정보보안

CH #1 Overview of Computer Security

컴퓨터 보안(Computer Security)

* 정보시스템 자원의 무결성, 가용성, 기밀성을 보전하고자 하는 목표달성을 위해 자동화된 정보시스템에 제공하는 보호
* 컴퓨터 보안, 사이버 보안 또는 정보 기술 보안은 하드웨어, 소프트웨어 또는 데이터의 도난이나 손상, 컴퓨터가 제공하는 서비스의 중단 또는 오용으로부터 컴퓨터 시스템을 보호하는 것을 의미.

CIA Triad (CIA 삼각형)

* **기밀성(Confidentiality)**
  + 데이터 기밀성 : 개인정보 또는 기밀정보가 인증되지 않은 개인에게 제공되거나 공개되지 않도록 보장
  + 프라이버시 : 개인은 자신과 관련된 정보가 다른 사람과 누구에게 공개되는지 제어할 수 있어야 함.
* **무결성(Integrity)**
  + 데이터 무결성 : 정보 및 프로그램은 지정되고 승인된 방식으로만 변경되어야 함.
  + 시스템 무결성 : 시스템은 의도적/무단 조작 없이 기능을 수행해야 함.
* **가용성(Availability) \*\* 가장 중요**
  + 시스템이 신속하게 작동하고 승인된 사용자에게 서비스가 거부되지 않도록 보장해야 한다.

추가 원칙

* **인증(Authenticity)**
  + 정말 그 사용자인지, 시스템이 받은 자료가 신뢰할 수 있는 출처에서 온 것인지 확인할 수 있는 것.
* **책임추적성(Accountability)**
  + 개체의 행동을 추적해서 찾아낼 수 있어야 한다는 것.

컴퓨터 보안이 어려운 이유?

* 보안은 단순한 문제가 아님
* 보안 기능에 대한 잠재적인 공격을 고려해야 함
* 지속적인 모니터링 필요
* 보안 실패가 발생할 때까지 보안 투자의 이점이 거의 감지되지 않음
* 높은 보안성은 종종 사용자 편의적인 운영을 방해하는 것으로 간주됨

Terminology

* 자산(Asset)
  + 보호해야 할 대상
* 보안 정책(Security Policy)
  + 시스템이나 조직이 민감하고 중요한 시스템 리소스를 보호하기 위해 보안 서비스를 제공하는 방법을 지정하거나 규제하는 일련의 규칙 및 관행
* 취약점(Vulnerability)
  + 악용될 수 있는 시스템 설계, 구현 또는 운영의 결함 또는 약점
* 위협(Threat)
  + 취약점을 악용하여 해를 입힐 수 있는 가능한 위험
* 공격(Attack)
  + 자산에 대한 위협 실행
* 대응(Countermeasure)
  + 위협, 취약점 또는 공격을 제거하거나 방지함으로써 피해를 최소화하는 기술
* 공격자(Adversary)
  + 시스템을 공격하거나 위협하는 개체

OSI Security Architecture

* 네트워크를 통해 전송되는 데이터에 대한 보안을 제공하기 위해 OSI 모델의 7개 계층 각각에서 사용할 수 있는 보안 서비스 및 보안 메커니즘을 정의

수동적 공격 (Passive Attack)

* 시스템에서 정보를 탈취하거나 사용하려고 시도하지만 시스템 리소스에는 영향을 미치지 않음.
  + 도청, 감시, 전송
  + 공격자의 *목표는 전송되는 정보를 얻는 것*

1. *메시지 내용 공개 (The release of message contents*
2. *트래픽 분석 (traffic analysis)*

능동적 공격 (Active Attack)

* 시스템 리소스를 변경하거나 시스템의 동작에 영향을 주려고 시도.
* 다양한 잠재적인 물리적, 소프트웨어 및 네트워크 취약성으로 인해 방지하기 어려움
  + Active Attacks의 4가지 유형

1. ***Masquerade****(위장) : 한 개체가 다른 개체인 것처럼 가장할 때 발생.*
2. ***Replay****(재생) : 승인되지 않은 효과를 생성하기 위한 데이터 단위의 수동 캡처 및 후속 재전송을 포함.*

*\*\* a 가 b로 보낸 메시지를 공격자가 가로챈 뒤, 메시지를 b에게 다량으로 전송한다. B는 다량의 메시지에 대한 처리를 수행하다 리소스를 모두 사용하게 된다.*

1. ***Modification of Message*** *(메시지 수정) : 메시지 조작*
2. ***Denial of Service(DoS)*** *: 통신설비의 정상적인 사용 또는 관리를 방해하는 행위*

Security Services

* Authentication(인증)
  + 통신이 믿을 수 있는지 확인 (신원 증명)
    - * X800 정의 인증 서비스
        + ***데이터 출처 인증*** *: 단일 메시지의 경우 수신자가 수신한 메시지가 출처라고 주장하는 곳에서 온 것임을 확인.*
        + ***피어 개체 인증*** *: 지속적인 상호작용의 경우 두 개체가 서로 인증되고, 제 3자가 합법적인 두 당사자 중 하나로 가정할 수 있는 방식으로 연결이 방해되지 않도록 함*
* Access Control (접근 제어)
  + 통신 링크를 통해 시스템 및 애플리케이션에 대한 접근을 제한하고 제어하는 기능
* Data Confidentiality(데이터 기밀성)
  + 데이터 전체를 암호화 하는 경우 : 출처가 불분명
  + 데이터 일부를 암호화 하는 경우 : 헤더 정보 무방비 노출
* ***따라서 적절히 암호화 해야함***
* Data Integrity(데이터 무결성)
  + 무결성 수준이 올라가면 가용성이 떨어짐
    - * 데이터 전체에 대해서 무결성을 제공해 줄 것인지, 실제로 통신과 직접적으로 관련되어 있는 메시지 내용에 대해서만 무결성을 제공해 줄 것인지는 적절히 조정해야 한다.
* Non-repudiation(부인 방지)
  + 보낸 사람이 진짜 보낸 사람인지, 받는 사람이 제대로 받는 사람인지 확인하는 것
* Availability Service(가용성 서비스)
  + 보안시스템을 적용해도 정상적인 동작을 해치지 않도록 보장하는 것

Security Techniques

* 암호 (Cryptography) : 데이터를 변형시켜서 못 보게 하는 것
* 스테가노그래피(Steganography) : 데이터를 감춰서 못 보게 하는 것

고전 대칭키 암호

* 대칭키 암호를 대치암호/치환암호로 나누는 방식
* 스트림암호/블록암호로 나누는 방식

***\*\*Symmetric Key는 사전에 공유되어야 한다.***

Kerckoff’s Principle

* Kerckhoff의 원리에 따라 ***공격자는 항상 암복호 알고리즘을 알고 있다***고 가정한다.
* 암호의 ***안전성은 키의 안전성에만 바탕을 둔다***.
* 달리 말해, ***키를 알아내는 것이 매우 어려워서 암호/복호 알고리즘을 비공개로 할 필요가 없어야 한다.***

<Cryptoanalysis attacks>

* 암호분석공격
* 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

  자동 생성된 설명
* 암호문단독 공격 (Ciphertext-only Attack)
* 텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

  자동 생성된 설명
  + 전수조사 공격 (brfute-force)
  + 통계적인 공격 (statistical)
  + 패턴 공격 (pattern)
* 알고있는 평문 공격 (Known-Plaintext Attack)

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

* + 평문과 암호문의 쌍만 가지고 있는 경우
  + Ciphertext-only 공격보다 유용함
* 선택 평문 공격 (Chosen-Plaintext Attack)

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

* + 공격자가 임의의 평문에 대한 암호문을 얻어볼 수 있는 경우. 이를 통해 통계적 유추가 가능함.
* 선택 암호문 공격 (Chosen-Ciphertext Attack)

텍스트, 도표, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

* + 암호문 -> 평문 (선택 평문 공격보다 강력함)

<대치암호(substitution ciphers)>

* 대치암호는 하나의 기호를 다른 기호로 대체한다.
  + 단일문자 암호 (a -> t)
  + 다중문자 암호 (a -> t or a -> z ….)



* Hello 의 l 을 o로 대치했기 때문에 단일문자 암호임을 쉽게 유추할 수 있다.

<덧셈암호(additive cipher)>

* 덧셈 암호는 가장 간단한 ***단일문자*** 암호이다. ***Shift cipher*** 또는 ***Caesar cipher***로도 불린다.
* 텍스트, 폰트, 라인, 측정 스틱이(가) 표시된 사진

  자동 생성된 설명
* 텍스트, 스크린샷, 라인, 도표이(가) 표시된 사진

  자동 생성된 설명
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명
* 텍스트, 폰트, 스크린샷이(가) 표시된 사진

  자동 생성된 설명

<곱셈 암호 (Multiplicative Ciphers)>

텍스트, 스크린샷, 라인, 도표이(가) 표시된 사진

자동 생성된 설명

곱셈 암호에 대한 키 공간은?

* 26과 서로소인 관계의 키만 가능 (1~26 중 26과의 최대공약수가 1이 되는 요소만 키로 활용)
* 텍스트, 폰트, 스크린샷이(가) 표시된 사진

  자동 생성된 설명

< 아핀 암호(Affine Ciphers) >

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

* + 덧셈 암호와 곱셈 암호를 한 번씩 적용한 암호 기법
* 아핀 암호는 키 쌍(곱셈 암호에 대한 키, 덧셈 암호에 대한 키) 을 사용하며 키 공간의 크기는 ***덧셈 암호 키 (26) x 곱셈 암호 키 (12) = 312*** 이다.

텍스트, 폰트, 스크린샷, 영수증이(가) 표시된 사진

자동 생성된 설명

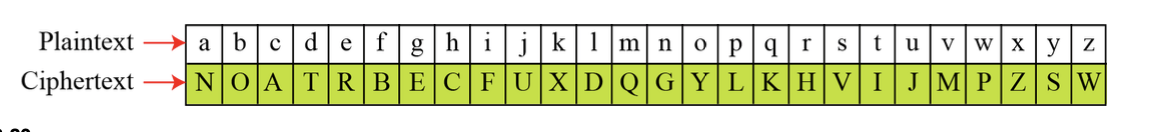
텍스트, 폰트, 영수증, 스크린샷이(가) 표시된 사진

자동 생성된 설명



* 암호화시엔 곱셈 이후 덧셈을 진행하지만, 복호화시엔 덧셈에 대한 복호화를 진행한 이후 곱셈에 대한 복호화를 진행한다.

단일문자 대치암호(Monoalphabetic Substitution Cipher)

* 키를 랜덤하게 배치하는 방법 

키 공간 => 26 !

<다중문자 암호 (Polyalphabetic Ciphers)>

* 평문 문자와 암호문 문자와의 관계는 일대다대응이다. 즉, 예를들어 a 는 D로 암호화될 수도 있고 N으로 암호화될 수도 있는것이다.
* 텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

  자동 생성된 설명
* 텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

  자동 생성된 설명
* 첫 번째 암호화 이후 각 문자에 대한 key는 이전 문자가 된다. 이에따라 키공간 역시 넓어지게 되는데, 만약, PlainText가 10개의 문자로 이루어져 있다고 한다면 키 공간은 26^10 을 이루게 된다.

<플레이페어 암호(Playfair Cipher)>

* 사각형, 직사각형, 번호, 폰트이(가) 표시된 사진

  자동 생성된 설명 텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

  자동 생성된 설명
* 두 개의 텍스트가 한 개의 쌍으로 암호화가 되는 구조

1. 평문을 2개의 알파벳 묶음으로 분리해준다
2. 두 알파벳이 같은 행에 있을 경우, 오른쪽으로 한 칸 씩 밀어주면 된다.
3. 두 알파벳이 사각형 모양을 그릴 경우 그 사각형을 기준으로 왼쪽과 오른쪽에 있는 알파벳으로 치환해 주면 된다.
4. 묶인 두 개의 알파벳이 같은 열에 있을 경우, 아래쪽으로 한칸 씩 밀어주면 된다.

\*\*만약, 연속으로 같은 알파벳이 나올경우에는 사이에 임의의 알파벳을 끼어 넣어준 뒤, 치환을 해준다. 또한, 2개의 묶음으로 나눠 떨어지지 않는 홀수개의 평문일 경우 맨 끝에 임의의 알파벳을 붙인 뒤, 치환해준다.

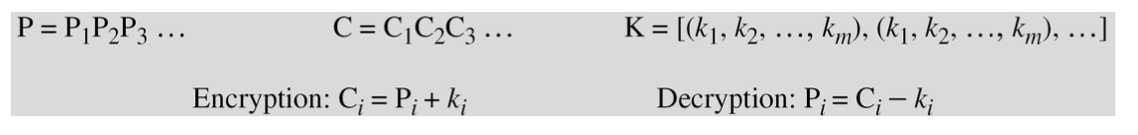
<Vigenere 암호 (Vigenere Cipher)>

* 키 값 자체를 하나의 텍스트로 두는 것(텍스트 자체가 키 값) 즉, 사람이 원하는 단어로 키를 구성하는 것
* Ex)

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

키 값(deceptive) 은 계속해서 반복적으로 나타난다.



이때 키 값의 길이 m 이 길어질수록 암호화 강도는 더 높아진다. 길이가 짧을 경우 키의 반복 주

기가 짧아지므로 암호문에서 패턴이 나타날 가능성이 높아진다. *(Kasiski Examination으로 특정 단어*

*들이 반복되는 구간을 찾을 수 있다)*

**++키의 길이 m =1 인 경우를 덧셈 암호 방식이라 볼 수 있다**

**텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명.**

* Vigenere Cipher가 다중문자 암호인 이유?
  + 단일문자 암호는 하나의 규칙을 사용해서 평문의 문자를 항상 같은 방식으로 바꾼다. 예를 들어, “A”는 항상 “D”로 바뀌고, “B”는 항상 “E”로 바뀐다. 즉, 같은 문자는 언제나 같은 결과를 내놓는다.
  + 반면, Vigenere Cipher는 여러 규칙을 번갈아 가면서 쓴다. 키가 여러 문자로 이루어져 있기 때문에, 평문이 암호화될 때 각 문자가 다른 방식으로 바뀐다. 예를 들어 “HELLO” 라는 단어를 암호화할 때 첫 번째 “L”과 두 번째 “L”이 다르게 바뀔 수 있다.
    - * 첫 번째 L은 키의 첫 번째 문자에 따라 바뀐다.
      * 두 번째 L은 키의 두 번째 문자에 따라 바뀐다.

<Kasiski Examination >

* + 암호문에서 두 번 이상 등장하는 문자열로부터 키의 길이를 유추하는 방식이다. 일반적인 영어 문장을 생각해보면 그 문장의 주제에 따라 반복해서 나오는 단어가 있기 마련이다.

**텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명**

예시에서 THIS 가 모두 WVOW로 암호화된 것을 볼 수 있다. 암호문을 가지고 키의 길이를 유추할 때 2번 등장한 WVOW가 동일한 평문일 것이라는 합리적 추론이 가능하고, WVOW 사이의 거리가 20이므로 키의 길이는 20의 약수인 20, 10, 5, 4, 2, 1 중 하나일 것이라고 추측할 수 있다.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

해당 예시의 경우 Difference의 최대 공약수는 m = 4 이다. 이때 블록의 키 값이 4개일 거라고 추측이 가능하게 되고, 키 공간의 크기는 26^4 이다.

<힐 암호(Hill Cipher)>

* 단일 문자 암호, 플레이페어 암호 모두 Ciphertext-only attack에 취약하다.
* 힐 암호의 경우 Ciphertext-only 로는 불가능하다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

* Plain Text를 Key 값에 맞춰서 행렬화 시켜버린다. 즉, 키 값이 4 \* 4 일 때 4를 기준으로 Plaint Text를 자른다.

텍스트, 스크린샷, 도표, 디자인이(가) 표시된 사진

자동 생성된 설명

복호화의 경우 암호문에 Key의 역행렬을 곱해주면 된다.

* 힐 암호는 Plain Text 와 Cipher Text의 일부 쌍만 알아도 쉽게 뚫릴 수 있다.

텍스트, 라인, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

m=3 임을 알고 있다고 가정하고, 위와 같이 세 개의 평문/암호문 블록 쌍을 가로챘다고 할 때, 이들 쌍을 합쳐 행렬 P 와 행렬 C를 만들어 낸다.

텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

이후 P의 역행렬과 C 행렬을 곱해 Key 행렬을 도출할 수 있다. 즉 Key 값을 유추할 수 있는 것이다. 이 유추된 Key 값을 가지고 Plain Text를 Cipher Text로 만들어 보는 과정을 반복하면 된다.

<Rotor 암호>

* 스크린샷, 텍스트, 도표, 디자인이(가) 표시된 사진

  자동 생성된 설명

평문 문자에 대한 암호문 문자 mapping 을 계속적으로 변환 시키는 암호이다.

한 문자를 Mapping하고나서 roter 를 1/6 바퀴 만큼 돌린다.

Ex) bee => BCA 로 암호화 된다.

<Enigma>

* 로터 암호 원리를 적용한 암호화 방식이 에니그마이다.

사전에 협의된 양만큼 Rotor를 돌려서 암호화 하는 방식이므로 Plain Text와 Cipher Text의 쌍을 가지고 있더라도 쉽게 분석할 수 없다.

<전치암호, Transposition Ciphers>

* 전치암호는 한 기호를 다른 기호로 대체시키지 않고, 대신에 그 기호의 위치를 바꾼다.

즉, 전치암호는 기호를 재정렬 시킨다.

<Keyless Transposition Ciphers>

* 텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

  자동 생성된 설명

Rail Fence 암호

* + 평문은 지그재그패턴으로 두 열에 배열하고 1열 끼리 연결, 2열 끼리 연결 후 이어주면 된다.
* 스크린샷, 측정 스틱이(가) 표시된 사진

  자동 생성된 설명

열의 개수를 합의하여 위와 같이 4 x 4 구조로 구성할 수도 있다.

이때 블록을 몇 개 단위로 나눌 것인가가 키 값이라고 보면된다.

추가정리본

<https://github.com/Youngeyaa/Symmetric-Key-Ciphers>

<Stream And Block Ciphers>

* 일반적으로 대칭키 암호를 두 개의 큰 부류, 스트림 암호와 블록 암호로 나눌 수 있다.

<Stream Ciphers>

* 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

  자동 생성된 설명

스트림 암호는 데이터를 한 비트 또는 한 바이트 씩 순차적으로 암호화하는 방식의 대칭 키 암호 방식이다. 이는 데이터를 작은 단위로 처리하며, 주로 실시간 통신이나 스트리밍 데이터와 같은 연속적인 데이터 처리에 적합하다.

스트림 암호는 한 번에 전체 메시지를 처리하지 않고, 계속해서 데이터를 입력받으면서 암호화 과정을 진행한다.

* ***덧셈 암호***와 ***단일문자 대치암호, Vigenere*** 또한 스트림 암호라고 볼 수 있다.
* 스트림 암호는 단일문자 암호이다. 그렇지 않다면 그 암호는 다중문자 암호이다.
* 키가 고정 돼 있고 거기에 맞춰서 평문이 사이퍼 텍스트로 변환되는 형태가 스트림 사이퍼 라고 볼 수 있다.

<Block Cipher>

* 범위를 지정해서 암호화를 수행하는 형태로, 평문을 특정 블록으로 나눠서 그 블록 단위로 암호화를 수행하면 Block Cipher로 구분이 된다.
* 블록의 크기가 1이면 Stream Cipher로 간주되기 때문에 블록의 크기는 2 이상이어야 한다. 결국 이 2개 이상의 텍스트로 이루어진 블록을 하나의 키 값으로 암호화를 수행한다라고 하면 그 방법들은 전부 Block Cipher에 속한다고 볼 수 있다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

\*\*\* Vigenere 암호가 Stream Cipher 인 이유?

* 반복되는 Key 텍스트가 Plain Text 문자 하나하나에 대응되어 암호화를 수행하기 때문.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

<현대 대칭키>

Modern Block Ciphers

* 현대 암호는 Stream Cipher를 더이상 사용하지 않는다.
* 현대 대칭키 블록 암호는 ***n 비트*** 평문 블록을 암호화 하거나 ***n 비트*** 암호문 블록을 복호화 한다. 암호화 혹은 복호화 알고리즘은 ***k비트*** 키를 사용한다.
  + *일반적으로 n은 64, 128, 256, 512 를 사용.*

텍스트, 스크린샷, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

Ex) 인코딩 과정에서 8비트 아스키 코드가 사용되고 64비트 블록 암호를 이용하여 암호화 하기를 원한다면, 100개의 문자로 구성된 메시지는 800bit 로 표현이 된다. 하지만, 평문은 64로 나누어져야만 하기 때문에 800 % 64 = ***32*** 만큼의 padding 이 추가되어야 한다.

+ 메시지 길이가 n 비트보다 짧으면 Padding, 길면 n 비트 블록단위로 분할

\*\* 평문이 129 비트인 경우 64 비트 키를 써서 블록 사이퍼를 수행하려면 남는 1비트 뒤에 63비트를 붙여주어야 한다. 이때 63비트를 붙이지 않고 1비트를 날려 버리는게 더 효율적이지 않은가?

* 복호화가 불가능하다.

현대 블록 암호는 대치 암호 혹은 전치 암호로 동작하도록 설계 되었다. (bit 단위로 동작)

* 대치 : 0 을 1로 바꾸거나 1을 9으로 바꿈
* 전치 : 0과 1로 이루어진 데이터를 재배열

***\*\*현대 블록 암호는 전수조사 공격에 안전하기 위하여 대치암호로 설계되어야 한다.***

<치환으로 모델화된 ***전치 블록 암호***>

* 텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

  자동 생성된 설명

텍스트, 폰트, 스크린샷, 영수증이(가) 표시된 사진

자동 생성된 설명

***<대치 블록 암호***는 치환으로 모델화 할 수 없다, 하지만 입력 값을 디코딩하고 출력 값을 인코딩할 수 있다면 대치암호를 치환암호로 모델화 할 수 있다.>

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷, 그린이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

* ***일반적으로 현대 블록 암호는 전체 키가 아니라 부분 키 대치 암호이다. DES의 경우 처럼 키의 크기를 전체 64비트가 아니라 조금 작은 크기로 설정한다는 의미이고, 키의 크기를 줄인 대신에 다른 테크닉들을 활용해 암호화 강도를 높이는 방법을 사용한다.***
* ***키의 값을 무작정 키우면 암호 강도는 올라가지만 연산량이 많아져 비효율 적이다.***

P-Boxes (Permutation Box)

* P-박스(치환 박스)는 문자 단위로 암호화를 수행하였던 고전 전치 암호를 병렬적으로 수행한다.
* *도표, 라인, 그래프, 스크린샷이(가) 표시된 사진

  자동 생성된 설명*

P-box 는 3개의 옵션이 존재한다

1. *Straight P-box*
2. *Compression P-box*
3. *Expansion P-box*

Straight P-box

* 5개가 들어왔을 때 permutation 시켜 5개가 나가는 형태 (전치 암호)라인, 도표, 스크린샷, 그래프이(가) 표시된 사진

  자동 생성된 설명

Compression P-box

* 5개가 들어가서 3개로 적어지는 형태

라인, 도표, 스크린샷, 그래프이(가) 표시된 사진

자동 생성된 설명

Expansion P-box

* 3개가 들어와서 5개로 늘어나는 형태

라인, 스크린샷, 도표, 그래프이(가) 표시된 사진

자동 생성된 설명

폰트, 라인, 텍스트, 그래프이(가) 표시된 사진

자동 생성된 설명

Compression P-Box의 경우 만약 32 비트가 들어왔다면 24비트로 나가게 되는 구조이다. 때문에

8개의 숫자가 날아갔다고 생각하면된다.

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

Expansion P-box의 경우 적은 비트가 입력이 들어와서 많은 비트가 출력이 되다보니 비트가 중

복 돼서 출력되는 경우가 생긴다. (1)

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

<P-Boxes: 역함수의 존재성 >

* 단순 P-box 는 역함수가 존재하지만, Compression P-box와 Expansion P-box는 역함수가 존재하지 않는다.

텍스트, 스크린샷, 번호, 라인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 라인, 도표이(가) 표시된 사진

자동 생성된 설명

<S-Box>

* S-Box는 대치 암호의 축소모형으로 생각할 수 있다.

텍스트, 폰트, 화이트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

S-Box는 선형, 비선형 형태로 나타낼 수 있다.

<S-boxes : 역함수의 존재성>

* S-Box는 입력 값과 출력 값 사이의 관계가 테이블 혹은 수학적 관계로 정의되는 대치 암호이다.
* S-Box는 역함수가 존재할 수도 있고, 존재하지 않을 수도 있다.
* 역함수가 존재하는 S-Box를 만드려면 ***입력 비트와 출력 비트의 개수를 동일화 시켜줘야 한다.***

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

<배타적 논리합 (Exclusive-Or) >

* 현대 블록암호의 중요한 구성요소는 배타적 논리합 연산이다.
* 일반적으로 암호화 과정과 복호화 과정에 동일한 값을 갖는 키를 배타적 논리합 연산에서 입력 값 중 하나라고 한다면 ***배타적 논리합 연산은 자기 자신을 역으로 갖는다.***
* 항등원 : 어떤 값을 그 연산에 적용해도 값이 변하지 않는 요소.
* 역원 : 어떤 값과 연산했을 때 항등원을 만들어내는 값

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

XOR에서의 역원: 폰트, 로고, 상징, 그래픽이(가) 표시된 사진

자동 생성된 설명

* + XOR에서의 항등원은 0이다. 어떤 값과 그 값의 역원을 XOR 연산하면 0이 나와야 한다.

그런데 XOR에서는 모든 값의 역원이 자기 자신이다. 예를 들어, A XOR A = 0, 즉 자기 자신을 XOR 하면 결과가 0이 된다.

XOR에서의 항등원: 폰트, 그래픽, 로고, 텍스트이(가) 표시된 사진

자동 생성된 설명 ***폰트, 텍스트, 상징, 그래픽이(가) 표시된 사진

자동 생성된 설명***

* + 항등원은 어떤 값을 그 연산에 적용해도 값이 변하지 않는 요소를 말한다. XOR 연산에서는 어떤 값 A에 0를 XOR하면 A가 그대로 유지되기 때문에, 0이 항등원이 된다.

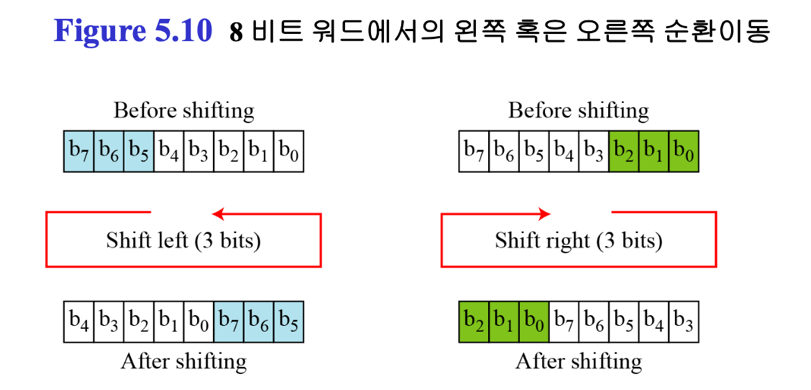
* ***따라서, XOR 에서 항등원은 0 이고, 역원은 자기 자신이다.***

***도표, 라인, 그래프, 디자인이(가) 표시된 사진

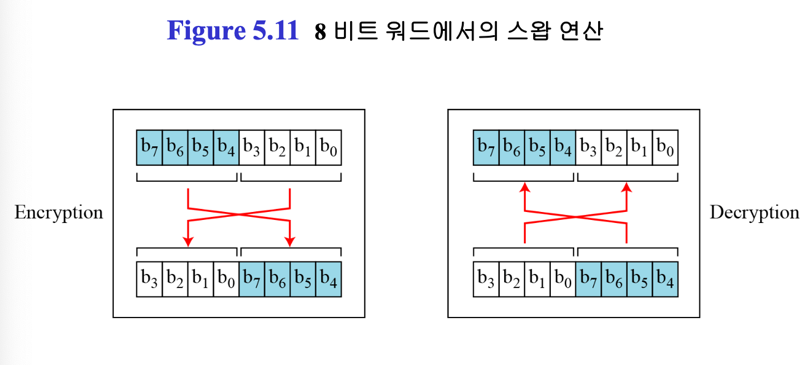
자동 생성된 설명***

* **암호화하는 키와 복구하는 키가 동일한 키 값을 쓸 수 있게 되는 구조가 XOR을 사용하는 가장 큰 장점이라 볼 수 있다.**

<순환이동, Circular Shift Operation>



<스왑, Swap>



* 스왑도 결국 Shift 연산이라고 볼 수 있다. 4bit 단위로 Shifting을 하면 위 그림처럼 4bit 단위로 스왑을 하게 되는 효과를 보인다.

<분할 및 결합, Split and Combine>

* 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

  자동 생성된 설명

<합성 암호, Product Ciphers>

* 앞 절에서 다루었던 대치, 치환, 그리고 기타의 구성요소(Shift, Swap, Split and Combine)를 결합한 복합적인 암호.
* 고려사항
  + 확산(Diffusion) : 암호문과 평문사이의 관계를 숨기는 것. *(평문의 한 비트가 바뀌더라도 전혀 다른 암호문이 나오게 됨)*
  + 혼돈(Confusion) : 암호문과 키의 관계를 숨기는 것. *(키가 한 비트 바뀌면 전체 암호문의 결과가 바뀌게 됨)*
  + 라운드(Rounds) : 한 스텝으로 암호화를 구성하면 쉽게 뚫릴 수 있으니 다중 스텝을 만든다는 개념. *(여러 라운드를 통해 암호화 수행)*

텍스트, 스크린샷, 도표, 평면도이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

Ex)

* 하나의 비트가 들어가서 4개의 비트 값이 나오는 구조
* 8이 키 값을 만나서 7, 8이 생성. Permutation을 통해 각각 4bit, 2bit 자리로 이동. 이 Permutated 된 Middle Text가 두 번째 키 값을 만나서 4개의 비트 값이 생성.

<Two Classes of Product Ciphers>

* 현대 블록 암호는 모두 합성 암호이며, 두 가지 종류로 분류된다.
  + Feistel 암호 (DES)
  + 비-Feistel 암호 (AES)

<Feistel 암호>

* Feistel 암호는 세 가지 타입의 구성요소를 가질 수 있다.
  + 자기 자신을 역으로 갖는 것 (self-invertible)
  + 역함수가 존재하는 것 (invertible)
  + 역함수가 존재하지 않는 것 (noninvertible)

텍스트, 스크린샷, 도표, 직사각형이(가) 표시된 사진

자동 생성된 설명

* Plain Text가 Cipher Text로 가는 중간 과정이 여러 개가 있는데, Mixer는 키 값을 통해서 Plain Text를 연산할 수 있게 만들어 준다.
* Mixer는 자기 자신을 역함수로 갖는다. (XOR 연산 구조)
  + 키 값을 XOR 연산이 되게 만들어 줌.

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

***Ex) 평문과 암호문의 길이가 4bit , 키의 길이가 3bit. 함수(Mixer)는 키의 첫 번째와 세 번째 비트를 입력 받는다. 두 비트는 십진수로 인식하고, 제곱한 뒤, 그 결과 값을 4bit 이진수로 출력한다. 만약 평문이 0111 이고 키가 101일 때 암호화와 복호화 결과 값은?***

텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명 스크린샷, 텍스트, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

* *장점 : Mixer를 잘 설계하면 키의 bit에 구애를 받지 않는다. 위와 같이 3bit 의 Key를 Mixer가 4bit로 늘려줬기 때문에 XOR 연산이 가능했다. 즉, Mixer만 잘 설계되어 있다면 입력으로 받는 Key 값은 크게 중요하지 않다.*

<Feistel 구조의 향상>

텍스트, 스크린샷, 도표, 직사각형이(가) 표시된 사진

자동 생성된 설명

* 함수의 입력 값으로 키만 사용하는 것이 아니라, 평문 혹은 암호문의 일부를 사용
* 위 암호화 과정을 여러 라운드에 걸쳐 반복하게 되면 공격자가 키 값을 유추하는 과정이 굉장히 어려워진다.

텍스트, 도표, 평면도, 스크린샷이(가) 표시된 사진

자동 생성된 설명

***<두 라운드를 갖는 Feistel 암호>***

스와퍼가 추가되긴 했지만 결국 첫 번째 라운드는 Left 비트를, 두 번째 라운드는 Right 비트에 대한 암호화를 수행한다고 볼 수 있다.

***해당 방식은 구조를 공개했을 때 쉽게 뚫릴 수 있다. 이를 대비하기 위해선 라운드 수를 늘려버림으로써 키 값 유추를 어렵게 할 수 있다.(경우의 수 증가) 이러한 구조를 띄는 요소가 DES 이다.***

<비-Feistel 암호>

* AES가 대표적. 이후에 설명

<블록 암호에 대한 공격>

* 키 사이즈가 매우 크기 때문에, 키에 대한 전수 조사 공격은 거의 불가능하다.
  + 차분 분석 (Differential Cryptanalysis)
* 차분 : 두 개의 다른 입력(평문)의 XOR 결과.

Ex)

라인, 도표, 상징, 스크린샷이(가) 표시된 사진

자동 생성된 설명

암호가 XOR 연산으로만 구성된다고 가정. 키의 값을 알지 못하더라도, 평문의 차분을 ***P1 XOR P2*** 로 정의, 암호문의 차분을 ***C1 XOR C2***로 정의하면 평문 차분과 암호문 차분 사이의 관계를 쉽게 발견할 수 있다.



***C1 XOR C2*** 에서 동일한 Key들이 XOR 되면 0 (XOR시 자기 자신이 역원) 이기 때문에 ***P1 XOR P2*** 와 동일함을 보일 수 있다.

\*전수조사 보다는 상대적으로 빠르게 찾을 수 있다.

ㄴ