서명 검증을 위한 Historical CRL

노종혁, 진승헌

ETRI, 정보보호연구본부

Historical CRL for Verifying the Signature

Jong-Hyuk Roh and Seunghun Jin

Information Security Research Division. ETRI

요 약

공개키 기반 구조에서 과거 시점에 서명한 문서를 검증하기 위해서는 서명한 시점의 인증서의 상태정보를 확인하여야 한다. 하지만, 기존의 상태 정보 제공 메커니즘을 이용하여 과거 시점에 대한 상태 정보를 획득할 경우에는 여러 가지 문제가 발생할 수 있다. 본 논문에서는 이러한 문제점에 대하여 자세히 설명하고 이를 해결하기 위한 방법을 제시한다.

I. Introduction

공개키 기반구조(PKI)에서는 전자서명 또는 신원확인 정보를 생성하는 서명자와 이를 검증하는 검증자가 존재한다. 검증자는 서명자의 정보를 검증하는 과정에서 서명자가 이용한 개인키에 부합되는 공개키가 필요하며, 이것은 인증기관이 발행한 인증서를 이용한다.

인증기관(Certification Authority)에서 인증서를 발행할 때에는 인증서에 만료일(expiration date)를 표기함으로써 인증서의 유효기간을 제한하고 있다. 그러나, 인증서의 유효기간이 만료되기 전에 개인키의 손실, 인증서 소유자의 정보 변경 등 다양한 이유로 소유자 또는 인증기관에 의해 인증서가 폐지될 수 있다. 그러므로, 검증자는 인증서를 사용하기 전에 반드시 인증서의 상태가 유효한지 확인하는 절차가 필요하며, 인증기관은 검증자에게 인증서의 유효성을 판단할 수 있는 정보를 제공하여야 한다[3].

인증서 상태 정보를 제공하는 수단으로, 가장 대표적이며 많이 사용되는 방법으로 폐지된 인증서의 목록을 주기적으로 생성하여 배포하는 CRL(Certificate Revocation List)이 있다. 그러나, CRL은 근본적으로 CRL 크기의 증가에 따른 통신상의 트래픽 증가 문제와 주기적인 특성으로 인한 실시간 인증서 상태 정보를 제공하지 못하는 문제를 가지고 있다. 이를 해결하기 위하여, Delta-CRL, CRL DP, CRS, CRT, OCSP 등 다양한 방법들이 제안되었다[3,4].

지금까지 제시되어 온 방법들은 인증서의 최근 상태 정보를 제공하기 위한 방법들이다. 하지만, 인증서의 검증은 현재의 인증서 상태 정보 획득만으로 부족한 경우가 존재한다. 즉, 과거에 서명된 문서에 대한 서명 검증이 필요한 경우에는 문서가 서명된 시점의 인증서 상태 정보가 필요하며, 인증기관은 검증자가 요구하는 시점의 정보를 제공할 수 있어야 한다. 하지만 현재 구축되어 있는 인증기관들은 이를 제공하지 않고 있다.

그리고, 인증서의 소유자는 특정 기간 동안 인증서를 사용하지 않기 위해 인증서의 효력을 정지(CertificateHold)할 수 있다. 기존 방법인 CRL 및 Delta-CRL, CRL-dp에서, 인증서가 효력 정지될 경우 폐지목록에 효력 정지된 인증서의 일련번호와 정지된 시간을 표현한다. 하지만, 인증서가 효력을 회복하는 경우에는 폐지목록에서 효력 정지되었던 인증서 목록을 제거할 뿐 효력을 회복한 시간은 폐지목록에 표현하지 않는다. 이러한 방식은 부인 봉쇄 서비스를 제공하는 PKI의 중대한 문제점을 초래하게 된다.

본 논문에서는 문서의 서명 시점과 검증하는 시점에 따라 발생할 수 있는 모든 경우를 파악하고,

기존의 방법으로 해결하지 못하는 부분에 대하여 지적하며, 이를 해결하기 위한 방법을 제시한다.

II. 전자 서명과 검증

서명자 Alice는 자신의 개인키로 문서 S를 서명하였다. 몇 개월 후, 검증자 Bob은 Alice가 서명한 문서 S를 검증해야 할 일이 발생하였다. Bob이 S를 검증하기 위해서는 Alice의 인증서를 검증하여야 한다. 우선, Bob은 Alice가 문서를 서명한 시점을 알아야 하고, 또한, 문서가 서명된 시점에 Alice의 인증서가 유효한지를 판단하여야 한다.

본 절에서는 전자 서명에 대한 설명과 서명을 검증할 시에 발생할 수 있는 여러 경우를 살펴보고, 기존의 방법으로 검증할 시에 발생하는 문제점을 지적한다.

1. 전자 서명

전자 서명이란 전자화된 문서의 메시지 내용이 수정 및 변조되지 않았음을 보장하는 동시에 메시지의 서명을 다른 사람이 아닌 서명자가 했음을 제 3자가 확인할 수 있게끔 하는 방식을 말한다. 전자 서명의 가장 일반적인 환경은 그림 1과 같다. 전자 서명 문서를 만드는 서명자는 인증기관으로부터 인증서와 함께 발급 받은 개인키를 이용하여 전자 서명을 생성한다. 검증자는 전자 서명된 문서가 변경되지 않았고, 인증서의 소유자가 서명했음을 검증하기 위하여 서명자 인증서의 공개키를 사용하여 전자 서명을 검증한다. 서명을 검증하기 이전에 인증서 자체에 대한 검증이 필요하므로, 인증서의 상태정보를 제공하는 곳으로부터 해당 인증서의 상태정보를 얻는다. 검증자는 인증서 상태정보를 이용하여 서명자의 인증서를 검증한 후 전자 서명 문서를 검증한다. 일반적으로 인증서 상태정보 제공자는 인증서를 발급한 인증기관이거나 인증기관과 신뢰관계를 맺고 있다. 인증서 상태정보를 제공하는 방법은 CRL, OCSP 등이 이용된다.

문서를 서명할 시에는 서명한 시간을 서명 문서에 포함시켜, 서명 검증자가 문서의 서명된 시점을 알 수 있도록 하여야 한다. 주로 문서를 서명하는 포맷은 RFC 2630 [6] CMS(Cryptographic Message Syntax)를 따르고 있으며, Signing Time attribute를 이용하여 서명 시간을 서명에 포함시킬 수 있다. Signing Time attribute를 포함하지 않는 경우에는 Time Stamp를 서명과 함께 보존함으로써, 서명에 대한 시점을 명확하게 하여야 한다[7].

그림 1 전자 서명과 인증서 상태정보 이용 과정

CMS의 *SignedData.crls* 필드를 이용하여, 서명자는 서명시 발행되어 있는 CRL을 서명에 포함시킬 수 있다. 이는 검증자가 서명 검증시 서명에 포함되어 있는 CRL을 참고하여 인증서의 상태정보를 얻도록 하기 위함이다. 하지만, CMS에 포함되어 있는 CRL을 이용하면 문제가 발생할 수 있다. 왜냐하면, 서명자가 서명하기 전에 자신의 인증서를 폐지하고 서명한 경우, 서명시 발행되어 있는 CRL에는 서명자의 인증서 폐지정보가 담겨 있지 않기 때문이다. 그러므로, 본 논문에서는 서명에 포함되어 있는 CRL 정보는 사용하지 않는다.

2. 서명 검증

본 절에서는 서명한 문서의 검증시에는 검증하는 시점에 따라 발생할 수 있는 문제점에 대하여 논한다. 검증을 위해 인증서의 상태 정보를 얻는 방법은 다양하지만 아래에서는 CRL과 OCSP에 대해서만 설명을 한다. 왜냐하면, CRL-DP, Delta-CRL 등은 CRL 획득시 네트워크 부하를 줄일 수 있지만, CRL의 주기적인 특성은 그대로 유지하기 때문이다.

1) CRL을 이용한 인증서 상태정보 획득

그림 2에서 인증서 C0의 유효기간은 t0에서 t5이다. t2에서 개인키 손상으로 인해 C0는 폐지되어 CRL3에는 폐지된 인증서 C0의 정보가 포함되어 있다. CRL4는 C0의 기간이 만료되었으므로 C0의 폐지정보를 포함하지 않는다. 서명 S0을 생성한 시점은 t1이고 서명을 검증하는 시점은 t3, t4, t6 이다. 서명을 한후에 인증서가 폐지 되었으므로 서명은 검증의 시점과 상관없이 유효하다

경우1) 검증 시점 t3: C0의 상태 정보를 얻기 위해 현재 발행되어 있는 CRL2를 획득한다. CRL2

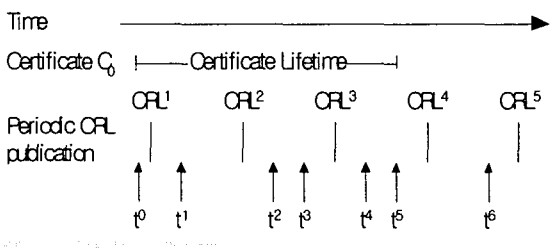

그림 2. 유효한 서명 문서의 검증

에는 C0의 폐지 정보가 나와 있지 않다. S0는 유효하다고 판단한다.

경우2) 검증 시점 t4: C0의 상태 정보를 얻기 위해 현재 발행되어 있는 CRL3을 획득한다. CRL3에는 C0가 폐지된 날짜인 t2가 명시되어 있다. 서명한 시점 t1은 t2보다 이전이므로 S0는 유효하다고 판단한다.

경우3) 검증 시점 t6: C0의 상태 정보를 얻기 위해 현재 발행되어있는 CRL4을 획득한다. C0의 기간이 만료되었으므로 CRL4에는 C0의 정보를 포함하지 않는다. 그러므로, 검증자는 CRL2 또는 CRL3을 획득하여 판단하여야 한다.

경우 3과 같이 인증서가 만료되어 CRL이 해당 인증서의 정보를 포함하지 않는 경우는, 서명 시점보다 이후에 발행되고 인증서의 만료 이전에 발행된 CRL을 획득하여 인증서 상태정보를 얻어야만 한다.

그림 3에서 C1의 유효기간은 t0에서 t5이다. t1에 인증서를 폐지하고 t2에 서명 S1을 생성하였으므로 S1은 유효하지 않다. CRL2와 CRL3에는 C1이 폐지되었다는 정보를 포함하고 있고, CRL4는 C1의 기간이 만료되었으므로 C1의 폐지정보를 포함하지 않는다.

경우4) 검증 시점 t3: C1의 상태 정보를 얻기 위해 현재 발행되어 있는 CRL1을 획득한다. CRL1에는 C1의 폐지 정보를 담고 있지 않다. 그러므로, S1은 유효하다고 판단한다.

경우5) 검증 시점 t4: C1의 상태 정보를 얻기 위해 현재 발행되어 있는 CRL2를 획득한다. CRL2에는 C1이 폐지된 날짜인 t1이 명시되어 있다. 서명한 시점 t2는 t1보다 이후이므로 S1은 유효하지 않다고 판단한다.

경우6) 검증 시점 t6: C1의 상태 정보를 얻기 위해 현재 발행되어 있는 CRL4를 획득한다. C1의 기간이 만료되었으므로 CRL4는 C1의 폐지정보를 포함하지 않는다. 경우 3과 같이, 검증자는 CRL2 또는 CRL3을 획득하여 검증한다. S1은 유효하지 않다고 판단한다.

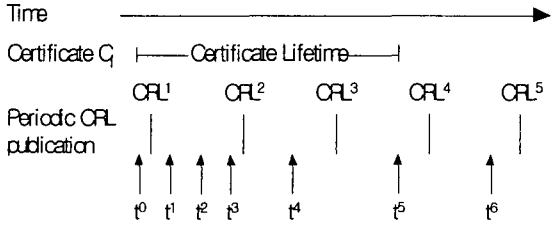

그림 3. 유효하지 않은 서명 문서의 검증

CRL은 근본적으로 인증서 상태 정보를 실시간으로 제공하지 못하므로, 경우 4)에서는 잘못된 판단을 하고 있다. 경우 4)에서 제대로 된 판단을 하기 위해서는 OCSP와 같은 실시간 정보를 제공하는 프로토콜을 사용하여야 한다.

일반적으로, 기존의 인증기관에서는 CRL을 발행할 때, 발행되어 있는 CRL 위에 엎어쓴다. 그러므로, 앞의 예제에서처럼 과거의 CRL을 획득할 방안이 없기에, 과거에 서명된 문서를 검증할 수 없다. 기존의 인증기관의 CRL 발급 메커니즘과 정책은 변경되어야 한다.

2) OCSP를 이용한 인증서상태정보 획득

OCSP는 CRL의 단점인 CRL 크기에 따른 네트워크 부하를 해결하고 실시간으로 인증서의 상태정보를 제공하는 프로토콜이다. 현재 OCSP는 IETF PKIX 워킹 그룹에서 버전 1이 RFC 2560으로 등록되어 있고 버전 2는 드래프트 상태이다[5].

그러나, OCSP는 인증서의 상태정보 제공을 위한 프로토콜일뿐OCSP Responder가 인증서의 상태정보를 어떠한 방법으로 수집, 관리 하는 가에 대해서는 표준에 언급하고 있지 않다. 그러므로, Responder가 인증서의 상태정보 수집을 위해 CRL을 사용한다면, 인증서의 실시간 상태정보를 제공할 수 없게 된다. 실시간 정보를 제공하기 위해서는, OCSP Responder가 CA(Certificate Authority)와 데이터 베이스를 연동하거나, CA로부터 실시간 정보를 얻기 위한 다른 메커니즘이 필요하다[1]. 본 논문에서는OCSP Responder가 실시간 정보를 수집하고 있다고 가정한다.

OCSP를 이용하여 서명을 검증하는 경우 발생하는 문제점 및 해결 방안에 대하여 설명한다. CRL과의 비교를 위하여 그림 2과 그림 3의 경우에 대하여 설명한다.

- 유효한 서명문서의 검증 (그림 2)

경우1) 검증 시점 t3: C0의 상태 정보를 얻기 위해 OCSP Responder에게 요청한다. OCSP 응답에는 현재 C0가 폐지되었다고 표현되지만 폐지된 시간 t1이 응답에 명시되어 있으므로, S0는 유효하다고 판단한다.

경우2) 검증 시점 t4: 경우 1과 같다.

경우3) 검증 시점 t6: OCSP 응답은 Responder를 어떻게 구현하였는 가에 따라, "revoked" 또는

"unknown"이 될 수 있다. 왜냐하면, RFC 2630에는 만료된 인증서에 대한 데이터를 표현하는 방법을 명시하지 않고 있기 때문이다. 그러므로, 판단은 Responder에 따라 다를 수 있다.

- 유효하지 않은 서명문서의 검증 (그림 3)

경우4) 검증 시점 t3: C1의 상태 정보를 얻기 위해 OCSP Responder에게 요청한다. OCSP 응답에는 현재 C1이 폐지되었다고 표현되어 있으며 폐지된 시간 t1은 서명시간인 t2보다 빠르다. S1은 유효하지 않다.

경우5) 검증 시점 t4: 경우 4와 같다.

경우6) 검증 시점 t6: 경우 3과 같다.

OCSP를 이용함으로써, CRL에서 해결하지 못한 문제인 경우 4를 해결할 수 있었다. 그러나, 경우 3과 경우 6에서 올바른 정보를 제공하기 위해서는, OCSP Responder가 만료된 인증서에 대한 검증 서비스를 제공하여야 한다. 즉, CA에서 발행한 모든 인증서에 대한 정보를 유지하고 있어야 한다. 서명 검증시 CRL과 OCSP 메커니즘의 특징을 비교하면 표 1과 같다.

표 1. CRL vs. OCSP

CRL	1) CRL의 주기적은 특성으로 인해 인증서의 최신 정보를 제공하지 못한다. 2) 검증시점에 만료된 인증서로 검증을 하기 위해서는, 서명 시점보다 이후에 발행되고 인증서만료 이전에 발행된 CRL을 획득하여 참고한다.
OCSP	1) 현재 시점의 인증서 상태정보를 제공하기 위하여, OCSP Responder는 CA의 인증서 상태 정보를 실시간에 획득할 수 있어야 한다. 2) OCSP 프로토콜에는 과거 시간에 대한 요청을 질의할 수는 없게 되어 있으나, 현재시점에 대한 질의로서 과거시점을 추정할 수 있다. 왜냐하면, 인증서가 폐지 되었다면 CRL과 마찬가지로 폐지된 날짜가 응답에 들어있기때문이다. 3) OCSP Responder는 만료된 인증서에 대한 검증 서비스를 제공하기 위하여, CA에서 발행한 모든 인증서에 대한 정보를 유지하고 있어야 한다.

3. CertificateHold

RFC 3280의 CRL에는 인증서 폐지 사유로 certificateHold가 있다. certificateHold란 인증서의 소유자가 특정한 사유로 인증서를 사용을 중지하고, holder가 원하는 시간에 인증서의 상태를 복원할 수 있는 기능이다[8]. 인증서의 소유자가 인증서를 hold하면, CA는 다음 발행하는 CRL에 해당 인증서를 list에 추가한다. 인증서의 holder가 인증서의 상태를 복원하면 다음에 발행하는 CRL에서는 복원된 인증서의 항목을 제거한다. 그러나, 이렇게 단순한 복원 방법은 과거에 서명한 문서를 검증할 시에 문제점을 초래하게 된다.

그림 4에서 인증서 C2의 유효기간은 t0에서 t7이다. 인증서의 소유자는 특별한 이유로 인해 t1에서 C2를 hold하고 t4에서 C2의 상태를 복원하였다. CRL2에는 C2가 hold되었다는 정보를 포함하고 있지만, CRL3에는 C2의 정보를 포함하고 있지 않다. 서명을 검증하는 시점은 t6이고, 서명을 생성한 시점이 t2 또는 t3 또는 t5인 경우에 대하여 설명을 하면 아래와 같다

경우1) 서명 시점 *t2*: C2의 상태 정보를 얻기 위해 현재 발행되어 있는 CRL3을 획득한다. CRL3에는 C2의 정보를 포함하지 않는다. 그러므로, S2는 유효하다고 판단한다. 그러나, 검증자가 certificateHold의 가능성을 생각한다면, CRL2를 획득하여야 한다. CRL2는 C2가 t1에 hold 되었다는 정보를 포함한다. t1은 문서를 서명한 t2 보다 이전이므로 서명 S2는 유효하지 않다고 판단한다.

경우2) 서명 시점 t3: C2의 상태 정보를 얻기 위해 현재 발행되어 있는 CRL3을 획득한다. CRL3에는 C2의 정보를 포함하지 않는다. Hold 상태에서 서명을 했음에도 불구하고, 서명 S3이 유효하다고 판단한다.

경우3) 서명 시점 t5: C2의 상태 정보를 얻기 위해 현재 발행되어 있는 CRL3을 획득한다. CRL3

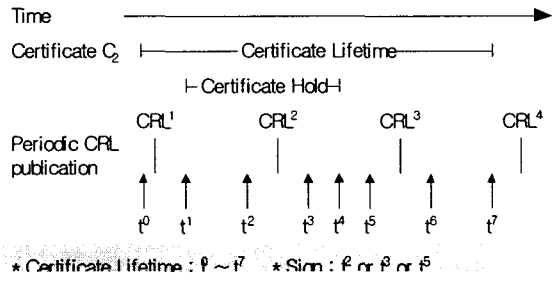

그림 4. CertificateHold와 서명검증

에는 C2의 정보를 포함하지 않는다. 서명 S5가 유효하다고 판단한다.

2.2절에서 검증 시점에 만료된 인증서로 검증을 하기 위해서는, 서명 시점보다 이후에 발행되고 인증서의 만료 이전에 발행된 CRL을 획득하여 참고하였다. 하지만, 인증서의 hold 가능성을 고려한다면, 경우 1)에서와 같이 서명 후 제일 먼저 발행된 CRL(경우 1에서 CRL2)을 획득하여야 한다.

경우 2는 잘못된 판단을 하고 있다. 이유는 인증서의 상태를 회복한 후에 발행되는 CRL은 인증서가 어느 시점에 회복을 하였는 가에 관한 정보를 포함하지 않기 때문이다.

그림 5에서는 certficateHold 기간이 CRL1과 CRL2가 발행되는 중간에 위치하고 있다. t2에 서명한 문서 S2는 유효하지 않고 t4에 서명한 문서 S4는 유효하다.

경우 4) 서명 시점 t2: 인증서 상태 정보를 얻기 위해 현재 발행되어 있는 CRL2를 획득한다. CRL2에는 C3의 정보를 포함하지 않는다. S4가 유효하다고 판단한다.

경우5) 서명 시점 t4: 경우 4와 같다.

certificateHold된 기간이 그림 5와 같이 짧은 기간에 이루어진다면 CRL에는 인증서가 certificateHold 되었다는 정보를 표현할 수 없다. 그러므로, CRL을 이용해서는 인증서의 상태 정보를 얻을 수 없다.

OCSP를 이용하여 인증서의 상태 정보를 얻을 시에는, 그림 4, 그림 5의 모든 경우에 대하여 서명을 유효하다고 판단하게 된다. 왜냐하면, OCSP의 응답은 검증 시점의 인증서의 상태를 알려주기 때문이다. 즉, OCSP는 과거의 인증서 상태 정보는 인증서가 폐지된 시간 외에는 어떠한 정보도 얻을 수 없다.

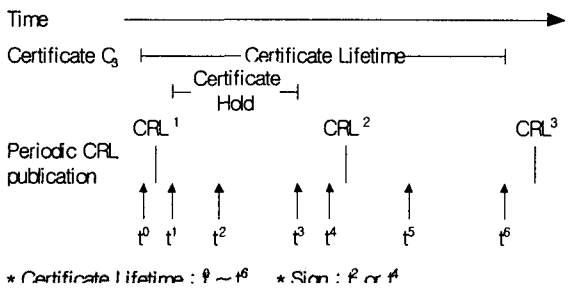

그림 5. 짧은 기간의 CertificateHold와 서명 검증

표 2. CRL vs. OCSP

방법	특징
CRL	1) 검증자는 서명 직후에 발행된 CRL을 획득하여 정보를 얻어야 한다. 2) CRL은 효력회복 시간에 대한 정보를 표현하지 못한다.
OCSP	1) OCSP는 현재 시간에 대한 인증서 상태 정보를 제공하므로 인증서가 폐지된 시간 외에는 어떠한 정보도 제공하지 못한다.

certificateHold의 경우, 서명 검증시 CRL과 OCSP 메커니즘의 특징을 비교하면 표 2와 같다.

III. Solution

본 장에서는 2장에서 제기된 문제점을 해결하기 위한 방법을 제시한다. 기존의 CRL에 extension field를 추가하고,OCSP의 프로토콜을 수정하여 과거 시점의 인증서 상태 정보를 정확하게 얻는 방법을 제시한다. 그리고, 인증기관은 과거 인증서 상태 정보를 제공하기 위해 발행한 CRL을 모두 관리하여야 한다. 이 부담을 줄이기 위해 본 논문은 Historical CRL을 제안한다.

1. Extension Field

1) CRL

2장에서 CRL을 이용하여 과거 시점의 인증서 상태 정보를 얻을 시에 발생할 수 있는 문제들을 살펴보았다. 첫번째 문제는 CRL 고유의 문제인 주기적인 특성으로 인하여 실시간 정보를 얻지 못한다는 점이다. 이 문제는 CRL에서는 극복할 수 없는 문제이다. 두번째 문제는 인증서의 certificateHold로 인한 문제이다. 인증서가 certificateHold 상태에서 복구하는 경우 복구 시간 정보를 CRL에 표현하지 못한다는 것과, CRL이 발행되는 사이에 인증서가 hold 되었다가 다시 복구된 경우에는, 이 정보를 CRL에 표현하지 못한다는 점이다. 본 논문에서는 인증서의 certificateHold 상태로 발생하는 문제점을 기존의 CRL에 extension field를 추가하여 해결한다.

인증서가 hold에서 복구된 후에 발행되는 CRL의 *crlExtensions* 필드에 그림 6과 같은 *invalidCertificates* 확장필드를 삽입 한다.

invalidCertificates 확장 필드는 다수의 정보를 표현하기 위해 SEQUENCE OF SEQUENCE 형태로 구성되어 있다. *userCertificate* 필드는 효력정지의 경험이 있는 인증서의 일련번호를 표현하

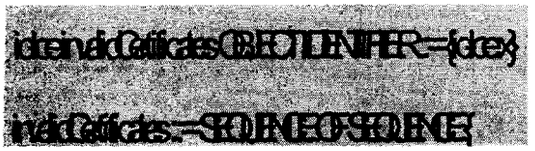

그림 6. invalidCertificates 확장 필드

그림 7. requestTime 확장 필드

고, *fromInvalid* 필드는 hold가 시작된 시간을 표현하며, *toInvalid* 필드는 hold가 끝나는 시간을 표현한다. 그림 4를 예로 들면, CRL2에는 C2가 hold 되었다는 정보가 *revokedCertificates* 필드에 표현되어 있으며 hold가 시작된 시간 t1은 *revokedCertificates.revocationDate* 필드에 명시된다. CRL3에는 C2의 상태가 복구 되었으므로, *revokedCertificates* 필드에서 C2의 정보를 삭제하고, *crlExtensions*에 *invalidCertificates* 필드를 추가한다. *userCertificate*는 C2의 일련번호, *fromInvalid*는 t1, *toInvalid*는 hold가 복구된 시간 t4를 명시한다. CRL4를 발행할때에는 C2의 *invalidCertificates* 필드를 삭제한다.

그림 5와 같은 경우에는 CRL2에만 *invalidCertificates* 확장 필드를 담는다. *userCertificate*는 C3의 일련번호, *fromInvalid*는 hold가 시작된 시간 t1, *toInvalid*는 hold가 복구된 시간 t3을 명시한다. 서명을 검증할 시에는 서명 시점 후 제일 먼저 발행한 CRL로부터 인증서의 상태정보를 얻어야 한다.

2) OCSP

과거에 서명된 문서를 OCSP를 이용하여 검증하기 위해서는, OCSP의 요청에 과거의 시간을 표현할 수 있는 필드가 필요하다. OCSP 요청의 *singleRequestExtensions* 필드에 그림 7과 같은 *requestTime* 필드를 추가한다. 또한, OCSP Responder는 CA가 발행한 모든 인증서에 대한 효력 정지 정보를 관리하여야 한다.

2. Historical CRL

2장에서 언급한것처럼, 과거의 인증서의 상태 정보를 얻기 위해서는 과거에 발행안 CRL을 획득할 수 있어야 한다. 하지만, 기존의 인증기관에서는 CRL을 발행하면 기존에 발행되어 있는 CRL을 삭제하고 현재 발행한 CRL만을 보관한다. 그러므로, 현재의 인증기관에서 제공하는 CRL 만으로는 과거에 서명된 문서를 검증할 수 있다.

인증기관이 발행한 모든 CRL을 관리하기에는 어느 정도 부담이 된다. 또한, 과거의 전자 서명 문서를 자주 검증하는 기관 또는 서버가 존재할 경우, 검증하려는 문서가 생성된 시간에 발행된 CRL을 일일이 획득한다는 것은 네트워크에 부담이 될 수 있다. 본 논문에서는 이러한 문제를 극복할 수 있는 Historical CRL을 제안한다.

Historical CRL은 특정 주기마다 발행되어 있는 CRL의 내용을 통합하여 하나의 CRL로 구성되어 있다. 예를 들어 인증기관이 CRL을 발행하기 시작한지 10년이 지났고, 하루에 두번씩CRL을 발행한다면, 인증기관이 보관하여야 할 CRL은 7300개이다. Historical CRL을 6개월마다 발행한 다면, 인증기관이 보관하여야 할 Historical CRL은 20개이다. Historical CRL의 크기는 기존 CRL의 크기보다 상대적으로 크지만, 인증기관이 보관해야 할 CRL의개수는 상대적으로 줄어든다. 또한, 과거의 전자 서명 문서를 자주 검증하는 기관 또는 서버가 존재할 경우, 20개의 Historical CRL을 미리 얻어 관리한다면 과거의 인증서 상태 정보를 얻기에 효율적이라 할 수 있다.

Historical CRL의 구조는 그림 8과 같다. *from*과 to는 Historical CRL에 표현된 정보들의 기간을 표현한다. 즉, from 시간에 발행한 CRL부터 to 시간에 발행한 CRL까지의 정보를 통합하여 저장하고 있다. revokedCertificates는 Historical CRL의 기간 내에 폐지된 인증서의 정보이다. invalidCertificates는 기간 내에 효력 정지 되었던 인증서의 효력 정지/복원 정보를 표현한다.

```
HistoricalCertificateList ::= SEQUENCE {
    tbsCertList             HistoricalTBSCertList,
    signatureAlgorithm      AlgorithmIdentifier,
    signatureValue          BIT STRING }

HistoricalTBSCertList ::= SEQUENCE {
    version         Version OPTIONAL,
    signature       AlgorithmIdentifier,
    issuer          Name,
    from            Time,
    to              Time,
    revokedCertificates     SEQUENCE OF SEQUENCE {
        userCertificate     CertificateSerialNumber,
        revocationDate      Time,
        crlEntryExtensions  Extensions OPTIONAL
    } OPTIONAL,
    invalidCertificates ::= SEQUENCE OF SEQ UENCE {
        userCertificate     CertificateSerialNumber,
        fromInvalid         [0] Time,
        toInvalid           [1] Time OPTIONAL
    } OPTIONAL,
    crlE xtensions [0] EXPLICIT Extensions OPTIONAL
}
```

그림 8. Historical CRL

IV. Conclusion

과거에 서명된 문서를 검증하기 위해서는 검증 시간에 인증서가 유효했는가를 판단하여야 한다. 본 논문에서는 인증서의 상태 정보를 얻는 대표적인 방법인 CRL과 OCSP를 이용하여 과거의 인증서 상태 정보를 얻을 경우에 발생하는 문제점들을 자세히 살펴보았다. CRL은 실시간 인증서 상태 정보를 얻지 못한다는 문제와 인증서의 효력정지로 인해 잘못된 정보를 제공할 수 있다. 본 논문은 효력정지로 인한 문제를 해결하기 위한 확장 필드를 제안하였다.

OCSP는 인증서의 실시간 정보를 제공할 수 있지만, 프로토콜 구조로 인해 과거의 상태 정보를 요청 할 수 없었다. 이를 해결하기 위해 Request에 과거의 시간을 표현할 수 있는 필드가 요구되며, 인증기관이 발행한 인증서에 대한 과거 정보를 유지 관리하여야 한다.

과거의 인증서 상태 정보를 제공하기 위하여, 인증기관은 과거에 발행한 모든 CRL을 관리하여야 한다. 이를 보다 효율적으로 관리하기 위하여 Historical CRL을 제안하였다. Historical CRL은 발행한 CRL을 통합한 CRL로써, 인증기관의 CRL 관리 부담을 줄일 수 있다.

참고문헌

[1] J. Roh and K. Lee, "A Model of Certificate Validation Server," *ICOIN 2003*, vol.2, Feb 2003.

[2] M. Naor and K. Nissim, "Certificate Revocation and Certificate Update," *IEEE Journal on Selected Areas in Communications*, vol.18, no.4, Apr. 2000.

[3] P. Wohlmacher, "Digital Certificates: A Survey of Revocation Methods," *Proceedings of the 2000 ACM workshops on Multimedia*, 2000.

[4] P. Gutmann, "PKI: It's Not Dead, Just Resting," *IEEE Computer*, vol.35, no.8, pp.41-49, Aug 2002.

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

[6] R. Housley, "Cryptographic Message Syntax," RFC 2630, June 1999.

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

[8] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 Public Key Infrastructure, Certificate and CRL Profile," RFC 3280, Apr 2002.

[9] A. Nash, W. Duane, C. Joseph and D. Brink, *PKI: Implementing and Managing E-Security*, Osborne/McGraw-Hill, 2001.

[10] R. Housley and T. Polk, *Planning for PKI*, John Wiley & Sons, 2001..