通信情報保護學會誌
第5卷 第3號, 1995. 9

CDMA 이동 통신을 위한 효율적인 인증 시스템 실현

염 흥 열*, 이 만 영**

요 약

본 고에서는 CDMA 이동 통신망에 적용 가능한 인증 시스템을 제시한다. 이를 위하여 먼저 IS-95에 표준화된 인증과 관련된 파라메타와 인증 절차를 분석하고, 인증 시스템에 응용될 수 있는 해쉬 함수의 개념과 원리를 제시하며, 이를 바탕으로 이동 통신에 적용 가능한 해쉬 함수에 바탕을 둔 인증 시스템을 제시하고, 제시된 인증시스템을 시뮬레이션한다.

제 1 장 IS-95 인증 파라메타 분석 및 인증 절차

본 장에서는 IS-95에 기초한 CDMA 이동 통신망에서의 인증[1-4] 관련 파라메타와 관련 채널, 인증 과정, 그리고 대표적인 호 처리 절차를 제시한다.[5]

1.1 인증 관련 파라메타

인증과 관련된 파라메타는 가입자의 전화번호와 관련된 *MIN(mobile identification* number), 이동 단말기 자체에 할당된 ESN (electronic serial number), 이동국의 비밀 영역에 저장되어 있는 인증용으로 이용되는 SSD__A(shared secret data__A)와 음성 보호용으로 이용되는 SSD__B, 그리고 기지국에서 최근에 보내온 32비트의 난수 RAND 등이 있다.

* 종신회원, 순천향대학교 공과대학 전자공학과
** 종신회원, 한양대학교 전자통신공학과,
한국통신정보보호학회 회장

비트 위치 :

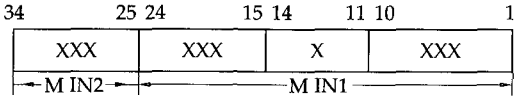


그림 1.1 MIN 구성

MIN은 34비트의 2진수로 10디지트(digit)의 전화번호(telephone number)로 부터 구해지며, 그림 1.1과 같이 10비트의 MIN2와 24비트의 MIN1으로 구성된다. 10디지트의 전화번호는 $D_1D_2D_3$ - $D_4D_5D_6$ - $D_7D_8D_9D_{10}$일때, 10디지트의 전화번호로 부터 MIN을 구하는 부호화 법칙은 다음과 같다.

① 전화번호의 첫 3디지트($D_1D_2D_3$)는 10bit의 MIN2를 생성한다. 디지트 '0'은 10진수 값 10으로 취급한다. MIN2는 $100D_1 + 10D_2 + D_3 - 111$ 계산한 후, 10진수로 표현된 결과를 2진수로 변환한 10비

트 값이다.

② 두번째 3디지트($D_4D_5D_6$)를 이용하여 MIN1이 계산된다. MIN1은 $100\,D_4 + 10\,D_5 + D_6 - 111$을 계산한 수, 10진수로 표현된 결과를 2진수로 변환한 10비트 값이다. 이 10비트 값이 MIN1의 상위 MSB(most significant bits)을 구성한다.

③ 전화번호의 마지막 4디지트($D_7D_8D_9D_{10}$)를 이용하여 MIN1의 하위 14 LSB(least significant bits)는 구해진다. 10진수로 표현된 D_7는 4비트의 2진수로 변환되어 14비트의 MIN의 상위 4비트를 구성한다. 나머지 3디지트($D_8D_9D_{10}$)는 과정 ①의 부호화 법칙에 따라 10비트로 부호화되어 나머지 하위 10비트가 된다.

예를 들어 전화번호가 321-456-7890인 경우의 MIN은 다음과 같은 절차로 구해진다. 먼저 MIN2는 다음과 같은 절차로 구해진다.

① $D_1 = 3,\ D_2 = 2,\ D_3 = 1$

② $100D_1 + 10D_2 + D_3 - 111 = 321 - 111 = 210$

③ 210를 2진수로 변환하면 "00 1101 0010"가 되며 이 값이 MIN2가 된다.

다음 MIN1는 다음과 같은 절차로 구해진다.

① 456 - 111 = 345 -> '0101 0110 01'

② 7 -> '0111'

③ ㉠ 890 -> 8 * 100 + 9 * 10 + 1 * 10 = 900

㉡ 900 - 11 = 789

㉢ 789 --> "11 0001 0101"

㉣ MIN1

= '0101 0110 0101 1111 0001 0101'

두번째 인증 파라메타인 ESN은 32비트의 2진수로서 이동국을 유일하게 확인한다. 이는

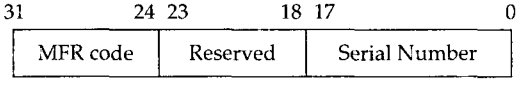


그림 1.2 ESN 구조

공장에서 세트되며, 변경 불가능한 값이다. ESN의 일반적 구조는 그림 1.2와 같다.

MFR 부호는 제조업자 부호이며 ESN의 상위 8비트 MSB를 구성한다. 6비트는 유보되어 있고, 일련번호(serial number)는 18비트로 구성된 제품의 일련번호이다.

SSD는 이동국내에 저장된 128비트의 특정 패턴으로서 기지국에서도 이용 가능한 정보이다. 이동국은 SSD를 반영구 메모리(semi-permanent memory)에 비밀스럽게 저장해야 한다. SSD는 인증용으로 이용되는 SSD_A와, CDMA 음성 정보보호용으로 이용되는 SSD_B 등이 있다.

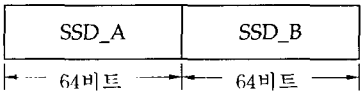


그림 1.3 SSD의 구성

random challenge data(RAND)은 기지국에서 보내온 인증을 위한 난수 값으로서 이동국내에 저장해야 할 32비트의 2진수이다. 이는 최근의 페이징 채널의 access parameters message내의 RAND 필드 의 값이다. RAND는 기지국에 의해 생성되는 32비트의 random challenge value 값으로서, 등록(registration) 인증, 이동국 발호 인증, 이동국 착호 인증 과정 등에서 이용된다.

AUTH는 페이징 채널의 'system parameter overhead message'내에 있는 정보 요소이다. AUTH = "01"인 경우 이동국이 기지국과 인증 과정을 완료한 후 서비스 액세스가 가능하고, AUTH = "00"인 경우 이동국은 인증 과정 수행 없이도 서비스 액세스가 가능함을 의미한다.

call history parameter(COUNT)는 이동국에서 유지되는 modulo-64 계수기(counter)로서, 이동국과 기지국이 서로 유지하며, 인증 과정이 수행될 때마다 일씩 증가한다. 이는 paging

channel에서 'parameter update order' 메시지가 수신되면 갱신된다.

A-key는 64비트 길이이며, 이동국에 할당되어 영구 보관되며, 이는 비밀 영역인 인증 메모리부에 저장되며, 이동국과 home location register/authentication center (HLR/AC)에서만 알려진 값이다.

1.2 인증을 위한 채널 분석

본 절에서는 유일-도전 응답 인증 과정에서 사용되는 reverse access channel에서의 'authentication challenge response message'의 구조를 제시한다. 인증 과정에서 이용되는 나머지 채널의 구조도 이와 거의 유사하다. 'authentication challenge response message'는 다음과 같은 서브필드 들로 구성된다.

MSG_TYPE : message type
ACK_seq : acknowledgement sequence number
MSG_seq : message sequence number
ACK_req : acknowledgement required indicator
VALID_ACK : valid acknowledgement indicator
ACK_TYPE : acknowledgement address type
MSID_TYPE : mobile station identifier field type
MSID_LEN : mobile station identifier field length
MSID : mobile station identifier
reserved : 6 bits

이동국은 ACK_TYPE 필드에 응답이 요구된 최근에 수신한 paging channel의 Addr_TYPE을 설정한다. MSID는 mobile station identifier로서 24bits의 MIN1, 10bits의 MIN2, 또는 32bits의 ESN로 구성된다. MSID_TYPE = 000이고 MSID_LEN = 9인 경우, MSID는 다음과 같이 설정된다.

MIN1 : 24bits
MIN2 : 10bits
ESN : 32bits
reserved : 6bits
전체 : 72bits(9byts)

1.3 인증 과정

인증 과정은 이동국의 정체성(identity)을 확인하기 위하여 이동국과 기지국간의 교환되는 일련의 프로토콜 과정이다. CDMA에서의 인증은 기본적으로 대칭형 암호 알고리듬 또는 해쉬 함수에 바탕을 두고 수행될 수 있다. 인증 과정은 이동국과 기지국이 동일한 비밀 공유정보(SSD : shared secret data)를 공유할 때만 완수되는 메시지 인증 방식에 기초한다. 인증 과정은 이동국이 기지국에 등록(registration)할때 수행되는 등록 인증, 호를 개시(originating) 할때 수행되는 발호 인증, 호를 수신(terminating)할때 사용되는 착호 인증, 상기의 과정들 중 하나가 실패했을 때 수행되는 유일-도전 응답(unique-challenge response) 인증 과정, 그리고 유일-도전 응답과정이 실패했을때 이동국과 기지국이 가지고 있는 비밀 공유 정보(SSD)를 갱신하기 위한 SSD갱신과 기지국 도전(base station challenge) 인증 과정이 있다.

이동국은 자신의 정체를 반드시 기지국에 인증 받은 후에 기지국에 등록할 수 있다. 등록 인증을 위한 인증 정보(AUTHR)의 계산은 RAND, ESN, MIN1, 그리고 SSD_AUTH필드로 구성되는 정보를 이용하며, 각 필드의 구체적인 값은 그림 1.4와 같다.

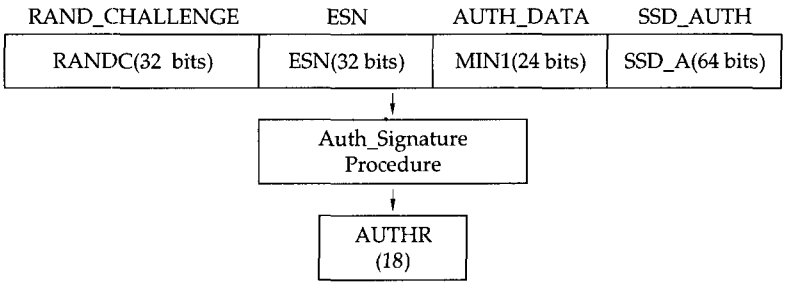


그림 1.4 AUTHR 서명문 계산 과정

등록 인증은 페이징 채널의 'system parameter overhead message' 내에 정보 요소 AUTH가 "01"인 메시지를 수신하고, 이동국이 기지국에 등록을 하려 할 때 수행된다.

등록을 위한 인증 과정에서 이용되는 RANDC는 paging channel의 'access parameter message' 내의 값이고, 등록 정보는 reverse access channel의 'registration message'를 이용한다.

등록인증 과정은 다음과 같이 수행된다.

① Authr_signature 과정의 입력 파라메타들을 설정한다.
② Save_register = false로 설정한다.
③ 그림 1.4와 같은 AUTHR 계산 과정을 이용하여 18-bit의 출력 Authr_signature를 계산한다.
④ COUNT, RANDC(RAND의 8 MSB 비트), AUTHR를 reverse 채널의 registration message를 이용하여 기지국으로 송신한다.

기지국은 'registration message'를 수신한 후, 다음 과정은 수행해야 한다.

① 수신된 RANDC, COUNT를 MIN/ESN를 관련된 내부 저장 값과 비교한다.
② 내부에 저장된 SSD_A의 값을 이용하여 AUTHR를 계산한다.
③ 이동국으로 부터 수신된 AUTHR과 기지국에서 계산된 AUTHR값을 비교한다.
④ 기지국은 모든 비교가 성공적으로 수행되었다면 이동국의 인증 시도가 성공된 것으로 판단한다. 성공하지 않으면 유일-도전 응답 과정을 수행한다.

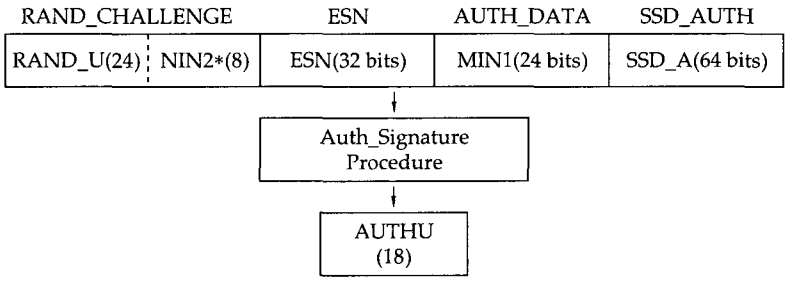


그림 1.5 AUTHU 계산 과정

상기 과정을 실패하면 기지국은 이동국에 유일-도전 응답 인증 과정을 수행해야 한다. 이동국에서 유일-도전 응답 인증을 위한 AUTHU는 그림 1.5와 같이 기지국에서 수신한

24비트의 RAND_U, MIN2의 8 MSB인 MIN2*, ESN, MIN1 그리고 SSD_A 정보를 이용하여 계산된다. 유일-도전 응답 인증 과정은 paging/access channel 또는 forward/reverse traffic channel 상에서 수행된다. unique challenge-response 과정은 다음과 같은 절차로 수행된다.

① 기지국은 임의의 난수 RAND_U를 생성한 후, paging channel의 'authentication message'의 정보 요소 RANDU를 이용하여 이동국에 전송한다.
② 이동국은 save_register = false로 한다.
③ 이동국은 그림 1.5와 같이 18비트의 AUTHU를 계산한다.
④ 이동국은 access channel의 'authentication challenge response message'에 AUTHU를 전송한다.
⑤ 기지국은 수신된 AUTHU와 자신이 가지고 있는 비밀 정보를 이용하여 계산된 AUTHU를 비교한다. 두 값이 동일하지 않으면 기지국은 이동국에 의해 시도된 더이상의 액세스를 거절할 뿐만 아니라 현재 진행중인 호를 절단한다. 그리고 기지국은 SSD 갱신 및 기지국 도전 인증 과정을 개시한다.

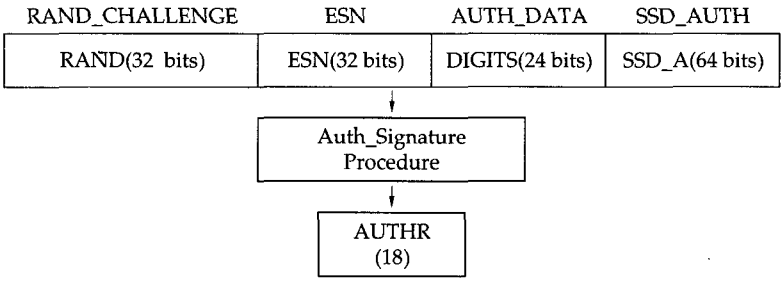


그림 1.6 발호 인증을 위한 AUTHR 계산 과정

이동국은 호를 시작할 때마다 발호(origination) 인증 과정을 수행해야 한다. 이동국 발호 인증을 위한 인증 정보(AUTHR)는 그림 1.6과 같이 RAND, ESN, DIGITS, 그리고 SSD_A를 이용하여 계산된다. RAND는 paging channel의 access parameter message의 RAND부이며, DIGITS는 'origination message'의 첫 6디지트를 이용한다.

이동국은 paging channel의 access parameter의 신호 요소 AUTH = "01"이고 호를 개시할 때 다음과 같은 절차로 등록 인증을 수행한다.

① 이동국은 입력 변수들 RAND, ESN, DIGITS, 그리고 SSD_A를 설정한다.
② 이동국은 18비트 AUTHR를 계산한다.
③ 이동국은 access channel에 AUTHR, Count, 그리고 RANDC 필드를 담은 origination 메시지를 기지국으로 전송한다.
④ 기지국은 내부에 저장되어 있는 값과 수신된 RANDC 값을 비교한다.
⑤ 기지국은 수신된 값과 저장된 COUNT 값을 비교한다.
⑥ 기지국은 자신이 저장하고 있는 정보를 이용하여 AUTHR를 계산하고, 이를 비교한다.
⑦ 기지국은 모든 비교들이 성공하면 트래픽 채널 할당 절차를 개시한다. 하나라도 성공하지 않은 경우, unique challenge-response나 SSD 갱신 절차를 수행한다.

이동국은 호를 수신할 때마다 호 수신을 위한 인증 과정을 수행해야 한다. 이동국 호 수신을 위한 인증 정보는 그림 1.7과 같이 계산된다.

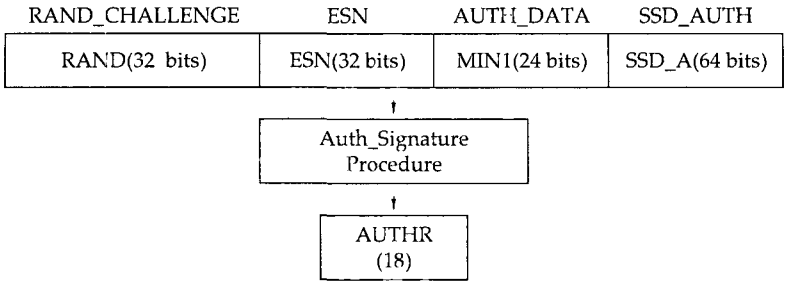


그림 1.7 착호 인증을 위한 AUTHR 계산 과정

paging channel의 access parameter의 신호 요소 AUTH = "01"이면 이동국은 호를 수신할 때마다 호 수신 인증 과정을 수행해야 한다. 호 수신을 위한 인증은 기지국에서 이동국으로 향하는 'page message'와 이동국에서 기지국으로 향하는 access channel의 'page response message'를 이용한다. 호 수신 인증 과정은 다음과 같다.

① 이동국은 그림 1.7과 같이 Auth_Signature 과정의 입력 파라메타를 셋한다.
② 이동국은 save_register = TRUE 한다.
③ 이동국은 18비트의 AUTHR를 계산한다.
④ 이동국은 access channel의 page response channel의 RANDC, COUNT, 그리고 AUTHR 필드를 채우고, 이를 기지국으로 전송한다.
⑤ 기지국은 내부에 저장되어 있는 RANDC 값과 수신된 RANDC값을 비교한다.
⑥ 기지국은 COUNT값을 비교한다.
⑦ 기지국은 자신이 저장하고 있는 정보를 이용하여 계산한 AUTHR과 수신된 AUTHR를 비교한다.
⑧ 기지국은 모든 비교들이 성공하면 채널 할당 절차를 개시하라. 하나라도 성공하지 않은 경우, unique challenge-response나 SSD 갱신절차를 수행한다.

shared secret data(SSD)의 갱신(update)과 base station challenge 인증 과정은 상기의 인증 과정이 실패한 경우에 수행된다. 이는 가입자의 비밀키중 하나인 A-key를 이용한다.

SSD 갱신 절차와 base station challenge 인증 과정은 그림 1.8과 같다.

① SSD 갱신이 필요한 경우, 기지국은 32비트의 난수 RANDSSD를 포함하는 SSD갱신 정보를 이동국에 전송한다.
② 이동국과 기지국은 그림 1.9와 같이 RANDSSD와 A_key를 이용하여 SSD_generation 과정을 수행하여 SSD_A_New와 SSD_B_New를 생성한다.
③ 이동국은 임의로 선택된 난수 RANDBS를 'base station challenge order message'를 이용하여 기지국에 전송한다.
④ 이동국은 그림 1.10과 같이 SSD-generation 과정에서 생성된 SSD_A_New와 RANDBS를 이용하여 Auth_signature 과정을 수행한 후, 18비트의 AUTHBS를 생성한다. 기지국도 SSD-generation 과정에서 생성된 SSD_A_New와 수신된 RANDBS를 이용하여 Auth_signature 과정을 수행

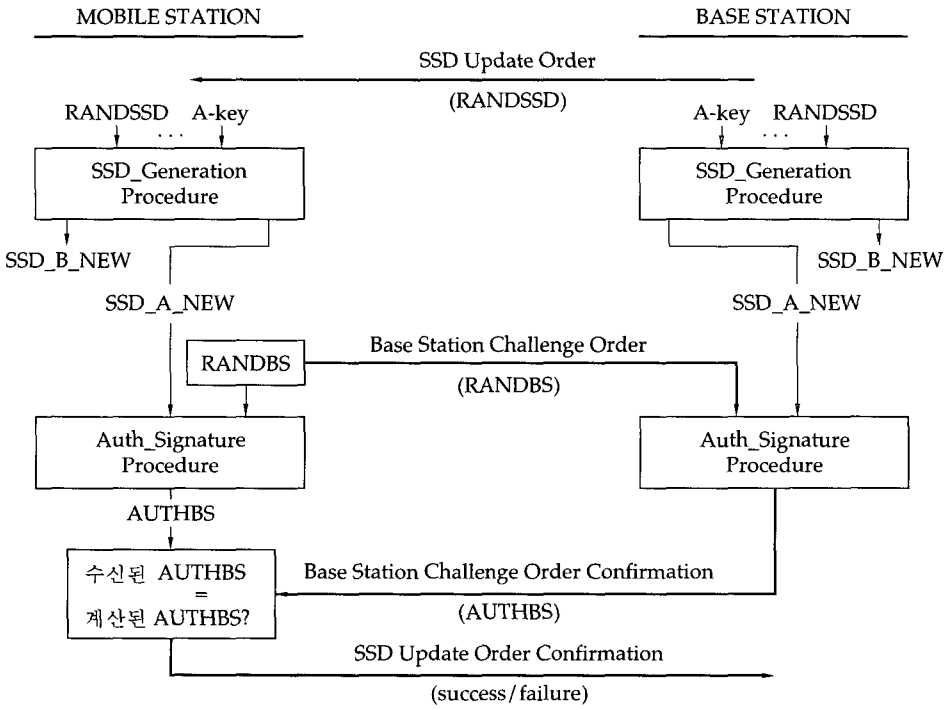


그림 1.8 SSD 갱신을 위한 신호 흐름도

한 후 18비트의 AUTHBS를 생성한다.

⑤ 기지국은 생성된 AUTHBS 정보를 'base station challenge order confirmation' 메시지를 이용하여 이동국에 전송한다.

⑥ 이동국은 수신된 AUTHBS와 계산된 AUTHBS를 비교한다. 같으면 SSD_A와 SSD_B를 갱신하고, 비교 결과만을 'SSD update order confirmation' 메시지를 이용하여 송신한다. 다르면 'SSD update reject order'를 송신한다.

⑦ 기지국은 'SSD update rejection order'를 수신하면 SSD update가 실패인 것으로 간주하고, 'SSD update confirmation order'를 수신하면 새로 생성된 SSD_A와 SSD_B로 갱신한다.

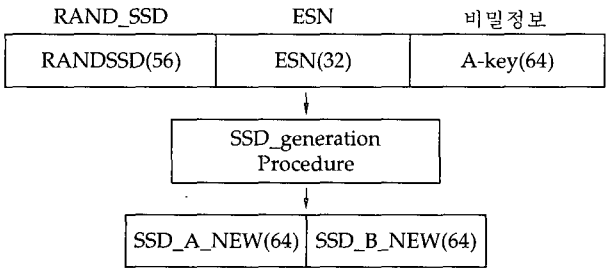


그림 1.9 새로운 SSD 생성 과정

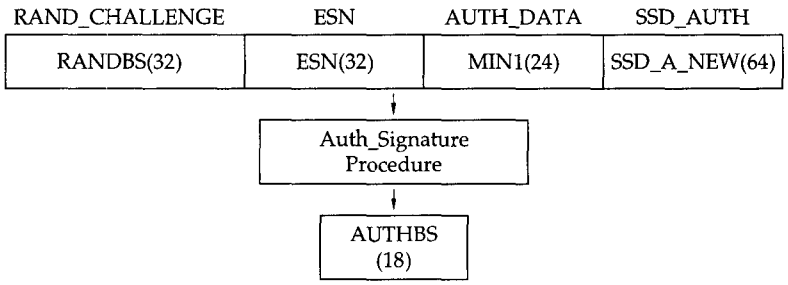


그림 1.10 AUTHBS 계산 과정

1.4 호 처리 절차

본 절에서는 이동국에서 호를 발신하는 절차와 이동국에서 호를 수신하는 절차 등을 기술하며, 인증과정이 포함되는 단계를 분명히 한다.

이동국 발신 호 처리는 다음과 같은 절차로 수행된다.

① 사용자의 호 시도를 검출한 이동국은 ESN, MIN, 수신 이동국 dialed digits, 서비스 형태, 그리고 인증 정보 등을 포함하는 'origination message'를 access channel로 전달한다. 여기서 등록 인증 정보가 포함된다.

② 이동국의 'origination message'를 수신한 기지국은 origination 인증 과정을 완료하고 여분의 트래픽 채널을 해당 가입자에 할당하고, 16개의 '1'과 8개의 '0'으로 구성된 'null traffic channel data'를 전송한다. 그리고 paging channel로 ESN, CDMA channel, 그리고 code channel 등을 포함하는 'channel assignment message'를 전송한다.

③ 'channel assignment message'를 수신한 이동국은 해당 메시지를 해석한 후, public long code를 이용하여 forward traffic channel를 수신할 준비를 한다. 유효한 N_{5m}번의 트래픽 채널로부터의 데이타를 수신한 이동국은 reverse traffic channel 상에 'traffic channel preamble'를 전송한다.

④ 기지국은 reverse traffic channel로부터 'traffic channel preamble'를 인식한 후, forward traffic channel로 'base station acknowledgement order' 메시지를 전송한다.

⑤ 이를 수신한 이동국은 'base station acknowledgement order' 메시지를 해석한 후, reverse traffic channel로 'null traffic channel data'를 전송한다.

⑥ 이를 수신한 기지국은 forward traffic channel로 'service option response order'를 전송한다. 이는 등록시에 등록 정보에 포함되어 있는 서비스 요구에 대한 응답이다.

⑦ 이를 수신한 이동국은 'service option response order'를 해석한 후, 서비스 선택 1인 경우 다음의 과정을 수행한다.

㉠ 이동국은 reverse traffic channel로 'origination continuation message'를 전송한다. 이를 수신한 기지국은 이에 대한 응답 프레임을 이동국으로 전송한다.

㉡ 기지국은 호 수신자에 링 신호를 전달하는 경우, 이동국에 'alert with information message(ring back

tone)' 를 전송한다. 이를 수신한 이동국은 오디오 경로로 ring back tone를 연결하여 이 상태를 사용자에 알린다.

㉢ 기지국은 호 수신자가 수화기를 hook-off하면 이 상태를 이동국에 'alert with information message(tone off)' 를 이용하여 전달한다. 이를 수신한 이동국은 사용자의 오디오 경로로 연결되어 있는 ring back tone를 제거한다. 그리고 이에 대한 응답 프레임을 전송한다.

⑧ 이후 트래픽 채널을 이용하여 두 사용자는 통화를 수행한다.

이동국에 호를 수신하는 절차는 다음과 같다.

① 기지국은 paging channel로 MIN을 포함하는 'page message' 나 'slotted page message' 를 전송한다. 이는 이동국 등록시에 결정된다.

② 이를 수신한 이동국은 access channel로 MIN과 ESN를 포함하는 'page response message' 를 전송한다. 여기서 착호 인증 과정 정보들이 포함된다.

③ 이를 수신한 기지국은 'page response message' 를 해석한 후, 남아있는 forward traffic channel를 준비하고, 여기로 16개의 '1' 과 8개의 '0' 으로 구성된 'null traffic channel data' 를 전송한다. 이 과정에서 기지국은 이동국의 정당성을 확인한다. 그리고 paging channel로 ESN, CDMA channel, code channel 등을 포함하는 'channel assignment message' 를 전송한다.

④ 'channel assignment message' 를 수신한 이동국은 해당 메시지를 해석한 후, public long code를 이용하여 forward traffic channel를 수신할 준비를 한다. 유효한 N_{5m}번의 트래픽 채널로 부터의 데이타를 수신한 이동국은 reverse traffic channel상에 'traffic channel preamble' 를 전송한다.

⑤ 기지국은 reverse traffic channel로부터 'traffic channel preamble' 를 인식한 후, forward traffic channel로 'base station acknowledgement order' 를 전송한다.

⑥ 이를 수신한 이동국은 'base station acknowledgement order' 메시지를 해석한 후, reverse traffic channel로 null traffic channel data를 전송한다.

⑦ 이를 수신한 기지국은 forward traffic channel로 'service option response order' 를 전송한다.

⑧ 이를 수신한 이동국은 'service option response order' 를 해석한 후, 서비스 선택 1인 경우 다음의 과정을 수행한다.

㉠ 기지국은 호 수신자에 링 신호를 전달하는 경우, 이동국에 'alert with information message (ring)' 를 전송한다. 이를 수신한 이동국은 오디오 경로로 ring 신호를 연결하여 이 상태를 사용자에 알린다.

㉡ 사용자가 수화기를 hook-off하면 이 상태를 감지한 이동국은 'connect order' 를 전달한다. 이를 수신한 기지국은 'connect order' 를 수신한 후, 이에 대한 응답을 전송한다.

⑨ 이후 트래픽 채널을 이용하여 두 사용자는 통화를 수행한다.

제 2 장 해쉬 함수와 메시지 인증 시스템

일반적으로 해쉬 함수는 모듈러 연산을 이용한 해쉬 기법, 대칭형 암호 알고리듬을 이용한 해쉬 기법, 그리고 MD5 등과 같은 전용 해

쉬 기법 등이 있다. 본 장에서는 해쉬 함수의 기본 특성 및 인증에 적용되는 방법 등을 살펴보고 기존의 해쉬 함수로 널리 이용되고 있는 MD-5 해쉬 함수와 ISO/IEC에서 표준 방식으로 고려 중인 여러 해쉬 기법들을 분석한다.[6-17]

2.1 해쉬 함수(hash function : H)의 정의와 요구조건[2]

일방향 해쉬 함수는 가변-크기의 메시지 M(variable-size message M)을 입력하여 고정된 크기의 메시지 해쉬 값 H(M)를 출력하는 함수이다. 이는 주로 디지틀 서명 기법과 결합하여 메시지 인증에 이용되며, 메시지 인증시에 송신되어야 할 데이타는 메시지 M과 메시지에 대한 해쉬 값 H(M)이다. 즉 해쉬 함수는 파일, 메시지, 또는 다양한 데이타 블럭의 지문(finger print)을 생성하기 위하여 이용된다. 해쉬 함수 H는 다음과 같은 특성을 갖도록 설계되어야 한다.

① 블럭의 크기가 가변인 입력 데이타에 적용 가능해야 한다.
② H의 출력은 고정-길이의 출력을 생성해야 한다.
③ x가 주어진 경우 H(x)의 계산은 용이해야 한다.
④ H(x) = m인 m이 주어진 경우, H(x) = m을 만족하는 x를 찾는 것이 계산적으로 불가능해야 한다. 이는 일방향성이라 한다.
⑤ 데이타 블럭 x가 주어진 경우, H(y) = H(x)인 y($\neq$x)를 찾는 것이 계산적으로 불가능해야 한다.
⑥ H(x) = H(y)인 임의의 쌍 (x, y)를 구하는 것이 계산적으로 불가능 해야 한다.

특성 ①은 메시지 인증 기법에 해쉬 함수를 실제 적용하기 위해 요구되는 요구사항이며, 특성 ④는 일방향 특성이다. 특성 ⑤는 x와 H(x)가 주어진 경우, H(y) = H(x)인 y를 구하는 것이 어렵다는 것과 등가이다. 만약 이 특성이 만족되지 않으면 ① 공격자가 통신로 중간에서 M, D_{ks}(H(M))을 관찰하고 가로챈 후, ② M으로 부터 H(M)을 계산하여, ③ H(M) = H(M′)인 자신에게 유리한 새로운 M′를 생성하여, ④ M', D_{ks}(H(M′))을 송신자에게 전송하는 공격이 가능해 진다. 특성 ⑥은 birthday attack에 대한 대비하기 위한 요구조건이다.

강한 해쉬 함수는 특성 ①에서 ⑥까지의 모든 특성을 만족하는 해쉬 함수를 지칭하고, 약한 해쉬 함수는 특성 ①에서 ⑤까지를 만족하는 해쉬 함수를 지칭한다.

2.2 메시지 인증 방식의 종류

메시지의 무결성을 검사하기 위한 메시지 다이제스트(message digest) 인증 기법은 관용 암호 시스템을 이용한 메시지 다이제스트 인증, 공개키 암호 시스템을 이용한 메시지 다이제스트 인증, 비밀 값과 해쉬 함수를 이용한 메시지 인증 기법 등으로 분류될 수 있다.

관용 암호시스템을 이용한 메시지 다이제스트 인증에서는 다음의 가정에 바탕을 두고 구성된다.

① 송신자와 수신자는 공통의 해쉬 함수를 공유한다.
② 송신자와 수신자는 공통의 비밀키를 공유한다.

송신자는 그림 2.1과 같이 메시지에 대한 해쉬 값을 구한 후, 해쉬 값만 비밀 키로 암호화하여 전송하는 방식이다. 이 방식의 특징은 H(M)의 계산이 용이한 반면 E_k(H(M))을 K가 알려져 있지 않다면 계산이 불가능하다. 수신단에서는 그림 2.2와 같은 과정을 통해 수신된 메시지를 검증한다.

수신단에서는 계산된 H(M)과 비밀키 K를 이용하여 복구된 H(M)이 동일한가를 검사하여 송신자가 비밀키 K의 소유를 확인함으로서 메시지가 정당한 상대로 부터 송신되었다는 것을 확인한다. 이 방식의 장점은 $E_k(H(M))$이 해쉬 값을 암호화한 값이므로 낮은 복잡도를 갖는 인증 시스템을 구현할 수 있다.

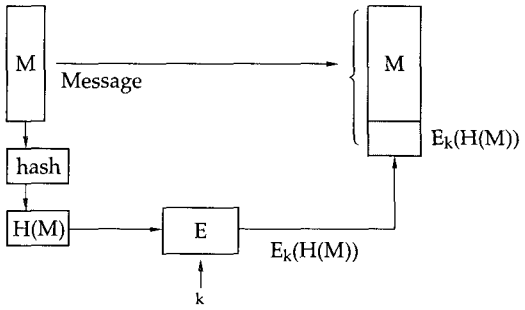


그림 2.1 관용키 암호시스템을 이용한 메시지 인증에서의 송신단 동작

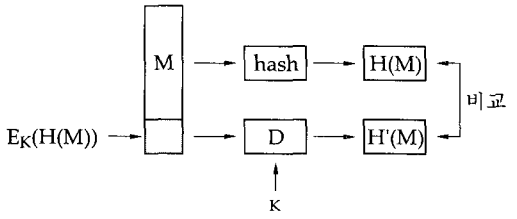


그림 2.2 관용키 암호시스템을 이용한 메시지 인증에서의 수신단 동작

공개키 암호시스템을 이용한 메시지 다이제스트 인증은 그림 2.3과 같이 메시지의 해쉬값을 공개키 암호시스템의 비밀키를 이용하여 서명문을 생성한 후 이를 전송한다.

송신단에서는 메시지의 해쉬 값을 구한 후 비밀키로 서명하여 전송하고, 수신단에서는 그림 2.4와 같이 수신된 메시지를 이용하여 계산된 해쉬 값과 수신된 서명문에서 송신자의 공개키를 이용하여 복구한 값이 동일한가를 확인함으로서 송신자 메시지의 정당성을 확인한다. 이 방식은 디지틀 서명 방식과 메시지 인증 기법이 결합되어 실현되며, 공개키 암호 시스템을 이용하므로 서명을 위한 키 분배가 요구되지 않는 특징이 있다.

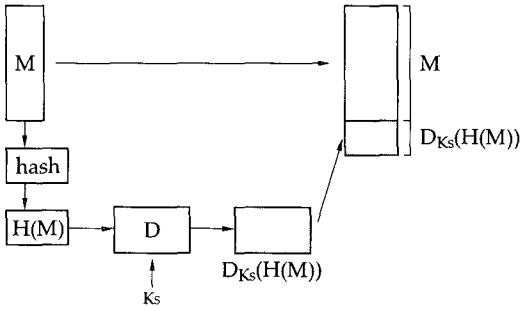


그림 2.3 공개키 암호시스템을 이용한 인증 기법에서의 송신단의 동작

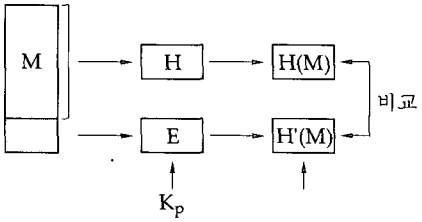


그림 2.4 공개키 암호시스템을 이용한 인증 기법에서의 수신단의 동작

암호 알고리듬을 이용하지 않은 메시지 인증 기법은 다음과 같은 이유로 이용될 전망이다.

① 암호화 소프트웨어의 동작은 매우 저속이다.
② 암호화 하드웨어 경비는 무시할 수 없다.
③ 암호화 하드웨어는 큰 크기의 데이타에 적합하다. 따라서 작은 블럭의 데이타인 경우 초기화 및 호출 시간이 매우 크다.
④ 암호화 알고리듬은 특허로 보호되어 있다. 예를 들어 RSA의 경우 특허로 보호되어 라인센스를 받아 사용할 수 있다.

⑤ 암호 알고리듬은 수출입 통제하에 있을 수 있다. 대표적인 예로 DES의 경우 수출 통제하에 있다.

암호 알고리듬을 이용하지 않은 메시지 인증 기법인 비밀 정보와 해쉬 함수를 이용한 메시지 인증은 두 통신 상대가 비밀 값 S_{AB}를 공유한다는 가정 하에 바탕을 두고 있다.

송신단은 그림 2.5와 같이 비밀 정보와 메시지를 쇄상(concatenate)하여 생성된 새로운 메시지에 대한 해쉬한 값을 계산하고 비밀 정보를 제외한 원래의 메시지와 해쉬 값을 수신단에 전송한다. 즉 송신 메시지는 $M||H(S_{AB}||M)$이 된다.

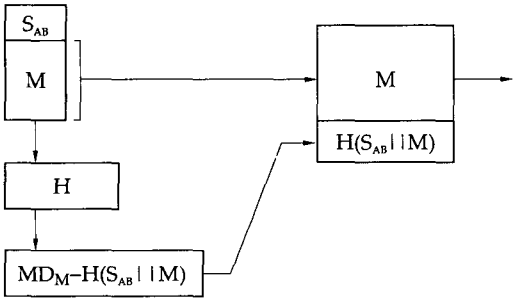


그림 2.5 비밀 정보를 이용한 메시지 인증 기법의 송신단에서의 동작

수신단에는 그림 2.6과 같이 자신이 가지고 있는 비밀 정보와 메시지를 쇄상하여 생성된 메시지에 대한 해쉬 값을 계산한 후, 수신된 해쉬 값과 비교하여 수신된 메시지의 정당성을 검증한다.

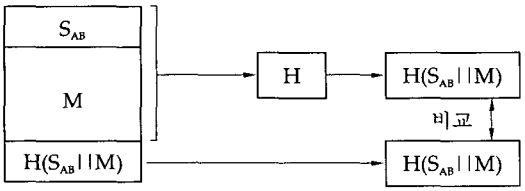


그림 2.6 비밀 정보를 이용한 메시지 인증 기법의 수신단에서의 동작

본 방식은 송신단과 수신단에서 해쉬 값 계산시에 암호 알고리듬이 수행되지 않으므로 인증 시스템의 고속 동작이 가능하다. 그리고 S_{AB}가 비밀이므로 불법의 사용자는 $H(S_{AB}||M)$를 생성할 수 없다는 것에 안전성의 근간이 있다. 이 방식은 CDMA 이동 통신의 인증 시스템으로 적용될 수 있다.

2.3 MD5 해쉬 알고리듬

본 절에서는 전용 해쉬 함수의 대표적인 방식인 MD5 해쉬 함수의 동작 절차를 분석한다. MD5 해쉬 함수는 128-비트 해쉬 값(메시지 다이제스트)를 생성하고, 고속 동작이 가능하며, 복잡도가 낮고, 32-비트 단위로 데이타 블럭을 처리하며, MD5 입력의 블럭 단위 512 비트 블럭이다. 이는 다음과 같은 다섯 단계로 수행된다.

(단계 1) 패딩 비트 부가

메시지의 길이가 448(mod 512) 비트가 되도록 패딩 비트 삽입하는 단계이다. 즉 패딩된 메시지의 길이가 512비트의 정수배 보다 64비트 작게 만드는 과정이다. 메시지의 길이가 원하는 길이인 경우도 (즉, 448 mod 512) 패딩 비트는 항상 부가된다. 패딩 비트의 수 1에서 512비트이며 패딩 패턴은 '1' + "0 … 0" 이다. 예를 들어, 메시지가 448비트인 경우 448비트의 메시지에 512비트의 패딩 비트를 부가하여 960비트의 데이타 블럭으로 재구성된다.

(단계 2) 정보 비트의 길이 부가

패딩 이전의 원래 메시지의 길이를 64비트로 표현하여 패딩 메시지에 부가하는 과정이다. 원래의 메시지의 길이가 2^{64}보다 크면 길이의 하위 64비트만을 사용한다.

따라서 단계 1과 단계 2의 결과로 메시지의 길이는 512비트 단위가 됨을 알 수 있다. 한편 512 비트는 16개의 32-비트 워드로 재구성될 수 있다. 이는 M[0], M[1], …, M[N-1]로 표현될 수 있다.

(단계 3) MD 버퍼의 초기화

알고리듬이 이용하는 4개의 32-비트 레지스터(A, B, C, D)는 다음과 같은 값으로 초기화된다.

A	=	01	23	45	67
B	=	89	AB	CD	EF
C	=	FE	DC	BA	98
D	=	76	54	32	10
		MSB			LSB

(단계 4)

128-비트(16워드) 블럭에서의 메시지 처리

MD5가 사용하는 4개의 비선형 함수의 입력은 32비트 워드이고, 출력 역시 32비트워드이다. 각 함수는 비트 단위 논리로 동작되며, 출력은 n번째 비트는 세입력의 각 n번째 입력 비트의 함수로 표현될 수 있다. 비선형 함수는 식(2.1)과 같다.

$$F(X,Y,Z) = (X \cdot Y) + (X' \cdot Z)$$
$$G(X,Y,Z) = (X \cdot Z) + (Y' \cdot Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X + Z') \qquad (2.1)$$

여기서, ·은 AND, +는 OR, ′는 NOT, ⊕는 EXOR 연산을 의미한다.

4비선형 논리함수의 진리표는 표 2.1과 같다.

표 2.1 네가지 비선형 함수의 진리표

X	Y	Z	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

MD5 알고리듬이 이용하는 64개의 난수 T[1], ···, T[64]는 식(2.2)에 의하여 계산될 수 있고, 이의 값은 표 2.2와 같다. 이는 입력 데이타의 규칙성을 배제하기 위한 32비트 패턴의 랜덤화된 패턴이다.

$$T[i] = \text{integer part of } 2^{32} * \text{abs}(\sin(i)) \qquad (2.2)$$

여기서, i는 라디안이고, $0 \le \text{abs}(\sin(i)) \le 1$, $0 \le 2^{32} * \text{abs}(\sin(i)) \le 2^{32}$이다.

표 2.2 난수 T[i]

T[1] = D76AA478	T[17] = F61E2562	T[33] = FFFA3942	T[49] = F4292244
T[2] = E8C7B756	T[18] = C040B340	T[34] = 8771F681	T[50] = 432AFF97
T[3] = 242070DB	T[19] = 265E5A51	T[35] = 69D96122	T[51] = AB9423A7
T[4] = C1BDCEEE	T[20] = E9B6C7AA	T[36] = FDE5380C	T[52] = FC93A039
T[5] = F57C0FAF	T[21] = D62F105D	T[37] = A4BEEA44	T[53] = 655B59C3
T[6] = 4787C62A	T[22] = 02441453	T[38] = 4BDECFA9	T[54] = 8F0CCC92
T[7] = A8304613	T[23] = D8A1E681	T[39] = F6BB4B60	T[55] = FFEFF47D
T[8] = FD469501	T[24] = E7D3FBC8	T[40] = BEBFBC70	T[56] = 85845DD1
T[9] = 698098D8	T[25] = 21E1CDE6	T[41] = 289B7EC6	T[57] = 6FA87E4F
T[10] = 8B44F7AF	T[26] = C33707D6	T[42] = EAA127FA	T[58] = FE2CE6E0
T[11] = FFFF5BB1	T[27] = F4D50D87	T[43] = D4EF3085	T[59] = A3014314
T[12] = 895CD7BE	T[28] = 455A14ED	T[44] = 04881D05	T[60] = 4E0811A1
T[13] = 6B901122	T[29] = A9E3E905	T[45] = D9D4D039	T[61] = F7537E82
T[14] = FD987193	T[30] = FCEFA3F8	T[46] = E6DB99E5	T[62] = BD3AF235
T[15] = A679438E	T[31] = 676F02D9	T[47] = 1FA27CF8	T[63] = 2AD7D2BB
T[16] = 49B40821	T[32] = 8D2A4C8A	T[48] = C4AC5665	T[64] = EB86D391

X ≪ S는 레지스터 X를 왼쪽으로 S비트 만큼 순회 치환(circular shift)한 32-비트 값이다. 즉 레지스터 X의 왼쪽 MSB S 비트는 순회 치환 후에 레지스터 X의 오른쪽 LSB S 비트가 되며, 나머지 부분은 오른쪽으로 S회 치환된다. 초기치 레지스터 A, B, C, D는 다음과 같은 절차로 갱신된다. 여기서 +는 모듈러 2^{32} 덧셈을 의미한다.

```
/* Process each 16-word (512-비트) block */
 For i = 0 to N/16-1 do      /* Copy block i into X. */
  For j = 0 to 15 do
      Set X[j] to M[i * 16 + j]
  end /* of loop on j */

/* Save A as AA, B as BB, C as CC, and D as DD */
  AA = A
  BB = B
  CC = C
  DD = D

/* Let[abcd k s i] denote the operation a = b + ((a + F(b, c, d) + X[k] + T[i])<<<s). */
 [ABCD  0   7   1] [DABC   1  12   2] [CDAB   2  17   3] [BCDA   3  22   4]
 [ABCD  4   7   5] [DABC   5  12   6] [CDAB   6  17   7] [BCDA   7  22   8]
 [ABCD  8   7   9] [DABC   9  12  10] [CDAB  10  17  11] [BCDA  11  22  12]
 [ABCD 12   7  13] [DABC  13  12  14] [CDAB  14  17  15] [BCDA  15  22  16]

/* Let[abcd k s i] denote the operation a = b + ((a + G(b, c, d) + X[k] + T[i])<<<s). */
 [ABCD  1   5  17] [DABC   6   9  18] [CDAB  11  14  19] [BCDA   0  20  20]
 [ABCD  5   5  21] [DABC  10   9  22] [CDAB  15  14  23] [BCDA   4  20  24]
 [ABCD  9   5  25] [DABC  14   9  26] [CDAB   3  14  27] [BCDA   8  20  28]
 [ABCD 13   5  29] [DABC   2   9  30] [CDAB   7  14  31] [BCDA  12  20  32]

/* Let[abcd k s i] denote the operation a = b + ((a + H(b, c, d) + X[k] + T[i])<<<s). */
 [ABCD  5   4  33] [DABC   8  11  34] [CDAB  11  16  35] [BCDA  14  23  36]
 [ABCD  1   4  37] [DABC   4  11  38] [CDAB   7  16  39] [BCDA  10  23  40]
 [ABCD 13   4  41] [DABC   0  11  42] [CDAB   3  16  43] [BCDA   6  23  44]
 [ABCD  9   4  45] [DABC  12  11  46] [CDAB  15  16  47] [BCDA   2  23  48]

/* Let[abcd k s i] denote the operation a = b + ((a + I(b, c, d) + X[k] + T[i])<<<s). */
 [ABCD  0   6  49] [DABC   7  10  50] [CDAB  14  15  51] [BCDA   5  21  52]
 [ABCD 12   6  53] [DABC   3  10  54] [CDAB  10  15  55] [BCDA   1  21  56]
 [ABCD  8   6  57] [DABC  15  10  58] [CDAB   6  15  59] [BCDA  13  21  60]
 [ABCD  4   6  61] [DABC  11  10  62] [CDAB   2  15  63] [BCDA   9  21  64]
```

```
/ * Then increment each of the four registers by the value it had before this block was
                                                                started * /
    A = A + AA
    B = B + BB
    C = C + CC
    D = D + DD

    end / * of loop on i * /
```

(단계 5) 해쉬 값 출력

단계 4의 반복 동작을 통해 생성되는 해쉬 값은 네개의 A, B, C, D 레지스터에 저장된 값이다.

MD5 해쉬 함수의 특징은 해쉬 값의 비트는 입력의 모든 비트의 함수이고, 기본 함수들(F, G, H, I)을 이용한 여러 단계의 복잡한 과정은 출력 해쉬 값이 서로 다르도록 한다. 즉 임의로 선택된 2개의 메시지에 대한 해쉬 값들은 서로 다르다. 이는 동일한 MD5 해쉬 값을 생성하는 두 메시지를 생성하는 것은 계산적으로 매우 어려움을 의미한다. 즉 2^{64}개의 반복 동작이 요구된다. 주어진 해쉬 값을 생성하는 또 다른 메시지를 생성하는 것도 매우 어렵다. 즉, 2^{128}의 동작이 요구된다.

2.4 DM 해쉬 함수

DM 해쉬 함수는 대칭형 암호 알고리듬을 이용한다. 이는 1985년 Davies와 Meyer에 의해 제안되어, 1989년 Quisquater와 Girault에 의해 개선된 방식으로 DP-10118에 draft proposal 형태로 표준화되었다.[7,14]

이는 그림 2.7과 같은 과정으로 실현된다.

① 메시지는 블럭 암호 시스템의 키 길이 m과 동일한 서브 메시지 길이를 갖는 t개의 m비트 블럭들로 재구성된다.

$$M_1, M_2, \cdots, M_t \quad (2.3)$$

② 그리고 식 (2.4)의 과정을 t회 수행한다.

$$H_i = E_{Mi}(H_{i-1}) \oplus H_{i-1}, \text{ for } i = 1, 2, \cdots, t \quad (2.4)$$

여기서, H_0 = I는 초기치로 임의로 선택된다. 출력 해쉬 값은 단계 t에서의 n비트 블럭인 H_t로서 이는 메시지 다이제스트로 이용될 수 있다.

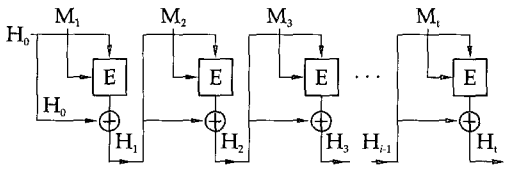


그림 2.7 DM 해쉬 함수

2.5 모듈러 연산에 바탕을 둔 해쉬 함수[3]

이 방식은 기본적으로 모듈러 지수 연산을 이용하며, 서명 역시 모듈러 지수 연산 상에서 동작된다. 본 절에서는 Jueneman의 기법을 분석한다. 이는 Quadratic congruence manipulation detection code(QCMDC)라 불리며, 1982년 Jueneman가 제안하였다. 기본적으로 알고리듬이 이용하는 변수들은 다음과 같다.

① m : 분할되는 메시지의 길이

② c : $c \geq 2^m$-1인 소수

③ H_0 : 초기치 (비밀키)

이 방식의 해쉬 값 생성 과정은 그림 2.8과 같다.

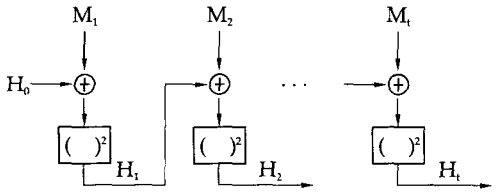


그림 2.8 QCMDC 해쉬 함수

① 메시지는 t개의 m-비트 블럭으로 식(2.5)와 같이 재구성된다. 여기서, $c \geq 2^m-1$ 이다.

$$M_1, M_2, \cdots, M_t \qquad (2.5)$$

② 위의 분할된 메시지를 이용하여 식 (2.6)과 같은 반복 동작을 수행한다.

$$H_i = (H_{i-1}+M_i)^2 \text{ module } c,$$
$$\text{for } i = 1, 2, \cdots, t \qquad (2.6)$$

출력 해쉬 값은 단계 t에서의 H_t로서 이는 메시지 다이제스트로 이용될 수 있다. 이 방식의 주요 특징은 키를 가지고 있는 해쉬 함수이며, 출력 해쉬 값의 길이가 모듈러 연산에서 이용되는 정수의 비트 길이와 동일하다는 것이다.

2.6 ISO/IEC 10118-2의 해쉬 함수[14]

해쉬 함수는 메시지 변조 및 삽입 검사하는 기능을 수행하며, 충돌 회피성을 가짐으로서 인증 기능에도 적용 가능하다. 본 방식은 그림 2.9와 같이 대칭형 암호 알고리듬을 이용하며 기본적으로 암호문의 블럭 길이와 동일한 길이를 갖는 해쉬 함수 값을 출력한다.

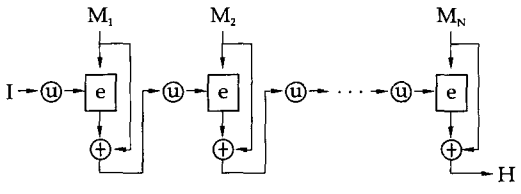


그림 2.9 ISO/IEC 10118-2의 해쉬 함수

암호문의 블럭 길이보다 두배 긴 해쉬 값을 출력하는 해쉬 함수는 그림 2.10과 같이 구성된다. 기본적으로 이 방식 역시 블럭 암호 알고리듬을 이용하여 실현된다. 이 방식은 암호문의 블럭 길이와 동일한 길이를 출력하는 해쉬 함수에 비해 birthday attack에 강하다.

여기서 ||는 쇄상을 의미한다.

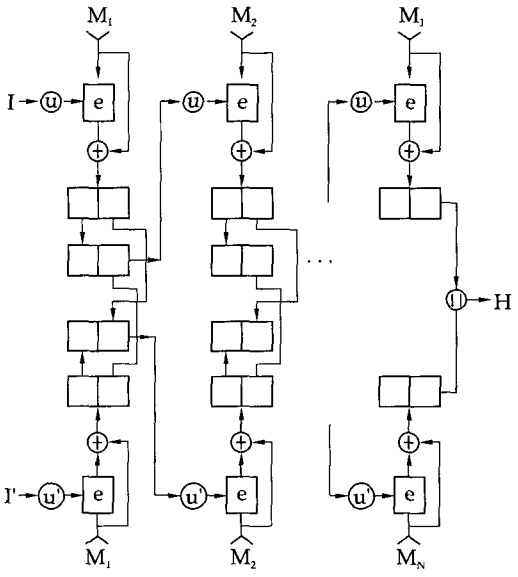


그림 2.10 암호문의 블럭 길이보다 2배 긴 해쉬 값을 출력하는 해쉬 함수

이 밖에 널리 이용되는 해쉬 함수는 미국의 표준 디지틀 서명에 이용되는 해쉬 함수인 SHA(secure hash function)들을 들 수 있다.

제 3 장 CDMA 이동 통신에 적용 가능한 인증 시스템

본 장에서는 속도 및 성능 측면에서 모듈러 연산에 기초한 해쉬 기법보다 우수하여 이동 통신망의 인증 시스템에 적용 가능한 전용 해

쉬 함수를 이용한 인증 기법과 대칭형 암호 알고리듬을 이용한 해쉬 함수에 바탕을 둔 인증 시스템을 제시한다. 이를 위하여 CDMA 이동 통신을 위한 인증 시스템으로 MD5을 변경하여 구현된 두가지 인증 시스템과 ISO/IEC 10118-2의 해쉬 함수를 이용한 인증 시스템을 제안한다. 제안된 인증 시스템은 고속 동작이 가능한 MD5 해쉬 함수와 대칭형 암호 알고리듬을 이용하여 설계되어 있으므로 기본적으로 속도가 빠르고 계산을 위한 복잡도가 낮다.

3.1 인증 시스템 설계를 위한 일반 사항

이동 통신망의 인증 시스템은 고속 동작이 가능하고 저전력 동작이 가능해야 하는 요구사항을 고려하면 복잡도가 낮은 인증 시스템을 이용해야 한다. 그리고 인증 시스템은 이동국의 비밀 정보 소지 여부로 이동국을 인증하는 메시지 인증 시스템을 이용할 수 있다. 메시지 인증 시스템은 여러 방식이 있으나 그중 해쉬 함수를 이용하여 실현된 메시지 인증 시스템이 속도 및 복잡도 측면에서 유리할 것이다.

CDMA 이동 통신의 인증 시스템에 이용되는 해쉬 함수는 birthday attack을 피할 수 있어야 하고, 새로운 공유 비밀 정보인 SSD_A와 SSD_B 생성과정에서 128비트의 해쉬 출력 값을 요구함을 고려하여 기본적으로 해쉬 출력 값의 비트 길이가 128비트인 해쉬 함수를 이용한다. CDMA 인증 단계에서 요구되는 인증자(authenticator)의 비트 길이가 128 이하인 경우, 해쉬 함수의 출력 128 비트중 왼쪽 비트들만을 이용한다.

해쉬 함수는 CDMA 이동 통신망에서의 발호, 착호, 유일 응답 과정, 그리고 기지국 도전 응답 과정에서 이용되는 18비트의 인증 데이타 생성 및 128비트의 새로운 비밀 공유 데이타인 SSD_A, SSD_B를 생성하는데 이용된다. 기지국 도전 응답과정에서는 128비트 새로운 SSD_A, SSD_B의 생성이 요구되므로 전체 해쉬 값을 이용하며, 나머지 과정에서는 왼쪽 18비트만을 이용한다. 나머지 과정에서 왼쪽 18비트만을 이용하여 실현하는 해쉬 함수를 이용해도 이동 통신망의 빠른 응답 시간을 고려하면 안전성에 문제가 없다.

CDMA 이동 통신의 인증을 위한 입력 블럭은 표 3.1과 같이 152비트의 데이타 블럭으로 구성되어 있다. 여기서 아래첨자 s와 p는 반영구와 영구를 의미한다. 공유 비밀 데이타는 SSD_A와 SSD_B로서, 이동국의 반영구적인 메모리에 저장된다. SSD_A는 이동 통신을 위한 인증용으로 이용되며, SSD_B는 음성 보호와 메시지의 비밀성을 위해 사용된다. $RAND_S$는 페이징 채널상에서 최근에 수신된 액세스

표 3.1 CDMA 이동 통신의 인증을 위한 입력 파라메타들

파라메타 / 과정	RAND-CHALLENGE (32-bits)	ESN (32bits)	AUTH_DATA (24bits)	SSD_AUTH (64bits)
등 록	$RAND_S$	ESN_P	MIN1	SSD_A
유일-도전응답	256*RANDU + (8 LSB_S of MIN2)	ESN_P	MIN1	SSD_A
발 호	$RAND_S$	ESN_P	Digits	SSD_A
착 호	$RAND_S$	ESN_P	MIN1	SSD_A
base station challenge	RANDBS	ESN_P	MIN1	SSD_A_New

SSA 생성	RANDSSD(56)	ESN(32)	A_key(64)	SSD(128)

파라메타 메시지의 RAND 영역의 값이다. 이는 이동국의 발호, 착호, 등록 확인 절차에 사용된다.

표 3.1에서 알수 있듯이 입력 파라메타는 RAND-CHALLENGE용 32비트, ESN용 32비트, AUTH-DATA용 24비트, SSD__AUTH 용 64비트로 전체 입력 비트의 길이는 152비트가 된다.

3.2 MD5 해쉬 함수를 이용한 인증 시스템

MD5을 이용한 인증 시스템은 두가지 방식이 존재한다. 첫 방법은 152비트의 입력 블럭에 '10000000' 패턴을 부가하여 5개의 32비트 블럭인 160비트의 입력 블럭을 생성한 후, 입력 블럭에 대해 4-라운드 동작을 수행하여 해쉬 값을 출력하는 방법이다. 두 번째 방법은 152비트의 입력 메시지 블럭에 360비트로 구성된 '10 … 0'를 쇄상하여 512비트의 입력 블럭으로 변경하여 원래의 16라운드 동작을 수행하는 방식이다. 전자는 기본적으로 참고문헌 [18]의 방식과 유사하나, MD5 기본 함수를 변경 없이 사용했다는 측면에서 다른다. 본 절에서는 속도 측면에서 유리할 것으로 예측되는 전자의 방식을 제시한다.

MD5 해쉬 함수를 변형하여 얻은 해쉬 함수는 다음과 같은 과정으로 구성된다.

(단계 1) 패딩 비트 부가

MD5을 이용하여 변형된 해쉬 방법을 위한 152-비트의 입력 블럭은 8-비트 패턴인 '*10000000*'가 부가되어 5개의 32비트 블럭인 160비트의 입력 블럭을 생성한다.

(단계 2) 해쉬 함수가 이용하는 네개의 버퍼 A, B, C, D의 초기화 네개의 버퍼 A, B, C, D는 다음과 같은 값으로 초기화된다.

A : 01 23 45 67
B : 89 AB CD EF
C : FE DC BA 98
D : 76 54 32 10

(단계 3) 단계 1에서 생성된 5개의 32-비트 워드로 구성된 160비트 메시지 블럭은 식(3.1)과 같다.

$$X[1], X[2], X[3], X[4], X[5] \quad (3.1)$$

해쉬 함수가 사용하는 4개의 비선형 함수는 입력이 32-비트 워드이며, 출력 역시 32-비트 워드이다. 각 함수는 비트 단위의 논리로 동작되며, 출력의 n번째 비트는 비선형 함수의 세 입력 블럭의 n번째 비트들의 함수이다. 네개의 비선형 함수는 식 3.2와 같이 MD5에서 이용되었던 함수와 동일하다.

$$\begin{aligned} F(X, Y, Z) &= (X \cdot Y) + (X' \cdot Z) \\ G(X, Y, Z) &= (X \cdot Y) + (Y' \cdot Z') \\ H(X, Y, Z) &= X \oplus Y \oplus Z \\ I(X, Y, Z) &= Y \oplus (X \oplus Z') \end{aligned} \quad (3.2)$$

여기서, ·은 AND, +는 OR, ′는 NOT, ⊕는 EXOR 연산을 의미한다.

4개의 기본 논리 함수의 진리표는 표 2.1과 같다. 알고리듬이 이용하는 20개의 난수 값 T[1], …, T[20]는 식 (3.3)에 의하여 계산될 수 있고, 이의 결과는 표 3.2와 같다. 이는 입력 데이타의 규칙성을 배제하기 위한 32-비트 패턴의 랜덤 패턴이다.

$$T[i] = \text{integer part of } 2^{32} * \text{abs}(\sin(i)) \quad (3.3)$$

여기서, i는 라디안 값이고, $0 \le \text{abs}(\sin(i)) \le 1$, 그리고 $0 \le 2^{32} * \text{abs}(\sin(i)) \le 2^{32}$이다.

표 3.2 T[i] 값

T[1] = D76AA478	T[11] = FFFF5BB1
T[2] = E8C7B756	T[12] = 895CD7BE
T[3] = 242070DB	T[13] = 6B901122
T[4] = C1BDCEEE	T[14] = FD987193
T[5] = F57C0FAF	T[15] = A679438E
T[6] = 4787C62A	T[16] = 49B40821
T[7] = A8304613	T[17] = F61E2562
T[8] = FD469501	T[18] = C040B340
T[9] = 698098D8	T[19] = 265E5A51
T[10] = 8B44F7AF	T[20] = E9B6C7AA

X<<S는 레지스터 X를 왼쪽으로 S비트 만큼 순회 치환(circular shift) 한 32-비트 값이다. 그리고 다음과 같은 4라운드 과정을 이용하여 초기치 레지스터 A, B, C, D들을 변경한다.

/ * Process each 16-word (512-비트) block * /

/ * Let [abcd k s i] denote the operation a = b + ((a + F(b,c,d) + X[k] + T[i])<<<s). * /
[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4] [ABCD 4 7 5]

/ * Let [abcd k s i] denote the operation a = b + ((a + G(b,c,d) + X[k] + T[i])<<<s). * /
[ABCD 1 5 6] [DABC 3 9 2] [CDAB 2 14 8] [BCDA 0 20 9] [ABCD 4 5 10]

/ * Let [abcd k s i] denote the operation a = b + ((a + H(b,c,d) + X[k] + T[i])<<<s). * /
[ABCD 2 14 11] [DABC 0 11 12] [CDAB 3 16 13] [BCDA 4 23 14] [ABCD 1 4 15]

/ * Let [abcd k s i] denote the operation a = b + ((a + I(b,c,d) + X[k] + T[i])<<<s). * /
[ABCD 3 6 16] [DABC 1 10 17] [CDAB 0 15 18] [BCDA 4 21 19] [ABCD 2 6 20]

(단계 4) 해쉬 값 출력

반복 동작을 통해 생성되는 메시지 다이제스트는 네개의 A, B, C, D 레지스터에 저장된 값이다.

현재까지 MD5 해쉬 함수를 공격할 수 있는 효율적인 알고리듬은 알려져 있지 않다. 제시된 해쉬 함수를 이용한 인증 시스템의 안전성은 MD5 해쉬 함수의 안전성에 바탕을 두고 있다.

여기서 제시된 해쉬 함수의 특징은 다음과 같다.

① 출력 해쉬 값의 각 비트는 입력 블럭의 모든 각각의 비트의 함수이다.

② 기본 함수들 (F, G, H, I)을 이용한 여러 단계의 복잡한 과정은 개개의 입력 블럭에 대한 출력 해쉬 값을 서로 다르게 한다. 즉 임의로 선택된 2개의 메시지에 대한 해쉬 값들은 서로 다르다.

③ 동일한 해쉬 값을 생성하는 두 메시지를 생성하는 것은 계산적으로 매우 어렵다. 이는 MD5 해쉬 함수의 안전성에 기초한다. 즉, 2^{64}개의 반복 동작이 요구된다.

④ 주어진 해쉬 값을 생성하는 또 다른 메시지를 생성하는 것이 매우 어렵다. 즉, 2^{128}의 동작이 요구된다.

3.4 블럭 암호 알고리듬을 이용한 인증 시스템

블럭 암호 알고리듬을 이용한 인증 시스템은 152비트의 데이타 블럭에 40비트의 패딩 패턴을 부가하여 192비트의 데이타 블럭을 생성한 후, 이 블럭을 해수 함수의 입력 블럭으로 이용한다. 따라서 입력 데이타 블럭은 3개의 64-비트 블럭인 192비트의 M_1, M_2, M_3 블럭으로 재구성될 수 있다. 그리고 이동통신의 기지국 도전 과정에서 이용되는 SSD_A와 SSD_B 정보를 생성해야 한다는 점을 고려하여 128비트의 출력 해쉬 값을 갖는 ISO/IEC 10118-2 해쉬 함수를 이용한다. 18비트의 출력은 128비트중 왼쪽 18비트이다. 사용 가능한 블럭 암호 알고리즘은 DES 또는 FEAL이다. 블럭 암호 알고리듬을 이용한 해쉬 함수의 초기치는 식(3.4)와 같은 값을 이용한다.

I(초기치) : 5252525252525252H
I′ : 2525252525252525H (3.4)

ISO/IEC의 블럭 암호 알고리듬을 이용한 해쉬 함수를 이용한 인증 시스템은 그림 3.1과 같다. 해쉬 함수는 6개의 DES 암호화 과정과 몇 개의 블럭 치환 과정이 요구되며 고속 동작이 가능한 해쉬 함수이다. 여기서 u, u′는 입력 블럭의 특정 비트를 특정 패턴으로 셋하는 함수들이다. 키의 64비트가 "X_1, X_2 …, X_{64}" 일때 "X_2, X_3"을 "10" 또는 "01"로 셋하는 함수이다. 즉 비트 2,3을 '10' 또는 '01'로 셋하는 함수이다. 여기서 비트 2,3은 블럭의 왼쪽 2번째와 3번째 비트로 가정한다.

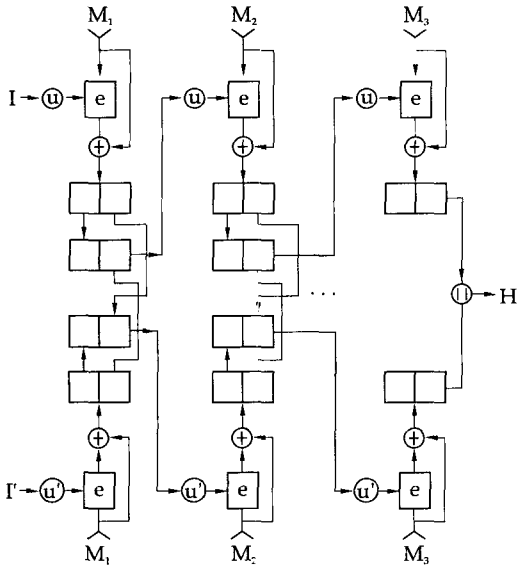


그림 3.1 비트길이가 2n인 해쉬 함수를 이용하여 생성된 인증 시스템

제 4 장 제안된 인증 시스템의 시뮬레이션

4.1 등록 인증 과정에 대한 시뮬레이션

본 절에서는 그림 4.1과 같이 $RAND_S$가 "01234567", ESN이 "89ABCDEF", MIN이 "565F15", SSD_A가 "FEDCBA9876543210"인 경우의 등록 인증 과정에 대한 인증자를 C언어를 이용한 프로그램으로 구했다. 152비트의 입력 블럭에 '10000000' 패턴을 부가하여 5개의 32비트 블럭인 160비트의 입력 블럭을 생성한 후, 입력 블럭에 대해 4-라운드 동작을 수행하여 해쉬 값을 출력하는 방법이다. 출력 18비트는 128비트의 해쉬 값 중 왼쪽 18비트만을 이용하였다.

$RAND_S$(32)	ESN(32)	MIN(24)	SSD_A (64)	PADDING	→	AUTHR(18)
01234567	89ABCDEF	565F15	FEDCBA9876543210	80	→	17C11

블럭 암호 알고리듬을 이용한 인증 시스템의 입력도 변형된 MD5을 이용하는 경우와 같은 입력 블럭을 이용한다. 그러나 여기서는 40비트의 패딩 패턴(8000000000)이 부가되어 192비트의 데이타 블럭을 생성한다. 따라서 입력 데이타 블럭은 3개의 64-비트 블럭인 192비트의 M_1, M_2, M_3 블럭으로 재구성된다. 시뮬레이션에 이용된 블럭 암호 알고리즘은 DES이다. 해쉬 함수의 초기치는 I = 5252525252525252H, I' = 2525252525252525H값을 이용하였다.

$RAND_S$(32)	ESN(32)	MIN(24)	SSD_A (64)	PADDING	→	AUTHR(18)
01234567	89ABCDEF	565F15	FEDCBA9876543210	8000000000	→	1CDA9

4.2 SSD_A, SSD_B 갱신 과정에 대한 시뮬레이션

본 절에서는 RANDSSD가 "0123456789ABCD", ESN이 "89ABCDEF", A_key가 "0123456789ABCDEF"인 경우의 128비트의 SSD_A, SSD_B 갱신 과정 과정을 C언어로 시뮬레이션했다. 첫번째 시뮬레이션은 4.1절에서와 마찬가지로 152비트의 입력 블럭에 '10000000' 패턴을 부가하여 5개의 32비트 블럭인 160비트의 입력 블럭을 생성한 후, 입력 블럭에 대해 4-라운드 동작을 수행하여 해쉬 값을 출력하는 방법이다. 계산된 SSD_A, SSD_B는 다음과 같다.

RANDSSD(56)	ESN(32)	A_key(64)	PADDING	→	SSD_A(64)	SSD_A(64)
0123456789ABCD	89ABCDEF	012345678ABCDEF	80	→	9FE55524 BA27308E	CFAD38CF 09EBBEBA

블럭 암호 알고리듬을 이용한 SSD 갱신 과정도 입력도 변형된 MD5을 이용하는 경우와 같은 입력 블럭을 이용하나 40비트의 패딩 패턴(8000000000)을 부가되어 192비트의 데이타 블럭을 생성한다. 따라서 입력 데이타 블럭은 3개의 64-비트 블럭인 192비트의 M_1, M_2, M_3 블럭으로 재구성된다. 사용된 블럭 암호 알고리즘은 DES이며, 3.4절에서 제시된 128비트의 해쉬 함수를 이용한 인증 시스템을 사용한다. 해쉬 함수의 초기치는 I = 5252525252525252, I′ = 2525252525252525 값을 이용하였다.

RANDSSD(56)	ESN(32)	A_key(64)	PADDING	→	SSD_A(64)	SSD_A(64)
0123456789ABCD	89ABCDEF	0123456789ABCDEF	80000000	→	906C2A9A26F7031F	0BAAB2F5857C3C0A

제 5 장 결 론

본 고에서는 고속 동작과 소프트웨어 실현이 가능한 CDMA 이동 통신을 위한 인증 시스템을 설계했다. 이를 위하여 IS-95 CDMA 이동 통신 시스템에 바탕을 둔 인증과 관련된 파라메타들을 정의하고, 이동국이 기지국에 등록(registration)하고, 호를 개시하며, 호를 수신하는 경우의 인증 프로토콜 및 관련 파라메타를 분석하였으며, 상기의 과정이 실패한 경우의 유일-도전 응답 과정, 그리고 상기의 유일-

도전 응답 과정의 실패로 인한 이동국과 기지국이 가지고 있는 비밀 공유 정보(SSD)를 갱신하기 위한 SSD 갱신 과정 등을 분석하였다. 또한 호 개시 및 수신을 위한 호 처리 절차를 분석하였다.

이를 바탕으로 속도 및 성능 측면에서 우수한 CDMA 이동 통신망의 인증 방식으로 적용 가능한 해쉬 함수에 바탕을 둔 인증 시스템과 대칭형 암호 알고리듬을 이용한 해쉬 함수에 바탕을 둔 인증 시스템을 제안했다. 제안된 방식들의 동작 원리와 안전성의 근간, 그리고 동작 과정을 제시했으며, 해쉬 함수의 입출력 불럭이 어떻게 이동 통신망에 적용될 수 있는가를 제시하였다. 그리고 제안된 방식을 컴퓨터 프로그램을 이용하여 시뮬레이션 하였다. 시뮬레이션 결과 인증 시스템이 고속으로 동작될 수 있음을 확인하였다. 본 연구의 결과는 CDMA 이동 통신의 인증 시스템으로 활용 가능하다.

참 고 문 헌

[1] Rhee, M.Y., Cryptography and Secure Communications, Mcgraw-Hill, 1993.

[2] Stallings, Data and Computer Communications, Macmillan, 4th ed., 1994.

[3] G.J.Simmons, editor : Contemporary Cryptology, IEEE press, 1991.

[4] Seberry, J and J. Pieprzyk, Cryptography : An Introduction to Computer Security, Prentice Hall, Sydney, 1989.

[5] TIA/EIA/IS-95, Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System, July 1993.

[6] Schneier,B., Applied Cryptography : Protocols, Algorithms, and Source Code in C, Wiley, 1994.

[7] NBS, "Data Encryption Standard", FIPS Pub-46, NBS, 1972.

[8] Miyaguchi,S., Shiraishi, S and Shimiqu, S, "Fast Data Encipherment Algorithm FEAL-8", Review of the Electrical Communication Laboratories,36,4,pp. 433-437, 1988.

[9] Matyas, S.M. and C.H.Meyer, "Electronics signature for Data Encryption Standard" IBM Tech. Disc. Bull., Vol.24, No.5, pp.2332-2334, 1981.

[10] Merkle,R.C, "One Way Hash Functions and DES", Proc. of Crypto '89, pp. 407-419,Aug, 1989.

[11] Webster,A.F. and S.E.Tavares, "On the Design of S-boxes", Proc. of Crypto'85, pp. 523-534, 1986.

[12] Diffie, W. and M.E.Hellman, "Exhaustive Cryptanalysis of NBS Data Encryption Standard", IEEE, Vol.10, No.6, pp.74-84, 1972.

[13] Davies, D.W. and Price, W.L., "Digital Signatures-an Update", 7th Int. Conf. on Computer Communication, pp. 845-849, 1984.

[14] Quisquate,J.J. and Girault,M, Manuscript of "2n-bit Hash-functions using n-bit Symmetric Block Cipher Algorithms", Eurocrypt'89, Oct. 1989.

[15] Quisquater,J.J. and J.P.Delescaille, "How easy is collision search? application to DES", Proc. of Crypto'89, 1990.

[16] Quisquater,J.J. and J.P.Delescaille, "How easy is colision search? New results and applications to DES", Proc. of Crypto'89, 1990.

[17] Mitchell,C.J., F.Piper and P.Wild, "Digital signatures", in contemporary cryptology : The science of information integrity, G.J. Simmons, editor, IEEE Press, pp.325-378, 1991.

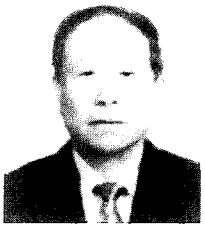
[18] 이국희, 박영호, 이상곤, 문상재, "CDMA 셀룰라 이동 통신을 위한 인증 시스템의 개발", 한국통신정보보호학회 종합학술대회 논문집, Vol.4, No.1, pp.81-90, 1994.

□ 著者紹介

염 흥 열(종신회원)

1981년 漢陽大學校 電子工學科 卒業(學士)
1983년 漢陽大學校 大學院 電子工學科 卒業(工學碩士)
1990년 漢陽大學校 大學院 電子工學科 卒業(工學博士)
1982년 12월 ~ 1990년 9월 韓國電子通信研究所 先任研究員
1990년 3월 ~ 현재 順天鄕大學校 工科大學 電子工學科 助教授

※ 관심분야 : 암호이론, 부호이론, 이동통신 분야



이 만 영(종신회원)

1924년 11월 30日生
서울大學校 電氣工學科 工學士(BSEE)
美國 Colorado 大學校 工學碩士(MSEE) 및 工學博士(Ph. D.)
美國 Virginia 州立大 工科大學 教授
美國 California Institute of Technology, JPL 責任研究員
國防科學研究所 第1副所長/韓國電子通信 社長/三星半導體通信 社長/漢陽大副總長
現 : 漢陽大 名譽教授/韓國通信情報保護學會 會長
著書 : Error Correcting Coding Theory, McGraw-Hill, New York, 1989.
Cryptography and Secure Communications, McGraw-Hill, New York, 1993.