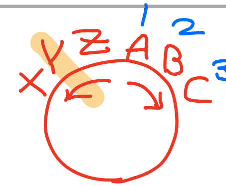
고전암호와 한계

시저 암호의 일반화 (덧셈 암호)



- 시저 암호(Caesar cipher)와 아우구스투스 암호(Augustus cipher)
  - 시저 암호: 메세지의 각 글자를 알파벳에서 세 자리 순환시켜서 암호화
  - 아우구스투스 암호: 메세지의 각 글자를 알파벳에서 키 값에 해당하는 자리 수 만큼 순환시켜서 암호화
- 시저 암호의 결함
  - 케르크호프스의 원리 (Kerckhoff's principle): 한 암호의 보안성은 오직 키의 비밀성에만 의존해야 하며, 암호체계의 비밀성에 의존해서는 안됨
  - But, 우리가 어떤 사람이 시저암호를 사용한다는 정보를 얻었다면, 그 사람의 암호 체계 전체를 파악할 수 있음
  - 복호화와 암호와에 키가 반드시 필요한 체계를 사용하면, 안전성 향상이 가능
    - 암호의 일반 체계를 알아내더라도 키가 없으면 메세지를 쉽게 읽을 수 없음
  - 아우구스투스 암호의 경우에도 25개의 키를 시도 시에 복호화 가능

알파벳 → 26글자

- 덧셈 암호 (additive cipher)

알파벳 → 숫자

  - 모듈로 연산 (modular arithmetic): 순환 개념을 사용하여 특정 정수로 나누었을 때 나머지로 합동을 정의하는 연산

1=1 2=2

13=1

    - $10+3 \equiv 1 \bmod 12$

25개 1 26

  - 시저 암호와 아우구스투스 암호는 알파벳 문자를 숫자로 대응시켰을 때 키 만큼의 합의 26-모듈로 연산으로 암호화 하는 것임

곱셈암호 기본

- 데시메이션 기법 (decimation method)
  - 키를 3이라고 하면, 평문의 알파벳을 나열한 상태에서 시작
    - a b c d e f g h i j k l m n o p q r s t u v w x y z
    - C F I L O R U X
  - 문자열 끝까지 돌아오면 다시 앞부분으로 돌아옴
    - C F I L O R U X A D G J M P S V Y
    - C F I L O R U X A D G J M P S V Y B E H K N Q T W Z
  - 최종적인 대응관계는 다음과 같음 (수와 함께 표시)

key = ☆

Enc(X) = ☆ × X mod 26

Enc(X) = ☆ + X mod 26

1~26

27 mod 26

30

평문	a	b	c	d	e	f	g	h	i	j	…	x	y	z
숫자 ×3	1	2	3	4	5	6	7	8	9	10	…	24	25	26
연산 결과	3	6	9	12	15	18	21	24	1	4	…	20	23	26
암호문	C	F	I	L	O	R	U	X	A	D	…	T	W	Z

곱셈암호 일반화

- 곱셈 암호 (multiplicative cipher)
  - 앞에서 제시한 데시메이션 기법은 평문 숫자에 3을 곱하고 26에 이르렀을 때, 앞으로 돌아가는 모듈로 곱셈 연산으로 생각할 수 있음
    - 모듈로 연산은 26을 반복적으로 빼는 것으로 생각할 수 있지만, 이것은 26으로 나눈 나머지를 보는 것과 같음
  - 모듈로 곱셈 연산
    - 모듈로 곱셈 연산이란 곱셈의 결과를 모듈로 동치로 나타내는 것을 의미함
- 곱셈 암호의 키
  - 만약 26을 키로 사용한다면?
    - 모듈로가 26일때 26을 곱하는 것은 0을 곱하는 것과같음 (26 ≡ 0 mod 26)
    - 이 경우에는 모든 문자를 Z 로 치환하게 됨
  - 2를 키로 사용한다면?
    - 26개의 문자가 13개의 문자로 치환되어 복호화할 수 없게 됨
    - 모든 다른 짝수 키도 마찬가지
  - 일단 지금까지 찾은 나쁜 키는 짝수 키들인 13개임
    - 최종적으로 자명한 키 1을 포함하여 좋은 키는 12개임, **why?**
      - 자명한 키 (trivial key): 평문과 암호문이 같은 키
      - 좋은 키: 26개의 문자를 서로 다른 문자로 치환할 수 있도록 하는 키

27 − 26 = 1

1~26 (26개)

13개 → 13개 ↗ 13개는

0 mod 26

0 = Z = 26

※ = 26

A B C D ...

1 2 3 4 ...

Z Z Z Z

ZZZZ ? → 평문!

2 26 서로소가 아닌 공약수 존재 ※ = 2

1 2 3 4 ... 13 14

2 4 6 8 ... 26 28 ...

$\underbrace{2 \ 4 \ 6 \ 8 \ \cdots \ 26}_{13개 문자}$

28 = 2

덧셈※ = 0

곱셈※ = 1

$Enc(X) = ✱ \times X \bmod 26$

곱셈 암호의 복호화 [1/2]

$X + ✱ - ✱$

$X \times ✱ \times \frac{1}{✱}$

- 곱셈 암호의 복호화 방법

$Enc(X) = ✱ + X \bmod 26$

$Dec(Y) = Y - ✱ \bmod 26$

  - 덧셈 암호에서는 복호화 할 때에 각 문자를 키에 해당하는 자리 만큼 왼쪽으로 이동하였음 → 모듈로 뺄셈으로 생각 가능
  - 곱셈암호에 대해서는 곱셈을 역으로 수행하는 나눗셈 연산을 생각할 수 있음
    - 예를들어, 암호문 문자 C 는 숫자 3 이고 3/3 = 1 이므로 이는 평문 a에 해당됨
    - 하지만, 암호문 문자 B의 경우에는 숫자 2이므로 2/3에 해당하는 문자가 없음
  - 모듈로 연산을 이용해서 합동인 숫자를 이용해 나눗셈 수행 가능
    - 2 ≡ 28 ≡ 54 mod 26
    - 2 ÷ 3 ≡ 54 ÷ 3 ≡ 18 mod 26
    - 따라서, B 를 r 로 복호화 할 수 있음

$Enc(10) = 3 \times 10 = 30 = 4 \bmod 26$

$4/3 =$ 정수! ⇒ 10?!

  - 효율성의 측면에서는 이렇게 문자에 대응되는 모듈로 합동인 숫자를 찾는 것은 비효율적임
    - 예를 들어, 키가 15인 경우 B는 2, 28, 54, 80, 106, 132, 158, 184, 210 이고, 15로 나누었을 때 문자에 대응되는 수는 210 뿐임

$4 \div 3 = 30 \div 3 = 10 \bmod 26$

$\boxed{18} \times 3 = 2 \bmod 26$

곱셈 암호의 복호화 [2/2]

$2 \times \langle x \rangle = 1$

$\frac{1}{2}$

$Enc(X) = X \times ☆ \bmod 26$

$Dec(Y) = Y \times \langle ☆ \rangle \bmod 26$

$Y = X \times ☆ \bmod 26$

$Y \times \langle ☆ \rangle = X \times \boxed{☆ \times \langle ☆ \rangle} \bmod 26$

$= 1$

$= X \bmod 26$

- 모듈로 곱셈에 대한 역원
  - 일반 곱셈에 대해서 a의 역원 <a> 는 다음을 만족
    - a × <a> = <a> × a = 1 → $\frac{1}{a}$ $(a \neq 0)$
  - 모듈로-q 곱셈에 대한 a 의 역원도 유사하게 정의할 수 있음
    - a × <a> ≡ <a> × a ≡ 1 mod q
  - 곱셈 암호의 복호화에 모듈로 곱셈에 대한 역원을 사용할 수 있음
    - 예를들어, 키가 3일 때 모듈로 동치인 수에 대해 3으로 나눗셈을 수행하여 얻는 정수 값과, 곱셈에 대한 역원을 모듈로 곱셈한 결과 값이 같음
    - 곱셈 암호에서 3에 대한 모듈로-26 역원은 9임: 3 × 9 ≡ 1 mod 26
    - B를 복호화 할 경우 **2** ÷ **3** ≡ 54 ÷ 3 ≡ 18 ≡ **2** × **9** mod 26
  - 곱셈 암호의 키가 k일 때, <k>가 존재한다는 것을 확신할 수 있을까?
    - 곱셈 암호에서 **나쁜키가 아닌 키**들에 대한 역원 <k>는 항상 존재
    - 모듈로-q의 곱셈에서 q와 서로소인 키 k의 역원 <k>는 항상 존재
    - 따라서, 곱셈 암호에서 나쁜키는 26과 1이 아닌 **공약수**가 존재하는 키임

$\Leftrightarrow$ 역원이 존재하지 X

$\boxed{\bmod 26}$

<k>

k와 26 서로소!

<k> 존재

유클리드 호제법

- 유클리드 호제법 (Euclidean algorithm)
  - 최대공약수를 찾는 효율적인 알고리즘
    - 유클리드 호제법에서 각 단계는 정수인 몫과 나머지가 있는 나눗셈이며, 이전 단계의 몫을 나머지로 나눔
    - 최종적으로 0이 아닌 마지막 나머지가 최대공약수가 됨
  - 예) 756과 210의 최대공약수 구하기
    - 756 = 3*210 + 126 → 210 = 126*1 + 84 → 126 =84*1 + 42 → 84 = 42 * 2 +0
    - 최종적으로 최대공약수는 42가 됨
  - 최대 공약수를 찾을 때 유클리드 호제법을 사용하는 이유
    - 1. 큰 수의 경우에는 두 수를 각각 소인수분해 해서 공통의 소인수를 찾는 것보다 유클리드 호제법이 더 빠름
    - 2. 유클리드 호제법을 잘 활용하면, 모듈러 곱셈에 대한 역원을 쉽게 찾을 수 있음
- 모듈러 곱셈에 대한 역원
  - 1을 3 * s 와 26 * t 꼴의 두 항의 합으로 표현해 보도록 함
  - 26 =3 * 8 + 2 → 2 = 26 - 3 * 8 → 2 = 26 * 1 - 3 * 8
  - 3 = 2 * 1 +1 → 1 = 3 - 2 * 1 → 1 = 3 - (26 * 1 - 3 * 8) * 1 → 1 = -(26 * 1) + (3 * 9)
  - 1 ≡ -(26 * 1) + (3 * 9) ≡ -(0 * 1) + 3 * 9 ≡ 3 * 9 mod 26
  - 따라서, <3> = 9

$1 = 3 \times s + 26 \times t$

$1 = 3 \times s \bmod 26$

$36 = 3^2 \times 2^2$, $24 = 2^3 \times 3$, $2^2 \times 3 = 12$

서로소 항상 1

$26 = 3 \times 8 + 2$

$2 = 26 \times 1 - 3 \times 8$

$3 = 2 \times 1 + 1 \Rightarrow 1 = 3 - 2 \times 1$

$1 = 3 \times 1 - 1(26 \times 1 - 3 \times 8)$

$= 26 \times (-1) + 3 \times 9$

$1 = 26 \times (-1) + 3 \times 9$ = s

$1 = 3 \times 9 \bmod 26$

아핀 암호

- 무차별 대입 공격 (Brute-force attack)
  - 모든 가능한 키를 조사해 복호화할 수 있는 키를 찾아내는 방법
    - 덧셈암호는 26개의 좋은 키가 있고, 곱셈 암호는 12개의 좋은 키가 있음 (자명한 키 포함)
  - 덧셈 암호, 곱셈 암호 중 어느 암호를 사용하더라도 무차별대입 공격을 쉽게 감행할 수 있음
- 둘 이상의 암호를 동시에 사용한다면? (k와 m을 키로 사용)
  - 키가 k인 덧셈 암호: C ≡ P + k *mod 26* → P ≡ C - k *mod 26* 26(25)
  - 키가 k인 곱셈 암호: C ≡ kP *mod 26* → P ≡ <k>C *mod 26* 12(11)
  - 1. 덧셈 암호를 두 번 적용
    - C ≡ P + k + m *mod 26* X
    - 공격자의 입장에서 키 값이 $\boxed{k+m}$인 덧셈 암호를 사용한 것과 같음
  - 2. 곱셈 암호를 두 번 적용
    - C ≡ kmP *mod 26*

  $k(P+m) = kP + \boxed{km} = q$

  - 3. 곱셈 암호와 덧셈 암호를 하나씩 동시에 사용 : **아핀 암호 (affine cipher)**
    - C ≡ kP + m *mod 26* → P ≡ <k>(C - m) *mod 26*
    - k의 가능한 값은 12이고, m의 가능한 값이 26개이기 때문에 총 12 * 26 = 312 개의 암호키가 존재

$C = kP + m$

$Dec(C) = \langle k \rangle (C - m)$
$= \langle k \rangle (kP + m - m)$
$= \underbrace{\langle k \rangle kP}_{1}$
$= P$

$Enc(P) = kP + m \bmod 26$

$Dec(C) = \langle k \rangle (C - m) \bmod 26$

곱암호

- 곱암호 (product cipher)
  - 데시메이션 기법(곱셈암호)과 이동 암호(덧셈 암호)를 결합한 암호 기법
  - 아트바시 암호 (atbash cipher) 아핀암호의 일종
    - 암호문 알파벳은 평문 알파벳을 역순으로 적어 놓은 것과 같음
    - 아래 표로부터 다음의 암호화 규칙을 찾을 수 있음
    - C ≡ 27 - P *mod 26* → C ≡ (-1)P + 27 *mod 26* → C ≡ 25P + 1 *mod 26*
    - 따라서 이는 k=25, m =1 인 kP + m 형태의 아핀 암호임

k=25

m=1

$Enc(P) = 27 - P = (-1) \cdot P + 27 \bmod 26$

$= (-1)P + 1 = 25P + 1 \bmod 26$

평문	a	b	c	d	e	f	g	h	I	j	…	x	y	z
숫자	1	2	3	4	5	6	7	8	9	10	…	24	25	26
연산 결과	26	25	24	23	22	21	20	19	18	17	…	3	2	1
암호문	Z	Y	X	W	V	U	T	S	R	Q	…	C	B	A

$25 \times 2 + 1 = 51 \bmod 26$

$= 25$

다중문자 치환 암호 [1/4]

P C

ja $\xrightarrow{k_1, k_2}$ $\xrightarrow{k_3, k_4}$

P_1 P_2 C_1 C_2

- 다중문자 치환 암호 (polygraphic substitution cipher)
  - 한 번에 복수의 문자를 치환하는 암호
  - 다중문자 치환 방식을 위해서는 블록의 크기를 정해 문자가 몇 개 단위로 치환되는 지를 결정
    - 다이어그래프(블록크기가 2인 암호), 트라이그래프(블록크기가 3인 암호), …
  - 다이어그래프는 16세기에 제안되었으나 19세기 들어서야 사용됨
- 힐 암호 (Hill cipher)
  - 1929년 레스터 힐 (Lester S. Hill)이 어떤 블록 크기에 대해서도 사용할 수 있는 힐 암호를 발명
  - 블록 크기가 2인 예:

    Ja/ck/y an/d/ Ji/ll/y an/d/ ev/ex

    - 먼저, 평문을 두 문자 씩 나누고 마지막 블록에 빈 공간이 있으면 임의의 문자로 채움
    - 이 때, 임의의 문자를 무효 문자 (Null) 혹은 채움 문자 (padding) 이라 함
    - ja ck ya nd ji ll ya nd ev ex → 현재 예에서는 x 가 채움 문자로 가정
    - 평문 블록의 첫 번째 문자를 P1, 두 번째 문자를 P2라 할 때 다음과 같은 공식으로 암호 문자 계산
      - $C1 \equiv k1*P1 + k2*P2 \ mod\ 26$; $C2 \equiv k3*P1 + k4*P2 \ mod\ 26$
      - k1,k2, k3, k4 는 1에서 26까지 수 중에서 선택하여 모두 다 키를 구성함

$C_1 = 3 \times 10 + 5 \times 1 = 35 = 9$

$C_2 = 6 \times 10 + 1 \times 1 = 61 = 9$

다중문자 치환 암호 [2/4]

- 힐 암호의 암호화
  - 예를 들어, k1 = 3, k2 = 5, k3 = 6, k4 = 1 이라고 하면 다음과 같은 암호화 공식을 얻을 수 있음
    - C1 ≡ 3*P1 + 5* P2 *mod 26*; C2 ≡ 6*P1 + 1*P2 *mod 26*
  - 앞에 주어진 평문을 다음과 같이 숫자로 전환하여 암호화 가능
    - 같은 문자라도 서로 다른 문자로 치환 되지만 (예: j, l), 같은 블록은 같은 문자들로 치환됨 (예: ya, nd)
- 힐 암호의 복호화

  $Enc(P) = Kp \cdot \bmod 26.\ Dec(C) = \langle k \rangle p \bmod 26$
  - 받은 메세지를 복호화 하기 위해서는 미지수가 두 개인 연립방정식을 풀어야 함

    존재!

    $\langle k_1 k_4 - k_2 k_3 \rangle$

    block 크기 2일 경우
    - C1 ≡ k1*P1 + k2*P2 *mod 26*; C2 ≡ k3*P1 + k4*P2 *mod 26*
  - k1*k4 - k2*k3 의 곱셈에 대한 역원이 존재한다면, 복호화 식은 다음과 같이 주어짐
    - P1 ≡ <k1*k4 - k2*k3>(k4*C1 - k2*C2) *mod 26*; P2 ≡ <k1*k4 - k2*k3>(k1*C2 - k3 C1) *mod 26*
- 힐 암호의 좋은키: 암호화 연립방정식의 해가 유일하게 존재하여 복호화가 가능한 키
  - 블록 크기가 2인 경우는 대략 45,000개, 블록 크기가 3인 경우는 대략 52,000,000,000 개 존재
  - 무차별 대입 공격이 어려움

평문	ja	ck	ya	nd	ji	ll	ya	nd	ev	ex
숫자	10,1	3,11	25,1	14,4	10,9	12,12	25,1	14,4	5,22	5,24
힐 암호	9,9	12,3	2,21	10,10	23,17	18,6	2,21	10,10	21,0	5,2
암호문	II	LC	BU	JJ	WQ	RF	BU	JJ	UZ	EB

다중문자 치환 암호 [3/4]

- 힐 암호의 복호화 (continued)
  - 블록 크기가 2인 경우의 힐 암호를 임의의 블록 크기로 일반화 할 경우 복호화 시에 블록크기 만큼의 미지수를 가지는 연립방정식이 됨

    3개 이상

  - 크래머 규칙 (Cramer's Rule)

    $m_1 = \langle 3\times 1 - 5\times 6\rangle \cdot 1$

    $= \langle 3-30\rangle \cdot 1 = \langle 25\rangle$

    -27

    - 일반적인 연립방정식의 해를 구하는 해법
    - 임의의 블록 크기의 힐 암호를 해독하기 사용될 수 있음
    - 연립방정식을 정사각 행렬의 행렬식들로 나타내어 계산함
  - 앞의 복호화 공식을 간단히 하기 위해서 복호화 식의 C1, C2의 계수들을 다음과 같이 나타냄
    - m1 = <k1*k4 - k2k3> (k4) ; m2 = <k1*k4 - k2k3> (- k2) ; m3 = <k1*k4 - k2k3> (- k3) ; m4 = <k1*k4 - k2k3> (k1)
  - m1, m2, m3, m4 를 이용해서 나타낸 복호화 식은 다음과 같음
    - P1 ≡ m1 * C1 + m2 * C2 mod 26 ; P2 ≡ m3 * C1 + m4 * C2 mod 26
    - 위의 연립 방정식은 암호화 시의 연립 방정식을 거꾸로 나타낸 역의 개념으로 생각 가능
    - 이 때, m1, m2, m3, m4 를 키 k1, k2, k3, k4 의 '역키 (inverse key)' 라고 함
    - 앞의 예에서 키 k1, k2, k3, k4 = (3, 5, 6, 1)의 역키는 ? **m1, m2, m3, m4 = (25, 5, 6, 23)**

다중문자 치환 암호 [4/4]

- 아핀 힐 암호 (affine Hill cipher)
  - 곱셈 암호와 덧셈 암호를 결합해 아핀 암호를 만들었던 것처럼 힐 암호에 덧셈을 추가한 방식의 암호
  - 블록 크기를 2로 가정하면, 새로운 암호의 암호화 방정식은 다음과 같음
    - C1 ≡ k1*P1 + k2*P2 + m1 mod 26
    - C2 ≡ k3*P1 + k4*P2 + m2 mod 26
    - 키가 (k1, k2, k3, k4, m1, m2)의 총 6개로 이루어졌고, 1~26까지 중에서 선택된 수임
    - 아핀 힐 암호에서도 k1*k4 - k2*k3와 26 의 최대공약수가 1이면 (모듈로 곱셈에 대한 역원이 존재하면), 좋은 키가 되며, m1, m2에 대해서는 별도의 조건이 없음

1~26

다표식 치환 암호

- 비즈네르 암호 (Vigenere cipher)
  - 16세기 지오반 바티스타 벨라소 (Giovan Battista Bellaso)라는 이탈리아 인이 만듦
    - 역사 상의 인용 오류 때문에 다른 암호를 만든 프랑스인 블레즈 드 비즈네르 (Blaise de Vigenere)의 이름으로 굳혀짐
  - 다표식 치환 암호 (polyalphabetic substitution cipher): 메세지 속 문자의 위치에 따라서 치환 규칙이 달라지게 하는 암호 방식
    - 비즈네르 암호는 다표식 치환 방법 중 하나임
    - 한자리 이상의 키로 정의되는 값들 만큼 글자들이 이동하여 암호화 수행
    - 예를들어, 키가 DUH라면, 평문의 글자들을 순서대로 3, 20, 7 자리 이동함
    - 3, 20, 7 자리 이동은 평문의 모든 글자가 암호화 될 때까지 반복됨
  - 비즈네르 암호는 시저 암호보다는 훨씬 안전하지만, 키의 길이를 알아내고 나면 깨기 쉬워 짐

평문	b	e	l	l	a	s	o	m	a	d	e	i	t
숫자	2	5	12	12	1	19	15	13	1	4	5	9	20
연산 결과	5	25	19	15	21	26	18	7	8	7	25	16	23
암호문	E	Y	S	O	U	Z	R	G	H	G	Y	P	W

정리

- 고전 암호는 오늘날 관점에서 보면 시대에 뒤쳐진 암호임
- 고전 암호는 알파벳을 대상으로 하지만, 오늘 날의 암호는 숫자, 그림, 음향 등 다양한 형태의 정보를 암호화함
- 덧셈 암호나 곱셈 암호는 키가 충분히 많지 않아 무차별 대입 공격에 쉽게 무너짐
- 이들 암호를 포함한 모든 단순 치환 암호는 빈도 분석법에 취약함
- 힐 암호와 아핀 암호 둘 다 알려진 평문 공격에 약함
- 하지만, 이 두 암호는 미국 정부의 블록 암호 표준을 포함해 현대 블록 암호의 기본 구성요소로 사용됨

알파벳-숫자 대응표

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26