# 암호

# 암호의 필요

- 문서의 보안
- 정확한 절달
- 송신자 확인
- 메시지의 정확성
- 수신 부인 방지

- 계산적으로 안전한 암호
- 빈도분석 : 평문에 등장하는 문자의 출현 빈도와, 암호문에 등장하는 문자의 출현 빈도가 일치하는 것

- 해독이 가능해지는 조건
    1. 암호화 알고리듬이 알려져 있는 경우
    2. 문자 출현율의 치우침 등 평문에 대해 통계적 성질의 데이터가 있는 경우
    3. 암호화의 예문을 많이 가지고 있는 경우

- 절대 안전한 암호 : 버넘암호
- 안전한 암호
    1. 절대 안전한 암호 : 버넘암호처럼 이론적으로 해독이 불가능한 것
    2. 계산적으로 안전한 암호 : 해독하는 데 채산이 맞지 않을 정도로 수고와 시간이 소요되며, 현대의 상용 암호에 사용됨

- 버넘암호 : 1회만 사용하는 난수를 바탕으로 하는, 즉 쓰고 버리는 키를 사용해 해독 불가능한 암호를 만들 수 있는데, 그런 암호문은 재현성이 없다.평문 P에 길이가 같은 난수의 줄을 덧붙여 암호문 C를 만드는 것
    1. 알파벳을 문자코드로 변환
    2. 1회만 사용하는 난수를 가산
    3. 26으로 나눈 나머지를 계산
    4. 문자코드를 이용하여 알파벳으로 변환

# 암호의 기초

- 클로드 섀넌 (Claude E. Shannon; 1916~2001) : 20세기의 영국 수학자. 1948년 [통신의 수학적 이론]이라는 논문을 써서 정보 이론의 아버지라 불리게 되었다.

## 기본 용어

- 평문 P (Plain Text) : 암호화되어 있지 않은 보통의 글
- 암호문 C (Cipher Text) : 암호화된 글
- 암호화 (Encryption / Encipherment) : 평문을 암호문으로 바꾸는 것
- 복호 (Decryption / Decipherment) :암호문을 평문으로 되돌리는 것

- 암호화키 $E_k$ (Encryption Key) : 암호화에 사용되는 키
- 복호키 $D_k$ (Decryption Key) : 복호에 사용되는 키

# 암호화키와 복호키의 관계

- 평문P → 암호화키$E_k$ → 암호문C : $C=E_k(P)$
- 암호문C → 복호키$D_k$ → 평문P : $P=D_k(C)$

# 순열/조합

순열 P (Permutation)

$_nP_r = n(n-1) \times (n-2) \times \cdots \times (n-r+1) = \frac{n!}{(n-r)!}$

조합 C (Combination)

$_nC_r=\frac{_nP_r}{r!}=\frac{n!}{(n-r)!r!}$

계승 ! (factorial)

$n!=n\times(n-1)\times\cdots\times3\times2\times1$

# 2진수(기수법)

# 배타적논리합(논리연산)

- OR(논리합)
- AND(논리곱)
- NOT(부정)
- NAND(부정논리곱)
- NOR(부정논리합)
- XOR(배타적논리합)
    1. 평문 $\oplus$ 암호화키 = 암호문
    2. 암호문 $\oplus$ 복호키 = 평문
    3. 암호문 $\oplus$ 평문 = 복호키
    4. 공통키암호 = 환자처리 + 전치처리 + XOR연산

# 의사난수열

아무 의미가 없는 수의 나열

# 대합

대합(involution)이란 2회 변환할 경우 원래 상태로 돌아가는 변환을 말한다.

```
예)
  1 -> 4 -> 1
  2 -> 3 -> 2
  3 -> 2 -> 3
  4 -> 1 -> 4
```

# 일방향함수

한쪽 방향으로는 계산이 가능해 답이 나오더라도 반대 방향으로 계산하는 것이 매우 곤란한 성질을 일방향이라 하며, 그런 성질을 가진 함수를 일방향함수라고 한다.

## 소인수분해문제

커다란 두 소수를 곱하여 그 결과를 얻는 것은 간단하다. 그렇지만 반대로 곱해진 수(합성수)로부터 원래의 두 소수를 구하기는 매우 어렵다. 합성수로부터 원래의 소수를 구하는 것을 소인수분해문제라고 한다.

### 소수/합성수

- 소수: 자기 자신과 1로만 나누어진다.
- 합성수: 소수가 아닌 수, 소수의 곱으로 표현 가능한 수.
- 1은 소수에 포함하지 않음 → 소인수분해의 일의성 유지

### 에라토스테네스의 체

자연수 N이 $\sqrt{N}$이하의 모든 소수로 나누어지지 않으면 자연수 N은 소수이다.

N=pq이라 할 때

$p\leqq\sqrt{N}$

$q\leqq\sqrt{N}$

$p<\sqrt{N}$ 그리고 $q>\sqrt{N}$

$pq > N$

### 소수 판정

에라토스테네스의 체는 확실하게 소수를 찾아내는 방법이다. 그렇지만 큰 수가 소수인지 아닌지 판정할 경

우에는 처리 시간이 오래 걸린다.

그래서 100% 확실하지는 않지만 확률적으로 거의 소수임을 판정하는 방법이 사용된다.

페르마의 방법에서는 어느 정수 a와, 소수인지 아닌지를 판정하는 수 n에 대하여 $a^{n-1}=1(\bmod n)$이면 n이 소수임을 확률적으로 판정할 수 있다. 그러나 소수가 아닌 수(합성수)를 소수로 판정할 위험이 있다.

그래서 페르마의 방법의 결점을 개선한 것이 밀러-라빈의 방법이다. 1회의 테스트로 잘못된 판정이 발생할 확률은 페르마의 방법의 4분의 1 이하로 확실하게 소수를 판정할 수 있다.

- 의사소수 : 소수로 판정될 가능성이 있는 수.

### 유클리드 호제법

### 확장된 유클리드 호제법

### 페르마의 소정리

n이 소수일 때 n과 서로소인 정수 a(n의 배수가 아닌 정수 a)에 대해 다음 식이 성립한다.

$a^{n-1}=1(\bmod n)$

즉, a를 n-1제곱한 것을 n으로 나누면 나머지가 1이 된다.

### 오일러의 정리

자연수 n과 서로소인 정수 a에 대해 다음의 식이 성립한다.

$a^{\varphi(n)}=1(\bmod n)$

식 안의 $\varphi$를 오일러함수라고 한다. 오일러함수값 $\varphi(n)$은 1부터 n까지의 자연수이면서, n과 서로소인 수의 개수를 나타내는 것이다.

그리고 $a^{\varphi(n)}\times a=a^{\varphi(n)+1}$이므로 틀림없이 다음 식도 성립한다.

$a^{\varphi(n)+1}=a(\bmod n)$

왜냐하면, $a^{\varphi(n)}=1(\bmod n)$이므로 a를 $(\varphi(n)+1)$ 제곱하면 a로 되돌아가는 것을 의미하기 때문이다.

나아가 거듭제곱을 해나가면 $2\varphi(n)$ 제곱에서 1이 되고 $(2\varphi(n)+1)$ 제곱에서 a로 돌아간다. 이것을 일반적으로 나타내면 자연수 n과 서로소인 정수 a에 대해 다음과 같이 된다.

$a^{k\varphi(n)}=1(\bmod n)$

$a^{k\varphi(n)+1}=a(\bmod n)$ (k는 음이 아닌 정수)

더불어, 1부터 (n-1)까지의 모든 정수 a에 다음과 같은 식이 성립한다.

$a^{k\varphi(n)+1}=a(\bmod n)$

n이 소수라면 1부터 n까지의 자연수에서 n과 서로소가 아닌 것은 n뿐. 즉, $\varphi(n)=n-1$ 이므로 $a^{\varphi(n)}=a^{n-1}=1(\bmod n)$이 되어 페르마의 소정리와 일치한다.

### 이산로그문제

다음과 같은 합동식을 생각해보자

$a^x=y(\bmod p)$

a와 x가 알려져 있는 경우 y를 구하기는 비교적 쉽다. 그렇지만 a와 y가 알려져 있더라도 y의 로그인 x를 구하기는 매우 곤란하다. 이것이 이산로그문제이다. 이산이란 연속의 반대말이며, 건너뛰는 값임을 나타낸다.

## 모듈로연산

- 의사난수 생성

$a=b(\bmod N)$

이것이 일반적인 합동식의 모양이며 모듈로 연산이라고 한다. 두 정수 a와 b ㅇ의 차가 N으로 나누어 떨어질 때 a와 b는 합동이라 읽는다. 등호 '='대신에 합동을 나타내는 부호 '$\equiv$'를 사용하는 경우도 있다.

### 모듈로연산의 덧셈과 뺄셈

### 모듈로연산의 곱셈과 나눗셈

# 고전적 암호

## 시저암호

평문의 각 문자를 순서대로 n문자 옮겨 암호화하는 알고리듬으로 만드는 암호
n의 값 만큼 문자를 뒤로 이동

## 환자식암호(대입암호)

시저 암호를 복작하게 해서 각 문자마다 옮기는 숫자를 변화시킨 것. 평문과 암호문의 각 문자를 1:1로 다른 문자에 대응시키는 것을 '단일환자암호'라 한다.

변환규칙 $\sigma$=$E_k$

키의 수 (알파벳개수)

$_{26}P_{26}=26!=26\times25\times24\times\cdots\times3\times2\times1\fallingdotseq4.03291461\times10^{26}$

## 다표식암호

평문을 n문자씩의 블록으로 나누고, 각 블록 안에서 문자를 옮기는 수를 바꾸는 것

변환규칙 $\delta$

키의 수 알파벳개수 26, 블록 내의 글자수 n

$26\times26\times\cdots\times26\times26=26^n$

## 전치식암호

평문을 n문자씩 블록으로 나누고, 각 블록 안에서 문자의 순서를 바꾸는 암호

치환규칙 $\tau$

키의 수 1블록을 n자로 했을 경우

$_nP_n=n\times(n-1)\times(n-2)\times\cdots\times3\times2\times1=n!$

# 공통키암호

Common Key Cryptography = Symmetric Key Cryptography = Secret Key Cryptography

n명의 이용자가 공통키암호로 서로 통신할 경우 $_{n}C_{2}=\frac{n(n-1)}{2}$

## 공통키 암호의 특징

- 키가 알려지지 않게 배송이나 보관에 주의할 필요가 있다.
- 계산량이 적어 고속으로 암호화와 복호가 이루어지기 때문에 대량의 데이터 통신에 적합하다.
- 다수의 키와 그 관리가 필요하므로 불특정 다수와 통신하기에 적합하지 않다

## 스트림 암호

# 블록암호

## 블록암호의 해독

- 전수조사해독법 : 가능한 모든 경우의 키를 조사하여 키를 찾는 방법
- 차분해독법 : 입력차분이 그대로 출력차분이 되는 XOR 연산의 성질을 이용해 키를 찾는 방법
- 선형해독법 : S-BOX를 선형근사(일차함수의 직선에 근사)시켜 확률적으로 출력을 추정하는 방법

# DES 암호의 구조

## DES의 암호화키 생성

## DES의 비선형함수 f의 구조

## DES에 의한 암호화와 복호의 기본 구성

## DES의 결점

- 키의 길이가 짧다. 키의 길이가 짧으면 처리 속도가 느려지거나 쉽게 해독된다.
- S-BOX의 설계 기준이 없기 때문에 미약한 구현이 나돌기 쉽다.

## 간이형 DES에 의한 암호화와 복호의 실제

### DES 암호문의 생성

### DES 암호문의 복호

## 3-DES

# AES

## AES암호의 개요

# 공개키암호

Public Key Cryptosystem = Asymmetric Key Cryptosystem

이용자 n명이 서로 암호 통신을 한다고 하더라도 키의 총수는 2n개 있으면 충분

- 자주 쓰이는 공개키암호방식의 종류
  - 소인수분해 : RSA암호, 라빈(Rabin)암호 등
  - 이산로그문제 : 엘가말(ElGamal)암호, 타원곡선암호, DSA인증 등
- 키의 개수 $2^n$

# RSA암호의 구조

## RSA암호의 탄생

RSA 암호는 1977년 공표된 세계 최초의 공개키 암호이다. RSA라는 이름은 이를 개발한 미국의 세 연구자 리베스트(Rivest), 샤미르(Shamir), 에이들먼(Adleman)의 머리글자로부터 유래한다.

암호의 강도를 보증하는 것은 소인수분해문제이다. 과학 전문지 "사이언스"에 이들 3인이 만든 문제가 게제되었는데, 그것은 어떤 수를 소인수분해하여 메시지를 해독하라는 것이었다.

그 수는 다음의 129자리 자연수이다.

```
114381625757888867669235779976146612010218296721242
362562561842935706935245733897830597123563958705058
989075147599290026879543541
```

이 소인수분해는 17년 후인 1994년에 약 1600대의 컴퓨터를 사용해 계산함으로써 메시지가 복호되었다. 일반적으로 17년이라는 매우 오랜 세월이 걸렸다고 생각되겠지만, RSA 개발자의 한 사람이었던 리베스트는 1000년은 걸릴 것이라고 예상했던 만큼 출제자들에게는 짧게 여겨졌던 것 같다. 참고로 해독된 메시지는 'THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE'이다.

현재 RSA암호에 사용되는 숫자는 10진수로 300자리 이상이다. 이것을 소인수분해하려면 천문학적인 시간이 걸리게 된다.

## RSA암호의 암호화와 복호

## RSA암호의 키 생성법

## 공개키와 비밀키 만드는 법

## RSA암호문의 생성

# 엘가말 암호

- 공개키암호와 이산로그문제

엘가말암호의 암호화와 복호

# 암호의 실제 적용

## 하이브리드 암호

- PGP
- SSL/TLS

## 해시함수와 메시지 인증코드

### 해시함수

해시함수를 사용하여 본래의 메시지로부터 해시값을 계산한다. 지문이 본인 확인의 수단으로써 유효한 것처럼 해시값은 메시지의 지문이라고 할 수 있다. 메시지를 요약한 것어로 데이터의 양이 적어지고, 고정된 크기를 가진다.

'메시지가 변조되지 않은 것을 수신자가 확인할 수 있다'는 성질을 무결성(완전성, Integrity)이라고 하며 송신자가 본래의 메시지와 해시값을 함께 송신함으로써 무결성이 보증된다. 즉, 메시지의 지문을 단서로 하여 변조의 유무를 체크하는 것이다. 수신자는 송신자와 같은 해시함수를 사용하여 메시지의 지문인 해시값을 계산하여 첨부된 해시값과 비교한다. 해시값이 같다면 변조되지 않은 것임을 알 수 있다.

해시함수는 일방향함수이다. 메시지로부터 해시값을 계산할 수는 있어도 반대로 해시값으로부터 메시지를 복원할 수 없게 하기 위함이다. 이런 성질을 비가역성이라 하며, 이런 성질을 가진 해시함수가 일방향해시함수이다.

또한, 해시함수값이 일치하는 듯한 서로 다른 메시지의 세트를 찾는 것은 어려워야 하는데, 이 조건을 강충돌내성이라 한다. 더욱이 어떤 메시지가 조어졌을 때 해시값이 흡사해지는 다른 메시지를 발견하는 하는 것은 어려워야 하는데, 이 조건을 약충돌내성이라 한다. 이런 목적으로 개발된 해시함수에는 MD5, SHA-1, SHA-256, SHA-512, RIPEMD-160 등이 있다.

### 메시지 인증코드의 구조

메시지의 무결성을 분명히 하고 인증하기 위한 절차가 메시지 인증코드이다.

송신자는 보내고 싶은 메시지와 함께 그 메시지로부터 생성된 MAC 값을 송신한다. MAC값이란 해시값과 마찬가지로 검사에 이용하는 값이다.

수신자는 수신한 메시지로부터 생성된 MAC값과 수신한 MAC값을 비교함으로써 메시지의 무결성과 인증을 담보한다. 이때 송신자측과 수신자측 모두 MAC값의 생성을 위해 공통키를 이용한다.

이 두 개의 MAC값이 동일할 때, 송신자로부터의 메시지가 중간에 변조되지 않았다는 것과(무결성), 송신자가 키를 공유한 올바른 송신자인 것을 확인할 수 있다(인증).

그러나 두 개의 MAC값이 다를 때는, 송신자로부터의 메시지가 중간에 변조되었다는 것과 송신자가 키를 공유한 올바른 송신자가 아닌 것을 확인할 수 있다.

메시지 인증코드는 국제금융거래나 온라인쇼핑 등에서 이용하는 SSL/TLS에서 이용되고 있다.

- 메시지 인증코드의 두 가지 결점
  1. 부인 방지(Non-Repudiation)를 할 수 없다
  2. 제3자에 대한 증명을 할 수 없다

## 전자서명

전자서명은 송신자가 자신의 비밀키로 메시지를 암호화한 것이고, 메시지완 같이 수신자에게 보낸다.

수신자는 송신자의 공개키로 서명을 복호하여 메시지를 얻는다. 그리고 복호한 메시지와 보내져온 또 다른 메시지와 비교한다.

양자가 동일하다면 무결성의 검증과 송신자의 인증이 동시에 이루어진 것이다. 또한, 송신자의 공개키로 복호하기 위해 제3자도 수신자와 같은 서명을 검증할 수 있어 제3자에 대한 증명이 가능함과 동시에 송신자의 부인 방지가 된다.

전자서명은 SSL/TLS의 서버 정당성을 인증하는 서버 인증서를 작성하기 위해서도 이용된다. 인증서라고 하는 것은 공개키(이 경우는 서버의 공개키)에 전자서명을 부가한 것이다. 또한, 다운로드용 소프트웨어에 전자서명을 부가하여 소프트웨어가 변조되는 것을 방지하기 위해서도 이용되고 있다.

## 인증서와 인증국

인증서란 공개키에 그 공개키의 전자서명을 부가한 것으로 인증국에 의해 발행된다. 공개키를 공개하고 싶은 이용자는 인증국(CA:Certification Authority)에 자신의 공개키를 등록하고, 동시에 인증서의 발행을 의뢰한다.

의뢰를 받은 인증국은 공개키를 공개하고 싶은 이용자의 정당성을 확인하여, 인증국의 기준에 합치한다면 공개키를 바탕으로 전자서명을 작성하고, 공개키와 전자서명을 묶어서 인증서를 작성한다. 공개키와 비밀키의 한 쌍은 이용자가 작성하는 경우와 등록할 때 인증국이 작성하는 경우가 있다.

인증서를 사용한 공개키 검증의 구조는 공개키가 이용자 A의 것임을 보증한다. 그렇기 때문에 이용자 A는 신뢰할 수 있는 제3자인 인증국에 공개키의 정당함을 증명받는다.

1. 이용자 A는 인증국에 자신의 공개키의 인증서 발행을 의뢰한다.
2. 인증국은 이용자 A의 본인 확인을 한 수 에 인증서를 발행한다. 발행된 증명서는 이용자 A의 공개키에 인증국이 전자서명을 부가한 것이다.
3. 인증국은 리포지토리(Repository;데이터 보관 장소)에 인증서를 보존한다.
4. 이용자 B가 리포지토리로부터 이용자 A의 인증서를 다운로드한다.
5. 이용자 B가 이용자 A의 인증서에 포함되는 전자서명을 인증국의 공개키로 복호한다.
6. 복호한 키를 인증서에 포함되는 공개키와 비교하여 검증한다. 이 2개의 키가 같다면 인증서에 포함되는 공개키는 이용자 A의 것임이 보증된다.

이상의 절차에 의하여 이용자 B는 보증된 이용자 A의 공개키를 얻을 수 있다. 보증된 이용자 A의 공개키를 사용한다면 이용자 A의 비밀키로 암호화한 전자서명이 부과된 메시지가 정당한 메시지임을 검증할 수 있다.

정당한 메시지란 이하의 세 가지 조건을 동시에 만족하는 것이다.

1. 메시지에 변조된 흔정이 없어야 한다.
2. 제3자가 이용자 A를 사칭하여 보낸 메시지가 아니어야 한다.
3. 이용자 A는 그 메시지를 자신이 보낸 것임을 부인할 수 없어야 한다.

공개키의 정당함을 증명함으로써 전자서명이 부과된 메시지의 정당함의 조건 1~3을 증명할 수 있다. 이것을 바탕으로 공개키암호기반(PKI)의 구조가 완성된다.

- 사칭 → MAC (Message Authentication Code)
- 부인 → 전자서명 (Digital Signature)
- 중간자공격 → 인증서와 인증국

## 공개키암호기반(PKI)

Public Key Infrastructure

# Etc

## 의사난수와 암호보안

난수, 바로 랜덤(무질서)한 숫자의 열음 보안기반기술의 하나이다. 예를들면, 공개키암호의 경우 정보를 암호화하거나 복호(해독)하는 데에 사용하는 키는 난수를 사용하여 작성한다. 매회 같은 키를 사용한다면 해독당할 가능성이 있기 때문에 공개키암호를 사용할 때마다 새로운 키를 생성하여 안전성을 높이고 있다. 이것은 난수를 알게 된다면 암호가 뚫려서 돈을 도난당하거나 개인 정보가 유출될 가능성이 있기 때문이다. 도청당하지 않을 암호로 만들기 위해서는 난수가 가지고 있는 예측불가능성(과거의 숫자의 열부터 다음의 수를 예측할 수 없다는 성질을 적극적으로 이용해야 한다.

난수는 의사난수와 최근 주목되고 있는 물리난수(진성난수)로 크게 나뉜다. 그 중에서 의사난수는 일정한 계산식에 바탕을 두어 생산되므로 같은 주기나 패턴이 출현하여 '완전하게 랜덤한 숫자의 열'은 될 수 없다. 그렇기 때문에 난수를 추정당해 보안이 뚫리는 위험이 있다. 대표적인 의사난수 생성기에는 선형합동법, 평방잉여법, M계열, BBS(Blum-Blum-Shub)법, 일방향해시함수를 사용하는 방법, 암호를 사용하는 방법 등이 있다.

반면에 물리난수는 자연계의 물리현상에 바탕을 두어 생성되는 난수이고 완전히 랜덤한 숫자의 열을 실현할 수 있어, 영원히 무질서한 숫자의 열을 연속하여 발생한다. 차후에는 보안기반을 구축하기 위해 물리난수가 활용되는 사례가 증가할 것이라고 예상된다.

## PGP

Pretty Good Privacy로, 직역하면 '대단히 좋은 프라이버시'란 의미로 1991년에 필 짐머만(Phil Zimmermann)이 고안한 것으로 광범위하게 사용되는 암호소프트웨어이다.

PGP는 현대의 암호소프트웨어에 필요한 기능을 거의 모두 갖추고 있다. 즉, 공통키암호(AES, 3-DES 등), 공개키암호(RSA, 엘가말 등), 전자서명(RSA, DSA), 일방향해시함수(MD5, SHA-1, RIPEMD-160 등), 인증서 작성 등이 가능하다.

## SSL/TLS

온라인쇼핑 등을 할 때 사용되는 통신 프로그토콜(통신상에서의 규칙)에서 통신 내용의 인증과 무결성을 체크하기 위해서 메세지 인증코드를 사용한다. 예를 들면, 웹브라우저에서 신용카드번호 등을 보낼 때에 통신을 암호화하는 프로토콜로서 SSL(Secure Socket Layer) 또는 TLS(Transport Layer Security)를 준비하여 번호의 교환을 암호화하여 도청을 방지할 수 있다. 또한, SSL/TLS에 의한 통신에서의 URL은 http:*가 아니라 https:*로 시작한다.

또한, 메일을 송신하기 위해서의 SMTP(Simple Mail Transfer Protocol)나, 메일을 수신하기 위해서의 POP3(Post Office Protocol)라고 하는 프로토콜도 SSL/TLS로 암호화하여 수호할 수 있다.

## 양자암호

절대적으로 안전한 암호로 평가되고 있다. 보통 광통신의 1비트에 상당하는 펄스(pulse)에는 빛의 최소알갱이(광자)가 1만개 이상 포함되어 있다. 양자암호에서는 광자 1개에 1bit의 정보를 올려서 광자의 편광상태(전자파의 진동방향)로 0과 1을 구별한다. 이렇게 함으로써 광자를 분해하는 것은 불가능하고, 광자를 도난당한 경우에도 관측에 의해서 광자의 편광상태가 변하는 성질로부터 도난당한 것을 알 수 있다(양자역학에 의해서 담보되는 '도청의 불가능성'). 이 '도청의 불가능성'과 암호화키를 1회용으로 쓰는 '원타임 패드에 의한 해독의 불가능성'을 조합함으로써 절대 안전한 암호로 인식되어 실용화를 향한 연구가 가속화되고 있다.

## 생체인증

생체인증은 개인의 고유 정보(지문, 정맥, 얼굴, 홍채, 손바닥 모양, DNA 등)를 본인 확인에 이용하는 것이다. 쉽게 접할 수 있는 예로써, ATM(현금자동지급기)이나, 병원에서 입/퇴실할 때 본인 확인 등에 이용되는 정맥인증시스템이 도입되어 손가락이나 손바닥으로 본인 확인을 한다.

From:
https://theta5912.net/ - **reth**

Permanent link:
**https://theta5912.net/doku.php?id=public:computer:cryptology**

Last update: **2018/02/14 23:03**