情報保護學會論文誌
第13券 第1號, 2003. 2

클라이언트-서버환경에 적합한 효율적인 인증서상태 및 경로검증 시스템

최 영 철*, 박 상 준*, 원 동 호**

An Efficient Certificate Status and Path Validation System for Client-Server Environment

Young-Chul Choi*, Sang-Joon Park*, Dong-Ho Won**

요 약

최근 공개키기반구조(Public Key Infrastructure)에 관한 연구가 활발해지면서 클라이언트나 서버의 부하를 줄이고 효율적인 연산이 가능하도록 인증서상태 및 경로검증에 관한 연구가 활발히 진행되고 있다. 그러나, 많은 관련 연구에도 불구하고, 인터넷뱅킹과 같이 실시간 처리가 필요한 대규모 클라이언트-서버 환경에서 서버가 수 많은 클라이언트들의 인증서를 동시에 검증할 수 있는 효과적인 메커니즘은 현재까지 거의 전무한 상태이다. 본 논문에서는 기존의 표준 또는 제안된 방법론들이 이러한 대규모 클라이언트-서버 환경에 적합하지 않음을 보이고, 이러한 환경에 적합한 새로운 형태의 인증서상태 및 경로검증 시스템을 제안하고자 한다.

ABSTRACT

As a research on PKI(Public Key Infrastructure) is being very popular, the study relating to certificate status and path validation is being grown with aim to reduce an overhead of the protocol and to provide an efficient operation. But in spite of a lot of related research, there is still almost no protocol that we can use for real-time based client-server environment with large scale like internet banking. In this paper, we shows that the existing standards or protocols are not suitable to be used for such a real-time based client-server environment with large scale, and then proposes an efficient certificate status and path validation system.

Keyword : *Certificate Status and Path Validation, Certificate Revocation List*

Ⅰ. 서 론

최근 공개키기반구조(Public Key Infrastructure, 이하 PKI라 칭함) 구축에 관한 연구가 활발해지면서 클라이언트나 서버의 부하를 줄이고 효율적인 연산이 가능한 인증서상태 및 경로검증에 관한 연구가 늘어나고 있다. 그러나, 많은 관련 연구에도 불구하고, 현재까지 제안된 주요 메커니즘들은 현실적인 고려사항을 충분히 반영시키지 못하거나, 잠재적인 문제점을 가지고 있는 경우가 대부분이다. 인증서상태검증은 인증서 표준이 제정된 이후로 계속적으로 연구되고 있는 분야로서 인증서 표준에 함께 포함된 인증서폐지목록(Certificate Revocation LIst, 이하 CRL이라 칭함)이 현재까지 가장 널리 사용되고 있는 기본적인 해결

* (주) 비씨큐어({ycchoi, sangjoon}@bcqre.com)
** 성균관대학교 정보통신공학부 정교수(dhwon@dosan.skku.ac.kr)

방법이다[1]. 이후, 단순한 CRL 방식만으로는 완전한 시스템 구축의 한계를 가지게 되자 보다 효율적인 방식의 CRL 사용이 가능토록 여러 가지 부가 기능들이 결합된 델타(Delta) CRL, CRL DP(Distribution Point), 최신(Freshest) CRL 등의 방법들이 개발되어 사용되고 있다. 그러나, 이러한 노력에도 불구하고 CRL 방법론이 잠재적인 문제점들을 근본적으로 해결할 수 없게 되자, 일부 학자들은 다른 측면의 접근방법을 가지고 새로운 대안을 제시하기 시작하였다. 이러한 새로운 방법론은 기존 방식의 효율성을 개선한 새로운 인증서상태검증 방법과 기존 방식의 잠재적 문제점을 근본적으로 해결하고자 하는 온라인상태검증 방법으로 대별된다. CRT(Certificate Revocation Tree)[2], NOVOMODO[3] 등이 전자에 해당하고, OCSP(Online Certificate Status Protocol)[4], SCVP(Simplified Certificate Validation Protocol)[5] 등이 후자에 해당하는 대표적인 방법이다. 그러나, 상기 방법론들 중에서 초당 동시 접속 사용자 수가 수 십명이 넘는 인터넷뱅킹이나 온라인주식트레이딩 시스템과 같이 실시간 처리에 기반을 둔 대규모 클라이언트-서버 환경에서 사용할 수 있는 방법론은 거의 전무한 상태이다.

CRL 방법론은 방법 자체가 가지고 있는 시간격차(time-gap) 문제와 물리적 파일 처리의 과부하로 인하여 사용에 많은 제약이 따르게 된다. 근래 표준화된 OCSP 방법론은 OCSP 서버 자체의 전자서명 수행과 수신 서버의 전자서명 검증 수행에 따른 과부하 때문에 실제 시스템 사용에 어려움을 가지게 된다.

결국 상기 문제를 해결할 수 있는 대안을 제시하지 못한다면 그러한 대규모 클라이언트-서버 환경에서 PKI 기반의 응용 시스템을 도입하는 것은 어렵게 될 것이다.

본 논문에서는 기존 메커니즘과는 다른 새로운 개념의 접근 방법을 통하여 이러한 실시간 기반의 대규모 클라이언트-서버 환경에 보다 완전하고 효율적으로 접근할 수 있도록 안전한 인증서상태 및 경로검증 시스템을 제안하고자 한다. 이를 위하여 기존의 방법론들이 실시간 기반의 대규모 클라이언트-서버 환경에 적절하지 않음을 새로운 평가 항목 정의를 통하여 설명하고, 제안 시스템의 필요성을 제시한다.

제안 시스템은 기존 방법론들에서 신뢰당사자 서버가 주로 수행해야 할 초당 수십 번의 전자서명 검증 연산 및 네트워크 접근등과 같은 부가 작업들을 클라이언트의 사전 검증 방법으로 전환하고 해당 검증 결과를 해쉬함수 연산 값으로 표현함으로서 신뢰당사자 서버가 네트워크 접근 없이 단지 한 번의 해쉬함수 연산만으로서 클라이언트의 인증서를 검증할 수 있도록 해준다.

본 논문 II절에서는 기 제안되거나 표준화된 메커니즘과 제안된 시스템들의 객관적이고 정량적인 비교 분석을 위한 실제 사용환경모델과 구체적 평가기준을 정의한다. 이때 해당 평가기준은 클라이언트-서버 환경 기반의 서버 관점에서 이루어짐을 전제한다. III절에서는 관련 연구 분석과 제안 시스템의 필요성을 기술한다. IV절에서는 제안 시스템을 상세 설명하고, V절에서는 제안된 시스템과 기존 시스템들 간의 완전성, 효율성, 안전성 측면에서 객관적이고 정량적 비교를 보여준다. 그리고, 마지막으로 VI절에서 결론을 맺는다.

II. 사용환경모델 및 평가기준 정의

보다 정확하고 효율적인 인증서상태 및 경로검증 시스템의 개발을 위해서는 무엇보다도 대상 응용의 정확한 환경모델 분석이 사전에 요구된다. 그러한 환경 모델 분석을 통하여 적절한 평가기준 요소가 마련되며, 이를 통하여 해당 제안 시스템의 정확한 평가가 이루어지게 된다. 즉, 본 절에서는 하나의 표준화된 방법론이 모든 모델에 적절하게 사용되기는 어렵기 때문에 본 논문에서 목표로 설정하고 있는 클라이언트-서버 환경모델을 정의하고, 이에 대한 시스템 평가를 위하여 해당 클라이언트-서버 환경모델에 알맞는 평가기준을 정의한다. 이러한 평가기준은 해당 환경모델에 대하여 기존 방법론들이 적절하게 사용될 수 없음을 보여주는 근거가 되며, 동시에 제안 시스템의 효율성을 보여주는 근거로 사용된다.

2.1 사용환경모델 정의

PKI가 적용되는 환경은 매우 다양하고 광범위하다. 그러나, 그 적용 부분을 응용 용도에 관계없이 적용시스템으로만 한정시켜 분석한다면 거의 모든 대상들이 클라이언트-서버 모델로 정의될 수 있다. 즉, 인증서 소유자(certificate holder)나 인증서를 사용하게 되는 신뢰당사자(relying party)는 클라이언트나 서버 중의 하나로 할당될 수 있으며, 이를 통하여 PKI 사용 환경모델은 [표 1]과 같이 클라이언트-서버 구성으로 모델화 될 수 있다.

〔표 1〕 사용환경모델 정의

구분	응용 예
클라이언트-클라이언트	개인간 거래(P2P)
클라이언트-서버	인터넷뱅킹 등(B2C)
서버-서버	기업간거래 등(B2B)

클라이언트-클라이언트 모델은 개인 간 거래 모델로서 P2P(Peer-to-Peer) 기반의 데이터 거래나 문서 공유 환경을 나타내는 것이라 할 수 있다. 이 모델의 특징은 인증서상태 및 경로검증의 주체가 클라이언트가 된다는 것이며, 이는 곧 성능보다는 시스템의 완전성 부분이 더욱 중요한 고려사항이 되어야 한다는 것을 의미한다. 즉, 이러한 클라이언트-클라이언트 환경에서는 OCSP나 SCVP와 같은 완전성이 뛰어난 온라인 상태확인 검증시스템을 사용하는 것이 효율적이라고 볼 수 있다.

클라이언트-서버 모델은 주로 개인과 기업 간 거래의 형태로서 최근 급성장하고 있는 인터넷뱅킹, 인터넷주식트레이딩, 인터넷쇼핑 등의 분야가 그 예이다. 이 모델은 본 논문의 제안 시스템이 적용 대상으로 삼는 목표 모델이 되며 인증서상태 및 경로검증의 주체는 서버가 된다. 여기서 서버는 동시에 수 많은 클라이언트들로부터 거래 요구를 받았을 경우 해당 클라이언트의 전자서명 검증이 성공적으로 완료되어야만 다음 거래를 처리하게 된다. 이 때, 서버는 동시 다발적인 인증서상태 및 경로검증을 수행해야 하며 이로 인해 본 모델에서는 완전성과 효율성이 모두 적절히 만족되어야 한다.

서버-서버 모델은 기업 간 거래 모델로서 인터넷 복권 등과 같은 응용에서 대리점과 본사 간의 웹 서버 간 통신 등이 그 예가 될 수 있다. 이 모델 또한 서버 간 거래이기 때문에 많은 거래처리가 이루어져야 하며 이로인해 클라이언트-서버 모델과 같이 완전성과 효율성이 모두 고려되어야 한다.

2.2 평가기준 정의

가정이 될 수 있는 환경모델을 정의하였다면 다음으로는 해당 환경모델에서 제안된 시스템이 얼마나 적절하게 적용될 수 있는지에 대한 시스템 평가기준이 필요하게 된다. 본 절에서는 환경모델에서 정의된 대규모 클라이언트-서버 환경에 적합한 평가기준을 제시하고자 한다. 즉, 인증서를 검증하는 주체는 대규모 클라이언트를 가지고 있는 서버가 되며, 해당 평가기준은 서버측 관점에서 기술된다. 전체 평가요소는 크게 완전성(completeness), 효율성(effectiveness), 안전성(security)으로 대별된다.

2.2.1 완전성(completeness)

■ 정의1 : 완전성
인증서상태검증에서 다음과 같은 두 조건을 만족하면 "해당 시스템은 완전성을 갖는다."라고 정의한다.
- 조건1 : 제안 방법은 시간격차 문제를 갖지 않는다.
- 조건2 : 제안 방법은 추가 가정을 갖지 않는다.

조건1의 시간격차 문제는 CRL 기반의 정보 다운로드(PULL 방식이라 정의함, 이하 PULL 방식이라 칭함) 방식이 갖는 가장 근본적인 문제를 말하는 것이다. CRL 유형의 접근법은 사용자가 자신의 인증서를 폐지하는 경우 인증기관이 해당 폐지 인증서의 일련번호(serial number)를 CRL에 포함시킨 후 디렉토리 서버와 같은 저장소에 해당 CRL을 공지하는 형식을 취한다.

이 경우 해당 CRL은 일반적으로 사용자 검색에 따른 PULL 방식으로 해당 저장소로부터 사용자까지 전달되거나, 또는 인증기관이 직접적으로 사용자의 저장소에 해당 CRL을 배포하는 방식(PUSH 방식이라 정의함, 이하 PUSH 방식이라 칭함)으로 전달될 수도 있다. 그러나 일반적으로 PUSH 방식은 사용자의 수가 너무 많거나 PKI 도메인 영역이 서로 다른 경우 구현 및 관리가 어렵기 때문에 일반적으로 사용되지는 않으며, 사실상 CRL 방법론이라 함은 PULL 방식을 사용하는 것으로 해석할 수 있다. 이때 PULL 방식은 항상 시간격차 문제를 갖게 된다. 시간격차 문제는 아래와 같이 두 가지 관점에서 분류할 수 있다.

- 주기적 CRL 발행에 따른 시간격차 문제 : 만약 인증기관이 CRL을 주기적으로 발행한다면(CRL의 다음갱신주기(next update)마다 새로운 CRL 발행) 해당 CRL이 발행된 이후 다음 CRL 갱신 전까지는 인증서 폐지 사실을 공지할 수 없게 되며, 결국 해당 CRL의 주기 동안 폐지되는 인증서의 상태정보를 확인할 수 없게 되는 시간격차 문제를 갖게 된다.
- 신뢰당사자의 주기적 다운로드에 따른 시간격차 문제 : PULL 방식에서 신뢰당사자는 인증기관의 저장소로부터 CRL을 다운로드 받는다. 이때, 일반적으로 주기적 방식의 다운로드를 수행하게 되는데 이 경우 CRL의 유효기간 동안 또는 주기적 다

운로드 간격 기간동안 신뢰당사자는 최신 인증서 상태 정보를 취할 수 없게 되는 시간격차 문제를 갖게 된다.

조건2는 제안 시스템이 인증서상태검증에 추가적 가정을 갖지 말아야 한다는 조건이다. 예를 들면, OCSP의 경우 실시간 응답 프로토콜이기 때문에 시간격차 문제는 가지지 않지만 OCSP 서버가 전자서명을 통하여 응답 메시지를 전달하기 때문에 해당 신뢰당사자는 다시금 OCSP 서버 인증서의 검증 문제를 해결해야만 하는 재귀적인 문제에 부딪히게 된다. 이 때, 해당 문제를 해결하기 위해 "OCSP 서버는 단기 주기(short-lived) 인증서를 사용할 수 있다."라고 정의하거나 "OCSP 서버는 인증기관에 의해 직접 관리되기 때문에 인증서상태검증을 생략할 수 있다."라고 정의한다면 이것은 실제 시스템의 구현을 어렵게 하고, 안전성을 저하시키는 추가 가정이 되는 것이다. 결국 보다 완전한 시스템을 위해서는 그러한 추가 가정을 하지 말아야 한다. 제안 시스템이 이러한 추가가정이 없는 경우 조건2는 만족된다. 어떤 시스템에서 조건1과 조건2가 만족되는 경우 우리는 "해당 시스템은 완전성을 제공한다."라고 정의한다.

2.2.2 효율성(efficiency)

■ 정의2 : 효율성
인증서상태검증이나 경로검증 관점에서 시스템A가 다음과 같은 평가요소에서 시스템B 보다 우월한(횟수가 보다 적은) 경우 "시스템A는 시스템B보다 효율적이다."라고 정의한다.
- 요소1 : 네트워크 접근의 횟수
- 요소2 : 파일(CRL) 접근의 횟수
- 요소3 : 암호연산의 횟수
  - 공개키연산의 횟수
  - 해쉬연산의 횟수

인증서상태 및 경로검증 프로토콜의 경우 제안한 시스템이 효율적임을 증명하기 위해서는 기본적인 평가 요소가 있어야만 한다. 서론에서 설명한 바와 같이 현재까지 제안되거나 표준화된 인증서상태 및 경로검증 방법론들이 대규모 클라이언트-서버 환경에서 사용하기 어렵다는 것은 상기 평가 요소를 기반으로 트랜잭션 양을 계산하면 그 이유가 명백해진다.

제안 시스템 역시 이러한 맥락에서 상기 평가 요소를 가지고 평가함으로써 대규모 클라이언트-서버 환경에서 효과적으로 사용될 수 있음을 증명할 수 있게 된다. 요소별 설명과정을 위하여 "인증서 소유자인 클라이언트가 전자서명 메시지를 자신의 인증서 및 인증경로와 함께(PKCS#7[6]사용) 신뢰당사자인 서버로 전송한다."라는 상황을 가정한다.

요소1은 가장 일반적인 항목으로서 서버가 해당 전자서명과 인증경로를 클라이언트로부터 수신한 후 해당 인증서의 상태 검증을 위하여 네트워크를 이용한 외부 시스템과 통신하는 횟수를 지칭한다. 예를 들어, 해당 인증서에 대응되는 CRL을 가져오기 위해서는 서버는 기본적으로 최소 '1'회의 네트워크 접근이 필요하게 된다. 네트워크 접근은 기본적으로 상대 시스템의 현재 상태, 네트워크의 대역폭(bandwidth), 그리고 미리 설정되는 타임아웃(timeout)의 길이에 따라 많은 성능 차이를 나타내므로, 네트워크 접근의 횟수가 많으면 많을수록 해당 시스템은 비효율적이게 된다. 특히, 실시간 환경에서는 이러한 과다한 네트워크 접근이 응답시간 지연으로 나타남으로써 급격한 시스템 성능 저하를 불러일으킬 수 있다.

요소2는 파일 접근의 횟수로서 일반적으로 CRL 파일에 접근하는 횟수를 의미한다. 다중 클라이언트를 갖는 서버에서 파일 접근의 횟수는 서버 성능에 큰 영향을 미치게 된다. 특히, 동시 접속자로 인하여 생성된 다중 프로세스가 동일한 파일에 접근하는 경우 해당 시스템의 하드디스크 성능과 전체 시스템 성능은 큰 영향을 받게 된다(일반적으로 안정성 제고를 위하여 세마포어(semaphore) 같은 임계영역(mutual exclusion) 제어방식을 사용하게 되며, 이 경우 시스템 성능은 저하된다). 그러므로, 해당 파일의 접근 횟수가 적으면 적을수록 시스템의 효율성은 높아지게 된다.

요소3은 암호연산의 횟수로서 인증서상태검증이나 경로검증을 수행하는 경우 해당 시스템은 공개키암호연산(전자서명 검증), 또는 해쉬연산 등과 같은 암호연산을 수행하게 된다. 암호 연산은 그 종류와 횟수에 따라 시스템 성능에 영향을 미치는 요소이기에 효율성의 올바른 평가를 위해서는 정확한 암호연산의 종류와 사용횟수가 평가되어야만 한다.

2.2.3 안전성(security)

■ 정의3 : 안전성
시스템A가 다음과 같은 조건을 만족하는 경우 "시스템A는 안전하다."라고 정의한다. 여기에서 정의되는 안전성은 암호알고리즘의 근본적인 안전성이 아니라 시스템 프로토콜 차원의 안전성을 의미한다.
- 조건1 : 위조 및 변조의 위협 방지
- 조건2 : 위장 및 행위 부인의 위협 방지
- 조건3 : 재사용의 위협 방지

인증서상태 및 경로검증 프로토콜의 경우 "제안한 시스템이 안전한가?"임을 증명하기 위해서 몇 가지 조건의 요구사항을 만족해야만 한다. 본 절에서 정의하는 안전성은 해당 시스템 프로토콜의 안전성을 정의하는 것이며, 사용되는 암호알고리즘이나 사용방식의 안전성을 의미하는 것은 아니다.

조건1은 위조 및 변조의 위협으로부터 안전해야 한다는 것이다. 인증서상태 및 경로검증의 주체가 되는 메시지는 외부의 위·변조로부터 안전해야만 한다. 즉, 제3자가 해당 메시지를 위·변조하여 인증서 상태검증이나 경로검증의 사실을 왜곡할 수 없어야 한다는 것이다.

조건2는 위장 및 행위 부인의 위협에 대한 방지로서 결과 메시지 생성자가 인증서상태검증이나 경로검증에 대한 메시지 생성 후 이에 대한 행위 사실을 부인할 수 없어야 한다는 것이다. 이는 메시지를 생성하는 신뢰기관의 부인방지를 의미한다.

조건3은 어떤 인증서가 폐지되었는데도 불구하고 제3자 또는 인증서 소유자가 그 이전의 유효한 상태검증 메시지를 재사용하여 서버를 속이는 행위에 대한 방지를 의미한다.

Ⅲ. 관련 연구 및 제안 시스템의 필요성

본 장에서는 제안 시스템을 설명하기 전에 필요한 관련 연구 현황과 제안 시스템의 필요성에 대하여 설명하고자 한다. 1절에서 관련 연구를 설명하고, 2절에서 최근 기존 제안 시스템을 문제점을 상세히 살펴보며, 이를 통하여 제안 시스템의 필요성을 설명한다.

3.1 관련 연구

3.1.1 CRL 기반 방법론

3.1.1.1 CRL 분배점(Distribution Point,DP)

일반적인 CRL 방법론에서 CRL의 크기를 줄이기 위하여 개발된 방법으로서 인증서확장영역에 해당 인증서가 폐지되는 경우 해당 인증서의 일련번호 범위또는 인증서 폐지 사유에 따라 등록될 CRL의 위치정보가 저장된다. 이후, 해당 인증서의 폐지여부를 검사하기 위해서는 해당 인증서의 CRL 분배점 정보를 참조하여 CRL 확인을 수행한다.

3.1.1.2 델타(Delta) CRL

델타 CRL 역시 CRL의 크기 증가 문제를 줄이기 위한 대안이나 CRL 분배점과 다른 점은 인증서 일련번호나 인증서폐지사유에 따라 CRL을 나누는 것이 아니라 CRL의 일정 크기에 따라 기본 CRL과 그 나머지가 되는 델타 CRL로 나눈다는 것이다. 신뢰당사자들은 최근 CRL 갱신 시 델타 CRL 만을 다운로드 받음으로써 다운로드의 부하를 감소시키는 모델이다. 물론 CRL 확인 시에는 기존의 기본 CRL을 반드시 함께 사용해야만 한다.

3.1.1.3 추가 발행(Over-issued) CRL

일반적인 CRL 메커니즘의 경우 주기적으로 발행되는 다음갱신주기(next update) 시간에 CRL 발행기관은 급작스러운 자원 사용에 따른 통신과부하 상태에 이르게 된다. 이러한 문제를 해결하기 위하여 CRL 발행기관은 주기적 발행뿐만 아니라 비주기적(또는 실시간) CRL 발행을 통하여 동일 시간 내에 여러 개의 유효한 CRL을 사용 가능케 한다. 이를 통하여 해당 CRL 발행기관은 갑작스런 통신 과부하 문제로부터 벗어날 수 있다[8].

3.1.1.4 간접(Indirect) CRL

간접 CRL은 인증기관이 아닌 CRL 전문 발행기관이 다수의 인증기관을 대행하여 CRL을 발행하는 메커니즘으로써 이를 통하여 신뢰당사자는 여러 개의 CRL을 관리하는 어려움으로부터 벗어날 수 있게 된다.

3.1.1.5 동적(Dynamic) CRL 분배점(DP)

동적 CRL 분배점 스킴은 기존 CRL 분배점이 갖고 있는 초기 분배 간격 설정 후 변경할 수 없는 분배 간격의 문제를 동적 방법으로 해결해주는 방식이다[9]. 이를 통하여 CRL 발행자는 운영 중에도 인증서 발행 추이에 따라 동적으로 분배 간격을 조정할 수 있다.

3.1.1.6 최신(Freshest) CRL

최신 CRL은 인증서확장영역에 포함되는 정보로서 델타 CRL의 최신 갱신 위치정보를 가지고 있다. 이를 통하여 신뢰당사자는 최신 델타 CRL을 다운로드 받을 수 있다[1].

3.1.2 실시간 상태 확인 기반 방법론

3.1.2.1 OCSP(Online Certificate Status Protocol)

OCSP는 온라인방식의 인증서상태확인 프로토콜로서 신뢰당사자가 인증서의 상태검증을 원하는 경우 해당 인증서의 일련번호를 OCSP 서버로 전송하면 OCSP

서버는 인증서상태 확인 후 전자서명된 상태 결과를 신뢰당사자에게 되돌린다. 그러므로, 신뢰당사자는 CRL의 시간격차 문제없이 인증서상태검증을 수행할 수 있다. 그러나, OCSP는 서버가 해당 결과에 전자서명을 수행하는 방식이기 때문에 OCSP 서버와 수신자 모두에게 전자서명 생성 및 검증에 따른 과부하를 발생시키게 된다.

3.1.2.2 SCVP(Simplified Certificate Validation Protocol)

SCVP는 OCSP의 기능 외에도 추가적으로 인증서 경로검증에 대한 기능도 신뢰당사자에게 제공한다. 그러므로, 신뢰당사자는 SCVP를 이용하여 자신이 수행해야 할 인증서상태 및 경로검증의 부하를 모두 줄일 수 있게 된다. 그러나, SCVP 역시 OCSP와 동일하게 응답결과에 전자서명을 수행해야 하기 때문에 OCSP와 동일한 문제점을 갖게 된다. 옵션사항으로서 전자서명이 없는 결과 메시지 전송이 가능하도록 정의되어있기는 하나 이는 안전성을 보장할 수 없는 트랜잭션이기 때문에 실제 사용하기 어려운 옵션이 된다.

3.1.3 비표준 신개념 기반 방법론

3.1.3.1 CRT(Certificate Revocation Tree)

CRT는 머클해쉬트리(Merkle hash tree)를 이용하여 인증서상태정보를 표현하는 방법으로서 폐지정보의 전체 크기가 크다고 할지라도 약20바이트(byte)이면 충분하게 된다. CRT를 이용함으로써 신뢰당사자는 보다 빠른 검색이 가능하며, 전체 데이터 크기가 작기 때문에 대규모 도메인에 적합하게 된다. 그러나, 하나의 인증서상태정보 갱신 마다 모든 트리 노드를 재계산하여 반영해야만 하는 문제점을 가지고 있다.

3.1.3.2 NOVOMODO

NOVOMODO는 초기 CRS(Certificate Revocation Status) 디렉토리 스킴으로 알려져 발표되었으며, 전자서명 없이 해쉬함수만을 이용하여 인증서상태정보를 확인할 수 있게 해준다. 그러나, NOVOMODO는 몇 가지 제약적인 요소를 갖게 된다. 첫 번째는 시간간격에 따라 인증기관이 비밀정보로 관리해야 할 데이터가 너무 많다는 것이다. 예를 들어 하루 주기로 인증서 폐지 관련 정보를 공고한다고 가정하면, 인증기관은 사용자 인증서 발급 시 20바이트 크기를 갖는 366개의 해쉬함수 연산결과를 인증서 유효기간동안 비밀정보로 관리해야만 한다. 즉, 인증기관의 가입자 수가 10만명이라고 가정한다면, 인증기관은 3,660만개의 비밀데이터를 지속적으로 관리해야 되는 것이다. 이는 실제 구현 환경에서 커다란 장애요인으로 대두된다. 두 번째는 유효성을 알리는 시간 간격을 미리 설정하여 그 정보를 배포하기 때문에 CRL과 같은 시간 간격에 따른 시간격차 문제를 동일하게 갖게 되는바 실시간 인증서상태 검증이 필요한 실시간 응용환경에서는 적용하기 어렵다.

3.2 제안 시스템의 필요성

상기 절에서 고찰한 방법론들 중 인터넷뱅킹이나 인터넷주식트레이딩 시스템 등과 같은 대규모 실시간 클라이언트-서버 환경에 사용할 수 있는 방법론은 거의 전무한 상태이다. CRL 방법론은 시간 격차 문제와 시스템 효율성 측면에 사용이 거의 불가능하다. 현재 제안되거나 표준화된 방법론들 중에서 사용할 수 있는 가장 유력한 방법은 OCSP 표준이며, 추가적으로 NOVOMODO 방법도 고려해 볼 수 있다. 그러나, 상기 2가지 방법을 객관적 관점에서 분석해보면 몇 가지 문제점들을 가지며, 이러한 문제점들은 사용의 중대한 장애요인으로 작용함으로서 현재까지도 이러한 방법론들이 인터넷뱅킹이나 인터넷주식 거래 등에 사용되지 않고 있는 실정이다. 본 절에서는 사용가능성이 있는 상기 2개의 방법론에 대하여 상세히 살펴보고자 한다.

[OCSP 방법론의 문제점]

- 문제1. 서비스 요청 서버의 과부하 문제 : 인터넷 뱅킹 환경에서 뱅킹서버는 클라이언트의 전자서명을 검증하기 전 해당 인증서의 검증요청을 OCSP 서버에 전달한다. 이후 요청 서버는 전자서명된 결과를 전달받으며, 이를 위해 1번의 전자서명 검증과정을 수행해야 한다. 즉, 클라이언트의 전자서명을 검증하기 위해 2회의 전자서명 검증과정을 거쳐야만 한다. 또한, 해당 결과를 검증하는 과정 중에 OCSP 서버의 인증서에 대한 유효성 확인 부분이 고려되어야 하며, 이는 추가 가정이 되는 오버헤드가 된다.
- 문제2. OCSP 서버 자체의 과부하 문제 : OCSP 서버는 응답결과를 생성하기 위하여 반드시 전자서명을 수행해야만 한다. 이 경우 OCSP 이용 대상자가 10개의 인터넷뱅킹 서버라고 가정하고, 각 뱅킹 서버의 초당 동시접속자 수가 30명이라고 가정한

다면, OCSP 서버는 초당 300건의 전자서명을 수행해야 한다는 결과가 나온다. 이는 OCSP 서버 자체에 과부하를 일으키게 되며, 결국 서비스 요청 서버에 대한 심각한 응답지연을 일으켜 인터넷뱅킹 서버의 시스템 장애를 일으킬 수 있다.

[NOVOMODO 방법론의 문제점]

- 문제1. 인증기관의 대규모 비밀정보 유지관리의 문제 : NOVOMODO 스킴의 핵심은 인증서 확장영역 내에 인증서 유효 정보와 인증서 폐지 정보를 구분하기 위하여 해쉬함수 결과 값을 표현할 수 있는 2개의 확장영역을 추가하여 이를 이용하는 것이다. 해당 인증기관은 사용자 인증서 발급 시 아래와 같은 절차를 거친다.
  - 인증기관은 서로 다른 난수 X_0, Y_0를 생성
  - $Y_1=H(Y_0)$ where Y_1 공개정보, Y_0 비밀정보
  - $X_1=H(X_0)$,$X_2=H(X_1)$, ... ,$X_{365}=H(X_{364})$ where X_{365}는 공개정보, X_0,X_1, ... X_{364}는 비밀정보
  - 인증기관은 인증서 발급 시 인증서 유효성을 알리기 위한 정보 X_{365}와 인증서 폐지 사실을 알리기 위한 정보 Y_1을 각각 인증서 추가 확장영역에 삽입
  - 인증기관은 공개된 특정 저장소에 인증서상태에 관한 정보를 매일 갱신. 만약 인증서가 유효하면 X_{364}, ..., X_0를 차례대로 매일 공지하고, 폐지되었다면 Y_0를 공지하여 이를 사용자에게 알림
  - 인증서를 검증하고자 하는 신뢰당사자는 인증기관이 매일 공지하는 인증서상태 정보를 가져와서 Y_0가 공지된 경우에는 인증서 내의 Y_1정보가 $Y_1=H(Y_0)$를 만족하는지 확인하며, 만약 X_{363}이 공지되었다면 $X_{364}=H(X_{363})$, $X_{365}=(X_{364})$의 만족여부를 확인함으로서 인증서상태 확인을 수행

  상기 스킴에서 보는 바와 같이 인증기관은 매일 인증서상태 확인 정보를 공고한다고 가정한다면 366개의 각 20바이트 해쉬함수 결과 값을 비밀정보로 유지해야 하며, 사용자가 수가 10만명이라고 가정하면 3,660만개의 비밀정보를 관리해야만 하는 심각한 난관에 부딪히게 된다.
- 문제2. 인증서 표준 프로파일의 변경 문제 : 제안하는 스킴은 인증서 표준에 정의된 확장영역에 추가적으로 20바이트 스트링 저장을 위한 추가 확장영역을 정의하여 사용해야만 한다. 이는 현실적으로 받아들이기 어려운 문제가 된다. 이미 전 세계에 수많은 인증기관이 기존 표준을 준용하여 시스템을 구축하고 이미 사용자가 수백만~수천만 명인 상황에서 인증기관 서버 시스템을 수정, 변경해야만 하는 문제를 쉽게 수용하기는 힘들기 때문이다.

결론적으로 현재 표준화되거나 제안된 방법론들 중에서 인터넷뱅킹이나 인터넷주식트레이딩 시스템 등과 같은 대규모 실시간 클라이언트-서버 환경에 사용할 수 있는 방법론은 거의 전무한 상태이다. 그러므로, 조금 더 현실성 있는 모델과 일반적이면서도 특수한 대규모 클라이언트-서버 환경에서 현실적으로 사용할 수 있는 시스템의 설계가 요구되는 것이다.

Ⅳ. 제안 시스템

본 장에서는 제안 시스템의 구성 요소, 기호 및 정의, 그리고 구체적인 시스템 프로토콜을 설명한다. 1절에서 본 제안 시스템을 위한 구성요소를 설명하고, 2절에서 기호 및 정의를 수행하고, 마지막 3절에서 구체적인 시스템 프로토콜을 기술한다.

4.1 구성요소

- 인증기관(Certification Authority, CA) : 가입자(클라이언트 또는 서버)에게 인증서를 발급해 주거나 관리 기능을 제공해주는 신뢰기관이다.
- 등록기관(Registration Authority, RA) : 인증서 발급 및 관리를 위한 사용자 신원확인을 수행하고 이에 대한 등록 업무를 담당하는 기관으로서 인증기관에 의해 운영될 수도 있으며, 제3의 독립기관에 의해 운영될 수도 있다.
- 검증기관(Validation Authority, VA) : 클라이언트-서버 환경에서 인증서를 발급받은 클라이언트가 서버에 인증서 기반의 로그온이나 전자서명을 전송하기 전 자신의 인증서에 대한 인증서상태 및 경로검증에 대한 검증서(Validation Certificate)를 발급받고자 하는 경우 이에 대한 검증서를 발급해주는 신뢰기관이다. 검증기관은 OCSP 서버를 운영하는 기관과 동등한 형태의 기관이며, 인증기관의 부속기관으로 존재하거나 독립적인 기관으로 존재할 수도 있다.
- 인증서소유자_서버(Certificate Holder_S, CS) : 인증기관의 가입자로서 대규모 클라이언트-서버 환경에서 서버 가입자를 의미한다.
- 인증서소유자_클라이언트(Certificate Holder_C, CC) :

인증기관의 가입자로서 대규모 클라이언트-서버 환경에서 클라이언트 가입자를 의미한다.

4.2 약어 및 기호 정의

[약어 정의]

- M : 메시지(Message) - 인증서소유자_클라이언트(이하 CC라 칭함)가 인증서소유자_서버(이하 CS라 칭함)에게 보내고자 하는 전자서명의 대상이 되는 원문이다.
- S : 전자서명(Digital Signature) - CC가 CS에게 보내고자하는 메시지에 대한 전자서명 결과로서 PKCS#1 형태의 구문을 따른다.
- RN : 난수(Random Number) - CS가 생성하는 난수로서 검증서 발급 및 사용 프로토콜에서 CC나 또는 악의적 공격자의 검증서 정보 재사용을 방지하기 위한 정보로 사용된다. 난수 생성은 X9.31[11]이나 FIPS 186-1[12]등과 같이 표준으로 정의된 절차를 따르는 것으로 가정한다.
- CSI : 인증서상태정보(Certificate Status Info.) - 검증기관(이하 VA라 칭함)이 CC의 검증서 발급요청에 대한 결과로 전달하는 정보들 중 CC의 인증서에 대한 상태를 나타내는 정보로서 유효(Valid), 폐지(Revocation), 효력정지(Suspension), 미확인(Unknown)들 중 하나의 결과 값을 갖는다.
- PVI : 경로검증정보(Path Validation Info.) - VA가 CC의 검증서 발급요청에 대한 결과로 전달하는 정보들 중 CC의 인증서에 대한 경로검증결과를 나타내는 정보로서 성공(Sucess), 실패(Failure)들 중 하나의 결과 값을 갖는다.
- VC : 검증서(Validation Certificate) - VA가 CC의 검증서 발급요청에 대한 결과로 전달하는 최종정보로서 메시지 형식은 다음과 같다 : $(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| H(M \| CSVA_{DH}) \| H(RN \| SVA_{DH})$.

[기호 정의]

- $\|$: 문자열을 연결하는 연접(concatenation)
- $CERT_X$: X가 인증기관(이하 CA라 칭함)로부터 발급받은 인증서
- $CERTP_X$: X의 인증서를 포함하는 전체인증경로로서 X가 인증서를 발급받을 때 CA가 $CERT_X$를 포함시켜 전달하는 정보임(PKCS#7 형식 사용). 이때 최상위 인증기관의 인증서는 표준[1]에서 권고하는 바와 같이 인증서신뢰리스트(Certificate Trust List, CTL) 형태로 사용자 소프트웨어와 함께 사전에 배포되어야 함.
- DN_X : $CERT_X$의 기본영역 내에 정의되어 있는 주체이름(subject name)
- PRI_X : 인증서 발급요청 시 X가 생성하는 개인키
- CS_{DH} : CS의 Diffie-Hellman(이하 DH라 칭함) 공개키 정보($CS_{DH}=g^a \bmod p$, p는 소수, g는 원시원소, a는 비밀정보)
- SCS_{DH} : CS_{DH}의 비밀정보 a($CS_{DH}=g^a \bmod p$에서 비밀정보 a)
- VA_{DH} : VA의 DH 공개키 정보($VA_{DH}=g^b \bmod p$, b는 비밀정보)
- SVA_{DH} : VA_{DH}의 비밀정보 b($VA_{DH}=g^b \bmod p$에서 비밀정보 b)
- $CSVA_{DH}$: CS와 VA가 공유하는 DH 비밀 공유키 ($CSVA_{PH}=g^{ab} \bmod p$, a,b는 비밀정보)
- $S_{PRI_X}(M)$: X의 개인키를 이용한 메시지 M의 전자서명. 서명표준은 PKCS#7를 준용하는 것으로 가정(즉, 연산결과는 $M \| S \| CERTP_X$ 구조를 가짐)
- $V_{CERT_X}(M \| S \| CERTP_X)$: $S_{PRI_X}(M)$의 연산결과인 PKCS#7 전자서명에 대한 전자서명 검증
- H(M) : 표준화되어 사용되고 있는 128비트 이상 출력 값을 갖는 충돌회피(collision-free) 해쉬함수를 이용한 메시지 M의 축약

4.3 제안 시스템

4.3.1 목표 요구 조건

> II장 2절의 평가기준 정의1, 정의2, 정의3을 모두 만족시키는 클라이언트-서버 모델에 가장 적합한 인증서상태 및 경로검증 시스템

제안 시스템은 현재까지 제안되거나 표준화된 인증서상태 및 경로검증 프로토콜의 문제점 및 효율성을 획기적으로 개선시킨 시스템이다. 이는 목표 요구 조건의 만족 여부를 분석함으로써 증명될 수 있다.

4.3.2 제안 시스템 개념

CC는 CS에게 전자서명 메시지를 보내기 전 인증기관의 부속 시스템으로 존재하는 VA로부터 자신의 인증서에 대한 상태정보와 경로검증 정보에 대한 검증서를 받아서 이를 전자서명 메시지와 함께 CS에 전달하며, CS는 전자서명 검증 전 해당 검증서를 우

선적으로 확인한 후 유효한 경우 해당 인증서의 공개키를 이용하여 전자서명을 검증하게 된다. 이때, 해당 CS는 사전에 VA와 DH[9] 방식의 키 공유를 수행하고 공유된 비밀키(secret key) $CSVA_{DH}$는 VA와 CS 간 개체인증 비밀정보로 사용된다. 또한, 해당 비밀정보 생성에 사용된 SCS_{DH}와 SVA_{DH}는 $CSVA_{DH}$ 정보와 함께 각각 CS와 VA에 의해 안전한 저장소에 보관·관리된다. SCS_{DH}와 SVA_{DH} 정보는 향후 해당 검증서가 VA에 의해 생성된 것인지 CS에 의해 생성된 것인지를 구분할 수 있는 부인방지 증명용 비밀정보가 된다.

4.3.3 프로토콜 동작 절차

본 프로토콜에서 CS, CC, VA는 모두 인증기관으로부터 이미 인증서를 발급받은 것으로 가정한다. 즉, CS, CC, VA는 자신의 인증서를 포함한 전체 인증경로(CERTP)를 가지고 있으며, 이것은 일반화된 인증서 발급절차를 따른다[1].

[초기 등록 프로토콜]

초기 등록 CS의 초기 셋업 프로토콜로서, 클라이언트로부터 전자서명을 수신하는 CS와 VA 간 개체인증용 비밀정보를 공유하게 된다. 즉, CS가 CC들로부터 전자서명을 수신받기 이전에 효율적인 인증서 상태 및 경로검증을 위하여 CS와 VA간 필요한 정보를 설정하고 공유하는 과정의 프로토콜로서 CS가 시스템 설치 후 한번만 수행하면 되는 초기화 과정이다. 본 프로토콜을 통하여 해당 CS와 VA는 상호 개체인증용 비밀정보 $CSVA_{DH}$를 공유하게 되며, 이 정보는 VA로부터 발행된 증명서의 위·변조를 막아준다. $CSVA_{DH}$ 정보는 DH 방식의 키분배 프로토콜을 통해 생성되며, 생성된 키는 메시지 인증을 위한 비밀정보로 사용된다. 동시에 공유된 비밀정보는 향후 장기(long-term) 비밀키로 사용된다.

단계1

- CS는 VA와 DH 키분배를 수행하기 위하여 키분배 정보인 $S_{PRICS}(CERT_{CS} \| TIME1 \| CS_{DH})$를 생성한다. 여기서 TIME1은 재사용 방지를 위한 정보로 사용되며, NONCE 형태로 대체될 수 있다.
- 여기서 CS_{DH}는 $CS_{DH}=g^a \bmod p$와 같으며, a는 비밀정보이고 이후 SCS_{DH}정보로 표기된다.
- 상기정보($S_{PRICS}(CERT_{CS} \| TIME1 \| CS_{DH})$)를 VA에게 전송한다.

단계2

- VA는 전송된 정보의 전자서명을 검증한다($V_{CERTCS}(S_{PRICS}(CERT_{CS} \| TIME1 \| CS_{DH}))$).
- 전자서명 검증에 성공하면 응답 메시지를 생성한후, 그 결과($S_{PRIVA}(CERT_{VA},TIME2,VA_{DH})$)를 CS에게 전송한다. 여기서 TIME2는 TIME1과 같은 기능을 갖는다.

단계3

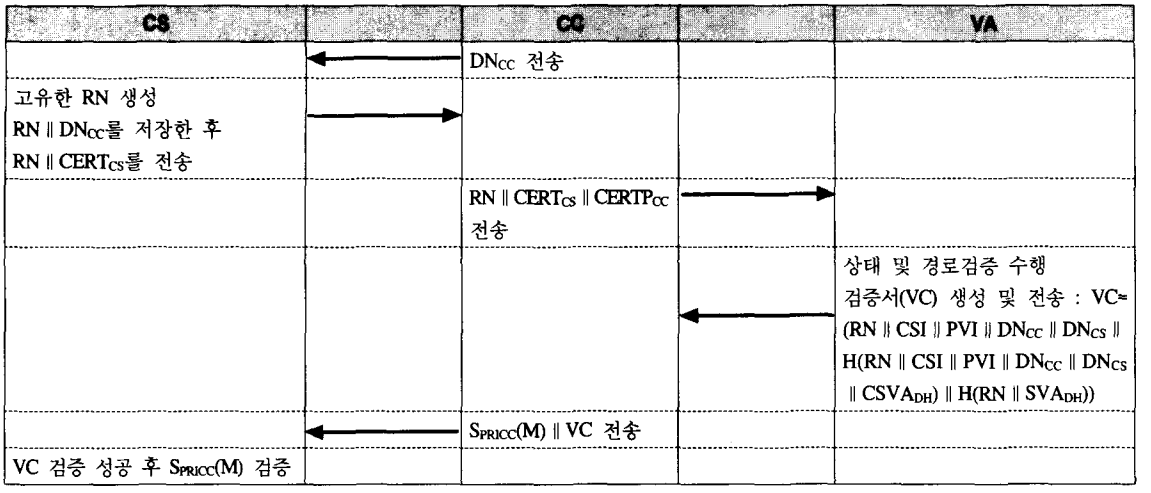
- VA는 DH 비밀공유키를 계산한 후 해당 결과($CSVA_{DH}=(CS_{DH})^{SVADH} \bmod p$)와 비밀정보 SVA_{DH}를 안전한 저장소에 저장 및 관리한다.
- CS는 VA의 메시지를 수신한 후 전자서명 검증을 수행한다($V_{CERTVA}(S_{PRIVA}(CERT_{VA},TIME2,VA_{DH}))$).
- 검증에 성공하면 DH 비밀공유키를 계산한 후 해당결과($CSVA_{DH}=(VA_{DH})^{SCSDH} \bmod p$)와 비밀정보 SCS_{DH}를 안전한 저장소에 저장 및 관리한다.

[검증서 발급 및 사용 프로토콜]

검증서 발급 및 사용 프로토콜은 CC가 CS에 전

CS		CC		VA
$S_{PRICS}(CERT_{CS} \| TIME1 \| CS_{DH})$ 생성 및 전송	→		→	
				수신 후 검증($V_{CERTCS}(S_{PRICS}(CERT_{CS} \| TIME1 \| CS_{DH}))$) $S_{PRIVA}(CERT_{VA} \| TIME2 \| VA_{DH})$생성
수신후 검증($V_{CERTVA}(S_{PRIVA}(CERT_{VA}, TIME2, VA_{DH}))$) $CSVA_{DH}=(VA_{DH})^{SCSDH} \bmod p$ 계산 $CSVA_{DH}$와 SCS_{DH}를 안전한 저장소에 저장 및 관리	←		←	$S_{PRIVA}(CERT_{VA} \| TIME2 \| VA_{DH})$전송 $CSVA_{DH}=(VA_{DH})^{SCSDH} \bmod p$ 계산 $CSVA_{DH}$와 SCS_{DH}를 안전한 저장소에 저장 및 관리

〔그림 1〕 초기 등록 프로토콜



〔그림 2〕 검증서 발급 프로토콜

자서명을 전송하기 전 또는 CS 서버에 인증서 기반의 시스템 로그온을 하기 전에 자신의 인증서상태정보와 경로검증정보에 대한 검증서를 VA로부터 발급받고 이를 사용하는 절차이다. 해당 검증서는 CC의 인증서상태정보와 인증경로검증정보를 포함하고 있으며 CS는 이 정보를 이용하여 네트워크 접근이나 부가적 암호연산 없이 해쉬함수 연산만으로서 인증서 검증을 수행할 수 있게 된다.

단계1

- CC는 CS의 웹 사이트에서 접속하여 인증서 기반 로그온 버튼을 클릭하거나, 또는 전자서명을 위하여 웹 페이지 폼(form) 필드에 중요 데이터를 입력한 후 전송 버튼을 클릭한다. 이때, CS는 CC의 클라이언트 측 실행 함수를 호출하며, 이와 동시에 CC에 기 설치된 소프트웨어 모듈이 동작하여 인증서내의 자신의 DN을 식별 정보로서 전송한다.

단계2

- CS는 DN 식별 정보 수신 후 서버 측 컴포넌트로 설치된 암호모듈을 사용하여 RN을 생성하고, 이를 자신의 인증서와 함께 CC에게 전달한다. 동시에 해당 RN과 CC로부터 수신한 DN_{CC}를 함께 데이터베이스에 저장한다. 여기서의 RN은 향후 CC나 제3자에 의한 재사용 공격(replay attack)을 방지하기 위한 도구로 사용된다. 그러므로, RN은 매 세션마다 다르게 생성되어야만 한다. 이때 RN은 약어기호에 정의한 바와 같이 X9.31이나 FIPS 186-1 등과 같은 안전성이 입증된 국제 표준을 따른다.

단계3

- CC는 CS로부터 RN ‖ $CERT_{CS}$를 수신한 후, 자신의 전체인증경로($CERTP_{CC}$)와 함께 전체 데이터(RN ‖ $CERT_{CS}$ ‖ $CERTP_{CC}$)를 VA로 전송한다.

단계4

- VA는 RN ‖ $CERT_{CS}$ ‖ $CERTP_{CC}$를 수신 후 $CERTP_{CC}$에 대한 인증서상태 및 경로검증을 수행한다. 이때 VA는 CA의 부속기관으로서 여러 가지 방법으로 인증서상태 확인을 수행할 수 있다. 이때의 동작절차는 기존 OCSP 서버나 SCVP 서버가 사용하는 방식을 준용하는 것으로 가정한다. 일반적으로 사용할 수 있는 방법론은 CA가 운영하는 디렉토리 서버의 이용 또는 CA의 데이터베이스와 동기화된 독립 데이터베이스의 이용 등이 있다.
- 검증 수행 후 확인된 결과는 CSI와 PVI의 정의 값으로 표현되며, VA는 검증서 VC를 구성한다 : (RN ‖ CSI ‖ PVI ‖ DN_{CC} ‖ DN_{CS} ‖ H(RN ‖ CSI ‖ PVI ‖ DN_{CC} ‖ DN_{CS} ‖ $CSVA_{DH}$) ‖ H(RN ‖ SVA_{DH}))
  - H(RN ‖ CSI ‖ PVI ‖ DN_{CC} ‖ DN_{CS} ‖ $CSVA_{DH}$) : RN ‖ CSI ‖ PVI ‖ DN_{CC} ‖ DN_{CS}에 대한 개체인증 및 메시지인증을 위한 해쉬 결과값으로서 해당 메시지가 VA에 의해 생성되었음을 증명할 수 있는 데이터이다. CS는 $CSVA_{DH}$값을 알고 있기 때문에 상기 해쉬값의 진위여부를 확인할 수 있으

며, 성공은 해당정보의 신뢰를 의미한다.

- $H(RN \| SVA_{DH})$: SVA_{DH}는 $H(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| CSVA_{DH})$에 대한 메시지근원지인증을 제공하는 중요정보가 된다. 즉, CS가 임의적으로 $H(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| CSVA_{DH})$을 생성한 후 해당 데이터가 VA에 의해 생성되었다고 주장하는 것을 방지하기 위한 부인방지 기능을 제공하는 것이다. 이때 SVA_{DH}는 굳이 신뢰기관에 위탁될 필요는 없다. 그 이유는 분쟁발생의 경우가 CS가 허위주장을 하는 경우이기 때문에 VA가 단지 판사에게 SVA_{DH} 정보만 공개하는 것으로서 해당 사실이 증명될 수 있기 때문이다.

단계5

- CC는 VA로부터 검증서 $VC=(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| H(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| CSVA_{DH}) \| H(RN \| SVA_{DH}))$를 수신한 후 해당 VC를 전자서명 메시지와 함께 CS에게 전송한다.
- 이때 CC는 PKCS#7 기반의 전자서명 메시지 표준 을 사용하며, CC의 인증경로($CERTP_{CC}$)는 해당 메시지 내에 포함된다.

단계6

- CS는 $(S_{PRICC}(M)) \| VC$를 수신한 후 VC를 이용하여 인증서상태 및 경로검증을 수행한다.
  - DN_{CC}의 정보와 $CERTP_{CC}$내 $CERT_{CC}$의 소유자이름이 동일한지 확인하며 같은 경우 다음 절차를 수행한다.
  - VC의 RN 정보와 기 저장된 $(RN \| DN_{CC})$의 저장정보를 비교하여 맞으면, $(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| H(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| CSVA_{DH}))$의 진위여부를 확인한다. 평문 전송된 $RN \| CSI \| PVI \| DN_{CC} \| DN_{CS}$와 CS가 저장·관리하고 있는 $CSVA_{DH}$를 이용하여 해쉬 결과값인 $H'=H(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| CSVA_{DH})$을 계산한 후, 이것이 전송된 해쉬값 $H''=H(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| CSVA_{DH})$과 같은 경우($H'=H''$) CS는 $RN \| CSI \| PVI \| DN_{CC} \| DN_{CS}$의 전송값을 신뢰할 수 있다. 즉, CS는 CSI 정보를 이용하여 인증서상태 정보를 확인하고, PVI를 이용하여 인증서경로검증 결과를 확인하게 된다.
- 상기 검사를 통과하면 CS는 $CERT_{CC}$ 내의 공개키를 이용하여 전자서명 검증을 수행한다.

V. 제안 프로토콜 분석

본 장에서는 두 가지 관점에서 제안 시스템을 분석하고자 한다. 1절에서는 제안 시스템의 설계 목표 요구조건과의 적합성을 보여주고, 2절에서는 기존 관련 연구와의 비교·분석을 통하여 제안 시스템의 우월성을 설명하고자 한다.

5.1 목표 요구조건 만족 여부 분석

제안 프로토콜은 초기 설계 목표인 목표 요구조건을 모두 만족하고 있으며, 그 상세 내용은 다음과 같다.

- 정의1 : 완전성
  - 조건1 : 만족. 제안 방법은 전자서명결과에 실시간 인증서상태정보와 경로검증 정보를 함께 전송하기 때문에 시간격차 문제를 갖지 않는다. VA는 CA의 부속기관으로서 항상 최신의 인증서상태 정보 결과를 알려준다.
  - 조건2 : 만족. 제안 방법은 전자서명 대신 DH 키분배를 이용한 공유비밀정보 방식을 사용하기 때문에 또 다른 인증서 검증이라는 재귀적 추가 가정을 갖지 않는다.

- 정의2 : 효율성
  - 요소1 : CS가 CC로부터 전자서명을 수신한 후 인증서상태검증이나 경로검증을 위해 수행하는 네트워크 접근 횟수는 '0'회 이다.
  - 요소2 : 환경구성(configuration) 파일의 존재를 고려하지 않는다면 접근 횟수 '0'회 이다. 즉, 인증서상태정보나 경로검증 정보로서 CRL 파일이나 기타 정보들을 사용하지 않기 때문에 파일 접근 횟수는 '0'회가 된다.
  - 요소3 : 본 제안 프로토콜에서 인증서상태검증 및 경로검증을 위해서 CS가 수행하는 암호연산의 횟수는 $H(RN \| CSI \| PVI \| DN_{CC} \| DN_{CS} \| CSVA_{DH})$의 진위여부를 확인하기 위한 해쉬연산 '1'회가 전부이다.

- 정의3 : 안전성
  - 조건1 : CSVA 메시지는 VA에 의해서만 생성될 수 있으며, CS는 해당 메시지의 위·변조 여부를 확인하기 위해서 사전에 CS와 VA간 공유된 공유비밀키($CSVA_{DH}$)를 사용한다. 그러므로, $CSVA_{DH}$ 값을 알지 못한다면 해당 데이터를 위·변조 할

수 없게 된다.

- 조건2 : CSVA는 오직 VA에 의해서만 생성될 수 있으며, 이는 $H(RN \| SVA_{DH})$ 값에 의해 증명될 수 있다. $H(RN \| SVA_{DH})$ 정보는 오직 VA에 의해서만 생성될 수 있으므로(SVA_{DH}는 오직 VA만이 알고 있는 비밀정보임), CSVA는 위장 및 행위부인방지인 조건2의 요구조건을 만족한다.
- 조건3 : CSVA는 RN의 사용으로 인하여 재사용 공격에 안전하다. 즉, CC가 CS로부터 초기값으로 수신한 RN이 최종 CSVA의 RN과 맞지 않는다면 CSVA 접수는 거부된다. 이는 재사용 공격의 가능성이 있기 때문이다.

5.2 기존 방식과의 비교 분석

본 절에서는 제안 프로토콜과 기존 방식들을 목표 요구 조건 관점에서 비교 분석하고자 한다. 이를 통하여 본 논문에서 제안 프로토콜의 가치를 객관적이고 정량적으로 분석하고자 한다. 비교 분석 전에 비교 대상이 되는 현존하는 기존 방식들의 내용을 간략히 살펴본다.

5.2.1 기존 방식 비교 분석

5.2.1.1 완전성 측면

기본적으로 CRL 기반의 인증서상태 검증 방법은 잠재적으로 모두 시간격차 문제를 가지게 된다. 단지 해당 시간격차를 줄임으로서 일반적 응용에서 사용할만한 허용 범위를 가지게 하는 것이 최선의 방안이 된다. 그러나, 해당 응용이 완전한 인증서상태 검증을 수행해야 한다면 CRL 기반 방법론을 대신하여 실시간상태확인기반의 방법론을 사용해야만 한다.

제안 프로토콜은 실시간 상태 확인 방법론과 마찬가지로 시간격차 문제를 가지지 않는다. 또한, 실시간상태확인기반 프로토콜은 일반적으로 추가 가정사항(OCSP나 SCVP 서버의 인증서상태 검증은 하지 않거나 단기주기(short-lived) 인증서를 사용하는 것으로 가정)을 가지게 됨으로써 또 다른 잠재적 오버헤드나 문제점을 가질 수 있게 되는 반면 제안 프로토콜은 그러한 추가적 가정사항을 갖지 않음으로써 보다 높은 완전성을 갖는다. 또한, OCSP나 SCVP의 경우 응답메시지를 전자서명 하는 구조를 취하기 때문에 해당 서버의 부하가 굉장히 높으며, 이는 서비스 장애 요인으로 작용한다.

5.2.2.2 효율성 측면

효율성 측면 분석의 첫 번째 요소는 네트워크 접근에 대한 효율성 평가이다. CRL 방법론은 기본적으로 네트워크 접근을 통한 CRL의 다운로드, 다운로드된 CRL의 전자서명 검증, CRL 파일의 접근 등의 작업을 수반하게 된다. 네트워크 접근은 해당 인증서상태검증 시 발생할 수도 있으며, 그렇지 않을 수도 있다. 즉, 인증서상태검증에 사용하는 CRL의 다음 갱신주기가 현재 시각보다 나중이라면 시스템은 새로운 CRL의 다운로드 없이 해당 인증서의 상태검증을 수행할 수 있다. 그러므로, CRL 기반에서 네트워크 접근은 '0'회 또는 '1'회가 된다. 실시간상태확인 기반에서는 직관적으로 최소 '1'회 이상이 된다. 기타 CRT의 경우는 CRL과 동일한 방식이므로 '0'회 또는 '1'회가 되며, NOVOMODO의 경우에는 상태검증 시 반드시 서버로부터 인덱스 값을 받아야만 하므로 최소 '1'회가 된다. 그러나, 제안 프로토콜은 별도의 네트워크 접근이 필요하지 않는다.

두 번째 요소는 파일 접근에 대한 효율성 평가이다. 파일접근은 시스템 성능에 영향을 미치는 중요 요소이므로 이에 대한 평가는 중요하다. 일반적으로 CRL 기반의 시스템은 최소 '1'회 이상의 파일 접근을 해야 하며, 실시간상태검증의 경우에는 별도의 파일 접근 없이 네트워크 프로토콜 차원에서 주로 검증 작업을 수행하게 할 것이다. 그러므로 파일 접근은 '0'회가 된다. 제안 프로토콜은 실시간상태검증 프로토콜과 같이 파일 접근이 필요 없다. 그러므로, 파일 접근 측면에서의 효율성은 실시간상태검증 프로토콜의 효율성과 동등하다.

마지막으로 효율성 측면의 세 번째 요소는 시스템이 수행해야 할 암호연산의 횟수이다. 상기 표에서 "요소3"은 각 방법론 별 암호연산의 횟수를 나타낸다. V(n) 연산은 전자서명 검증 연산으로서 공개키 연산 n번 수행을 의미한다. 상기 표에서 인증경로검증은 레벨2로 간주한다. 즉, 사용자 인증서의 경로검증을 위해서 상위 인증기관 및 최상위 인증기관 인증서 검증을 수행(V(2))하는 것으로 가정한다. [표 2]의 CRL 방법론 부분에서 첫 부분 V(2)는 CRL 서명검증 '1'회(V(1)), 클라이언트 인증서 서명검증 '1'회(V(1))를 의미하는 것이고 나머지 부분의 V(2)는 상위 인증기관 ARL(Authority Revocation List, 이하 ARL이라 칭함) 서명검증 '1'회, 상위 인증기관 인증서 서명검증 '1'회(V(1))를 의미한다. 여기서 최상위 자가서명 인증서의 검증은 비교연산으로 가정하며 상기 표에

〔표 2〕 인증서상태 및 경로검증 프로토콜 비교 분석

방법론	완전성		효율성			안전성			비고
	조건1	조건2	요소1	요소2	요소3	조건1	조건2	조건3	
CRL 기반									
CRL DP	X	○	0/1	1	V(2)+V(2)	○	○	○	
Delta CRL	X	○	0/1	N	N*V(2)+V(2)	○	○	○	
Over-issued CRL	X	○	0/1	1	V(2)+V(2)	○	○	X	
Indirect CRL	X	○	0/1	1	V(2)+V(2)	○	○	○	
Dynamic CRL DP	X	○	0/1	1	V(2)+V(2)	○	○	○	
Freshest CRL	X	○	0/1	N	N*V(2)+V(2)	○	○	○	
실시간상태확인기반									
OCSP	○	X	1	0	V(1)+V(2)	○	○	○	
SCVP	○	X	1	0	V(1)	○	○	○	
비표준신개념기반									
CRT	X	○	0/1	1	(V(2)+H(h))×2	○	○	○	
NOVOMODO	X	○	1	0	H(a)×2	○	○	○	
제안 프로토콜									
ECSPVS	○	○	0	0	H(1)	○	○	○	

※ N : 기본 CRL로부터 최신의 델타 CRL까지의 연결되는 CRL 수, h : Merkle Hash Tree의 높이, V(n) : 공개키암호연산(전자서명검증) n회, H(n) : 해쉬연산 n회, a : 유효성 확인을 위하여 반복해야 할 해쉬함수 연산의 수

〔표 3〕 프로토콜별 서버 측 처리 예상 속도 비교(암호연산)

방법론	예상수행속도 (msec/cert)	비고
CRL 기반		
CRL DP	1.28	
Delta CRL	1.92	N=2로 가정
Over-Issued CRL	1.28	
Indirect CRL	1.28	
Dynamic CRL DP	1.28	
Freshest CRL	1.92	N=2로 가정
실시간상태확인기반		
OCSP	0.96	
SCVP	0.32	
비표준신개념기반		
CRT	1.3	h=5로 가정
NOVOMODO	0.6	a=150로 가정
제안 프로토콜		
ECSPVS	0.002	

서는 고려하지 않는다. 이하 나머지 부분들도 각각 동일한 방식의 연산 횟수를 갖는다. 제안 프로토콜은 지금까지 제안된 방식 중 가장 효율적인 해쉬연산 '1'회만의 암호연산을 수행한다. 일반적으로 해쉬함수는 전자서명 생성 연산보다 약10,000배 정도 빠르다고 알려져 있으며,[3] 상기 결과를 가지고 아래와 같은 조건 하에서 상대적 성능 개선도를 비교할 경우 [표 3]과 같은 결과를 얻을 수 있다. 아래의 전자서명 및 해쉬함수 연산 속도는 실제 암호 라이브러리의 테스트 결과를 기반으로 한 것이다.[10]

- 전자서명검증(1024비트) 연산속도 : 0.32msec
- 해쉬함수(128비트 SHA-1) 연산속도 : 0.002msec

2.2.3 안전성 측면

안전성 측면에서 CRL 방법론과 실시간상태확인 방법론은 모두 전자서명을 사용하고 있기 때문에 평가요소에 정의한 요구조건을 만족한다. CRT의 경우에는 전자서명과 해쉬함수를 통하여 해당 조건을 만족하고 있으며, NOVOMODO의 경우에는 본 논문에서 제안하는 방법과 동일한 방법을 사용하고 있기 때문에 안전성 부문을 모두 만족한다. 다만, 추가발행 CRL의 경우 동일한 시점에 여러 개의 유효한 CRL이 존재할 수 있기 때문에 재사용 위협에 직면할 수 있다. 즉, 어떤 인증서가 폐지된 경우 신뢰당사자가 해당 폐지 사실을 포함하고 있는 CRL을 이용하지 못하고 기존의 유효한 CRL을 계속적으로 사용하게

되는 경우 잘못된 인증서 검증을 수행하게 된다.

Ⅵ. 결 론

PKI 시스템은 안전한 정보인프라 구축에 필수적인 기반 시스템이지만 그것의 폭 넓은 사용에는 아직도 많은 현실적인 제약이 남아 있다. 그러한 어려움들 중의 하나는 인터넷뱅킹이나 인터넷주식트레이딩과 같이 실시간 기반의 대규모 클라이언트-서버 환경에서 피크 타임(peak time)에 수 많은 접속자들이 동시에 접속하여 인증서기반의 로그온이나 전자서명 메시지를 전송하는 경우 신뢰당사자가 되는 서버는 과도한 인증서상태 및 경로검증 때문에 시스템 정지 현상이 발생할 수도 있다는 것이다. 이 때문에 현재 대규모 클라이언트-서버 환경에서는 다른 임시적 수단을 통한 인증서상태검증 또는 경로검증 방법을 사용하고 있거나, 인증서 상태 검증을 배제한 불완전한 인증서 검증을 수행하고 있는 실정이다. 더욱이, 최근 표준화된 OCSP나 SCVP와 같은 프로토콜의 경우에도 해당 문제를 명쾌하게 해결하지 못하고 있기 때문에 이에 대한 문제 해결이 필요한 상태이다.

본 논문에서는 실시간 기반의 대규모 클라이언트-서버 환경에서 신뢰당사자가 되는 서버의 자원 소모를 극단적으로 줄여 서버 장애를 방지할 수 있는 효율적인 인증서상태 및 경로검증 시스템을 제안하였다. 제안 시스템의 기본적 개념은 클라이언트가 전자서명 시 인증서상태 및 경로검증 정보를 CA의 부속기관인 VA로부터 검증서 형태로 발급받아 이것을 전자서명과 함께 서버로 전송하게 하는 것이다. 이때 서버에서는 부가적인 네트워크 접근이나 파일 접근 없이 단순 해쉬연산 만으로서 해당 인증서상태 및 경로검증을 수행할 수 있게 된다. 제안 시스템은 본 논문에서 정의한 완전성, 효율성, 안전성 측면에서 기존에 제안되거나 표준화된 프로토콜보다 우월하며 실제 환경에서 효율적으로 구축 사용될 수 있는 시스템이다. VA는 CA의 부속기관으로서 별도의 개인키를 가질 필요가 없는 신뢰기관이 된다. 결론적으로 제안 시스템은 인터넷뱅킹이나 인터넷주식트레이딩과 같은 실시간 기반의 대규모 클라이언트-서버 기반에서 현실적으로 사용 가능한 효과적인 시스템이 될 것이다.

참 고 문 헌

[1] R. Housley, W. Ford, W. Polk, D. Solo., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", IETF RFC3280, April 2002.

[2] Paul Kocher, "A Quick Introduction to Certificate Revocation Tree(CRTs)", Technical Report, Valicert, 1999.

[3] Silvio Micali, "NOVOMODO, Scalable Certificate Validation And Simplified PKI Management", 1st Annual PKI Research Workshop, Preproceedings, pp. 9~19, April 2002

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

[5] A. Malpani, R. Housley, T. Freeman, "Simple Certificate Validation Protocol(SCVP)", IETF Internet Draft, June 2002.

[6] RSA Inc. PKCS #7: Cryptographic Message Syntax Standard. Version 1.5, November 1993.

[7] W. Diffie and M. E. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, 22, 1976

[8] David. A. Cooper, "A model of certificate revocation", Proceeding of the 15th Annual Computer Security Applications Conference, December 1999.

[9] ITU and ISO/IEC, "Final Proposal Draft Amendment on Certificate Extensions", April 1999.

[10] http://www.eskimo.com/~weidai/benchmarks.html.

[11] ANSI X9.31, "1998 Digital signatures using reversible public cryptography for the financial services industry (rDSA)", 1998.

[12] NIST FIPS(Federal Information Processing Standards Publication) 186-1, "Digital Signature Standard", December, 1998.

〈著者紹介〉

최 영 철 (Young-Chul Choi) 정회원
1996년 2월 : 성균관대학교 정보공학과 졸업(학사)
1998년 2월 : 성균관대학교 대학원 정보공학과 졸업(석사)
2003년 2월 : 성균관대학교 대학원 전기전자 및 컴퓨터공학부 졸업(박사)
1997년 10월~2000년 3월 : 한국정보보호센터(KISA) 연구원
2000년 5월~현재 : (주) 비씨큐어 해외사업부장
<관심분야> 암호 프로토콜, PKI, DRM, 전자지불시스템

박 상 준 (Sang-Joon Park) 정회원
1984년 2월 : 한양대학교 수학과 졸업(학사)
1986년 2월 : 한양대학교 대학원 수학과 졸업(석사)
1999년 2월 : 성균관대학교 대학원 정보통신공학과(박사)
1986년 2월~1999년 12월 : 한국전자통신연구원 선임연구원
2000년 1월~2000년 10월 : 한국전자통신연구원 부설 국가보안기술연구소 책임연구원
2000년 11월~현재 : (주) 비씨큐어 부사장
<관심분야> 암호 알고리즘 설계 및 분석, 암호키관리, PKI, DRM

원 동 호 (Dong-Ho Won) 종신회원
성균관대학교 전자공학과 졸업(학사, 석사, 박사)
1978년~1980년 : 한국전자통신연구소 전임연구원
1992년~1994년 : 성균관대학교 교학처장
1996년~1998년 : 국무총리실 정보화추진위원회 자문위원
1999년~2001년 : 성균관대학교 전기전자 및 컴퓨터공학부장 정보통신대학원장
현재 : 성균관대학교 정보통신공학부 교수, 성균관대학교 연구지원처장, 정보통신부 지정 정보보호인증기술연구센터 센터장
<관심분야> 암호이론