Cryptanalysis
(암호분석)

Chapter 2 – Part 1

2020.3

Contents

- Probability
- Birthday paradox
- Euler totient function $\phi(n)$
- GCD and Euclidean algorithm
- Finite field $\mathrm{GF}(p^n)$

Probability

random variable

- **Probability** $P(X = x_i)$ of an occurrence x_i (event)
  - is defined by the likelihood that it(=x_i) happens

$$0 \leq P(X = x_i) \leq 1$$

never happens

always happens

  - Examples
    - (fair coin): $P(X = \mathrm{HEAD}) = P(X = \mathrm{TAIL}) = \frac{1}{2}$
    - (jack diamond): $P(X =$  $) = \frac{1}{52}$

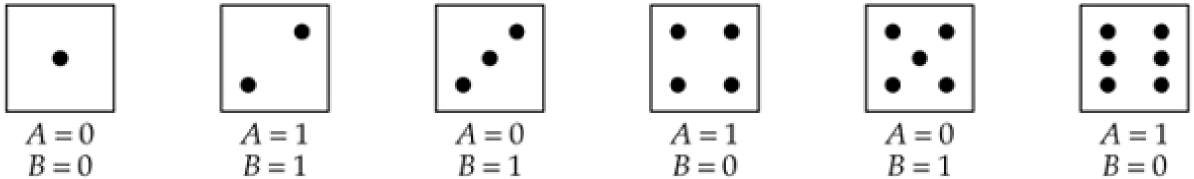
- Permutations and Choices
  - the number of (all) **Permutations** of n objects: $n!$
  - the number of permuted choices (k out of n): $\frac{n!}{(n-k)!}$
  - Binomial coefficients (n **choose** k): $\binom{n}{k} = \frac{n!}{(n-k)!k!}$

Dependence

- Independent and dependent events
  - A, B are **independent** events if one event has occurred does not affect the probability that the other event will occur.
$$P(A|B) = P(A)$$
  - Otherwise, A, B are said to be **dependent** events.

- Example (Dice game)
  - event A: Alice gets a point for even numbers
  - event B: Bob gets a point for prime numbers
  - Events A and B are dependent (why?)

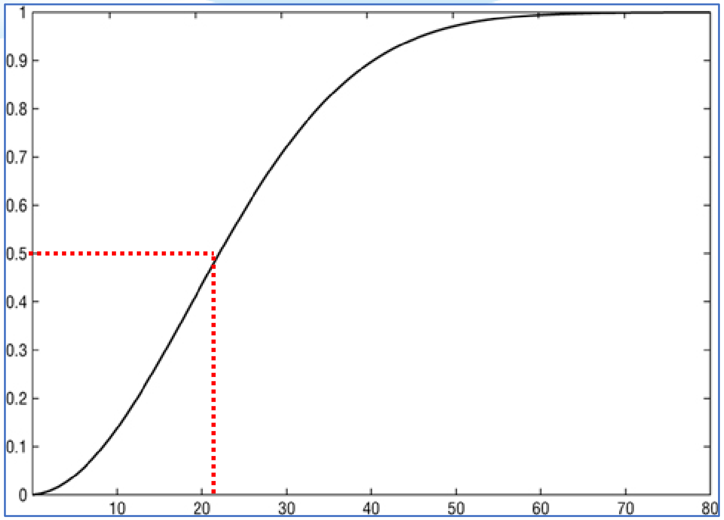
Calculate probabilities
$P(A), P(B), P(A|B), P(B|A)$



Birthday collision

- Birthday collision
  - Birthday collision of two people occurs with probability 1/365.
  - The probability P that there are no birthday collisions with k people is $e^{-k(k-1)/2n}$ $(n = 365)$
  - No collision probability P:
    $$P = \frac{n(n-1)(n-2)\cdots(n-k+1)}{n^k} = \left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{k-1}{n}\right)$$
  - Taylor series for approximation:
    $$e^{-1/n} \approx 1 - \frac{1}{n}$$
  - Approximate P:
    $$P = e^{-1/n}e^{-2/n}\cdots e^{-\frac{k-1}{n}} = e^{-k(k-1)/(2n)}$$

Birthday paradox



- Birthday paradox
  - There is at least one birthday collision among 23 or more people with probability more than 0.5.
  - Condition that no collision probability $P = e^{-k(k-1)/(2n)} = ½$:
  - $\ln\left(\frac{1}{2}\right) = -\frac{k(k-1)}{2n}$ implies $k(k-1) = n\, 2\ln 2$
  - Approximately, $k(k-1) = k^2$ yields $k \approx 1.1774\sqrt{n}$
  - If $k > 1.1774\sqrt{365} \approx 22.49$, a birthday collision among k people is found with probability ≥ ½.

Roughly, $k = \sqrt{n}$

Collisions in Cryptographic Algorithms

- Collision in an n-bit Block cipher
  - We only have to look at about $2^{n/2}$ ciphertexts to expect to find the pattern twice.

- Collision in hashes
  - preimage attack: (2^n complexity)
    find an input message knowing only its hash value
  - collision attack: ($2^{n/2}$ complexity)
    find random messages that have the same hash value

Integers

- Congruence
  - $a \equiv b \ (mod\ m)$: a is congruent to b modulo m if m divides $(a-b)$
  - $a \equiv b \ (mod\ m)$ is equivalent to $a = b + km$
- Residue class
  - A set of all congruent to each other modular m is called its residue class.
  - example: $\{\dots, -6, -3, 0, 3, 6, \dots\}$ is a residue class modulo 3
- Complete Set of Residues(CSR)
  - contains integers exactly one from each residue
  - example: $\{0,1,2,\dots,9\}$ is CSR of 10
- Reduced Set of Residues(RSR)
  - subset of CSR containing only numbers relatively prime to m
  - example: $\{1,3,7,9\}$ is CSR of 10

Euler's totient theorem

- Euler totient function $\phi(n)$
  - the size of the RSR
  - equivalently, the number of positive integers less than n that are relatively prime to n.
  - examples: $\phi(5) = 4,\ \phi(6) = 2,\ \phi(10) = 4$
- Euler's theorem (Fermat's little theorem)
  - m: positive integer
  - a: integer relatively prime to m $(1 \leq a < m)$

then,

$$a^{\phi(m)} \equiv 1 \pmod{m}.$$

GCD

- GCD: Greatest Common Divisor
  - For nonnegative integers a, b,
    $g = \gcd(a, b)$ is the largest common divisor of a and b

  - Properties of GCD:
    - If $b \neq 0$, $\gcd(a, b) = \gcd(b, a \ mod \ b)$
    - If $b = 0$, $\gcd(a, b) = a$

$$\boxed{a = b \cdot q + r \ (0 \leq r < b)}$$

$$\boxed{\gcd(a, b)} = \boxed{\gcd(b, r)}$$

GCD function

$$a = b \cdot q + r \; (0 \leq r < b)$$

$$\gcd(a, b) = \gcd(b, r)$$

- Euclidean Algorithm

```
9 = gcd(63, 90)

 63 = 90 * 0 + 63
 90 = 63 * 1 + 27
 63 = 27 * 2 + 9
 27 =  9 * 3 + 0
  9 =  0 * ? + ?
```

a b r

```
def gcd(a,b):

    while b != 0:
        r = a % b
        a = b
        b = r

    return a
```

Extended Euclidean Algorithm

- Extended Euclidean Algorithm
  - There exist integers x and y such that

$$ax + by = \gcd(a, b)$$

```
def gcd(a,b):

    while b != 0:
        r = a % b
        a = b
        b = r

    return a
```

Initially,

$$a = 1 * a + 0 * b,$$
$$b = 0 * a + 1 * b.$$

For each step,

$$a = u_a * a + v_a * b,$$
$$b = u_b * a + v_b * b.$$

Finally,

$$\gcd(a, b) = a = u_a * a + v_a * b.$$

input parameter a, b

updated final value of a

Extended Euclidean Algorithm

- In the Euclidean algorithm,
  - Store Input value a, b (keep them unchanged)
  - Set $a' = a,\ b' = b$
  - Initially,

$$a' = u_a * a + v_a * b = 1 * a + 0 * b$$
$$b' = u_b * a + v_b * b = 0 * a + 1 * b$$

  - Update values as

$$a' = b' = u_b * a + v_b * b$$
$$b' = r = a' - b' \cdot q$$
$$= u_a * a + v_a * b\ u_b * a - q * (u_b * a + v_b * b)$$
$$= (u_a - u_b q) * a + (v_a - v_b q) * b$$

  - Finally, when $b' = 0$, the algorithm returns

$$a' = \gcd(a, b) = u_a * a + v_a * b$$

Initially,

$u_a = 1,\ v_a = 0,$
$u_b = 1,\ v_b = 0.$

Updating...

$u_a = u_b,$
$v_a = v_b,$
$u_b = u_a - u_b q,$
$v_b = v_a - v_b q.$

Extended Euclidean Algorithm

```
def xgcd(a,b):

    an, bn = a, b
    ua, va, ub, vb = 1, 0, 0, 1

    while bn != 0:
        q = an//bn
        an, bn = bn, an-bn*q
        ua, va, ub, vb = ub, vb, ua-ub*q, va-vb*q

    return an, ua, va
```

반드시 필요하지 않지만
변수를 새로 만들면 중간과정을 인쇄하여
확인하기 좋다.

Python에서는 한번에 어려 변수를 업데이트
할 수 있어 편리하다

이렇게 중간과정이 출력되로록
함수를 고쳐보자!

```
37 = 1*37 + 0*41,  41 = 0*37 + 1*41
41 = 0*37 + 1*41,  37 = 1*37 + 0*41
37 = 1*37 + 0*41,  4 = -1*37 + 1*41
4 = -1*37 + 1*41,  1 = 10*37 + -9*41
gcd(37,41) = 1 = 10*37 + -9*41
```

Multiplicative Inverse

- Multiplicative inverse
  - For a and m with $\gcd(a, m) = 1$,
    x is called the multiplicative inverse of a modulo m if
    $$ax = xa = 1 \ (mod\ m)$$
  - The multiplicative inverse of a is denoted by a^{-1}.

- Algorithm for calculating a^{-1}
  - Using the Extended Euclidean algorithm, find integers x and y such that
    $$ax + my = \gcd(a, m) = 1.$$
  - Then $a^{-1} = x \ mod\ m$ is the multiplicative inverse of a.

Multiplicative Inverse

```
def modInv(a,m):
    if gcd1(a,m) != 1:
        return None
    c, d = a, m
    uc, vc, ud, vd = 1, 0, 0, 1
    while c != 0:
        q = d//c
        c, d = d-q*c, c
        uc, vc, ud, vd = ud-q*uc, vd-q*vc, uc, vc
    return ud % m
```

Extended Euclidean algorithm과 거의 같다.

Finite Field

- A finite field is a field which consists of finite elements.
  - Example 1: $(Z_p,\ +,\ \times)$ (p: prime) is also called $\mathrm{GF}(p)$.
    $$Z_p = \{0, 1, 2, \dots, p-1\}$$
    - Z_p satisfies all conditions for field.
    - For any x in Z_p, $-x \pmod p$ is the additive inverse of x
    - For $x \neq 0$ in Z_p, $x^{-1} \pmod p$ is the multiplicative inverse of x
  - Example 2: Galois field $\mathrm{GF}(p^n)$ with p^n elements.
    - A set of polynomials of degree $n-1$ with multiplication modulo $m(x)$ (irreducible polynomial of degree n) having coefficients in $\mathrm{GF}(p)$.

Example $\mathrm{GF}(2^3)$

- $\mathrm{GF}(2^3)$ consists of 2^3 elements of the form

$$\mathrm{GF}(2^3) = \{a_0 + a_1x + a_2x^2 \mid a_i = 0, 1\}.$$

  - Their coefficients belong to the finite field $\mathrm{GF}(2) = \{0, 1\}$.
  - Choose an irreducible polynomial $m(x) = x^3 + x + 1$ in $\mathrm{GF}(2)[x]$.
  - Multiplications are defined as modulo $m(x)$.
    eg., $(x^2 + 1) \cdot (x^2 + 1) = x^4 + 1 = x \cdot m(x) + x^2 + x + 1$
  - The multiplicative inverse can be found as in the following table.

$\times$	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
0	0	0	0	0	0	0	0	0
1	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
x	0	x	x^2	x^2+x	$x+1$	1	x^2+x+1	x^2+1
$x+1$	0	$x+1$	x^2+x	x^2+1	x^2+x+1	x^2	1	x
x^2	0	x^2	$x+1$	x^2+x+1	x^2+x	x	x^2+1	1
x^2+1	0	x^2+1	1	x^2	x	x^2+x+1	$x+1$	x^2+x
x^2+x	0	x^2+x	x^2+x+1	1	x^2+1	$x+1$	x	x^2
x^2+x+1	0	x^2+x+1	x^2+1	x	1	x^2+x	x^2	$x+1$