확인할 수 있다.

KES-T 시스템은 아직 몇 가지 문제점을 남기고 있다. 첫째, 다양한 사용 분야들을 지원하려면, 그 분야들을 위한 전자 서명 정책들에 관련한 지원 기능들이 추가되어야 한다. 데모 시스템은 타임스탬프 토큰의 발행에 대한 3초와 15초의 시간 제약에 대한 정책만을 구현하고 있어 전자 서명 정책에 관련한 기능들을 충분히 검증하였다고 하기 힘들다. 둘째, KES-T 시스템은 전자 서명에 사용된 인증서의 철회 여부를 검색하기 위하여 CRL(certificate revocation list)만을 사용하도록 설계되어 있지만, 전자 서명에 사용된 인증서의 철회 여부를 효과적으로 검색하기 위하여 OCSP(online certificate status server)도 지원하도록 확장되는 것이 바람직하다. 셋째, 디지털 문서 보관소와 같이 매우 장기간 유효한 전자 서명을 필요로 하는 분야들을 지원하려면 KES-T 시스템은 ES-A 구조와 같은 확장된 구조를 지원하도록 기능 확장을 필요로 한다.

4. 결론 및 추후 연구 계획

본 논문은 수명이 긴 전자 서명의 형태의 하나인 ES-C 구조의 전자 서명을 생성하고 검증하는 서비스를 제공하는 KES-T 시스템을 제안하고, 제안한 KES-T 시스템을 테스트하기 위한 KES-T 라이브러리와 KES-T 데모 시스템을 개발하였다. KES-T 시스템은 한국표준과학연구원에서 운영하는 웹 기반의 시점확인 서비스 시스템인 Kriss TSA에 연동하여 동작한다. KES-T 시스템이 생성하는 전자 서명은 Kriss TSA와 공모하지 않고는 전자 서명의 시점을 조작할 수 없다. 또한 사용자가 서명에 사용한 사용자의 인증서의 유효 기간이 지나더라도 그 서명의 유효성(validity)을 오래 지속할 수 있다. 왜냐하면

KES-T 시스템이 발행하여 서명 결과에 포함된 타임스탬프 토큰은 사용자의 유효한 서명이 타임스탬프 발행 시점에 이미 존재하였다는 사실에 대한 증거가 될 수 있기 때문이다.

따라서 KES-T 시스템이 제공하는 서비스는 오랫동안 전자 서명의 유효성을 검증해야 하는 고액 전자 결재, 온라인 경매, 문서 보관소 시스템 등의 다양한 전자 서명 관련 시스템들을 개발하는 데 핵심적인 요소로 사용될 수 있다.

장기간 유효한 전자 서명을 필요로 하는 응용 분야는 잠재적으로 다양하다. 본 논문의 연구 결과 시스템은 기존의 전자 서명 방식이 지원하지 못하는 고액 전자 상거래, 전자 경매 시스템, 전자 복권 시스템, 장기 문서 보존, 문서 공증 등의 다양한 응용 분야에서 효과적으로 활용될 수 있다. 장기간 유효한 전자 서명을 사용하는 전자 서명 응용 시스템들이 필요로 하는 전자 서명 생성 및 검증 기능을 자체적으로 내장한 형태로 개발하여 사용할 수도 있겠지만 공신력 있는 기관이 제공하는 수명을 갖는 전자 서명의 생성과 검증에 관련된 서비스를 사용하는 것도 의미가 있는 방법이 될 수 있을 것이다. 전자 서명에 국가 기관에서 운영하는 공식 시점 확인 서비스와 결합하여 수명이 긴 ES-C 구조의 전자 서명을 생성하고 검증하는 온라인 서비스를 제공하는 시스템의 개발은 국내에서는 최초로 판단된다.

추후 연구로는 KES-T 시스템의 기능 확장 및 웹 서비스화에 대한 연구가 필요하다. KES-T 시스템은 현재 수명이 긴 전자 서명의 형태인 ES-C 구조를 생성하지만 쉽게 ES-X-Long, ES-X Type1, ES-A(archival electronic signature) 등의 보다 수명이 긴 전자 서명 구조를 생성하는 시스템으로 확장될 수 있다. 왜냐하면 ES-X-Long, ES-X Type1, ES-A 등의 구조는 ES-C 구조에 점진적으로 더 필요한 요소들을 추가하는 구조들이기 때문이다.

KES-T 시스템은 클라이언트 서버 방식으로 구현되었지만 KES-T 라이브러리는 높은 이식성을 가지므로 KES-T 클라이언트를 웹 버전으로 변환하는 것도 어려운 일이 아니다. 예를 들어 추후 KES-T 시스템을 웹 서비스(web service)의 한 형태로 구현할 수도 있을 것이다.

참고문헌

[1] Housley, R. (2002), "Cryptographic Message Syntax", RFC 3369.

[2] Schneier, Bruce. (1996), Applied Cryptography 2nd Edition, Willy.

[3] Adams, C., Cain, P., Pinkas, D., Zuccherato, R. (2001), "Internet X.509 Public Key Infrastructure Time-Stamp Protocol(TSP)", RFC 3161.

[4] 장혜진, 김관웅, 이상태, 이호성. (2004), "웹 기반의 타임스탬프 토큰 발행 서버의 구축", 한국감성과학회 2004 춘계학술대회 프로시딩.

[5] 장혜진 (2004), "국제 표준을 만족하는 웹 전자 공증 시스템의 개발", 한국 산학 기술학회 논문지 Vol. 5, No. 1, pp. 21-25.

[6] Pinkas, D., Ross, J., Pope, N. (2001), "Electronic Signature Formats for long term electronic signatures", RFC 3126.

[7] Verisign Inc. (2008), "Authenticated Content Signing", http://www.verisign.com/stellent/groups/public/documents/data_sheet/003201.pdf.

[8] Housley, R., Ford, W., Polk, W., Solo. D. (1999), "Internet X.509 Public Key Infrastructure Certificate and CRL Profile". RFC 2459.

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

[10] Mills, D. L. (1992), "Network Time Protocol (Version 3) - Specification, Implementation and Analysis". RFC 1305.

[11] Mills, D. L. (1996), "Simple Network Time Protocol (SNTP) Version 4 - for IPv4, IPv6 and OSI". RFC 2030.

[12] Rivest, R. L., Shamir, A., and Adeleman, L. M. (1983), "Cryptographic Communications System and Method", U.S Patent #4,405,829.

[13] Eastlake, D., 3rd., Jones, P. (2001), "US Secure Hash Algorithm 1 (SHA1)", RFC 3174.

[14] Rivest, R. L. (April 1992), "The MD5 Message Digest Algorithm", RFC 1321.

[15] John Viega, Matt Messier, and Pravir Chandra (June 2002) Network Security with OpenSSL, O'Reilly.

[16] ISO 8824:1987 (1987), "Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation (ASN.1)", International Organization for Standardization.

[17] ISO 8825:1987 (1987), "Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation (ASN.1)", International Organization for Standardization.

장 혜 진(Hai Jin Chang) **[정회원]**

- 1994년 2월: 서울대학교 대학원 계산통계학과 박사 졸업(전산학 전공)
- 1994년 3월 ~ 현재: 상명대학교 공과대학 컴퓨터소프트웨어공학과

<관심분야>
통신 보안, 로봇 분산 제어 시스템, DRM.