Mod-5

Message Authentication & Digital Signature.

Quick Work
Page No.:
Date:
M T W T F S S

Symmetric Encryption for message Authentication

→ Assume only sender and Receiver share a key.
→ Single key for both encryption and decryption.
→ Sender encrypt plaintext using Receiver's secret key which can be later used by the Receiver to decrypt ciphertext.

Scheme 1:

A: Encrypt with $K_{public-B}[M]$, send to B.
B: Decrypt with $K_{private-B}$ to obtain M.

→ Authentication: ✗ Confidentiality: ✓

Scheme 2:

A: Encrypt with $K_{private-A}[M]$, send to B
B: Decrypt with $K_{public-B}$ to obtain M.

→ Authentication: ✓ Confidentiality ✗.

Scheme 3:

A: Encrypt with $K_{public-B}$ [Encrypt with $K_{private-A}[M]$]
B: Decrypt with $K_{privateB}$ then with $K_{public-A}$

→ Authentication: ✓ Confidentiality ✓

Compression :- A function that maps arbitrarily long binary strings to fixed length binary string.

Ease of Computation :- Given a Hash function and an input it should be easy to calculate the output

Hash Function Property

→ It is mathematically impossible to extract the original message from the digest.

→ A slight change to original message causes a drastic change in the Resulting digest.

→ The Result of the Hashing algorithm is always the same.

Cryptographic Hash Function Criteria

- Preimage Resistance
- Second preimage Resistance
- Collision Resistance

Quick Work
Page No.:
Date:
M T W T F S S

1] Preimage Resistance.

- A cryptographic hash function must be preimage Resistance.
- Given a Hash function h and $Y = h(m)$ it must be extreamly difficult for eve. to find message, m, such that $h(m') = Y$.
- This property protects against an attacker who only hash value and is trying to find the input.

2] Second preimage Resistance.

- This ensures that a message cannot easily be forged.
- Given a specific message and its digest it is impossible to create another message with same digest.
- Given M and $h(m)$
  Find M' such that $M \neq M'$ but $h(M) = h(M')$

3] Collision Resistance.

- It should be computationally infeasible to find two different input that produce the same hash output.
- Given Nothing
  Find M' such that $M \neq M'$ but $h(M)' = h(M)$

Quick Work
Page No.:
Date:
M T W T F S S

Birthday Attack:

- Is based on birthday paradox i.e. the probability that in a set of n Randomly Chosen people, some pair of them will have the same birthday.
- Applied to Hash Function attack, this mean you have 50% Chance to break the collision Resistance.

Features of SHA-512:

- A Hashing algorithm used to convert text of any length into a fixed size string.
- Each output produces a SHA-512 length of 512 bits (64 bytes).
- This algorithm is commonly used for
  - email addresses Hashing.
  - password Hashing.
  - Digital Record Verification.
- with a Message digest of 512 bit, SHA-512 expected to be resistant to all attacks, including Collision Attacks.

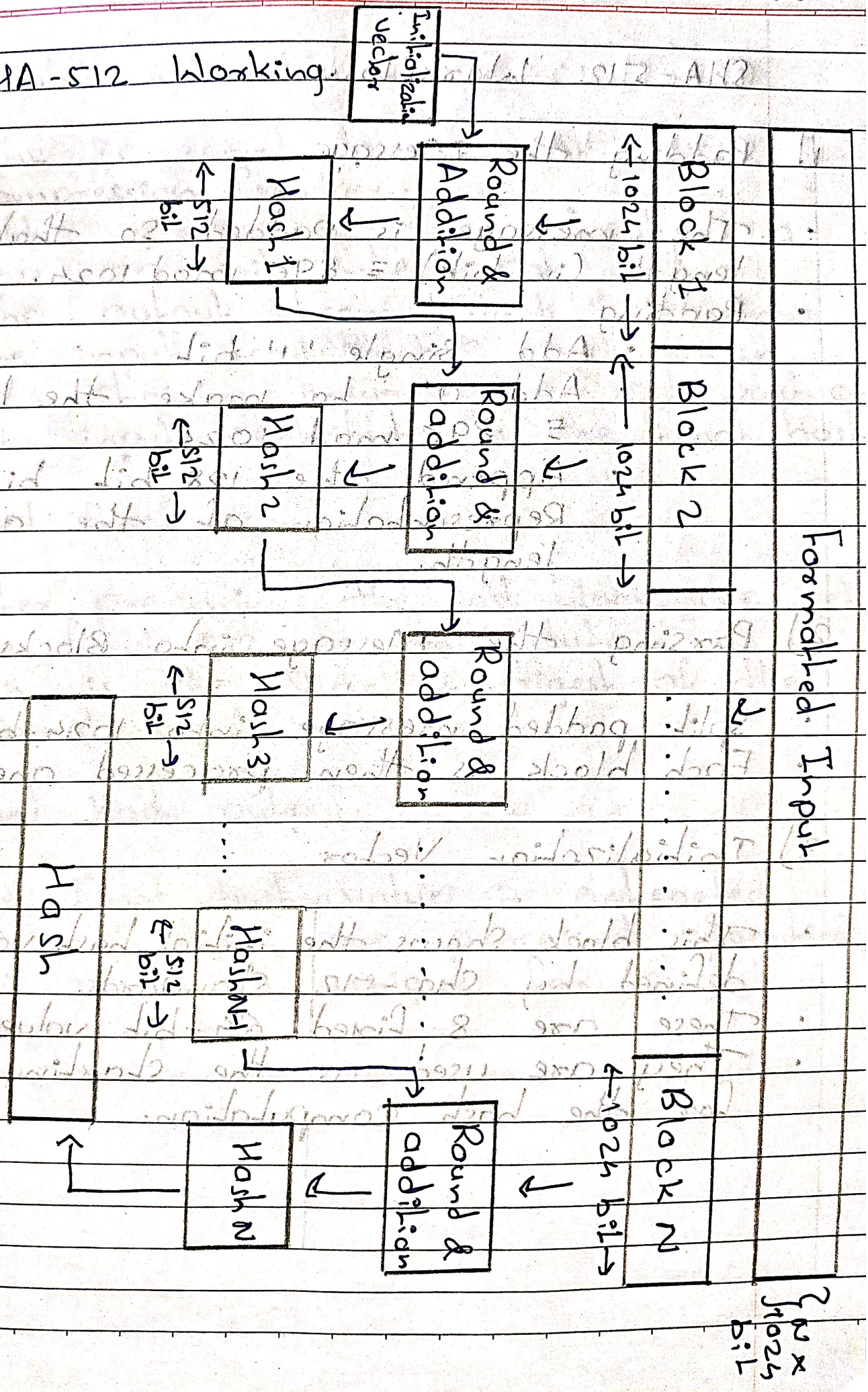
Quick Work

Page No.:

Date:

M T W T F S S

SHA-512 Working.



Quick Work
Page No.:
Date:
M T W T F S S

SHA-512 Workflow

1] Padding the Message.

- The message is padded so that its length (in bits) ≡ 896 mod 1024.
- Padding
  - Add single '1' bit
  - Add '0's to make the length ≡ 896 mod 1024
  - Append the 128 bit binary representation of the original length.

2] Parsing the Message into Blocks.

- Split padded message into 1024-bit blocks.
- Each block is then processed one by one

3] Initialization Vector

- This block shows the initial hash values defined by Sha-512 standards.
- These are 8 fixed 64-bit value [total 512bit]
- They are used as the starting point for the hash computation.

4] Round and addition.

- The 80 rounds of the SHA-512 compression function.
- After processing, the output is a new 512-bit intermidiate Hash.
- The output from Block 1 becomes the input for Block 2, and so on.
- This chaining ensures that each part of the input contributes to the final hash.

5] Final Block.

- After processing the last block, we get the final 512-bit hash output.
- This is the SHA-512 digest of the original input message.

6] Final Hash output.

- The Final hash value is extracted after all block are processed sequentially.
- This value is unique for every unique input.

Quick Work
Page No.:
Date:
M T W T F S S

MAC [Message Authentication code].

- Similar to message digest.
- Symmetric key cryptography is used.

working of MAC.

M ⟶ H, [MAC code]
↑
Shared key (K)

M + H → sent to Receiver.

Calculate this own MAC H_2 with key (K)

If $H_1 = H_2$ ⇒ No change in MAC.

$H_1 \neq H_2$ ⇒ Message is changed.

Significance of MAC.

1] Receiver can know if message is changed or not

2] Receiver has assurance that message is from correct Sender [because same key for Sender and Receiver].

HMAC. [Hased Based MAC].

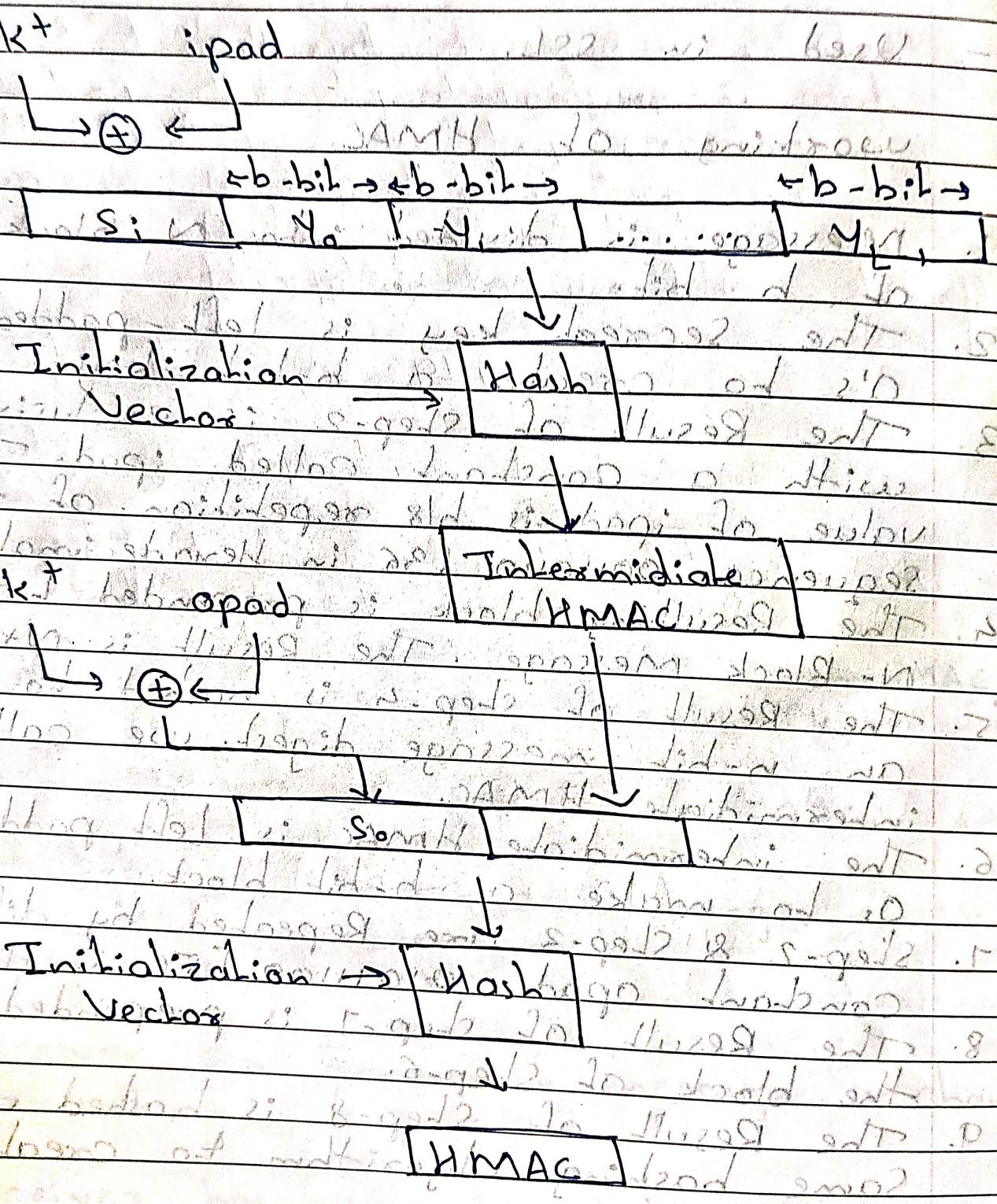
- Used in SSL.

working of HMAC.

1. Message is divided into N Block. each of b bit.
2. The Secreat key is left-padded with 0's to creake a b-bit key.
3. The Result of step-2 is exclusive-ored with a Constant called ipad. The value of ipad is b/8 repetition of the sequence 00110110 (36 in Hexadecimal).
4. The Resulting block is prepended to the N-Block Message. The Result is N+1 Blocks
5. The Result of step-4 is hashed to creake an n-bit message digest. we call it as intermidiate HMAC.
6. The intermidiate HMAC is left padded with 0s to make a b-bit block.
7. Step-2 & Step-3 are Repeated by different Constant opad 01011100 [5C in Hexadecim
8. The Result of step-7 is prepended to the block of step-6.
9. The Result of step-8 is hashed with Same hashing algorithm to creake Final n-bit HMAC.

Quick Work
Page No.:
Date:
M T W T F S S

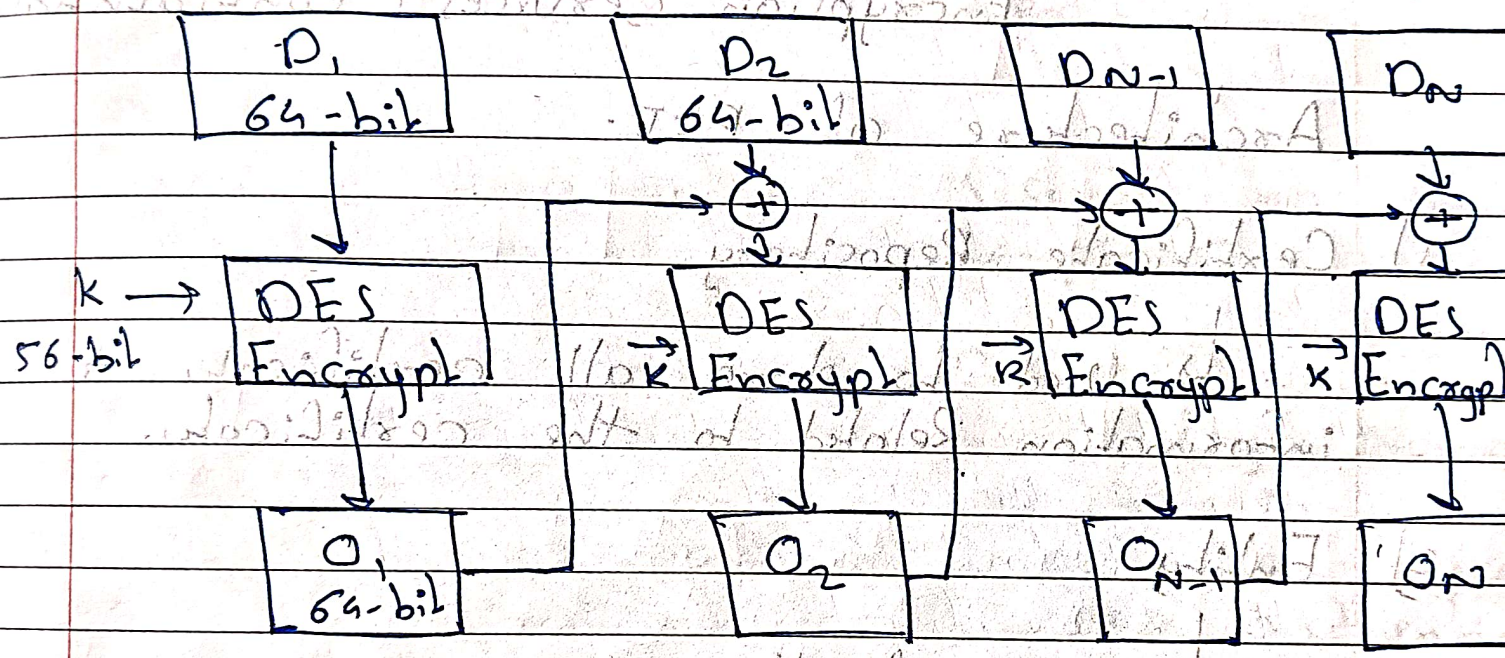
Diagram:



Quick Work
Page No.:
Date:
M T W T F S S

DAA [Data Authentication Algorithm].

- Based on DES
- Cipher chaining mode with initialization vector of zero.



- $D_1, D_2, \ldots D_n$. If Necessary, the final block is padded on the right with zeros to form a full 64-bit block.

$$O_1 = E[K, D_1]$$
$$O_2 = E[K, [D_2 \oplus O_1]]$$
$$\vdots$$
$$O_N = E[K, [D_N \oplus O_{N-1}]]$$

Quick Work
M T W T F S S

PKI [Public key Infrastructure].

- Standard followed for managing storing and Revoking the digital certificates.
- Follows Asymmetric key cryptography.
- Include : Message digest [Integrity]
  Digital Signature [Authentication, Non-Repu.]
  Encryption services [Confidentiality]

Architecture of PKI.

1] Certificate Repository
   ↳ Used to store all certificate + information Related to the certificate.

2] Entity
   ↳ User of the PKI

3] Registration Authority [RA]
   ↳ It is for Registration and verification purpose.

4] Certification Authority [CA].
   ↳ It will decide to give the certificate to the user with time duration.

Quick Work
Page No.:
Date:
M T W T F S S

Digital Signature	Digital Certificates.
Proves the authenticity & integrity of a message.	Verifies the identity of sender.
Created by Sender	Issued by CA Certification Authority.
Valid as long as keys are trusted	Usually has an expiration date.
Uses private key to create the Signature	Contains your public key + Name or domain.
Used in email Signing, document Verification, Blockchain	SSL/TLS, authentication, VPN, etc.

Quick Work
Page No.:
Date:
M T W T F S S

Kerberos.

- It is a Network authentical protocol.
- Client Server Architechture.
- Symmetric key
- Requires 3rd party for Key.

Key distribution Centre (KDC)	
Authentication Server	Ticket Granting Server

User A	Network Service

Step1. Login Request to Authentication Server.
- User A wants to login.
- Sends username to Authentication Server.

Step2: AS verifies Identity.
- As checks if user A exists.
- If Yess, it Send back:
  - A Ticket Granting Ticket
  - A Session key for user

Step3: Requesting a Service Ticket.
- User A now wants to use a Network Service
- Sends the TGT + Service Request to TGS

Step 4: TGS check Servic Ticket.

- TGS check TGT
- If valid, it sends back Service Ticket to user A.

Step5: Access Network Services.

- User A sends the service ticket to Network service
- If valid, Access is granted.