Foundations of computer security

Norbert Oláh

2022.

Content

Substitution cypher

- Caesar cypher
  - plain text (m)
  - key (k)
  - cyphertext $c = (m + k) mod\ 26 \rightarrow$ english abc
  - decripted text $m = (c - k) mod\ 26$
- Keyword Caesar encryption
  - plain text (m)
  - key (k=8 security)
    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
    O P Q V X Y Z S E C U R I T Y A B D F G H J K L M N
  - comparing to the shifted alphabet it contains significantly more variations (n!, where n is the length of alphabet)
  - cannot be sabotaged by rotation

Affin cypher

- Encrypting by letters
- The key is a pair of numbers K=(a,b) $\rightarrow$ GCD(a,26)=1
- $c = (a * m + b) mod\ 26$
- $m = (a_1 * c - a_1 * b) mod\ 26$ where $a_1 * a = 1 mod\ 26$

Residue class

If $a \in Z$, then the set of integers (mod m) congruent with a its called the (mod m) residue class represented by a.
Notation: $(a)_m$
Z_n it's a kind of set which elements are residue classes
Z_6=$(0)_6, (1)_6, (2)_6, (3)_6, (4)_6, (5)_6,$

Residue class properties

- $(a)_m + (b)_m = (a + b)_m$
- $(a)_m * (b)_m = (ab)_m$
- if we implement a given operation containing any two set of residue class the result of the operation still remain within the set.
- Example:
  $(2)_6 + (5)_6 = (2 + 5)_6 = 1_6$ $(3)_6 + (3)_6 = 0_6$ $(4)_6 + (5)_6 = 3_6$
  $(2)_6 * (5)_6 = 4_6$
- They forming algebraic structure.

Algebraic structures

There is given an $S = (x, y, z...)$ set and within this set an operation have been defined. (usually addition and multiplication)

- Semigroup
- Group
- Abelian group
- Ring
- Field

Addition and multiplication as the Characteristic of algebraic structures +, ·

- Associative
  The set of residue classes inherits the associative property of integers.
  Example: $(xoy)oz = xo(yoz)$.
- Commutative
  The set of residue classes inherits the commutative property of integers
  Example: $xoy = yox$.
  The addition and multiplication are both commutative among integers.

Characteristic of algebraic structures +, ·

- J neutral element (identity)
  We call any e element of S identity element or neutral element, if any cases of $x \in S$:
  e o x = x o e = x
  + $(0)_m$ the neutral element that is $(a)_m+(b)_m=(0)_m$
  · $(1)_m$ the neutral element that is $(a)_m \cdot (b)_m=(1)_m$
- J inverse
  If e identity exists in S and if such y element belongs to x, that
  x o y = y o x = e,
  then we call y the inverse of x.
  In the set of real numbers the inverse of 2 is 1/2.

Characteristic of algebraic structures +, ·

- Distributive
  The set of residue classes inherits the distributive property of integers
  $x\Delta(y+z) = x\Delta y + x\Delta z$
  and
  $(y+z)\Delta x = y\Delta x + z\Delta x$
  Among integers, the multiplication is distributive relating to the addition.

We call an $S = x, y, ...$ set an algebriac structure if there is at least one operation is being defined in it.

Groups

- Semigroup
  An S set is semigroup if the **associative** property is being defined in it.
- Group
  An S set is group, if the **associative** property is being defined furthermore the neutral or identity elements exist and every element has its inverse.
- Abelian group
  An S set is an Abelien group, if the **associative and commutative** property is being defined in it furthermore the neutral or identity elements exist and every element has its inverse.

Group

- Ring

Ring	$(Z_m$ base set;+)	$(Z_m$ base set;·)
	associative	associative
	commutative	distributive
	J identity	commutative (ring with identity)
	J inverse	J identity (ring with identity)

- test

Field	$(Z_m$ base set;+)	$(Z_m$ base set;·)
	associative	associative
	commutative	commutative
	J identity	J identity
	J inverse	J inverse
		distributive

Exercise

- $Z_6 = (0)_6, (1)_6, (2)_6, (3)_6, (4)_6, (5)_6,$
- $Z_5 = (0)_5, (1)_5, (2)_5, (3)_5, (4)_5$

Exercise

Field	(Z_6 base set;+)	(Z_6 base set;·)
	associative	associative
	commutative	commutative
	J identity	J identity
	(0_6)	
	J inverse	J inverse
	(0)6 ->(0)6	(1)6 ->(1)6
	(1)6->(5)6	(2)6->no
	(2)6 ->(4)6	(3)6->no
	(3)6->(3)6	(4)6->no
	(4)6->(2)6	(5)6->(5)6
	(5)6->(1)6	

Exercise

Field	(Z_5 base set;+)	(Z_5 base set;·)
	associative	associative
	commutative	commutative
	J identity	J identity
	(0_5)	
	J inverse	J inverse
	(0)5 ->(0)5	(1)5 ->(1)5
	(1)5->(4)5	(2)5 ->(3)5
	(2)5 ->(3)5	(3)5 ->(2)5
	(3)5->(2)5	(4)5 ->(4)5
	(4)5->(1)5	

RSA

- It was published in 1977.
- Designers: Ron Rivest, Adi Shamir and Leonard Adleman
- It can be found in most Public Key Infrastructure (PKI) products, SSL / TLS certificates
- Secure email: PGP, Outlook

RSA encryption scheme

Asymmetric encryption scheme: AE = (Key; Enc; Dec)

- Key:
  1. We randomly choose two large primes: p; q.
  2. Calculate the modulus of RSA: $n = p \cdot q$.
  3. We calculate the Euler ϕ function: $\phi(n) = (p-1)(q-1)$.
  4. We choose a random e integer: $1 < e < \phi(n)$ and $(e, \phi(n)) = 1$. (e is the encryption exponent)
  5. Calculate d: $1 < e < \phi(n)$ and $ed \equiv 1 (mod \phi(n))$. ($d$ is the decryption exponent)

     $PK = (n; e), SK = d$ and $\phi(n); p; q$ secret parameters

     $P = C = \mathbb{Z}_n$
- $Enc_{PK}(m) = m^e (mod n)$ beside $\forall m \in P$ and $PK = (n, e)$.
- $Dec_{SK}(c) = c^d (mod n)$ beside $\forall c \in C$ and $SK = d$.

Euclid's algorithm

845	68	29	10	9	**1**	0
-	12	2	2	1	9	

That way the two numbers are relativ prime, ie.:$(845, 68) = 1$

Euclid's algorithm pseudo code

- Euklidesz(a, b, d)
- $d \leftarrow a$
- If($b \neq 0$)
- Then Euklidesz($b, a \ mod \ b, d$)
- Return (d)

Extended Euclid's algorithm

The greatest common divisor of two integers a and b can be expressed by the numbers $x, y \in Z$ in the following form:

$(a, b) = a * x + b * y$

Always!

$x_0 = 1 \; x_1 = 0$

$y_0 = 0 \; y_1 = 1$

Formula:

$x_{i+1} = x_i * q_i + x_{i-1}$

$y_{i+1} = y_i * q_i + y_{i-1}$

$x = (-1)^n * x_n$

$y = (-1)^{n+1} * y_n$

extended Euclid's algorithm example

k	0	1	2	3	4	
r_k	544	119	68	51	17	0
q_k	-	4	1	1	3	
x_k	1	0	1	1	2	
y_k	0	1	4	5	9	

$(-1)^4$ and $(-1)^{4+1}$

17=544*2+119*-9

Extended Euclid's algorithm pseudo code

- ExtendedEuclid(a, b, d, x, y)
- $x_0 \leftarrow 1$, $x_1 \leftarrow 0$, $y_0 \leftarrow 0$, $y_1 \leftarrow 1$, $s \leftarrow 1$
- While $(b \neq 0)$
- $r \leftarrow a \ mod \ b$, $q \leftarrow a \ div \ b$
- $a \leftarrow b$, $b \leftarrow r$
- $x \leftarrow x_1$, $y \leftarrow y_1$
- $x_1 \leftarrow q * x_1 + x_0$, $y_1 \leftarrow q * y_1 + y_0$
- $x_0 \leftarrow x$, $y_0 \leftarrow y$
- $s \leftarrow -s$
- End While
- $x \leftarrow s * x_0$, $y \leftarrow -y_0$
- $(d, x, y) \leftarrow (a, x, y)$
- Return (d, x, y)

Modular exponentiation

By using the following method shown in the example implementing relatively few operations we will get the value of a^b modulo m, where a integer, b integer is greater than 1 and m pozitive integer.

Algorithm:

1. step: The exponent is being written as a sum of the powers of 2: $b = 2^{b_1} + 2^{b_2} + ... + 2^{b_r}$

2. step: Calculate the following value by the repeated squaring: $a^{2^0}, a^{2^1}, ...a^{2^r}$

$$a^{2^{k+1}} = a^{2^k * 2} = \left(g^{2^k}\right)^2$$

3. step: We get the wanted power:

$$a^b = a^{2^{b_1}} * a^{2^{b_2}} * ... * a^{2^{b_r}} \pmod{m}$$

Example

$6^{73} \pmod{100}$

$73 = 2^6 + 2^3 + 2^0$

$6^{2^0} \equiv 6 \pmod{100}$
$6^{2^1} \equiv 36 \pmod{100}$
$6^{2^2} \equiv 96 \pmod{100}$
$6^{2^3} \equiv 16 \pmod{100}$
$6^{2^4} \equiv 56 \pmod{100}$
$6^{2^5} \equiv 36 \pmod{100}$
$6^{2^6} \equiv 96 \pmod{100}$

$6^{73} = 6^{2^6} * 6^{2^3} * 6^{2^0} = 99 * 16 * 6 \equiv 16 \pmod{100}$

Exercise

$129^{97} \pmod{171}$

Exercise

$129^{97} \pmod{171}$

$97 = 2^6 + 2^5 + 2^0$

$129^{2^0} \equiv 129 \pmod{171}$
$129^{2^1} \equiv 54 \pmod{171}$
$129^{2^2} \equiv 9 \pmod{171}$
$129^{2^3} \equiv 81 \pmod{171}$
$129^{2^4} \equiv 63 \pmod{171}$
$129^{2^5} \equiv 36 \pmod{171}$
$129^{2^6} \equiv 99 \pmod{171}$

$129^{97} = 129^{2^6} * 129^{2^5} * 129^{2^0} = 99 * 36 * 129 \equiv 108 \pmod{171}$

Modular exponentiation pseudo code

- Mod_exp($base$, exp, mod)
- $base = base\%mod$,
- if($exp == 0$)
- return 0;
- else if($exp == 1$)
- return base;
- else if($exp\%2 == 0$)
- return Mod_exp($base * base\%mod$, $exp/2$, mod);
- else
- return base * Mod_exp($base$, $exp - 1$, mod) $\%mod$;

Thank you for your attention!