효율적인 공개키 프레임워크에 대한 실용적 개선과 응용

정회원 양 종 필*, 신 원**, 이 경 현***

Practical Improvement of An Efficient Public-Key Framework and Its Application

Jong-Phil Yang*, Weon Shin**, Kyung Hyune Rhee*** *Regular Members*

요 약

J. Zhou 등이 제안한 공개키 프레임워크는 인증서의 유효 기간은 여러 짧은 갱신 기간으로 나뉘어 지며, 각 인증서는 인증서 소유자(또는 조직적인 환경에서는 소유자의 관리자)의 제어를 통하여 각 갱신 기간의 종료 지점에서 취소될 수 있다. J. Zhou 등의 공개키 프레임워크는 인증서 검증자의 계산 및 통신 부담을 줄임으로써 효율성을 상당히 증진시키지만 실제 환경에 구현하여 적용할 경우 많은 문제점을 내포하고 있다. 따라서, 본 논문에서는 J. Zhou 등의 공개키 프레임워크 내의 보안 파라메타들을 재조명한 후, 실제 구현 환경에 적합하도록 개선하며, J. Zhou 등의 공개키 프레임워크에서 불필요한 신뢰기관을 제거함으로써 보다 실용적인 공개키 프레임워크를 제안한다. 그리고, 개선된 공개키 프레임워크를 기반한 실질적인 응용을 소개한다.

Key Words : Public Key Infrastructure, Certificate Revocation, Authenticatoin, Security Protocol

ABSTRACT

J. Zhou et al. proposed a new public-key framework, in which the maximum lifetime of a certificate is divided into short periods and the certificate could be expired at the end of any period under the control of the certificate owner(or his manager in a corporate environment). However, J. Zhou et al.'s public-key framework is not suitable on implementation in real world. Therefore, we review some security parameters to change them into more suitable ones for implementation and remove an unnecessary trust party of J. Zhou et al.'s public-key framework. Then, we propose an improved scheme for realistic solution. Moreover, we present a practical application based on the improved framework.

Ⅰ. 서 론

오늘날 공개키 기반구조(Public Key Infrastructure, PKI)는 전자서명을 통한 인증, 부인방지 서비스의 제공과 함께 대칭키와 비대칭키의 결합을 통한 기밀성 보장 및 키 관리 서비스의 사용을 위한 기반구조로써 널리 이용되어지고 있다[1]. 공개키 기반구조에서 인증서는 공개키와 소유자의 신원정보를 결합한 것으로 인증기관(Certificate Authority, CA)이라 불리는 신뢰된 제 3자에 의하여 발급된다. X.509 인증서는 IETF PKIX 워킹그룹에 의하여 제정된 인증서 표준 형식으로써 X.509 인증서에 기반한 PKI는 인터넷 표준으로 자리잡고 있다[11]. 한편, CA에 의해 발급된 사용자 인증서는 만기일 이전에 취소되어질 수 있을 것이며 이러한

※ 이 논문은 2003학년도 부경대학교 기성회 학술연구비에 의하여 연구되었음.
* 큐슈대학교 시스템정보과학부(bogus@itslab.csce.kyushu-u.ac.jp)
** 동명정보대학교 정보보호학과(shinweon@tit.ac.kr) *** 부경대학교 전자컴퓨터정보통신공학부(khrhee@pknu.ac.kr), 교신저자
논문번호 : KICS2005-08-357, 접수일자 : 2006년 1월 8일, 최종논문접수일자 : 2006년 3월 15일

사용자 인증서에 대한 취소 정보의 효율적이고 적시적인 분배는 PKI에서 매우 중요한 부분이다. 그러나, 현재 존재하는 인증서 취소 기법들은 PKI 사용자뿐만 아니라 인증기관에게 상당한 작업처리와 통신상, 저장공간의 오버헤드를 요구한다: CRL[11], OCSP[5], delta CRL[2], indirect CRL[1], Certificate Revocation Tree[1][8], Certificate Revocation System[13]. 따라서, 인증서 취소 정보를 효율적이고 적시적으로 분배할 수 있는 메카니즘의 개발은 더욱 효율적인 PKI 환경을 구현하기 위하여 해결되어져야 하는 문제로 거론되어지고 있다.

J. Zhou 등은 인증서의 유효 기간을 짧은 기간들로 나누고, 인증서가 인증서 소유자(혹은 조직적인 환경에서 관리자)의 통제아래 각 기간의 끝 지점에서 취소될 수 있는 새로운 공개키 프레임워크를 제안하였다[3]. 그러나, J. Zhou 등의 공개키 프레임워크에서 CA는 인증서의 취소를 직접적으로 제어할 수 없기 때문에 악의적인 사용자는 비록 인증서가 취소되었지만 성공적으로 다른 사용자와 인증업무를 수행할 수 있다. 비록 J. Zhou등은 보안 서버(Security Server, SS)라는 새로운 신뢰 기관을 통한 대안을 제시하였지만, 보안 서버는 단지 소개된 취약점을 극복하기 위한 불필요한 신뢰 기관일 뿐이다. 또한, 인증서의 유효 기간을 짧은 기간들로 나누는 것은 인터넷 환경에서 타임 동기화 문제를 발생시킬 수 있다.

본 논문에서는 J. Zhou 등의 공개키 프레임워크 내의 보안 파라메타들을 재조명한 후, 실제 구현 환경에 적합하도록 개선하며, J. Zhou등의 공개키 프레임워크에서 불필요한 신뢰기관을 제거함으로써 보다 실용적인 공개키 프레임워크를 제안한다. 그리고, 개선된 공개키 프레임워크의 응용으로써, J. Yang 등이 제안한 유·무선 장치간의 인증된 채널 설정 프로토콜을 개선한다[4]. 본 논문은 다음과 같이 구성된다. 2장에서 J. Zhou 등의 공개키 프레임워크에 대하여 소개한 후, 3장에서는 J. Zhou 등의 공개키 프레임워크 내의 보안 파라메타들을 재조명하고, 실질적 구현 환경에 적합한 개선된 공개키 프레임워크를 제안한다. 4장에서 취약성 원도우와 통신비용에 대한 성능 분석을 수행하며, 5장에서 제안된 공개키 프레임워크를 기반한 새로운 인증된 채널 설정 프로토콜을 제안한다. 최종적으로, 6장에서 결론을 맺는다.

Ⅱ. J. Zhou 등의 공개키 프레임워크

J. Zhou 등은 인증서 취소에 대한 효율적인 방안을 제시하고자 두 가지 모델을 제안하였다. 기본 프레임워크(Basic Framework)라는 해쉬체인에 기반한 새로운 인증서 취소 상태 검증 기법을 제안하였으며, 기본 프레임워크의 문제점을 해결하기 위하여 관리자 제어된 인증서(Manager Controlled Certificate)를 제안하였다[3].

2.1 J. Zhou 등의 기본 프레임워크(J. Zhou et al.'s Basic Framework, ZBF)

본 절에서는 J. Zhou 등이 제안한 기본 프레임워크(ZBF)에 대한 소개를 한다. ZBF에서의 공개키 인증서의 취소 유무에 대한 상태 관리는 오직 인증서 소유자에 의해서만 제어된다. $SIGN_A(M)$은 메시지 M에 대한 통신 개체 A의 전자 서명문을 나타낸다. 사용자 U의 공개키 인증서는 아래와 같은 절차로 생성된다.

(Step 1) U는 자신을 위한 키쌍을 생성한다 : SK_U는 비밀키, PK_U는 공개키.

(Step 2) U는 다음과 같은 인증서 파라메터를 정의한다. T는 인증서의 유효 기간, D는 유효 기간의 시작 지점, L은 인증서 유효성 갱신 기간. 또한, $j = T/L$을 정수 값이라고 가정하면, 인증서 유효성 갱신 지점들은 아래와 같이 정의될 수 있다.
$D_1 = D + L, D_2 = D + 2*L, \cdots, D_j = D + j*L$

(Step 3) U는 일방향 해쉬체인을 생성한다. 이때, r은 오직 U만이 알고 있어야한다.
$H^0(r) = r,\ H^i(r) = H(H^{i-1}(r)),\ i = 1,2,\cdots,j$

(Step 4) U는 $(PK_U, D, H^j(r), j, L)$을 CA에게 전송한다.

(Step 5) CA는 U의 요청을 인증한 후 아래와 같이 U의 인증서를 생성하여, U에게 전달한다.
$CERT_U = SIGN_{CA}(U, PK_U, D, H^j(r), j, L)$
여기서, $SIGN_X$는 통신 개체 X의 전자 서명을 의미한다. 기존의 공개키 인증서와 비교할 때, $CERT_U$는 $(H^j(r), j, L)$와 같은 추가 파라메타를 포함한다. 추가된 파라메타는 ZBF에서 $CERT_U$의 유효성을 제어하는데 사용되어 진다.

$CERT_U$의 다음 유효성 갱신 지점이 D_e라고 가정하면, U는 SK_U를 사용하여 전자 서명문을 생성할

시 $i=j-(D_e-D)/L$인 $(H^i(r),i)$를 전자 서명문에 첨부시킨다. U의 전자서명 검증을 위해 검증자 V는 먼저 $CERT_U$의 취소 유무를 확인해야 한다. $CERT_U$를 검증하는 지점이 D_V라고 가정하면, V는 아래의 검증과정을 수행하여 $CERT_U$의 상태를 확인한다.

(Step 1) V는 $CERT_U$ 상의 CA의 서명을 검증한다. 만약 올바르다면, V는 U의 공개키는 PK_U이고, 시작 유효일은 D, 유효 기간은 $T=j*L$, 갱신 기간는 L, 해쉬체인상의 마지막 해쉬값은 $H^j(r)$임을 확인한다.

(Step 2) V는 $0\le i<j$ 와 $H^{j-i}(H^i(r))=H^j(r)$을 검증한다. 만약 올바르다면, V는 $H^i(r)$은 $H^j(r)$로 끝나는 일방향 해쉬체인상의 올바른 해쉬값임을 믿게 된다.

(Step 3) V는 $D_V\le D+(j-i)*L$을 검증한다. 만약 올바르다면, V는 $CERT_U$가 현재 유효하고 $D_e=D+(j-i)*L$까지 유효할 것이라고 결론짓는다.

위의 방법에서, U는 전자 서명문 생성시에 관련된 $H^i(r)$을 배포하는 것에 의해 $CERT_U$의 유효성을 제어할 수 있다. 또한, V는 CA로부터 취소 정보를 획득하는 절차없이 $CERT_U$의 취소 유무 상태를 확인할 수 있으므로, CA는 인증서의 유효성을 검증하는 절차로부터 제외된다. 또한, 인증서 소유자 U는 $CERT_U$의 만기일을 제어하기 위하여 해쉬체인의 루트 값인 r값을 유일하게 알고 있어야 하므로, U의 장치는 r과 SK_U를 안전하게 보관해야만 한다.

2.2 J. Zhou 등의 관리자 제어된 인증서(J. Zhou et al.'s Manager Controlled Certificate, ZMCC)

ZBF에서는 인증서 소유자 U가 인증서 유효 기간 T까지 $CERT_U$의 유효성을 제어한다. 이는 비밀키 훼손 등으로 인하여 발생하는 인증서 소유자에 의한 인증서 취소의 필요성에 관해서만 고려하고 있다. 하지만, 인증서는 회사 퇴직이나 직책의 변동 등 여러 가지 다른 이유로 인하여 인증서 소유를 반대하는 관리자에 의하여 취소되어질 필요가 있다. J. Zhou 등은 위와 같은 상황을 해결하기 위한 방안으로 사용자들을 대신하여 해쉬체인의 루트를 생성하는 보안서버(Security Server, SS)를 사용하는 ZMCC을 제안하였다.

ZMCC에서는 $CERT_U$의 유효성 갱신 지점이 다가오면, SS는 사용자 U에게 현재 요구되는 해쉬값을 분배한다. U는 해쉬체인을 계산하는 것에 의해 전송받은 해쉬값의 유효성을 쉽게 검증할 수 있다. 만약, SS가 어떠한 이유로 인하여 U의 인증서를 취소하기를 원한다면 U의 해쉬값의 배포를 중지함으로써 $CERT_U$를 취소시킨다. 따라서, $CERT_U$는 다음 유효성 갱신 지점에서 취소될 것이다.

Ⅲ. J. Zhou 등의 공개키 프레임워크에 대한 실용적인 사용 방안

3.1 실제 구현 환경에서의 고려사항

공격자에 의해서 사용자 U의 SK_U와 r이 노출되거나 사용자가 악의적일 경우, ZBF의 취약성 윈도우(window of vulnerability)[1]는 인증서의 예정된 만기일까지 지속되는 심각한 문제점을 가지고 있다. 따라서, J. Zhou 등은 SS라는 신뢰기관에게 인증서 취소에 대한 처리를 위임하는 ZMCC를 제안하였다. 하지만, SS는 단지 앞서 언급된 취약점을 극복하기 위해 추가된 부가적 신뢰기관일 뿐이며, 실질적 시스템 구현 시에 SS를 안전하게 유지하기 위한 추가적인 비용이 요구될 것이다. 또한, SS는 소규모 조직 내의 인증서 상태 관리를 위해서 적용되므로, 인터넷과 같은 글로벌 환경에서 SS를 사용하는 것은 바람직하지 못하다. 본 절에서는 J. Zhou 등의 공개키 프레임워크를 실질적 구현을 위한 개선 방안을 소개한다.

〔고려사항 1〕 구현 환경에서 L 파라메타의 적용

ZBF에서 인증서의 취소 유무 상태를 확인하기 위해서는 D, L 그리고 T와 같은 다수의 시간 파라메타들이 존재한다. 특히 L 파라메타가 아주 짧게 선택되어지면 사용자와 검증자가 소유한 각각의 로컬 시스템내의 시스템 클럭의 차이로 인하여 서명 검증이 실패할 확률이 매우 높아진다. 따라서, 시간 동기화 문제를 제거하기 위하여 서버 지원된 서명(Server-supported signatures)[6]이나 S/Key 시스템[7]과 같이 해쉬체인의 반복수 감소는 시간 파라메터에 독립적이어야 한다. 즉, 사용자가 유효한 해쉬값을 생성할 필요가 있을 때마다 해쉬체인의 반복횟수가 줄어드는 방식을 사용해야 한다.

1) 취약성 윈도우(Window of vulnerability)는 검증자가 상대방의 인증서에 대한 취소 사실을 알지 못하고 지속적으로 유효하다고 판단하는 최대 시간을 의미한다[10].

〔고려사항 2〕 추가적인 보안 파라메타 r의 관리

ZBF에서 사용자 U는 SK_U와 r 모두 안전하게 보관해야 하는 반면, ZMCC에서는 사용자는 SK_U만을 안전하게 보관하며, SS는 조직내의 멤버들에 대한 다수의 r들을 안전하게 보관해야 하는 부담을 가진다. 실질적 시스템에서 U는 SK_U를 보호하기 위하여 사용자-정의 패스워드를 사용한다. ZBF에서 U는 패스워드로써 r을 기억해야 하거나, 패스워드로 r 값을 보호해야 한다. 따라서, U는 SK_U의 보호를 위한 패스워드와 r의 보호를 위한 패스워드 모두를 기억해야 한다. SK_U의 보호를 위한 패스워드는 SK_U와 직접적인 관련이 없기 때문에, SK_U의 보호와 r의 보호를 위하여 동일한 패스워드로 사용하는 것은 두 개의 패스워드를 사용하는 것 보다 더욱더 사용자 친화적인 방법이 될 수 있다.

〔고려사항 3〕 보안서버(SS)의 제거

ZMCC에서 SS는 조직 내의 사용자들을 대신하여 r값을 소유함으로써, 사용자들에 대한 인증서 취소를 수행한다. SS는 글로벌 영역을 위한 PKI에서 정의되지 않는 추가적인 신뢰기관이며, 특수한 소규모 조직을 위해서만 안전하게 인증서 취소 업무를 수행할 수 있다. 특히, SS를 통한 ZMCC는 ZBF에서의 취약성 윈도우의 지속성에 대한 문제를 해결하기 위한 방안으로 간주될 수 있다. 따라서, ZMCC에서 SS를 제거함과 동시에 SS의 역할을 대신할 새로운 해결책을 제시하는 것이 바람직하다. 본 논문에서는 SS를 제거하기 위하여 사용자가 제한된 시간 구간(time period)에서 자신의 인증서를 제어할 수 있는 해결책을 제안한다.

(Definition. 1) 제어 윈도우(Control Window)란 검증자가 오직 인증서 송신자의 해쉬체인 검증을 통해서만 송신자의 인증서 취소 유무 상태를 신뢰할 수 있도록 하는 허용된 시간 구간을 의미한다.

송신자의 인증서를 수신한 검증자는 CA에게 인증서 취소 정보를 질의한다. 만약 그 인증서가 취소되지 않았으면, 검증자는 현재의 로컬 시간을 "유효성 시작 지점"으로 설정하고 유효성 시작 지점에서 제어 윈도우만큼 증가시킨 시간을 "유효성 종료 지점"으로 설정한다. 그리고, 검증자는 종료 지점까지 송신자의 인증서를 캐쉬한다. 따라서, 검증자는 "유효성 종료 지점"까지 송신자의 인증서 상태를 송신자가 전송한 해쉬값의 검증을 통해서 판단한다. 즉, 인증서 송신자는 제어 윈도우동안은 단지 해쉬값을 계산하는 것으로 자신의 인증서 상태를 제어할 수 있음을 의미하며, 검증자는 제어 윈도우동안 송신자의 인증서 상태에 대한 제어능력(해쉬값)을 신뢰함으로써 CA로부터 인증서 취소 정보를 더 이상 질의할 필요가 없다. 또한, 제어 윈도우를 사용함으로써 J. Zhou 등이 제안한 공개키 프레임워크에서 SS를 제거할 수 있다.

3.2 개선된 공개키 프레임워크

앞 절에서는 J. Zhou 등의 공개키 프레임워크를 실질적 구현에 적합하게 하기 위하여 3가지 사항을 고려하였다. 본 절에서는 소개되었던 고려사항을 개선한 "인증서 생성" 및 "인증서 검증"을 위한 새로운 공개키 프레임워크를 제안한다.

〔인증서 생성〕

(Step 1) 사용자 U는 자신을 위한 키쌍을 생성한다 : SK_U와 PK_U.

(Step 2) U는 사용자-정의 패스워드 r을 생성한다. r은 SK_U를 암호화와 일방향 해쉬체인을 생성하기 위해서 사용된다.

$H^0(r)=r,\ H^i(r)=H(H^{i-1}(r)),\ i=1,2,\cdots,j.$

(Step 3) U는 $(PK_U, H^j(r), j)$을 CA에게 전송한다.

(Step 4) CA는 U의 요청을 인증한 후 아래와 같이 U의 인증서를 생성하여, U에게 전달한다.

$CERT_U = SIGN_{CA}(U, PK_U, H^j(r), j, CW)$

여기서, CW는 CA의 보안 정책에 의해 선택된 제어 윈도우(Control Window)를 나타낸다.

〔인증서 검증〕

(Step 1) 사용자 U는 검증자 V에게 $CERT_U$를 전송한다.

(Step 2) V는 $CERT_U = SIGN_{CA}(U, PK_U, H^j(r), j, CW)$ 내의 CA의 전자 서명을 검증한다. 만약 검증이 성공하면, V는 CA에게 취소 정보(예, CRL)를 질의하여 $CERT_U$의 취소 유무를 판단한다. 만약 $CERT_U$가 취소되지 않았다면, V는 U의 공개키가 PK_U이며, 일방향 해쉬체인의 마지막 해쉬값이 $H^j(r)$임을 알게 된다. 그리고,

- V는 현재 자신의 로컬 시간을 $CERT_U$에 대한 "유효성 시작 지점" 설정하고, 시작 지점에서 $CERT_U$내의 CW만큼 증가시킨 시간을 "유효성 종료 지점"로 설정한다.

– V는 "유효성 종료 지점"까지 $CERT_U$를 캐쉬한다.

(Step 3) U가 V에게 전자 서명문을 전송할 때, U는 패스워드 r을 입력하여 SK_U를 복호화하고 현재 반복수에 해당하는 해쉬값을 계산한다. 실질적으로 U의 장치는 해쉬체인상의 다음 반복수를 저장하고 있어야 한다. U는 복호된 SK_U로서 계산된 전자 서명문과 현재의 해쉬값 $(H^i(r), i)$를 V에게 전송한다.

(Step 4) V는 U로부터 전자 서명문을 받은 시간이 "유효성 종료 지점"을 지났는지를 확인한다. 만약 시간이 지나지 않았다면, V는 $H^{j-i}(H^i(r)) = H^j(r)$을 만족하는지 검사한다. 만족할 경우 V는 CA에게 인증서 취소 유무 정보에 대한 질의를 수행하지 않고, $CERT_U$가 현재 유효한 것으로 판단한다. 따라서, V는 PK_U로서 수신한 U의 전자 서명문을 검증한다.

Ⅳ. 개선된 공개키 프레임워크 분석

4.1 취약성 윈도우

본 절에서는 아래와 같이 3가지 유형에서의 취약성 윈도우(window of vulnerability)를 분석한다.

– CASE 1 : ZBF에서의 악의적 사용자
– CASE 2 : ZMCC에서의 정직한 사용자 또는 악의적 사용자
– CASE 3 : 제안된 공개키 프레임워크에서의 악의적 사용자

그림 1에서 사용자 U의 인증서 유효 기간(T)은 t1에서 t6까지라고 가정한다. U는 $CERT_U$를 t2에 검증자 V에게 전송하며, $CERT_U$가 t3에 취소된다고 가정한다. V는 수신한 $CERT_U$의 취소 유무 상태에 대한 검증 및 캐쉬를 다음의 3가지 유형에서 사용되는 각각의 기법에 따라서 수행할 것이다.

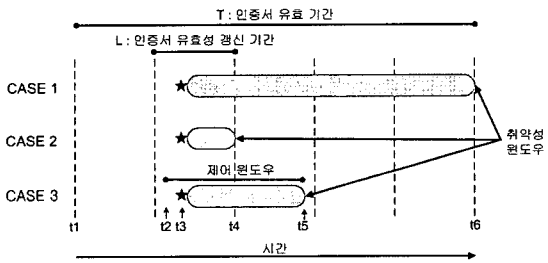

그림 1. 취약성 윈도우(Windows of vulnerabilities)

〔CASE 1〕 ZBF에서의 악의적 사용자

V는 정직한 사용자 U로부터 수신한 해쉬값을 검증하며, $t2 \le t4$인가를 확인한다. 여기서, V는 t4를 $D+(j-i)*L$를 통해서 계산 가능하다. 만약 모든 검사가 성공적이면, V는 t4까지 $CERT_U$를 캐쉬한다. 즉, $CERT_U$가 t4까지 취소되지 않을 것으로 간주한다. 따라서, V는 t4까지는 U가 전송한 해쉬값을 통해서 $CERT_U$의 취소 유무에 대한 상태 검증을 수행한다. 여기서, 악의적인 사용자를 고려해 보자. 비록 $CERT_U$는 t3에서 취소되었지만 악의적 사용자 U는 언제나 자신 혼자서 다음 유효성 갱신 기간을 위한 해쉬값을 생성 가능하며, V는 단지 U로부터 수신한 해쉬값에 의존하여 $CERT_U$의 상태 검증을 수행한다. 결국, 취약성 윈도우는 t3에서 t6까지 된다.

〔CASE 2〕 ZMCC에서의 정직한 사용자 또는 악의적 사용자

V는 CASE 1과 동일한 방식으로 $CERT_U$의 상태와 시간을 검증한다. 만약 $CERT_U$가 올바르다면, V는 $CERT_U$를 캐쉬하고, t4까지 취소되지 않을 것이라고 간주한다. 비록 U가 악의적 사용자일지라도 단독으로 다음 인증서 유효성 갱신 기간을 위한 해쉬값을 생성할 수 없다. 즉, SS가 취소된 $CERT_U$를 위한 해쉬값을 U에게 분배하지 않기 때문에, 취약성 윈도우는 t3에서 t4까지 된다.

〔CASE 3〕 제안된 공개키 프레임워크에서의 악의적 사용자

V는 Ⅲ장에서 제안된 [인증서 검증]의 (Step 1)과 (Step 2)에 의해서, t2를 "유효성 시작 지점"으로 t5를 "유효성 종료 지점"으로 설정하고 $CERT_U$를 t5까지 캐쉬한다. 따라서, V는 t5까지 $CERT_U$의 취소 유무 상태를 악의적 사용자 U가 전송한 해쉬값의 검증 결과에 의존한다. 따라서, 취약성 윈도우는 t3에서 t5까지 된다.

그림 1과 같이, CASE 3의 취약성 윈도우 보다 CASE 2의 취약성 윈도우가 작음을 알 수 있다. 하지만, CASE 2는 SS라는 부가적인 신뢰 개체를 의존함으로써 얻어진 결과이며, SS는 인터넷과 같은 글로벌 환경에서 적용 불가능하기 때문에 본 논문에서 제안되는 공개키 프레임워크가 인터넷 환경에서 좀 더 현실적인 대안이 될 것으로 사료된다. 또한, CASE 3에서의 취약성 윈도우의 크기는 사용자의 인증서 내에 정의된 제어 윈도우의 크기에 전적으로 의존될 수 있다. 그러므로, 제어 윈도우의 크기는 충분히 주의 깊게 이뤄져야 할 것이다.

4.2 통신 비용

본 절에서는 CRL, ZBF, ZMCC 그리고 제안된 공개키 프레임워크에서의 일간의 통신비용(daily communication cost)을 분석한다. CA는 발행된 인증서들에 대한 취소 정보를 분배하기 위해서 기본적으로 CRL을 사용하는 것으로 가정한다. 또한, 아래의 파라메터들을 사용한다.

- n : 발행된 인증서들의 전체 추정 수 (n= 300,000).
- p : 만기일 이전에 취소될 인증서들의 추정 비율 (p= 0.1).
- q : 하루에 발생되는 인증서 상태 검증 질의 횟수.
- t : CRL이 하루에 갱신되는 횟수. t= 2이면, 12 시간마다 주기적인 갱신이 발생.
- L : ZBF 또는 ZMCC에서의 인증서 유효성 갱신 기간. L= 1이면, 하루를 의미.
- C : 제안된 공개키 프레임워크에서의 제어 윈도우의 크기. C= 2이면, 제어 윈도우의 크기는 2일.
- l_{sn} : 인증서의 시리얼 번호(serial number)를 위해 요구되는 비트 수 (l_{sn}= 20).
- l_{sig} : 전자 서명문의 비트 수 (l_{sig}= 1024).
- l_{hash} : 일방향 해쉬함수를 통해 계산된 해쉬값의 비트 수 (l_{hash}= 160).

n, p, q, l_{sn}을 위한 값들은 [13]과 동일한 값들을 사용했으며, l_{sig}와 l_{hash}는 [8]과 동일하다. 또한, 본 논문에서 t, L, C를 새로이 정의하였다. CRL, ZBF, ZMCC 그리고 제안된 공개키 프레임워크에서 하루에 각각 소요되는 평균 통신비용은 다음과 같다.

〔CRL의 일간 통신 비용〕

모든 인증서 상태 검증 질의에 대해서, CA는 CRL 전체를 응답으로 전송해야 한다.

$$t \cdot q \cdot p \cdot n \cdot l_{sn} + t \cdot q \cdot l_{sig}$$

〔ZBF의 일간 통신 비용〕

사용자들은 L마다 q개의 해쉬값들을 검증을 위해 전송한다.

$$\frac{q \cdot l_{hash}}{L}$$

〔ZMCC의 일간 통신 비용〕

SS들은 주기적으로 CA로부터 CRL을 다운로드 해야만 하며, 사용자들에게 q개의 해쉬값들을 전송해줘야 한다. 그리고, 사용자들 또한 수신 받은 q개의 해쉬값들을 검증을 위해 전송한다.

$$t \cdot q \cdot p \cdot n \cdot l_{sn} + t \cdot q \cdot l_{sig} + 2 \cdot \frac{q \cdot l_{hash}}{L}$$

〔제안된 공개키 프레임워크의 일간 통신 비용〕

검증자들은 인증서를 최초 검증하기 위해서 "유효성 시작 지점"에서 CA에게 CRL을 질의한다. 그리고, 사용자들은 q개의 해쉬값들을 검증을 위해서 전송한다.

$$\frac{q \cdot p \cdot n \cdot l_{sn}}{C} + \frac{q \cdot l_{sig}}{C} + q \cdot l_{hash}$$

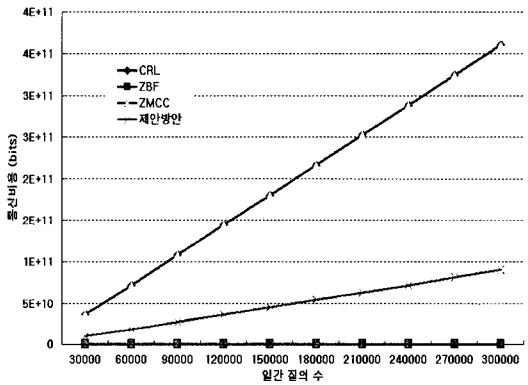

그림 2. 질의 수 변화에 따른 일간 통신비용

그림 2는 소개된 4가지 방안들에서 통신비용을 계산하기 위하여 공통적으로 사용되는 "인증서 상태 검증 질의 횟수"의 변화에 따른 전체 통신비용의 증가량을 보여주고 있다. J. Zhou 등의 주장과는 달리 [3]의 ZMCC는 CRL과 거의 유사하게 아주 높은 통신비용을 요구하며, ZBF의 경우 가장 효율적인 통신비용을 나타내고 있다. 하지만, 앞 절에서 소개된 바와 같이, ZBF의 취약성 윈도우즈에 대한 취약점으로 인하여, 실질적인 시스템으로의 적용은 불가능하기 때문에 효율적인 통신비용의 결과는 의미가 없다 하겠다. 반면, 제안된 공개키 프레임워크는 CRL과 ZMCC에 비하여 적은 통신비용을 요구하는 것을 알 수 있다.

V. 개선된 유무선 장치간의 인증된 채널 설정 프로토콜

본 장에서는 제안된 공개키 프레임워크를 통하여, [4]에서 제안된 유·무선간의 인증된 채널을 설정하기 위한 프로토콜을 통신 및 계산 비용 측면에서 좀 더 효율적으로 개선하고자 한다.

5.1 E2ESP

J. Yang 등은 기존의 무선 인터넷 환경에서 가장 주요한 문제점인 통신 개체사이의 종단간의 인증을 수행하기 위하여 End-to-End Security Protocol(E2ESP)를 제안하였다[5]. E2ESP에서는 CRL-Agent라는 신뢰되는 개체가 무선 단말기의 사용자들이 수행해야 할 인증서 취소 유무 상태 검증에 대한 처리를 사용자들을 대신하여 수행함으로써, 사용자들은 통신을 하고자 하는 상대방의 인증서 취소 유무를 판단하기 위한 대역폭의 낭비를 최소화하는 기법이다. 하지만, CRL-Agent라는 부가적인 신뢰 개체에 의존하는 단점을 가지고 있다. 본 논문에서는 기 제안된 E2ESP를 제안된 공개키 프레임워크를 사용하여 CRL-Agent을 효율적으로 제거하고자 한다. 본 장에서는 다음과 같은 용어들이 사용된다.

- $E_K(M), D_K(M)$: 메시지 M을 키 K로서 암호화 및 복호화.
- $seqN$: 전송되는 메시지의 순차 번호.
- ref : 얼마나 자주 세션키들이 갱신되는 가를 의미한다. 새로운 세션키는 $n=2^{ref}$ 메시지마다 새로이 계산된다. 즉, 새로운 세션키의 $seqN$는 $0, n, 2n, 3n, \cdots$.
- $Krls$: 유선 장치인 서버가 제공 가능한 모든 갱신 주기들의 리스트.

본 장에서 제안하는 개선된 E2ESP(Modified E2ESP, ME2ESP)는 E2ESP와 같이 서버와 무선장치 사용자간의 비대칭적 상호 인증을 제공한다. 여기서, 비대칭적이라는 의미는 서버는 사용자의 패스워드를 통한 인증을 수행하고, 사용자는 서버를 인증서를 통해서 인증한다. 특히, 사용자 인증 및 채널 보호를 위한 키 교환을 위해서 EKE(Encrypted Key Exchange)를 사용한다[9][12][14]. 본 장을 위한 가정 사항은 아래와 같다.

- 시스템을 위한 공개 파라메터 : 통신 개체들은 큰 소수 p와 생성원 g를 "g가 mod$p$의 원시근(primitive)"으로 협정한다. 여기서, p와 g는 공개값.
- 유선 장치인 서버 S는 공개키 인증서($CERT_S$)를 소유하고 있다.
- 사용자는 접속하고자 하는 서버에 이미 등록되어 있음을 가정한다. 여기서, 등록의 의미는 사용자의 패스워드 관련 정보를 아래와 같은 절차를 통하여 서버가 안전하게 저장하고 있다.

① 사용자 U 는 자신을 위한 패스워드를 $Upass$로 설정하고, 임의의 랜덤비트열 λ를 생성한다.

② U는 $E_{Upass}(\lambda)$를 계산하여, 무선 장치에 저장한다. 또한, $v = g^{H(\lambda, U, S)} \bmod p$를 계산하여 서버에게 등록한다.

③ 서버는 사용자 인증을 위해서, 사용자 패스워드에 관련된 정보인 v를 시스템의 패스워드 파일에 안전하게 저장한다.

본 장에서 제안되는 ME2ESP는 새로운 연결 프로토콜(New Connection Protocol, ME2ESP-NCP)와 축약된 연결 프로토콜(Reduced Connection Protocol, ME2ESP-RCP)의 두 가지 형태의 프로토콜로 구성되어 있다. 사용자가 최초로 서버와 인증된 채널을 설정하고자 할 때는 ME2ESP-NCP를 통하여 안전한 채널을 설정하게 되며, 그 후에 서버의 인증서내의 제어 윈도우에 해당하는 기간 동안 발생하는 인증된 채널 설정 시에는 ME2ESP-RCP를 통하여 안전한 채널을 설정한다.

5.2 새로운 연결 프로토콜(ME2ESP-NCP)

단순화를 위해서 mod$p$ 연산은 생략한다. ME2ESP-NCP를 통한 유·무선 장치간의 인증된 채널 설정을 위한 절차는 다음과 같다.

(Step 1) 사용자 U는 서버 S에게 단순한 로그인 요청 메시지를 전송한다.

[LoginReq] $U \rightarrow S$: "User Login Request"

(Step 2) S는 랜덤한 큰 수 정수 x를 선택하고, g^x를 계산한다. 물론, x는 서버의 채널 설정을 위한 임시 비밀정보가 된다. 그리고, S는 랜덤 챌린지로서 r_S를 생성하고, ServerRep1을 생성하여 U에게 전송한다.

[ServerRep1] $S \rightarrow U$: $CERT_S, SIGN_S(g^x, Krls, r_S)$

(Step 3) U는 $CERT_S$내의 인증기관(CA)의 전자서명을 검증하고, CA에게 인증서 취소 유무 검증을 위한 상태 정보를 질의한다. 만약, 두 개의 검증이 성공하면, U는 $CERT_S$내의 공개 정보인 PK_S와 $H^j(r)$을 알게 된다. U는 $CERT_S$의 제어 윈도우에 의해서 "유효성 시작 지점"과 "유효성 종료 지점"을 설정한다. U는 ServerRep1내의 전자 서명된 부분을 PK_S로 검증하여, 유효한 $g^x, Krls, r_S$를 얻고, 자신의 패스워드를 입력하여 $E_{Upass}(\lambda)$를 복호화한다. 그리고, U는 $Krls$로부

터 적절한 ref를 선택하고, 랜덤 챌린지 r_C를 생성한다. U는 마스터 비밀 키(MS)와 세션키(SK)를 아래와 같이 계산한다. 단, 여기서 초기 $seqN$는 0로 설정한다.

$$MS = H((g^x)^{H(\lambda,U,S)}),$$
$$SK = H(MS, r_S, r_C, seqN)$$

그리고, U는 UserRep를 S에게 전송한다.

[UserRep] $U \rightarrow S$: $U, E_{SK}(ref, r_S), r_C$

(Step 4) S는 $MS = H(v^x)$, $SK = H(MS, r_S r_C, seqN)$를 계산하고, UserRep에서 SK로 암호화된 부분을 복호화한다. S는 복호된 r_S와 ServerRep1에서 자신이 전송한 r_S를 비교한다. 만약 두 값이 동일하면, S는 사용자 U를 인증한다. 최종적으로, S는 Finish를 U에게 전송한다.

[Finish] $S \rightarrow U$: $E_{SK}(r_C)$

(Step 5) U는 Finish를 복호화하여, r_C와 비교를 한다. 두 값이 동일하면, U는 생성된 세션키가 안전한 채널을 위해서 사용될 수 있음을 알게 된다. 최종적으로, U는 S에 대한 $H(PK_S), H^j(r), j, MS$를 "유효성 종료 지점"까지 캐쉬한다. 여기서, MS의 경우는 U의 무선 장치 내에 안전하게 저장되어야 한다.

$seqN$는 U와 S사이에 교환되는 메시지마다 매번 갱신된다. 이때, ref에 의하여 SK는 단지 $seqN$만이 갱신된 상태로 주기적으로 재계산될 것이다.

5.3 축약된 연결 프로토콜(ME2ESP-RCP)

ME2ESP-NCP가 성공적으로 수행된 후, 서버의 인증서 내의 제어 윈도우에 의해서 정의된 "유효성 종료 지점"까지, 사용자는 서버와의 인증된 채널을 설정하기 위하여 ME2ESP-RCP를 수행할 수 있다.

(Step 1) U는 LoginReq를 S에게 전송한다.

(Step 2) S는 현재의 $H^i(r), i$를 계산하고, ServerReq2를 U에게 전송한다.

[ServerRep2] $S \rightarrow U$: $PK_S, H^i(r), i, Krls, r_S$

(Step 3) U는 ServerRep2를 수신한 현재 시간이 "유효성 만료 지점"을 지났는가를 검사한다. 만약, 지나지 않았다면, U는 $H^{j-i}(H^i(r)) = H^j(r)$를 검사한다. 유효할 경우, U는 PK_S를 해쉬 처리하여, 캐쉬된 값과 비교한다. 만약 참이면, U는 $CERT_S$는 아직 유효하며, S의 공개키는 PK_S라고 믿는다. U는 랜덤 챌린지로서 r_C를 생성하고, $seqN$를 0으로 설정한다. 그리고, U는 안전하게 저장되었던 MS를 로드하여, SK를 ME2ESP-NCP와 동일한 방법으로 생성한다. 그 이후의 단계는 ME2ESP-NCP와 동일하게 수행된다.

ME2ESP-RCP에서는 사용자가 더 이상 인증서 내의 CA 전자 서명문의 유효성 및 취소 유무 상태 검증을 위한 통신 및 계산 비용을 낭비할 필요가 없다. 또한, 사용자는 MS를 계산하기 위한 지수 연산을 수행할 필요도 없다. 따라서, 사용자는 신속하고 안전하게 서버와의 인증된 채널을 설정 가능하다.

5.4 ME2ESP의 특징

본 장에서 제안된 ME2ESP는 기존의 PKI를 고려하여 설계된 E2ESP에 비하여, 무선 장치에서의 통신 및 계산 비용을 감소시켰으며, CRL-Agent라는 부가적인 신뢰개체를 효율적으로 제거하였다. 또한, 만약 전송되는 메시지에 대한 메시지 인증이 요구되는 환경에서는, 세션키를 유도하는 함수의 간단한 변경을 통하여 메시지 인증 코드(MAC)를 위한 대칭키의 생성이 가능하다.

ME2ESP-NCP 수행 이후, 사용자의 무선 장치는 제어 윈도우 기간 동안 MS를 안전하게 보관하고 있어야 하는 부담을 가지고 있다. 따라서, 만약 제시된 ServerRep2를 아래와 같이 변경하면, 사용자의 무선 장치는 MS를 안전하게 보관할 필요가 없어진다.

[ServerRep2] $S \rightarrow U$: $PK_S, H^i(r), i, g^x, Krls, r_S$

하지만, 이 경우에는 사용자의 무선 장치는 ME2ESP-RCP 프로토콜 수행 시에 새로운 MS 계산을 위한 지수 연산을 부가적으로 수행해야하는 단점을 가진다.

Ⅵ. 결 론

본 논문에서는 J. Zhou 등이 제안한 공개키 프레임워크의 몇 가지 보안 파라메타들을 재조명하였고 실제 PKI 환경에서의 활용 가능하도록 개선하였다. 또한, J. Zhou 등의 공개키 프레임워크에서 불필요한 신뢰기관을 제거함과 동시에 대안으로 제어 윈도우라는 새로운 형태의 인증서 취소 처리를 위한 메커니즘을 제안하였다. 또한, 본 논문에서는 기 제안되었던 유·무선 장치간의 인증된 채널 설정을

위한 프로토콜을 통신 및 계산 비용을 향상시킨 새로운 프로토콜로서 재설계하기 위하여, 제안된 새로운 공개키 프레임워크를 적용하였다.

참 고 문 헌

[1] C. Adams and S. Lloyd, "*Understanding public-key infrastructure: concepts, standard, and deployment considerations,*" Indianapolis: Macmillan Technical Publishing, (1999).

[2] D. Cooper, "A more efficient use of delta-CRLs," *Proceeding of 2000 IEEE Symposium on Security and Privacy*, pp.190-202, (2000).

[3] J. Zhou, F. Bao and R. Deng, "An Efficient Public-Key Framework," *5th International Conference on Information and Communications Security*, LNCS 2836, pp.88-99, (2003).

[4] J. Yang, W. Shin and K. Rhee, "An end-to-end authentication protocol in Wireless Application Protocol," *ACISP 2001*, LNCS 2119, pp.247-259, (2001).

[5] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "*X.509 Internet public key infrastructure on-line certificate status protocol (OCSP),*" RFC 2560, (1999).

[6] N. Asokan, G.Tsudik and M.Waidner, "Server-Supported Signatures," *European Symposium on Research in Computer Security*, pp.131-143, (1996).

[7] N. Haller, "The S/Key One-time Password System," *Proceeding of ISOC Symposium on Network and Distributed System Security*, pp.151-157, (1994).

[8] M. Naor and K. Nissim, "Certificate revocation and certificate update," *Proceedings 7th USENIX Security Symposium*, San Antonio, Texas, pp.217-228, (1998).

[9] Peter Buhler, Thomas Eirich, Michael Stenier and Michael Waidner, "Secure Password-Based Cipher Suite For TLS," *In Symposium on Network and Distributed Systems Security* (NDSS '00), pp.129-142, (2000).

[10] P. McDaniel and S. Jamin, "Windowed certificate revocation," *Proceedings of IEEE INFOCOM'2000,* Tel-Aviv, Israel, pp.1406-1414, (2000).

[11] R. Housley, W. Ford, W. Polk and D. Solo, "*Internet X.509 public key infrastructure certificate and CRL profile,*" RFC 2459, (1999).

[12] S. Halevi and H. Krawczyk, "Public-Key Cryptography And Password Protocols," *In 5th ACM Conference on Computer and Communication Security*, San Francisco, California, pp.122-131, (1998).

[13] S. Micali, "*Efficient Certificate revocation,*" Technical Memo MIT/LCS/TM-542b, (1996).

[14] Steven M. Bellovin, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," *Proceedings of the IEEE Symposium on research in Security and Privacy*, Oakland, , pp.72-84, (1992).

양 종 필 (Jong-Phil Yang) 정회원

1999년 부경대학교 전자계산학과 졸업(학사)

2001년 부경대학교 대학원 전자계산학과 졸업(석사)

2005년 부경대학교 대학원 전자계산학과 졸업(박사)

2005년~현재 일본 큐슈대학교 시스템정보과학부 객원연구원

<관심분야> 유비쿼터스 컴퓨팅 보안, 비밀 분산, 공개키 기반 구조, 익명성

신 원 (Weon Shin) 정회원

1996년 부경대학교 전자계산학과 졸업(학사)

1998년 부경대학교 대학원 전자계산학과 졸업(석사)

2001년 부경대학교 대학원 전자계산학과 졸업(박사)

2002년~2005년 (주)안철수연구소 선임연구원

2005년~현재 동명대학교 정보보호학과 전임강사

<관심분야> 소프트웨어 보안, 악성코드 확산, 이동 에이전트 시스템, 암호 프로토콜 응용

이 경 현 (Kyung Hyune Rhee) 정회원

1982년 경북대학교 수학교육과 (이학사)
1985년 KAIST 응용수학과(이학석사)
1992년 KAIST 수학과(이학박사)
1982년~1993년 ETRI 선임연구원
1995년~1996년 Univ. of Adelaide, Australia, 방문교수
2001년~2002년 Univ. of California, Irvine, 교환교수
2002년~2003년 국제간 정부기구 CPSC 교학부장, Manila, Philippines
1993년~현재 부경대학교 전자컴퓨터정보통신공학부 교수
<관심분야> 암호이론, 멀티미디어 정보보호, WSN 보안, 암호 프로토콜 응용