

제 9 장

메시지 인증 코드



박 종 혁 교수

Tel: 970-6702

Email: jhpark1@seoultech.ac.kr

1절 메시지 인증 코드

2절 메시지 인증 코드 이용 예

3절 메시지 인증 코드의 실현 방법

4절 인증암호

5절 HMAC

6절 메시지 인증 코드에 대한 공격

7절 메시지 인증 코드로 해결할 수 없는 문제

제1절 메시지 인증 코드

1.1 올바른 송금 의뢰

1.2 메시지 인증 코드란?

1.3 메시지 인증 코드의 이용 순서

1.4 메시지 인증 코드의 키 배송 문제

1.1 올바른 송금 의뢰

- 앨리스: 은행 A의 고객
- 밥: 은행 B의 고객
- A은행에 앨리스로부터 송금 의뢰가 도착

내용:

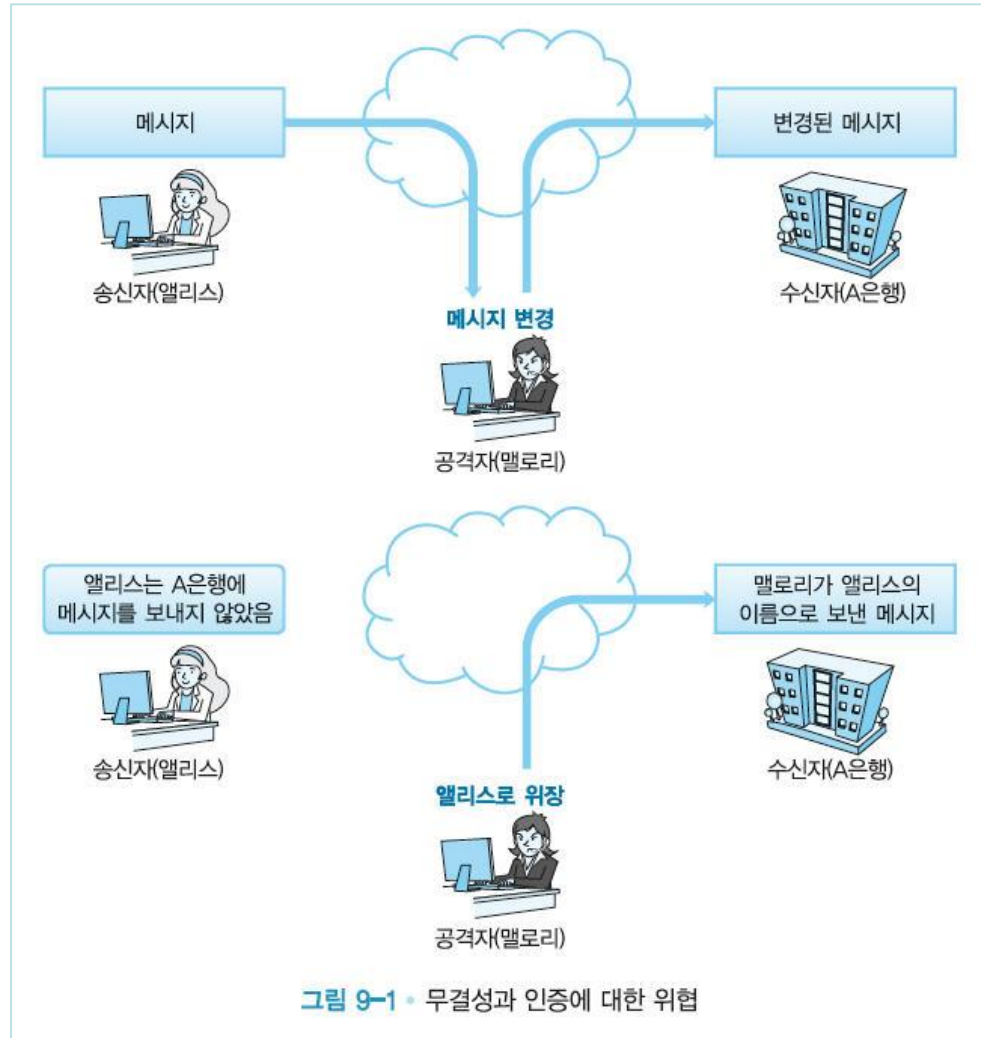
“내 계좌 앨리스-5374에서 B은행의 계좌 밥-6671로 1억 원을 송금바랍니다”

A 은행이 해야 할 일

- 메시지 출처: **인증**
- 통신 중 내용 변경 유무: **무결성**

- 메시지 인증(authentication)이란, 「메시지가 올바른 송신자로부터 온 것이다」 라는 성질

무결성과 인증에 대한 위협



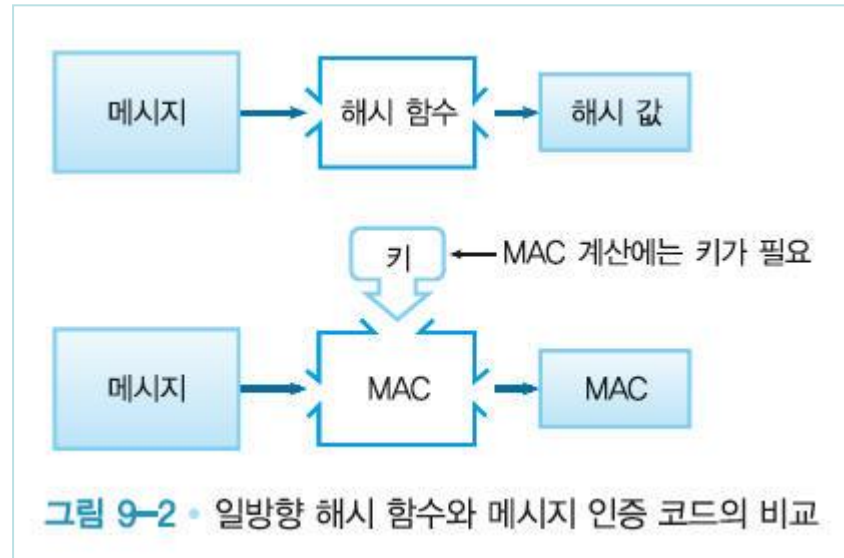
1.2 메시지 인증 코드란?

- 메시지 인증 코드(message authentication code)란, 무결성을 확인하고, 메시지에 대한 인증을 위한 코드
- 첫 글자를 따서 MAC이라 한다
- 입력: 메시지, 공유하는 키
- 출력: 고정 비트 길이의 코드

일방향 해시와 메시지 인증 코드

- 일방향 해시: 키를 사용하지 않는다
- 메시지 인증 코드: 키를 사용

일방향 해시 함수와 메시지 인증 코드의 비교



1.3 메시지 인증 코드의 이용 순서

- 1) 앨리스와 수신자 A은행: 사전에 키(K) 공유
- 2) 앨리스: 송금 의뢰 메시지(M) 작성 MAC 값($MAC_K(M)$)을 계산
- 3) 앨리스: 수신자 A은행으로 송금 의뢰 메시지와 MAC 값을 전송
- 4) 수신자 A은행: 수신한 송금 의뢰 메시지를 기초로 해서 MAC 값을 계산
- 5) 수신자 A은행: 앨리스로부터 수신한 MAC 값과 자신이 계산한 MAC 값을 비교
- 6) 수신자 A은행:
 - **인증성공**: 2개의 MAC 값이 동일하면 송금 의뢰가 틀림없이 앨리스로부터 온 것이라고 판단
 - **인증실패**: 2개의 MAC 값이 동일하지 않으면 앨리스로부터 온 것이 아니라고 판단

메시지 인증 코드의 이용 순서

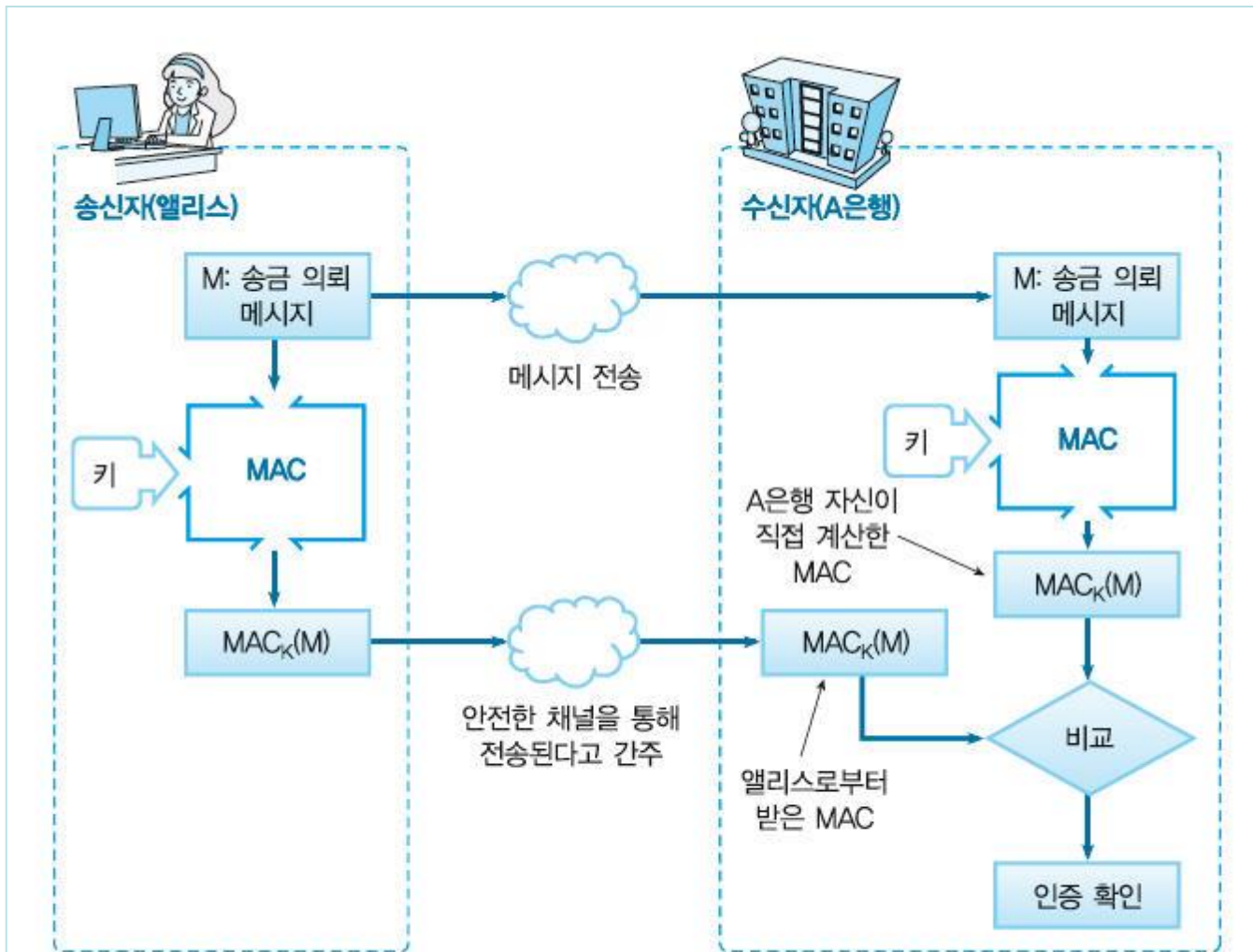


그림 9-1 • 메시지 인증 코드의 이용 순서

1.4 메시지 인증 코드의 키 배송 문제

- 대칭 암호 때의 「키 배송 문제」와 같은 문제가 메시지 인증 코드에도 발생
- 키 배송 문제를 해결
 - 공개 키 암호
 - Diffie-Hellman 키 교환
 - 키 배포 센터
 - 키를 안전한 방법으로 별도로 보내기

제2절 메시지 인증 코드 이용 예

2.1 SWIFT

2.2 Ipsec

2.3 SSL/TLS

2.1 SWIFT

- SWIFT는 Society for Worldwide Internet Financial Telecommunication(국제은행간 통신협회)
- 은행과 은행은 SWIFT를 통해서 거래 메시지 교환
- 메시지 인증 코드를 사용

2.2 Ipsec

- Ipsec
 - 인터넷 기반의 통신 프로토콜인 IP(Internet Protocol)에 보안 기능을 추가한 것
 - 통신 내용 인증과 무결성을 위해 메시지 인증 코드를 이용

2.3 SSL/TLS

- SSL/TLS
 - 웹에서 온라인 쇼핑을 할 때 사용되는 통신 프로토콜
 - 통신 내용의 인증과 무결성 확인을 위해 메시지 인증 코드를 이용

제3절 메시지 인증 코드의 실현 방법

3.1 일방향 해시 함수를 이용한 실현

3.2 블록 암호를 이용한 실현

3.3 그 밖의 방법으로 실현

3.1 일방향 해시 함수를 이용한 실현

- SHA-1나 MD5와 같은 일방향 해시 함수를 이용하여 메시지 인증 코드를 실현
- HMAC

3.2 블록 암호를 이용한 실현

- 트리플 DES나 AES와 같은 블록 암호를 사용해서 메시지 인증 코드를 실현
 - 블록 암호 키를 메시지 인증 코드의 공유키로 사용
 - CBC 모드로 메시지 전체를 암호화
 - 메시지 인증 코드에서는 복호화를 할 필요가 없으므로 최종 블록 이외는 폐기
 - 최종 블록을 MAC 값으로 이용

3.3 그 밖의 방법으로 실현

- 스트림 암호
- 공개 키 암호

4.1 GCM과 GMAC

인증 암호

- 인증 암호(AE혹은 AEAD)
 - 2000년 이후 연구 진행
 - 대칭 암호와 메시지 인증 코드를 조합하여 기밀성 · 무결성 · 인증을 동시에 충족시키는 구조
- Encrypt-then-MAC
- Encrypt-and-MAC
- MAC-then-Encrypt

Encrypt-then-MAC

- 평문을 대칭 암호로 암호화한 후 암호문의 MAC 값을 계산
- 메시지 인증 코드 입력에 암호문을 부여
- 선택 암호문 공격을 막을 수 있다

Encrypt-and-MAC

- 평문을 대칭 암호로 암호화하고, 그와는 별도로 평문의 MAC 값을 얻는 방법

MAC-then-Encrypt

- 미리 평문의 MAC 값을 얻고, 평문과 MAC 값 양쪽을 정리하여 대칭 암호로 암호화하는 방법

4.1 GCM과 GMAC

- GCM(Galois/Counter Mode)
 - 인증 모드의 일종
 - AES와 같은 128비트 블록 암호를 CTR 모드로 이용하여 MAC 값을 얻기 위하여 덧셈과 곱셈을 반복하는 해시 함수를 사용
 - CTR 모드는 1씩 늘어가는 숫자를 암호화하기 때문에 각 블록을 병렬 처리하여 실행속도를 높일 수가 있음
 - CTR 모드와 MAC 값 생성에 공통 키를 사용하기 때문에 키 관리도 편리

4.1 GCM과 GMAC

- GMAC(Galois/Counter Mode MAC)
 - GCM을 메시지 인증 코드 전용으로 사용

5.1 HMAC이란?

5.2 HMAC의 순서

5.1 HMAC이란?

- HMAC은 일방향 해시 함수를 이용하여 메시지 인증 코드를 구성하는 방법
- HMAC의 일방향 해시 함수는 모듈형으로 골라서 사용
 - HMAC-SHA1: SHA-1
 - HMAC-MD5: MD5
 - HMAC-RIPEMD: RIPEMD-160

RFC 2104 정의

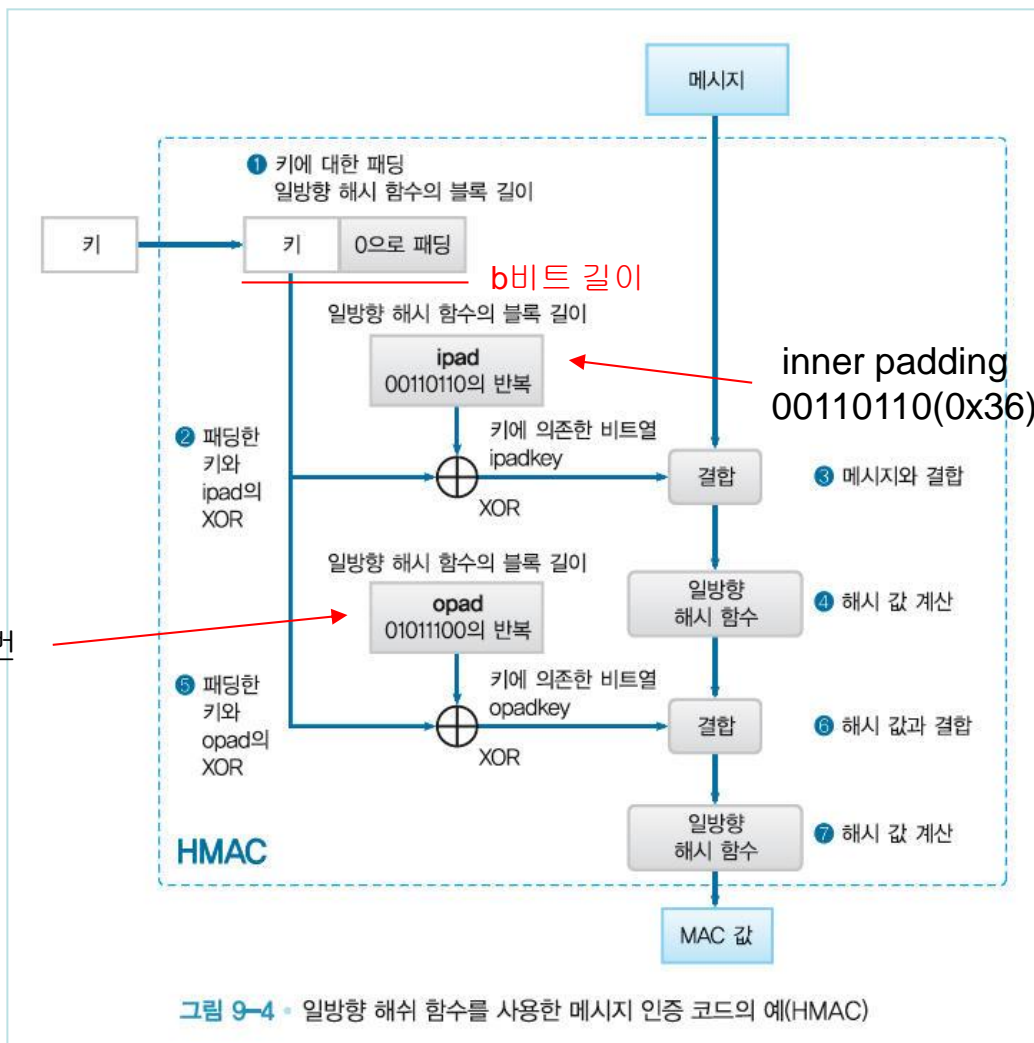
$$\text{HMAC}(K, m) = \text{H} \left((K' \oplus \text{opad}) \parallel \text{H} \left((K' \oplus \text{ipad}) \parallel m \right) \right)$$
$$K' = \begin{cases} \text{H}(K) & K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

5.2 HMAC의 순서

- 1) 키 패딩
- 2) 패딩한 키와 ipad의 XOR
- 3) 메시지 결합
- 4) 해시 값의 계산
- 5) 패딩한 키와 opad의 XOR
- 6) 해시 값과의 결합
- 7) 해시 값의 계산

일방향 해시 함수를 사용한 메시지 인증 코드의 예(HMAC)

outer padding
01011100(0x5C)을 $b/8$ 번
반복한 2진 수열



제6절 메시지 인증 코드에 대한 공격

6.1 재전송 공격

6.2 키의 추측에 의한 공격

6.1 재전송 공격

- 재전송(replay)공격
 - 보존해 둔 정당한 MAC 값을 반복해서 송신하는 공격

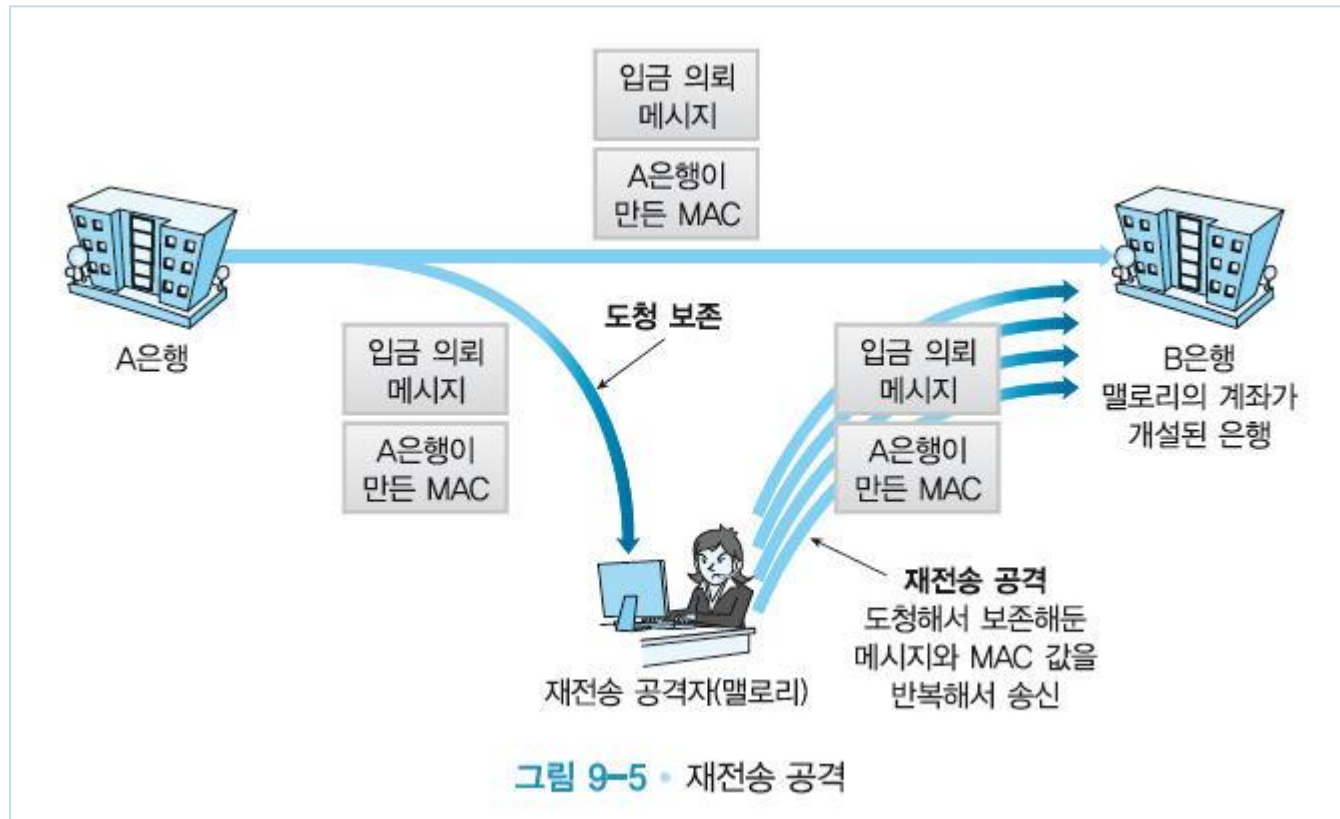
재전송 공격의 예

- 1) 맬로리는 A은행과 B은행의 통신을 도청하고 있다.
- 2) 맬로리는 A은행에 가서 B은행에 있는 자신의 계좌 M-2653으로 100만원을 입금한다. 그러면 A은행은 다음과 같은 입금 의뢰 메시지를 만든다.
「계좌 M-2653으로 100만원을 입금하라」
A은행은 이 입금 의뢰 메시지의 바른 MAC 값을 계산하여 그것을 메시지에 추가해서 B은행에 보낸다.
- 3) B은행은 받은 메시지를 기초로 MAC 값의 계산을 행하고, 그 계산 결과와 보내 온 MAC 값을 비교한다. 그리고 2개의 MAC 값이 같다는 것으로부터 A은행에서 온 메시지는 올바른 입금 의뢰라고 판단하고, 맬로리의 계좌 M-2653으로 100만원을 입금한다.

재전송 공격의 예

- 4) 맬로리는 A은행에서 B은행으로 보내진 입금 의뢰 메시지와 MAC 값을 도청하여 자신의 컴퓨터에 보존한다.
- 5) 맬로리는 보존해 둔 입금 의뢰 메시지와 MAC 값을 한 번 더 B은행에 보낸다.
- 6) B은행은 받은 메시지를 기초로 MAC 값의 계산을 행하고, 그 계산 결과와 보내 온 MAC 값을 비교한다. 그리고 2개의 MAC 값이 같다는 것으로부터 이 메시지는 A은행에서 온 올바른 입금 의뢰라고 판단(오해)하고, 맬로리의 계좌 M-2653에 100만 원을 입금해 버린다.
- 7) 맬로리는 (5)를 100번 반복한다.
- 8) B은행은 (6)을 100번 반복해 버린다.
- 9) 맬로리는 B은행의 자신의 계좌에 $100\text{만원} \times 100 = 1\text{억원}$ 이 입금되면 바로 인출한다.

재전송 공격



재전송 공격 방어

- **순서 번호(sequence number)**
 - 송신 메시지에 매회 1씩 증가하는 번호(순서 번호, sequence number)를 붙이기
 - 마지막 통신시 순서번호를 저장
- **타임스탬프(timestamp)**
 - 송신 메시지에 현재 시각 넣기
 - 송수신자 사이의 동기화 필요
- **비표(nonce)**
 - 송신자에게 일회용의 랜덤한 값을 전송
 - 메시지와 비표를 합해 MAC 값을 계산
 - 비표 값은 통신 때마다 교체

6.2 키의 추측에 의한 공격

- 메시지 인증 코드에 대한 공격
 - 전사공격
 - 생일 공격
- MAC 값만 획득한 공격자가 키를 추측하지 못하도록 해야 한다
 - 해시 함수의 일방향성
 - 해시 함수의 충돌내성
 - 키 생성에 의사난수 생성기 사용

제7절 메시지 인증 코드로 해결할 수 없는 문제

7.1 제 3자에 대한 증명

7.2 부인방지

7.1 제 3자에 대한 증명

- 앨리스로부터 메시지를 받은 밥이 「이 메시지는 앨리스가 보낸 것이다」라는 것을 제삼자인 검증자 빅터에게 증명할 수 없다
- 이유:
 - 일단 키를 빅터에게 알려줘야 한다
 - 앨리스와 밥 모두가 키를 가지고 있으므로 둘 중 누가 작성했는지 말할 수 없다

7.2 부인 방지

- 밥이 MAC 값이 딸린 메시지를 받았고, 「이 메시지는 앨리스로부터 온 것이다」라는 걸 확실히 알 수 있다
- 하지만 앨리스가 전송 자체를 부정할 경우 제3자에게 이 사실을 증명할 수 없다
- 앨리스의 송신자체에 대한 부정을 부인(repudiation)이라고 한다
- 메시지 인증 코드로는 부인 방지(nonrepudiation)를 할 방법이 없다

Q & A

Thank You!