情報保護學會論文誌
第17卷 第3號, 2007. 6

프라이버시와 완전한 전방향 안전성을 제공하는 UMTS 키 동의 프로토콜

김대영[1†], 최용강[1], 김상진[2], 오희국[1‡]
[1]한양대학교, [2]한국기술교육대학교

A UMTS Key Agreement Protocol Providing Privacy and Perfect Forward Secrecy

Daeyoung Kim[1†], Yonggang Cui[1], Sangjin Kim[2], Heekuck Oh[1‡]
[1]Hanyang University, [2]Korea University of Technology and Education,

요 약

3G 이동 통신 기술 중 하나인 UMTS(Universal Mobile Telecommunications System)에서는 무선 구간의 안전한 통신을 위해 UMTS AKA(Authentication and Key Agreement) 프로토콜을 사용한다. 그러나 이 프로토콜은 SN(Serving Network)과 HN(Home Network)의 네트워크 대역폭 소모 문제, SQN(SeQuence Number) 동기화 문제 등 여러 가지 문제점이 제기되었다. 본 논문에서는 UMTS AKA 프로토콜의 문제점을 개선한 새로운 타원곡선 기반 UMTS AKA 프로토콜을 제안한다. 제안하는 프로토콜은 IMSI(International Mobile Subscriber Identity)의 노출로 인한 문제점으로부터 프라이버시 보호를 강화하고, ECDH(Elliptic Curve Diffie Hellman) 기법을 통해 완전한 전방향 안전성을 제공한다.

ABSTRACT

In the UMTS (Universal Mobile Telecommunication System), which is one of 3G mobile communication standards, the protocol called UMTS AKA (Authentication and Key Agreement) is used to authenticate mobile stations. However, the UMTS AKA protocol has some weakness, including network bandwidth consumption between a SN (Serving Network) and a HN (Home Network) and SQN (SeQuence Number) synchronization. In this paper, we propose a new improved protocol for UMTS that overcomes UMTS AKA weakness. Our protocol solves the privacy problem caused by IMSI (International Mobile Subscriber Identity)'s disclosure and provides perfect forward secrecy using ECDH (Elliptic Curve Diffie Hellman).

Keywords : *UMTS, 키 동의 프로토콜, Perfect Forward Secrecy, Privacy*

I. 서 론

유럽을 비롯한 많은 나라에서 사용하고 있는 UMTS는 3세대 이동 통신 기술 중의 하나로서, W-CDMA (Wideband-Code Division Multiple Access)를 기반으로 사용하며 3GPP(The 3rd Generation Partnership Project)

접수일: 2007년 1월 11일; 채택일: 2007년 4월 10일
* 이 논문은 2006년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임 (No. R01-2006-000-10957-0).
† 주저자, kimdy@cse.hanyang.ac.kr
‡ 교신저자, hkoh@cse.hanyang.ac.kr

에 의해 표준화되고 있다. GSM(Global System for Mobile Communication)을 발전시킨 UMTS의 특징 중 하나는 안전성 강화로써, 크게 무선 구간에서의 액세스 보안, 핵심 망에서의 보안, 사용자 영역에서의 보안 등으로 분류할 수 있다. 그 중에서 보안에 취약한 무선 구간에서의 액세스 보안이 가장 중요시되고 있는데, 이로 인해 UMTS의 표준화 기구인 3GPP에서는 무선 구간에서의 안전한 통신을 위한 UMTS AKA프로토콜을 개발하였다[1]. UMTS AKA 프로토콜은 사용자와 네트워크 간의 상호 인증을 제공하고, 암호화키와 무결성키를 확립하여 준다. 그러나 UMTS AKA 프로토콜은 SN과 HN 사이의 대역폭 소모 문제, SQN 동기화 문제 등의 여러 가지 문제점들이 제기되고 있다. 이 문제점을 해결하기 위해 기존의 프로토콜에서 동기화를 제거한 AP-AKA 프로토콜[2], 해시 체인을 이용한 Harn/Hsin 프로토콜[3], 임시키를 사용한 UMTS X-AKA 프로토콜[4] 등이 제안되었다.

본 논문에서는 안전성을 한 단계 높이기 위해 전방향 안전성이 보장되도록 개선하였으며, IMSI 노출 때문에 발생하는 프라이버시 문제점을 극복한 타원 곡선 기반의 새로운 UMTS AKA 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 이해를 돕기 위한 수학적 배경과 UMTS 네트워크의 전체적인 구조에 대해 설명하고, 3장에서는 UMTS-AKA 프로토콜을 비롯한 기존에 제안된 프로토콜에 대해 기술한다. 4장에서는 제안하는 UMTS AKA 프로토콜을 자세히 기술하고, 제안하는 프로토콜의 안전성과 효율성을 분석한다. 마지막으로 5장에서는 결론을 맺는다.

Ⅱ. 연구 배경

2.1. 수학적 배경

본 논문에서는 앞으로 다음과 같은 표기법을 사용한다. 1) q는 매우 큰 소수를 의미한다. 2) $\overrightarrow{G}$는 타원 곡선 위의 위수가 q인 순환군이다. 3) P는 $\overrightarrow{G}$의 임의의 생성자이다. 4) a와 b는 $\overrightarrow{Z_q}$의 임의의 원소들이다.

정의 1 (Discrete Logarithm Problem (DLP) in $\overrightarrow{G}$). $\overrightarrow{G}$의 원소 P와 aP가 주어졌을 때, $a(\in\overrightarrow{Z_q})$를 계산하는 문제를 말한다.

정의 2 (Computational Diffie-Hellman Problem (CDHP) in $\overrightarrow{G}$). $\overrightarrow{G}$의 원소 P, aP, bP가 주어졌을 때, $abP(\in\overrightarrow{G})$를 계산하는 문제를 말한다.

현재까지 DLP, CDHP를 다항시간 내에 계산하는 것은 계산적으로 어렵다고 알려져 있다. 이 논문에서 제안하는 프로토콜의 안전성은 위의 문제들을 다항시간 내에 계산하는 것이 어렵다는 가정에 기반하고 있다.

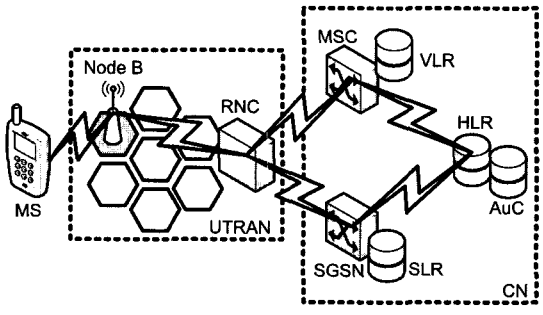
2.2. 전방향 안전성

- 전방향 안전성(forward secrecy): 세션키 생성에 참여하는 일부 참여자들의 장기간 비밀키가 노출되어도 이전 세션키를 얻는 것이 계산적으로 어려워야 한다.
- 완전한 전방향 안전성(perfect forward secrecy): 세션키 생성에 참여하는 모든 사용자들의 장기간 비밀키가 노출된다 하더라도 이전 세션키를 얻는 것이 계산적으로 어려워야 한다.

2.3. UMTS 네트워크 구조

〔그림 1〕은 UMTS 네트워크 구조와 액세스 보안에 대해 나타내고 있다[1].

- IMSI(International Mobile Subscriber Identity): *IMSI*는 가입자 식별 값으로 USIM(Universal Subscriber Identity Module)과 AuC(Authentication Center)에 저장되어 있다. *IMSI*는 초기에 가입자를 식별할 때 사용되어 진다.
- USIM : UMTS-AKA 프로토콜을 수행하기 위해 필요한 가입자의 비밀키 K와 *IMSI* 등의 사용자 정보와 암호 알고리즘 등을 저장하는 모듈이다.
- MS(Mobile Station): USIM이 삽입되는 단말기로



〔그림 1〕 UMTS 네트워크 구조

써, 암·복호화와 무결성을 검증하는 기능이 있다.

- Node B(or Base station): MS와 RNC사이에 통신 연결을 담당하는 역할을 한다.
- RNC(Radio Network Controller): Node B를 제어하고, SGSN(Serving GPRS Support Node)이나 MSC/VLR(Mobile Switching Center/Visitor Location Register)와 연결되며 암·복호화와 무결성 검증을 수행한다.
- SGSN: GPRS(General Packet Radio Service) 서비스 지역 내에서 MS와의 데이터 패킷 전달을 담당하는 노드이다.
- MSC/VLR: MSC는 CS(Circuit Switch) 서비스를 제공하고 VLR은 방문 가입자에 대한 정보를 저장하는 역할을 한다.
- HLR(Home Location Register): 가입자에 대한 정보가 저장되며, SGSN이나 MSC/VLR에게 전달하게 된다.
- AuC: 가입자의 인증과 무선구간에서 암호화를 지원을 위한 시스템으로 가입자 정보 *IMSI*와 비밀키 K를 저장하고 있다.

Ⅲ. 관련 연구

3.1. 표기법

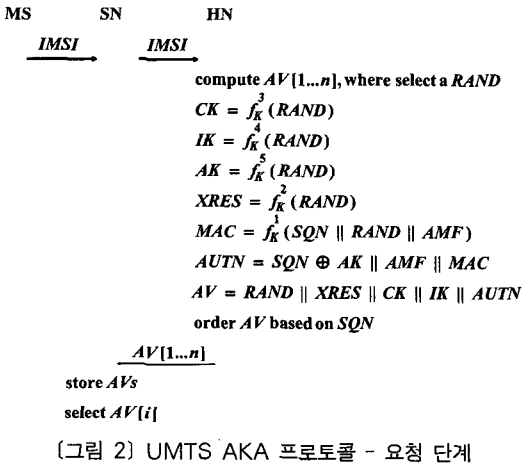
본 논문에서 설명하는 프로토콜의 표기법은 다음과 같이 통일해서 기술한다.

〔표 1〕 프로토콜 표기법

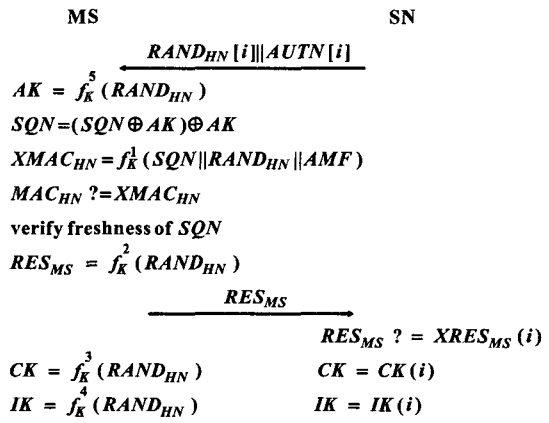
표 기	의 미
$RAND_A$ or N_A	A에서 생성한 난수
$cTID_{MS}$ or $pTID_{MS}$	$IMSI$의 프라이버시 보호를 위해 사용되는 임시 ID
ID_A	A의 식별자
$H(M)$	메시지 M을 해시한 값
f_K^1	MAC(Message Authentication Code) 값을 계산하는 메시지 인증 함수
f_K^2	RES or $XRES$ 값을 계산하는 메시지 인증 함수
f_K^3, f_K^4, f_K^5	각각의 키를 생성하는 함수
MAC_A	A가 f_K^1 함수로 생성한 MAC 값
$XMAC_A$	A가 생성한 메시지 인증 코드 값에 대응하는 검증 값
AMF	다중 인증 알고리즘과 키를 지원하거나 암호화키와 무결성키의 수명을 관리하는 인증 관리 필드 (Authentication Management Field)
SQN	동기화를 수행하기 위한 순서 번호 (SeQuence Number)
T_A	재사용 공격 방지를 위해 A가 생성한 타임스탬프
SK	ECDH 기법으로 생성한 세션키
TK	제안하는 프로토콜과 X-AKA에서 사용되는 티켓 키
CK	f_K^3 함수에 의해 생성되는 암호화키
IK	f_K^4 함수에 의해 생성되는 무결성키
AK	f_K^5 함수에 의해 생성되는 익명성키
RES_A or $XRES_A$	A가 f_{SK}^2 함수로 생성하는 인증 응답 값과 그에 대응하는 기대 값

3.2. UMTS AKA 프로토콜

UMTS AKA 프로토콜은 MS가 SN에 방문했을 때 HN을 통해 상호 인증을 수행하고, MS와 SN 사이에 암호화키와 무결성키를 확립시켜주는 키 동의 프로토콜이다[1]. 이 프로토콜은 MS의 USIM과 HN의 AuC가 서로 비밀키를 공유하고 있고, SN과 HN은 MAPsec (Mobile Application Part Security)이나 IPsec(IP Security)과 같은 네트워크 도메인 보안을 통해 서로 안전한 통신을 수행한다고 가정한다. 이러한 가정은 앞으로 나올 모든 관련 프로토콜에 포함된다. 그리고 재전송 공격을 방지하기 위해서 MS와 HN은 각각 SQN을 유지하고 있다. 〔그림 2〕는 UMTS AKA 프로토콜의 요청 단계를 나타내고, 〔그림 3〕은 UMTS AKA 프로토콜의



〔그림 2〕 UMTS AKA 프로토콜 - 요청 단계



〔그림 3〕 UMTS AKA 프로토콜 - 인증 및 키 동의 단계

인증 및 키 동의 단계를 나타낸다.

UMTS AKA 프로토콜의 진행과정을 살펴보면 다음과 같다. 먼저 MS는 인증을 요청하기 위해 SN을 거쳐 HN에게 *IMSI*를 보낸다. 인증 요청을 받은 HN은 다음과 같이 AV(Authentication Vector)를 n개 생성한다. 각 AV를 생성하기 위해서는 먼저 $RAND_{HN}$을 생성하고 이 값으로부터 〔그림 2〕와 같이, CK, IK, AK, $XRES_{MS}$, MAC_{HN} 등을 계산한다.

그 다음에 HN은 생성한 값으로부터 $AUTN$(Authentication Token)을 만든다. $AUTN$은 $AUTN = SQN \oplus AK \| AMF \| MAC_{HN}$과 같이 구성되어 있다. 그리고 HN은 생성한 AV들을 모두 SN에게 보내게 된다. AV들을 받은 SN은 우선 AV들을 저장하고 FIFO(First In First Out) 방식으로 i번째 AV로부터 $RAND_{HN[i]}$와 $AUTN[i]$ 값을 선택해 MS에게 보낸다. MS는 처음에 $RAND_{HN}$로부터 AK를 계산하고, SQN을 구한 다음에 $XMAC_{HN}$을 계산해 메시지를 검증하게 된다. 그리고 SQN이 올바른 범위에 있는 값인지 SQN의 유효성에 대해서도 검증하게 된다. 이 때 SQN이 올바른 범위 내에 있지 않다면 MS는 동기화 실패 메시지를 SN에게 보내게 되어 동기화를 다시 수행하게 된다. 메시지 검증이 올바르게 수행되었다면 MS는 RES_{MS}값을 계산해 SN에게 보내게 되고, SN은 RES_{MS}와 $XRES_{MS}$ 값을 비교해 MS를 확인하고 서로 확립된 CK와 IK를 이용해 안전하게 통신을 하게 된다.

3.3 AP-AKA 프로토콜

Zhang 등이 제안한 AP-AKA 프로토콜은 SQN을 사용하지 않음으로써, MS와 HN의 동기화 과정을 제거한 특징을 가진다[2]. 또한, 기존 UMTS AKA 프로토콜을 개선해 MS와 SN간의 상호 인증에 대한 안전성을 향상시켰다. 이 프로토콜의 진행과정은 다음과 같다. SN은 UMTS AKA 프로토콜들과 달리 사용자에게 먼저 $RAND_{SN}$를 보내게 된다. 요청을 받은 MS는 $RAND_{MS}$를 생성하고, $RAND_{MS}$, $MAC_{MS} = f_K^1(RAND_{MS} \| RAND_{SN} \| ID_{SN})$를 SN에게 보낸다. 응답을 받은 SN은 $IMSI$, $RAND_{SN}$, $RAND_{MS}$, MAC_{MS}를 HN에게 전달한다. HN은 MAC_{MS}를 검증하고 n개의 AV를 생성해 SN에게 보내게 된다. 이 때, 각 AV는 다음과 같이 구성된다.

$$AV = \{RAND_{HN} \| XRES \| SK \| AUTH\}$$

여기서 $XRES$, SK, $AUTH$는 다음과 같으며,

$$XRES = f_K^2(RAND_{HN})$$
$$SK = f_K^3(RAND_{HN})$$
$$AUTH = i \| RAND_i \| MAC_{HN}$$

$RAND_i$, MAC_{HN}은 다음과 같다.

$$RAND_i = f_K^4(i \| RAND_{MS})$$
$$MAC_{HN} = f_K^1(RAND_{HN} \| i \| RAND_i)$$

이 때 i는 AV의 색인번호이다.

SN은 사용하지 않은 AV들 중 낮은 색인번호 i를 가지는 AV를 선택해 MS에게 $RAND_{HN}$과 함께 보낸다. MS는 MAC_{HN}을 검증하고 SN에게 $RES = f_K^1(RAND_{HN})$를 보낸다. 마지막으로 SN은 RES와 $XRES$를 비교해 값이 같으면 프로토콜 진행이 모두 끝나게 된다. 이 때, 세션키는 $SK = CK \| IK$로 구성된다. 그리고 UMTS AKA 프로토콜과 마찬가지로 암호화키와 무결성키로 통신을 수행한다.

3.4. Harn-Hsin 프로토콜

Harn-Hsin 프로토콜은 기존의 프로토콜처럼 여러 개의 AV를 사용하지 않고 해시 체인 기법을 적용한 프로토콜로써, 앞으로는 HH-AKA라 한다[3]. HH-AKA 프로토콜을 여러 개의 해시 체인을 사용하지만 간단하게 설명하기 위해 하나의 긴 해시 체인을 사용한다고 가정한다. 이 프로토콜의 진행과정은 다음과 같다.

먼저 MS는 SN에게 $IMSI$, $h^m(b_M)$, T_{MS}, MAC_{MS}를 전달한다. 여기서 MAC_{MS}는 다음과 같다.

$$MAC_{MS} = f_K^1(IMSI \| h^m(b_M) \| T_{MS})$$

이 때, h^m은 해시 체인의 m번째 값을 나타내고, b_M은 해시 체인의 시드를 나타낸다. SN은 MS로부터 받은 메시지를 그대로 HN에게 보내고, HN은 MAC_{MS}를 검증하고 하나의 AV를 생성하게 된다. AV는 다음과 같이 구성된다.

$$AV = IMSI \| h^m(b_M) \| RAND_{HN} \| AK \| CK \| IK$$

여기서 AK, CK, IK는 다음과 같다.

$$AK = f_K^6(RAND_{HN})$$
$$CK = f_K^3(RAND_{HN})$$
$$IK = f_K^4(RAND_{HN})$$

HN은 생성한 AV를 SN에게 보내고, SN은 또 다른 해시 체인 $h^n(b_S)$를 생성하고 MS에게 $RAND_{HN} \| h^n(b_S) \| MAC_{SN}$을 전달한다. 여기서 MAC_{SN}은 다음과 같다.

$$MAC_{SN} = f_{AK}^1(RAND_{HN} \| h^n(b_S))$$

MS는 MAC_{SN}를 검증하고 UMTS AKA 프로토콜과 동일한 방법으로 CK, IK를 생성한다. MS는 SN에게 $h^{m-i}(b_M)$을 보내고 SN은 MS로부터 받은 해시 체인을 검증하고 다음과 같이 새로운 암호화키와 무결성키를 생성한다.

$$CK_N = f_{CK}^3(h^{m-i}(b_S) \| h^{m-i}(b_M))$$
$$IK_N = f_{CK}^4(h^{m-i}(b_S) \| h^{m-i}(b_M))$$

그 다음에 MS에게 $h^{m-i}(b_M)$을 전달한다. MS도 SN으로부터 받은 해시 체인을 검증하고 CK_N, IK_N를 생성해 안전한 통신을 수행하게 된다.

3.5 UMTS X-AKA 프로토콜

Huang 등이 제안한 UMTS X-AKA 프로토콜은 임시 키 발급 메커니즘을 이용해 대역폭 소비 문제를 해결하고 저장 공간 오버헤드를 줄였다[4]. 이 프로토콜은 커버로스와 같이 티켓 키를 발급해 사용한다. 프로토콜의 진행과정은 다음과 같다.

먼저, MS는 SN을 거쳐 HN에게 $IMSI$, T_{MS}, $MAC_{MS} = f_K^1(T_{MS})$를 보낸다. HN은 MAC_{MS}를 검증하고 $MAC_{HN} = f_K^1(RAND_{HN} \| AMF)$을 생성한 다음 $TK = f_K^x(T_{MS})$와 $AUTH_{HN}$을 SN에게 보내게 된다. 이 때, f_K^x는 임시 티켓 발급을 위한 키 생성 함수이며 $AUTH_{HN}$은 다음과 같다.

$$AUTH_{HN} = MAC_{HN} \| RAND_{HN} \| AMF$$

SN은 $RAND_{SN}$을 생성하고 $AUTH_{SN}$을 만들어 MS에게 보낸다. 여기서 $AUTH_{SN}$은 다음과 같으며,

$$AUTH_{SN} = MAC_{SN} \| RAND_{SN} \| RAND_{HN} \| AMF \| j$$

MAC_{SN}은 다음과 같다.

$$MAC_{SN} = f_{TK}^1(MAC_{HN} \| RAND_{SN} + j \times RAND_{HN})$$

MS는 MAC_{SN}와 MAC_{HN}을 검증하고 j번째가 맞는지 확인한다. 그 다음에 MS는 SN에게 $RES = f_{TK}^2(RAND_{SN})$를 보내고, SN은 $XRES$와 비교해 RES를 검증하게 된다. 끝으로 MS와 SN은 $CK = f_{TK}^3(RAND_{SN})$, $IK = f_{TK}^4(RAND_{SN})$를 생성해 안전한 통신을 수행한다.

3.6 프로토콜 비교

UMTS AKA와 AP-AKA는 AV들을 사용해 HN의 접속을 최소화하지만, AV들의 사용은 SN과 HN의 대역폭 소모를 야기 시키고 SN의 저장 공간 오버헤드를 발생시킨다. 이것을 해결한 HH-AKA와 X-AKA는 AV들을 사용하지 않는다. 이 프로토콜들은 AV들 대신에 각각 해시 체인과 커버로스와 같은 티켓 키를 이용한다. HH-AKA는 여러 개의 해시체인을 사용하면 UMTS AKA와 유사한 대역폭 소모와 저장 공간 오버헤드가 발생하게 된다. 하지만 하나의 긴 해시 체인을 사용하게 된다면, 이 문제를 해결할 수 있다.

UMTS AKA에서는 HN이 MS를 인증하지 못하지만 다른 프로토콜에서는 HN이 MAC_{MS}를 검증함으로써 인증한다. 또한 MS는 MAC_{HN}나 MAC_{SN}을 통해 HN을 인증한다. HH-AKA에서는 MAC_{SN}의 적시성을 검증할 수 없지만, UMTS AKA와 AP-AKA에서는 MAC_{HN}에 난스와 타임스탬프가 포함되어있고 X-AKA에서는 타임스탬프를 사용해 TK를 계산한다. 이것들을 이용해 MS는 MAC값들의 최신성을 검증할 수 있다.

이전에 제안된 프로토콜을 살펴보면, 장기간 비밀키인 K가 노출되면 이전의 통신 내용이 모두 노출된다. 하지만 AV들에 포함된 단기간 비밀키가 노출되면 다른

〔표 2〕 프로토콜 비교

	UMTS AKA	AP-AKA	HH-AKA	X-AKA	제안하는 프로토콜
AVs 사용여부	○	○	hash chain	ticket key	ticket key
MS와 HN의 동기화	counter	×	×	clock	clock
SN과 HN사이에 네트워크 대역폭 소모	○	○	×*	×	×
SN의 저장공간 오버헤드 여부	○	○	×*	×	×
MS와 HN의 상호 인증	MS ← HN	○	○	○	○
MS와 SN의 상호 인증	○	○	○	○	○
장기간 비밀키 노출의 영향	○	○	○	○	×
단기간 비밀키 노출의 영향	×	×	×**	○***	×***

* : 단일 해시 체인을 사용할 경우
** : 해시체인의 루트가 노출되었을 경우
*** : TK_{HN}이 노출되었을 경우

AV들에 영향을 주지 않는다. HH-AKA에서는 해시 체인 루트, *CK*, *IK*가 모두 노출되었을 경우에만 이전의 통신 내용이나 이후의 통신 내용을 알 수 있게 된다. X-AKA에서는 *TK*가 노출된다면 *TK*를 이용한 이전의 통신 내용을 모두 알 수 있게 된다. 하지만 제안하는 프로토콜에서는 이러한 모든 문제가 발생하지 않는다.

Ⅳ. 제안하는 프로토콜

제안하는 프로토콜은 이전에 제안된 프로토콜들의 장점을 가지면서도 프라이버시 강화와 완전한 전방향 안전성을 제공하는 UMTS AKA 프로토콜이다. 이 프로토콜은 타원 곡선 암호방식을 기반으로 완전한 전방향 안전성은 보장한다. 먼저 제안하는 프로토콜의 가정을 살펴보고 프로토콜에 대해 설명한다.

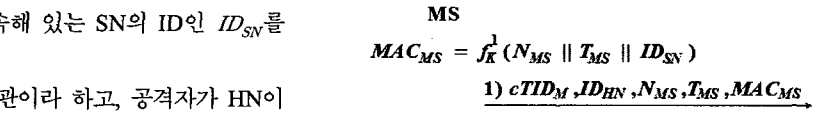
4.1 프로토콜의 가정

- MS는 자신의 HN사이에 비밀키 *K*, 임시 ID인 TID_{MS}, 암호알고리즘을 공유하고 있다.
- SN과 HN의 통신 구간은 MAPsec이나 IPsec과 같은 Network Domain Security 메커니즘을 통해 안전한 채널을 이룬다고 가정한다.
- MS는 현재 자신이 속해 있는 SN의 ID인 ID_{SN}를 알 수 있다.
- HN과 SN은 신뢰 기관이라 하고, 공격자가 HN이나 SN의 제어권을 획득하여 생기는 문제는 고려하지 않는다.
- 암호알고리즘의 문제점으로 생기는 프로토콜의 문제점이나 GSM과 UMTS가 공존하여 생기는 문제점은 고려하지 않는다[5].

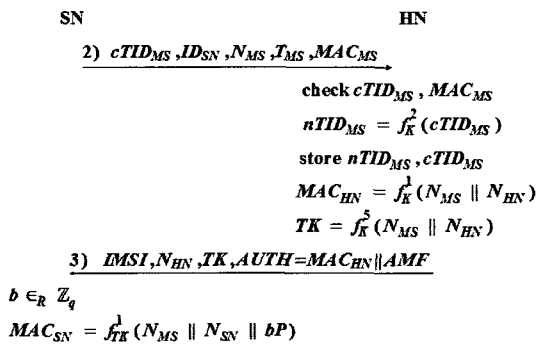
4.2 프로토콜

이 프로토콜에서 MS와 SN은 다음과 같은 키들을 확립한다. *TK*는 *SK*를 생성하기 위해 사용되는 키로써, 수명을 부여하여 시간제한을 할 수 있고 카운터를 공유하여 횟수를 제한할 수 있다. 본 논문에서는 시간제한 방식으로 설명한다. *SK*는 완전한 전방향 안전성을 보장하기 위해 ECDH 기법을 이용하여 생성한 키로써, *CK*와 *IK*를 생성할 때 사용한다. 마지막으로 안전한 통신을 하기 위해 사용되는 암호화키인 *CK*와 무결성키인 *IK*를 생성하게 된다. 다음은 제안하는 프로토콜 중에 초기 프로토콜로써, 〔그림 4, 5, 6〕은 프로토콜의 진행 과정을 나타낸다.

- 단계 1. MS는 먼저 N_{MS}를 생성하고, $MAC_{MS} = f_K^1(N_{MS} \| T_{MS} \| ID_{SN})$을 계산한다. 이 때, 타임스탬프인 T_{MS}는 재전송 공격을 방지하기 위해 사용된다. MS는 SN에게 $cTID_{MS}$, ID_{HN}, N_{MS}, T_{MS}, MAC_{MS}를 전송한다. 이 때, $cTID_{MS}$는 현재 MS의 임시 ID로



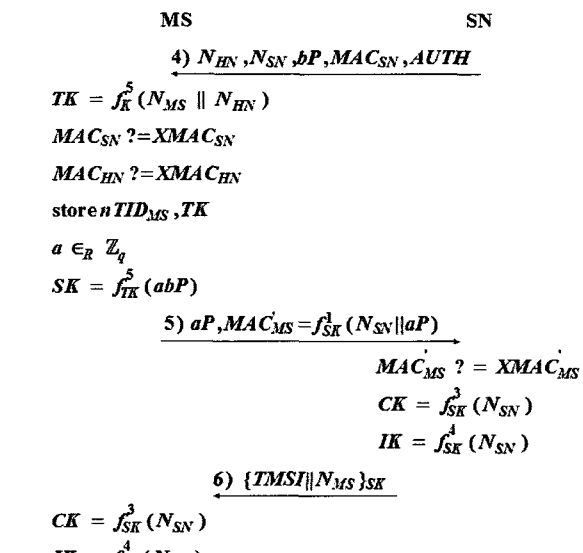
〔그림 4〕 초기 프로토콜 - 1 (MS-SN)



〔그림 5〕 초기 프로토콜 - 2 (SN-HN)

$f_K^2(IMSI)$이나 $f_K^2(pTID_{MS})$를 계산한 값이다. $pTID_{MS}$는 이전에 사용된 MS의 임시 ID를 나타낸다. 이렇게 매번 세션을 시작할 때 새로운 아이디를 발급하고 사용함으로써 프라이버시를 강화하였다.

- 단계 2. SN은 ID_{HN}을 확인하고, MS로부터 받은 메시지를 MS의 HN에게 전달한다. 이 때, ID_{HN}대신에 ID_{SN}을 포함시켜 보내게 된다. 이렇게 함으로써, HN은 MS와 SN이 보낸 ID_{SN}이 일치하는 지 확인할 수 있다.
- 단계 3. HN은 수신한 $cTID_{MS}$가 데이터베이스에 있는지 검색한다. 데이터베이스에는 각 가입자의 $IMSI$, $cTID_{MS}$, $pTID_{MS}$가 저장되어 있다. 이렇게 $pTID_{MS}$를 유지해 만약 동기화 문제가 발생하면 이전 아이디를 검색해봄으로써, TID_{MS}의 동기화 문제를 해결할 수 있다. HN은 MAC_{MS}를 검증하는데, 이 과정에서 SN으로부터 받은 ID_{SN}이 MAC_{MS}에 포함된 ID_{SN}과 맞는지 확인하고 타임스탬프인 T_{MS}의 유효성을 확인한다.
- 단계 4. HN은 새로운 $nTID_{MS} = f_K^2(cTID_{MS})$와 N_{HN}을 생성한다. $nTID_{MS}$는 데이터베이스에서 현재 임시 ID 필드에 저장되고, $cTID_{MS}$는 이전 임시 ID필드에 저장된다. HN은 $MAC_{HN} = f_K^1(N_{MS}\|N_{HN})$과 $TK = f_K^5(N_{MS}\|N_{HN})$를 계산하고, $IMSI$, N_{HN}, $AUTH = MAC_{HN}\|AMF$를 SN에게 보낸다. 이 데이터들은 가정한 것처럼 인증된 안전한 채널을 통해 보내지게 된다.
- 단계 5. SN은 ECDH 기법을 사용하기 위해 $b \in Z_q$를 선택하고 bP를 계산한다. 그 다음 N_{SN}을 생성하여 $MAC_{SN} = f_{TK}^1(N_{HN}\|N_{SN}\|bP)$를 계산한다. 마지막으로 N_{HN}, N_{SN}, bP, MAC_{SN}, $AUTH$를 MS에게 보낸다.

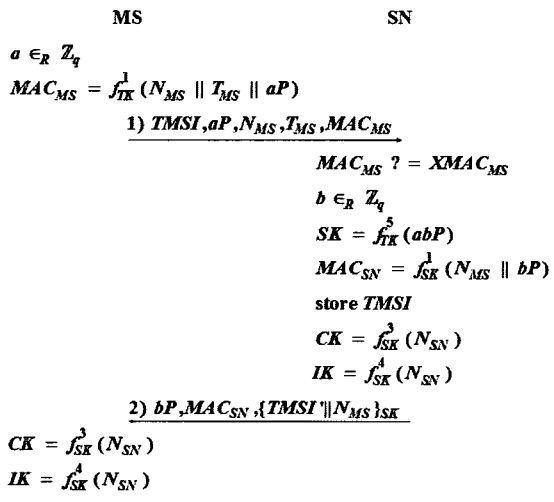


〔그림 6〕 초기 프로토콜 - 3 (MS-SN)

- 단계 6. MS는 먼저 MAC_{HN}을 검증하고, TK를 생성한다. 그리고 MS는 MAC_{SN}을 검증하고, 검증이 끝나면 다음에 사용할 임시 ID를 계산하고 TK와 함께 저장한다. 그 다음에 MS는 $a \in Z_q$를 선택해 aP와 세션키 $SK = abP$를 계산한다. 마지막으로 aP, $MAC'_{MS} = f_{SK}^1(N_{SN}\|aP)$를 SN에게 보낸다.
- 단계 7. SN은 SK를 계산하고 MAC'_{MS}를 검증한다. 그리고 실제 통신에 사용하게 될 ID인 $TMSI$를 생성해 암호화 메시지 $TMSI\|N_{MSSK}$를 계산해 MS에게 보낸다. 마지막으로 SN은 안전하게 통신을 하기 위한 $CK = f_{SK}^3(N_{SN})$, $IK = f_{SK}^4(N_{SN})$를 계산한다.
- 단계 8. MS는 SN으로부터 보내진 암호화 메시지를 복호화하고, SN과 마찬가지로 CK, IK를 계산한다. 그리고 MS와 SN은 이 키들을 이용해 암호화 및 무결성을 보장하게 되어 안전하게 통신을 할 수 있다.

다음은 초기 프로토콜 이후에 진행되는 MS와 SN 사이의 프로토콜로써, 〔그림 7〕은 프로토콜의 진행과정을 나타낸다.

- 단계 1. MS는 먼저 $a \in Z_q$를 선택하고 aP를 계산한다. 그 다음에 N_{MS}를 생성하고, $MAC_{MS} = f_{TK}^1$



〔그림 7〕 초기 프로토콜 이후에 진행되는 프로토콜

$(N_{MS} \| aP \| T_{MS})$를 계산한다. 마지막으로 MS는 $TMSI$, N_{MS}, T_{MS}, aP, MAC_{MS}를 SN에게 보내게 된다.

- 단계 2. SN은 $TMSI$로 TK를 검색해 MAC_{MS}를 검증한다. 검증이 끝나면 $b \in Z_q$를 선택하고 bP, abP를 계산한다. SN은 세션키 $SK = f^5_{TK}(abP)$를 생성하고, $MAC_{SN} = f^1_{SK}(N_{MS} \| bP)$를 계산한다. 그리고 새로운 $TMSI$도 생성하게 된다. 여기서도 동기화 문제 발생 시 해결하기 위해 이전에 사용되었던 $TMSI$를 유지하게 된다. 마지막으로 SN은 MS에게 bP, MAC_{SN}, $\{\mathrm{TMSI}' \| \mathrm{N_{MS}}\}_{\mathrm{SK}}$를 보낸다.
- 단계 3. MS도 K_{MS}를 계산하고, MAC_{SN}를 검증하게 된다. 이전 프로토콜과 마찬가지로 MS와 SN은 CK, IK를 생성해, 안전한 통신을 할 수 있게 된다.

4.3 안전성 분석

- MS와 HN의 상호 인증: HN은 MS와 HN이 미리 공유하고 있는 키 K로 생성한 MAC_{MS}을 통해 MS를 인증하며, MS는 동일한 키로 생성한 MAC_{HN}을 통해 HN을 인증한다. 따라서 사용되는 MAC이 안전하면 제안하는 프로토콜에서 MS와 HN은 안전하게 상호 인증할 수 있다.
- MS와 SN의 상호 인증: 먼저 초기 프로토콜에서는 MS는 SN의 상호 인증과 관련하여 다음과 같은 가정을 사용하고 있다.

가정 1. MS는 SN의 ID를 알 수 있다.

가정 2. HN은 프로토콜 진행에 나타나 있지 않지만 SN과 MAPsec이나 IPsec을 이용하여 안전하게 상호 인증한다. 이 때, HN은 MS가 MAC_{MS}에 포함된 ID를 활용한다.

따라서 SN은 HN으로부터 메시지를 받으면 HN이 MS를 인증하였다는 것을 알게 되며, SN은 HN을 신뢰하므로 SN 또한 MS를 인증하게 된다. 추가로 SN은 나중에 MAC_{MS}'를 통해 인증의 확신을 확대할 수 있다. 반면에 MS는 TK를 HN이 최근에 생성하였다는 것을 확인할 수 있다. 따라서 HN에 대한 신뢰를 바탕으로 TK로 생성된 MAC_{SN}을 통해 SN을 인증할 수 있다.

- 완전한 전방향 안전성 만족: 제안하는 프로토콜은 ECDH 기법을 이용해 MS와 SN간에 세션키를 생성함으로써 완전한 전방향 안전성을 만족하게 된다. 만약, K가 노출된다면 공격자는 이전 메시지를 통해 TK를 생성할 수 있다. 하지만 aP 또는 bP를 알아도 SK를 생성할 수 없다. 결론적으로 EC-CDHP에 의해 aP나 bP가 노출되어도 세션키에 사용되는 값 abP를 계산할 수 없기 때문에 완전한 전방향 안전성을 만족하게 된다. 또한, MS와 SN사이에 MAC값들을 통해 aP와 bP를 검증하기 때문에 중간자 공격에 대해서도 안전하다.
- 프라이버시 보호 강화: UMTS-AKA 프로토콜을 비롯한 이전에 제안된 프로토콜은 MS가 새로운 SN을 방문할 경우 적어도 한번은 IMSI를 평문형태로 전송해야 된다. 그러나 제안하는 프로토콜은 IMSI가 평문형태로 전송되지 않는다. MS는 항상 HN과 미리 공유하고 있는 비밀키 K를 이용해 새로 생성한 임시 ID인 TID_{MS}를 전송하기 때문에 이전 프로토콜과는 달리 프라이버시 보호가 강화된다.
- 재전송 공격: 제안하는 프로토콜에서는 MAC값에 포함된 타임스탬프와 난스를 통해 메시지의 최근성을 검증하기 때문에 공격자의 재전송 공격에도 안전하다.
- TID_{MS}의 동기화 공격: 먼저, MS가 SN을 통해 HN으로 보내는 메시지 1을 공격자가 막는다고 가정하자. MS는 HN으로부터 메시지 4를 받지 못하기 때문에 일정 시간이 지나게 되면 세션이 종료되고 새로운 세션을 시작하기 때문에 아무런 문제가 없

다. 그 다음으로 HN이 SN을 통해 MS에게 메시지 3을 보낼 때, 공격자가 이 메시지를 막는다고 가정하자. 이 때, HN은 MS의 새로운 $nTID_{MS}$를 갱신하게 된다. 반면에 MS는 HN의 메시지 4를 받지 못하기 때문에 이전에 사용된 $cTID_{MS}$를 그대로 유지하게 된다. 이 경우에 MS가 새로운 세션을 다시 시작하면, HN은 이전에 사용된 $cTID_{MS}$와 새롭게 갱신한 $nTID_{MS}$도 같이 유지하고 있기 때문에 MS를 식별하고 다시 동기화 할 수 있어 TID_{MS}의 동기화 공격에도 안전하다.

4.4 효율성 분석

- SN과 HN의 네트워크 대역폭 소모 감소: UMTS AKA 프로토콜처럼 인증 데이터 n개를 생성해 SN과 HN사이에 교환하는 것이 아니라 제안하는 프로토콜은 하나의 인증 데이터를 생성하기 때문에 보다 효율적이다. 그리고 TK_{HN}에 대한 사용기간이나 사용 횟수를 제한할 수 있다. 이 방식은 X-AKA와 비슷한 접근 방식을 가진다고 할 수 있다.
- SN의 저장공간 오버헤드 감소: SN은 각 MS에 대한 인증 데이터를 하나만 저장하기 때문에 X-AKA와 마찬가지로 저장공간에 대해 보다 효율적이다.
- 공개키 연산 사용: 완전한 전방향 안전성을 제공하기 위해서, ECDH 기법을 사용한다. 따라서 매번 MS는 프로토콜을 수행하기 위해서, 2번의 타원 곡선 연산을 사용하게 된다. 이전에 제안된 프로토콜은 MS의 계산 능력과 배터리 능력 제한으로 공개키 연산을 사용하지 않았지만, 최근에는 기술의 발달로 인해 공개키 연산을 사용하는 추세이다[6]. 제안하는 프로토콜은 인증서와 같은 것을 필요로 하지 않고, 한 번의 인증 시에 2번의 타원 곡선 연산만을 필요로 하기 때문에 MS에게 크게 부담이 가지 않는다. 뿐만 아니라 MS가 안전한 저장장치를 가지고 있으면 여러 개의 ECDH 일회용 공개키를 사전에 생성해 놓을 수도 있다.

V. 결 론

본 논문에서는 프라이버시를 강화하고 완전한 전방향 안전성을 제공하는 UMTS AKA 프로토콜을 제안하였다. 제안하는 프로토콜은 암호화된 임시 ID를 사용해 프라이버시를 강화하였고, ECDH 키 동의 프로토콜을 사용해 완전한 전방향 안전성을 제공하였다. 또한 MS와 HN 사이의 상호 인증과 MS와 SN에 대한 상호 인증을 각각 제공한다. 마지막으로 제안하는 프로토콜은 SN과 HN사이의 네트워크 대역폭 소모를 감소시키고, SN의 저장 공간 오버헤드를 감소시킨다.

참고문헌

[1] 3GPP TS 33.102 (v7.0.0), Security architecture, Release 7, 2005.

[2] M. Zhang and Y. Fang, "Security Analysis and Enhancement of 3GPP Authentication and Key Agreement Protocol," *IEEE Trans. on Wireless Communications*, Vol. 4, No. 2, pp.734-742, 2005.

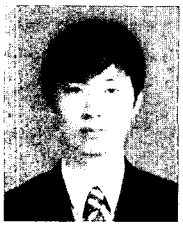
[3] L. Harn and W.J. HSin, "On the Security of Wireless Network Access with Enhancements," *Proc. of the ACM Workshop on Wireless Security*, pp.88-95, 2003.

[4] C. Huang and J. Li, "Authentication and Key Agreement Protocol for UMTS with Low Bandwidth," *Proc. of the 19th IEEE Conf. on AINA*, pp.392-397, 2005.

[5] U. Meyer and J. Li, "A Man-in-the-Middle Attack on UMTS," *Proc. of the ACM Workshop on Wireless Security*, pp. 90-97, 2004

[6] G. Kambourakis, A. Roukas, and S. Gritzalis, "Performance Evaluation of Public key based Authentication in Future Mobile Communication System," *EURASIP J. on Wireless Communications and Networking*, Vol. 2004, No. 1, pp. 184-1997, 2004

〈著者紹介〉



김 대 영 (Daeyoung Kim) 학생회원
2005년 2월: 한양대학교 전자컴퓨터공학부(학사)
2007년 2월: 한양대학교 컴퓨터공학과(석사)
2007년 3월~현재: (주) 엠파스 시스템본부 시스템개발부 ECS개발팀
<관심분야> 네트워크 보안
URL: http://kzero.net

최 용 강 (Yonggang Cui) 학생회원
2004년 7월: 대련민족학원 컴퓨터학부(학사)
2005년 8월~현재: 한양대학교 컴퓨터공학과 석사 과정
<관심분야> 네트워크 보안

김 상 진 (Sangjin Kim) 종신회원
1995년 2월: 한양대학교 전자계산학과(학사)
1997년 2월 한양대학교 전자계산학과(석사)
2002년 8월 한양대학교 전자계산학과(박사)
2003년 3월~현재: 한국기술교육대학교 인터넷미디어공학부 조교수
<관심분야> 암호기술 응용
URL:http://infosec.kut.ac.kr/sangjin/

오 희 국 (Heekuck Oh) 종신회원
1983년: 한양대학교 전자공학과(학사)
1989년: 아이오와주립대학 전자계산학과(석사)
1992년: 아이오와주립대학 전자계산학과(박사)
1993년~1994년: 한국전자통신연구원 선임연구원
1995년 3월~현재: 한양대학교 컴퓨터공학과 부교수
<관심분야> 암호프로토콜, 네트워크 보안
URL:http://infosec.hanyang.ac.kr/~hkoh/