정보보호개론

제3장 암호알고리즘 개요 2부

1. 해시함수

임의 길이의 이진 문자열을 고정된 길이의 이진 문자열(해시값, 메시지 다이제스트, 메시지 지문)로 매핑하여 주는 결정적 함수를 해시함수라 한다. 동일한 값을 해시하면 그 결과는 항상 같다. 해시함수 H는 기본적으로 다음 3가지 요구사항을 만족해야 한다.

- 요구사항 1. 압축: 해시값의 길이보다 큰 길이의 데이터도 해시값의 크기로 줄일 수 있어야 한다.
- 요구사항 2. 계산의 용이성: x가 주어지면 $H(x)$를 계산하기 쉬워야 한다.
- 요구사항 3. 일방향성(one-wayness): 입력을 모르는 해시값 y가 주어졌을 때, $H(x') = y$를 만족하는 x'를 찾는 것은 계산적으로 어려워야 한다.

위 3가지 요구사항을 만족하는 해시함수를 **일방향 해시함수**(OWHF, One-Way Hash Function)라 하며, 요구사항 3을 다른 말로 원상 회피성(preimage resistance)이라 한다. 해시함수는 항상 압축하는 것은 아니다. 해시값의 크기보다 작은 값도 해시할 수 있으며, 해시값의 길이는 입력과 상관없이 항상 같다. 일방향성 요구사항을 좀 더 면밀히 살펴보면 y 값을 최초로 계산할 때 입력으로 사용한 x를 포함하여 y로 매핑되는 임의의 값을 찾을 수 없어야 한다는 것을 말한다. 이 때문에 계산의 용이성 요구사항과 달리 x'로 표기한 것이다. 해시값의 길이가 n 비트일 때 2^n개의 서로 다른 입력을 검사하여야 x'을 찾을 수 있으면 이 해시함수는 일방향성에 대해 안전한 해시함수이다. 따라서 x가 n비트인 특정 y 값으로 해시되기 위한 확률은 $1/2^n$이어야 한다.

해시값의 크기는 고정되어 있고, 해시함수는 임의 길이의 값을 입력으로 받을 수 있기 때문에 동일한 값으로 매핑되는 정의역 값은 무수히 많을 수 있다. 이처럼 서로 다른 두 개의 값 x와 x'의 해시값 $H(x)$와 $H(x')$이 같으면 충돌(collision)이 발생하였다고 말한다. 응용에 따라 충돌을 찾기 쉬우면 곤란한 경우가 많다. 그러므로 이와 같은 응용에서는 다음 2가지 요구사항을 추가로 만족해야 한다. 다음 두 가지 조건을 모두 만족하는 해시함수를 **충돌회피 해시함수**(CRHF, Collision-Resistant Hash Function)라 한다.

- 요구사항 4. 약한 충돌회피성(weak collision-resistance): x가 주어졌을 때 $H(x') = H(x)$인 $x'(\neq x)$을 찾는 것은 계산적으로 어려워야 한다.
- 요구사항 5. 강한 충돌회피성(strong collision-resistance): $H(x') = H(x)$인 서로 다른 임의의 두 입력 x와 x'을 찾는 것은 계산적으로 어려워야 한다.

생일로 비유하면 홍길동이 주어졌을 때 홍길동과 생일이 같은 사람을 찾는 것이 어려워야 한다는 것이 약한 충돌회피이고, 생일이 같은 임의의 두 사람을 찾는 것이 어려워야 한다는 것이 강한 충돌회피이다. 약한 충돌회피성은 다른 말로 제2원상 회피성(second-preimage resistance)이라 한다. 사람의 생일이 균일하게 분포되어 있다고

가정하였을 때 23명이 있으면 이 중에 생일이 같은 쌍이 있을 확률이 50%가 넘는다. 생각보다 이 숫자가 작기 때문에 이를 생일 파라독스(paradox)라 한다. 이 파라독스에 때문에 해시값의 길이가 n비트일 때 $2^{n/2}$개만 모으면 충돌을 50%의 확률로 찾을 수 있다. 더 정확하게는 생일 파라독스에 의하면 해시값의 길이가 n이고, q개의 해시값을 계산하였을 때 충돌을 찾을 확률은 대략 $\frac{q^2}{2^{n+1}}$이다.

1.1 해시함수의 용도

해시함수는 전자서명과 함께 많이 사용한다. 길이가 긴 데이터에 대해 직접 전자서명을 하면 계산 비용이 너무 높을 수 있고, 전자서명 값이 매우 길어질 수 있어서 데이터 대신에 데이터의 해시값에 전자서명을 하게 된다. 이때 사용하는 해시함수는 반드시 충돌회피 해시함수이어야 한다. Sig.$A(H(M))$이 메시지 M에 대한 A의 전자서명값이라고 할 때, A가 $H(M) = H(M')$인 M'을 찾을 수 있다면 A는 나중에 M에 서명한 것이 아니라 M'에 서명한 것이라고 주장할 수 있다. B가 충돌을 찾으면 A의 서명키를 모르는 상태에서 A의 서명을 위조한 것이 된다.

해시함수는 무결성 서비스를 위해 사용할 수 있다. 파일시스템에 파일 F와 그 파일에 대한 해시값 $H(F)$을 함께 저장할 수 있다. 바이러스 등에 의해 F가 F'로 변경되었다면 해시값 $H(F)$ 값을 통해 그 사실을 발견할 수 있다. 하지만 지능적인 악성 소프트웨어는 F를 F'으로 변경하면서 해시값도 새롭게 계산해 함께 저장할 수 있다. 이 경우에는 파일의 변경을 해시값을 이용하여 발견할 수 없다. 따라서 해시함수도 암호키를 사용하도록 설계할 수 있다. 암호키를 사용하는 해시함수를 메시지 인증 코드(**MAC**, Message Authentication Code)라 하고, 이와 구분하기 위해 일반 해시함수를 조작 탐지 코드(MDC, Manipulation Detection Code)라 한다. 하지만 MDC라는 용어는 거의 사용하지 않는다. 파일시스템 예에서 $H(F)$ 대신에 $MAC.K(F)$를 유지하면 키 K를 모르는 경우 파일과 MAC 값을 함께 수정할 수 없다. MAC에서 사용하는 암호키는 비밀키로써 생성한 자와 확인하는 자가 같은 키를 가지고 있어야 한다,

리눅스와 같은 다중사용자 운영체제나 다중 사용자가 접속할 수 있는 웹 서비스는 패스워드를 이용하여 사용자를 인증하는 것이 일반적이다. 이때 패스워드를 평문 상태로 유지하면 해당 정보가 저장된 파일이나 데이터베이스가 노출되면 모든 사용자의 패스워드가 노출되는 문제점을 가지고 있다. 따라서 보통 패스워드 대신에 패스워드의 해시값을 저장한다. 해시함수의 일방향성 때문에 패스워드가 저장된 파일이 노출되더라도 패스워드를 알아내기 힘들어진다. 이와 관련된 내용은 11장에서 다시 자세히 설명한다.

2. MAC

해시함수의 용도를 설명하면서 MAC을 이미 소개하였다. MAC은 해시함수와 유사하게 주어진 메시지를 일정한 크기의 출력으로 매핑하여 주는 결정적 함수이지만 해시함수와 달리 메시지와 키를 받아 결괏값을 계산한다. 따라서 키를 모르면 MAC 값을 생성할 수 없고, 주어진 MAC 값을 확인할 수 없다. 이 때문에 MAC 값은 송신자의 인증과 무결성을 동시에 제공하는 함수이지만 해당 키를 가지고 있는 모든 사용자는 해당 MAC 값을 생성할 수 있기 때문에 부인방지 기능을 제공하기는 어렵다.

MAC은 보통 해시함수보다 무결성을 더 안전하게 제공하기 위해 사용한다. 하지만 MAC을 이용하지 않고 전자서명을 통해 무결성을 제공할 수 있다. MAC은 계산비용이 저렴하지만 같은 키를 가지고 있는 사용자만 확인할 수 있으며, 이 때문에 부인방지에 약하다. 이와 달리 전자서명을 이용하면 계산비용이 높지만 대응되는 확인키를 가지고 있는 누구나 확인할 수 있으며, 서명키는 한 사용자만 비밀로 유지하는 것이기 때문에 부인방지 서비스가 가능하다. 참고로 해시값을 대칭키로 암호화하여도 유사한 효과를 얻을 수 있다.

MAC은 새로운 전용 알고리즘을 개발하여 만들 수 있지만 보통 대칭 암호알고리즘이나 해시함수를 이용하여 만든다. 해시함수를 이용하여 만든 MAC을 HMAC이라 한다.

2.1 비밀성과 무결성을 동시 제공

보통 암호알고리즘은 비밀성만 제공하고 무결성은 제공하지 않는다. 하지만 무결성을 제공하지 못하면 복호화한 값이 원래 평문인지 확신할 수 없기 때문에 조작이 될 가능성이 있다. 따라서 지금은 메시지를 암호화할 때 비밀성과 무결성을 동시에 제공할 수 있는 방법을 사용하며, 이와 같이 메시지를 암호화하는 것을 **인증 암호화**(authenticated encryption)라 한다[1].

인증 암호화를 제공하는 방법은 크게 다음 3가지 방법이 있다.

- 방법 1. 암호화할 때 메시지의 해시값 또는 MAC 값을 포함하는 방법

$$E.K_1(M||\text{MAC}.K_2(M))$$

- 방법 2. 메시지에 독립적으로 대칭 암호알고리즘과 MAC을 적용하는 방법

$$E.K_1(M), MAC.K_2(M)$$

- 방법 3. 메시지를 암호화한 후에 암호문에 대한 MAC 값을 계산하는 방법

$$C = E.K_1(M), MAC.K_2(C)$$

방법 1에서 해시함수보다 MAC을 사용하는 것이 더 안전하므로 다른 방법과 비교할 때 MAC을 사용하는 것만 고려하여 비교한다. 또 용도가 다르면 다른 키를 사용하는 것이 안전하므로 암호화할 때 사용하는 키와 MAC을 계산할 때 사용하는 키는 다른 키를 사용해야 한다. 이것이 번거로우면 하나의 키로부터 독립적인 두 개의 키를 계산하여 사용할 수 있다. 방법 1과 2는 무조건 복호화를 해야 무결성을 확인할 수 있지만 방법 3은 복호화하지 않고 무결성을 확인한 다음에 무결성이 확인된 경우에만 복호화를 할 수 있기 때문에 방법 3이 다른 두 방법보다 효과적이다. 더구나 방법 2에서 메시지에 대한 MAC 값은 비밀성을 제공하지 않기 때문에 M에 대한 정보가 일부 노출될 수 있는 문제도 있다. 이 때문에 3가지 방법 중 세 번째 방법이 가장 안전하고 효과적인 방법이다. 세 번째 방법을 다른 말로 encrypt-then-mac 방법이라 한다.

3. 전자서명

전자서명은 우리가 손으로 하는 일반적 서명을 정보 서비스로 바꾼 것이다. 전자서명의 특성을 살펴보기 전에 일반 서명과의 차이점을 먼저 생각하여 보자. 전자서명은 정보화된 서비스이므로 서명의 확인은 전자적으로 프로그램된 알고리즘을 통해 검증하게 된다. 반면에 일반 서명은 보통 눈으로 기존 서명과 대조하여 이루어진다. 더욱이 전자서명은 모든 사용자에게 동일한 안전성을 제공하지만, 일반서명은 서명에 따라 위조 가능성이 다르다. 일반서명은 서명하고자 하는 문서 위에 하지만 전자서명은 문서와 별도로 존재한다. 하지만 한 문서의 전자서명은 다른 문서의 전자서명으로 사용할 수 없어야 하므로 전자서명은 문서를 이용하여 서명 값을 생성한다. 따라서 전자서명은 동일한 사용자라 하더라도 문서마다 다른 값이 생성되지만 같은 사용자의 일반서명은 모든 문서마다 동일하다. 끝으로 전자서명은 디지털 값이므로 원본과 복사본 개념이 없다.

전자서명처럼 기존에 컴퓨팅 요소를 사용하지 않고 제공하던 서비스를 정보화하면 서비스가 원래 만족해야 하는 요구사항뿐만 아니라 정보화 때문에 새롭게 생기는 요구사항까지 만족해야 한다. 원래 일반 서명이 가지고 있는 요구사항은 다음과 같다.

- 요구사항 1. 인증(authentic): 누가 서명하였는지 확인이 가능해야 한다.

- 요구사항 2. 위조불가(unforgeable): 위조가 불가능해야 한다.
- 요구사항 3. 부인방지(non-repudiation): 나중에 부인할 수 없어야 한다.

전자서명은 요구사항 1과 3을 충족하기 위해 서명자마다 다른 서명키를 사용한다. 따라서 같은 문서를 다른 사용자가 서명하면 다른 서명 값이 생성된다.

다음 요구사항들은 일반 서명에서는 필요가 없었지만, 정보화되면서 추가된 요구사항이다.

- 요구사항 4. 재사용불가(not reusable): 서명을 다른 용도로 사용할 수 없어야 한다.
- 요구사항 5. 변경불가(unalterable): 서명된 문서의 내용을 변경할 수 없어야 한다.

전자서명은 서명한 문서와 별도로 존재하며 디지털 정보이기 때문에 한 문서의 전자서명을 다른 문서의 전자서명으로 사용하는 것과 전자서명을 그대로 유지한 상태에서 문서를 바꾸는 것이 가능하지 않아야 한다. 두 요구사항은 모두 서명을 생성할 때 문서를 이용하여 생성하면 만족할 수 있다. 따라서 전자서명은 사용자의 서명키와 서명할 문서를 입력으로 받아 서명 값을 생성하여야 한다. 어떤 문서와 그것의 전자서명 전체를 한 번만 사용할 수 있어야 하는 경우도 있다. 이때에는 서명할 때 서명 시간을 포함하는 방법을 많이 사용한다.

3.1 전자서명 방식

전자서명은 보통 사용자가 자신의 서명키와 서명할 문서를 가지고 혼자 전자서명 알고리즘을 수행하여 서명 값을 생성한다. 이렇게 하는 방식을 직접 서명 방식이라 한다. 이 경우 서명키가 도난 또는 분실되었을 때 공격자에 의해 또는 서명자가 직접 악의적으로 서명한 시각을 조작할 가능성이 있다. 이를 방지하기 위해 중재 서명 방식을 사용할 수 있다. 중재 서명 방식은 더는 알고리즘이 아니며, 프로토콜 형태로 다음과 같이 진행된다.

$$\begin{aligned} &\text{Msg 1.} \quad A \to S: \quad T_A, M, \text{Sig.}A(T_A||H(M)) \\ &\text{Msg 2.} \quad S \to A: \quad T_S, \text{Sig.}S(T_S||\text{Sig.}A(T_A||H(M))) \end{aligned}$$

사용자 A가 문서 M과 서명 시각 T_A을 함께 전자서명하여 중재자에게 전달하면 중재자가 서명 시각을 포함하여 해당 서명의 유효성을 확인한 후에 사용자의 서명값과 중재자의 현재 시각 T_S을 포함하여 서명하게 된다. 한 메시지에 대해 사용자와 중재자가 같이 서명하는 방식이다. 따라서 이 중재자가 신뢰할 수 있는 기관이고 사용자와 공모하지 않는다면 사용자는 서명 시각을 고의적으로 조작할 수 없다.

3.2 전자서명 알고리즘의 분류

전자서명도 알고리즘이므로 결정적 서명 알고리즘과 확률적 서명 알고리즘으로 분류할 수 있다. 하지만 전자서명은 대칭과 비대칭 암호알고리즘처럼 반드시 확률적 서명 알고리즘이 되어야 안전성이 높아지는 것은 아니다. 메시지의 비밀성이 목적이 아니므로 결정적 알고리즘이더라도 위조가 계산적으로 어려우면 문제가 되지 않는다.

전자서명은 서명키와 서명할 데이터를 입력으로 받아 서명 값을 생성하게 되며, 공개키 기반이므로 데이터가 매우 크면 서명 비용이나 서명 크기 때문에 데이터를 직접 입력 받아 서명 값을 생성하기 어렵다. 따라서 보통 전자서명은 서명키와 서명할 데이터의 해시값을 입력으로 받아 서명 값을 생성하게 된다. 이와 같은 전자서명 알고리즘을 첨부 형태 전자서명(DSS with appendix)이라 한다. 이 방식에서는 전자서명 값만 가지고는 서명을 확인할 수 없고, 서명된 원 데이터와 확인키가 있어야 서명을 확인할 수 있다.

반대로 매우 작은 데이터들은 데이터를 직접 입력 받아 서명값을 생성할 수 있으며, 앞서 설명한 방식과 달리

이 경우 서명 값으로부터 서명한 데이터를 얻어 낼 수 있으므로 이와 같은 전자서명을 메시지 복구 가능 전자서명(DSS with recovery)이라 한다. 이 방식의 근본 문제는 매우 작은 크기의 메시지만 사용할 수 있다는 것이다.

4. 암호알고리즘의 안전성

4.1 추상적 안전성

암호알고리즘에 대한 해독이란 알고리즘을 분석하여 그것의 허점을 발견하는 것을 말한다. 이를 통해 암호알고리즘의 안전성을 분석하게 된다. 지금까지 살펴본 5가지 알고리즘에서 우리가 바라는 안전성을 추상적으로 설명하면 다음과 같다.

대칭 암호알고리즘은 비밀키를 모르는 상태에서 암호문이 주어졌을 때 그것의 평문을 알아내거나 비밀키를 찾아낼 수 없어야 한다. 평문 전체 또는 키 전체를 알아내지 못하더라도 그중 일부만 알아낼 수 있으면 알고리즘 안전성에 문제가 있는 것이다. 공개키 암호알고리즘은 공개키는 알지만 개인키를 모르는 상태에서 암호문이 주어졌을 때 그것의 평문을 알아내거나 개인키를 찾아낼 수 없어야 한다. 해시함수는 해시값의 역을 취할 수 있거나 충돌을 찾아낼 수 없어야 한다. 실제 정의역이 너무 광범위하고 많이 압축하기 때문에 역을 찾는 것은 별로 고려하지 않는다.

MAC은 MAC 키를 모르는 상태에서 MAC 값을 위조할 수 없거나 키를 알아낼 수 없어야 한다. 또 해시함수와 마찬가지로 충돌을 찾을 수 없어야 한다. 충돌을 찾을 수 있다는 것은 보통 위조할 수 있다는 것을 의미하기 때문에 위조할 수 없어야 한다는 것에 충돌을 찾을 수 없어야 한다는 것이 포함되어 있다고 생각해도 된다. 전자서명도 확인키를 모르는 상태에서 서명을 위조하거나 서명키를 알아낼 수 없어야 한다. MAC이나 전자서명의 경우에는 공격자가 원하는 값에 대한 위조를 얻지 못하더라도 위조가 가능하면 큰 문제가 될 수 있다. 참고로 이와 같은 안전성은 해독을 통해 깨질 수 있지만, 사회공학이나 사용자 부주의로 깨질 수 있으며, 이 경우에는 해독되었다고 하지 않고 노출(compromise)되었다고 한다.

4.2 현대 암호학의 특징

과거에는 안전성에 대한 형식적인 증명 없이 암호기술을 설계하여 발표하면 암호학자들과 공격자들이 허점을 발견하고자 노력하게 되며, 허점이 발견되면 그 허점을 보완하는 형태로 암호기술이 개발 및 발달하였다. 이렇게 기술이 개발되는 것을 "design-break-patch"라 한다. 이 때문에 오랫동안 허점이 발견되지 않으면(survived long scrutiny) 안전성이 증명된 것으로 간주하였다.

하지만 오늘날에는 엄격한 안전성 증명을 중요하게 생각한다. 이것은 암호기술과 관련된 학문적 이론이 발전했으며, 안전성을 증명하는 기술도 발달하였기 때문이다. 안전성 증명은 보통 증명하고자 하는 것에 대한 정의를 내리고, 안전성 모델(공격자 모델)을 정립하고, 필요한 가정을 나열한 후 이와 같은 능력을 가진 공격자가 존재하더라도 주어진 가정 하에서는 정의된 안전성이 보장됨을 형식적으로 증명하게 된다. 그렇지만 안전성에 대한 형식적 증명(formal proof)이 이루어졌다고 하여 안전성에 대해 완벽한 보장(bullet-proof, iron-clad guarantee)이 제공되는 것은 아니다. 정의가 잘못된 것일 수 있고, 가정이 잘못될 수 있거나 가정이 현실과 조금 거리가 있을 수 있다. 그럼에도 불구하고 안전성 증명은 매우 중요하다. 지금은 형식적 증명이 없는 암호기술은 실제 현장에서 사용을 하지 않는다. 다만, 실제 현장에서 사용하는 기술은 그 응용에서 요구하는 효율성도 갖추어야 한다. 따라서 현장에서 사용하는 것보다 안전한 알고리즘도 있지만 이들은 실용성 때문에 사용하지 않고 있을 수 있다.

현대 알고리즘의 안전성은 보통 현재의 컴퓨팅 능력을 고려한 안전성이다. 따라서 컴퓨팅 능력이 획기적으로 좋아지거나 지금까지 알려진 가장 빠른 해독 방법보다 더 빠른 방법을 발견하면 지금의 안전성 수준은 아무런 의미가 없어질 수 있다. 특히, 양자 컴퓨팅(quantum computing)이 현실화되면 지금까지 사용한 여러 가정들이 무의미해질 수 있다. 이 때문에 최근에는 양자 컴퓨팅이 실용화되어도 안전한 암호기술(post-quantum cryptography)을

개발하고 있다. 이에 대해서는 4.7 절에서 좀 더 구체적으로 설명한다.

4.3 안전성 관련 정의

암호알고리즘의 안전성은 해독하기 위해 요구하는 노력에 의해 측정이 된다. 하지만 우리가 사용하는 알고리즘은 (unconditionally secure)을 제공하지 못한다. 무조건적 안전성이란 무한한 컴퓨터 자원을 가져도 암호알고리즘을 해독할 수 없는 경우를 말한다. 특히, 제한된 크기의 키를 사용하면 키에 대한 전수조사를 통해 사용한 키를 발견하는 것이 불가능한 것은 아니기 때문에 우리가 보통 사용하는 알고리즘은 **계산적 안전성**(computationally secure)을 가진다. 계산적 안전성이란 공격자의 능력이 가장 현실적으로 높다고 가정[1]하였을 때 공격자가 암호알고리즘을 해독하기 위한 노력이 불합리하게 많은 컴퓨터 시간을 요구할 경우를 말한다. 예를 들어 대칭 암호알고리즘이 128 비트 키를 사용하고, 전수조사 외에는 다른 해독 방법이 없다면 이 알고리즘은 2^{128}개의 키를 검사해야 복호화를 할 수 있기 때문에 계산적으로 안전한 알고리즘이다.

대칭 암호알고리즘과 달리 공개키 암호알고리즘은 어려운 수학 문제를 기반하고 있다. 알고리즘이 그것이 기반하고 있는 수학 문제와 등가임을 증명할 수 있다면 이 알고리즘은 증명 가능 안전성(provably secure)을 가지고 있다고 한다. 예를 들어 RSA 공개키 암호알고리즘이 인수분해 문제와 등가라는 것이 증명되어 있다면 RSA는 증명 가능 안전성을 가지고 있다고 할 수 있다. 하지만 실제 RSA을 해독하는 문제와 인수분해 문제가 등가라는 것이 증명되어 있지는 않지만 많은 학자들이 같을 것으로 예측하고 있다. 어려운 문제라는 것은 지금까지 알려진 해결 방법이 지수 시간 알고리즘밖에 없는 경우를 말하기 때문에 증명 가능 안전성도 계산적 안전성에 해당한다[2].

Shannon은 1949년에 암호알고리즘이 **완벽한 안전성**(perfect security)을 가지기 위한 조건을 제시하였고, 이와 같은 안전성을 가지기 위해서는 키 길이가 메시지의 길이보다 커야 한다는 것을 증명하였다[3]. Shannon이 제시한 완벽한 안전성 조건을 확률을 이용하여 수학적으로 표현하면 다음과 같다.

$$\Pr(\mathcal{M} = m|\mathcal{C} = c) = \Pr(\mathcal{M} = m)$$

즉, 암호문 c가 주어졌을 때 그것의 평문이 m일 확률은 원래 평문이 m일 확률과 같아야 한다는 것을 말한다. 원래 평문이 m이 될 확률은 응용과 사용하는 방법에 따라 달라질 수 있다. 실제 위 확률식이 성립하면 다음 확률식도 성립한다.

$$\Pr(\mathcal{C} = c|\mathcal{M} = m) = \Pr(\mathcal{C} = c)$$

즉, 어떤 평문 m이 주어졌을 때 이것이 특정 암호문 c로 암호화되기 위한 확률은 암호문이 c가 될 확률과 같아야 한다는 것을 말한다.

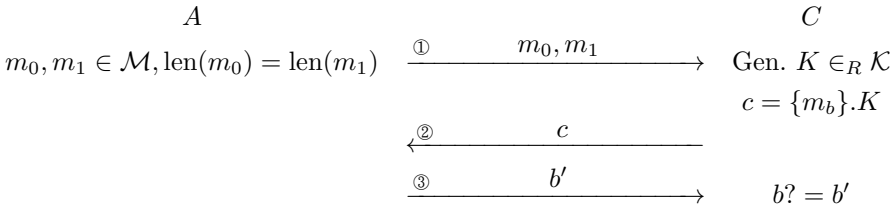
Shannon의 증명에 의해 완벽한 안전성을 갖춘 알고리즘은 키 길이 때문에 현실적으로 사용하기 힘들다. 예를 들어 항상 평문과 같은 길이의 랜덤 키를 생성하여 평문과 XOR하여 암호화하는 one-time pad 암호알고리즘은 완벽한 안전성을 갖추고 있는 알고리즘이다. 하지만 항상 다른 키를 사용해야 하며, 키 길이가 메시지와 항상 같아야 하기 때문에 실제 응용에서 사용할 수 있는 알고리즘은 아니다. 이 때문에 one-time pad 대신에 이와 유사하게 암호화하는 스트림 방식(키 길이는 고정되어 있지만 이 키를 이용하여 메시지와 같은 길이의 랜덤한 키 스트림을 생성하여 암호화함)의 암호알고리즘을 사용하고 있다.

무한한 컴퓨터 자원을 가져도 해독할 수 없으면 좋겠지만 계산적 안전성을 가져도 실용적으로 사용하는데 문제가 없다. 따라서 완벽한 안전성 대신에 **의미론적 안전성**(semantic security)라는 개념을 사용하여 현대 암호알고리즘을 분석한다. 의미론적으로 안전하다는 것은 암호문이 주어졌을 때 효율적으로 계산할 수 있는 모든 것은 암호문이 없어도 계산할 수 있어야 한다는 것을 말한다. 여기서 효율적으로 계산할 수 있다는 것은 계산적 안전성을 고려한다는 것을 의미한다. 하지만 이 자체를 형식화된 형태로 증명하기 어렵기 때문에 이보다 많이 사용하는 개념이 **구별 불가**

[1] 현존하는 가장 고성능 하드웨어를 이용하여 해독 전용 시스템을 구축할 때 소요되는 경제적 비용과 이를 통해 해독하는데 소요되는 시간을 함께 고려한다.

안전성(indistinguishability)이다.

구별 불가 안전성이란 도전자와 공격자 간에 다음과 같은 게임으로 생각할 수 있다.



이 게임은 도전자 입장에서 $b = 0$인 게임과 $b = 1$인 게임으로 나누어진다. 도전자는 정해진 알고리즘을 이용하여 암호키를 임의로 생성한 후에 공격자에게 게임 시작을 알리면 공격자는 서로 다르지만 길이가 같은 평문 두 개를 생성하여 도전자에게 전달한다. 도전자는 게임의 종류에 따라 m_0 또는 m_1을 암호화하여 공격자에게 전달하면 공격자는 이 암호문이 어떤 평문을 암호화한 것인지 맞추는 게임이다. 공격자의 능력에 따라 이 과정을 1번 하는 것이 아니라 q번 반복을 할 수 있다. 전체 반복이 끝난 후 공격자는 $b = 0$ 게임에 참여하였는지 $b = 1$ 게임에 참여하였는지 맞추는 게임이 된다.

답이 둘 중 하나이므로 공격자가 답을 맞출 확률은 최소한 50%이다. 따라서 모든 효율적인 공격자에 대해 맞출 확률이 정확하게 50%임을 증명할 수 있다면 이 알고리즘은 의미론적으로 안전한 알고리즘이다. 하지만 맞출 확률이 50%보다 넘다 하더라도 그 초과된 부분이 무시할 수 있을 정도로 작다면 우리는 이 알고리즘은 의미론적으로 안전한 알고리즘이라 한다.

안전성을 논할 때 한 가지 또 알고 있어야 하는 개념이 **NM 특성**(Non-Malleability)이다. NM 특성을 만족하는 알고리즘에 의해 생성된 암호문은 그것의 평문을 알고 있는지 여부와 상관없이 의미 있는 다른 평문으로 복호화되도록 이 암호문을 다른 암호문으로 변경할 수 없어야 한다. 예를 들어 100원을 인출하라는 평문을 암호화한 암호문을 변경하여 1,000원을 인출하라고 바꿀 수 있다면 이 암호알고리즘은 NM 특성을 만족하지 못한다. One-time pad는 완벽한 안전성을 갖춘 알고리즘이지만 키 길이가 비현실적일 뿐만 아니라 XOR을 사용하기 때문에 기본적으로 NM 특성을 만족하지 못한다. XOR의 특성 때문에 암호문의 특정 비트를 토글한 후 복호화하면 평문의 해당 위치의 비트가 토글된다.

NM 특성은 앞서 언급한 의미론적 안전성보다 강한 특성이며, 무결성이 비밀성과 함께 제공되어야 만족할 수 있는 특성이다. 안전한 암호알고리즘과 안전한 MAC 함수를 이용하여 인증 암호화를 하면 NM 특성을 만족하는 암호문을 만들 수 있다. 즉, 수정을 할 수 있어도 수정한 것을 알 수 있으면 NM 특성을 갖춘 알고리즘이라 한다.

4.4 암호알고리즘에 대한 공격

모든 암호알고리즘은 기본적으로 전사공격(brute force)에 대해서는 안전하여야 한다. 전사공격이란 가능한 모든 경우를 검사하여 공격하는 것을 말하며, 보통 가능한 모든 키를 검사하는 방식을 말한다. 예를 들어 n비트 암호키를 사용하는 대칭 암호알고리즘은 2^n 노력이면 키를 무조건 찾을 수 있다. 하지만 전사공격은 암호해독에 해당하지 않는다. 현재 컴퓨팅 수준을 고려하면 키 길이가 128비트 이상이면 안전하다고 말한다. 해시함수는 생일 파라독스 때문에 충돌회피가 해시값의 길이에 제곱근에 비례하므로 해시값의 길이는 256비트 이상이 되어야 한다. 대칭 암호알고리즘의 경우 전사공격이 가장 효과적인 공격이면 가장 안전한 알고리즘이 된다.

암호알고리즘 안전성을 증명할 때 공격자의 능력을 어떻게 가정하였는지에 따라 안전성의 강도가 달라진다. 가장 능력이 적은 공격자부터 가장 능력이 많은 공격자까지 분류하면 다음과 같다.

- 암호문 단독 공격(ciphertext-only attack): 공격자가 암호문만 가지고 있는 경우를 말한다. 가지고 있는 암호

문의 개수에 따라 능력의 차이가 있다.

- 기지 평문 공격(known-plaintext attack): 공격자는 특정한 개수의 평문과 암호문 쌍을 얻게 된다. 하지만 공격자는 평문이나 암호문을 선택할 능력은 없다.
- 선택 평문 공격(chosen-plaintext attack): 공격자는 특정한 개수의 평문과 암호문 쌍을 얻게 되는데, 공격자는 자신이 원하는 평문을 선택할 수 있다.
- 선택 암호문 공격(chosen-ciphertext attack): 공격자는 특정한 개수의 평문과 암호문 쌍을 얻게 되는데, 공격자는 자신이 원하는 암호문을 선택할 수 있다. 선택 암호문 공격은 보통 선택 평문 공격을 포함한다고 한다.
- 적응적(adaptive) 선택 평문 공격: 적응적이지 못한 경우에는 정해진 개수만큼 한번에 요구하여 쌍을 얻어야 하지만 적응적인 공격에서는 기존에 얻은 쌍들을 바탕으로 자신이 원하는 평문에 대한 암호문을 결정할 수 있다.
- 적응적 선택 암호문 공격: 기존에 얻은 쌍들을 바탕으로 자신이 원하는 암호문을 선택하여 평문을 얻을 수 있는 경우를 말한다.

앞서 설명한 게임은 선택 평문 공격을 설명하는 게임이다. q번 반복하는 게임에서는 공격자는 자신이 선택한 q개의 평문에 대한 암호문을 얻을 수 있다. 이 게임에서 실제 공격자는 각 게임을 반복할 때 다른 두 개의 평문을 전달하는 것이 아니라 $m_0 = m_1$인 같은 두 개의 평문을 전달할 수 있다. 선택 암호문 공격은 보통 공개키 암호알고리즘의 안전성을 분석할 때 사용하며, 비슷한 게임으로 표현하면 다음과 같다.

A		C
	$\xleftarrow{①} +K$	Gen. $(+K, -K)$
$c_i \in \mathcal{C}$	$\xrightarrow{②} c_i$	$m_i = D(-K, c_i)$
	$\xleftarrow{③} m$	
$m_0, m_1 \in \mathcal{M}, \text{len}(m_0) = \text{len}(m_1)$	$\xrightarrow{④} m_0, m_1$	$c = E(+K, m_b)$
	$\xleftarrow{⑤} c$	
$c_i \in \mathcal{C}, c_i \neq c$	$\xrightarrow{⑥} c_i$	$m_i = D(-K, c_i)$
	$\xleftarrow{⑦} m_i$	
	$\xrightarrow{⑧} b'$	$b? = b'$

이 게임에서 알 수 있듯이 공격자는 공개키를 가지고 있기 때문에 얼마든지 자신이 원하는 메시지를 암호화할 수 있으며, 알고리즘이 결정적 알고리즘이면 ⑤의 c를 직접 계산하여 비교할 수 있기 때문에 선택 암호문 공격에 대해 안전할 수 없다. 참고로 암호 기술에서 선택 평문이나 선택 암호문 공격을 할 때 자신이 원하는 평문에 대한 암호문이나 암호문에 대한 평문을 얻게 해주는 것을 오라클(oracle)이라 한다.

이와 같은 공격의 복잡성 정도는 보통 공격자가 공격에 성공하기 위해 필요한 평문 암호문 쌍의 개수에 의해 정의된다. 이것을 데이터 복잡도라 한다. 데이터 복잡도 외에 복잡성을 측정하기 위해 공격에 성공하기 위해 필요한 시간을 말하는 시간 복잡도나 공격 알고리즘을 수행할 때 필요로 하는 공간의 크기를 말하는 공간 복잡도를 고려한다. 이 중 시간 복잡도는 컴퓨팅 능력에 의해 결정되는 요소이다.

4.5 암호알고리즘에 대한 공격 결과

키를 사용하는 암호알고리즘에 대한 공격 결과는 다음과 같이 표현된다.

- 완전 성공(total break): 비밀키를 발견한 경우를 말한다. 공개키 방식에서는 개인키를 찾은 경우를 말한다.

- 광역 성공(global deduction): 암호키를 발견하지 못하였지만 복호화할 수 있는 알고리즘을 발견한 경우를 말한다. 전자서명과 MAC은 서명이나 MAC 값을 무조건적으로 위조가 가능한 경우를 말한다.
- 인스턴스 성공(instance deduction): 어떤 한 암호문으로부터 또는 어떤 특정 종류의 암호문들로부터 그것의 평문을 얻어낸 경우를 말한다. 전자서명과 MAC은 목표하는 사용자의 서명이나 MAC 값을 제한적으로 위조할 수 있는 경우를 말하며, 다음과 같이 추가적으로 분류한다.
  - 선택적 위조(selective forgery): 특정 종류의 메시지에 대해서만 위조할 수 있는 경우를 말한다.
  - 존재 위조(existential forgery): 어떤 메시지에 대한 서명이나 MAC 값을 위조하였지만 공격자가 위조한 메시지를 결정할 수 있는 능력이 없는 경우를 말한다.
- 정보 추출(information deduction): 암호문으로부터 평문의 일부나 암호키와 관련된 정보를 얻어낸 경우를 말한다. 전자서명이나 MAC은 개인키, MAC키 외에 위조할 서명이나 MAC 값의 일부를 얻어낸 경우를 말한다.

해시함수의 경우 해시값의 길이가 n 비트이면 역방향이나 약한 충돌회피는 2^n의 노력이 필요하지만 충돌을 찾는 것은 생일 파라독스 때문에 $2^{n/2}$가 필요하다. 따라서 해시함수의 안전성은 보통 충돌을 찾는 것에 집중하게 된다.

4.6 안전성에 대한 기타 고려사항

암호알고리즘 자체가 안전하다고 증명되어 있더라도 그것을 구현한 소프트웨어나 하드웨어가 잘못 구현되어 허점이 있을 수 있다. 특히, 알고리즘의 입력과 출력 외에 다른 정보를 이용하여 공격할 수 있으며, 이를 **부채널 공격**(side channel attack)이라 한다. 대표적인 부채널 공격에는 시간 정보를 이용하는 것과 전력 소비를 이용하는 것이 있다. 시간 정보는 입력이 주어졌을 때 출력을 계산하는데 소요되는 시간을 이용하여 키를 찾는 공격을 말하며[4], 전력 소비의 경우에는 입력이 주어졌을 때 출력을 계산하기 위해 소요된 전력의 차이를 이용하여 키를 찾는 공격을 말한다[5]. 최근에는 심화 학습(deep learning) 기술의 발전으로 부채널을 통해 얻은 정보를 더 효과적으로 분석할 수 있게 되었다. 이것을 방어하기 위해 부채널로 노출되는 정보를 얻을 수 없도록 하거나 그것이 어려우면 부채널로 노출되는 정보를 줄이는 방법을 사용한다. 이와 같은 부채널 공격 때문에 절대로 비전문가가 직접 암호알고리즘을 소프트웨어나 하드웨어로 구현하여 사용해서는 안 된다.

4.7 양자 컴퓨팅과 암호 기술

아직 이 현실화되어 있지는 않지만 양자 컴퓨팅의 특성과 동작 원리는 잘 알려져 있다. 특히, 가능한 입력 범위가 n일 때, 특정 출력을 주는 입력을 양자 컴퓨팅을 이용하면 $O(\sqrt{n})$ 비용으로 찾을 수 있다는 것이 증명되었다. 이 알고리즘을 Glover 알고리즘이라 한다[6]. 따라서 대칭키가 128비트이면 기존 컴퓨팅 환경에서 전수조사 비용이 $O(2^{128})$이지만 양자 컴퓨팅이 현실화되면 비용이 $O(2^{64})$로 줄어들게 된다. Peter Shor는 비슷한 시기에 양자 컴퓨팅을 이용하면 다차 시간에 인수분해와 이산대수 문제를 해결할 수 있음을 보였다[7].

하지만 이 두 개의 발견이 양자 컴퓨팅을 이용하면 NP-hard 문제를 다차 시간에 풀 수 있다는 것은 아니다. 인수분해 문제를 해결하는 다차 시간 알고리즘이 아직 발견되지 않았지만 이 문제가 NP-complete 문제라고 증명되어 있지 않다. 인수분해가 NP-complete 문제이면 양자 컴퓨팅을 이용하면 모든 NP-hard 문제를 다차 시간에 해결할 수 있다는 것을 의미한다. 이 때문에 지금의 지배적인 생각은 양자 컴퓨팅을 이용하더라도 NP-hard 문제는 여전히 계산적으로 해결하기 어려운 문제라고 여겨지고 있다.

Glover 알고리즘 때문에 대칭 암호알고리즘이나 해시함수는 지금보다 키 길이나 해시값의 길이를 두 배로 늘리면 여전히 양자 컴퓨팅이 현실화되어도 사용할 수 있다. 하지만 공개키 암호알고리즘은 Shor 알고리즘 때문에 지금 사용하는 인수분해, 이산대수, 타원곡선에 기반한 알고리즘을 더 이상 사용할 수 없다. 이 때문에 **양자 내성**

암호(post-quantum cryptography, quantum-resistant cryptography) 기술이란 양자 컴퓨팅 시대에 사용할 수 있는 공개키 암호알고리즘을 말한다.

양자 내성 암호에 대한 연구는 크게 두 가지 형태로 진행되고 있다. 첫째, NP-Hard 문제에 기반한 공개키 암호알고리즘(lattice-based, multivariate-based)에 대한 연구이다. NP-Hard 문제 기반 공개키 암호알고리즘은 이미 오래전에 발견된 기술이지만 지금 사용하고 있는 알고리즘과 비교하였을 때 효율성이 떨어지기 때문에 사용하지 않고 있는 기술이다.

둘째, 해시함수 기반 공개키 기술을 연구하고 있다. 앞서 언급한 바와 같이 해시함수는 해시값의 길이만 늘리면 양자 컴퓨팅 환경에서도 안전하게 사용할 수 있다. 해시함수 기반 공개키 기술도 오래전에 발견된 기술이며, NP-Hard 문제 기반 공개키 기술과 마찬가지로 지금 널리 사용하고 있는 공개키 알고리즘에 비해 효율성이나 사용 편리성이 떨어진다. 해시함수 기반 전자서명에 대해서는 11장에서 조금 더 자세히 살펴본다.

NIST는 2017년부터 현재양자 내성 암호 표준을 진행하고 있으며[8], 현재 3라운드가 진행되고 있다. 공개키 암호알고리즘에는 McEliece(code based), Crystals-Kyber(lattice-based), NTRU(lattice-based), SABER(lattice-based), 4종류의 알고리즘이 3라운드까지 진출하였고, 전자서명 알고리즘에는 Crystals-DILITHIUM(lattice-based), FALCON(lattice-based), Rainbow(multivariate-based), 3종류의 알고리즘이 3라운드까지 진출하고 있다. 최종 후보들을 보면 대부분 격자 기반(lattice-based) 문제에 의존하는 알고리즘이다.

부록

A.1 양자 보안

은 양자 이론이나 기술을 이용한 보안 기술을 말하며, 양자 내성 암호는 양자 보안 기술은 아니다.양자 내성 암호는 양자 컴퓨팅 시대에도 사용할 수 있는 공개키 또는 전자서명 암호알고리즘을 말한다. 현재 개발되고 있는 양자 보안은 크게 양자 난수 생성기(QRNG, Quantum Random Number Generator), 양자 키 분배(QKD, Quantum Key Distributor), 양자 직접통신(QSDC, Quantum Secure Direct Communication)로 분류된다. 최근에는 양자 보안 기술과 기존 보안 기술을 결합하여 사용하는 양자 하이브리드 기술에 대한 연구도 많이 진행하고 있다. QRNG, QKD, QSDC, 양자 하이브리드 기술은 모두 양자 컴퓨터가 필요한 기술은 아니다.

양자 난수 생성기는 양자 기술을 이용하여 난수를 생성하는 방법이며, 이를 위해 특수 칩을 제작하여 사용한다. 암호기술에서 보통 사용하는 암호적으로 안전한 의사난수 생성기와 달리 실제 난수를 생성한다. 하지만 양자 난수 생성기를 사용하나 암호학적으로 안전한 의사난수 생성기를 사용하나 보안 측면에서 차이는 없다. 암호학적으로 안전한 의사난수 생성기를 통해 생성한 난수를 예측하여 공격하는 것은 계산적으로 가능하지 않기 때문이다.

양자 키 분배는 양자 이론을 이용해 빛의 가장 작은 단위인 광자에 암호키와 관련한 정보를 실어 전달함으로써, 도청이 불가능하도록 설계된 키 분배 알고리즘을 말한다. QRNG을 이용하여 대칭키를 생성하였을 경우 보통 기존 키 확립 프로토콜을 사용하여 원격에 있는 두 사용자가 키를 확립하지만 QKD는 대칭키를 양자 채널을 통해 확립한다. 따라서 일반적인 통신 매체가 아니라 양자 정보를 전달할 수 있는 광섬유와 같은 특수 매체가 필요하며, 기존 통신과 달리 도청을 감지할 수 있다. 대칭키를 확립한 이후에는 AES 등을 이용하여 암호 통신을 한다.

양자 직접통신은 양자 기술을 이용한 통신 매체를 말하며, 이 매체를 사용하면 별도 대칭키를 교환하여 암호알고리즘으로 메시지를 암호화하여 교환할 필요가 없다. 매체 자체가 도청이 가능하지 않다. QKD는 키만 양자 채널로 교환하는 반면에 QSDC는 메시지를 양자 채널로 교환한다.

참고문헌

[1] Jonathan Katz, Moti Yung, "Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation," Int'l Workshop on Fast Software Encryption, FSE 2000, LNCS 1978, pp. 284–299, Springer, 2001.

[2] D. Aggarawal, U. Maurer, "Breaking RSA Generically is Equivalent to Factoring," Advances in Cryptology, Eurocrypt 2009, LNCS 5479, pp. 36–53, Springer, 2009.

[3] Claude E. Shannon, "Communication Theory of Secrecy Systems," Bell System Technical Journal, Vol. 28, No. 4, pp. 656–715, Sep. 1949.

[4] Paul C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Advances in Cryptology, CRYPTO 1996, LNCS 1109, pp. 104–113, Springer, 1996.

[5] Paul C. Kocher, Joshua Jaffe, Benjamin Jun, "Differential Power Analysis," Advances in Cryptology, CRYPTO 1999, LNCS 1666, pp. 388–397, Springer, 1997.

[6] Lov K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," Proc. of the 28th Annual ACM Symp. on the Theory of Computing, pp. 212–219, Jul. 1996.

[7] Peter W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," Proc. of the Annual Symp. on Foundations of Computer Science, pp. 124–134, Nov. 1994.

[8] `https://csrc.nist.gov/projects/post-quantum-cryptography`

퀴즈

1. 해시함수와 관련된 다음 설명 중 틀린 것은?

   ① 약한 충돌을 찾는 것이 강한 충돌을 찾는 것보다 쉽다.
   ② 정의역이 치역보다 훨씬 범위가 크기 때문에 충동의 발생은 불가피하다.
   ③ 입력과 무관하게 항상 출력은 고정된 크기이다.
   ④ 약한 충돌은 x가 주어졌을 때 $H(x) = H(x')$인 x와 다른 x'을 찾는 것이다. 생일에 비유하면 홍길동이 주어졌을 때 홍길동과 생일 같은 사람을 찾는 것을 말한다.

2. 해시값의 길이가 $n$bit인 해시 함수를 이용한 임의의 입력 x의 해시값이 특정값이 될 확률은 어떻게 되어야 이 해시함수가 일방향성을 만족할 수 있는가?

   ① $\frac{1}{n}$
   ② $\frac{1}{2^n}$
   ③ $\frac{1}{2}$
   ④ $\frac{1}{x}$

3. 메시지의 무결성을 보장하기 위해 MAC을 사용하는 것과 전자서명을 사용하는 것을 비교한 다음 내용 중 틀린 것은?

   ① MAC을 사용하는 것이 더 효율적이다.
   ② 전자서명을 사용하는 것이 부인방지 측면에서 우수하다.
   ③ 잔지서명은 검증할 수 있는 사용자와 생성할 수 있는 사용자가 비대칭적이다.
   ④ MAC을 사용하는 것이 다수가 검증할 수 있도록 하는 것에 유리하다.

4. 최근에는 메시지를 암호화할 때 안전성을 위해 인증 암호화를 한다. 인증 암호화와 관련된 설명 중 틀린 것은?

   ① 가장 널리 안전하게 사용하는 인증 암호화 기법은 encrypt-then-mac 방법이다.
   ② encrypt-then-mac 방법은 암호문을 복호화한 후에 평문을 이용하여 MAC 값을 확인한다.
   ③ NM 특성이 만족하지 않는 대칭 암호알고리즘의 경우에도 인증 암호화를 사용하면 NM 특성을 만족하게 된다.
   ④ 인증 암호화는 무결성과 비밀성을 동시에 제공하는 암호화 방법이다.

5. 전자서명에서 해시함수를 사용하는 이유가 아닌 것은?

   ① 메시지 자체를 암호화하면 서명값의 크기가 메시지 크기에 비례한다.
   ② 메시지 자체를 암호화하면 서명값을 계산하는 비용이 메시지 크기에 비례한다.
   ③ 사용하는 알고리즘에 따라 메시지 자체를 암호화하면 위조가 쉬워질 수 있다.
   ④ 메시지 자체를 암호화하면 메시지의 비밀성이 보장되지 않는다.

6. 부채널 공격이 주는 교훈이 아닌 것은?

   ① 암호알고리즘을 직접 소프트웨어로 구현하여 사용하거나 하드웨어로 제작하여 사용하는 것은 위험할 수 있다.
   ② 현재 안전하다고 알려져 있더라도 지금까지 생각하지 못한 전혀 다른 형태의 공격이 가능할 수 있다.
   ③ 전문 업체에 의해 구현 및 제작된 소프트웨어 암호라이브러리나 하드웨어 모듈을 활용해야 안전하다.
   ④ 하드웨어보다는 소프트웨어를 이용하여 암호알고리즘을 사용해야 한다.

연습문제

1. 원격에 있는 두 사용자가 동전 던지기를 하기 위해 다음과 같은 프로토콜을 사용할 수 있다.

   - 단계 1. Alice는 매우 큰 양의 정수 X를 임의로 선택하여 그것의 해시값 $H(X)$를 Bob에게 전달한다.
   - 단계 2. Bob은 짝수, 홀수 중 하나를 선택하여 Alice에게 자신의 선택을 알린다.
   - 단계 3. Alice는 X를 Bob에게 전달한다. Bob의 추측이 맞으면 Bob이 이긴 것이 된다.

   이 프로토콜과 관련하여 다음 각각에 대해 답하시오.

   ① 선택 가능한 X의 범위가 이 프로토콜의 안전성에 어떤 영향을 주는지 설명하시오.
   ② Bob이 무엇을 할 수 있으면 항상 이길 수 있는지 제시하고, 해시함수의 어떤 특성 때문에 그것이 가능하지 않은지 제시하시오.
   ③ Alice가 무엇을 할 수 있으면 항상 이길 수 있는지 제시하고, 해시함수의 어떤 특성 때문에 그것이 가능하지 않은지 제시하시오.

   위 프로토콜을 블록방식의 대칭 암호알고리즘을 사용하여 다음과 같이 변경하였다.

   - 단계 1. Alice는 블록 크기의 랜덤한 값 X와 대칭키 K를 임의로 선택하여 X를 K로 채우기 없이 ECB 모드로 암호화한 암호문을 Bob에게 전달한다.
   - 단계 2. Bob은 짝수, 홀수 중 하나를 선택하여 Alice에게 자신의 선택을 알려준다.
   - 단계 3. Alice는 K를 Bob에게 전달한다.

   ④ 이 프로토콜에서 Alice가 무엇을 할 수 있으면 항상 이길 수 있는지 제시하고, 이 측면에서 이 프로토콜의 안정성을 논하시오.

2. 전자서명은 공개키 기술을 이용하는 것이므로 전자서명 알고리즘을 이용하여 매우 큰 메시지를 서명하게 되면 두 가지 문제점이 있다. 두 가지 문제점을 간단히 설명하시오. 이 문제를 극복하기 위해 해시함수를 이용한다. 이때 해시함수는 충돌회피 해시함수이어야 한다. 그 이유도 간단히 설명하시오.

3. 중재 서명방식은 서명자의 서명키가 노출되었을 때 발생하는 문제점을 극복하기 위한 방식이다. 이 문제는 노출되지 않더라도 서명자의 부정행위를 방지할 수도 있다. 15장에서 자세히 소개하는 블록체인 기술은 첨가만 가능한 분산 데이터베이스를 제공하여 준다. 이 기술이 중재 서명방식 대신에 어떻게 사용할 수 있는지 자신의 생각을 제시하시오.

4. 어떤 메시지 M의 무결성을 확인하기 위해 메시지와 그 메시지의 해시값 $H(M)$을 유지할 수 있다. 하지만 능동적인 공격자는 메시지만 변경하지 않고 해시값까지 변경할 수 있다. 이 때문에 보다 안전하게 무결성을 제공하기 위해서는 해시함수 대신에 MAC을 사용할 수 있다. 이를 포함하여 다음 세 가지 형태의 방법을 사용할 수 있다.

   ① MAC만 사용: $MAC.K(M)$
   ② 일반 해시함수와 대칭키 암호알고리즘 사용: $\{H(M)\}.K$

③ 해시함수와 전자서명 알고리즘 사용: Sig.–$K(H(M))$

위 3가지 방법의 차이점을 간단히 논하시오. 특히 1), 2)와 3)의 차이를 논하시오.

5. 대칭 암호알고리즘의 복호화 속도를 일부러 느리게 만들었다. 대칭 암호알고리즘 안전성에 어떤 영향을 주는지 설명하시오.

6. 대칭키 길이가 32비트인 암호알고리즘을 전사공격하는데 하루가 소요된다고 가정하자. 그러면 복호화 속도가 같은 대칭키 길이가 40비트인 암호알고리즘을 전사공격하는데 소요되는 시간을 제시하시오.

7. 전사공격은 가능한 모든 키를 검사하는 것을 말한다. 대표적인 대칭 암호알고리즘인 DES의 키 길이는 56비트이고, AES의 키 길이는 128비트이다. DES를 전사공격한다는 것은 2^{56}개의 가능한 모든 키를 검사하는 것을 말하며, 평균적으로 2^{55}번 시도를 하면 키를 찾을 수 있다. 전사공격을 실제 하기 위해서는 임의의 키로 복호화하였을 때 해당 복호화가 성공적인지 여부를 공격자가 확인할 수 있어야 한다. 다음 각각에 대해 답변하시오.

   ① 공격자가 소유한 하나의 프로세서가 1초에 10^7개의 복호화를 할 수 있다고 가정하였을 때와 10^{13}개의 복호화를 할 수 있다고 가정하였을 때 DES와 AES를 전사공격하기 위해 소요되는 시간을 계산하시오.

   ② 공격자가 같은 프로세서를 1,000개 보유가 있을 때 전사공격하기 위해 소요되는 시간을 계산하시오.

   실제 컴퓨터 프로그래밍을 이용하여 소요되는 시간을 년, 일, 시간, 분, 초로 제시하시오.

8. XOR을 이용한 one-time pad는 Shannon에 의해 암호문 단독 공격에 대해서는 완벽한 안전성을 제공한다고 증명되어 있다. 하지만 one-time pad는 3가지 문제점을 가지고 있다. 첫째 키가 너무 길다. 둘째, 매번 다른 키를 사용해야 한다. 셋째, NM 특성을 만족하지 못한다. 이와 관련하여 다음 각각에 대해 답변하시오.

   ① One-time pad는 $C = M \oplus K$ 형태로 암호화한다. 만약 같은 길이의 메시지 M_1과 M_2를 같은 길이의 키 K로 암호화하여 C_1과 C_2를 얻었다. C_1과 C_2를 획득한 공격자는 어떤 정보를 얻을 수 있는지 설명하시오.

   ② One-time pad로 암호화된 $C = M \oplus K$가 있을 때, 공격자가 C를 C와 같은 길이의 P로 XOR한 $C' = C \oplus P$로 바꾸었을 때 복호화 결과를 제시하시오.

   ③ One-time pad의 키 길이 문제 때문에 의사난수발생기 PRG를 이용하여 $C = M \oplus \mathrm{PRG}(K)$ 형태로 동작하는 암호알고리즘을 만들 수 있다. 즉, 사용자는 작은 길이의 키만 유지하면 되고, 필요한 실제 키 길이는 PRG를 이용하게 된다. 이 암호알고리즘이 안전하기 위해서는 PRG가 예측이 가능하지 않아야 한다. 초기 무선AP는 802.11b WEP이라는 프로토콜을 사용하여 무선AP와 연결된 기기간에 암호 통신을 하였다. 그것의 동작 방식을 간단하게 표현하면 다음과 같다. 무선AP와 무선기기는 104비트 크기의 대칭키 K를 공유한다. 메시지 M은 $C = M \oplus \mathrm{PRG}(IV||K)$와 같이 암호화한다. 여기서 IV는 24비트 카운터이며, 매번 다른 값으로 메시지를 암호화하기 위해 IV를 사용하고 있다. 암호화된 메시지는 IV와 함께 상대방에게 전달($IV||C$)된다. 참고로 IV는 0부터 시작하며, AP를 강제 초기화하면 이 값은 다시 0이 된다. 802.11b WEP의 문제점 두 가지를 제시하시오.

9. One-time pad를 포함하여 암호기술에서는 XOR 연산을 많이 사용한다. 이 연산을 많이 활용하는 이유는 표본공간 $U = \{0,1\}^n$(n비트 크기의 모든 메시지)이고 U에 정의된 확률변수 Y가 있을 때, X가 U에 정의된 균등확률변수(uniform random variable)이면 $Z := Y \oplus X$도 균등확률변수가 된다. 즉, Y의 확률분포와 무관하게 랜덤한 값으로 XOR하면 그 결과는 랜덤한 값이 된다는 것이다. 다시 말하면 랜덤하지 않은 메시지를 랜덤한 키로 XOR하면 그 결과는 랜덤한 값이 된다는 것이다. 균등확률변수란 모든 $a \in U$에 대해 $\Pr[X = a] = 1/|U|$가 되는 확률변수를 말한다. 주사위는 모든 면이 나올 확률이 같기 때문에 균등확률변수이다. $n = 1$일 때 $\Pr[Z = 0] = 1/2$임을 보이시오.

10. 서버는 각 사용자와 MAC 키를 공유하고 있으며, 메시지를 교환할 때 공유된 키로 계산된 MAC을 첨부해야 한다. 서버는 수신한 MAC의 유효성을 확인하기 스스로 MAC 값을 계산한 뒤에 최상위 비트부터 차례로 수신한 MAC 값과 비교한다. 서버는 비교하다 틀린 비트를 발견하면 즉시 MAC에 문제가 있다고 알려준다. 이 문제에서 설명한 방식의 문제점과 해당 문제점을 극복하는 방법을 설명하시오.