Page No.
Date

INS Mod 3

* Basic Concepts of Access control.

→ 'Access' is a key concept that implies flow of information from a subject to an object and object to a subject.

→ Access control policies make sure of the fundamental requirements of a secure system through authentication and authorisation.

→ Access control involves mechanisms for selectively restricting access to resources or places, such as determining who can access specific files or system resources

→ It functions to manage which active subjects (user or processes) have access to passive objects (data or resources) for specific access operations.

→ Examples are - Password Encryption, antivirus software, firewalls, routers, and data integrity preservation tools.

→ Main aim of access control mechanisms is to control the actions executed by subjects, preventing any actions that could compromise data.

Date

* Goals of Access control

① Granting Access : Means allowing authorised users or subjects to access specific resources or perform certain actions.

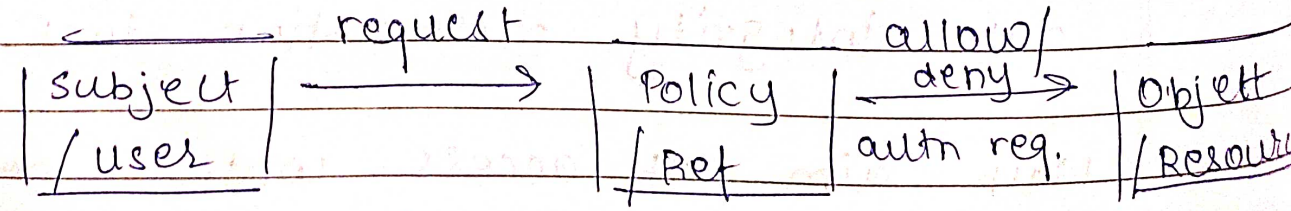
② Limiting Access : Practice of controlling and restricting the level of access that users or have processes have to resources.

③ Preventing Access : Denying Access to unauthorised users or subjects.

④ Revoking Access : Taking away access from users or subject who previously had access to certain data.

---

* Abstract Access control Model.



Page No.
Date

→ Access control involves an active subject seeking access to a passive object for specific Access operations.

→ Subjects, in access control models, are the entities initiating access requests and can be human users, software entities, processes, modules etc.

→ Objects represent the resources containing data that may require access control. Objects may include memory, files, directories, network nodes.

→ A reference monitor / policy is responsible for granting or denying access and making auth. decisions.

→ Eg. O.S, firewalls, JVM (Java Virtual Machine)

* Access Modes.

There are two main access modes.

① Observe: In this mode, a subject can only view the content of the object. common actions in this mode include reading and searching.

② Alter: This mode allows the subject to modify the content of the object including actions like writing, appending and deleting.

* Access control Models.

① Mandatory Access Control (MAC)

→ It is a non-discretionary access control model.

→ Access is determined by a central authority based on information clearances and security levels.

→ Commonly used in govt. and military environment.

→ MAC begins with assigning security labels to all resource objects, which contains a classification (eg. top-secret, confidential) and a category (includes a management level, department or project.)

→ When a user tries to access a resource in a MAC system, the user checks classification and category and compares the requirements.

→ Access is only allowed if the requirements match otherwise it is denied.

② Discretionary Access control (DAC) is

→ DAC is a method where the owner or administrator of a system, data or resource sets access policies.

→ In DAC, each user has control over who can access their own data, making it a common strategy in desktop O.S.

→ Instead of using security labels, DAC uses Access control Lists (ACL) associated with resource objects. ACLs lists users and groups that have permission to access the resource and specify the level of access for each user or group.

→ For eg. user A has read-only access to a file belonging to user B, but C might have write access to the same file.

Page No.
Date

→ DAC offers flexibility but increases the risk of data being accessed by users who should not have access.

⑤ Role-Based Access Control (RBAC)

→ RBAC grants access based on pre-defined business functions rather than individual user identities.

→ Goal of RBAC is to provide users access only to data necessary for that specific role in the organisation.

→ Takes a real-world approach to access control, linking access to a user's job function within the organisation.

→ Roles are assigned to the organisation and users are assigned roles.

→ Example - 'Accountant' role grants permissions for accountants in the system.

'Developer' role is alloted to all the software engineers in the organisation.

④ Attribute based Access control (ABAC) is a dynamic access control method.

→ Access is determindd based on a set of attributes and environmental conditions associated with both users and resources.

→ May include factors like user identity, time of day, location and more.

× ×

* Access Control Mechanism:-

→ Access control involves two main steps:-

① Authentication: Aims to detelmine the identity of a user and answer the question "Are you who you say you are?"

May also involve verifying IP, machine and request time of the user

Various methanisms like passwords, cryptographic methods, smart cards, biometrics enforce authentication.

② Authorization: Follows Authentication and answe the question "What are you allowed to do?

→ Enforces limitations on the user's action after access is granted.

Authentication	Authorization.
① Verifies credentials and determines whether users are who they claim to be	① Grants or denies permission to determin what user can and cant do.
② users are verified	② users are validated.
③ Works through passwords biometrics, OTPs etc.	③ Works through admin panels.
④ Eg - Login / OTP / Single sign on (SSO)	④ Eg - RBAC, JSON Web token (JWT), OAuth.
⑤ Visible to user	⑤ Not visible to user
⑥ Partially changeable by user	⑥ Not changeable by user.
⑦ Data moved through ID tokens	⑦ Data moved through access tokens.

Page No.
Date

* Access Control Matrix

→ An access control matrix is a fundamental control structure used to implement a protection model.

→ It is represented as a matrix where each row represents a subject (user, processes, procedure) and each column represents an object (resource).

→ The matrix characterizes the possible rights of each subject concerning each object. The entry M[s,o] defines the operations subject 's' can perform using object 'o'

→ The access control matrix is a valuable tool for visualising and managing rights within a system, covering permissions like read, write, execute and more.

	OS	A.P	A.D	I.D	P.D.
Bob	rx	rx	r	-	-
Alice	rx	rx	r	rw	rw
Sam	rx	rx	rw	rw	rw.

→ Advantages.

① Clarity of definition
② Easy to verify.

→ Disadvantage

① Poor scalability
② Poor handling of changes

* Access control Lists (ACL)

→ ACL is an object-centered description of access rights, representing the columns of the access control matrix associated with each object, while omitting empty entries.

→ Ideally, its a list per object showing all subjects with access and their respect rights.

→ It simplifies permissions management by listing subjects with their access rights for a particular object.

→ Missing subjects are usually given default access making it easy to set objects as public.

→ Example entry: 'bill.doc' : { Bob : {read, write)
Alice : read
}

'exit.exe' : { Alice : (execute),
Bob : (execute),
}

→ Advantages

① Easy access to object access rights

→ Disadvantages

① Poor overview of access rights
② Difficulty of revoking
③ Difficulty of sharing.

* Capabilities

→ Can be seen as storing the rows of the access control matrix with the corresponding subject, excluding empty entries.

Date

→ When a subject attempts an operation, their row in the access control matrix is consulted to determine if the operation is permitted.

→ These are unforgeable tokens granting users access to objects while specifying the allowed level of access.

Example - Alice : { 'edit.exe' : execute,
'fun.com' : read.
}

Bob : { 'bill.doc' : read, write,
'edit.exe' : execute,
'fun.com' : execute'
}

→ Advantages

① Easy ownership transfer
② Easy inheritance of access rights

→ Disadvantages

① Poor view of access rights per object
② Difficulty of revocation

Page No.
Date

Confused Deputy.

Scenario

① Resources

- compiler (can write into any file)
- BILL file (contains critical biling information)

② Users

- Alice (can invoke compiler, ~~can~~ can provide debug filenames)

③ Permissions.

- compiler can write into any file
- Alice cannot write directly to the BILL file to prevent corruption.

④ Scene.

- Alice invokes the compiler, providing 'BILL' as the debug filename.
- The command should fail because Alice lacks priviledge to access the BILL file.

compiler acts as Alice's deputy, confused about Alice's and his own permission.

(5) Outcome

- Expected failure of Alice's command, protecting the BILL file
- Potential risk of trashing the BILL file due to confusion of priviledge management.

* ACLs vs Capabilities.

ACL	Capabilities
① users manage their own files.	① Delegation and avoiding confused deputy.
② Protection is data-oriented.	② Not data-oriented.
③ Easy to change rights of resources	③ Complex to implement
④ Difficult to avoid C.D	④ Easy to avoid C.D.
⑤ Used often in practice	⑤ Less common but has specific uses

Date

* Covert channel.

→ It is a communication path which is unintended by system designers.

→ Arises in network communication, challenging to eliminate due to inherent system complexities.

→ Exploited by a process to transfer information discreetly, violating established security policies.

• MLS (Multi-Level Security)

→ MLS is designed to control and restrict legitimate communication channels.

→ Despite design efforts, there may be alternative ways for information to flow, for eg- shared resources signali information.

• Example scenario.

→ Alice (TOP SECRET) and Bob (confident are sharing file space.

→ Alice employs the covert channel, creates file XYZ. to signal "1" and removes it with signal "0".

→ Bob, with lower clearance level, monitors the file's existence to receive covert signals from Alice.

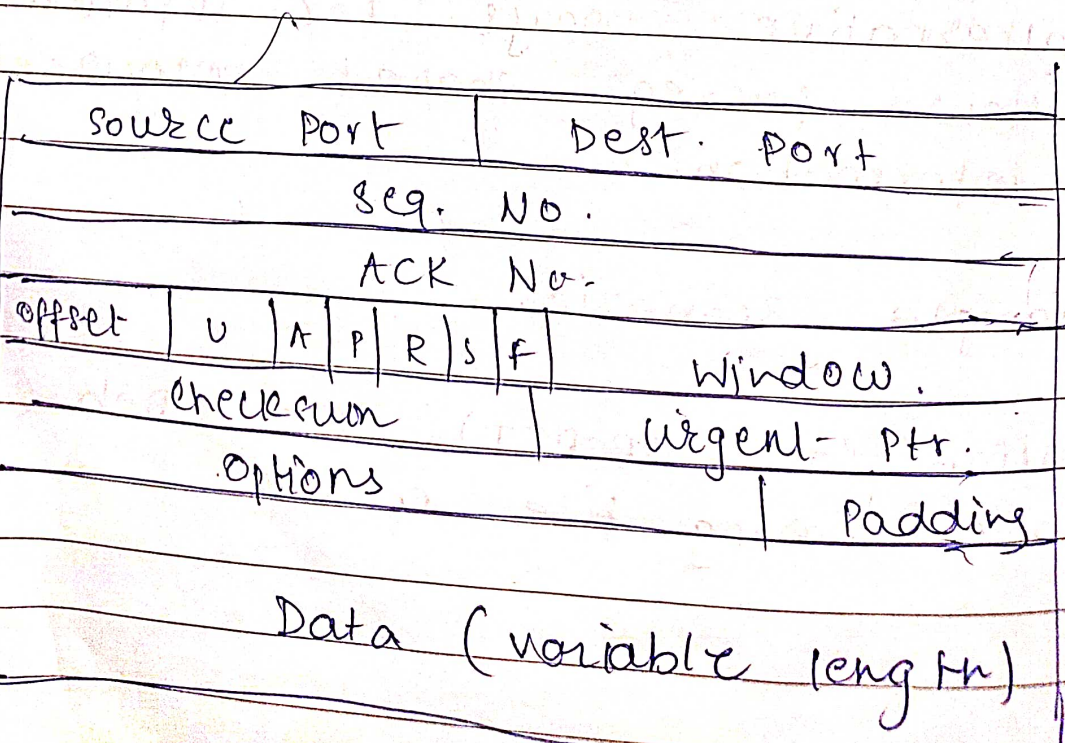
Alice : 'Create 'Delete 'Create

Bob : Check Check Check Check

Data 0 1 0 1 1

Time

- Real World Covert Channel.



→ Data can be hidden in TCP header reserved field.

- Sequence Number
- Acknowledgement number.

---

* Authentication

→ Network Administrators require login IDs and passwords to access network devices.

→ Instead of managing account information locally, using an authentication protocol is a much more secure protocol.

→ An authentication protocol is a cryptographic protocol designed for the secure transfer of authentication data between two entities. ; typically a client and a server.

Examples of A.P.

① TACACS+ : Terminal Access controller Access-control System.

② Radius : Remote Authentication Dial In User Service.

LDAP: Light weight Directory Access Protocol.

* Key Management.

→ Managing keys is a crucial aspect of both symmetric and asymmetric cryptography.

→ Symmetric key cryptography requires a shared secret key for encryption and decryption.

→ Asymmetric key cryptography involves public and private key pairs.

→ Symmetric key cryptography is efficient for encrypting large messages

* Methods of symmetric key distribution

→ Using TTP (Trusted Third Party): Centralise server facilitating secure key exchange.

→ Without TTP (Diffie Hellman): Parties generate a shared secret without trusted intermediary.

→ Using KDC (Key Distribution center): central authority managing key distribution.

→ Using CA's (certificate authorities): Issuing digital certificates to validate key ownership.

→ PKI (Public Key Infrastructure): framework integrating policies and procedures for key management.

* Key Distribution center (KDC)

→ Creates a secret key (eg. ~~KAS~~ K_{AS}, K_{BS}) for each member.

→ These secret keys are for communication exclusively between the member and the KDC.

• Session Symmetric Key (K_{AB})

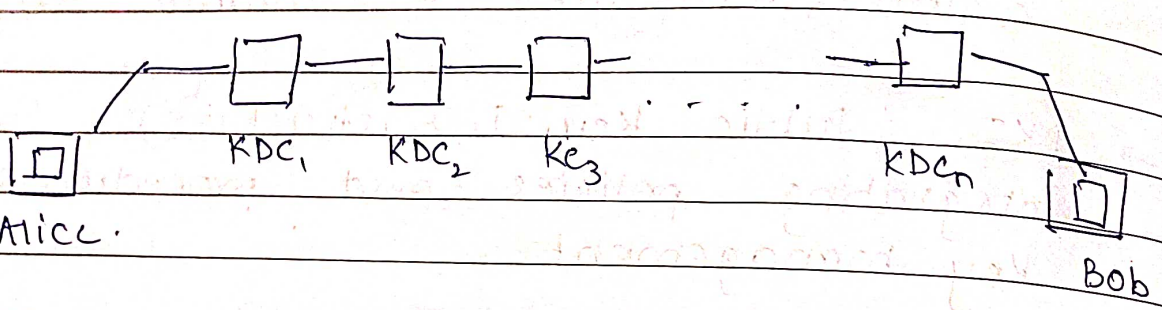
→ Established along with the KDC.

→ Used for communication b/w two parties, such as Alice and Bob.

→ Agreed upon with the server (Bob's) consent.

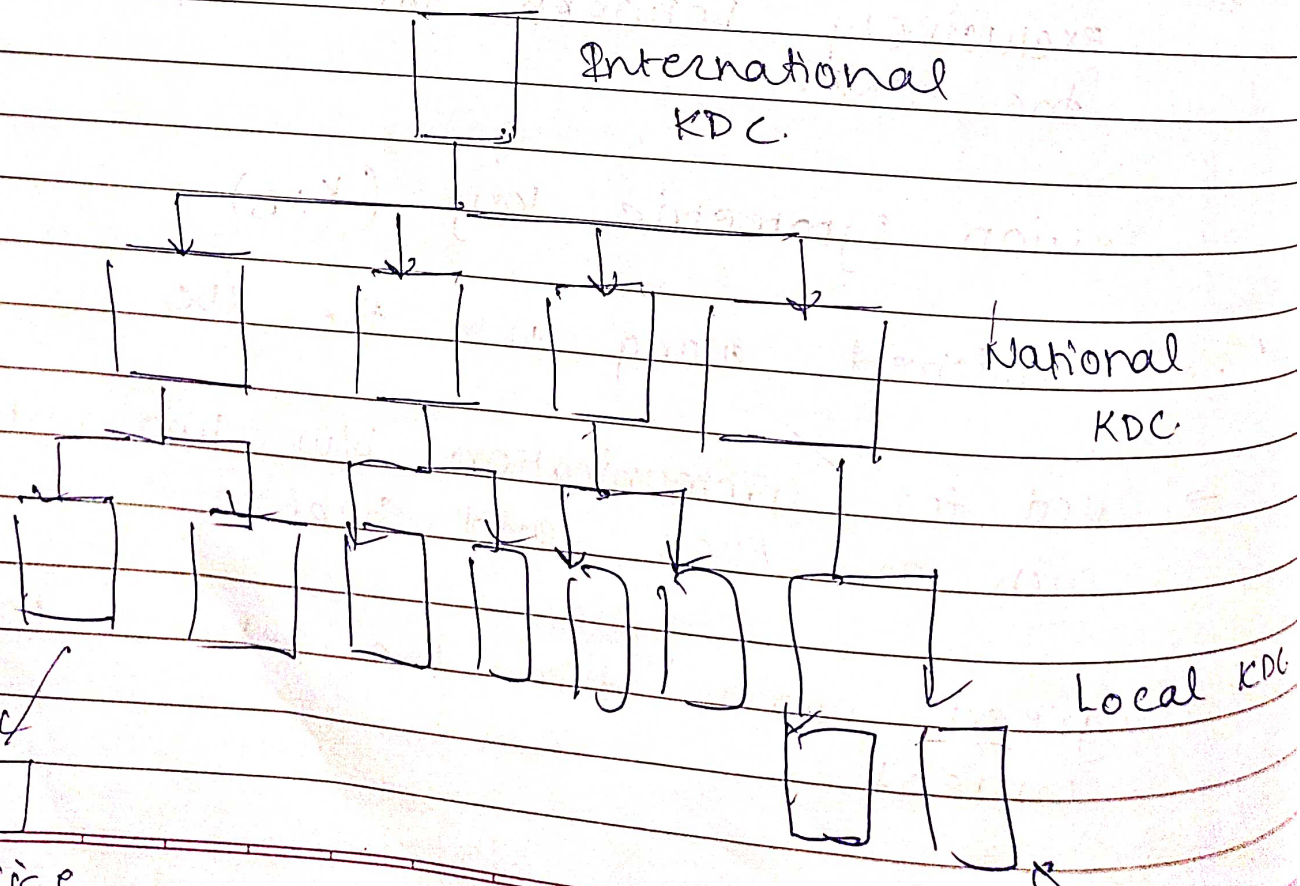
* Types of KDC.

① Flat Multiple KDC.

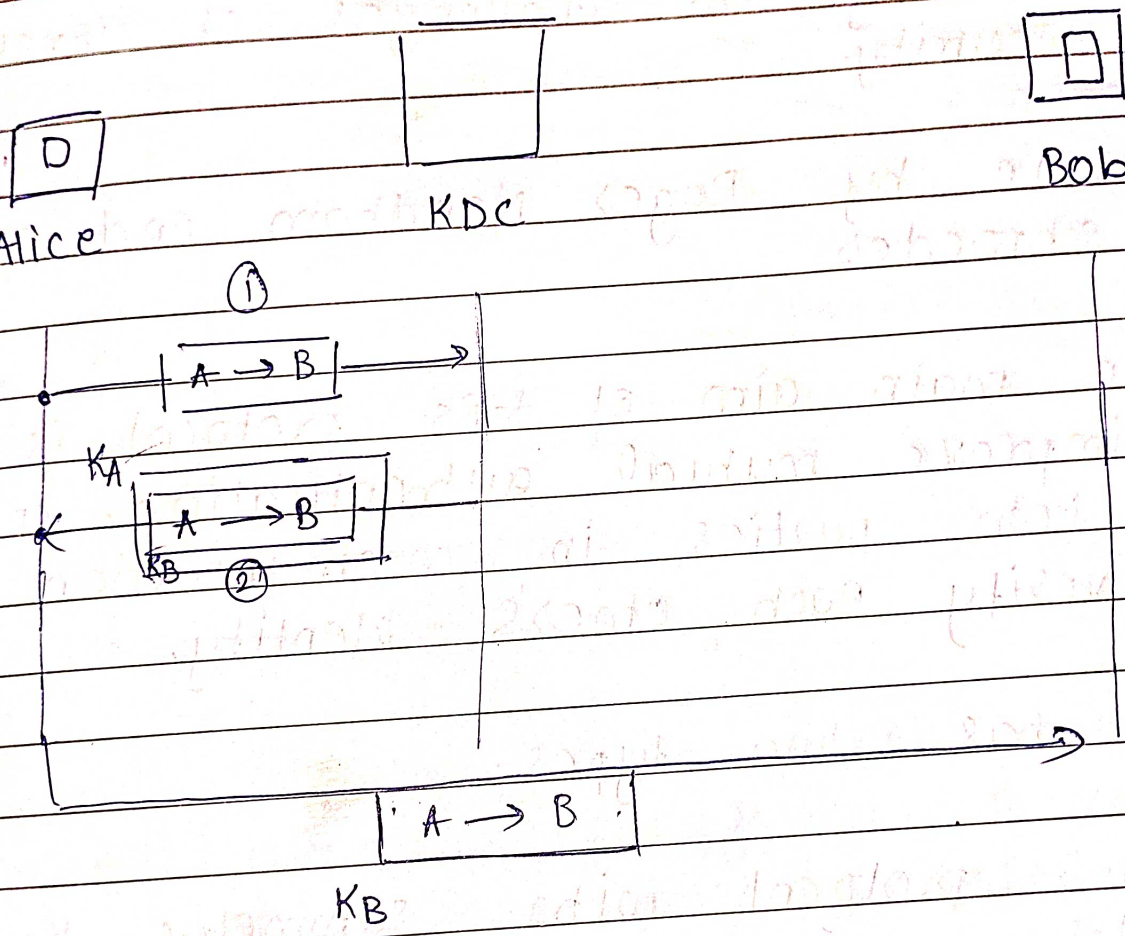


→ Multiple KDC operate independantly, each managing its own set of keys and handling authentication protocols.

② Hierarchial KDC.



→ There is a top-level KDC & which oversees everything. Below it there are subordinate KDC each managing their specific domain.



Working of KDC.

* Needham - Schroeder Protocol.

→ Needham - Schroeder protocol is crafted to secure communication in insecure networks addressing the challenges of unreliable security.

→ Made by Roger Needham and Michael Schroeder

→ The main aim of this protocol is to improve mutual authentication, ensuring both parties in communication can verify each other's identity.

→ It has two types.

① N-S protocol with symmetric key
② N-S protocol with asymmetric key.

• Protocol Identities.

→ Alice initiates communication with Bob

→ K_{AB} is a server key trusted by both parties.

Page No.
Date

→ K_{AS} - Symmetric Key known to Alice and server
K_{BS} - Symmetric Key known to Bob and server
K_{AB} - Session Key known to Alice and Bob.

→ N_A and N_B are nonces generated by A and B respectively.

• Nonces.

→ Short for 'numbers used once' are randomly generated values included in messages.

→ Nonces are employed in encryption protocol. to prevent replay attacks.

→ If someone captures a packet, they can resend it without decrypting. To counter this, nonce (randomly generated value) is added to this data. allows it to be used only once.

→ If a nonce is generated and sent by party A in one step and then returned by party B in a later step. A can verify that B's message is fresh and is not a Replay attack.

Page No.
Date

Steps for N-S protocol (Symmetric Key)

① Alice initiates communication with KDC.

- Alice sends a message to the Key Distribution center (KDC) containing her identity (A), Bob's identity (B), and her nonce (N_A)

∴ Message : $A \longrightarrow KDC : A, B, N_A$

② KDC generates session Key and Ticket for Bob

- KDC generates the session key (K_{AB}). It appends Alice's identity and encrypts it with Bob's secret key (K_{BS}), creating a ticket for Bob.

- The ticket along with the session key (K_{AB}), Bob's identity (B) and Alice's Nonce (N_A) is encrypted with Alice's secret key (K_{AS}) and sent back to Alice.

Encrypted Ticket : $S \longrightarrow A : \{ N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}} \}_{K_{AS}}$

Page No.
Date

③ Alice forwards ticket to Bob.

Forwarded Ticket : $A \rightarrow B : \{K_{AB}, A\}K_{BS}$.

④ Bob authenticates and responds.

- Bob decrypts the ticket using the key (K_{BS}) shared with the KDC, authenticating the data. Bob then sends Alice a nonce (N_B) encrypted under session key (K_{AB})

Encrypted Nonce : $B \rightarrow A : \{N_B\}K_{AB}$.

⑤ Alice verifies and responds.

- Alice performs an operation on the nonce provided by Bob, decrements it by 1, re-encrypts it with the session key $\{K_{AB}\}$ and sends it back to Bob.

$A \rightarrow B : \{N_B - 1\}K_{AB}$.

Page No.
Date

* Kerberos

- User Access Threats.

① Users pretending to be others
② Users altering the network address of a workstation.
③ Users eavesdropping on exchanges and employing replay attacks.

- Authentication Solutions

→ Authentication using public key was proposed, where N pairs would require N keys for public authentication.

→ This method would require approx N^2 keys making it impractical for a large no. of users due to scalability issues.

→ It was orignated in MIT and is based on the Needham and Schroeder Protocol.

→ It uses UDP port 88 by default.

Page No.
Date

- Components of Kerberos

① Client - Workstation (CWS): Represents the user initiating the authentication process. (eg. Alice)

② Authentication Server (AS): Register's users. Verifies them and issues a secret key, for communication with the Ticket Granting Server (TGS)

③ Ticket - Granting Server: Issues Tickets for real server (eg. Bob) and provides a session key (K_{AB}) for secure communication.

④ Real server: Provides services for user after authentication.

---

* Steps in Kerberos.

① User Authentication.

→ Alice enters her username, which is received in plaintext by the authentication server (A.S).

Page No.
Date

→ A.S creates a package for Alice, containing a randomly generated session key (K_{A-TGS}), Alice's user ID, expiration time and encrypts it using the symmetric key (K_{AS-TGS}) shared between A.S and TGS.

→ Output is a Ticket Granting Ticket, which can only be decrypted by TGS.

→ Symmetric Key K_{A-TGS} is derived from Alice's password.

→ TGT is issued when the user logs in and serves as user credentials for obtaining tickets to access network resources.

$$TGT = (K_{A-TGS}, \text{"Alice"}, \text{timestamp})_{K_{AS-TGS}}$$

→ Client Machine destroys the password entered by Alice to prevent theft by attackers.

$$((TGT)_{K_{AS-TGS}}, K_{A-TGS})_{K_{A-AS}}$$

Page No.
Date

② Accessing Bob's Email server.

→ Alice, after login wants to access Bob's email server.

→ Alice sends TGT to TGS to obtain ticket for Bob

→ Client machine creates a message for TGS including ID of server (Bob), expiration, encrypted by session key K_{A-TGS}

③ TGT and Ticket Processing.

→ TGS obtains K_{A-TGS} from TGT using symmetric key K_{AS-TGS}.

→ TGS verifies TGT and the timestamps.

→ TGS creates a session key K_{A-B} for secure communication b/w Alice and Bob.

→ TGS issues two tickets to Alice.

a) K_{A-B} combined with Bob's ID and encrypted with Key K_{A-TGS}. (Ticket to Alice)

K_{A-B} combined with Alice's ID and encrypted with Bob's secret key K_{TGS-B} (ticket to Bob).

④ Forwarding Bob's Ticket

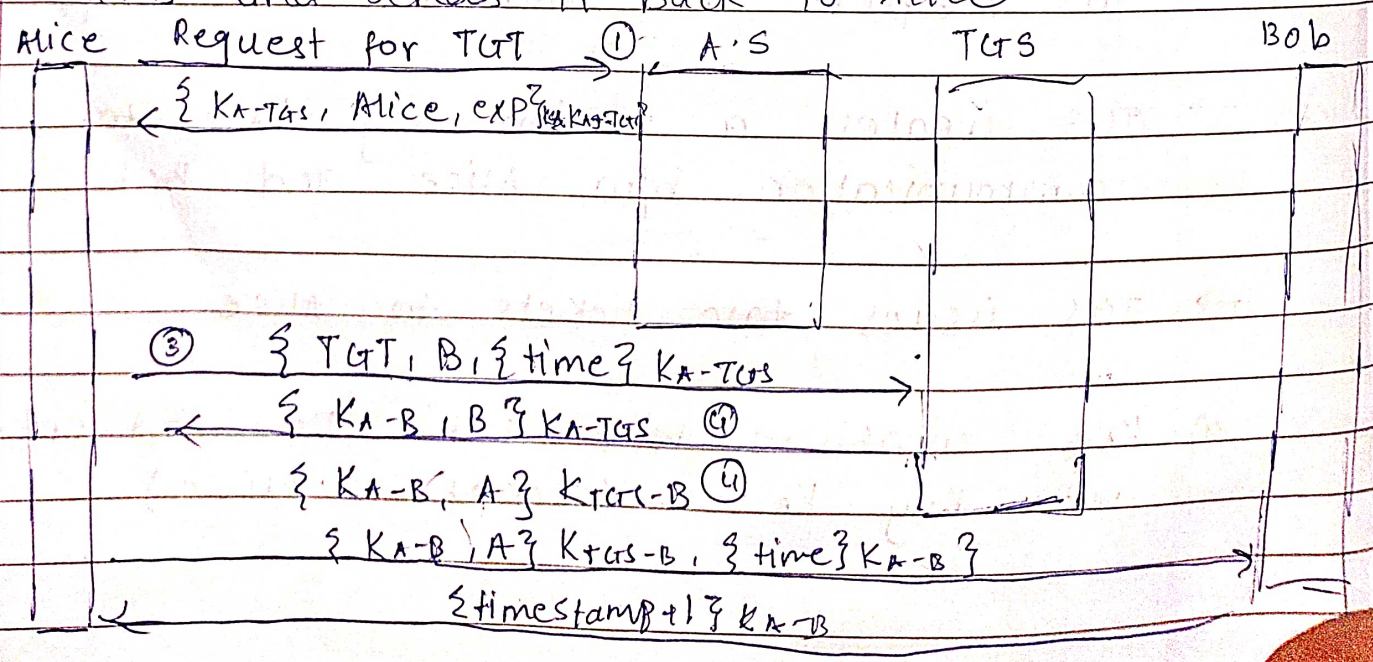
→ Alice forwards Bob's ticket ($\{K_{A-B}, A\}K_{TGS-B}$) received from TGS.

→ To guard against replay attacks, Alice adds timestamp encrypted with K_{A-B}.

⑤ Bob's Processing and Management.

→ Bob generates K_{A-B} and decrypts the timestamp.

→ For acknowledgement, Bob adds +1 to the timestamp and encrypts the result with K_{A-B} and sends it back to Alice.



→ Advantages.

① Statelessness.

② Anonimity of Alice

③ Replay Attack Prevention.

→ Disadvantages

① Depends on security of KDC/TTP.

② Clock Synchronization is required.

---

* Zero Knowledge Proofs.

→ Developed by Fiege, Fiat and Shamir.

→ Typically involves multiple rounds and has high computational costs.

Concept: ZKP allows Alice to prove to Bob that she knows a secret without revealing any info about the secret.

→ Bob can verify that Alice knows the secret without revealing any information about what the secret was.

→ In traditional authentication, user provides a password which is not confidential and can be exploited by dishonest verifiers.

* Steps in Fiat-Shamir Protocol:

① RSA Setup.

→ Bob, the verifier or trusted third party (TTP) chooses two large prime numbers, p and q and calculates $N = p * q$. N is made public while p and q are kept secret.

② Alice's key generation

→ Alice, the claimant, chooses a secret s such that s is co-prime to N and $1 \leq s \leq N-1$

→ Alice computes $v = s^2 \bmod N$. Here s is her private key, v is her public key.

③ Witness Generation

→ Alice selects a random number r b/w 0 & to N-1

→ She computes $x = r^2 \bmod N$, which is reffered as witness.

→ Alice sends x to Bob.

④ Challenge from Bob.

→ Bob sends challenge e to Alice, where e is either 0 or 1.

⑤ Response Calculator.

→ Alice calculates $y = r * s^e \bmod N$.

→ Alice sends y to Bob to demonstrate that she knows the secret key s.

⑥ Verification By Bob.

→ Bob calculates y^2 and checks if it's congruent to $x * v^e \bmod N$.

→ If the two values match then Alice knows the secret key s, else

These rounds are repeated multiple times to increase security.

→ Alice must pass this test in every round if she fails even in one round she is not authenticated.

→ If an intruder comes, he/she can fool Bob with a probability 1/2.

→ With 20 rounds intruder probability goes from 1/2 to $1/2^{20}$. Which is very less.

---

* CAPTCHA

→ CAPTCHA stands for Completely Automated Public Turing Test to tell Computers and Humans Apart.

→ It is an Inverse Turing Test, generated and scored by a computer.

→ Designed to be passable by humans, but challenging for machines, even with access to source code.

→ Functions as an access control mechanism restricting access to resources to humans only, excluding bots.

Page No.
Date

. Requirements of CAPTCHA:

① Must be easy for most humans to pass

② Must be difficult or impossible for computer to pass, even if computer has access to CAPTCHA software.

③ Randomness in generating CAPTCHA.

④ Different types of CAPTCHA can be implemented based on individual requirements.

. Features of CAPTCHA.

① Accessibility: Should be accessible to visually impaired users as well, allowing audio CAPTCHA.

② Image security: CAPTCHA images of text should be distorted randomly to prevent simple automated attacks.

③ Script security: Should ensure there are no easy ways around it at script level, avoiding vulnerabilities.

- Applications of CAPTCHA.

① Preventing Comment Spam on Blogs.

② Protecting Website Registration.

③ Protecting Email Addresses from Scrapers.

④ Search Engine Bots.

* Types of CAPTCHA

① Text-Based.

→ Involves distortion of text which is not readable by OCR and can be interpreted by humans only.

→ Eg. Gimpy, ez-Gimpy., Simards HIP.

② Graphic-Based

→ Involves patterns, graphics, images which are recognisable only by humans and not by machines.

→ Eg. Bongo, Pix, 3D.

③ Audio/Video Based Cap CAPTCHA-

a) Video - Based.

→ Uses combination of both audio and video ~~which~~ and gives instructions based on the video.

→ Instructions can only be followed perfectly by humans

× ×

* Authentication Methods.

→ User Authentication involves methods to validate users attempting to access a computer system or resources, ensuring they are authorised.

→ It authenticates a user to a machine to verify the identity of someone or something to be a particular user.

· Types of User Authentication

① Something you Know: Eg. Usernames or Passwords.

② Something you own: Smart card, security tokens

Date

③ Something you are: Eg. Biometrics (Fingerprint, Facial Recognition)

- Ideal Passwords.

→ An ideal password is something which the user knows but its hard to guess.

→ Example - PIN for an ATM, web app password, social security number, DOB, Mother's maiden name etc.

- Popularity of Passwords.

→ Passwords are free compared to biometric devices and smart cards.

→ Easier for administrator to reset a compromised password.

- Types of Passwords

① Cognitive Passwords.

→ Created through experience-based questions (eg- fav. color, college name).

Page No.
Date

② One-Time Passwords: Used in sensitive cases (eg- token devices, prepaid cards)

③ Passphrases: A sequence of characters no longer than a password. entered into an application that transforms it into a password.

④ Personal Identification Number (PIN): An arbitary string of numeric characters.

· Common Attacks on Passwords

① Dictionary Attack: Uses a list of common words to try against accounts, exploiting weak password choices.

② Brute Force Attack: Attempts to generate likely passwords, starting with common ones and trying variations.

③ Traffic Interception: Monitors network traffic to capture passwords.

④ Man in the Middle / Social Engineering: Inserts itself in interactions, often through impersonation or luring users to a fake site.

Date

⑤ Key-Logger Attack: Installs software to track Keystrokes, capturing usernames and passwords

* Bad Passwords vs Good Passwords

- Bad Passwords

→ Default Passwords: (supplied by vendor at time of installation to be changed later.

→ Dictionary Words: Chameleon, RedSox, Sandbags etc.

→ Words with numbers appended: (password1, bunnyhop!, IntenseCrabTree etc.

→ Words with simple obfuscation - @ssword, gold$fish. etc.

→ Doubled Words: crabcrab, stopstop, treetree, passpass etc.

- Good Passwords

→ Allow long passphrases.

Page No.
Date

→ Randomly generated password.

* SALT.

. Hashing with SALT

→ Salting involves adding a random piece of data (salt) to password before hashing.

→ Ensuring same password hashes to different values at different times.

. Unique Hashes for same Passwords.

→ Users with the same password will have different entries in password file due to unique salt

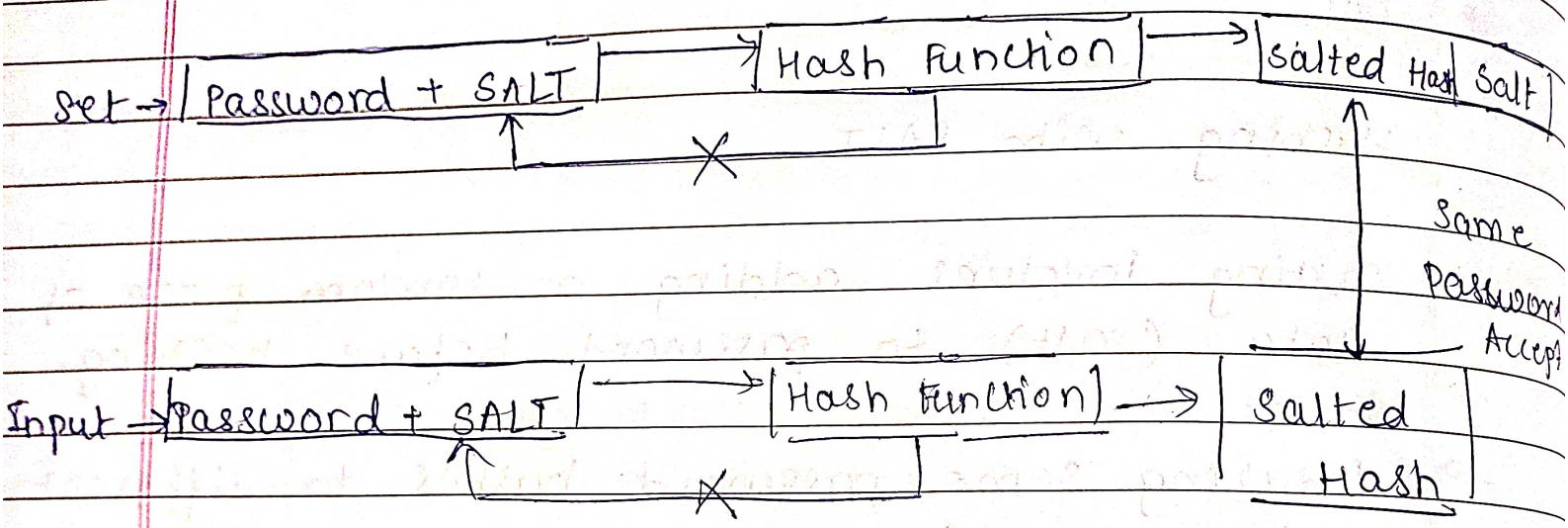
. SALT storage

→ Salt is stored along with the hashed password as a complete hash entry in the password file.

. Random Salt Selection.

→ A random salt is chosen for each password and hash is computed as $y = hash(password, s)$.

→ A 12 bit random salt generates 2^{12} (4096) different values for the same password.



* Biometrics.

→ Biometrics involves automated methods of recognizing a person based on physiological or behavioral characteristics.

* Types of Biometrics.

① Biological/Physical : a) fingerprint
b) Iris
c) Retina
d) Facial Recognition
e) Hand Geometry.

Page No.
Date

② Behavioural Examples: a) Handwriting
b) Gait Recognition
c) Mouse Gesture Recognition
d) Odor Recognition.

- Advantages of Biometrics.

① Security and Accuracy: Biometric data cannot be forgotten, exchanged or stolen.

② Convenience: No need to remember password or easy cards.

③ Accountability: Provided better accountability of who accessed the system.

- Biometric characteristics -

① Universality: No biometric applies to everyone.

② Uniqueness: Biometrics should distinguish individuals with virtual certainity.

③ Permanence, Physical characteristics should resist aging.

4. Collectibility: Easy to collect without causing any harm.

(5) Performance Reliance

(6) Degree of Acceptibility is high.

(7) Difficult to fool the authentication system.

* Biometric System Phases

(1) Enrollment Phase

→ Raw biometric is captured and entered into the database.

→ Distinguishing characteristics extracted and converted into a biometric identifier record.

(2) Recognition Phase

→ Compare preset owner's data with the visitor's data.

→ Access is granted if the data matches (almost.)

Page No.
Date

- Fraud and Insult Rate

→ Fraud Rate is the rate at which mis-authentication occurs.

- Example - System interprets Bob as Alice.

→ Insult rate is the rate at which system fails to authenticate the subject.

→ Equal Error Rate: Rate at which fraud and insult rates are the same.

x x

* Fingerprints

⇒ Fingerprints are stable and do not change over time

→ Fingerprint images are captured and enhanced using image processing techniques.

→ Features like arches, whorls, loops, minutiae and furrows are recorded.

Page No.
Date

- Fingerprint Enrollment

① Capture fingerprint image
② Enhance the image
③ Identify 'points' such as ridges and valleys.

- Fingerprint Comparison

① Minutae-based Matching: Compares specific points in the template and input minutiae

② Correlation-based Matching: Calculates the association b/w equivalent pixels in two fingerprint images.

③ Ridge Feature-based Matching: Captures ridges, suitable for low-quality fingerprint images.

* Hand Geometry.

→ Measures shape of hand using 16 measures

→ Quick and robust, suitable for many situations.

→ Advantages: Quick enrollment and recognition.

→ Disadvantages: Not suitable for very young or old individuals.

* Iris Patterns.

→ Iris recognition identifies individuals based on patterns in the ring shaped pupil of the eye.

→ considered a robust method.

· Registration Process.

① Locate eye and capture black and white photo.

② Process image using 2-D wavelet transform, resulting in a 256 byte Iris-code.

③ Compare Iris codes based on hamming distance between them.

④ Hamming distance is the number of non-matching bits divided by total number of bits compared.

5. Threshold of acceptance is used to determine match or non-match.

* Token Based Authentication.

→ A security token is a peripheral device used to gain access of an electronically restricted resource.

→ Generates a new random value everytime its used, serving as an alternative to passwords.

→ Working of Authentication Token Device.

① Creation of Token.

• Authentication Token Creation.

→ Authentication tokens are created by authentication server along with a random seed for the tokens.

→ The random seed is automatically placed or pre-programmed inside each token by the server.

Page No.
Date

② server Management-

→ The server keeps a copy of the seed against the user ID in the user database.

→ The seed can be conceptually considered as a user password, but the difference is that the user password is known to the user while seed value is unknown to the user.

③ Functionality

- Random seed: Each authentication token is pre-programmed with a unique number called a random seed. to ensure uniqueness of the output.

- Dynamic values.: The token generates a new random value every time its used for authentication.

x x

Page No.
Date

* Single Sign-On.

Definition: SSO login allows the user to log in to an application with a single set of credentials and then automatically be signed into multiple applications without need for login creds.

→ SSO is a property of access control for multiple-related but independent software systems.

→ Single Sign Off is a reverse property. When you log out of one software, you get logged out of all.

→ Key Benefits of SSO.

① Productivity Improvement: Eliminates the need to re-enter user creds.

② Password Fatigue Reduction: ~~Remembering~~ Remembering different usernames and passwords is an issue.

③ Cost Reduction: Reduces complaints and costs for password helpdesk.

Page No.
Date

* Working of SSO.

① User Access.

→ The user accesses a specific application (for eg- foo.com)

② Authentication server

→ user is redirected to the central authentication server., where an authentication-related cookie is generated.

③ Navigating to Other Apps.

→ user navigates to other applications hosted on different domains.

④ Redirection to Authentication server

→ For each new app, the user is redirected to the central authentication server.

⑤ Authentication Check.

→ The authentication server checks whether the user already has an authentication related cookie.

Page No.
Date

(5) Access Granted.

→ If the user has a valid cookie, the authentication server redirects the user back to the respective app., providing access without login.