이동통신 환경에서의 효율적인 상호인증 및 세션키 공유 프로토콜

이 승원*, 홍 성민**, 윤 현수**, 조 유근*

* 151-742 서울 특별시 관악구 신림동 산 56-1
서울대학교 컴퓨터공학과
** 305-701 대전 광역시 유성구 구성동 373-1
한국과학기술원 전산학과

Efficient Key Establishment Protocol in Mobile Communication

Seungwon Lee(*), Seong-Min Hong(**)
Hyunsoo Yoon(**), Yookun Cho(*)

* Department of Computer Engineering, SNU
San 56-1 Shilim-Dong Kwanak-Ku, Seoul, 151-742, KOREA
E-mail: {leesw,cho}@ssrnet.snu.ac.kr
** Department of Computer Science, KAIST
373-1 Gusong-Dong Yousong-Gu Taejon, 305-701, KOREA
E-mail: {smhong,hyoon}@camars.kaist.ac.kr

요약 : 이동통신 환경에서, 이동단말기와 기지국 사이에 안전한 통신을 제공하기 위한 많은 프로토콜들이 제안되어왔다. 그러나 이동단말기의 부족한 계산능력 때문에 공개키를 이용한 프로토콜은 많은 장점에도 불구하고 그 실용성을 검증받기 힘들었다. 본 논문은 이동단말기의 부족한 계산능력으로도 공개키를 이용해 상호인증 및 세션키 공유를 효율적으로 수행할 수 있는 프로토콜을 제안한다. 제안된 프로토콜은 최근 제안된 SASC 프로토콜을 적용하여 이동단말기의 부족한 계산능력을 보충하였다.

제 1 절 서론

이동통신(mobile communication)은 대기를 전달매체로 사용하기 때문에 기존의 유선통신과는 달리 별도의 회선을 설치할 필요가 없어서 설치비용이 적다. 그리고 이동통신은 사용자가 공간적 제약을 받지 않고 사용할 수 있다는 편리함과 유동성을 제공한다. 이동통신이 제공하는 이러한 경제성과 편리함 때문에 이동통신은 차세대 통신 환경의 주역으로 주목을 받고 있는 통신 시스템이다. 그러나, 이동통신이 차세대의 통신 주역으로 성장하기 위해서는 몇가지 해결해야 할 문제들이 있다. 그러한 문제들중의 하나가 기지국 (base station)과 이동단말기 (mobile station) 사이에서 통신내용의 기밀성을 유지하는 것과 기지국과 이동단말기 사이에 상호인증을 제공하는 문제이다.

이동통신 환경에서 데이타의 기밀성과 상호인증을 제공하기 위한 프로토콜은

공개키 암호알고리즘을 이용하는 방식과 대칭키 암호알고리즘을 이용하는 방식으로 나눌 수 있다. 대칭키 암호알고리즘은 공개키 암호알고리즘에 비해서 처리 속도가 월등히 빠르다는 장점을 가지고 있다. 그러나, 대칭키 암호알고리즘을 이용하여 기밀성과 인증을 제공하는 프로토콜을 구현하기 위해서는 각 이동단말기가 자신의 비밀 정보를 가지고 있어야 한다는 제한을 가지게 된다. 이러한 제한은 이동단말기가 한 영역(roam)에서 다른 영역으로 이동할 경우 이동단말기의 비밀 정보 역시 현재 기지국에서 다른 영역의 기지국으로 이동해야 한다는 것을 의미한다. 이러한 상황은 각 기지국 사이에 이동단말기의 비밀정보를 안전하게 전달할 수 있는 방법과 기지국이 자신의 영역을 떠난 이동단말기의 비밀 정보를 어떻게 없애는가 하는 문제를 야기시킨다. 게다가 현재 구현 가능한 부인 방지(non-repudiation)서비스는 오직 공개키 암호알고리즘을 이용해서만 구현할 수 있고 익명성(anonymity)도 공개키 암호알고리즘을 이용하여 구현하는 것이 대칭키 암호알고리즘을 이용하여 구현하는 것보다 훨씬 쉽다. 이동통신 환경에서 기밀성과 상호인증을 제공하는 프로토콜들의 대부분은 상호 인증과 키 공유를 위해서는 공개키 암호알고리즘을 이용하고 데이타를 전달할 때 기밀성을 유지하기 위해서는 대칭키 알고리즘을 이용한다 [1].

이동통신 환경에서 연구된 상호 인증과 키 공유 프로토콜들이 사용하는 공개키 암호알고리즘들은 MSR(Modulo Square Root), RSA, ElGamal, Diffie-Hellman 키 교환 프로토콜 등이 있다 [2, 3, 4]. MSR알고리즘은 Rabin에 의해서 제안된 공개키 암호알고리즘으로 모듈라 제곱승과 모듈라 제곱근을 이용하여 다른 공개키 암호 시스템에 비해 좋은 성능을 보이는 알고리즘이다 [5]. 그러나 MSR을 이용한 키공유 프로토콜은 보안상 여러가지 헛점들이 있는 것으로 알려졌고, 이러한 단점을 보안하기 위해 Diffie-Hellman 키교환 프로토콜과 합쳐진 프로토콜이 제안되었다. Diffie-Hellman 키교환 프로토콜은 키 공유의 목적으로 만들어진 알고리즘이기 때문에 상호인증을 하기 위해서는, MSR과 같은 다른 공개키 암호알고리즘과 병행해서 사용해야 한다. ElGamal 암호알고리즘은 서명을 확인하는데 다른 연산에 비해 많은 시간이 걸리고 암호알고리즘과 서명을 생성하는 알고리즘이 다르다는 단점을 가지고 있다. 특히 암호알고리즘과 전자 서명 생성 알고리즘이 다르다는 점은 특정 목적의 하드웨어(special purpose hardware)를 만들 경우 RSA에 비해 더 많은 회로가 요구되고 소프트 웨어로 구현할 경우에도 더 많은 용량의 기억장치를 요구하게 된다. RSA알고리즘은 암호화하는 알고리즘과 전자 서명을 만드는 알고리즘이 동일하고 연산이 단순하기 때문에 하드웨어로 구현하기가 용이하다. 게다가, RSA알고리즘은 전세계에서 널리 사용되고 있는 알고리즘으로 대부분의 보안 프로토콜과 소프트웨어가 지원하므로 확장이 매우 용이한 암호알고리즘이다. 그러나, RSA도 이동단말기의 계산량이 너무 많다는 단점때문에 현재 기술로는 이동통신 환경에서 상호 인증과 키 공유를 위한 실질적으로 프로토콜에 이용하기가 힘들었다 [1].

본 논문에서는 기존 프로토콜에서는 이동단말기의 과도한 계산량 때문에 이

동통신 환경에서 이용할 수 없었던 RSA를 효율적으로 이용하는 상호인증과 키 공유 프로토콜을 제안한다. 제안 프로토콜은 이동단말기의 계산량을 줄이기 위해 서버를 이용하는 비밀 계산(server aided secret computation : SASC) 기법을 이용한다. 제안 프로토콜이 사용하는 SASC 기법은 기존의 SASC 기법에 비해 훨씬 적은 통신 비용과 서버의 계산량을 줄인 최근에 새롭게 제안된 SASC 기법이다 [6]. 제안 프로토콜은 기존의 프로토콜에 비해 8 20배 이상의 속도 향상을 기대할 수 있고 이동통신 환경에 잘 적용되므로 공개키 방식의 장점을 이동통신 환경에 적합하다.

본 논문의 구성은 다음과 같다. 2절에서는 이동통신 환경에서 키 공유 프로토콜들을 설명한다. 3절에서는 본 논문에서 제안된 키 공유 프로토콜과 SASC기법에 대해 기술하고, 4절에서는 제안된 프로토콜의 성능을 기존의 프로토콜들과 비교 분석한다. 마지막 5절에서 결론을 맺는다.

제 2 절 관련 연구

본 절에서는 기존 연구에서 제안된 이동통신 환경에서의 키 공유 프로토콜들 중 공개키 암호알고리즘과 대칭키 암호알고리즘을 병행해서 사용하는 대표적인 프로토콜들에 대해 설명한다. 이 프로토콜들은 믿을 수 있는 제 삼자가 발행한 인증서를 통해 상대방의 공개키를 확인할 수 있다고 가정한다. MSR + DH 프로토콜 [7]은 기지국과 이동단말기가 상대방의 공개키를 확인한 후 Diffie-Hellman 키 교환기법으로 키를 공유하는 프로토콜이다. 개선된 BY 프로토콜 [8]의 경우는 이동단말기가 세션키를 생성하여 기지국에 알려주지만 개선된 AD 프로토콜 [9]의 경우에는 이동단말기와 기지국이 같이 세션키를 생성하여 두 세션키를 통해 새로운 세션키를 공유한다.

2.1 MSR + DH 프로토콜 [7]

Rabin에 의해 제안된 MSR 알고리즘은 기존의 공개키 암호알고리즘에 비해 상당히 빠른 속도로 암호 연산과 서명 생성을 할 수 있는 암호알고리즘이다. MSR을 이용한 초기의 MSR 프로토콜은 이동단말기가 자신의 인증서 (누구나 믿을 수 있는 인증기관(Certificate Authority)이 발행한)를 기지국과 공유된 세션키로 암호화하여 전달하는 프로토콜이다. 이 프로토콜은 기지국의 공개키를 확인 할 수 있는 방법이 없고 메세지의 재전송(replay) 공격에 취약하다는 것이 밝혀진 뒤에 Carlsen에 의해 Improved MSR 프로토콜(IMSR)로 수정되었다 [7]. IMSR 프로토콜은 MSR 프로토콜에 challenge-response 기법을 추가하고 기지국의 공개키의 인증서를 추가함으로써 MSR의 보안 문제를 해결했다. MSR + DH 프로토콜은 IMSR 프로토콜을 Diffie-Hellman 키 교환 프로토콜과 결합한 프로토콜이다. MSR + DH 프로토콜의 세부사항은 아래와 같다.

1. $B \rightarrow M : B, N_B, PK_B, Cert(B)$
2. $M \rightarrow B : \{x\}_{PK_B}, \{N_B, M, PK_M, Cert(M)\}_x$

위 프로토콜에서 B는 기지국을 나타내고 M은 이동단말기를 나타낸다. 화살표는 데이타의 전송 방향을 나타내고 PK는 공개키를 나타낸다. $\{X\}_K$는 X가 K를 키로 사용하여 암호화 되었음을 의미한다. 1번 절차에서 기지국은 자신의 공개키를 인증서와 함께 이동단말기에게 전달하고 이동단말기는 기지국의 공개키를 확인한 후 자신의 공개키와 기지국이 보낸 난수를 이동단말기가 정한 세션키로 암호화하여 기지국에 전달한다. 메세지 전달이 끝난 후 기지국과 이동단말기는 상대방의 공개키와 자신의 비밀키로 공유되는 세션키를 각자 계산한다.

2.2 개선된 BY 프로토콜

Beller와 Yacobi가 제안한 BY 프로토콜 [8]은 기지국과 이동단말기 모두 공개키을 가지고 인증과 키 공유를 하는 프로토콜이다. 이 프로토 콜이 man-in-the-middle 공격에 취약하다는 것이 밝혀진 뒤에, Boyd와 Mathuria에 의해 man-in-the-middle 공격에 이겨낼 수 있도록 수정되었다 [1]. 아래는 Boyd와 Mathuria가 수정한 개선된 BY 프로토콜의 세부 사항이다.

1. $B \rightarrow M : B, PK_B, Cert(B), N_B$
2. $M \rightarrow B : \{x\}_{PK_B}, \{M, PK_M, Cert(M)\}_x, \{h(B, M, N_B, x)\}_{PK_M^{-1}}$
3. $B \rightarrow M : \{N_B\}_x$

위 프로토콜에서 기지국은 먼저, 자신의 이름(B), 자신의 공개키 (PK_B), 공개키에 대한 인증서($Cert(B)$)등을 이동단말기에 보낸다. 기지국의 메세지를 받은 이동단말기는 기지국의 공개키를 인증서를 통해 확인 한 후, 세션키(x)를 생성하고 자신의 공개키(PK_M), 인증서($Cert(M)$)를 세션키로 암호화한 후 자신의 서명($\{hash(B, M, N_B, x)\}_{PK_B^{-1}}$)과 함께 기지국에 전달 한다. 메세지를 받은 기지국은 세션키를 해독한 후 이동단말기의 공개키를 확인 하고 자신을 증명하기 위해 난수(N_B)를 세션키로 암호화해서 이동단말기에게 전달한다.

2.3 개선된 AD 프로토콜

Aziz와 Diffie가 제안한 AD 프로토콜 [9]은 프로토콜 진행중에 대칭키 암호알고리즘을 결정하고 기지국과 이동단말기가 각자 만든 세션키를 통해 새로운 세션키를 만드는 프로토콜이다. 이 프로토콜도 man-in-the-middle 공격에 취약한 것이 알려지자 Boyd와 Mathuria에 의해 man-in-the-middle 공격에 대항할 수 있도록 수정되었다 [1]. 아래는 Boyd와 Mathuria에 의해 개선된 AD프로토콜 세부사항이다.

1. $M \rightarrow B : Cert(M), N_M, alg\_list$

2. $B \rightarrow M : Cert(B), N_B, \{x_B\}_{PK_M}, sel\_alg, \{hash(x_B, M, N_M, sel\_alg)\}_{PK_B^{-1}}$
3. $M \rightarrow B : \{x_M\}_{PK_B}, \{hash(x_M, B, N_B)\}_{PK_M^{-1}}$

위 프로토콜에서 alg_list는 기지국이 선택할 수 있는 대칭키 방식을 나타내는 리스트이고 sel_alg는 기지국이 선택한 대칭키 암호화 방식을 나타낸다. 그 외의 다른 기호들은 위의 다른 프로토콜에서 의미하는 것과 동일하다. 프로토콜이 수행된 이후에 이동단말기와 기지국 사이의 공통 세션키는 $x_M \oplus x_B$이다. 이때 x_M은 이동단말기가 만든 세션키를 x_B는 기지국이 만든 세션키를 나타낸다.

제 3 절 제안 프로토콜

3.1 RSA 서명을 위한 SASC 프로토콜

RSA에서 서명을 생성하기 위해서는, 매우 큰 소수 p, q를 구하고 $n = pq$를 계산한다. 그리고나서, $\phi(n)(= (p-1)(q-1))$과 서로 소인 임의의 정수 ν를 고른 후에 $s\nu \equiv 1 \bmod \phi(n)$이 성립하는 s를 찾는다. 이러한 초기 설정이 끝나면, 서명자는 N, ν를 공개하고, s는 자신만이 아는 개인키의 역할을 하게 된다. 임의의 메시지 m에 대한 서명은 $m^s \bmod n$이 된다.

Matumoto 등이 [10]에서 제안한 RSA-S1 프로토콜의 기본 구조는 다음과 같다. RSA 서명 생성을 위한 SASC 프로토콜의 목적은 클라이언트가 비밀키 s를 서버에 알려주지 않으면서 $m^s \bmod n$을 계산하는 것이다. 클라이언트는 우선 $s \equiv \sum_{i=1}^{m} a_i x_i \bmod \phi(n)$이 성립하는 벡터(vector) $X = (x_1, x_2, \cdots, x_m)$와 $A = (a_1, a_2, \cdots, a_m)$를 찾는다. 이들 중 X를 서버에게 전해주고, 서버는 $z_i \equiv m^{x_i} \bmod N$(, where $1 \leq i \leq m$)들을 계산하여 클라이언트에게 돌려준다. 클라이언트는 $m^s \equiv \Pi_{i=1}^{m} (z_i)^{a_i} \bmod n$을 계산함으로써 RSA 서명을 얻는다.

그러나, 이러한 기존의 SASC 프로토콜들은 서버의 계산량이 매우 많고, 특히, 통신량이 매우 많아서 이동통신 환경에서 사용하기에는 힘들다. 또한 확률적 공격법 [11]을 이용하면 기존에 제시된 보안인수보다 훨씬 큰 보안인수들을 사용해야 하기때문에, 이러한 단점이 더 커진다. Hong, Shin, Lee, 그리고 Yoon은 [6]에서 SASC의 이러한 단점들을 극복하면서 성능이 뛰어난 새로운 접근방식의 server-aided RSA signature generation 프로토콜을 제안하였다. 따라서, 본 논문에서는 Hong 등의 방법을 사용한다. (이후로 기존의 나눔(decomposition) 기법을 사용하는 SASC 프로토콜을 나눔 SASC라 부르고, Hong 등의 감춤(blinding) 기법을 사용하는 프로토콜을 감춤 SASC라 부르도록 한다.)

본 논문에서 제안하는 상호인증 및 키 공유 프로토콜은 감춤 SASC 프로토콜을 사용하므로, 본 절에서 간략히 설명한다. 자세한 설명은 [6]을 참조하면 된다. 감춤 SASC 프로토콜은 사전계산 부분과 실제 서명 부분으로 나뉜다. 사전계산은 클라이언트의 비밀정보 s가 생성될 때 한 번만 수행하면 된다.

사전계산

1. 클라이언트는 다음 식들을 만족하는 r_i, r'_i들을 임의로 선택한다: $R' = \overline{r'_1}||\overline{r'_2}|| \cdots ||\overline{r'_k}, r'_i = \overline{r'_i} \times 2 + 1$, and $r_i \in \{a_R\}_1^l$. '||'은 붙임(concatenation)을 의미하고, $\{a_R\}_1^l$은 R에 대한 이진 덧셈사슬 [12]을 나타낸다. R과 R'은 임의의 수이다.

2. 클라이언트는 다음 식을 만족하도록 t를 계산한다: $t \equiv {r'_k}^{-1}(\cdots({r'_1}^{-1}(s_1 - r_1) - r_2) - \cdots - r_k) - R \bmod \lambda(N)$. s_1은 임의의 정수로서 $s = s_1 + s_2$가 되고, s_2는 서명생성 시에 이용된다.

3. 클라이언트는 $w_p \equiv q(q^{-1} \bmod p) \bmod N$과 $w_q \equiv p(p^{-1} \bmod q) \bmod N$을 미리 계산해 놓는다.

서명 생성 다음은 서명생성 프로토콜이다.

Client		Server
	m, n, and t $\longrightarrow$	
		computes Z : $Z = m^t \bmod n$
	Z $\longleftarrow$	
computes z_p and z_q : $z_p = (\cdots((Z \bmod p \times {m_p}^R)^{r'_k} \times {m_p}^{r_k})^{r'_{k-1}} \cdots)^{r'_1} \times {m_p}^{r_1} \bmod p$ $z_q = (\cdots((Z \bmod q \times {m_q}^R)^{r'_k} \times {m_q}^{r_k})^{r'_{k-1}} \cdots)^{r'_1} \times {m_q}^{r_1} \bmod q$ computes σ_p and σ_q : $\sigma_p = s_2 \bmod (p-1) + \varrho_p(p-1)$, $\sigma_q = s_2 \bmod (q-1) + \varrho_q(q-1)$, where $\varrho_p \in_R \{0, \ldots, q-2\}$, and $\varrho_q \in_R \{0, \ldots, p-2\}$.		
	σ_p, σ_q $\longrightarrow$	
		computes y_p and y_q : $y_p = m^{\sigma_p} \bmod n$ $y_q = m^{\sigma_q} \bmod n$
	y_p, y_q $\longleftarrow$	

computes $S = w_p S_p + w_q S_q \bmod n$, where
$S_p = y_p z_p \bmod p$ and $S_q = y_q z_q \bmod q$.
If $S^v \bmod n = m$, then transmits S.

위에서 설명한 프로토콜에서 각각 R과 R'의 길이인 b_R과 b'_R과 k는 보안인수로서, 적절한 탐색영역을 유지하면서 계산효율을 가장 높일 수 있는 값으로 선택한다. 그러한 값들 중 하나로, [6]에서 저자들이 추천한 보안인수는 $<b_R$=4,$b_{R'}$=30,k=5>이다.

3.2 Improved BY scheme에의 적용 [8, 1]

Beller와 Yacobi가 제안한 BY 프로토콜 [8]이 man-in-the-middle 공격에 이겨낼 수 있도록 Boyd와 Mathuria에 의해 수정되었다 [1]. 본 논문에서는 Boyd와 Mathuria에 의해 개선된 프로토콜에 감춤 SASC 프로토콜을 적용시킨다.

제안 프로토콜에서 이동단말기(Mobile Station)는 클라이언트가 되고, 기지국(Base Station)은 서버로서의 역할을 한다. 감춤 SASC 프로토콜에서 클라이언트가 서버에게 처음에 보내주는 t는 클라이언트의 공개키에 대한 인증기관의 공개키 인증서에 포함되어 있다고 가정한다. 각 기호들은 2절에서 사용한 것들과 이전 절의 감춤 SASC 프로토콜에서 사용한 것들을 사용한다.

1. $B \rightarrow M$: $B, PK_B, Cert(B), N_B$
2. $M \rightarrow B$: $\{x\}_{PK_B}, \{M, PK_M, Cert(M)\}_x, h(B, M, N_B, x)(= m)$
3. $B \rightarrow M$: $\{N_B\}_x$, Z
4. $M \rightarrow B$: σ_p, σ_q
5. $B \rightarrow M$: y_p,y_q
6. $M \rightarrow B$: $S = \{h(B, M, N_B, x)\}_{PK_M^{-1}}$

위에 기술한 프로토콜에서 1번과 2번 절차는 개선된 BY 프로토콜과 거의 동일하다. 다만, 2번 절차에서 마지막 메세지가 서명이 아닌 서명하고자 하는 메세지로 바뀌었다. 이는 이동단말기의 계산량을 줄이기 위해 SASC 프로토콜이 시작되는 시점이다. 원래의 SASC 프로토콜에 의하면, 이 때, t 값도 함께 기지국에 전달되어야 하지만 t 값은 메세지와 관계없이 일정한 값이므로 $Cert(M)$에 원래 포함되어 있다고 가정하였으므로, 따로 전달될 필요가 없다. 3번 절차의 $\{N_B\}_x$는 개선된 BY 프로토콜에서와 같다. 이를 제외한 3번부터 6번까지의 절차는 SASC를 위해 포함된 절차로서, 이전 절에서 설명한 값들과 동일하다. 이 때의 메세지 m은 $h(B, M, N_B, x)$이다. 원래의 SASC 프로토콜에서 기술되어 있듯이 6번 절차에서 이동단말기가 기지국에 서명을 전달하기 전에, 생성된 서명이 올바른 것인지 확인한 후, 올바른 것이 아니면 전달하지 않는다. 이는 SASC에 대한 능동적 공격을 막기 위해서이다.

3.3 Improved AD scheme에의 적용 [9, 1]

Aziz와 Diffie가 제안한 AD 프로토콜 [9]이 man-in-the-middle 공격에 이겨낼 수 있도록 Boyd와 Mathuria에 의해 수정되었다 [1]. 본 논문에서는 Boyd와 Mathuria에 의해 개선된 프로토콜에 감춤 SASC 프로토콜을 적용시킨다.

제안 프로토콜에서 이동단말기(Mobile Station)는 클라이언트가 되고, 기지국(Base Station)은 서버로서의 역할을 한다. 감춤 SASC 프로토콜에서 클라이언트가 서버에게 처음에 보내주는 t는 클라이언트의 공개키에 대한 인증기관의 공개키 인증서에 포함되어 있다고 가정한다. 각 기호들은 2절에서 사용한 것들과 이전 절의 감춤 SASC 프로토콜에서 사용한 것들을 사용한다.

다음 프로토콜에서는 SASC 프로토콜이 두 번 사용된다. 한 번은 m_1에 대한 복호화에 사용되고, 다른 한 번은 m_2에 대한 서명확인에 사용된다. SASC 프로토콜에 이용되는 값들인 $Z, \sigma_p, \sigma_q, y_p$, 그리고 y_q들은 아랫첨자를 이용해 어느 계산에 사용되는 값들인지를 표시하였다. 예를들어, Z_{m_1}은 m_1을 복호화 하는데 사용되는 SASC 프로토콜에 이용되는 Z 값을 의미한다. $m_1 = \{x_B\}_{PK_M}$이다.

1. $M \rightarrow B : Cert(M), N_M, alg\_list$
2. $B \rightarrow M : Cert(B), N_B, sel\_alg, \{hash(x_B, M, N_M, sel\_alg)\}_{PK_B^{-1}}, Z_{m_1}$
3. $M \rightarrow B : \sigma_{p_{m_1}}, \sigma_{q_{m_1}}, \{x_M\}_{PK_B}, hash(x_M, B, N_B)(= m_2)$
4. $B \rightarrow M : y_{p_{m_1}}, y_{q_{m_1}}, Z_{m_2}$
5. $M \rightarrow B : \sigma_{p_{m_2}}, \sigma_{q_{m_2}}$
6. $B \rightarrow M : y_{p_{m_2}}, y_{q_{m_2}}$
7. $M \rightarrow B : S_{m_2}(= \{hash(x_M, B, N_B)\}_{PK_M^{-1}}$

위에 기술된 프로토콜이 수행된 이후에 이동단말기와 기지국 사이의 공통 세션키는 $x_M \oplus x_B$이다. 위에서 언급 하였듯이, 두 번의 SASC 프로토콜이 이용되는데, 각각이 절차 3과 절차 4에서 중첩되어 있다.

이번 절에서 제안하는 프로토콜의 기본 원칙은 앞 절에서 개선된 BY 기법에 적용한 것과 동일하다. 즉, 원래의 개선된 BY 기법에서 서명수행과 복호화에 SASC 프로토콜을 적용시킨 것이다. 다만, 절차 2에서 기지국이 이동단말기에게 $\{x_M\}_{PK_M}(= m_1)$을 건네주어야 하는데, 기지국은 이미 절차 1에서 이동단말기로부터 $Cert(M)$을 받았기때문에 이동단말기 M에 대한 t를 알고 있다. 따라서, 바로 m_1에 대한 Z_{m_1}을 이동단말기에 보낼 수 있다.

제안 프로토콜들의 안전성은 사용한 SASC 프로토콜과 상호인증 및 키 공유 프로토콜에 달려 있다. 추가로 발생하는 안전성에 대한 위협은 없다.

제 4 절 성능분석

본 절에서는 이동통신 환경을 목적으로 개발된 기존의 상호인증 및 세션키 공유 프로토콜들의 성능과 제안 프로토콜의 성능을 비교한다. 성능 척도는 이동단말기에서 요구되는 모듈라 곱셈의 횟수를 이용하였다.

비교대상 프로토콜들은 다음과 같다: Beller, Chang, 그리고 Yacobi에 의해 개발되고 Carlsen에 의해 수정된 MSR+DH 프로토콜 [13, 7, 1], Beller와 Yacobi의 ElGamal 알고리즘을 이용한 프로토콜 [8], 그리고 Aziz와 Diffie에 개발된 일반적인 공개키 알고리즘을 이용하는 프로토콜들과 비교하였다. 비교대상이 되는 모든 프로토콜들은 Man-in-the middle attak을 이겨낼 수 있도록 Boyd와 Mathuria에 의해 수정된 것을 사용하였다.

MSR+DH 프로토콜과 개선된 BY 기법은 공개키 인증서로 ElGamal 알고리즘을 사용한다고 가정하였다. 개선된 AD 기법은 공개키 인증서로 ElGamal을 사용한 것과 RSA를 사용한 것 모두를 비교하였다. 제안 프로토콜은 공개키 인증서와 암호 알고리즘 모두 RSA를 사용하였다. RSA 알고리즘을 사용할 때, 암호화할 때와 서명을 확인할 때는 작은 지수를 사용한다. 즉, 3을 공개키로 사용한다고 가정하였다. 또한, RSA 복호화와 서명생성 시에는 CRT(Chinese Remainder Theorem)를 사용한다고 가정한다.

표 1에 기존 프로토콜과의 성능비교가 나타나 있다. 512비트 키를 사용할 경우의 성능이다. 즉 ElGamal의 경우에 공통 modulus p가 512비트이고, RSA의 경우에 공개키 modulus N이 512비트이다. 표 1에서 공개키 연산횟수는 각 프로토콜에서 공개키를 사용하는 연산, 즉, 암호화와 서명확인 횟수를 의미하고, 개인키 연산횟수는 개인키를 이용하는 복호화와 서명생성 횟수를 의미한다. 제안 프로토콜에서 SASC의 보안인수로는 [6]에서 저자들이 사용한 $<b_R$=4,$b_{R'}$=30,k=5>를 이용하였다.

프로토콜	공개키 연산	개인키 연산	사용 알고리즘	모듈라곱셈 횟수
MSR+DH [1]	2	1	ElGamal+DH	3750(=1500*2+750)
IBY [8, 1]	2	1	ElGamal	3750(=1500*2+750)
IAD [9, 1]	3	2	ElGamal	6000(=1500*3+750*2)
IAD [9, 1]	3	2	RSA	406(=2*3+200*2)
Proposed IBY	2	1	RSA	29(=3*2+23)
Proposed IAD	3	2	RSA	52(=2*3+23*2)

표 1: 각 프로토콜들 간의 성능비교. (공개키 연산 = 암호화 혹은 서명확인, 개인키 연산 = 복호화 혹은 서명생성)

표 1에서 보면, 제안 프로토콜은 기존의 것들에 비해 8 ~ 20배 이상의 성능향상을 보인다. 이처럼, 많은 성능향상을 보이는 것은 부분적으로는, 알고리즘의 선택에 있다. 예를들어, ElGamal의 경우 개인키 연산에 비해 공개키 연산이 매우 힘든 연산인데, 표 1에서 볼 수 있듯이, 두 번 이상의 공개키 연산이 필요하다. 이는 인증기관의 인증서를 확인하는 데 주로 사용된다. 또한, RSA는 공개키를 아주 작은 숫자로 설정할 수 있기때문에 공개키 연산에서 매우 효율적이다.

그러나, 서론에서도 언급했듯이, RSA는 서명생성이나 암호문의 복호화에 MSR+DH

프로토콜들처럼 온전한 모듈라 멱승 연산을 요구하기때문에 실제로 사용하기에 힘들었다. 본 논문에서 제안하는 프로토콜은 SASC 프로토콜을 이용하여, 기지국의 연산능력을 사용함으로써, 위와 같은 전체 성능향상을 기대할 수 있다.

SASC를 이용하지 않는 기존의 프로토콜들 중에서도 개선된 BY 기법 같은 경우는 ElGamal 알고리즘을 이용하여 사전계산을 통한 성능향상을 기대하였으나, 이는 실제로 계산량이 줄어든 것이 아니다. 다만, 사용자가 사용하지 않는 동안에 미리 계산하여 두는 것이므로, 연속사용 시 효과를 볼 수 없고, 계산량이 줄지 않으므로 전력소모량은 변함이 없다. 게다가, ElGamal 암호시스템은 암/복호화 알고리즘과, 서명 생성/확인 알고리즘이 각각 따로 구현되어야 하므로, 이동단말기의 하드웨어/소프트웨어 구현 시 비용상승을 초래할 수 있다. MSR+DH 프로토콜들의 경우도 암/복호화 알고리즘, 서명 생성/확인 알고리즘, 키 교환 알고리즘이 각각 따로 구현되어야 한다.

제 5 절 결론

본 논문에서는 이동통신 환경에서, 이동단말기와 기지국 사이에 안전한 통신을 제공하기 위한, 효율적인 상호인증 및 세션키 공유 프로토콜을 제안하였다. 이동통신에서의 암호 프로토콜을 설계할 때, 이동단말기의 부족한 계산능력으로도 합당한 시간 내에 연산할 수 있도록 해야 한다. 따라서, 공개키를 이용한 상호인증 및 세션키 공유 프로토콜들은, 대칭키 암호시스템에 비해 많은 장점들을 지녔음에도 불구하고, 그 실용성을 검증받기 힘들었다.

본 논문에서는 공개키의 이러한 단점을 극복하기 위해, SASC 프로토콜을 적용하였다. 이는 스마트 카드의 부족한 계산능력을 보충하기 위해 많은 연구가 이루어지고 있는 프로토콜로서, 이동통신에서의 이동단말기와 비슷한 입장에 처해있는 스마트 카드로 하여금 서버 (카드 리더, 현금지급기 등)의 계산능력을 빌릴 수 있도록 하는 기법이다. 기존의 SASC 프로토콜들은 클라이언트의 계산량을 현저히 줄이는 반면, 통신량이 매우 많고, 서버의 계산량이 커서 이동통신 환경에서는 적용하기가 어려웠다. 그러나, 최근 제안된 새로운 방식의 SASC 프로토콜은 이러한 단점을 극복한 것으로서, 이동통신 환경에 적용이 가능하다. 본 논문에서 제안한 방식은 기존의 방식에 비해 8 20배 이상의 속도 향상을 기대할 수 있으므로, 공개키 방식의 장점을 충분히 활용하는 데 기여할 것이다.

참고 서적

[1] C. Boyd and A. Mathuria, "Key establishment protocols for secure mobile communications: A selective survey," in *ACISP'98, Lecture Notes in Computer Science*, vol. 1438, pp. 344–355, 1998.

[2] R.L.Rivest, A.Shamir, and L.Adleman, "A method for obtaining digital signatures and public key cryptosystems," *CACM*, vol. 21, pp. 120–126, 1978.

[3] T.ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469–472, july 1985.

[4] W.Diffie and M.E.Hellman, "New directions in cryptography," *IEEE Trans. Computers*, vol. IT-22, pp. 644–654, June 1976.

[5] M. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," *MIT/LCS/TR-212*, 1979.

[6] S.-M. Hong, J.-B. Shin, H.Lee-Kwnag, and H. Yoon, "A new approach to server-aided secret computation," in *ICISC'98, to be appeared*, 1998.

[7] U.Carlsen, "Optimal privacy and authentication on a portable communications system," *ACM Operating Systems Review*, vol. 28, no. 3, pp. 16–23, 1994.

[8] M.J.Beller and Y.Yacobi, "Fully-fledged two-way public key authentication and key agreement for low-cost terminals," *Electronics Letters*, vol. 29, pp. 999–1001, May 1993.

[9] A.Aziz and W.Diffie, "Privacy and authentication for wireless local area networks," *IEEE Personal Communications*, vol. 1, pp. 25–31, 1994.

[10] T.Matsumoto, K.Kato, and H.Imai, "Speeding up secret computations with insecure auxiliary devices," in *Crypto'88*, pp. 497–506, 1988.

[11] S.-M. Hong, S. Jean, and H. Yoon, "Probabilistic attack on server-aided secret computation protocol," in *Submitted*, 1998.

[12] D.E.Knuth, *The art of computer programming Vol.2.* Addition-Wesley,Inc., 1981.

[13] M.J.Beller, L.-F.Chang, and Y.Yacobi, "Privacy and authentication on a portable communications system," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 821–829, August 1993.