情 報 保 護 學 會 誌
第 12 券 第 6 號, 2002. 12

cdma2000 1xEV-DO의 시큐리티 고찰

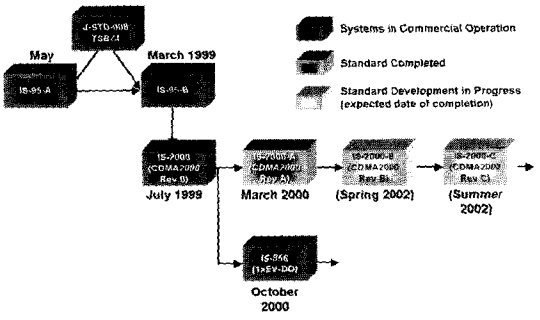
신 상 욱*, 류 희 수*, 정 교 일*

요 약

cdma2000 1xEV(Evolution)-DO(Data Only)는 기존의 IS-2000 무선 프로토콜과 달리 패킷 데이터 서비스를 위한 전용 프로토콜로, 최대 2.4Mbps의 전송 속도를 제공한다. cdma2000 1xEV-DO는 이전의 프로토콜과 달리 무선 인터페이스 계층 구조에 따로 분리된 시큐리티 계층(security layer)을 가지며, 이 시큐리티 계층에서 패킷 데이터 서비스를 위한 인증과 암호화 서비스를 제공한다. 본 고에서는 패킷 데이터 서비스를 위한 전용 프로토콜인 cdma2000 1xEV-DO의 시큐리티 계층의 4가지 프로토콜인 키 교환 프로토콜, 인증 프로토콜, 암호화 프로토콜, 시큐리티 프로토콜을 분석한다.

Ⅰ. 서 론

CDMA(Code Division Multiple Access)는 무선 구간에서 기지국과 단말기 사이의 신호 전송 방법이다. CDMA 이동통신 방식은 현재 미국 위주의 동기식과 유럽 위주의 비동기식으로 구분된다. 동기식 CDMA 이동통신 방식은 크게 2G인 IS-95 계열과 3G인 IS-2000 계열로 구분된다. IS-95는 무선 구간의 접속 방식을 CDMA로 최초로 정의한 무선 구간 프로토콜이다. IS-2000은 cdma2000 방식의 무선 구간 프로토콜로 IMT-2000을 지향하고 있다. [그림 1]은 CDMA 무선 인터페이스 표준을 보여준다[9].



[그림 1] CDMA 무선 인터페이스 표준

IS-2000[1-5]은 1x 또는 3x라고 불리며, Rev(Revision).0~Rev.B는 일반적인 음성과 데이터 서비스에 대해 최대 307kbps(1x), 1.04Mbps(3x)의 전송 속도를 지원한다. HDR(High Data Rate), HRPD(High Rate Packet Data), 1xEV(Evolution)-DO(Data Only)라고 불리는 IS-856[6]은 기존의 IS-2000 무선 프로토콜과는 다른 패킷 데이터 서비스를 위한 전용 프로토콜로, 최대 2.4Mbps의 전송 속도를 제공한다. IS-2000 Rev.C 이후의 기술은 1xEV-DV(Data and Voice)라고 불리며 현재 표준화가 진행 중이며, 3Mbps~5Mbps의 데이터 전송 속도를 제공할 예정이다.

cdma2000 1xEV-DO는 이전의 프로토콜과 달리 무선 인터페이스 계층 구조에 따로 분리된 시큐리티 계층(security layer)을 가진다. 이 시큐리티 계층에서 패킷 데이터 서비스를 위한 인증과 암호화 서비스를 제공한다. 본 고에서는 패킷 데이터 서비스를 위한 전용 프로토콜인 cdma2000 1xEV-DO의 시큐리티 계층을 분석한다. cdma2000 1xEV-DO의 시큐리티 계층은 키 교환 프로토콜, 인증 프로토콜, 암호화 프로토콜, 시큐리티 프로토콜로 구성되어 있다. 현재 3GPP2(3rd Generation Partnership Project 2)에서 표준 개발 중인 cdma2000 1xEV-DO 시큐리티 계층의 각 프로토콜에 대해 분석한다.

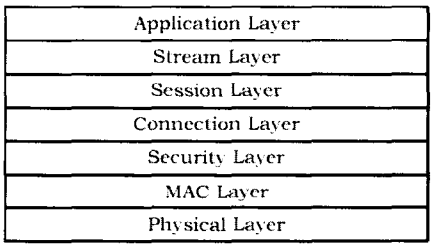
* 한국전자통신연구원 ({sinsu, hsryu, koil}@etri.re.kr)

먼저 2장에서는 cdma2000 1xEV-DO의 무선 인터페이스의 구조와 시큐리티 계층에 대해 간단히 기술하고, 3장에서는 시큐리티 계층에서 키 관리 프로토콜, 인증 프로토콜, 암호화 프로토콜, 시큐리티 프로토콜 각각에 대해 기술한다. 마지막 4장은 결론이다.

Ⅱ. cdma2000 1xEV-DO

2.1 일반적인 개요

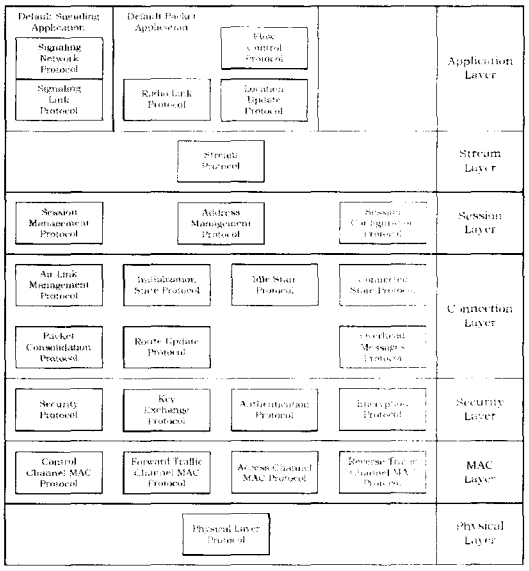
cdma2000 1xEV-DO 무선 인터페이스는 그림 2에 보여진 것처럼 계층화된 구조를 가진다. 이것은 계층 또는 프로토콜의 수정을 그 계층으로 고립화시켜준다. 각 계층은 하나 이상의 프로토콜로 구성되며, 이들 프로토콜 각각은 개별적으로 협상된다.



〔그림 2〕 무선 인터페이스 계층 구조

1. 응용 계층(application layer) : 응용 계층은 다양한 응응을 제공한다. 무선 인터페이스 프로토콜 메시지 전송을 위한 디폴트 시그널링 응용(default signaling application)을 제공한다. 또한 사용자 데이터 전송을 위한 디폴트 패킷 응용(default packet application)을 제공한다.
2. 스트림 계층(stream layer) : 스트림 계층은 다양한 응용 스트림의 다중화(multiplexing)를 제공한다. 스트림 0은 시그널링에 한정되고 디폴트 시그널링 응용에 디폴트이다. 스트림 1과 2, 3은 디폴트로 사용되지 않는다.
3. 세션 계층(session layer) : 세션 계층은 주소 관리, 프로토콜 협정, 프로토콜 구성, 상태 유지 관리 서비스를 제공한다.
4. 연결 계층(connection layer) : 연결 계층은 무선 링크 연결 설정과 관리 서비스를 제공한다.
5. 시큐리티 계층(security layer) : 시큐리티 계층은 인증과 암호화 서비스를 제공한다.
6. MAC(medium access control) 계층 : MAC 계층은 물리 계층으로 수신하고 전송하기 위해 사용되는 절차를 정의한다.
7. 물리 계층(physical layer) : 물리 계층은 forward와 reverse 채널을 위한 채널 구조, 주파수, 파워 출력, 변조(modulation), 인코딩 방법을 제공한다.

각 계층은 하나 이상의 프로토콜로 구성된다. 프로토콜들은 무선 링크의 peer 개체에게 정보를 전달하기 위해 시그널링 메시지 또는 헤더를 사용한다. 프로토콜들이 메시지를 전송할 때 이들 메시지들을 전송하기 위해 Signaling Network Protocol (SNP)를 사용한다. 〔그림 3〕은 각 계층에 정의된 디폴트 프로토콜들을 보여준다.



〔그림 3〕 디폴트 프로토콜

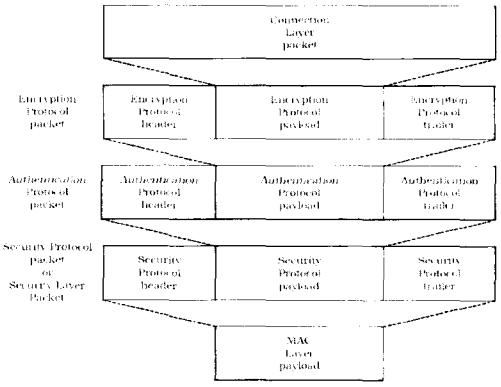
2.2 시큐리티 계층

무선 인터페이스는 제어 채널, 액세스 채널, forward 트래픽 채널, reverse 트래픽 채널에 의해 전송되는 액세스 터미널 트래픽의 인증과 암호화를 위해 사용될 수 있는 시큐리티 계층을 지원한다.

시큐리티 계층(security layer)은 키 교환(key exchange), 인증(authentication), 암호화(encryption)

기능을 제공한다. 시큐리티 계층은 이들 기능들을 제공하기 위해 키 교환 프로토콜, 인증 프로토콜, 암호화 프로토콜, 시큐리티 프로토콜을 사용한다.

- 키 교환 프로토콜 : 인증과 암호화를 위한 시큐리티 키 교환을 위해 액세스 네트워크(access network: AN)와 액세스 터미널(access terminal: AT)에 의해 수행되는 절차를 제공한다.
- 인증 프로토콜 : 트래픽 인증을 위해 액세스 네트워크와 액세스 터미널에 의해 수행되는 절차를 제공한다.
- 암호화 프로토콜 : 트래픽 암호화를 위해 액세스 네트워크와 액세스 터미널에 의해 수행되는 절차를 제공한다.
- 시큐리티 프로토콜 : 인증 프로토콜과 암호화 프로토콜에 의해 사용될 수 있는 cryptosync, timestamp와 같은 공개 파라메터 생성 절차를 제공한다.



〔그림 4〕 시큐리티 계층 캡슐화

〔그림 4〕는 연결 계층 패킷, 시큐리티 계층 패킷, MAC 계층 패킷들간의 관계를 보여준다. 세션 구성이 디폴트 시큐리티 계층을 설정하거나 또는 구성된 시큐리티 프로토콜이 헤더(header) 또는 트레일러(trailer)를 요구하지 않는다면, 시큐리티 계층 헤더 또는 트레일러는 나타나지 않을 수 있다. MAC 계층에 의해 추가되는 필드들이 시큐리티 계층 헤더와 트레일러의 존재를 지시한다. 암호화 프로토콜은 평문의 실제 길이를 숨기기 위해 트레일러를 추가할 수 있고 암호화 알고리즘에 의해 사용되는 패딩을 추가할 수도 있다. 암호화 프로토콜 헤더는 초기화 벡터와 같은 변수들을 포함할 수 있다. 인증 프로토콜 헤더 또는 트레일러는 인증되는 인증 프로토콜 패킷 부분을 인증하기 위해 사용되는 전자 서명을 포함할 수 있다. 시큐리티 프로토콜 헤더 또는 트레일러는 인증과 암호화 프로토콜에 의해 필요한 변수들을 포함할 수 있다.

〔그림 4〕는 인증과 암호화되는 시큐리티 계층 패킷의 부분을 보여준다. 인증은 암호화 프로토콜 패킷에 대해 수행된다. 이것은 인증이 실패할 때 불필요한 복호화를 피할 수 있다.

Ⅲ. 시큐리티 계층의 프로토콜

3.1 시큐리티 프로토콜

3.1.1 디폴트 시큐리티 프로토콜

디폴트 시큐리티 프로토콜은 인증 프로토콜과 MAC 계층간에 패킷 전송을 제외하면 어떤 서비스를 제공하지 않는다. 이 프로토콜의 PDU(protocol data unit)는 시큐리티 계층 패킷이다. 각 시큐리티 계층 패킷은 인증 프로토콜 패킷으로 구성된다. 디폴트 시큐리티 프로토콜은 헤더와 트레일러를 추가하지 않는다.

3.1.2 Generic 시큐리티 프로토콜

Generic 시큐리티 프로토콜은 다음의 작업을 수행한다.

- 전송시에 이 프로토콜은 인증 프로토콜과 암호화 프로토콜에 의해 사용될 수 있는 cryptosync를 제공한다.
- 수신시에 이 프로토콜은 시큐리티 프로토콜 헤더에 포함된 정보를 사용하여 cryptosync를 계산하여 cryptosync를 공개적으로 활용가능하게 한다.

이 프로토콜의 공개 데이터는 Cryptosync이고, PDU는 시큐리티 계층 패킷이다. 각 시큐리티 계층 패킷은 인증 프로토콜 패킷과 시큐리티 프로토콜 헤더로 구성된다.

프로토콜은 인증 프로토콜 패킷으로부터 시큐리티 계층 패킷을 구성해서 MAC 계층에게 패킷을 전달해야 한다.

- 프로토콜이 인증되거나 암호화되는 인증 프로

토콜 패킷을 인증 프로토콜로부터 수신하면, 시큐리티 프로토콜 헤더에서 CryptosyncShort를 인증 프로토콜 또는 암호화 프로토콜에 의해 사용되는 Cryptosync 값의 최하위 16비트로 설정해야 한다. 시큐리티 프로토콜은 80ms 단위로 CDMA 시스템 시간의 64 비트 표현에 기반하여 Cryptosync 값을 선택해야 한다. Cryptosync는 시큐리티 계층 패킷이 물리 계층에 의해 전송될 시간보다 더 이후의 시간을 명시하지 말아야 하고 현재 CDMA 시스템 시간보다 더 이른 시간을 명시하지 말아야 한다. 시큐리티 프로토콜은 인증 프로토콜 패킷의 앞에 시큐리티 프로토콜 헤더를 추가해야 한다.

- 프로토콜이 인증되지 않거나 암호화되지 않는 인증 프로토콜 패킷을 인증 프로토콜로부터 수신하면, 프로토콜은 인증 프로토콜 패킷에 시큐리티 프로토콜 헤더를 추가하지 말아야 한다.
- 이 프로토콜은 인증 프로토콜 패킷에 시큐리티 프로토콜 트레일러를 추가하지 말아야 한다.

이 시큐리티 프로토콜은 MAC 계층으로부터 수신된 시큐리티 계층 패킷을 사용하여 다음처럼 인증 프로토콜 패킷을 구성해서 인증 프로토콜로 패킷을 전달해야 한다.

- 프로토콜이 인증되거나 암호화되는 시큐리티 프로토콜 패킷을 MAC 계층으로부터 수신하면, 시큐리티 프로토콜 헤더에 주어진 Cryptosync Short를 사용하여 64 비트 Cryptosync를 계산하고, 시큐리티 프로토콜 헤더를 제거하여 인증 프로토콜 패킷을 구성해야 한다.

  Cryptosync =
  (SystemTime - (SystemTime[15:0] - CryptosyncShort) mod 2^{16}) mod 2^{64}

  여기서 SystemTime은 80ms 단위로 현재 CDMA 시스템 시간이고, SystemTime[15:0]은 SystemTime의 최하위 16비트이다.
- 프로토콜이 인증되지 않거나 암호화되지 않는 시큐리티 프로토콜 패킷을 MAC 계층으로부터 수신하면, 시큐리티 계층 프로토콜을 인증 프로토콜 패킷으로 설정해야 한다.

3.2 키 교환 프로토콜

3.2.1 디폴트 키 교환 프로토콜

디폴트 키 교환 프로토콜은 어떤 서비스를 제공하지 않고 디폴트 인증 프로토콜과 디폴트 암호화 프로토콜이 선택되면 디폴트 키 교환 프로토콜이 선택된다. 이 프로토콜은 다른 계층 또는 프로토콜 대신 payload를 전달하지 않는다.

3.2.2 DH 키 교환 프로토콜

DH 키 교환 프로토콜은 Diffie-Hellman에 기반한 세션 키 교환을 위한 방법을 제공한다. 이 프로토콜의 공개 데이터는 다음과 같다.

- FACAuthKey와 길이 : forward assigned channel(즉 forward 트래픽 채널)에 사용되는 인증 키
- RACAuthKey와 길이 : reverse assigned channel(즉 reverse 트래픽 채널)에 사용되는 인증 키
- FACEncKey와 길이 : forward assigned channel(즉 forward 트래픽 채널)에 사용되는 암호화 키
- RACEncKey와 길이 : reverse assigned channel(즉 reverse 트래픽 채널)에 사용되는 암호화 키
- FPCAuthKey와 길이 : forward public channel(즉 제어 채널)에 사용되는 인증 키
- RPCAuthKey와 길이 : reverse public channel(즉 액세스 채널)에 사용되는 인증 키
- FPCEncKey와 길이 : forward public channel(즉 제어 채널)에 사용되는 암호화 키
- RPCEncKey와 길이 : reverse public channel(즉 액세스 채널)에 사용되는 암호화 키

이 프로토콜은 구성 메시지들의 처리를 정의하기 위해 Generic Configuration Protocol을 사용한다. 액세스 터미널과 액세스 네트워크가 KeyLength attribute를 위한 값에 동의하면, 액세스 터미널과 액세스 네트워크는 다음을 수행한다.

- SKey를 0으로 설정하고 그 길이를 Keylength attribute 값으로 설정한다.
- FACAuthKey, RACAuthKey, FACEncKey, RACEncKey, FPCAuthKey, RPCAuthKey, FPCEncKey, RPCEncKey를 0으로 설정하고 그 길이를 160으로 설정한다.

키 교환 프로토콜은 공개 세션키 교환을 위해 KeyRequest와 KeyResponse 메시지를 사용하

고, 비밀 세션키를 계산했다는 것을 지시하기 위해 ANKeyComplete와 ATKeyComplete 메시지를 사용한다.

(1) DH 키 교환 절차

액세스 터미널과 액세스 네트워크는 세션 구성 과정동안 다음의 키 교환 절차를 수행해야 한다.

A. 액세스 터미널

KeyRequest 메시지를 수신하면, 액세스 터미널은 다음을 수행한다.

- 액세스 터미널은 KeyLength와 $2^{KeyLength}-2$ 사이의 랜덤한 정수 ATRand를 선택하여 KeyResponse 메시지의 ATPubKey 필드를 다음처럼 설정한다.

$$ATPubKey = g^{ATRand} \bmod p$$

여기서 g와 p는 DH 키 교환 프로토콜을 위한 KeyLength 의존 프로토콜 상수이다. KeyLength는 DH 키 교환 프로토콜의 세션 구성 과정동안 명시된다.
- 액세스 터미널은 KeyRequest 메시지 수신의 $T_{KEPATResponse}$ 초 이내에 KeyResponse 메시지를 전송해야 한다.
- 액세스 터미널은 다음처럼 세션키 SKey를 계산해야 한다.

$$SKey = ANPubKey^{ATRand} \bmod p$$

액세스 터미널은 KeyResponse 메시지를 전송한 후에 $T_{KEPKeyCompAN}$를 KeyRequest 메시지로 액세스 네트워크에 의해 보고된 Timeout에 의해 명시된 시간으로 설정한다. 액세스 터미널은 $T_{KEPKeyCompAN}$ Timeout 값으로 AN Key Computation Timer를 시작한다. 액세스 터미널은 관련된 KeyRequest와 KeyResponse 메시지의 TransactionID 필드와 일치하는 TransactionID를 가진 ANKeyComplete 메시지를 수신하면 AN Key Computation Timer를 disable 시킨다.

AN Key Computation Timer가 만료되면, 액세스 터미널은 오류를 선언해야 한다.

관련된 KeyRequest와 KeyResponse 메시지의 TransactionID 필드와 일치하는 TransactionID를 가진 ANKeyComplete 메시지를 수신하면, 액세스 터미널은 다음을 수행해야 한다.

- 액세스 터미널은 다음처럼 64비트 TimeStamp Long을 계산해야 한다.

TimeStampLong =
(SystemTime - (SystemTime[15:0] - TimeStampShort) mod 2^{16}) mod 2^{64}

여기서 SystmeTime은 80ms 단위로 현재 CDMA 시스템 시간이고 SystemTime[15:0]은 SystemTime의 최하위 16비트이다. TimeStampShort는 ANKeyComplete 메시지로 수신된 16비트 필드이다.
- 액세스 터미널은 계산된 SKey와 TimeStam Long 그리고 ANKeyComplete 메시지의 TransactionID와 Nonce 필드를 사용하여 [표 1]에 보여진 것과 같은 message bits를 구성한다.

[표 1] message bits

필드	길이(비트)
SKey	KeyLength
TransactionID	8
Nonce	16
TimeStampLong	64

- 액세스 터미널은 이전 단계에서 구성된 message bits를 SHA-1[7]을 사용하여 160-비트 메시지 다이제스트를 계산한다.
- 계산된 메시지 다이제스트가 ANKeyComplete 메시지의 KeySignatrue 필드와 일치하면, 액세스 터미널은 다음 두가지 이벤트 이후의 $T_{KEPSigCompAT}$ 초 이내에 1로 설정된 Result 필드를 가진 ATKeyComplete 메시지를 전송한다.
  - ANKeyComplete 메시지의 수신
  - SKey 계산 완료
- 그렇지 않다면, 액세스 터미널은 오류를 선언하고 0으로 설정된 Result 필드를 가진 ATKeyComplete 메시지를 전송한다.

B. 액세스 네트워크

액세스 네트워크는 KeyRequest 메시지를 전송하여 키 교환을 개시해야 한다. 액세스 네트워크는 KeyLength와 $2^{KeyLength}-2$ 사이의 랜덤한 정수 ANRand를 선택하여 다음처럼 KeyRequest 메시지의 ANPubKey 필드를 설정한다.

$$ANPubKey = g^{ANRand} \bmod p$$

여기서 g, p, KeyLength는 DH 키 교환 프로토콜의 세션 구성 과정동안 명시된다.

액세스 네트워크가 관련된 KeyRequest 메시지의 TransactionID 필드와 일치하는 TransactionID를 가진 KeyResponse 메시지를 $T_{KEPANResponse}$ 내에 수신하지 못하면, 액세스 네트워크는 오류를 선언하고 키 교환 절차를 중단한다.

관련된 KeyRequest 메시지의 TransactionID 필드와 일치하는 TransactionID를 가진 KeyResponse 메시지를 수신한 후에 액세스 네트워크는 다음을 수행한다.

- 액세스 네트워크는 KeyResponse 메시지로 액세스 터미널에 의해 보고된 Timeout에 명시된 시간으로 $T_{KEPKeyCompAT}$ 를 설정한다. 액세스 네트워크는 $T_{KEPKeyCompAT}$ timeout 값으로 AT Key Computation Timer를 시작한다.
- 액세스 네트워크는 다음처럼 세션키 SKey를 계산한다.

  $SKey = ATPubKey^{ANRand} \bmod p$
- 액세스 네트워크는 계산된 SKey, TimeStamp Long, TransactionID, Nonce 필드를 사용하여 〔표 2〕처럼 message bits를 구성한다. TimeStampLong은 80ms 단위의 현재 CDMA 시스템 Time의 64비트 표현에 기반하여 설정된 64비트 값이다. TimeStampLong은 물리 계층에 의해 전송될 시간보다 더 이후의 시간을 명시하지 말아야 하고 현재 CDMA 시스템 Time보다 더 이른 시간을 명시해서도 않된다.

〔표 2〕 message bits

필드	길이(비트)
SKey	KeyLength
TransactionID	8
Nonce	16
TimeStampLong	64

- 액세스 네트워크는 이전 단계에서 구성된 message bits를 SHA-1을 사용하여 160-비트 메시지 다이제스트를 계산한다.
- 액세스 네트워크는 이전 단계에서 계산된 message bits를 KeySignature 필드로 설정하고 CDMA 시스템 Time의 하위 16비트를 TimeStampShort로 설정한 ANKeyComplete 메시지를 전송한다. 액세스 네트워크는 $T_{KEPSigCompAN}$ 의 timeout 값을 가진 AT Signature Timer를 시작한다.

관련된 KeyRequest와 KeyResponse 메시지의 TransactionID와 일치하는 TransactionID를 가진 ATKeyComplete 메시지를 수신하면, 액세스 네트워크는 AT Key Computation Timer와 AT Key Signature timer 모두를 diable 해야 한다.

액세스 네트워크는 다음 이벤트 중의 하나가 발생하면 오류를 선언하고 키 교환 절차를 중단한다.

- AT Key Computation Timer와 AT Key Signature timer 모두가 만료된다.
- 0의 값으로 설정된 Result 필드를 가진 ATKeyComplete 메시지를 수신한다.

(2) 인증 키와 암호화 키 생성

인증과 암호화를 위해 사용되는 키들은 세션키 SKey로부터 생성된다. 표 3은 SKey의 8개 서브 필드를 정의한다. 서브 필드들의 길이는 같다.

〔표 3〕 SKey의 서브 필드

서브 필드	길이(비트)
K0	KeyLength / 8
K1	KeyLength / 8
K2	KeyLength / 8
K3	KeyLength / 8
K4	KeyLength / 8
K5	KeyLength / 8
K6	KeyLength / 8
K7	KeyLength / 8

액세스 네트워크와 액세스 터미널은 표 4에 보여진 것처럼 message bits를 구성한다. 여기서 TimeStampLong, Nonce는 KeySignature 생성에 사용된 것과 동일하다.

액세스 터미널과 액세스 네트워크는 message bits들을 SHA-1을 사용하여 160비트 메시지 다이제스트를 계산한다. 계산된 메시지 다이제스트들을 FACAuthKey, RACAuthKey, FACEncKey, RACEncKey, FPCAuthKey, RPCAuthKey, FPCEncKey, RPCEncKey로 설정한다.

〔표 4〕 인증과 암호화 키 생성을 위한 message bits

	MSB		LSB
FACAuthKey 생성을 위한 message bits	K0 (KeyLength / 8)	Nonce (16비트)	TimeStampLong (64비트)
RACAuthKey 생성을 위한 message bits	K1 (KeyLength / 8)	Nonce (16비트)	TimeStampLong (64비트)
FACEncKey 생성을 위한 message bits	K2 (KeyLength / 8)	Nonce (16비트)	TimeStampLong (64비트)
RACEncKey 생성을 위한 message bits	K3 (KeyLength / 8)	Nonce (16비트)	TimeStampLong (64비트)
FPCAuthKey 생성을 위한 message bits	K4 (KeyLength / 8)	Nonce (16비트)	TimeStampLong (64비트)
RPCAuthKey 생성을 위한 message bits	K5 (KeyLength / 8)	Nonce (16비트)	TimeStampLong (64비트)
FPCEncKey 생성을 위한 message bits	K6 (KeyLength / 8)	Nonce (16비트)	TimeStampLong (64비트)
RPCEncKey 생성을 위한 message bits	K7 (KeyLength / 8)	Nonce (16비트)	TimeStampLong (64비트)

(3) 메시지 포맷

- KeyRequest 메시지 : 액세스 네트워크는 세션 키 교환을 개시하기 위해 KeyRequest 메시지를 전송한다.

필드	길이(비트)
MessageID	8
TransactionID	8
Timeout	8
ANPubKey	KeyLength

  - MessageID : 액세스 네트워크는 이 필드를 0x00으로 설정한다.
  - TransactionID : 액세스 네트워크는 각각의 새로운 KeyRequest 메시지에 대해 이 값을 증가시켜야 한다.
  - Timeout : 공유 비밀 계산 timeout. 액세스 네트워크는 이 필드를 세션키 계산을 위해 액세스 네트워크가 요구하는 최대 시간을 초 단위로 설정한다.
  - ANPubKey : 액세스 네트워크의 ephemeral 공개 Diffie-Helleman 키
- KeyResponse 메시지 : 액세스 터미널은 KeyRequest 메시지에 대한 응답으로 Key Response 메시지를 전송한다.

필드	길이(비트)
MessageID	8
TransactionID	8
Timeout	8
ATPubKey	KeyLength

  - MessageID : 액세스 터미널은 이 필드를 0x01로 설정한다.
  - TransactionID : 액세스 터미널은 액세스 터미널이 응답하는 KeyRequest 메시지의 TransactionID 필드의 값으로 이 필드를 설정한다.
  - Timeout : 공유 비밀 계산 timeout. 액세스 터미널은 이 필드를 세션키 계산을 위해 액세스 터미널이 요구하는 최대 시간을 초 단위로 설정한다.
  - ATPubKey : 액세스 터미널의 ephemeral Diffie-Helleman 공개키
- ANKeyComplete 메시지 : 액세스 네트워크는 KeyResponse 메시지에 대한 응답으로 ANKeyComplete 메시지를 전송한다.

필드	길이(비트)
MessageID	8
TransactionID	8
Nonce	16
TimeStampShort	16
KeySignature	160

  - MessageID : 액세스 네트워크는 이 필드를 0x02로 설정한다.
  - TransactionID : 액세스 네트워크는 대응하는 KeyRequest 메시지의 TransactionID 값으로 이 필드를 설정한다.
  - Nonce : 액세스 네트워크는 KeySignature 계산을 위해 사용되는 임의로 선택된 16비트 nonce로 이 필드를 설정한다.
  - TimeStampShort : KeySingature 계산에 사용된 SystemTimeLong의 하위 16비트로 이 필드를 설정한다.
  - KeySignature : 세션키의 20바이트 서명 값
- ATKeyComplete 메시지 : 액세스 터미널은

ANKeyComplete 메시지에 대한 응답으로 ATKeyComplete 메시지를 전송한다.

필드	길이(비트)
MessageID	8
TransactionID	8
Result	1
Reserved	7

- MessageID : 액세스 터미널은 이 필드를 0x03로 설정한다.
- TransactionID : 액세스 터미널은 대응하는 KeyRequest 메시지의 TransactionID 값으로 이 필드를 설정한다.
- Result : 액세스 터미널은 ANKeyComplete 메시지의 KeySignature 필드 값이 유효하다면 이 필드를 1로 설정하고 그렇지 않다면 0으로 설정한다.
- Reserved : 액세스 터미널은 이 필드를 0으로 설정한다. 액세스 네트워크는 이 필드를 무시한다.

(4) Configuration attribute와 프로토콜 상수들

Configurable attribute들은 표 5에 보여진다. 액세스 터미널은 〔표 5〕에서의 이탤릭체로 표시된 값들을 디폴트로 사용한다.

〔표 5〕 configurable values

attribute ID	attribute	values	meaning
0x00	Session Key Length (KeyLength)	*0x00*	디폴트로 96바이트(768비트) Diffie-Hellman 키. KeyLength = 768
		0x01	128바이트(1024비트) Diffie-Hellman 키. KeyLength = 1024
		0x02 ~ 0xff	Reserved

〔표 6〕은 프로토콜의 수치 상수들을 보여준다.

〔표 7〕과 〔표 8〕은 DH 키 교환 프로토콜에 사용되는 g와 p 상수 값들이다. 이 값들은 IETF의 IKE[8]에 정의된 것을 그대로 사용한다.

〔표 6〕 프로토콜 수치 상수

constant	meaning	value
$N_{KEPType}$	키 교환 프로토콜의 Type 필드	0x05
N_{KEPDH}	키 교환 프로토콜의 subtype 필드	0x0001
$T_{KEPSigCompAN}$	ANKeyComplete 메시지를 전송한 후에 ATKeyComplete 메시지를 수신하기까지의 time	3.5초
$T_{KEPSigCompAT}$	ANKeyComplete 메시지를 수신한 후에 ATKeyComplete 메시지를 전송하기까지의 time	3초
$T_{KEPANResponse}$	KeyRequest 메시지를 전송한 후에 KeyResponse 메시지를 수신하기까지의 time	3.5초
$T_{KEPATResponse}$	KeyRequest 메시지를 수신한 후에 KeyResponse 메시지를 전송하기까지의 time	3초

〔표 7〕 768비트 KeyLength에 대한 common primitive base와 common prime modulus

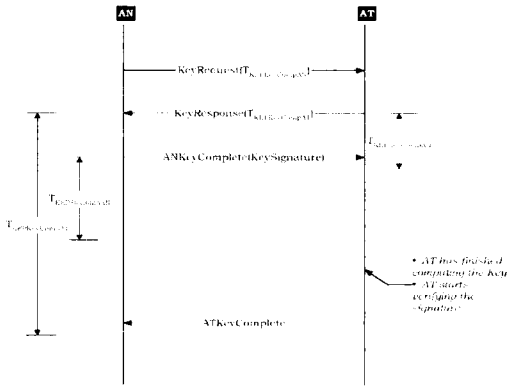
constant	meaning	value
g	common primitive base	0x02
p	common prime modulus (MSB first)	0xFFFFFFFF 0xFFFFFFFF 0xC90FDAA2 0x2168C234 0xC4C6628B 0x80DC1CD1 0x29024E08 0x8A67CC74 0x020BBEA6 0x3B139B22 0x514A0879 0x8E3404DD 0xEF9519B3 0xCD3A431B 0x302B0A6D 0xF25F1437 0x4FE1356D 0x6D51C245 0xE485B576 0x625E7EC6 0xF44C42E9 0xA63A3620 0xFFFFFFFF 0xFFFFFFFF

〔표 8〕 1024비트 KeyLength에 대한 common primitive base와 common prime modulus

constant	meaning	value
g	common primitive base	0x02
p	common prime modulus (MSB first)	0xFFFFFFFF 0xFFFFFFFF 0xC90FDAA2 0x2168C234 0xC4C6628B 0x80DC1CD1 0x29024E08 0x8A67CC74 0x020BBEA6 0x3B139B22 0x514A0879 0x8E3404DD 0xEF9519B3 0xCD3A431B 0x302B0A6D 0xF25F1437 0x4FE1356D 0x6D51C245 0xE485B576 0x625E7EC6 0xF44C42E9 0xA637ED6B 0x0BFF5CB6 0xF406B7ED 0xEE386BFB 0x5A899FA5 0xAE9F2411 0x7C4B1FE6 0x49286651 0xECE65381 0xFFFFFFFF 0xFFFFFFFF

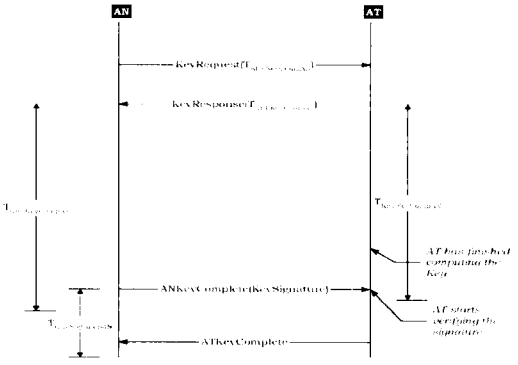
(5) 메시지 흐름

[그림 5]는 액세스 네트워크가 빨리 키와 서명을 계산하여 액세스 터미널에게 전송하는 flow diagram의 예를 보여준다. 액세스 터미널은 키 계산을 완료하기 위해 시간이 필요하다. 이 경우 AT Signature Computation Timer가 만료되지만 AT Key Computation Timer는 만료되지 않는다.



[그림 5] 호 흐름(call flow)의 예: 타이머 $T_{KEPSigCompAN}$ 는 만료되지만 $T_{KEPKeyCompAT}$ 는 만료되지 않는다.

[그림 6]은 액세스 네트워크가 키 계산을 위해 더 긴 시간을 요구하는 flow diagram의 예를 보여준다. 이 경우 AT Key Computation Timer는 만료하지만 AT Signature Computation Timer는 만료되지 않는다.



[그림 6] 호 흐름(call flow)의 예: 타이머 $T_{KEPSigCompAN}$ 는 만료되지 않지만 $T_{KEPKeyCompAT}$ 는 만료된다.

3.3 인증 프로토콜

3.3.1 디폴트 인증 프로토콜

디폴트 인증 프로토콜은 암호화 프로토콜과 시큐리티 프로토콜 사이의 패킷 전송을 제외하면 아무런 서비스도 제공하지 않는다. 이 프로토콜의 PDU는 인증 프로토콜 패킷이다.

암호화 프로토콜 패킷을 수신하면, 시큐리티 프로토콜로 전달한다. 시큐리티 프로토콜로부터 시큐리티 프로토콜 패킷을 수신하면, 암호화 프로토콜 패킷을 인증 프로토콜 패킷으로 설정하여 암호화 프로토콜로 전달한다. 디폴트 인증 프로토콜은 헤더와 트레일러를 추가하지 않는다.

3.3.2 SHA-1 인증 프로토콜

SHA-1 인증 프로토콜은 ACAuthKey, 시큐리티 계층 payload, CDMA System Time, sector ID로 구성된 message bits를 SHA-1 해쉬함수에 적용하여 액세스 채널 MAC 계층 패킷의 인증 방법을 제공한다. 이 프로토콜의 PDU는 인증 프로토콜 패킷이다.

암호화 프로토콜 패킷을 수신하면, 각 액세스 채널 암호화 프로토콜 패킷 앞에 인증 계층 헤더를 추가하여 액세스 채널 인증 프로토콜 패킷을 만들어 시큐리티 프로토콜로 전달한다.

프로토콜이 시큐리티 계층으로부터 액세스 채널 시큐리티 프로토콜 패킷을 수신하면, 인증 프로토콜 헤더를 제거하여 암호화 프로토콜을 구성하여 암호화 프로토콜로 전달한다.

(1) 인증 절차

액세스 채널에 의해 전달되는 패킷에 다음의 인증 절차를 적용한다. 다른 패킷들에 대해서는 디폴트 인증 프로토콜을 적용한다.

A. 액세스 터미널

액세스 채널로 향하는 암호화 프로토콜 패킷을 수신하면 액세스 터미널은 다음처럼 ACPAC(Access Channel Packet Authentication Code)를 계산한다.

- 액세스 터미널은 다음처럼 ACAuthKey를 구성한다.
  - 키 교환 프로토콜이 공개 데이터로 RPCAuth Key를 정의하지 않으면, 액세스 터미널은

ACAuthKey를 ACAuthKeyLength에 의해 명시된 길이의 0으로 설정한다.

- 그렇지 않으면, 액세스 터미널은 다음을 수행한다.
  - RPCAuthKey의 길이가 ACAuthKey의 길이와 같다면, ACAuthKey는 RPCAuthKey이다.
  - RPCAuthKey의 길이가 ACAuthKey의 길이보다 크다면, ACAuthKey는 RPCAuthKey의 하위 ACAuthKeyLength 비트이다.
  - RPCAuthKey의 길이가 ACAuthKey의 길이보다 작다면, ACAuthKey는 ACAuthKey Length가 되도록 RPCAuthKey의 끝(LSB)을 0으로 채운다.

• 액세스 터미널은 다음처럼 Cryptosync를 구성한다.

- 시큐리티 프로토콜이 공개 데이터로 Cryptosync를 정의하지 않는다면, 액세스 터미널은 Cryptosync 필드를 0으로 설정한다. 그렇지 않다면, 시큐리티 프로토콜에 의해 공개 데이터로 주어진 값을 사용한다.
- 시큐리티 프로토콜이 공개 데이터로 Cryptosync Length를 정의하지 않는다면, 액세스 터미널은 64로 설정한다. 그렇지 않다면, 시큐리티 프로토콜에 의해 공개 데이터로 주어진 값을 사용한다.

• 액세스 터미널은 다음의 표9처럼 ACPAC 계산을 위한 message bits를 구성한다.

〔표 9〕 ACPAC 계산을 위한 message bits

필드	길이(비트)
ACAutheKey	ACAuthKeyLength
인증 프로토콜 payload	가변
SectorID	128
Cryptosync	CryptosyncLength

여기서 SectorID는 Overhead Message Protocol에 의해 공개 데이터로 제공된다.

• 액세스 터미널은 위에서 구성된 message bits를 SHA-1에 적용하여 160비트 메시지 다이제스트를 계산한다. 메시지 다이제스트의 하위 64비트를 ACPAC 필드로 설정한다.

B. 액세스 네트워크

액세스 채널로부터 인증 프로토콜 패킷을 수신하면, 액세스 네트워크는 액세스 채널 MAC 계층 패킷 인증 코드(ACPAC)를 계산하여 인증 프로토콜 헤더에 주어진 ACPAC를 검증한다.

• 액세스 네트워크는 다음처럼 ACAuthKey를 구성한다.

- 키 교환 프로토콜이 공개 데이터로 RPCAuth Key를 정의하지 않으면, 액세스 터미널은 ACAuth Key를 ACAuthKeyLength에 의해 명시된 길이의 0으로 설정한다.
- 그렇지 않으면, 액세스 터미널은 다음을 수행한다.
  - RPCAuthKey의 길이가 ACAuthKey의 길이와 같다면, ACAuthKey는 RPCAuthKey이다.
  - RPCAuthKey의 길이가 ACAuthKey의 길이보다 크다면, ACAuthKey는 RPCAuthKey의 하위 ACAuthKeyLength 비트이다.
  - RPCAuthKey의 길이가 ACAuthKey의 길이보다 작다면, ACAuthKey는 ACAuthKey Length가 되도록 RPCAuthKey의 끝(LSB)을 0으로 채운다.

• 액세스 네트워크는 다음처럼 Cryptosync를 구성한다.

- 시큐리티 프로토콜이 공개 데이터로 Cryptosync를 정의하지 않는다면, 액세스 터미널은 Cryptosync 필드를 0으로 설정한다. 그렇지 않다면, 시큐리티 프로토콜에 의해 공개 데이터로 주어진 값을 사용한다.
- 시큐리티 프로토콜이 공개 데이터로 Cryptosync Length를 정의하지 않는다면, 액세스 터미널은 64로 설정한다. 그렇지 않다면, 시큐리티 프로토콜에 의해 공개 데이터로 주어진 값을 사용한다.

• 액세스 네트워크는 표 9처럼 ACPAC 계산을 위한 message bits를 구성한다.

• 액세스 터미널은 위에서 구성된 message bits를 SHA-1에 적용하여 160비트 메시지 다이제스트를 계산한다. 메시지 다이제스트의 하위 64비트를 ACPAC 필드로 설정한다.

계산된 ACPAC가 프로토콜 헤더의 ACPAC 필드 값과 일치하면, 인증 계층 payload를 암호화 프로토콜로 전달한다. 그렇지 않다면, 프로토콜은 Failed indication을 이슈하고 시큐리티 계층 패킷을 버린다.

(2) 인증 프로토콜 헤더와 configuration attributes

SHA-1 인증 프로토콜 헤더는 다음과 같다.

필드	길이(비트)
ACPAC	64

SHA-1 인증 프로토콜은 트레일러를 추가하지 않는다. SHA-1 인증 프로토콜의 configurable attribute들은 [표 10]에 보여진다. 프로토콜 수치 상수들은 [표 11]과 같다.

[표 10] configurable values

attribute ID	attribute	values	meaning
0x00	ACAuthKey Length	0x00A0	인증 키 길이의 디폴트 값 (비트 길이)
		0x0000 ~ 0xFFFF	액세스 채널 인증 키 길이 (비트 길이)

[표 11] 프로토콜 수치 상수

constant	meaning	value
$N_{AIPType}$	인증 프로토콜의 Type 필드	0x06
$N_{AIPSHA1}$	인증 프로토콜의 subtype 필드	0x0001

3.4 암호화 프로토콜

현재 암호화 프로토콜은 디폴트 암호화 프로토콜만이 정의되어 있다. 디폴트 암호화 프로토콜은 시큐리티 계층 패킷 payload를 변경하지 않으며, 암호화 프로토콜 헤더와 트레일러를 추가하지 않는다. 즉, 이 프로토콜의 암호문은 연결 계층 패킷과 동일하다. 필요하다면 end-to-end 암호화가 응용 계층에서 제공될 수 있다. 이 프로토콜의 수치 상수는 다음과 같다.

[표 12] 프로토콜 수치 상수

constant	meaning	value
N_{EPType}	암호화 프로토콜의 Type 필드	0x07
$N_{EPDefault}$	암호화 프로토콜의 subtype 필드	0x0000

Ⅳ. 결 론

본 고에서는 cdma2000 1xEV-DO의 시큐리티를 분석하였다. cdma2000 1xEV-DO는 이전의 무선 프로토콜과 달리 시큐리티 계층을 분리시켜 패킷 데이터에 대한 인증과 암호화 서비스를 제공한다. 시큐리티 계층은 키 교환 프로토콜, 인증 프로토콜, 암호화 프로토콜, 시큐리티 프로토콜로 구성된다. 키 교환 프로토콜은 DH 키 교환 프로토콜이 정의되어 있으며, 인증 프로토콜은 SHA-1 기반 인증 프로토콜이 정의되어 있다. 현재 암호화 프로토콜은 정의되어 있지 않다.

향후 연구로 제한된 성능을 가지는 AT에 768비트와 1024비트의 DH 키 교환 프로토콜이 적절한지 좀더 분석되어야 하며, 이에 대한 대안으로 ECDH를 고려해볼 수 있으며, 이 경우 DH와 ECDH에 대한 면밀한 비교 분석이 수행되어야 할 것이다. 또한 현재 정의되어 있지 않은 암호화 프로토콜에 대한 연구가 계속 수행되어야 할 것이다.

참 고 문 헌

[1] 3GPP2 C.S0001, "cdma2000 - Introduction"
[2] 3GPP2 C.S0002, "Physical Layer Standard for cdma2000 Spread Spectrum Systems"
[3] 3GPP2 C.S0003, "Medium Access Control (MAC) Standard for cdma2000 Spread Spectrum Systems"
[4] 3GPP2 C.S0004, "Signaling Link Access Control (LAC) Standard for cdma2000 Spread Spectrum Systems"
[5] 3GPP2 C.S0005, "Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems"
[6] 3GPP2 C.S0024, "cdma2000 High Rate Packet Data Air Interface Specification"
[7] FIPS PUB 180-1, Federal Information Processing Standards Publication 180-1
[8] IETF RFC 2409, The Internet Key Exchange (IKE)
[9] Frank Quick, "Security in cdma2000", ITU-T Workshop on Security, Seoul (Korea), 13-14 May, 2002

〈著 者 紹 介〉

신 상 욱(Sang Uk Shin)

1995년 2월 : 부산수산대학교(현 부경대학교) 전자계산학과 (학사)
1997년 2월 : 부경대학교 전자계산학과(석사)
2000년 2월 : 부경대학교 전자계산학과(박사)
2000년 4월~현재 : 한국전자통신연구원 선임연구원
관심분야 : 정보보호론, 컴퓨터 보안

류 희 수 (Heuisu Ryu)
정회원

1990년 2월 : 고려대학교 수학과 (학사)
1992년 2월 : 고려대학교 수학과 (석사)
1999년 5월 : Johns Hopkins University 수학과(박사)
2000년 7월~현재 : 한국전자통신연구원 선임연구원
관심분야 : 타원곡선 암호, 대수학, 이동통신 정보보호

정 교 일 (Kyo-il Chung)
정회원

1981년 : 한양대학교 전자공학과 (공학사)
1983년 : 한양대학교 산업대학원 전자계산학과 (공학석사)
1997년 : 한양대학교 대학원 전자공학과 (공학박사)
1982년~현재 : 한국전자통신연구원 정보보호연구본부 정보보호기반연구부장 / 책임연구원
관심분야 : IC Card, Security, Biometrics, 국가기반보호, 신호처리