情報保護學會論文誌
第13券 第4號, 2003. 8

IC 칩을 내장한 무선 단말기에 적용 가능한 키 분배 프로토콜

안 기 범*, 김 수 진*, 한 종 수*, 이 승 우*, 원 동 호**

Key Distribution Protocol Appropriate to Wireless Terminal Embedding IC Chip

Gi-Bum An*, Soo-Jin Kim*, Jong-Su Han*, Seung-Woo Lee*, Dongho Won**

요 약

현재 co-processor를 탑재한 IC 칩이 계속 출시되고 있어 IC 칩의 연산 능력이 나날이 발전하고 있다. 또한, 무선 단말기 시장에는 간편하고 다양한 서비스를 제공하기 위해 IC 칩(Integrated Circuit Chip)을 내장한 무선 단말기 제품이 많이 출시되고 있다. 하지만 현재 IC 칩에 탑재된 co-processor의 연산 능력은 아직 유선 통신 환경의 연산 능력에 미치지 못하고 있어 기존 유선 통신 환경의 키 분배 프로토콜을 무선 통신 환경에 그대로 활용하기 어렵다. 따라서 본 논문에서는 무선 단말기의 제한적인 연산 능력을 고려하여 암호 전용 연산을 하는 co-processor를 무선 단말기에 탑재함으로써 연산 능력을 보완하고, 기존의 이동 통신 환경에서의 키 분배 프로토콜에서 제공하지 않는 보안 요구 사항을 만족하며, 사용자와 서버 양측에 연산 부담을 줄일 수 있는 무선 단말기 환경에 적합한 키 분배 프로토콜을 제안한다.

ABSTRACT

Computational power of IC chip is improved day after day producing IC chips holding co-processor continuously. Also a lot of wireless terminals which IC chip embedded in are produced in order to provide simple and various services in the wireless terminal market. However it is difficult to apply the key distribution protocol under wired communication environment to wireless communication environment. Because the computational power of co-processor embedded in IC chip under wireless communication environment is less than that under wired communication environment. In this paper, we propose the key distribution protocol appropriate for wireless communication environment which diminishes the computational burden of server and client by using co-processor that performs cryptographic operations and makes up for the restrictive computational power of terminal. And our proposal is satisfied with the security requirements that are not provided in existing key distribution protocol.

keyword : *IC chip, key distribution, wireless communication environments*

Ⅰ. 서 론

최근 정보통신 기술의 발전과 휴대 단말기 사용의 보편화로 무선 인터넷 등과 같은 기술을 이용하는 무선 통신 사용자가 급격히 증가하고 있다. 그로 인해 유선 통신 환경의 전자상거래, 전자 결제 등의 다양한 서비스를 무선 통신 환경에서도 제공하기 위한 노력이 계속되고 있다. 무선 통신 환경에서도 이러한

* 성균관대학교 정보통신공학부 정보통신보호연구실({gbahn, kimsj, jshan, swlee}@dosan.skku.ac.kr)
** 성균관대학교 정보통신공학부 정교수(dhwon@dosan.skku.ac.kr)

서비스를 성공적으로 제공하기 위해서는 보안 문제가 우선 해결되어야 한다. 즉, 유선 통신 환경과 동일하게 무선 통신 환경에서도 안전한 서비스를 제공하기 위해 기밀성(confidentiality), 인증(authentication), 무결성(integrity), 부인 봉쇄(non-repudiation) 등을 제공하여야 한다.

하지만 무선 통신 환경은 유선 통신 환경과 달리 많은 제약 조건이 있다. 즉, 적은 용량의 배터리, 작은 크기의 화면, 낮은 성능의 CPU, 적은 메모리 등의 단말기 환경의 제약 조건과 무선 통신망의 제한된 통신 속도, 통신 에러율 등의 무선 통신 환경의 제약 조건이 있다.

최근 무선 통신 환경과 단말기 기술의 급속한 발전으로 기존 단말기의 제약 조건을 극복하고 다양한 기능을 가지는 무선 단말기가 개발되어 판매되고 있으며, 그러한 무선 단말기를 통해 다양한 서비스를 제공하고 있다. 하지만 이러한 서비스가 널리 이용되기 위해서는 서비스 제공자와 사용자 사이에 신뢰가 전제되어야 한다. 서비스 제공자와 사용자간의 신뢰를 무선 통신 환경 상에서 제공하기 위해서는 정보보호 기술이 적용되어야만 한다. 하지만 현재 유선 통신 환경에서 사용되고 있는 정보보호 기술을 그대로 무선 통신 환경에서 이용하기가 어렵다. 그 중 대표적인 문제점으로는 무선 단말기의 제한적인 연산 능력이다. 따라서 현재 많은 무선 단말기 제조 업체에서 무선 통신 환경과 유선 통신 환경의 단말기의 격차를 줄이기 위해 많은 노력을 기울이고 있다.

이를 보완하기 위해서 IC 칩을 내장한 무선 단말기가 개발되었다. 또한 현재 개발되고 있는 IC 칩의 경우도 종전의 단순한 저장 기능과 연산 기능에 만족하지 않고 자체적인 co-processor를 탑재함으로써 기존의 IC 칩과 연산 능력에서 차별을 두고 있다. 하지만 유선 통신 환경에서와 같은 연산 능력은 갖추지 못하고 있는 실정이다. 따라서 여러 IC 칩 제조회사에서는 IC 칩의 자체적인 연산 능력을 향상시키기 위해 많은 노력을 기울이고 있다. 이에 대한 노력으로 Gemplus사의 스마트 카드 제품인 GemXplore Trust 경우, 자체적인 RSA 암호 연산이 가능한 co-processor를 탑재하여 RSA 1024비트의 전자 서명이 가능하고 WAP(Wireless Application Protocol)에서 사용하기 적합하도록 개발되어 판매되고 있다.

본 논문에서는 무선 단말기의 제한적인 연산 능력을 보완하기 위해 암호 전용 연산을 하는 co-processor를 탑재한 IC 칩을 무선 단말기에 내장하여 부족한 연산 능력을 보완하고, 사용자(무선 단말기)와 서버(기지국)의 연산 부담을 기존의 이동 통신 환경의 키 분배 프로토콜과 비교하여 최소화함으로써 무선 단말기 환경에 적합한 키 분배 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 연구 배경에 대해 설명하고, 3장에서는 무선 통신 환경에서의 키 분배 프로토콜의 보안 특성에 대해 살펴보고, 4장에서는 본 논문에서 제안하는 키 분배 프로토콜에 대해서 상세히 서술한다. 5장에서는 기존의 프로토콜과 보안 요구 사항과 계산량에 대해 비교하고, 제안하는 프로토콜의 안전성을 분석하며, 6장에서는 결론과 향후 연구 방향에 대해 서술한다.

II. 연구 배경

기존의 무선 통신 환경에서 사용할 수 있는 키 분배 프로토콜의 경우 연산 과정이 무선 단말기 내에서 이루어진다. 그렇지만 무선 단말기의 연산 능력이 유선 통신 환경의 단말기에 비해 상대적으로 적은 편이므로 유선 통신 환경에서의 키 분배 프로토콜을 무선 통신 환경에서 그대로 적용할 수 없는 한계를 갖게 된다. 또한 무선 단말기 내의 메모리에 개인키, 인증서, 개인 비밀 정보 등의 중요 정보가 연산 과정에서 노출될 수 있다는 문제점을 내포하고 있다. 이러한 문제점은 co-processor를 탑재한 IC 칩을 단말기에 내장함으로써 극복할 수 있다.

IC 칩의 초기 형태는 스마트카드로서 1974년 Innovation S.A의 Roland Moreno에 의해서 처음으로 스마트카드에 대한 특허가 출원되었다. 그 이후, 최초의 스마트카드는 1977년 BULL사와 Motorola사의 합작으로 등장하게 되었다. 일반적인 스마트카드는 신용카드 크기 정도의 플라스틱에 IC 칩을 장착한 것으로 단순한 메모리 기능만을 갖는 메모리카드에 비해 더욱 안전하고, 개방형 네트워크를 통하여 사용자를 인증할 경우 일회용 패스워드(one-time password)를 사용하므로, 네트워크 상에서 전송되는 정보의 도청(eavesdropping)에 의한 문제를 해결할 수 있다.

일반적으로 스마트카드의 메모리는 개인 식별 번호(PIN: Personal Identification Number)가 입력되지 않으면 내용을 읽을 수 없도록 되어 있으며, 위·변조 방지를 위한 불법 변조 방지(tamper-resistant) 특성을

갖도록 설계되어 있다. 또한 스마트카드는 사용자가 단 한번의 로그인으로 네트워크를 통하여 많은 컴퓨터에 접근하는 방법을 제공하는 장점을 가지고 있다.

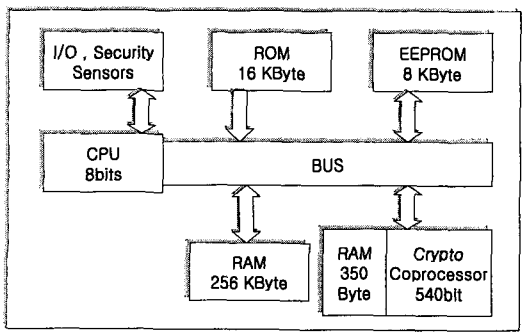
이와 같은 장점으로 인해, 스마트카드의 IC 칩은 현재 보안 모듈의 형태로 개발되고 있으며, 무선 환경에서 사용할 수 있는 대표적인 스마트카드로는 WIM (WAP Identity Module)과 SIM(Subscriber Identity Module)이 있다.

WIM이란 WAP 포럼에서 규정하고 있는 방식으로 개인키와 인증기관의 인증서와 같은 중요 정보를 저장하고 있으며, 저장된 정보를 이용하여 암호 연산을 수행하고 마스터 시크릿(master secret)를 계산·저장하는 등의 동작을 수행한다. 또한 WIM은 WTLS(Wireless Transport Layer Security) 계층에서의 암호 연산과 응용 계층에서의 전자 서명을 지원한다.

SIM은 무선 단말기의 무선 전자 과금·결제 솔루션 분야에서 사용되고 있다. SIM 카드는 전화번호, 가입자가 가입한 이동 통신 서비스 제공 사업자, 개인 전화번호부 등을 포함하는 카드로 기본적인 개인 인증 기능을 수행하며, 가입자 로밍, 인증 모듈 등의 기능이 추가되고 있다. 현재 SIM과 같은 계열로는 CDMA 2000에서 사용할 수 있는 UIM(User Identity Module)과 IMT 2000에서 사용할 수 있는 USIM(Universal Subscriber Identity Module)이 대표적이다.

IC 칩은 크게 CPU, ROM, EEPROM, RAM, Security Sensor로 구성되며, 암호 전용 연산 co-processor를 탑재하게 되면 다음과 [그림 1]과 같은 구조를 갖게 된다.

일반적으로 암호 연산용으로 사용되는 arithmetic crypto co-processor는 모듈러 연산을 위한 연산 모듈을 내장하고 있으며, co-processor는 모듈 내부에 암호



〔그림 1〕 IC 칩의 구조

알고리즘을 내장하여 이를 이용하여 데이터의 암호화 등 암호 연산에 필요한 연산 과정을 IC 칩 내에서 수행함으로써 무선 단말기의 연산 부담을 줄일수 있다. 또한 이를 이용하여 암호 연산을 무선 단말기 내에서 수행하는 것이 아니라 대부분 IC 칩 내에서 수행함으로써 연산 과정이 보다 안전하게 이루어질 수 있다.

개인 비밀 정보가 기존의 단말기 메모리에 저장되어 있는 경우, 개인키, 인증서 등이 저장 과정 또는 연산 과정에서 노출될 우려가 있다. 이러한 개인 비밀 정보는 외부에서 정보의 주입, 변경 등 접근이 어려운 IC 칩 내의 불가침 메모리 영역에 저장하고, co-processor와 관계된 암호 연산 과정에서 주요 정보를 IC 칩 내부에서만 이용할 수 있게 함으로써 외부의 침입으로부터 안전한 암호 연산을 지원하며 중요 정보가 노출되는 것을 예방할 수 있다.

또한, 기존의 무선 통신 환경에서의 키 분배 프로토콜의 경우 양방향 개체 인증을 제공하지 않거나 key freshness를 보장하지 않아 재전송 공격(replay attack)에 대한 취약성을 내포하고 있으며 부인 봉쇄 기능을 제공하지 않는 경우도 있다([표 2] 참고). 이러한 경우 무선 통신 환경의 비 대면적인 특성을 갖는 무선 전자상거래, 무선 전자 결제 등 다양한 서비스를 안전하게 이용할 수 없다.

따라서, 본 논문에서 제안하는 키 분배 프로토콜에서는 무선 단말기를 처음 등록하는 과정에서 비밀 정보를 사용자와 서버만 비밀리에 공유하고 이 비밀 정보를 이용하여 보다 안전한 키 분배 과정을 수행할 수 있으며, 기존의 무선 통신 환경의 키 분배 프로토콜에서 제공하지 않는 보안 요구 사항을 추가 제공한다. 또한, 키 분배 프로토콜을 수행할 때, 단말기에 개인 식별 번호를 입력하여 사용하게 함으로써 단말기를 분실한 경우에도 개인 비밀 정보의 누출로 인한 위험을 줄일 수 있다.

Ⅲ. 키 분배 프로토콜의 보안 요구 사항

유럽의 차세대 이동통신 표준인 UMTS(Universal Mobile Telecommunication Systems)에서는 안전한 보안 서비스를 제공하기 위해서 ASPeCT 프로젝트에서 연구된 키 분배 프로토콜의 보안 특성과 보안 요구 사항을 아래와 같이 정의하였다.[1,6]

- 개체 인증(entity authentication)

실시간으로 키 분배 프로토콜에 참여하고 있는 상대방의 신원을 확인하는 과정으로 명시적 키 인증(explicit key authentication)에 의해 제공될 수 있고 또한 객체사이에서 공유된 비밀 정보의 소유 여부를 통해 제공될 수 있으며 다른 객체에 대한 위장(masquerade)을 방지할 수 있어야 한다.

- 공개키 인증서의 상호 교환(exchange of certified public keys)
  공개키 기반의 키 분배 프로토콜을 지원하기 위해서는 프로토콜에 참여하는 상대방의 공개키의 정당성에 대한 확인 과정이 필요하며 이를 위해 공개키 인증서를 상호 교환한다.
- 세션키에 대한 상호 동의(mutual agreement of a secret key)
  사용자와 서버 사이에서 교환되는 데이터를 보호하기 위해서 세션키는 사용자와 서버가 상대방과 자신의 정보를 이용하여 각각 생성해야 한다.
- 세션키의 상호 제어(joint control of the secret key)
  다른 주체가 우연 또는 고의적으로 약한 키를 선택하는 것을 방지하기 위해 사용자와 서버가 세션키에 영향을 미치는 정도가 동일해야 한다. 즉, 세션키를 생성하기 위해서 사용되는 정보의 정도가 동일해야 한다.
- 키 인증(key authentication)
  키 인증은 묵시적 키 인증(implicit key authentication)과 명시적 키 인증(explicit key authentication)으로 분류할 수 있다.
  - 묵시적 키 인증 : 키의 소유 여부는 알려져 있지 않다고 하더라도 키 분배 프로토콜에 참여한 상대방만이 세션키를 생성할 수 있음을 보장
  - 명시적 키 인증 : 사용자는 자신이 통신을 하고자 하는 상대방만이 세션키를 계산할 수 있고 상대방이 실제로 그 키를 가지고 있음을 확인할 수 있음
- key freshness
  매 세션마다 다른 세션키를 생성하여 이전 세션의 전송 정보의 재사용하는 재전송 공격을 방지하기 위해 필요하다.
- 사용자의 익명성(anonymity)
  공격자에 의해 임의의 사용자의 신분이나 위치를 추적 당하는 것을 방지하기 위해 익명성을 보장해야 한다.
- 부인 봉쇄(non-repudiation)
  중요한 데이터의 전송이나 수신한 사실을 부인할 수 없어야 한다.

Ⅳ. 제안하는 키 분배 프로토콜

본 장에서는 논문에서 제안하고 있는 키 분배 프로토콜에서 사용될 시스템 파라미터를 설명하고, 사용자가 단말기를 처음 등록할 때 개인 식별 번호와 임시 개인 식별 정보 TID_A를 공유하는 등록 과정과 사용자와 서버간에 공유된 개인 식별 번호를 이용하여 세션키를 분배하는 키 분배 프로토콜을 기술한다.

4.1 시스템 파라미터

본 논문에서는 사용되는 시스템 파라미터는 다음과 같다.

- A, U, V : 사용자
- B, S : 서버
- E : 공격자
- p : $GF(p)$를 정의하는 큰 소수
- q : $q|p-1$
- g : Z_p에서 위수 q를 갖는 원시원소
- P_X : 개체 X의 공개키($P_X \equiv g^{S_X} \bmod p$)
- S_X : 개체 X의 개인키
- r_X : 개체 X가 생성한 랜덤 수
- SK : 세션키
- E_K : 대칭키 K를 이용한 암호화 연산
- D_K : 대칭키 K를 이용한 복호화 연산
- PE_{P_X} : 개체 X의 공개키를 이용한 암호화 연산
- PD_{S_X} : 개체 X의 개인키를 이용한 복호화 연산
- Sig_X : 개체 X의 서명
- Ver_X : 개체 X의 서명 검증
- ID : 개인 식별 정보(identity)
- $h(\ )$: 일방향 해쉬 함수
- PIN : 개인 식별 번호
- TID_A : 사용자 A의 임시 ID
- TS_X : 개체 X가 생성한 타임스탬프
- RN_B : 등록 과정에서 서버 B가 생성한 랜덤 수
- $||$: 연접(concatenation)

4.2 등록 과정

등록 과정은 사용자가 무선 단말기를 처음 등록하여 서버와의 안전한 키 분배 프로토콜을 수행하기 위해 사용자가 선택한 개인 식별 번호와 익명성을 제공하기 위한 임시 ID를 사용자와 서버 양측에 공유하는 과정이다. 무선 통신 환경에서는 사용자의 위치 등 개인의 프라이버시를 보장해 주어야 하므로 실제 ID를 사용하지 않고, 실제 ID와 연결 관계를 갖는 임시 ID를 사용한다. 임시 ID 생성시, 서버 B의 랜덤 수 RN_B를 포함함으로써 임시 ID 변경시 동일한 임시 ID가 생성되는 것을 방지하고, 재전송에 의한 공격을 예방한다. 상대방의 공개키 인증서의 유효성을 검증하는 과정을 사용자나 서버가 직접 수행하는 경우 무선 통신 환경에서 높은 부하를 가져올 수 있다. 이를 방지하기 위해 실시간으로 공개키 인증서의 상태 정보를 검증할 수 있는 OCSP(Online Certificate Status Protocol)을 이용할 수 있다. 인증서를 검증하기 위해 OCSP를 이용하는 경우 사용자는 인증서의 유효성에 대한 결과를 받아 이를 사용할 수 있으므로 이와 관련된 모든 인증서의 유효성 검증 절차를 수행하지 않아 연산 부담을 줄일 수 있다. 현재 우리나라에서는 6개의 공인인증기관에서 무선 OCSP 서비스를 제공하고 있다. 등록 과정은 초기 사용자와 서버간의 비밀 정보와 TID를 교환하는 과정으로 초기에 한번만 수행된다.

등록 과정은 다음과 같다.

① 사용자 A는 OCSP 서버에게 서버 B의 공개키 인증서의 유효성 검증을 요청하고 유효한 인증서로 검증되면, PIN을 선택하고 자신의 ID_A, $h(PIN)$을 서버 B의 공개키를 이용하여 암호화 하고 서버 B에게 전송한다.

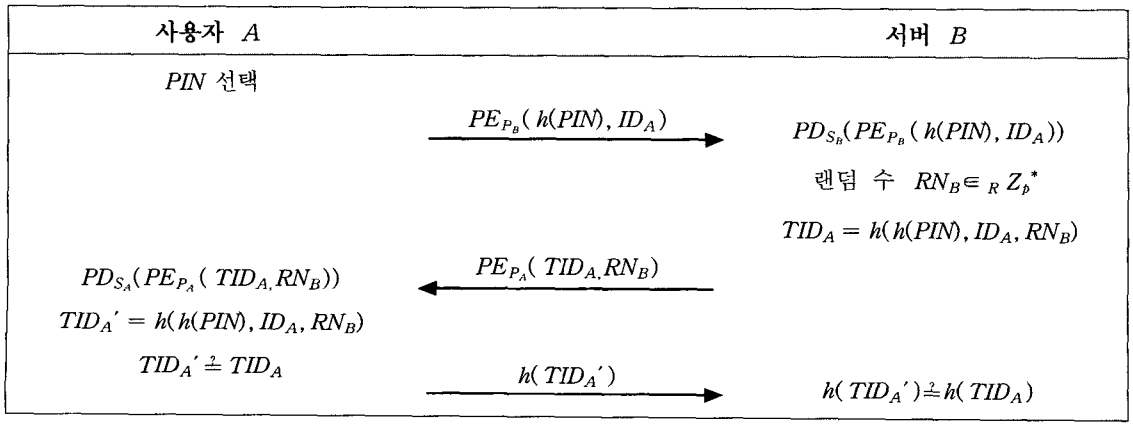
$$PE_{P_B}(h(PIN), ID_A)$$

② 서버 B는 자신의 개인키 S_B를 이용하여 사용자 A로부터 전송 받은 암호문을 복호화하고, 사용자 A에 대한 $h(PIN)$과 Z_p^*상에서 랜덤 수 RN_B을 선택하여 임시 ID인 TID_A 생성하고, RN_B와 $h(PIN)$를 자신의 데이터베이스에 저장한다. 서버 B는 OCSP 서버에게 사용자 A의 공개키 인증서의 유효성 검증을 요청하고 유효한 인증서로 검증되면, 자신이 생성한 TID_A와 랜덤 수 RN_B를 사용자 A의 공개키를 이용하여 암호화한 후 사용자 A에게 전송한다.

$$PD_{S_B}(PE_{P_B}(h(PIN), ID_A))$$
$$TID_A = h(h(PIN), ID_A, RN_B)$$
$$PE_{P_A}(TID_A, RN_B)$$

③ 사용자 A는 자신의 개인키 S_A를 이용하여 전송받은 암호문을 복호화하고, 랜덤 수 RN_B를 이용하여 임시 ID인 TID_A'를 생성하고 서버 B로부터 전송 받은 TID_A와 동일한지 비교·확인 한 후, TID_A'의 해쉬 값을 서버 B에게 전송한다.

$$PD_{S_A}(PE_{P_A}(TID_A, RN_B))$$



〔그림 2〕 등록 과정

$$TID_A' = h(h(PIN), ID_A, RN_B)$$
$$TID_A' \stackrel{?}{=} TID_A$$

④ 서버 B는 자신이 생성한 사용자 A의 TID_A의 해쉬 값과 사용자 A로부터 전송 받은 해쉬 값과 동일한지 비교·확인한다.

$$h(TID_A') \stackrel{?}{=} h(TID_A)$$

4.3 양방향 개체 인증을 제공하는 키 분배 프로토콜

기존의 이동 통신 환경에서의 키 분배 프로토콜 경우 사용자 A는 서버 B에 대한 인증을 수행할 수 있는 반면, 서버 B는 사용자 A에 대한 인증을 수행할 수 없다는 단점을 가진다. 즉, 일방향 개체 인증만을 제공한다. 따라서 비 대면이라는 특징을 갖는 무선 통신 환경의 특성 때문에 보다 안전한 양방향(상호) 개체 인증을 제공해야 하는 무선 통신 환경에서의 전자상거래 및 전자 금융 거래에서의 키 분배 프로토콜로는 적합하지 않다.([표 2]참고)

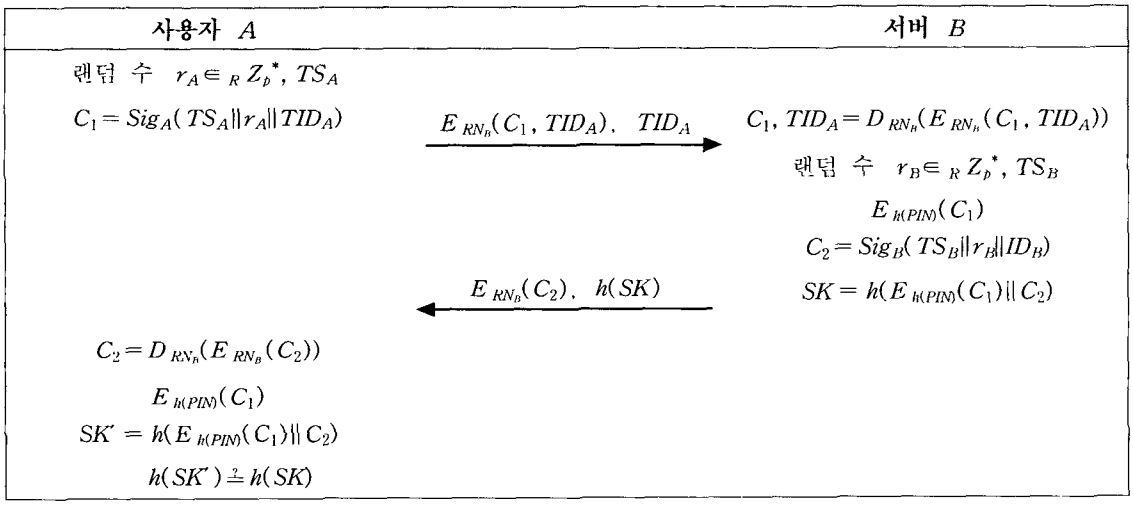
다라서 본 절에서는 양방향 개체 인증을 제공하는 키 분배 프로토콜을 제안한다. 제안하는 키 분배 프로토콜은 4.2절의 등록과정에서 서버 B가 선택한 랜덤 수 RN_B와 사용자 A가 생성한 $h(PIN)$을 이용하여 양방향 개체 인증을 제공하면서 통신 횟수는 기존의 프로토콜과 동일한 2회인 프로토콜을 제안한다. 또한 기존의 이산 대수의 어려움을 기반으로 하는 전자 서명 방식의 경우 서명 과정에서 대부분 1회의 모듈러 멱승 연산과 모듈러 멱승의 역원을 구하는 과정에서 계산 시간과 계산량이 증가된다. 따라서 본 프로토콜에서 사용하는 전자 서명 방식은 서명 과정에서의 계산량이 모듈러 멱승 1회이며 모듈러 멱승의 역원을 구하는 과정이 없어 계산 시간과 계산량을 개선한 Nyberg-Rueppel[13]의 전자 서명 방식을 이용한다.

본 논문에서 제안하는 양방향 개체 인증을 제공하는 키 분배 프로토콜의 동작 과정은 다음과 같다.

① 사용자 A는 Z_p^*상에서 랜덤 수 r_A을 선택하고 타임스탬프 TS_A를 생성한다. 그리고 랜덤수 r_A, 타임스탬프 TS_A와 TID_A를 연접하고 자신의 개인키를 이용해 서명한 C_1을 계산한다. C_1과 TID_A을 RN_B으로 암호화한 후, 서버 B에게 TID_A와 함께 전송한다.

$$C_1 = Sig_A(TS_A \| r_A \| TID_A)$$
$$E_{RN_B}(C_1, TID_A),\ TID_A$$

② 서버 B는 TID_A를 확인하고 RN_B를 이용하여 복호화하고 사용자 A임을 확인한다. 그리고 Z_p^*상에서 랜덤 수 r_B, 타임스탬프 TS_B 생성한다. 서버 B는 사용자 A로부터 전송 받은 C_1을 자신이 소유한 $h(PIN)$로 암호화하고 자신이 생성한



[그림 3] 양방향 개체 인증을 제공하는 키 분배 프로토콜

r_B, TS_B와 ID_B를 연접하여 자신의 개인키를 이용해 서명한 C_2를 계산한다. 그리고 $E_{h(PIN)}(C_1)$과 C_2를 이용해 세션키 SK를 계산한다.

$$C_1, TID_A = D_{RN_B}(E_{RN_B}(C_1, TID_A))$$
$$E_{h(PIN)}(C_1)$$
$$C_2 = Sig_B(TS_B \| r_B \| ID_B)$$
$$SK = h(E_{h(PIN)}(C_1) \| C_2)$$

③ 서버 B는 C_2를 RN_B로 암호화한 $E_{RN_B}(C_2)$와 세션키의 해쉬 값인 $h(SK)$를 사용자 A에게 전송한다.

$$E_{RN_B}(C_2),\ h(SK)$$

④ 사용자 A는 서버 B로부터 전송 받은 $E_{RN_B}(C_2)$을 RN_B를 이용하여 복호화하고, C_1을 자신이 소유한 $h(PIN)$을 이용하여 암호화한 $E_{h(PIN)}(C_1)$와 C_2를 이용해 세션키 SK'를 계산한다. 그리고 계산한 세션키 SK'를 해쉬하여 서버 B로부터 전송받은 $h(SK)$와 비교·확인한다.

$$C_2 = D_{RN_B}(E_{RN_B}(C_2))$$
$$E_{h(PIN)}(C_1)$$
$$SK' = h(E_{h(PIN)}(C_1) \| C_2)$$
$$h(SK') \stackrel{?}{=} h(SK)$$

4.4 키 분배 프로토콜의 보안 특성 분석

본 절에서는 3절에서 기술하고 있는 보안 요구 사항을 중심으로 제안하는 양방향 개체 인증을 제공하는 키 분배 프로토콜이 세션키 설정에 필요한 통신 횟수, 상호 개체 인증, 공개키 인증서의 상호 교환, 키 인증, key freshness, 키 확인(key confirmation), 익명성, 부인 봉쇄의 특징을 제공하는지에 대해 분석한다.

(1) 개체 인증 : 사용자 A와 서버 B는 모두에게 양방향 개체 인증을 제공한다. 사용자 A는 서버 B가 전송한 세션키의 해쉬 값인 $h(SK)$와 자신이 계산한 세션키의 해쉬 값인 $h(SK')$과 비교하여 서버 B임을 인증할 수 있다. 서버 B는 사용자 A가 자신과 공유한 비밀 정보 RN_B를 이용하여 암호화한 $E_{RN_B}(C_1, TID_A)$을 복호화함으로써 사용자 A임을 인증할 수 있다.

(2) 공개키 인증서의 교환 : 본 논문에서 제안하고 있는 프로토콜의 키 분배 과정에서는 등록 과정을 통해 공유한 비밀 정보를 이용하여 세션키를 설정하기 때문에 공개키 인증서를 교환할 필요가 없다.

(3) 키 인증 : 사용자 A는 명시적 키 인증을 할 수 있고, 서버 B는 묵시적 키 인증을 할 수 있다. 사용자 A의 경우, 세션키를 설정하는 과정에서 비밀 정보를 알고 있는 객체만이 세션키를 계산할 수 있으므로 서버 B로부터 전송 받은 세션키의 해쉬 값을 비교·확인함으로써 실제로 계산된 키임을 확인할 수 있다. 서버 B의 경우, 동일하게 세션키를 설정하는 과정에서 사용자 A가 자신과 공유한 비밀 정보를 이용해야만 세션키를 계산할 수 있다는 것을 확인할 수 있으므로 사용자 A만이 키를 계산할 수 있음을 확인할 수 있다.

(4) 키 동의 및 key freshness : 세션키를 설정하는 과정에서 세션키는 사용자 A와 서버 B가 각각 생성한 랜덤 수 r_A와 r_B를 이용하여 계산하게되므로 키 동의와 key freshness를 보장한다.

(5) 익명성 : 사용자 A와 서버 B의 익명성이 보장된다. 사용자 A의 경우, 임시 ID인 TID_A를 사용하므로 익명성을 보장받을 수 있다. 서버 B의 경우, 기존의 프로토콜과 달리 공개키 인증서를 전송하지 않으므로 익명성을 보장받을 수 있다.

(6) 부인 봉쇄 : 부인 봉쇄는 사용자 A와 서버 B 모두에게 제공된다. 사용자 A와 서버 B가 세션키를 설정하는 과정에서 각각 자신의 개인키를 이용하여 서명한 C_1, C_2와 타임스 탬프 TS_A, TS_B가 사용됨으로써 후에 분쟁이 발생한다 하더라도 서명을 검증함으로써 분쟁을 해결할 수 있다.

본 논문에서 제안하는 양방향 개체 인증을 제공하는 키 분배 프로토콜은 통신 횟수가 2회이며 사용자와 서버간의 양방향 개체 인증을 제공한다. 또한 키 분배 수행과정에서 공개키 인증서를 교환할 필요가 없으며 일방향 키 확인, 양방향 key freshness, 양방향 익명서 및 양방향 부인 봉쇄를 제공한다.

위의 보안 요구 사항의 분석 내용을 바탕으로 양방향 개체 인증을 제공하는 키 분배 프로토콜의 보안 요구 사항에 대한 분석 결과는 아래 [표 1]과 같다.

〔표 1〕 양방향 개체 인증을 제공하는 키 분배 프로토콜의 보안 요구 사항 분석

	키 분배 프로토콜	
통신 횟수	2회	
개체 인증	양방향	
공개키 인증서 교환	교환하지 않음	
키 인증	A	명시적 키 인증
	B	묵시적 키 인증
키 확인	일방향(사용자 A)	
키 구축	키 동의	
Key freshness	양방향	
익명성	양방향	
부인 봉쇄	양방향	

V. 안전성 분석 및 기존 키 분배 프로토콜과의 비교

5.1 공격자 모델

본 논문에서 제안한 키 분배 프로토콜의 안전성을 증명하기 위해서는 키 분배 프로토콜에 대한 공격자 모델을 정의하여야 한다. 암호 프로토콜에서의 공격자 모델은 주로 수동적 공격자(passive attacker)와 능동적 공격자(active attacker)로 나뉘어 지며, 각 공격자의 정의는 다음과 같다.[13]

- 수동적 공격자 : 합법적인 주체간의 암호 프로토콜 수행 시 실제 프로토콜에 참여하지 않으며 단지 두 주체 사이의 통신 내용을 도청함으로써 공격을 수행하는 공격자
- 능동적 공격자 : 단순히 참가자들의 통신 내용을 도청하는 것뿐만 아니라 전송되는 메시지를 위·변조하거나 새로운 메시지를 삽입하는 등 실제 통신에 참여하는 강력한 공격자

본 논문에서 고려할 능동적 공격자 모델은 Active Impersonation 공격자, Forward Secrecy에 대한 공격자, Key-Compromise Impersonation 공격자, Known Key Security 공격자, Off-line dictionary attack에 대한 공격자이며 각각에 대한 정의는 다음과 같다.

[정의 1] Active Impersonation attack
공격자가 자신을 임의의 다른 사용자로 위장하여 프로토콜에 참여하고, 정당한 사용자 U와 키 분배를 성공적으로 수행하는 경우에 active impersonation이 가능하다고 한다.

[정의 2] Forward Secrecy
사용자 U와 V의 개인키가 노출되더라도, 공격자가 두 사용자 사이에 설정된 과거 세션키를 계산할 수 없는 경우에 Forward Secrecy를 만족한다고 한다. Forward Secrecy는 노출되는 사용자의 비밀키에 따라 다음과 같이 분류할 수 있다.

- Half Forward Secrecy : 한 사용자의 개인키가 노출된 경우에만 세션키가 안전
- Full Forward Secrecy : 두 사용자의 개인키가 모두 노출된 경우에도 세션키가 안전

[정의 3] Key-Compromise Impersonation attack
사용자 U의 개인키가 노출되었을 때, 공격자 E가 누구에게나 사용자 U로 위장할 수 있고 사용자 U에게 임의의 사용자 V로 위장할 수 있을 때 Key-Compromise Impersonation이 가능하다고 한다. 그러나 공격자 E가 누구에게나 사용자 U로 위장할 수 있지만 사용자 U에게 임의의 다른 사용자로 위장할 수는 없는 경우에는 키 분배 프로토콜이 Compromise Impersonation resilience 특성을 갖는다고 한다.

[정의 4] Known Key Security
두 사용자 U, V 사이의 과거 세션키가 노출되더라도 현재 세션키의 안전성에는 아무런 영향을 미치지 않는 경우에 Known Key Security를 만족한다고 한다. Known Key Security에 대한 공격은 다음과 같이 두 가지로 분류할 수 있다.

- KKP(Known Key Passive) 공격 : 과거의 세션키와 전송 정보 및 현재 세션키 전송 정보를 이용하여 현재의 세션키를 획득하려는 공격 방법
- KKI(Known Key Impersonation) 공격 : 세션에 직접 참여하여 과거의 세션키와 전송 정보 그리고 현재 세션의 전송 정보를 이용하여 사용자 U에게 사용자 V로 위장하여 세션키를 설정하려는 공격 방법

[정의 5] Off-line dictionary attack
공격자는 과거에 있었던 정당한 사용자들간의 통신을 기록하고, 사전(dictionary)으로부터 패스워드를

검사하면서 기록된 통신에 사용된 패스워드와 일치하는 패스워드를 찾는 공격 방법이다.

5.2 안전성 분석 결과

5.2.1 수동적 공격자에 대한 안전성

본 논문에서 제안하고 있는 키 분배 프로토콜은 기본적으로 그 안전성이 이산 대수 문제에 기반하므로 수동적 공격자가 공개 정보와 전송 정보를 이용하여 세션키를 구하는 어려움은 이산 대수 문제를 푸는 어려움과 동일하다.

5.2.2 능동적 공격자에 대한 안전성

▪ Active Impersonation에 대한 안전성

공격자 E가 세션을 시작한 경우, 공격자 E가 사용자 A로 위장하여 서버 B와 세션키를 설정하는 것은 수동적 공격자의 어려움과 동일하다. 공격자 E가 서버 B와 세션을 시작하는 경우에는 정당한 사용자의 개인키, 개인 식별 정보의 해쉬 값인 $h(PIN)$와 비밀 정보 RN_B를 알지 못하면 정당한 세션키를 생성할 수 없으므로 위장이 불가능하다. 또한, 공격자 E가 서버 B로 위장하여 사용자 A와 세션키를 설정하는 경우, 세션키를 설정하는 과정에서 공격자 E는 정당한 서버의 개인키, $h(PIN)$와 RN_B를 알지 못하기 때문에 동일한 세션키를 계산하지 못한다. 그리고 공격자 E가 사용자 A에게 정당하지 않은 전송 정보 $h(SK'')$를 전송하고 이를 사용자 A가 확인하는 과정에서 공격 여부를 알 수 있다.

사용자 A가 서버 B로 위장하여 다른 임의의 사용자 U와 세션을 시작하는 경우, 사용자 A는 임의의 사용자 U와 서버 B사이의 공유된 개인 식별 번호의 해쉬 값과 RN_B'와 서버 B의 개인키를 알 수 없기 때문에 정당한 세션키를 생성할 수 없다. 또한 서버 B가 사용자 A로 위장하여 다른 임의의 서버 S와 세션을 설정하려는 경우 사용자 A의 개인키와 사용자 A가 서버 S가 공유한 RN_S'를 알 수 없기 때문에 정당한 세션키를 생성할 수 없으며 이는 이산 대수 문제를 푸는 어려움과 동일하다.

▪ Forward Secrecy에 대한 안전성

사용자 A의 개인키가 노출된 경우에 공격자 E는 $h(PIN)$과 RN_B를 알 수 없으므로 정당한 과거의 세션키를 구할 수 없고, 서버 B의 개인키가 노출된 경우에도 사용자 A의 경우와 동일하게 공격자 E는 $h(PIN)$과 RN_B를 알 수 없으므로 과거의 세션키의 안전성에는 아무런 영향이 없다. 따라서 Half Forward Secrecy를 제공한다.

또한 사용자 A와 서버 B의 개인키가 모두 노출된다 하더라도 $h(PIN)$와 RN_B를 알 수 없음으로 현재의 세션키를 구하거나 과거의 세션키를 구하는 어려움은 수동적 공격자와 동일하므로 Full Forward Secrecy를 제공한다.

▪ Key-Compromise Impersonation에 대한 안전성

사용자 A의 개인키 S_A가 노출된 경우에도 공격자 E는 $h(PIN)$과 RN_B를 알 수 없기 때문에 서버 B에게 사용자 A로 위장할 수 없다. 또한 공격자 E는 사용자 A에게 다른 임의의 서버로 위장할 수 없다. 공격자 E가 사용자 A에게 임의의 다른 서버로 위장하기 위해서는 $h(PIN)$과 RN_B을 알아야 한다. 즉, 공격자 E가 사용자 A에게 자신이 선택한 전송 정보 $E_{RN_B'}(C_2')$를 계산하여 전송한다 하더라도 정당한 세션키의 해쉬 값을 구할 수 없으므로 다른 임의의 서버로 위장하는 것은 불가능하다.

또한 공격자 E가 서버 B에게 임의의 사용자 U로 위장하는 경우에도 공격자 E는 사용자 A의 개인키의 소유와 관계없이 사용자 U의 개인키와 사용자 U와 서버 B사이에 공유한 비밀 정보 RN_B'와 $h(PIN')$을 알아야 하기 때문에 임의의 사용자로 위장할 수 없어 Compromise Impersonation resilience 특성을 갖게 된다.

▪ Known Key attack에 대한 안전성

세션키를 설정하기 위해서 매 세션마다 서버 B가 선택한 랜덤 수 r_B, 사용자 A가 선택한 랜덤 수 r_A와 해쉬 함수가 사용되므로 이전 세션의 전송 정보와 세션키가 노출되더라도 현재의 세션키를 구하는데 아무런 도움을 주지 못한다. 따라서 KKP 공격자의 어려움은 이전 세션에 대한 아무런 정보가 주어지지 않은 수동적 공격자의 어려움과 동일하다.

세션키 생성 과정에서 이전 세션의 전송 정보와 세션키를 획득한 공격자 E가 사용자 A 또는 서버 B로 위장하기 위해서 이전의 세션의 전송 정보를 재전송하는 하는 경우, 세션에 참여한 상대방이 매 세션마다 선택한 랜덤 수로 인해 정당한 세션키를 생성할 수 없다. 따라서 재전송을 이용한 KKI 공격에 대해 안전하다.

• Off-line dictionary attack에 대한 안전성

공격자 E가 합법적으로 위장할 수 있는 세션을 수행하기 위해 정당한 사용자들간의 전송 정보를 이용하여 $h(PIN)$과 RN_B를 구하려는 경우, 사용자 A에 의해 전달되는 전송 정보 $E_{RN_B}(C_1, TID_A)$, TID_A를 가지고 off-line dictionary attack에 성공하여 RN_B을 찾았다 하더라도 키 분배 프로토콜에서 RN_B은 서버와 사용자간의 상호간의 개체 인증을 수행하기 위한 비밀 정보일 뿐 세션키를 계산하는데는 아무런 영향을 주지 않는다. 그리고 실제 세션키를 계산하기 위해서는 세션키 계산에 필요한 비밀 정보 $h(PIN)$을 알아야 하므로 정당한 세션키를 계산할 수 없다. 서버 B에 의해 사용자 A에게 전달되는 전송 정보 $E_{RN_B}(C_2)$, $h(SK)$이용하여 공격을 수행하는 경우 실제 세션키 계산에 필요한 비밀 정보 $h(PIN)$를 알 수 없고, 전송 정보 $h(SK)$에 대한 off-line dictionary attack을 수행하여 세션키을 구하려는 경우의 어려움은 해쉬 함수의 역함수를 구하는 것과 동일하므로 공격을 수행하는 것은 어렵다. 또한 세션키를 계산하기 위해 필요한 비밀 정보인 $h(PIN)$는 사용자 A와 서버 B사이의 전송 정보에서 전혀 노출되지 않으므로 off-line dictionary attack을 할 수 없다.

5.3 기존 이동 통신 환경의 키 분배 프로토콜과의 비교

5.3.1 기존 키 분배 프로토콜과 보안 요구 사항 비교

본 절에서는 기존에 발표된 이동 통신 환경에서의 키 분배 프로토콜들과 본 논문에서 제안하는 프로토콜의 보안 요구 사항을 비교한다. 기존의 프로토콜은 BCY 프로토콜,[2] PACS 프로토콜,[3] 1.5-move 프로토콜,[4] LM 프로토콜[5]있으며, 보안 요구 사항 중 상호 개체 인증, 공개키 인증서의 상호 교환, 키 동의, 키 인증, key freshness, 익명성, 부인 봉쇄를 비교・분석한다.[6]

BCY 프로토콜은 관용 암호 방식과 공개키 암호 방식을 조합한 형태로 많은 개선을 통해 Varadharajan와 Mu에 의해 개선된 형태의 표준 프로토콜과 유사한 구조를 가지는 프로토콜로 제안되었다.[7] 공개키 인증서의 상호 교환, 키 동의, 사용자의 익명성만 제공하고 있어 사용상의 제약이 많다. 또한 임의의 객체가 사용자나 서버의 역할을 할 수 있다는 문제점을 가져 active impersonation attack이 가능하여 안전성에 문제를 갖는다.

PACS 프로토콜은 공개키 인증 기반 프로토콜로 사용자가 자신의 서명을 이용해 세션키를 생성하고 서버의 공개키를 이용해 전송하는 키 전송 방식이다. 서버는 사용자로부터 전송된 전송 정보를 복호화 과정을 통해 세션키를 얻고 사용자의 공개키 인증서와 서명을 검증하게 된다. 이 과정에서 서버측의 연산 부담이 증가하게 된다. 또한 랜덤 수를 사용하지 않기 때문에 key freshness를 제공하지 않으며, 사용자는 자신이 계산한 세션키를 이용해 자신의 공개키 인증서를 암호화하여 서버에게 전송하므로 사용자에 대한 익명성을 보장하게 된다.

1.5-move 프로토콜은 서버의 개인키를 알고 있는 공격자가 사용자의 개인키를 모르더라도 정당한 사용자로 위장할 수 있다는 심각한 문제점을 가지고 있으며, PACS와 동일하게 키 전송 형태를 가지며, 부인 봉쇄를 전혀 제공하지 않는다.

LM 프로토콜은 signcryption[8]을 이용하여 사용자

〔표 2〕 기존의 프로토콜과 조건 비교・분석

	BCY	PACS	1.5-move	LM	제안하는 프로토콜
통신 횟수	2	2	2	2	2
상호 개체 인증	제공안함	일방향	일방향	일방향	양방향
공개키 인증서 상호 교환	양방향	양방향	양방향	양방향	교환하지 않음
키 구축	키 동의	키 전송	키 전송	키 동의	키 동의
키 인증	사용자 : 묵시적 서버 : 묵시적	사용자 : 명시적 서버 : 묵시적	사용자 : 명시적 서버 : 묵시적	사용자 : 명시적 서버 : 묵시적	사용자 : 명시적 서버 : 묵시적
key freshness	제공안함	제공안함	제공안함	양방향	양방향
익명성	일방향	일방향	일방향	일방향	양방향
부인 봉쇄	제공안함	일방향	제공안함	일방향	양방향

〔표 3〕 기존의 프로토콜과 계산량 비교

	BCY		PACS		1.5-move		LM		제안하는 프로토콜	
	사용자	서버	사용자	서버	사용자	서버	사용자	서버	사용자	서버
공개키 암/복호화			1	1						
세션키 생성	1	1	1		2		1	1		
서명			1	2			1		1	1
기타	1	1			1	2		1		
합계	2	2	3	3	3	2	2	2	1	1

의 익명성을 제공한다. 사용자와 서버는 초기에 전송 측의 메시지를 보호하기 위해 암호화 키를 설정하고 복호화된 메시지와 수신 측의 개인키를 사용하여 복호화 키를 생성하는 형태를 가지고 있다. 수정된 *signcryption* 기법의 경우 처음 제안된 기법보다는 계산량이 다소 증가하지만 기존의 서명 기법과 암호 기법을 각각 사용하는 경우보다는 계산량이 적고 사용자의 익명성을 제공한다.

본 논문에서 제안한 키 분배 프로토콜에서의 개체 인증은 세션키를 생성할 때 사용자 A와 서버 B사이에 공유된 비밀 정보 RN_B와 $h(PIN)$를 이용하게 되므로 양방향 개체 인증을 제공한다. 기존의 프로토콜의 경우 사용자와 서버간의 공개키 인증서를 상호 교환하게 되는데 제안한 키 분배 프로토콜의 경우 상호간 공개키 인증서를 교환할 필요가 없다. 인증서를 검증하는 과정은 등록 과정에서만 수행되고 그 또한 OCSP 서버를 이용하여 공개키 인증서를 검증할 수 있으므로 사용자와 서버의 연산 부담을 줄일 수 있다. 세션키 생성 시 사용자와 서버가 각각 선택한 랜덤 수를 사용하기 때문에 키 동의 방식을 제공하며, 세션키 생성시 키 인증의 경우 사용자에게는 명시적 키 인증, 서버에게는 묵시적 키 인증을 제공한다. 사용자와 서버간에 교환되는 전송 정보를 임의의 공격자가 도청하더라도 사용자와 서버의 양방향 익명성을 보장하며, 세션키 생성시 상대방의 서명을 사용하게 되므로 양방향 부인 봉쇄 특성을 제공한다. 또한, 키 확인 과정은 서버가 생성한 세션키의 해쉬 값을 통해 사용자에게 제공된다.

이를 보안 요구 사항으로 분류하여 기존의 이동 통신 환경의 키 분배 프로토콜과 비교하면, [표 2]와 같다.[6]

5.3.2 계산량 비교

무선 통신 환경의 단말기의 경우, 유선 통신 환경의 단말기에 비해 연산 능력이 적다. 따라서 무선 통신 환경에서의 키 분배 프로토콜을 설계하려면 무선 단말기의 계산량을 최소화하여야 한다. 키 분배 프로토콜을 수행할 때 해쉬 연산, 대칭 암호 암/복호화 연산 등의 경우 연산에 필요한 시간은 적기 때문에 연산에 필요한 대부분의 시간은 주로 모듈러 멱승이 필요한 서명 생성/검증, 공개키의 암/복호화, 세션키 계산에 필요한 모듈러 멱승 등의 연산에 주로 소요된다. 따라서 본 논문에서는 모듈러 멱승의 횟수를 기준으로 기존의 키 분배 프로토콜과 계산량을 비교한다.

기존의 키 분배 프로토콜의 모듈러 멱승 횟수를 살펴보면 다음과 같다. BCY 프로토콜은 모듈러 멱승의 횟수가 각각 사용자 2번, 서버 2번이고 PACS 프로토콜은 사용자 3번, 서버 3번이다. 또한 1.5-move 프로코톨은 사용자 3번, 서버 2번이고, LM 프로토콜은 사용자 2번, 서버 2번이다.

하지만 본 논문에서 제안하는 키 분배 프로토콜을 수행하기 위해서는 사용자 1번, 서버 1번의 모듈러 멱승을 필요로 때문에 사용자와 서버의 연산 부담을 줄였으며, 낮은 연산 능력을 갖는 무선 단말기 등의 사용에 적합하다.

[표 3]은 기존의 키 분배 프로토콜과 계산량을 비교 분석한 것이다.

Ⅵ. 결 론

무선 통신 환경은 유선 통신 환경과 비교했을 때 많은 면에서 상이한 특성을 갖는다. 사용자의 이동이 빈번하게 일어나게 되므로 인증이 실시간으로 이루

어져야 하며 유선 통신 환경과는 달리 연산 능력이 비대칭적이다. 또한, 무선 통신 환경의 사용자는 유선 통신 환경에 비해 낮은 연산 능력을 갖기 때문에 사용자 측의 연산 부담을 줄여야 한다. 3세대 이동통신의 무선 단말기는 성능이 우수한 CPU와 메모리를 사용하게 되므로 연산 속도와 처리 능력이 기존의 두선 단말기보다 뛰어나지만 아직 유선 통신 환경의 연산 능력과 메모리에 미치지 못한다. 이를 보완하기 위해서 암호 연산 전용 co-processor를 탑재한 무선 단말기를 활용하는 경우 다음과 같은 장점을 갖게 된다. 첫째, 무선 단말기의 암호 연산 부담을 암호 전용 연산 co-processor에 부가함으로써 무선 단말기의 연산 부담을 줄인다. 둘째, 무선 단말기를 암호 연산에 이용하는 경우 연산 과정에서 개인키, 인증서, 개인 비밀 정보 등이 저장 과정 또는 연산 과정에 누출될 위험을 IC 칩의 불가침 메모리 영역을 이용함으로써 예방할 수 있다.

본 논문에서 제안하는 프로토콜은 무선 통신 환경에서의 단말기의 제한적인 연산 능력을 고려하여 모듈러 멱승 연산을 수행하는 암호 전용 co-processor를 탑재한 IC 칩을 무선 단말기에 내장함으로써 모듈러 멱승 연산을 무선 단말기에서 수행하는 것이 아니라 IC 칩에서 수행하여 무선 단말기의 연산 부담을 감소시켰다. 또한, 키 분배 프로토콜을 수행 전 사용자가 사전 계산(precomputation)이 가능한 $E_{h(PIN)}(C_1)$을 사전에 계산함으로서 휴대폰, PDA 등과 같은 무선 단말기에 더욱 효율적으로 사용될 수 있다.

기존의 프로토콜에서 제공하지 않은 보안 요구 사항을 만족하여 높은 수준의 보안이 요구되는 무선 전자상거래, 무선 전자결제 등에 적합하다.

하지만 본 논문에서 제안한 키 분배 프로토콜은 이산 대수 문제를 기반으로 하고 있어 키 사이즈가 크기 때문에 대역폭 사용의 효율적 측면과 서버가 사용자에 대한 비밀 정보를 저장해야 한다는 문제점을 가지게 된다. 키 사이즈(key size)의 경우 타원 곡선 암호시스템(Elliptic curve cryptosystem)과 비교하여, 이산 대수 문제의 어려움을 기반으로 하는 암호시스템의 경우 키의 길이가 1024비트, 타원 곡선 암호시스템의 경우 160비트의 길이의 키를 사용한다. 동일한 환경에서 각각의 암호시스템의 서명 연산 속도를 비교해 보면 이산 대수의 경우 1.931msec, 타원 곡선의 경우 0.603mesc이 소요되어 약 3배정도 빠르다는 것을 알 수 있다.[14] 이를 바탕으로 상대적으로 작은 키 사이즈를 사용하는 타원 곡선 암호시스템으로의 추가적인 연구 과정과 비밀 정보의 축소 등을 통해 보다 효율적인 프로토콜을 고려하여야 할 것이다.

참 고 문 헌

[1] G. Horn, K. M. Martin, and C. J. Michell, "Authentication Protocols for Mobile Network Environment Value-Added Service", draft, available at http://isg.rhbnc.ac.uk/cjm/Chris_Mitchell.htm.

[2] M. J. Beller, L. F. Chang, and Y. Yacobi, "Privacy and Authentication on a Portable Communications System", *Proceeding of GLOBECOM'91*, pp. 1922 ~1927, IEEE Press, 1991.

[3] JTC, "PACS(Personal Access Communications System) Air Interface Standard", J-STD-014, 1995. 6.

[4] Y. Zheng, "An Authentication and Security Protocol for Mobile Computing", proceedings of IFIP, pp.249~257, 1996. 9.

[5] KookHeui Lee and SangJae Moon, "AKA Protocols for Mobile Commu- nications", Australasian Conference, ACISP 2000, LNCS 1841, pp.400~411, Brisbane, Australia, 2000. 7.

[6] 최영근, 김순자, "이동 시스템에서의 효율적인 인증 및 키 교환 프로토콜", 통신정보보호논문지, pp.73~82, 2001. 4.

[7] V. Varadharajan, Y. Mu, "On the Design of Security Protocols for Mobile Communications", *Australasian Conference, ACISP'96 Conference*, pp.134 ~145, Springer-Verlag, 1996.

[8] Y. Zheng, "Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption)", Advances in Cryptology, Proceedings of CRYPTO '97, pp.165~179, 1997. 8.

[9] S. Vellovin and M. Merrit, "Encrypted key exchange: password based protocols secure against dictionary attacks", In proceedings of the Symposium of Security and Privacy, pp.72~84, IEEE, 1992.

[10] T, Wu, "Secure Remote Password Protocol", In Proceedings of the Internet Society Network and Distributed System Security Symposium, pp.97~111, 1998.

[11] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks", Eurocrypt'00, LNCSl Vol, 1807, pp.139

~155, Springer-Verlag 2000.

[12] Jaeseung Go and Kwangjo Kim, "Wireless Authentication Protocols Preserving User Anonymity", SCIS-2001, vol.1/2 pp.159~164, Jan. 23~26, 2001.

[13] K. Nyberg and R. A. Rueppel, "Message recovery for signature scheme based on the discrete logarithm problem". Eurocrypt'94 Proceeding, Springer-Verlag, 1995.

[14] 오수현, 이승우, 심경아, 양형규, 원동호, "타원곡선에 기반한 표준 키 분배 프로토콜의 안전성 분석 및 응용 분야에 관한 연구", 정보보호학회 논문지, pp.103~118, 2002. 6.

[15] 이동훈, 황효선, 임채훈, "타원곡선 암호의 기초와 응용", (주)퓨쳐시스템 암호체계센터, Technical Report, 2001. 8. 1.

[16] 곽진, 이승우, 조석향, 홍순좌, 원동호, "시간정보를 이용한 인증서 상태 검증정보 제공에 관한 연구", 한국정보처리학회 춘계학술발표 논문집 제9권 1호 pp.829~832, 2002. 4.12.

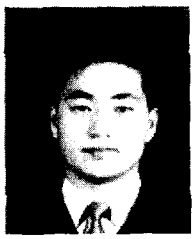
[17] 곽진, 이승우, 조석향, 원동호 "온라인 인증서 상태 검증 프로토콜(OCSP)의 최근 연구 동향에 관한 분석", 한국정보보호학회 논문지 제12권 2호 pp.50~61, 2002. 4.

〈著者紹介〉

안 기 범 (Gi-Bum An) 학생회원
2002년 2월 : 성균관대학교 시스템경영공학부 졸업(공학사)
2002년 3월~현재 : 성균관대학교 정보통신공학부 석사 과정

김 수 진 (Soo-Jin Kim) 학생회원
2002년 2월 : 성균관대학교 자연과학부 수학과 졸업(이학사)
2002년 3월~현재 : 성균관대학교 정보통신공학부 석사 과정

한 종 수 (Jong-Su Han) 학생회원
2002년 2월 : 성균관대학교 정보공학과 졸업(공학사)
2002년 3월~현재 : 성균관대학교 정보통신공학부 석사 과정



이 승 우 (Seung-Woo Lee) 학생회원
2001년 2월 : 강남대학교 전자계산학과 졸업(공학사)
2003년 3월 : 성균관대학교 정보통신공학부 졸업(공학석사)
2003년 3월~현재 : 성균관대학교 정보통신공학부 박사 과정

원 동 호 (Dongho Won) 정회원
성균관대학교 전자공학과 졸업(학사, 석사, 박사)
1978년~1980년 : 한국전자통신연구원 전임연구원
1985년~1986년 : 일본 동경공업대 객원연구원
1988년~1999년 : 성균관대학교 교학처장, 전기전자 및 컴퓨터공학부장, 정보통신대학 원장, 정보통신기술연구소장
1996년~1998년 : 국무총리실 정보화추진위원회 자문위원
2002년~2003년 : 한국정보보호학회장
현재 : 성균관대학교 정보통신공학부 교수, 정통부 지정 정보보호인증기술연구센터장, 성균관대학교 연구처장