Neso Academy Course: Cryptography and Network Security

ep 001

- need for network security:
    - most of networks are public
    - good guys, also bad guys

ep 002

- security
    - confidentiality
    - integrity
    - availablity
- confidentiality
    - privacy
    - authorized access
    - use cryptography
- integrity
    - it be what it was
- availablity
    - availablity
- authenticity
    - right permissions
- accountability
    - keeping track of activities
- security breach
    - low level
        - limited access and minor damage
    - medium level
        - huge access and significant damage
    - high level
        - everything access and catastrophe damage

ep 003

- threat
    - a potential for violation of security, which exists when there is a circumstance, capabitliy, action, or event that could breach security and cause harm. that is, a threat is a possible danger that might exploit a vulnerability.
- attack
    - an assault on system security that drives from an intelligent threat. that is, an intelligent act that is a deliberate attempt to evade security services and violate the security policy of a system.
- osi security architecture
    - security attack
        - action that compromises the security
    - security mechanism
        - detect, prevent and recover from a security attack
    - security services
        - enhance the security, counter security attack, provide the service
- attacks
    - passive
    - active
- security services
    - authentication
    - access control
    - data confidentiality
    - data integrity
    - nonrepudiation
- security mechanisms
    - encipherment
    - digital signature
    - access control
    - data integrity
    - authentication
    - traffic padding
    - routing control
    - notarization

ep 004

- passive attack
    - attempt to learn or make use of information form the system.

    - does not affect system resources.
    - eavesdropping or monitoring of transmissions.
    - goal: obtain information that is being transmitted.
    - types:
        1. release the message contents
        2. traffic analysis
- active attack
    - involve modification of the messages
    - types:
        1. masquerade
        2. replay
        3. modification of message
        4. denial of service (dos)

1. masquerade
    - hacker acts like the ligitimates
    - ex: stealing someone userpass
2. replay
    - resending what sender sends to receiver for some reason
    - ex: sending password hashed again!
3. modification of message
    - changing some data of original message
4. denial of service
    - many request to receiver to can't receive legitimate messages

- passive attack
    - hard to detect
    - just encryption can prevents from these attacks
- active attack
    - hard to prevent
    - usually not hard to detect

ep 005

- security service
    - processing or communication service that is provided by a system to give a specific kind of protection of system resources. security services implement security policies and are implemented by security mechanism.
- authentication
    - proves the identity of the sender
    - types:
        - peer entity
            - chekcing the sender of message on transmit
        - data origin

            - chekcing the sender of message at receiver system
    - access control
    - data confidentiality
    - data integrity
    - nonrepudiation
        - proving that who does what
        - types:
            - source level
            - destination level

ep 006

- security mechanisms
    - specific
    - pervasive
- specific security mechanism
    - encipherment
        - cryptography
    - digital signature
        - authentication with something in message
    - access control
    - data integrity
    - authentication exchange
        - exchanging digital signature
    - traffic padding
        - some additional data in network for fooling the attacker
    - routing control
        - determining the secure path in physical routers
    - notarization
        - deploying some trusted third parties
        - ex: ssl
- pervasive
    - trusted functionality
    - security labels
    - event detection
    - security audit trail
    - security recovery

ep 007

- network security model

    - cryptography famous image

1. design an algorithm encryption & decryption
2. generate the secret information
3. develop methods for distribution and sharing of information
4. specify the protocol

- gatekeeper function
    - someone that all should pass from it to go to the network
    - ex: firewall

ep 008

- cryptography
    - plaintext --> encryption --> ciphertext --> transmit --> ciphertext --> decryption --> plaintext

> encryption & decryption both needs key as input key is the most vital and secret thing

- cryptography
    - symmetric (private key)
        - same key for encryption & decryption
    - assymmetric (public key)
        - different key for encryption & decryption
- ciphertext should be very very unitelligible
- encryption
    - unconditionaly secure
        - no type of attack can broke the key
    - computationaly secure
        - not possible in limited time

ep 009

- assymmetric cryptography (public key)
    - everyone has 2 key (public key and private key)
    - public key is just for locking the message
    - private key is just for unlocking the message
    - sender knows the receiver public key and by that lock the message and put message on network
    - receiver receives the encrypted message and it can unlock it with its own private key

ep 010

- assumption
    - encryption & decryption algorithm are public
    - cipher text is public
- cryptanalysis attacks
    1. ciphertext only
    2. known plaintext
    3. chosen plaintext (can be adaptive or not)
    4. chosen ciphertext (can be adaptive or not)
    5. chosen pair (can be adaptive or not)

ep 011

- brute-force attack
    - searching all key space
    - until an intelligible translation of the ciphertext obtained

> brute-force tools:
>
> - Aircrack-ng
> - DaveGrohl
> - John the ripper
> - Cain and Abel
> - Hashcat
> - Rainbowcrack
> - Crack
> - Hydra
> - Ophcrack

- solution to brute-force attack
    - captcha!
        - completely automated public turing test to tell computers and human apart
        - random image showing that only human can understand it and by that identify the robot

ep 012

- classical cryptography techniques
    - substitution technique
    - transposition technique
    - these are not secure
- substitution technique
    - letters replaced by other letters
    - symmetric cryptography

- transposition technique
    - applying permutation in plaintext letters
- substitution technique algorithm
    - caesar cipher
    - monoalphabetic cipher
    - polyalphabetic cipher
    - playfair cipher
    - hill cipher
    - one time pad cipher
- transposition
    - rail fence
    - row column transposition

ep 013

- caesar cipher
    - every letter replaces with three after that letter
    - c = e(p, k) = (p + k) % 26 = (p + 3) % 26
    - p = d(c, k) = (c - k) % 26 = (c - 3) % 26

ep 014

- shift cipher
    - every letter replaces with three after that letter
    - c = e(p, k) = (p + k) % 26 (k is variable in this case)
    - p = d(c, k) = (c - k) % 26 (k is variable in this case)
- caesar cipher - pros and cons
    - pros
        - simple
        - easy to implement
    - cons
        - just 25 key in key space (easy brute-force)

ep 015

- monoalphabetic cipher
    - any mapping of alphabets to itself
    - this eliminate brute-force technique
- monoalphabetic cipher - pros and cons
    - pros

        - better than caesar cipher (no brute-force threat)
    - cons
        - letter frequency analysis attack is so possible

> when key is a sentence each letter in that maps to first to first second to second and repeated letter in plaintext use repeated maps.

ep 016

- playfair cipher
    - the key is 5x5 matrix for all alphabetic letters
    - (i and j letters goes to same coordinate)
    - key can be a word and then we fill other coordinates in matrix alphabetically sorted.
    - digraph is 2 letter besides each other like "ex"
    - we can use from "z" for last letter if needed
    - we can use from "x" in the middle for same letters digraph, ex: "ll" -> "lxl"
    - encryption
        - seperating digraph on plain text
            - plaintext digraph are on same row or column
                - we choose the next of each in that row or column
            - plaintext digraph are not on same row or column
                - we find the rectangle of them and choose other vertex in horizontal order

ep 017

- solving problems

ep 018

- solving problems

ep 019

- hill cipher
    - seperating the plaintext to digraph, trigraph, ...
    - based on letter count seperated = N -> k = Matrix-NxN
    - usually needed filler is "x"
    - c = e(k, p) = p x k % 26
    - p = d(k, c) = c x k^-1 % 26

ep 020

- mathematics review:

    - $k^{-1} = \frac{1}{det(k)} \times adj(k)$
    - note the k^{-1} is multiplicative (in ring) inverse
        - one way is to check all possibles numbers in ring

ep 021

- vigenere cipher
    - type of polyalphabetic cipher
    - c_i = (p_i + k_i) % 26
    - p_i = (c_i - k_i) % 26
- vigenere cipher cryptanalysis
    1. first we should find the length of the key (hard step)
    2. second we can find the key itself

> key and the plaintext share the same frequency distribution of letters so a statistical technique can be applied!

- autokey system for vigenere cipher
    - we use from plaintext to make the key in time!

ep 022

- vernam cipher
    - types of polyalphabetic cipher
    - length of the key = length of the plaintext
    - c_i = p_i xor k_i
    - p_i = c_i xor k_i
- vernam cipher cryptanalysis
    - if we find the repeated key we can find the xor of 2 plaintext

ep 023

- one time pad
    - type of vernam pad
    - random key for every seperation
    - so the key is not repeated
    - every key used just once

> the security of the one-time pad is entirely due to the randomness of the key. this method is unbreakable! but also not practical because of key sharing is a difficult process!

- one-time pad drawbacks
    - generating good random key

    - sharing the key with receiver
- perfect secrecy
    - absolutely nothing will be revealed about the plaintext by ciphertext

ep 024

- rail fence cipher
    - type of transposition technique
    - write the text diagonally in depth value, and read horizontally!

ep 025

- row column transposition (columnar transposition)
    - the key determines the columns counts and also columns order
    - if the key is a word order of the letter in alphabet determine the order of columns
    - write row by row
    - read column by column in key order
    - we can repeat this process and the repeated count is in key

ep 026

- steganography
    - hiding the message in the message
    - in counter to the cryptography the message can be intelligible
- steganography
    - character marking
    - invisible ink
    - pin punctures
    - typewriter color ribbon
- drawbacks of steganography
    - lot of overhead
    - once the system is discovered, it becomes wortheless

ep 027

- lsb steganography
    - lsb -> least significant bits
    - hide message on lsbs
    - it can be hide in even an image!

> tool for this: OpenStego

ep 028

- solving problems

ep 029

- in cryptography the number theory and abstract algebra are the important concept in mathematics and not rest of mathematics.
- which topic in this course we consider
    - the division algorithm
    - the euclidean algorithm the extended euclidean algorithm
    - modular arithmetic
    - groups, rings, fields and finite fields
    - polynomial arithmetic
    - prime numbers
    - fermat's and euler's theorem
    - testing for primality
    - the chinese remainder theorem
    - discrete logarithms
- computers
    - classic computer
    - quantum computer

ep 030

- prime numbers
    - exactly 2 divisors (1 and N)
    - all numbers have prime factors
        - ParseError: KaTeX parse error: Double subscript at position 10: n = p_1^m_1 \tiems p_2^m_…
    - P = {2, 3, 5, 7, 11, ...}
    - only even number: 2
    - smallest prime number: 2
    - 1 is not a prime number also not a composite number

ep 031

- modular arithmetic
    - integer numbers (Z set)
    - wrap around after reaching a certain value called modulus
- congruence
    - three line equality symbol
    - ex: 15 = 3 (mod 12) [15 is congruent to 3]

ep 032

- important property
    - [(a mod n) +-x (b mod n)] mod n = (a +-x b) mod n
- modular arithmetic properties
    - commutative laws
        - (a + b) mod n = (b + a) mod n
        - (a x b) mod n = (b x a) mod n
    - associative laws
        - [(a + b) + c] mod n = [a + (b + c)] mod n
        - [(a x b) x c] mod n = [a x (b x c)] mod n
    - distributive laws
        - [a x (b + c)] mod n = [(a x b) + (a x c)] mod n
    - identities
        - (0 + a) mod n = a mod n
        - (1 x a) mod n = a mod n
    - additive inverse
        - exists "-a" for "a" where a + (-a) = 0 mod n
- ring Z_n
    - {1, 2, 3, 4, ..., n}

ep 033

- modular exponentiation
    - exponentiation over modulus
    - a^b(mod m)

ep 034

- solving problems

ep 035

- gcd (method 1)
    - (greatest common divisor) aka hcf (highest common factor)
    - euclid's algorithm
        - gcd(a, b) = gcd(b, a%b)

ep 036

- gcd (method 2)
    - it is just before! wtf!

ep 037

- co-prime numbers
    - aka relatively prime numbers
    - gcd(a, b) = 1 <--> co-prime numbers

ep 038

- euler totient function
    - aka phi function
    - phi(n) = positive integer less than n that are relatively prime to n

ep 039

- solving problems
- phi formula
    - n is prime
        - phi(n) = n-1
    - n = a x b
        - phi(n) = n x (1 - 1/p_1)(1 - 1/p_2)(1 - 1/p_3)...

ep 040

- fermat's little theorem
    - p is prime and a is not divisble by p --> a^(p-1) = 1 (mod p)

ep 041

- euler's theorem
    - gcd(a, n) = 1 --> a^{phi(n)} = 1 (mod n)

ep 042

- primitive roots
    - {a^1 mod n, a^2 mod n, a^3 mod n, ..., a^{n-1} mod n} are distincts // a is primitive root

ep 043

- multiplicative inverse mod n
    - a x a^-1 = 1 (mod n)
    - if gcd(a, n) != 1 (not relatively prime) there is no a^-1 and if gcd(a, n) = 1 defenitely there is some a^-1

ep 044

- solving problems

- extended euclidean algorithm
    - t_1 = 0, t_2 = 1, t = t_1 - t_2 x q
    - shift also t_1, t_2, t besides a, b, a%b

ep 045

- solving problems

ep 046

- solving problems

ep 047

- chinese remainder theorem (crt)
    - solving the system
        - X = a_1 (mod m_1)
        - X = a_2 (mod m_2)
        - X = a_3 (mod m_3)
        - ...
        - X = a_n (mod m_n)
        - gcd(m_1, m_2, m_3, ..., m_n) = 1
    - if system is not above form you should convert it to that form if this is impossible problem has no solution
    - solution
        - X = (a_1 x M_1 x M_1^-1(mod m_1) + a_2 x M_2 x M_2^-1(mod m_2) + ... + a_n x M_n x M_n^-1(mod m_n)) mod M
        - M = m_1 x m_2 x m_3 x ... x m_n
        - M_1 = M/m_1, M_2 = M/m_2, ..., M_n = M/m_n
        - you should verify the answer

ep 048

- solving problems

ep 049

- one-way function
    - fun analogy: consider making the coffee, we need hot water and coffee pouder and we can make coffee very EASY, but consider we want to generate pure hot water and coffee pouder from the coffee maked it is very HARD to do that, so this process is a one-way process!
    - solving the result of equation 5^K mod 17 = ? is an easy equation but solving the problem of 5^? mod 17 = K is a hard equation, thus using these 2 equation we can have a one-way function!

> solving the problem of 5^? mod 17 = K is a hard equation this is because "?" can be very large one when the mod number (this case 17) is very large and also K is equally distributed by choosing random "?"! this is a case of I just compute one of the hard computing case and you should do many hard computing case to find what I did!

ep 050

- the discrete logarithm problem (dlp)
    - ex: log2(9) mod 11 = ?
        - solution
            - converted: 2^? = 9 (mod 11)
            - simple testing 1, 2, 3, ...
            - "6" satisfies! but...

ep 051

- factoring : fermat's algorithm
    - n = P.Q
    - this algorithm just works fine when X and Y are close
    - formula:
        - P = (X - Y), Q = (X + Y)
        - n = X^2 - Y^2 then X = \sqrt(n + Y^2) should be integer thus we can check number from 1 to ...

ep 052

- testing for primality (fermat's test)
    - a^p - a = k.p for all a in (0 < a < p)
    - this test has not complete accurecy!

ep 053

- testing for primality (miller-robin test)
    - this is also probablistic test: it can say that this number is probably a prime number or not
    - if p is a prime number then "something", so if "something" is false then p is not prime number but if "something" is right then p can be prime or composite!
    - "something" test:
        1. find n - 1 = 2^k x m (k=?, m=?)
        2. choose 'a' (1 < a < n-1)
        3. compute b_0 = a^m (mod n), b_1 = b_0^ (mod n), ..., b_i = b_{i-1} ^2 (mod n)
        4. result of each b_k: +1 --> composite number, -1 --> probably prime number, other value --> go forward

ep 054

- group
    - a group 'G' denoted by {G, .}, is a set under some operations (.) if it satisfies the CAIN properties.
    - CAIN properties
        - closure
        - associative
        - identity
        - inverse
- abelian group
    - CAIN properties plus communicative property

ep 055

- cyclic group
    - a group having at least one generator element
    - a generator element is the one that can generate all elements in group by repeating operation

ep 056

- ring
    - a ring R denoted by {R, +, *}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in R the following axioms are obeyed:
        - ring is abelian group on '+' operation
        - ring has CA on '*' operation
        - distributive laws for these 2 operation
- commutative ring
    - a ring plus commutative law for '*' operation
- integral domain
    - a commutative ring plus
        - identity law for '*' operation
        - a * b = 0 then a = 0 or b = 0 (no zero divisors law)
- fields
    - a integral domain plus
        - exists a^-1 for all a (except 0) where a * a^-1 = 1 (multiplicative inverse)

> you can do subtraction by '+' operation a - b = a + (-b)! you can do division

> by '*' operation a / b = a * (b^-1)

- finite field
    - field with finite number of elements

ep 057

- ciphers
    - block cipher
    - stream cipher
- confusion
    - making the relationship between encryption key and the ciphertext as complex as possible.
- diffusion
    - making each plaintext bit affect as many ciphertext bits as possible.
    - ex: changing one bit in pt, significant change on ct.
- stream cipher
    - bit by bit or byte by byte ciphering
- block cipher
    - fixed-length packed bits calls block (block by block ciphering)

ep 058

- feistel cipher structure
    - symmetric block cipher
    - splitting plaintext to left and right
    - some operation on left and right and swapping
    - repeating this porcess again as a new round
    - roll back same rounds on decryption
    - number of round should be determined
- aes and des also are block ciphers, aes algorithm is today in use, but des algorithm is not because it is not secure!

ep 059

- des (data encryption standard)
    - symmetric block cipher
    - it is type of feistel
    - it is replaced by aes now
    - block size: 64 bit (plaintext & ciphertext)
    - main key size: 64 bit

    - subkey size: 56 bit
    - roundkey size: 48 bit
    - number of round: 16 round
    - 64 bit is input & output of each round
    - in the last round output we do 32 bit swap

ep 060

- initial permutation - input as 8x8 square - right rotation to square - shuffling the rows one pass one - shifting on size of 4 rows
    - inverse initial permutation
        - input as 8x8 square
        - left rotation to square
        - shuffle the columns one pass one
        - shifting on size of 1 column
- single round des
    - input is 64 bit output of previous round and 48 bit permuted key
    - mangler function
        - 32 bit right half of input
        - expansion to 48 bit
        - xor with 48 bit key
        - passing 48 bit output to the 8 sbox
        - it converts 48 bit to 32 bit again
        - passing result to pbox
        - xor the result with the 32 bit left half input
    - right 32 bit place on left 32 bit of round output
    - mangler function output place on right bit of round output

ep 061

- mangler function
    - expansion function
        - input as a 8row x 4column square
        - appending 1 shifted leftmost column to right of square
        - appending 1 shifted rightmost column to left of square
        - output now is a 48 bit
    - simple xor with round key 48 bit output
    - sbox
        - 48 bit input
        - 8 small sbox
            - 6 input
            - 4 output
            - it answer based on a table that map (leftmost-rightmost

                2bit, middle 4bit) => 4 bit result
            - 32 bit output
        - pbox
            - input 32 bit
            - this done by a table of coordinate
            - output 32 bit

ep 062

- key scheduling in des
    - permuted choice 1
        - 64 bit key input
        - bits 8, 16, ..., 64 are dropped
        - 56 bit output (effective key)
    - for each round key generating we do
        - i = 1, 2, 9, 16
            - 1 left circular shift on previous result
        - i = others
            - 1 left circular shift on previous result
    - permuted choice 2
        - 56 bit input
        - 8 bit dropped (not said which bits)
        - 48 bit output (roundkey)
- decryption is same as encryption but just we reverse key used in encryption!

ep 063

- avalanche effect property in des
    - every encryption has this property
    - changing one bit in plaintext change many bits on ciphertext

> des has strong avalanche effect with 1 bit change in pt -> on average 34 bit change on ct, and 1 bit change in key -> on average 35 bit change on ct!

- strength of des
    - 56 bit key -> 2^56 = 7.2 x 10^16
    - brute-force attack is impractical
    - but des is not secure and it has a crack less than three days
    - also in sbox and pbox we should hide tables! that we know this is not good!
- timing attack
    - how long it takes a given implementation to perform decryptions on

        various ciphertext.
    - des is secure against timing attack

ep 064

- solving problems

ep 065

- aes
    - advance encryption standard
    - symmetric block cipher
    - widely in use today!
    - input plaintext 128 bit (16 byte)
    - output ciphertext 128 bit (16 byte)
    - input go to initial transformation
    - then result go to rounds
    - round
        - each round contains 4 transformation
            - substitute byte
            - shift rows
            - mix column
            - add round keys
        - final round has 3 transformation
    - all round keys are 128 bit
    - initial transformation gets round key index 0
    - key size determines the round counts
    - types (the number is the key size)
        - aes-128 (number of round = 10)
        - aes-192 (number of round = 12)
        - aes-256 (number of round = 14)

ep 066

- aes encryption & decryption
    - a word is 32 bit
    - initial transformation
        - xor with round key 0
    - in last round we have 3 transformation because we have not mix columns step
    - encryption and decryption are same again but key indexes is inverse

ep 067

- aes rounds

        - 4 transformation
            - substitute byte
            - shift rows
            - mix column
            - add round keys
    - we look 128 byte as a 4x4 array
    - substitute byte
        - this is like sbox and have a table that (x, y) coordinate to new (x', y') coordinate
    - shift rows
        - first row untouched
        - second row shift left by 1
        - third row shift right by 3
        - last row shift right by 1
    - mix columns
        - multiply 4x4 matrix to a known 4x4 matrix
    - add round key
        - simple xor operation

ep 068

- aes key scheduling
    - it needs very memorizing