2026/07/09, 2026 보안 기초 세미나

암호학과 네트워크 보안

- 11장 메시지 무결성과 메시지 인증 -

김 민 식 (minsik@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 보충
- 메시지 무결성(Message Integrity)
- 랜덤 오라클 모델(Random Oracle Model)
- 메시지 인증(Message Authentication)

목 차

- 보충
- 메시지 무결성(Message Integrity)
- 랜덤 오라클 모델(Random Oracle Model)
- 메시지 인증(Message Authentication)

보충

- 대칭 키와 비대칭 키 암호시스템 비교

구분	대칭 키 암호시스템	비대칭 키 암호시스템
키 구조	송신자와 수신자가 같은 키 사용	공개키와 개인키 서로 다른 키 쌍 사용
키 개수	통신 쌍마다 1개의 비밀키 필요	사용자마다 공개키 1개, 개인키 1개 필요
키 배포	비밀키를 안전하게 공유해야 하므로 어려움	공개키는 공개 채널로 배포 가능함
속도	빠름	상대적으로 느림
주요 용도	대용량 데이터 암호화	공개키로 암호화하는 경우: 암호화에 사용할 비밀키 전달 개인키로 암호화하는 경우: 전자서명 생성, 인증서 서명

목 차

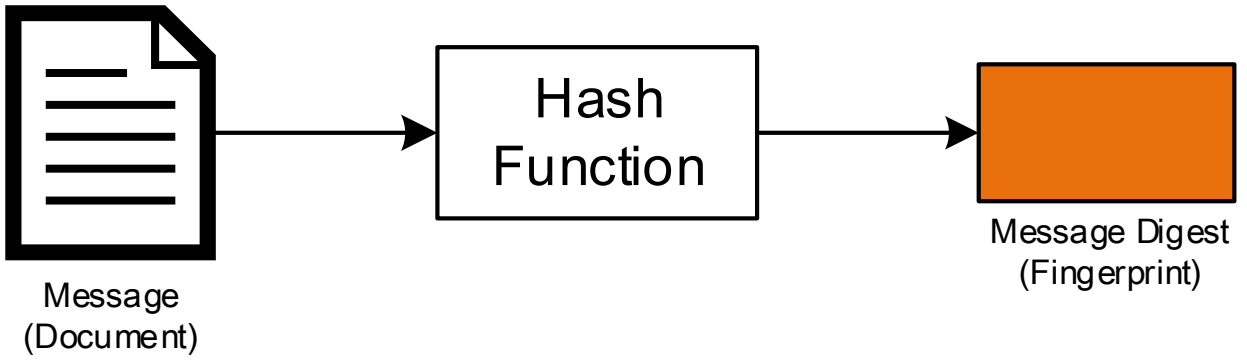
- 보충
- 메시지 무결성(Message Integrity)
- 랜덤 오라클 모델(Random Oracle Model)
- 메시지 인증(Message Authentication)

메시지 무결성

- 무결성(Integrity)
  - 무결성은 정보가 변조·훼손되지 않고 정확성과 완전성을 유지하는 특성
- 메시지 무결성(Message Integrity)
  - 메시지가 전송·저장 과정에서 변조·삭제·삽입되지 않고 원래 내용을 유지하는 성질

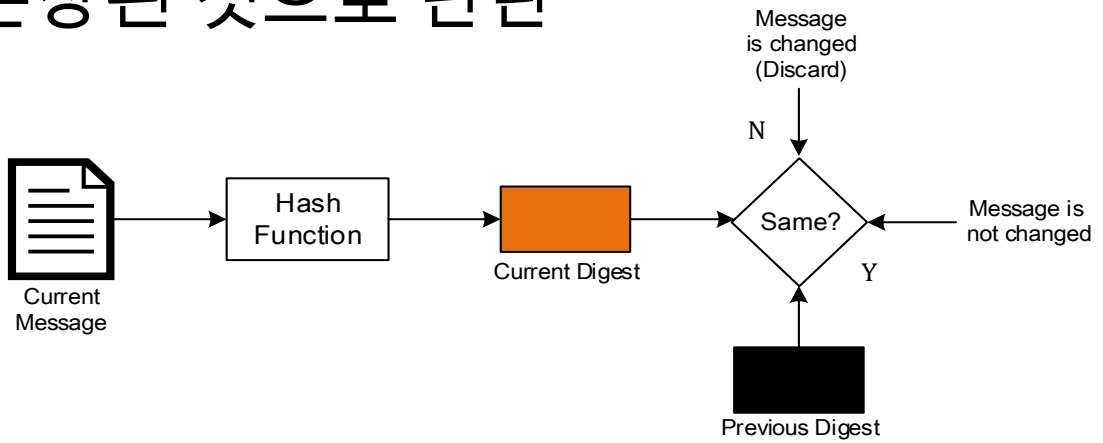
메시지 무결성

- 무결성 확인 방법 (1/3)
  - 다이제스트(Digest)
    - 메시지·문서와 같은 데이터에 해시 함수를 적용해 생성한 고정 길이의 결과값
  - 핑거프린트(Fingerprint)
    - 생성된 다이제스트를 데이터의 식별이나 무결성 확인을 위한 기준값으로 사용하는 것



메시지 무결성

- 무결성 확인 방법 (2/3)
  - 핑거프린트를 이용한 확인 과정
    1. 원본 메시지에 해시 함수를 적용하여 기준 다이제스트 생성
    2. 기준 다이제스트를 원본 메시지의 핑거프린트로 사용
    3. 수신 메시지에 동일한 해시 함수를 적용하여 새로운 다이제스트 생성
    4. 기준 핑거프린트와 새로 계산한 다이제스트 비교
    5. 두 값이 같으면 메시지가 전송 중 변경되지 않은 것으로 판단, 다르면 변조·손상된 것으로 판단

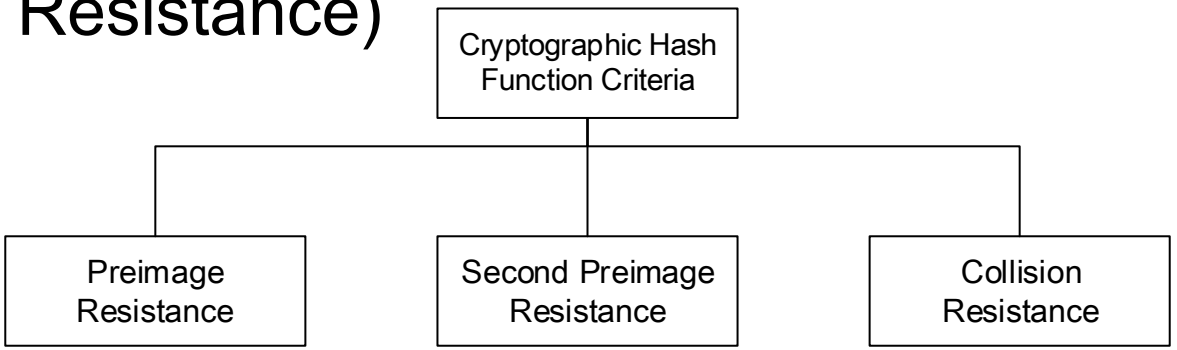


메시지 무결성

- 무결성 확인 방법 (3/3)
  - 문서와 메시지 차이
    - 문서(Document)
      - 정보의 기록과 보관을 목적으로 하는 정적인 정보 단위
    - 메시지(Message)
      - 정보의 송수신을 목적으로 하는 동적인 정보 단위
  - 문서 핑거프린트와 메시지 다이제스트 차이
    - 문서 핑거프린트(Document Fingerprint)
      - 물리적으로 서로 연결되어있음
    - 메시지 다이제스트(Message Digest)
      - 물리적으로 연결 해제(또는 전송)할 수 있음

메시지 무결성

- 암호학적 해시 함수(Cryptographic Hash Function)
  - 정의
    - 임의 길이의 데이터를 고정 길이의 해시값으로 변환하며, 원문 복원과 동일한 해시값 생성이 어렵도록 설계된 함수
  - 보안 기준
    - 프리이미지 저항성(Preimage Resistance)
    - 제2 프리이미지 저항성(Second Preimage Resistance)
    - 충돌 저항성(Collision Resistance)



메시지 무결성

- 암호학적 해시 함수(Cryptographic Hash Function)
  - 프리이미지 저항성(Preimage Resistance) (1/3)
    - 주어진 해시값 y에 대해 $h(M) = y$를 만족하는 입력 메시지 M을 찾기 어려운 성질
    - 예제 11.1

어떤 사용자의 비밀번호에 SHA-256 해시 함수를 적용하여 다음과 같은 다이제스트를 저장한다고 하자.

$$D = \text{SHA-256(비밀번호)}$$

서버에는 원래 비밀번호가 저장되어 있지 않고, SHA-256으로 생성된 다이제스트 D만 저장되어 있다.
이때 제3자가 다이제스트 D만 보고 원래 비밀번호를 쉽게 알아낼 수 있는가?

- 풀이
  - SHA-256은 입력값을 고정 길이의 다이제스트로 변환하는 암호학적 해시 함수임
  - 다이제스트는 원문을 직접 포함하지 않고, 입력값이 같은지 비교하기 위한 값이므로, 제3자가 다이제스트 D만 보고 이를 생성한 원래 비밀번호를 찾아내기 어려움

메시지 무결성

- 암호학적 해시 함수(Cryptographic Hash Function)
  - 프리이미지 저항성(Preimage Resistance) (2/3)
    - 예제 11.2

StuffIt와 같은 전통적인 손실이 없는 압축 방법을 암호학적 해시 함수로 사용할 수 있는가?

- 풀이
  - StuffIt은 여러 파일을 하나로 묶고 크기를 줄여 저장·전송하는 무손실 압축 프로그램이자 압축 형식임
  - ZIP과 유사하며, 압축을 해제하면 원본 파일을 그대로 복원할 수 있음
  - 원본 복원이 가능하므로 프리이미지 저항성을 만족하지 못하며, 출력 길이도 고정되지 않아 암호학적 해시 함수로 사용할 수 없음

메시지 무결성

- 암호학적 해시함수(Cryptographic Hash Function)
  - 프리이미지 저항성(Preimage Resistance) (3/3)
    - 예제 11.3

검사합(checksum)함수를 암호학적 해시 함수로 사용할 수 있는가?

- 풀이
  - 검사합은 데이터에 일정한 계산을 적용해 생성한 값으로, 저장·전송 과정에서 발생한 오류를 확인하는 데 사용함
  - 수신한 데이터의 검사합을 다시 계산하여 기존 검사합과 비교하고, 값이 다르면 데이터에 오류가 발생한 것으로 판단함
  - 검사합은 우연한 오류 검출이 목적이며, 같은 검사합을 갖는 데이터를 쉽게 만들 수 있어 충돌 저항성 등의 암호학적 안전성을 만족하지 못하므로 암호학적 해시 함수로 사용할 수 없음

메시지 무결성

- 암호학적 해시 함수(Cryptographic Hash Function)
  - 제 2 프리이미지 저항성(Second Preimage Resistance)
    - 주어진 메시지 M에 대해 $h(M) = h(M')$을 만족하는 다른 메시지 $M'(\neq M)$을 찾기 어려운 성질
    - 예제 11.4

어떤 계약서 파일 M에 SHA-256 해시 함수를 적용하여 다음과 같은 다이제스트를 저장한다고 하자.

$$D = \text{SHA-256}(M)$$

공격자는 원본 계약서 M과 다이제스트 D를 모두 알고 있다. 이때 공격자가 원본 계약서와 내용이 다른 위조 계약서 M'을 만들어, $D = \text{SHA-256}(M')$ 를 만족시킬 수 있는가?

- 풀이
  - SHA-256은 입력값이 조금만 바뀌어도 다이제스트가 예측하기 어렵게 크게 달라지는 특성을 가짐
  - 따라서 계약서의 금액, 날짜, 문장, 공백 등을 조금씩 바꾸더라도 원본과 같은 다이제스트가 나오도록 조정하기 어려움
  - 가능한 다이제스트 값이 2^{256}개로 매우 많기 때문에, 위조 메시지가 원본과 같은 다이제스트를 가질 확률은 매우 낮음

메시지 무결성

- 암호학적 해시 함수(Cryptographic Hash Function)
  - 충돌 저항성(Collision Resistance)
    - $h(M) = h(M')$을 만족하는 서로 다른 두 메시지 M과 M'을 찾기 어려운 성질
    - 예제 11.5

공격자가 서로 다른 두 문서 M과 M'을 만든다고 하자($M \neq M'$), 이때 공격자가 두 문서에 SHA-256을 적용했을 때 같은 다이제스트(SHA-256(M) =SHA-256(M'))가 나오도록 만들 수 있는가?

- 풀이
  - 충돌 저항성은 서로 다른 두 입력이 같은 다이제스트를 갖는 쌍을 찾기 어렵게 하는 성질임
  - 이 경우 공격자는 특정 원본 메시지를 기준으로 하는 것이 아니라, 서로 다른 두 문서 M, M'를 모두 직접 선택함
  - SHA-256은 입력이 조금만 달라져도 다이제스트가 예측하기 어렵게 변하므로, 두 문서의 내용을 조정하더라도 같은 다이제스트가 나오도록 맞추기 어려움
  - SHA-256의 다이제스트는 256비트이므로 가능한 다이제스트 값은 2^{256}개로 매우 많아 맞추기 어려움

목 차

- 보충
- 메시지 무결성(Message Integrity)
- 랜덤 오라클 모델(Random Oracle Model)
- 메시지 인증(Message Authentication)

랜덤 오라클 모델

- 랜덤 오라클 모델(Random Oracle Model)
  - 정의
    - 해시 함수가 이상적인 무작위 함수처럼 동작한다고 가정하여 암호 시스템의 안전성을 분석하는 이론적 모델
  - 동작 원리
    1. 새로운 메시지가 입력되면 고정 길이의 다이제스트를 무작위로 생성하여 대응표에 기록함
    2. 이전에 입력된 메시지가 다시 입력되면 새 값을 생성하지 않고 기록된 동일한 다이제스트를 반환함
    3. 새로운 메시지의 다이제스트는 이전에 생성된 다이제스트들과 독립적으로 무작위 선택됨

랜덤 오라클 모델

- 랜덤 오라클 모델(Random Oracle Model)
  - 예제 11.6 (1/3)

16비트 메시지 다이제스트를 생성하는 랜덤 오라클의 현재 대응표가 다음과 같다고 하자.

a. 메시지 AB1234CDB765BDAD에 대한 메시지 다이제스트를 생성하시오. 단, 동전을 16번 던진 결과는 HHTHHHTTHTHHTTTH이며, 앞면 H는 1, 뒷면 T는 0으로 대체한다.
b. 메시지 4523AB1352CDEF45126에 대한 메시지 다이제스트를 생성하시오 .

<처음 3개의 다이제스트를 생성한 후의 Oracle 표>

Message	Message Digest
4523AB1352CDEF45126	13AB
723BAE38F2AB3457AC	02CA
AB45CD1048765412AAAB6662BE	A38B

랜덤 오라클 모델

- 랜덤 오라클 모델(Random Oracle Model)
  - 예제 11.7 (2/3)

16비트 메시지 다이제스트를 생성하는 랜덤 오라클의 현재 대응표가 다음과 같다고 하자.

a. 메시지 AB1234CDB765BDAD에 대한 메시지 다이제스트를 생성하시오. 단, 동전을 16번 던진 결과는 HHTHHHTTHTHHTTTH이며, 앞면 H는 1, 뒷면 T는 0으로 대체한다.
b. 메시지 4523AB1352CDEF45126에 대한 메시지 다이제스트를 생성하시오 .

- 풀이
  a. 메시지 AB1234CDB765BDAD는 오라클 표에 존재하지 않으므로 동전을 16번 던져 새로운 값을 생성함
     - $HHTHHHTTHTHHTTTH \rightarrow 1101110010110001_2 \rightarrow DCB1_{16}$

<처음 3개의 다이제스트를 생성한 후의 Oracle 표>

Message	Message Digest
4523AB1352CDEF45126	13AB
723BAE38F2AB3457AC	02CA
AB45CD1048765412AAAB6662BE	A38B

→

<4개의 수를 제공하는 Oracle 표>

Message	Message Digest
4523AB1352CDEF45126	13AB
723BAE38F2AB3457AC	02CA
AB1234CD8765BDAD	DCB1
AB45CD1048765412AAAB6662BE	A38B

랜덤 오라클 모델

- 랜덤 오라클 모델(Random Oracle Model)
  - 예제 11.8 (3/3)

16비트 메시지 다이제스트를 생성하는 랜덤 오라클의 현재 대응표가 다음과 같다고 하자.

a. 메시지 AB1234CDB765BDAD에 대한 메시지 다이제스트를 생성하시오. 단, 동전을 16번 던진 결과는 HHTHHHTTHTHHTTTH이며, 앞면 H는 1, 뒷면 T는 0으로 대체한다.
b. 메시지 4523AB1352CDEF45126에 대한 메시지 다이제스트를 생성하시오 .

- 풀이
  b. 메시지 4523AB1352CDEF45126은 오라클 표에 이미 존재하므로 동전을 다시 던지지 않고, 저장된 메시지 다이제스트 13AB를 반환함

<4개의 수를 제공하는 Oracle 표>

Message	Message Digest
4523AB1352CDEF45126	13AB
723BAE38F2AB3457AC	02CA
AB1234CD8765BDAD	DCB1
AB45CD1048765412AAAB6662BE	A38B

랜덤 오라클 모델

- 랜덤 오라클 모델(Random Oracle Model)
  - 예제 11.9

해시 함수 $h(M) = M \ mod \ n$을 사용하는 오라클에서 $h(M_1)$과 $h(M_2)$가 이미 생성되어 있고
$M_3 = M_1 + M_2$일 때, $h(M_3)$를 구하고 이 방식이 랜덤 오라클의 요구조건을 위배하는 이유를 설명하시오.

  - 풀이
    - $M_3 = M_1 + M_2$ 이므로 $h(M_1)$과 $h(M_2)$를 알고 있으면 M_3의 해시값을 새로 계산하지 않고도 구할 수 있음

$$h(M_3) = (M_1 + M_2) \ mod \ n = (M_1 mod \ n + M_2 \ mod \ n) \ mod \ n = \big(h(M_1) + h(M_2)\big) \ mod \ n$$

    - 이는 새로운 입력의 출력값이 기존 값과 무관하게 무작위로 결정되어야 한다는 랜덤 오라클의 요구조건을 위배함

랜덤 오라클 모델

- 비둘기 집 원리(Pigeonhole Principle)
  - 원리
    - 만약 $n+1$마리의 비둘기가 n개의 비둘기 집에 들어가 있다면 적어도 한 비둘기 집에는 두 마리의 비둘기가 들어있다는 의미
  - 해시에서의 원리
    - 가능한 메시지의 수가 가능한 다이제스트의 수보다 많으므로 서로 다른 메시지가 같은 다이제스트에 대응하는 충돌이 반드시 존재함
    - 가능한 메시지의 집합과 가능한 다이제스트의 집합 사이의 관계는 다대일 관계임
  - 필요성
    - 다이제스트 길이에 따른 충돌 가능성을 분석하고, 전자서명·파일 무결성 검증에 사용할 안전한 해시 함수를 선택하는 기준으로 활용됨

랜덤 오라클 모델

- 비둘기 집 원리(Pigeonhole Principle)
  - 예제 11.10

메시지 길이가 6비트이고 다이제스트 길이가 4비트인 해시 함수에서 가능한 메시지와 다이제스트의 개수를 구하고, 비둘기집 원리에 따라 하나의 다이제스트에 최소 몇 개의 메시지가 대응하는지 구하시오.

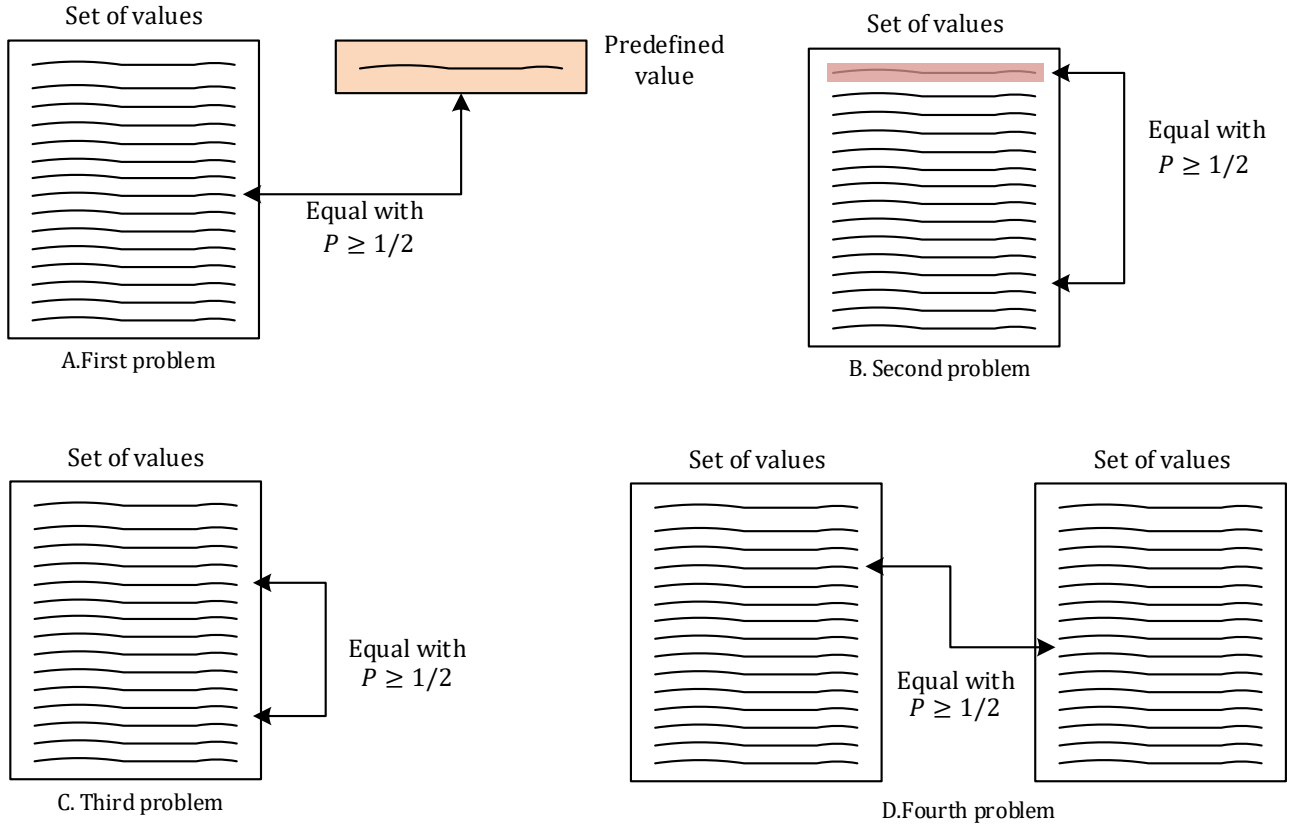
- 풀이
  - 가능한 다이제스트의 개수는 $2^4 = 16$이고 가능한 메시지 개수는 $2^6 = 64$이므로 64개의 메시지를 16개의 다이제스트에 대응시키면 $64 \div 16 = 4$이므로 다이제스트 하나당 평균 4개의 메시지가 대응함
  - 따라서 비둘기집 원리에 의해 하나의 다이제스트에는 최소 4개의 메시지가 대응함

랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 원리
    - 가능한 생일이 365개일 때 사람 수가 증가할수록 같은 생일을 가진 두 사람이 존재할 확률이 빠르게 증가한다는 원리
  - 해시에서의 원리
    - 가능한 값이 n개일 때 약 $\sqrt{n}$개의 표본만으로도 두 값이 충돌할 확률이 약 50%에 도달한다는 원리
  - 필요성
    - m비트 해시 함수의 충돌 탐색 난이도가 2^m이 아니라 약 $2^{m/2}$임을 계산하여 실제 보안 수준을 평가하기 위한 원리
    - 전자서명, 인증서, 파일 무결성 검증에서 안전한 해시 함수 선택 기준으로 활용

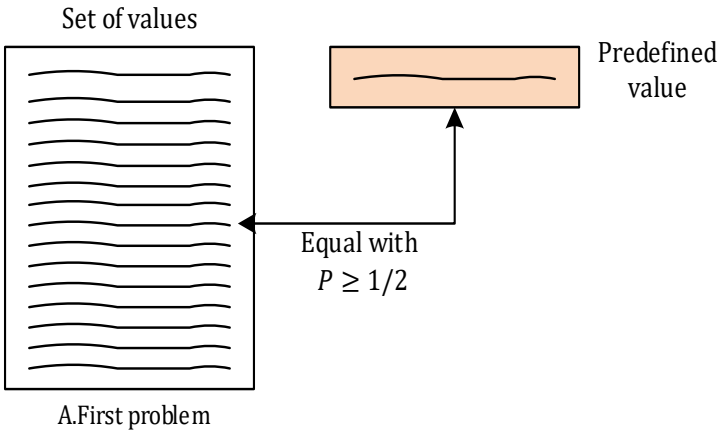
랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 4가지 생일 문제



랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 첫 번째 문제-프리이미지 공격 난이도
    - 교실 안에 학생이 k명 있을 때 적어도 한 학생이 특정 생일일 확률 P가 $1/2$보다 크거나 같으려면 k의 최소값이 무엇인가?
  - 일반화
    - N개의 값(0부터 $N-1$)에 균등하게 분포된 확률 변수가있다. 적어도 한 개의 확률 변수가 특정한 값 (0과 $N-1$사이의 미리 지정된 값)을가질 확률이 $1/2$ 이상이 되려면 최소한도 몇 개의 확률 변수가 필요한가?

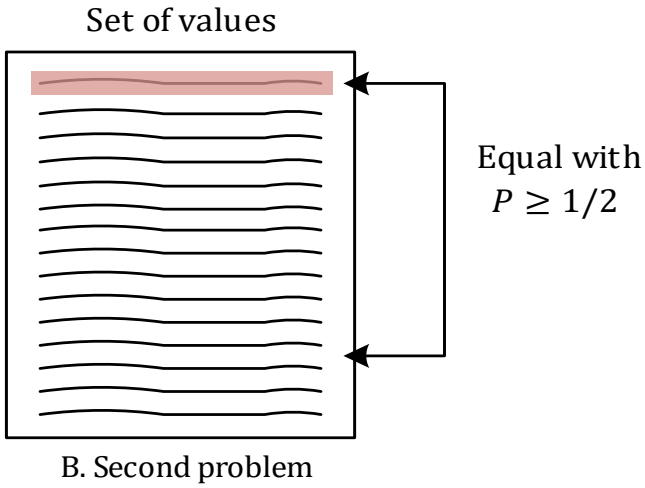


랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 첫 번째 문제-프리이미지 공격 난이도
    - 풀이-확률 P
      1. 한 학생이 특정 생일일 확률은 가능한 생일이 N개이고 각 생일의 확률이 같으므로: $\frac{1}{N}$
      2. 한 학생이 특정 생일이 아닐 확률은 전체 확률 1에서 특정 생일일 확률을 빼면: $1-\frac{1}{N}$
      3. k명의 학생 모두 특정 생일이 아닐 확률은 각 학생의 생일이 독립적으로 결정되므로: $\left(1-\frac{1}{N}\right)^k$
      4. 적어도 한 학생이 특정 생일일 확률은 전체 확률 1에서 k명 모두 특정 생일이 아닐 확률을 빼면: $P=1-\left(1-\frac{1}{N}\right)^k$
      5. $P \geq \frac{1}{2}, N=365$를 대입하면 $k \geq \frac{\ln\left(\frac{1}{2}\right)}{\ln\left(\frac{364}{365}\right)} \approx 252.65$이므로 최소 253명이 필요함

랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 두 번째 문제-제2 프리이미지 공격 난이도
    - 교실 안에 학생이 k명 있을 때 적어도 한 학생이 교사가 미리 선정한 학생과 같은 생일일 확률 P가 $1/2$ 보다 크거나 같으려면 k의 최소값이 무엇인가?
  - 일반화
    - N개의 값(0부터 $N-1$)을 갖는 균등 분포 확률 변수가 있다. 적어도 한 개의 확률 변수가 미리 선정된 확률 변수 값(0과 $N-1$사이의 미리 지정된 값)을 가질 확률이 $1/2$ 이상이 되려면 최소한 몇 개의 확률 변수가 필요한가?

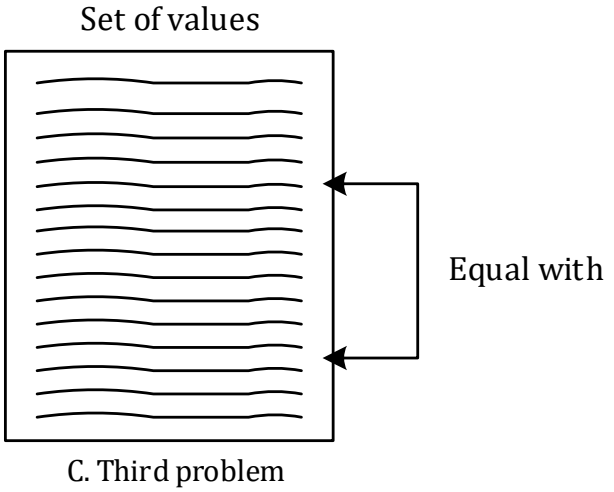


랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 두 번째 문제-제2 프리이미지 공격 난이도
    - 풀이-확률 P
      1. 미리 선정된 학생 1명의 생일을 기준값으로 설정
      2. 나머지 한 학생이 기준 학생과 같은 생일일 확률은: $\frac{1}{N}$
      3. 나머지 한 학생이 기준 학생과 다른 생일일 확률은: $1-\frac{1}{N}$
      4. 전체 k명 중 기준 학생을 제외한 $k-1$명이 모두 다른 생일일 확률은: $\left(1-\frac{1}{N}\right)^{k-1}$
      5. 따라서 적어도 한 학생이 기준 학생과 같은 생일일 확률 P는 $P=1-\left(1-\frac{1}{N}\right)^{k-1}$
      6. $P \geq \frac{1}{2}, N=365$를 대입하면 $k \geq 1+\frac{\ln\left(\frac{1}{2}\right)}{\ln\left(\frac{364}{365}\right)} \approx 253.65$이므로 최소 254명이 필요함

랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 세 번째 문제-충돌 공격 난이도
    - 교실 안에 학생이 k명 있을 때 적어도 두 학생이 같은 생일일 확률 P가 $1/2$보다 크거나 같으려면 k의 최솟값이 무엇인가?
  - 일반화
    - N개의 값(0부터 $N-1$)을 갖는 균등 분포확률 변수가 있다. 적어도 두 개의 확률 변수 값이 동일할 확률이 $1/2$ 이상이 되려면 최소한 몇 개의 확률 변수가 필요한가?



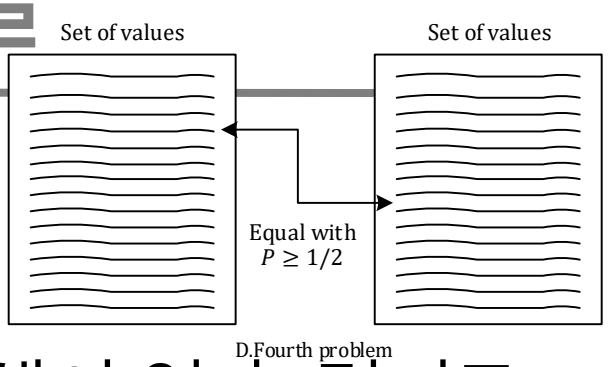
랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 세 번째 문제-충돌 공격 난이도
    - 풀이-확률 P (1/2)
      1. 가능한 값의 개수를 N, 선택하는 표본의 개수를 k라고 설정
      2. 첫 번째 표본은 어떤 값을 가져도 되므로 확률은: 1
      3. 두 번째 표본이 첫 번째 표본과 다른 값을 가질 확률은: $1-\frac{1}{N}$
      4. 세 번째 표본이 앞의 두 표본과 모두 다른 값을 가질 확률은: $1-\frac{2}{N}$
      5. 같은 방식으로 k번째 표본이 앞의 $k-1$개 표본과 모두 다른 값을 가질 확률은: $1-\frac{k-1}{N}$
      6. 따라서 k개의 표본이 모두 서로 다른 값을 가질 확률 Q는 $Q=1\times(1-\frac{1}{N})\times(1-\frac{2}{N})\times\cdots\times(1-\frac{k-1}{N})$
      7. 적어도 두 표본이 동일한 값을 가질 확률 P는 전체 확률 1에서 Q를 빼면 되므로: $P=1-Q$

랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 세 번째 문제-충돌 공격 난이도
    - 풀이-확률 P (2/2)
      8. 식을 간단히 근사하면 :$P \approx 1 - e^{-\frac{k(k-1)}{2N}}$
      9. $k(k-1) \approx k^2$로 근사하면: $P \approx 1 - e^{-\frac{k^2}{2N}}$
      10. $P \geq \frac{1}{2}$가 되도록 계산하면: $1 - e^{-\frac{k^2}{2N}} \geq \frac{1}{2}$
      11. 따라서 $k \approx 1.18\sqrt{N}$ 이 됨
      12. 생일 문제에 $N = 365$를 대입하면: $k \approx 1.18\sqrt{365} \approx 22.5$
      13. 따라서 최소 정수 k는 23임

랜덤 오라클 모델



- 생일 문제(Birthday Problems)
  - 네 번째 문제-또 다른 충돌 공격 난이도
    - 두 개의 반이 있는데 각 반에는 k명의 학생이 있다. 적어도 첫 번째 반에서 선택된 한 학생과 다른 반에서 선택된 한 학생의 생일이 같을 확률 P가 $1/2$ 이상이 되기 위해 필요한 k의 최솟값은 무엇인가?
  - 일반화
    - N개의 값$(0$부터 $N-1)$을 갖는 균등 분포확률 변수가 있다. 여기서 각각 k개의 확률 변수로 이루어진 두 집단을 생각해 보자. 이때 적어도 첫 번째 집단에서 선택한 한 확률 변수의 값이 다른 집단에서 선택한 한 확률 변수의 값과 동일할 확률이 $1/2$ 이상이 되려면 k의 최솟값은 무엇인가?

랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 네 번째 문제-또 다른 충돌 공격 난이도
    - 풀이-확률 P (1/2)
      1. 가능한 값의 개수를 N이라고 하고, 두 집단 A와 B에 각각 k개의 확률변수가 있다고 설정
      2. 첫 번째 집단 A에서 선택한 한 값과 두 번째 집단 B의 한 값이 서로 다를 확률은: $1-\frac{1}{N}$
      3. A에서 선택한 한 값이 B의 k개 값 모두와 다를 확률은: $\left(1-\frac{1}{N}\right)^k$
      4. A에도 값이 k개 있으므로, 두 집단 사이에서 같은 값이 하나도 나타나지 않을 확률 Q는 근사적으로: $Q \approx \left[\left(1-\frac{1}{N}\right)^k\right]^k$
      5. 두 집단 사이에서 적어도 한 쌍의 값이 같을 확률 P는 전체 확률 1에서 Q를 빼면 되므로 :$P \approx 1-\left[\left(1-\frac{1}{N}\right)^k\right]^k$

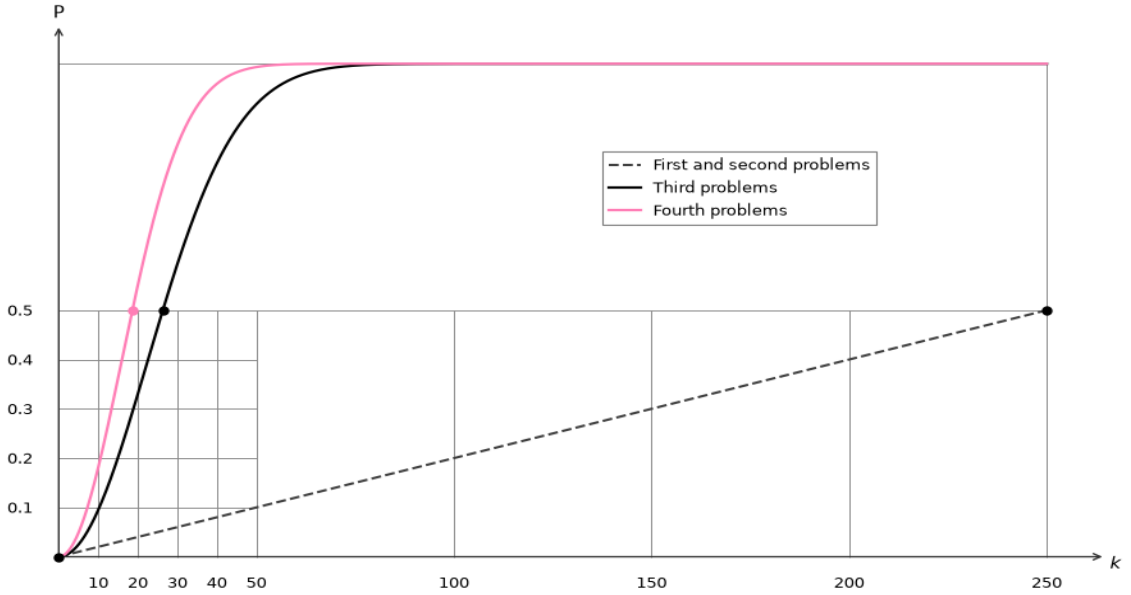
랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 네 번째 문제-또 다른 충돌 공격 난이도
    - 풀이-확률P (2/2)

6. 이를 지수함수로 근사하면: $P \approx 1 - e^{-\frac{k^2}{N}}$
7. 확률 P가 $\frac{1}{2}$이상이 되는 조건을 계산하면: $\frac{k^2}{N} \geq ln2$
8. 따라서 $k \geq \sqrt{Nln2}$ 이므로 $k \approx 0.83\sqrt{N}$
9. 생일 문제에 $N = 365$를 대입하면: $k \approx 0.83\sqrt{365} \approx 15.9$
10. 따라서 k의 최솟값은 16임

랜덤 오라클 모델

- 생일 문제(Birthday Problems)
  - 문제에 대한 그래프



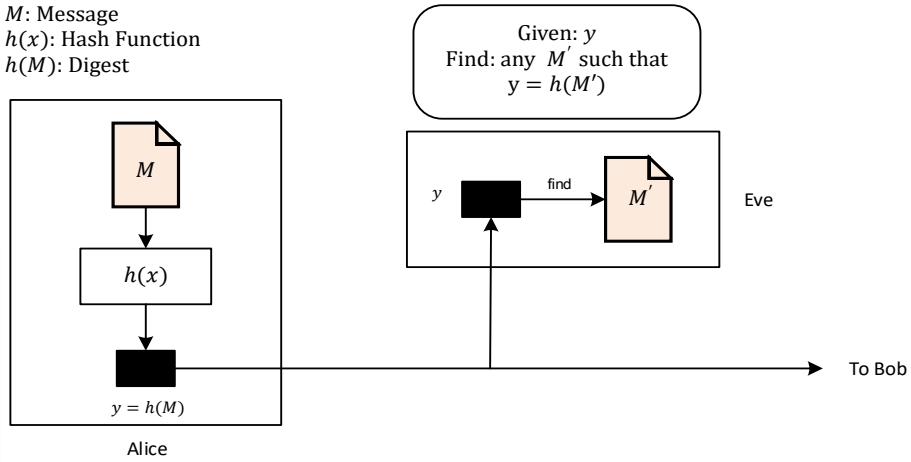
- 가로축은 학생 수 k, 세로축은 같은 생일이 존재할 확률 P를 나타냄
- 문제 1·2는 k가 N에 비례하므로 확률 증가 속도가 비교적 느림
- 문제 3·4는 k가 $\sqrt{N}$에 비례하므로 적은 인원으로도 확률이 빠르게 증가함
- $P = 1/2$일 때 문제 1·2는 약 253~254명, 문제 3은 23명, 문제 4는 16명 필요
- 충돌 공격이 프리이미지 및 제2 프리이미지 공격보다 적은 계산량으로 가능함을 보여주는 그래프

랜덤 오라클 모델

- 랜덤 오라클 모델에 대한 공격
  - 프리이미지 공격 (1/2)
    - 정의
      - 공격자는 다이제스트 $D = h(M)$을 가로채고 $D = h(M')$이 되는 임의의 메시지 M'를 탐색하는 공격
    - 동작 방법

주어진 정보: $y = h(M)$
구하려고 하는 것: $y = h(M')$을 만족하는 M'

Preimage_Attack(D)
{
  for($i = 1$ *to* k)
  {
   create($M[i]$)
   $T \leftarrow h(M[i])$ // T는 임시 다이제스트이다.
   if ($T = D$) return $M[i]$
  }
  return failure
}



1. 임의의 메시지 $M[i]$ 생성
2. 생성한 메시지의 다이제스트 $T = h(M[i])$계산
3. T와 주어진 다이제스트 D비교
4. $T = D$이면 해당 메시지 $M[i]$ 반환
5. 최대 k번 반복 후 일치값이 없으면 공격 실패

랜덤 오라클 모델

- 랜덤 오라클 모델에 대한 공격
  - 프리이미지 공격 (2/2)
    - 성공 확률 및 난이도
      - k개의 메시지를 생성했을 때 성공 확률: $P \approx 1 - e^{-k/2^n}$
      - 성공 확률이 50% 이상이 되기 위한 메시지 수: $k \approx 0.69 \times 2^n$
      - 프리이미지 공격의 난이도는 약 2^n에 비례함
    - 예제 11.11

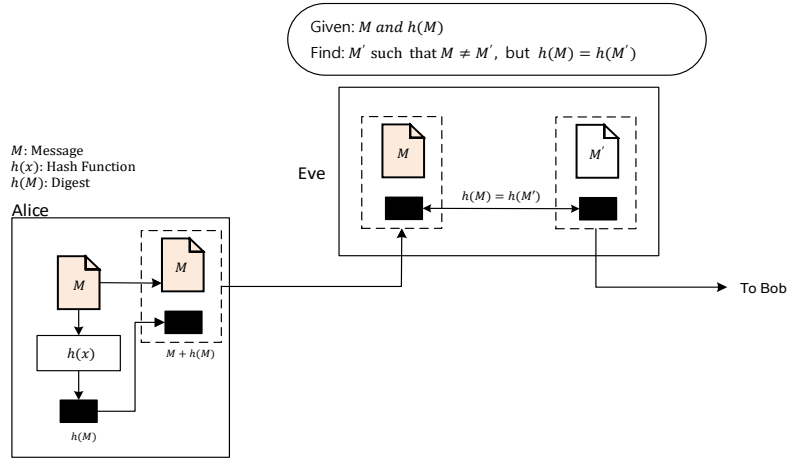
암호학적 해시함수는 64비트 다이제스트를 사용한다. 공격자는 원래의 메시지를 찾을 확률이 0.5 이상이 되도록 하려면 얼마나 많은 다이제스트를 생성해야 하는가?

    - 풀이
      - 생성해야 할 다이제스트의 수는$k \approx 0.69 \times 2^n \approx 0.69 \times 2^{64}$
      - 공격자가 초당 2^{30}개의 메시지를 생성한다고 하더라도 0.69×2^{34}초가 필요함
      - 따라서 64비트 메시지 다이제스트는 프리이미지 공격에 대해 안전함

랜덤 오라클 모델

- 랜덤 오라클 모델에 대한 공격
  - 제2 프리이미지 공격 (1/2)
    - 정의
      - 공격자가 하나의 다이제스트 $D = h(M)$과 이에 대응하는 메시지 M을 가로챈 상황에서 $h(M') = D$를 만족하는 다른 메시지 M'을 탐색하는 공격
    - 동작 방법

주어진 정보: M과 $h(M)$
구하려고 하는 것: $h(M) = h(M')$을 만족하는 $M'(\neq M)$



Second_Preimage_Attack $(\boldsymbol{D}, \boldsymbol{M})$
{
  for $(i = 1\ to\ k\ -\ 1)$
{
  create $(M[i])$
  $T \leftarrow h(M[i])$ // T는 임시 다이제스트이다.
  if $(T = D)$ return $M[i]$
}
  return failure
}

1. 주어진 메시지 M을 제외한 새로운 메시지 $M[i]$생성
2. 생성한 메시지의 임시 다이제스트 $T = h(M[i])$계산
3. 임시 다이제스트 T와 주어진 다이제스트 D비교
4. $T = D$이면 동일한 다이제스트를 갖는 메시지 $M[i]$반환
5. $k - 1$번 반복 후 일치하는 메시지가 없으면 공격 실패

랜덤 오라클 모델

- 랜덤 오라클 모델에 대한 공격
  - 제2 프리이미지 공격 (2/2)
    - 성공 확률 및 난이도
      - $k-1$개의 메시지를 생성했을 때 성공 확률: $P \approx 1 - e^{-(k-1)/N}$
      - 성공 확률이 50% 이상이 되기 위한 메시지 수: $k \approx 0.69N + 1$ 혹은 $k \approx 0.69 \times 2^n + 1$
      - 제2 프리이미지 공격의 난이도는 약 2^n에 비례함

랜덤 오라클 모델

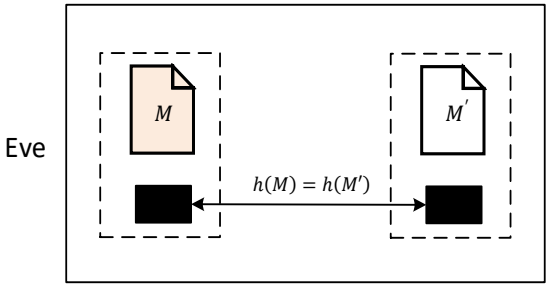
- 랜덤 오라클 모델에 대한 공격
  - 충돌 공격 (1/2)
    - 정의
      - 동일한 다이제스트를 갖는 서로 다른 두 메시지 M과 M'를 탐색하는 공격
    - 동작 방법

주어진 정보: 해시 함수 h
구하려고 하는 것: $h(M) = h(M')$을 만족하는 쌍 (M, M')

Collision_Attack
{
  for $(i = 1\ to\ k)$
  {
  create $(M[i])$
  $D[i] \leftarrow h(M[i])$ // $D[i]$는 생성된 다이제스트 목록임
  for $(j = 1\ to\ i - 1)$
  {
  if $(D[i] = D[j])$ return $(M[i]$ and $M[j])$
  }
  }
  return failure
}

M: Message
$h(x)$: Hash Function
$h(M)$: Digest

Find: $M\ and\ M'$ such that $M \neq M'$, but $h(M) = h(M')$



1. 새로운 메시지 $M[i]$ 생성
2. 메시지의 다이제스트 $D[i] = h(M[i])$계산
3. 새로 생성한 $D[i]$를 이전에 생성한 다이제스트 $D[j]$와 차례로 비교
4. $D[i] = D[j]$이면 동일한 다이제스트를 갖는 두 메시지 $M[i]$와 $M[j]$반환
5. k번 반복 후 같은 다이제스트가 없으면 공격 실패

랜덤 오라클 모델

- 랜덤 오라클 모델에 대한 공격
  - 충돌 공격 (2/2)
    - 성공 확률 및 난이도
      - k개의 메시지를 생성했을 때 성공 확률: $P \approx 1 - e^{-k(k-1)/2N}$
      - 성공 확률이 50% 이상이 되기 위한 메시지 수: $k \approx 1.18 \times N^{1/2}$ 혹은 $k \approx 1.18 \times 2^{n/2}$
      - 충돌 공격의 난이도는 약 $2^{n/2}$에 비례함
    - 예제 11.12

어느 암호학적 해시함수가 64비트 다이제스트를 사용한다고 하자. 공격자가 동일한 다이제스트를 갖는 서로 다른 두 개의 메시지를 만들 수 있는 확률이 0.5 이상 되려면 얼마나 많은 다이제스트가 필요한가?

- 풀이
  - 생성해야 할 다이제스트의 수는 $k \approx 1.18 \times 2^{n/2} \approx 1.18 \times 2^{32}$임
  - 공격자가 초당 2^{20} 메시지를 확인할 수 있다면 1.18×2^{12}초(2시간 이내)로 끝낼 수 있음
  - 따라서 길이가 64비트인 메시지 다이제스트는 충돌공격에 안전하지 못함

랜덤 오라클 모델

- 랜덤 오라클 모델에 대한 공격
  - 또 다른 충돌 공격 (1/2)
    - 정의
      - 정상 메시지 후보 집합 $M_1, M_2, \dots, M_k$와 위조 메시지 후보 집합 $M'_1, M'_2, \dots, M'_k$를 생성하고, 두 집합 사이에서 동일한 다이제스트를 갖는 메시지 쌍을 탐색하는 공격
    - 동작 방법

주어진 정보: 해시 함수 h, 원본 메시지 집합, 위조 메시지 집합
구하려고 하는 것: 두 집합 사이에서 $h(M[i]) = h(M'[j])$를 만족하는 메시지 쌍 $(M[i], M'[j])$

1. 원본 메시지 M으로부터 변형된 정상 메시지 집합 $M[1], M[2], \dots, M[k]$생성
2. 위조 메시지 M'으로부터 변형된 위조 메시지 집합 $M'[1], M'[2], \dots, M'[k]$생성
3. 정상 메시지 집합의 각 다이제스트 계산 $D[i] = h(M[i])$
4. 위조 메시지 집합의 각 다이제스트 계산 $D'[j] = h(M'[j])$
5. 정상 메시지의 다이제스트 $D[i]$와 위조 메시지의 다이제스트 $D'[j]$를 서로 비교 만약 $D[i] = D'[j]$이면 동일한 다이제스트를 갖는 메시지 쌍 $(M[i], M'[j])$ 반환
6. 모든 조합을 비교해도 동일한 다이제스트가 없으면 공격 실패

Alternate_Collision_Attack $(\boldsymbol{M[k], M'[k]})$
{
  for $(i = 1\ to\ k)$
  {
    $D[i] \leftarrow h(M[i])$
    $D'[i] \leftarrow h(M'[i])$
    if $(D[i] = D'[j])$ return $(M[i], M'[j])$
  }
  return failure
}

랜덤 오라클 모델

- 랜덤 오라클 모델에 대한 공격
  - 또 다른 충돌 공격 (2/2)
    - 성공 확률 및 난이도
      - 두 집단에 각각 k개의 메시지가 있을 때 성공 확률: $P \approx 1 - e^{-k^2/N}$
      - 성공 확률이 50% 이상이 되기 위한 메시지 수: $k \approx 0.83 \times N^{1/2}$ 혹은 $k \approx 0.83 \times 2^{n/2}$
      - 또 다른 충돌 공격의 난이도는 약 $2^{n/2}$에 비례함

랜덤 오라클 모델

- 랜덤 오라클 모델에 대한 공격
  - 공격 유형별 난이도

Attack	Value of k with $P = 1/2$	Order
Preimage	$k \approx 0.69 \times 2^n$	2^n
Second preimage	$k \approx 0.69 \times 2^n + 1$	2^n
Collision	$k \approx 1.18 \times 2^{n/2}$	$2^{n/2}$
Alternate collision	$k \approx 0.83 \times 2^{n/2}$	$2^{n/2}$

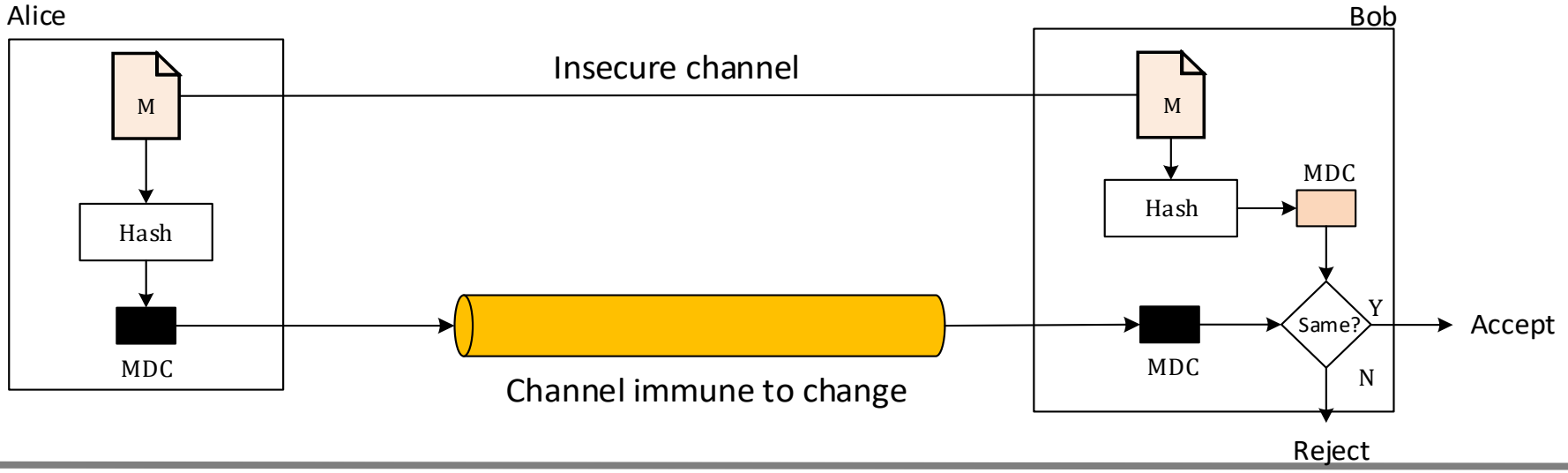
  - 프리이미지·제2 프리이미지 공격의 난이도는 2^n수준이고 충돌 공격의 난이도는 $2^{n/2}$ 수준임
  - 따라서 같은 n비트 해시 함수에서 충돌 저항성의 실제 보안 수준은 프리이미지 저항성보다 낮으며, 충돌 공격이 상대적으로 보안 수준이 낮음

목 차

- 보충
- 메시지 무결성(Message Integrity)
- 랜덤 오라클 모델(Random Oracle Model)
- 메시지 인증(Message Authentication)

메시지 인증

- 변경 감지 코드(MDC, Modification Detection Code)
  - 정의
    - 메시지에 암호학적 해시 함수를 적용해 생성하며, 수신 측에서 다시 계산한 값과 비교하여 메시지의 변경 여부를 확인하는 다이제스트
  - 활용
    - 파일·프로그램·다운로드 데이터의 해시값을 비교하여 손상이나 변경 여부를 확인할 때 사용함



메시지 인증

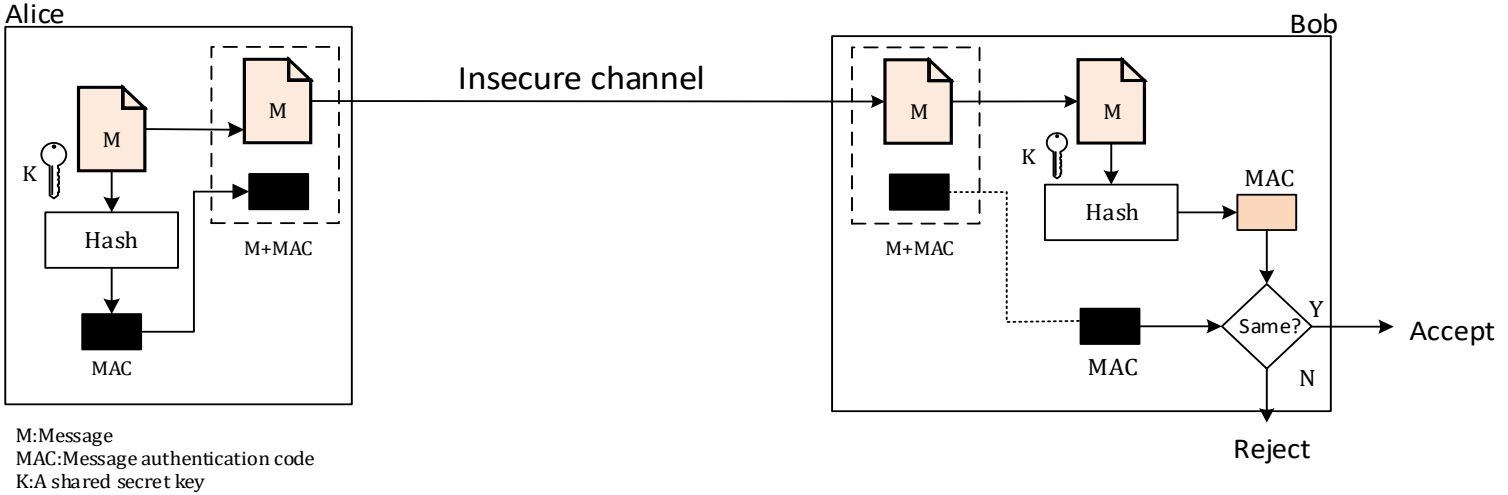
- 변경 감지 코드(MDC, Modification Detection Code)
  - 동작 방법
    1. 송신자가 메시지에 암호학적 해시 함수를 적용하여 MDC를 생성함
    2. 송신자가 메시지와 MDC를 함께 수신자에게 전송함
    3. 수신자가 받은 메시지로부터 새로운 MDC를 생성함
    4. 새로 생성한 MDC와 수신한 MDC를 비교함
    5. 두 값이 같으면 메시지가 변경되지 않은 것으로 판단, 다르면 변경 또는 손상된 것으로 판단함

메시지 인증

- 변경 감지 코드(MDC, Modification Detection Code)
  - 보안성
    - n비트 다이제스트 사용 시 프리이미지 공격에 약 2^n, 충돌 공격에 약 $2^{n/2}$번의 시도가 필요한 구조임
    - 충분히 긴 다이제스트와 안전한 해시 함수 사용 시 우연한 충돌과 원문 복원이 어려운 방식임
  - 문제점
    - 비밀키가 없어 공격자가 변조된 메시지의 MDC를 다시 계산할 수 있는 문제가 존재함

메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - 정의
    - 메시지와 공유 비밀키를 이용해 생성하며, 메시지의 무결성과 송신자 인증을 확인하는 값
  - 활용
    - 비밀키를 공유하는 송수신자가 네트워크 메시지의 변조 여부와 송신자를 인증할 때 사용함



메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - 동작 방법
    - 송신자
      1. 메시지 M과 공유 비밀키 K를 MAC 알고리즘에 입력하여 MAC 생성
      2. 생성된 MAC을 원래 메시지 M 뒤에 첨부$(M + MAC)$
      3. 메시지와 MAC을 안전하지 않은 채널로 전송
    - 수신자
      1. 수신한 $M + MAC$에서 메시지와 MAC을 분리
      2. 수신 메시지 M과 동일한 비밀키 K로 새로운 MAC 계산
      3. 직접 계산한 MAC과 수신한 MAC 비교하고 두 값이 같으면 메시지 수락, 다르면 거부

메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)

*K: 공유 비밀키
*M: 메시지
*H: 해시 함수
*T: 생성된 메시지 인증 코드(MAC)

  - 키 적용 방식
    - 프리픽스 MAC(Prefix-MAC)
      - 비밀키를 메시지의 앞부분에 붙인 뒤 해시 함수를 적용하는 방식

$$T = H(K \,||\, M)$$

      - 일반 해시 함수에 단순히 키를 붙이는 방식은 길이 확장 공격 등의 취약점이 발생할 수 있어 실제로는 HMAC 사용 권장함
    - 포스트픽스 MAC(Postfix-MAC)
      - 비밀키를 메시지의 끝부분에 붙인 뒤 해시 함수를 적용하는 방식

$$T = H(M \,||\, K)$$

      - 키를 메시지 뒤에 붙이므로 길이 확장 공격은 줄일 수 있지만, 단순히 $M \,||\, K$를 해시하는 구조이기 때문에 해시 함수가 충돌에 취약하면 서로 다른 메시지가 같은 MAC을 가질 수 있음

메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - 보안 위협 (1/2)
    - 전수조사 공격
      - 키의 길이가 짧을 경우 가능한 모든 키를 대입하여 가로챈 MAC과 동일한 값을 생성하는 비밀키 탐색
      - 비밀키가 노출되면 변조한 메시지에 대한 올바른 MAC을 생성하여 메시지 위조 가능
    - 프리이미지 공격
      - 해시 함수의 프리이미지 저항성이 약할 경우 가로챈 MAC과 동일한 해시값을 생성하는 입력값 탐색
      - 해당 입력값에서 비밀키를 알아내거나 유효한 MAC을 생성하여 메시지 위조 가능

메시지 인증

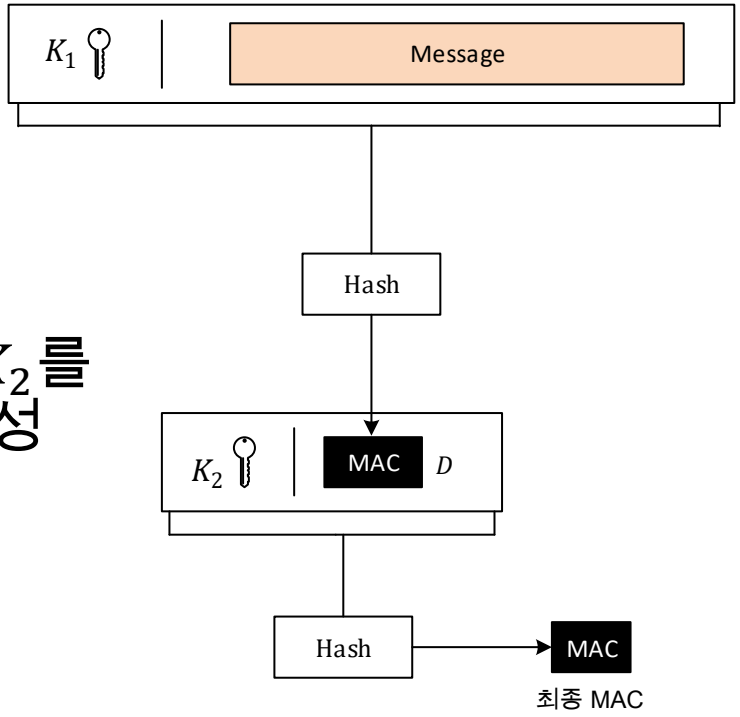
- 메시지 인증 코드(MAC, Message Authentication Code)
  - 보안 위협 (2/2)
    - 알려진 메시지 공격
      - 공격자가 여러 개의 메시지와 이에 대응하는 MAC 쌍을 수집하여 키 또는 MAC 생성 규칙 분석
      - 분석 결과를 이용해 새로운 메시지에 대응하는 유효한 MAC을 생성하여 메시지 위조 가능

메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - 보안성
    - k비트 비밀키 사용 시 키 전수조사 공격에 약 2^k번의 시도가 필요한 구조임
    - 비밀키를 모르는 공격자가 메시지와 올바른 MAC을 함께 위조하기 어려움
  - 문제점
    - 비밀키가 유출되면 공격자가 정상적인 MAC 생성 가능함
    - 송신자와 수신자가 같은 키를 공유하므로 부인 방지 제공 불가함

메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - 중첩 MAC(Nested MAC) (1/2)
    - 개요
      - MAC의 안전성을 높이기 위해 설계됨
    - 동작 방법
      - 메시지 M과 첫 번째 비밀키 K_1을 이용해 내부 다이제스트 생성
      - 내부 다이제스트 D와 두 번째 비밀키 K_2를 이용해 다시 해시하여 최종 MAC을 생성
    - 개선된 점
      - 해시 연산을 두 번 중첩하여 내부 다이제스트의 직접 노출 방지
      - 서로 다른 두 키를 사용하여 단순 프리픽스·포스트픽스 MAC보다 위조 난이도 향상



메시지 인증

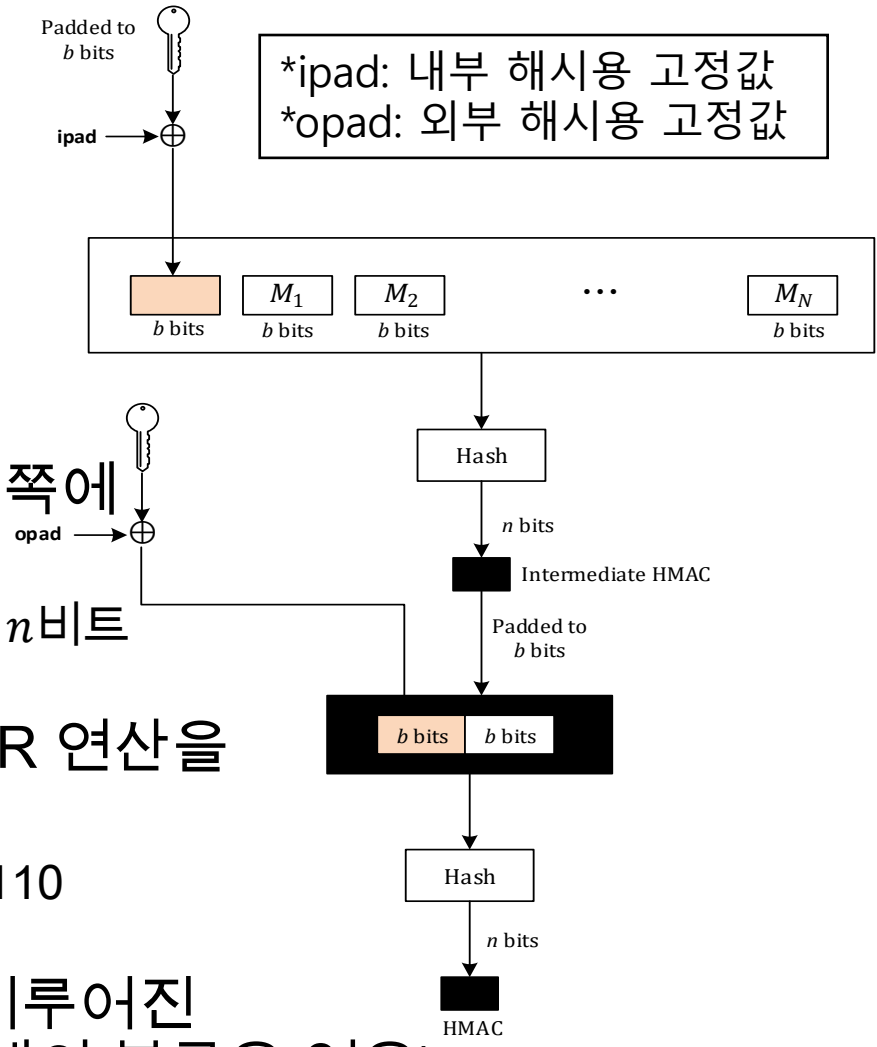
- 메시지 인증 코드(MAC, Message Authentication Code)
  - 중첩 MAC(Nested MAC) (2/2)
    - 활용
      - 중첩 MAC은 HMAC의 설계 기반으로 활용되며, 실제로는 HMAC-SHA-256 등의 형태로 API 요청 인증, IPsec·VPN 패킷 인증에 사용됨
    - 보안성
      - 서로 다른 두 비밀키와 이중 해시를 사용하여 내부 다이제스트의 직접 노출 방지
      - 단순히 키와 메시지를 한 번 해시하는 방식보다 길이 확장 공격과 MAC 위조에 대한 저항성 향상
    - 문제점
      - 해시 연산을 두 번 수행하므로 일반 MAC보다 연산량 증가
      - K_1, K_2두 키를 안전하게 관리해야 하는 부담이 있어 현대에는 주로 HMAC 구조가 활용됨

메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - HMAC(Keyed-Hash Message Authentication Code)
    - 정의
      - 암호학적 해시 함수와 공유 비밀키를 결합하여 메시지의 무결성과 송신자 인증을 제공하는 메시지 인증 코드 방식
    - 특징
      - 내부 패딩값(ipad)와 외부 패딩값(opad)를 이용한 내부·외부의 이중 해시 구조로 중간 다이제스트의 직접 노출을 방지함
      - NIST의 FIPS 198-1에 정의된 표준 방식이며, 현재 명세를 SP 800-224로 이전하는 절차가 진행 중임
    - 활용
      - IPsec·DNS 인증·JWT 등 인터넷과 소프트웨어 기반 통신의 메시지 인증에 사용함

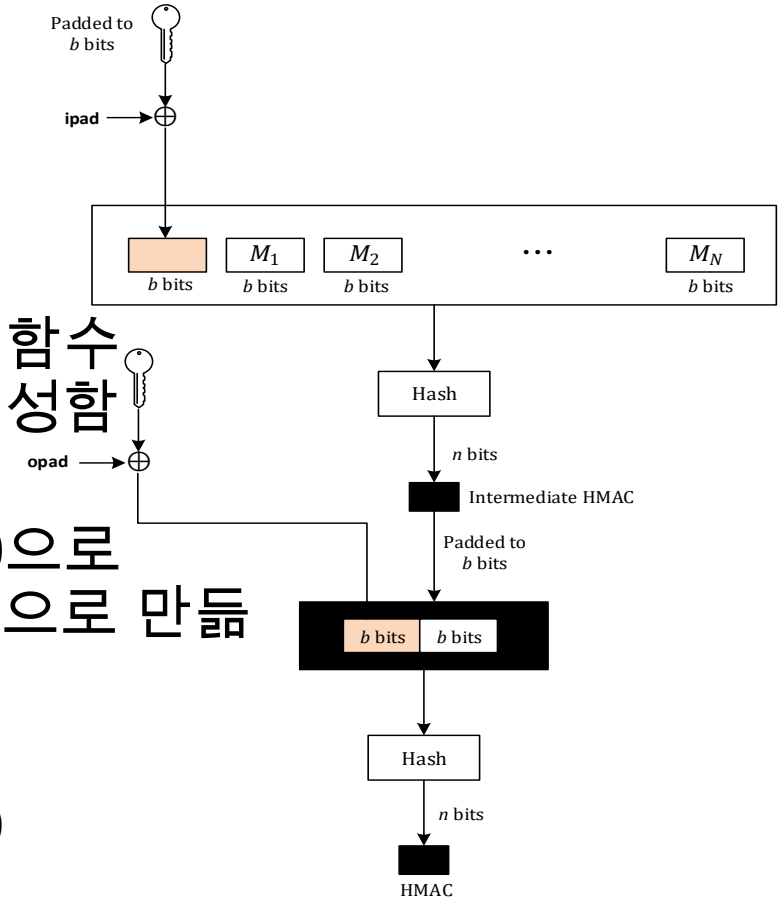
메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - HMAC
    - 동작 방법 (1/2)
      1. 메시지를 길이가 b비트인 N개의 블록으로 나눔
      2. 비밀키가 b비트보다 짧으면 오른쪽에 0비트를 추가하여 b비트로 맞춤
         - 패딩을 하기 이전의 비밀 키의 길이는 n비트 이상을 권장
      3. 단계 2의 결과를 ipad 상수와 XOR 연산을 수행함
         - ipad는 $b/8$번의 연속된 2진 열 00110110 (16진수 36)으로 이루어짐
      4. 단계 3에서 얻은 블록을 N개로 이루어진 블록 메시지 맨 앞에 붙임($N+1$개의 블록을 얻음)



메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - HMAC
    - 동작 방법 (2/2)
      5. 단계 4에서 얻은 $N+1$ 결과를 해시 함수에 적용하여 n비트 다이제스트를 생성함
         - 이 다이제스트를 중간 HMAC라고 부름
      6. n비트로 된 중간 HMAC의 왼쪽에 0으로 이루어진 열을 패딩하여 b비트 블록으로 만듦
      7. 단계 2와 단계 3을 다른 상수인 opad(output pad)로 반복함
         - opad는 $b/8$번의 연속된 2진 열 01011100 (16진수 5C)으로 이루어짐
      8. 단계 7에서 얻은 결과를 단계 6에서 얻은 블록의 앞에 붙임
      9. 단계 8의 결과에 같은 해시 함수를 적용하여 최종 n비트 HMAC를 생성함



메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)

*태그: HMAC 계산이 끝난 뒤 나오는 최종 인증값

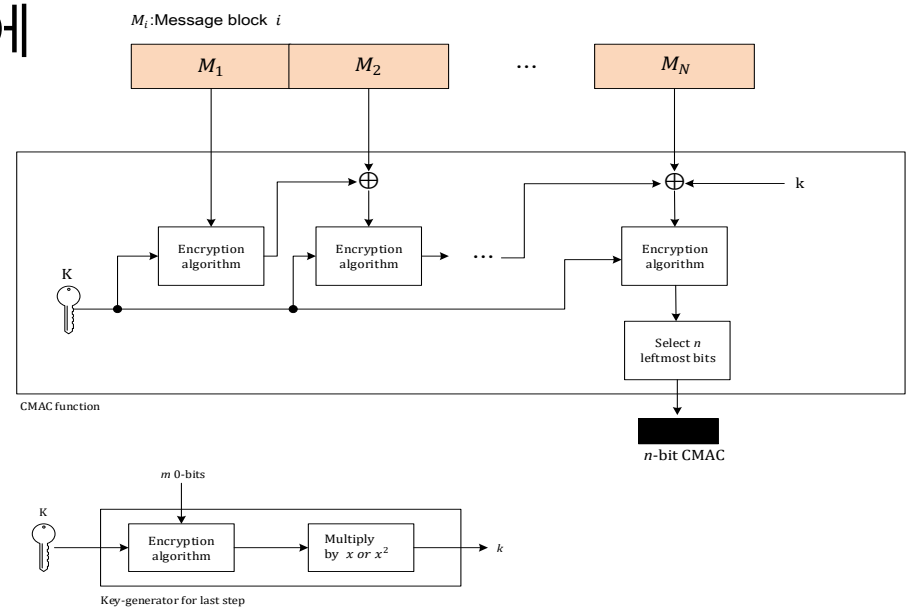
  - HMAC
    - 보안성
      - 충분히 긴 비밀키 사용 시 키 전수조사 공격에 높은 저항성 제공
      - 내부·외부 해시 구조를 사용하여 단순 키 결합 방식과 길이 확장 공격을 방지
      - 최종 n비트 HMAC을 사용할 경우, 공격자가 태그를 한 번 무작위로 맞힐 확률은 약 2^{-n}
    - 문제점
      - 비밀키가 유출되면 공격자가 정상적인 HMAC을 생성할 수 있는 문제
      - 112비트 미만의 비밀키 사용 시 키 전수조사 공격에 취약해질 수 있음

메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - CMAC(Cipher-based Message Authentication Code)
    - 정의
      - 블록 암호와 공유 비밀키를 이용하여 메시지의 무결성과 송신자 인증을 제공하는 메시지 인증 코드 방식
    - 특징
      - 하나의 비밀키에서 두 개의 보조키 K_1, K_2를 생성하여 사용함
      - 해시 함수 없이 블록 암호 연산만으로 인증 태그를 생성하는 구조임
      - FIPS 113의 기존 블록 암호 기반 MAC을 개선하여 NIST SP 800-38B에서 CMAC으로 표준화됨
    - 활용
      - AES 기반 장치와 블루투스 같은 임베디드·무선 통신의 인증 및 키 생성에 사용함

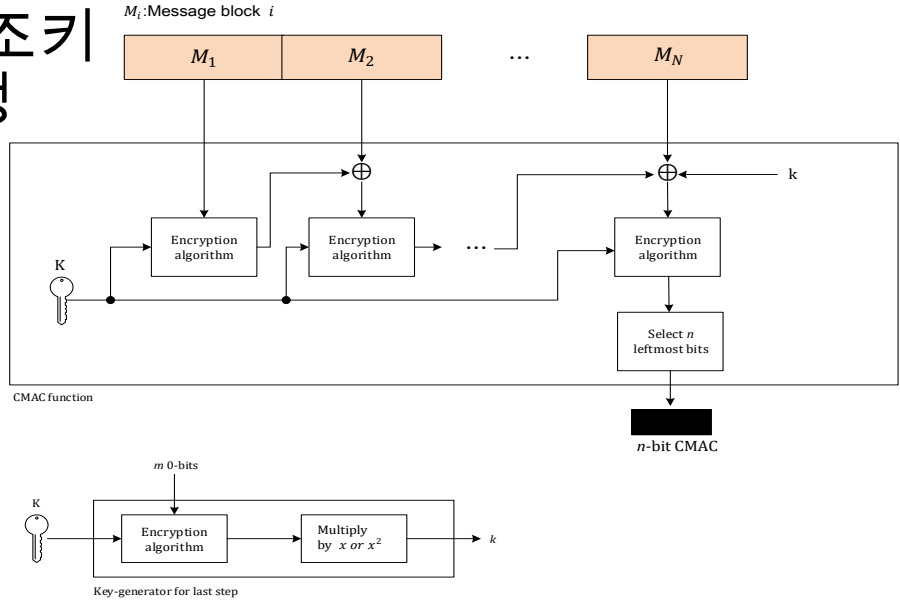
메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - CMAC(Cipher-based Message Authentication Code)
    - 동작 방법 (1/2)
      1. 메시지를 길이가 m비트인 N개의 블록 $M_1, M_2, \ldots, M_N$으로 분할
      2. 마지막 블록이 m비트보다 짧으면 마지막 메시지 비트 뒤에 1비트를 추가하고, 남은 부분을 0으로 채워 블록 길이에 맞춤
      3. m비트의 0 블록을 비밀키 K로 암호화하여 보조키 생성에 사용할 값 생성
      4. 패딩이 없는 경우 해당 값에 x, 패딩이 있는 경우 x^2를 곱하여 보조키 k생성



메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - CMAC(Cipher-based Message Authentication Code)
    - 동작 방법 (2/2)
      5. 첫 번째 메시지 블록 M_1을 비밀키 K로 암호화
      6. 이전 암호화 결과와 다음 메시지 블록을 XOR한 후 다시 암호화하는 과정 반복
      7. 마지막 블록 처리 시 생성한 보조키 k를 적용한 후 최종 암호화 수행
      8. 최종 암호화 결과의 왼쪽 n비트를 선택하여 CMAC 생성



메시지 인증

- 메시지 인증 코드(MAC, Message Authentication Code)
  - CMAC(Cipher-based Message Authentication Code)
    - 보안성
      - AES와 같은 안전한 블록 암호 및 충분한 키 길이 사용 시 키 전수조사 공격에 높은 저항성 제공
      - t비트 태그 사용 시 무작위 위조 성공 확률이 약 2^{-t}인 구조
    - 문제점
      - 비밀키 유출 시 공격자가 정상적인 CMAC을 생성할 수 있는 문제
      - 태그 길이가 짧으면 무작위 위조 성공 가능성 증가
      - 동일한 키로 지나치게 많은 메시지를 처리할 경우 충돌 가능성이 증가하는 한계

Thanks!

김 민 식(minsik@pel.sejong.ac.kr)