4 Authentication Protocols and Digital Signature Schemes

Syllabus

At the end of this unit, you should be able to understand and comprehend the following syllabus topics :

- User Authentication and Entity Authentication
  - Password Based
  - Challenge Response Based
    - Needham Schroeder Authentication protocol
    - Kerberos Authentication protocol
- Digital Signature
  - Attacks on Digital Signature
  - Digital Signature Scheme: RSA

4.1 User and Entity Authentication

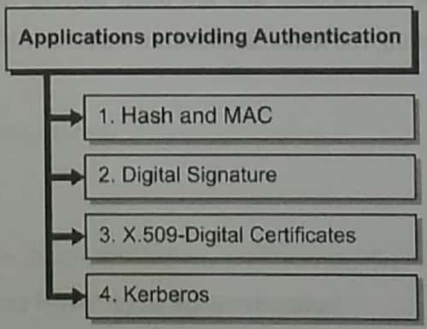

Applications providing Authentication
1. Hash and MAC
2. Digital Signature
3. X.509-Digital Certificates
4. Kerberos

Fig. 4.1.1 : Applications provide Authentication

Authentication is a way to build trust in communicating parties or systems. Authentication serves as the crucial protection mechanism towards securing a communication – ensuring that the right set of parties are involved in the communication and no one else.

Broadly speaking, there are four ways that provide major authentication services today. You have learnt about the first three so far. Let's learn the fourth one - Kerberos. But before that, let's learn about some remote user authentication principles and the ways in which various authentication mechanisms work.

Remote User Authentication Principles

Overtime, various authentication methods have evolved to address :

- The ease of authentication.
- Make it hard to break authentication.
- Make authentication techniques suitable for various devices.

At a high level, authentication methods are categorised as following.

4.1.1 Types of Authentication Methods

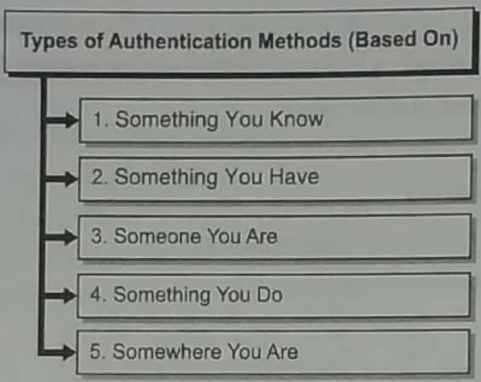


Fig. 4.1.2 : Types of Authentication Methods

1. Something You Know

> ✍ ***Definition** : The authentication methods based on "Something You Know" rely upon your secret knowledge about something.*

These secrets could be memorised or practiced and are often easy to recall for you. Some of the examples of authentication methods based on "Something You Know" are;

- Passwords
- PIN
- Passphrases
- Secret questions
  - Mother's maiden name
  - First pet name
  - First car purchase year
  - First school name
  - City in which you were born
  - Or other secret questions whose answers would be likely known just to you
- Lock key combination

Advantages of "Something You Know" type authentication

- Easy to implement by developers in the product (OS, Applications or Websites)
- Easy to recall for the user
- Easy to authenticate for the user
- Easy to change for the user
- Chances of errors are low
- Can be used by non-person entities (hardware devices, applications, OS, programs, processes, etc.)

Disadvantages with "Something You Know" type authentication

Easy to crack (or break)

2. Something You Have

✍ ***Definition :*** *The authentication methods based on "Something You Have" rely upon your possession of something.*

You could possess something that could let you authenticate using it. You often don't require to remember anything while using it for authentication purpose. Some of the examples of authentication methods based on "Something You Have" are,

- Physical keys
- Badge
- Swipe Card (for example, Debit or Credit Card)
- Digital Certificates
- Security Keys (for example, Private Key)
- OTP (that you get via SMS)
- Tokens

Advantages of "Something You Have" type authentication

- Easy to use
- Does not often require remembering secrets
- Chances of errors are low
- Not easy to crack
- Can be used by non-person entities (hardware devices, applications, OS, programs, processes, etc.)

Disadvantages of "Something You Have" type authentication

- Difficult to change
- Possibility of loss or theft
- Requires distribution methods (provisioning) to reach to the user securely.

3. Someone You Are

✍ ***Definition :*** *The authentication methods based on "Someone You Are" rely upon your physical characteristics.*

Your body has several physical characteristics that can be used to uniquely identify and authenticate you. These characteristics do not much change over time (as you age) and can serve authentication purpose for near lifetime. These characteristics are called Static Biometrics.

Definition : *Static Biometrics are physical characteristics that can be used for authentication.*

Generally speaking,

Definition : *The measurement and analysis of unique physical or behavioural characteristics is called biometrics.*

However, there can be scenarios where re-provisioning (re-calibrating) your biometric details might be required. For example, if you use a particular finger for fingerprint and your that particular finger is damaged permanently, you might have to choose another finger or another type of biometrics for authentication. Some of the examples of authentication methods based on "Someone You Are" are;

- Fingerprint
- Palm Scan
- Hand Geometry
- Retina Scan
- Iris Scan
- Facial recognition

Advantages of "Someone You Are" type authentication

- Easy to use
- Does not often require remembering secrets
- Difficult to crack

Disadvantages of "Someone You Are" type authentication

- Difficult to implement correct.
- Chances of errors are high (recall you trying fingerprints several times at Aadhar enrolment centre?).
- Difficult to change (requires physical presence for re-provisioning).
- Cannot be used by non-person entities (hardware devices, applications, OS, programs, processes, etc.).

4. Something You Do

Definition : *The authentication methods based on "Something You Do" rely upon your way of performing a given task.*

This authentication method also collects biometric patterns but while performing a given task. Unlike static biometric patterns, these are dynamic biometric patterns which are used for authentication purpose.

Definition : *Dynamic Biometrics are job performing characteristics that can be used for authentication while an individual is performing a given task.*

Some of the examples of authentication methods based on "Something You Do" are,

- Voice print (or pattern).
- Keystroke Dynamics (how hard you press the keys and how fast).
- Handwriting characteristics (remember old movies where handwriting matches were done?).

Advantages of "Something You Do" type authentication

- Easy to use.
- Does not often require remembering secrets.

Disadvantages of "Someone You Are" type authentication

- Difficult to implement correct.
- Not too difficult to crack (consider replaying a recorded voice).
- Chances of errors are high (have you tried voice to text yet?).
- Difficult to change (requires physical presence for re-provisioning).
- Cannot be used by non-person entities (hardware devices, applications, OS, programs, processes, etc.).

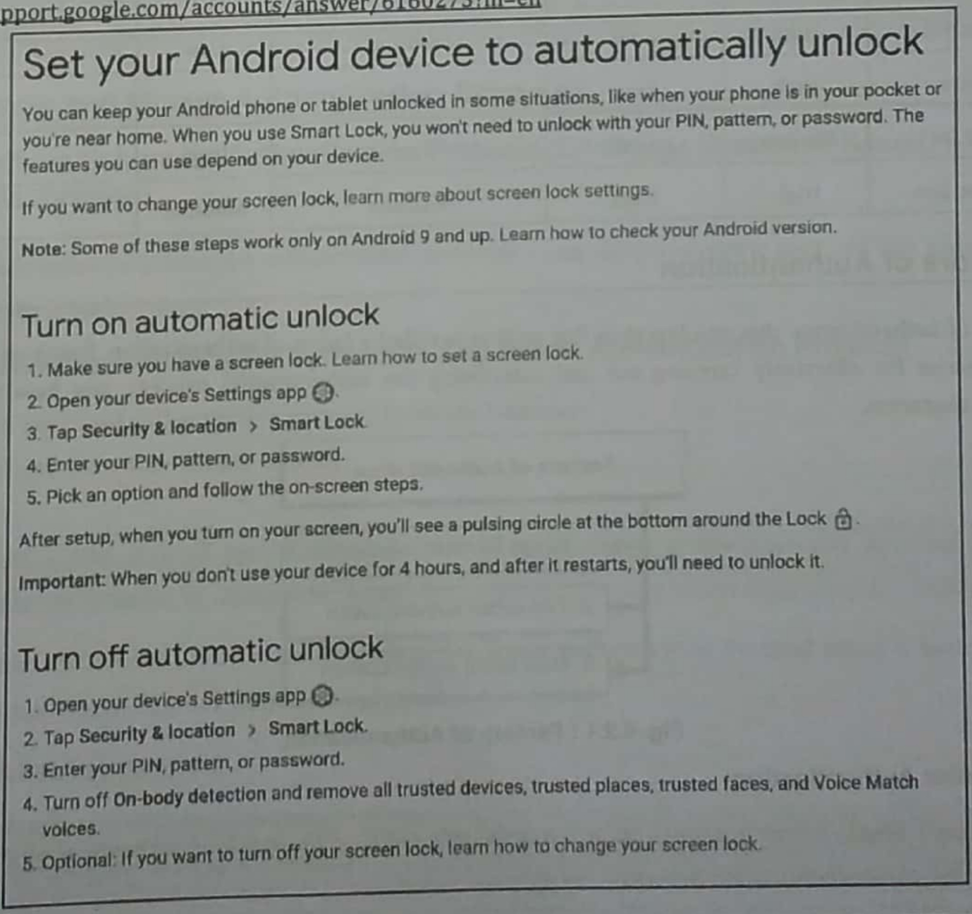
5. Somewhere You Are

✍ ***Definition :*** *The authentication methods based on "Somewhere You Are" rely upon your physical location.*

Increasingly, the devices and systems have location awareness. Your mobile phone or laptops (via network connectivity) precisely know where you are located at a particular moment. "Somewhere You Are" uses the location information for authentication.

For example, Google Smart Lock for Android allows you to set Trusted Locations. Say, if you are at home, your phone may not require you to unlock it (via PIN, pattern, password or fingerprint) before use.

https://support.google.com/accounts/answer/6160273?hl=en



Set your Android device to automatically unlock

You can keep your Android phone or tablet unlocked in some situations, like when your phone is in your pocket or you're near home. When you use Smart Lock, you won't need to unlock with your PIN, pattern, or password. The features you can use depend on your device.

If you want to change your screen lock, learn more about screen lock settings.

Note: Some of these steps work only on Android 9 and up. Learn how to check your Android version.

Turn on automatic unlock

1. Make sure you have a screen lock. Learn how to set a screen lock.
2. Open your device's Settings app ⚙.
3. Tap **Security & location** > **Smart Lock**.
4. Enter your PIN, pattern, or password.
5. Pick an option and follow the on-screen steps.

After setup, when you turn on your screen, you'll see a pulsing circle at the bottom around the Lock 🔒.

Important: When you don't use your device for 4 hours, and after it restarts, you'll need to unlock it.

Turn off automatic unlock

1. Open your device's Settings app ⚙.
2. Tap **Security & location** > **Smart Lock**.
3. Enter your PIN, pattern, or password.
4. Turn off **On-body detection** and remove all trusted devices, trusted places, trusted faces, and Voice Match voices.
5. Optional: If you want to turn off your screen lock, learn how to change your screen lock.

"Somewhere You Are" is also quite widely used in corporate IT. In many environments, if you are on an office network (LAN or Wi-Fi), you can login using only a password, but if you are out of the office you must use VPN or an additional mechanism for authentication.

Advantages of "Somewhere You Are" type authentication

- Easy to use.
- Does not often require remembering secrets.
- Can be used for authenticating non-person entities (mobile, laptops or other location aware devices).

Disadvantages of "Somewhere You Are" type authentication

- Not too difficult to crack (consider theft and someone carrying the device to a trusted location).
- Cannot be used for authenticating individuals.
- Requires network connectivity for location awareness.

4.1.2 Comparison between the Authentication Types

Table 4.1.1 : Comparison between the Type of Authentication

Type of authentication	Ease of use	Ease of Change	Ease of implementation	Error rate	Support non-person entities
Something You Know	High	High	High	Low	Yes
Something You Have	High	Low	Medium	Low	Yes
Someone You Are	High	Low	Low	High	No
Something You Do	Medium	Low	Low	High	No
Somewhere You are	High	High	Medium	Medium	Yes

4.2 Factors of Authentication

Each type of authentication that you learnt in this section is called a factor of authentication. Based on the number of factors you choose for effectively carrying out and completing the authentication process, you have three types of authentication scenarios.

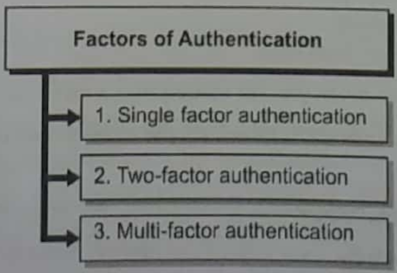

Factors of Authentication
1. Single factor authentication
2. Two-factor authentication
3. Multi-factor authentication

Fig. 4.2.1 : Factors of Authentication

1. Single Factor Authentication

✍ ***Definition :*** *Single factor authentication requires only ONE of the types of authentication for successfully carrying out the authentication process.*

For example, you could just use password or token. This is most widely used. It is often treated as a weak form of authentication.

2. Two-factor Authentication

> ***Definition :*** *Two-factor authentication requires you to use any TWO types of authentication, one after another, for successfully carrying out the authentication process.*

It is considered as a strong form of authentication. For example, for your online transactions, you are first required to give the account password and then you receive an OTP. You are required to put the correct OTP for successfully authenticating your account details and carrying out and completing your transaction. Another example could be ATM. You must possess you Debit Card (Something You Have) and put in the right PIN (Something You Know) for withdrawing money.

3. Multi-factor Authentication

> ***Definition :*** *Multi-factor authentication requires you to use MORE than two types of authentication.*

It is often used in a high security environment. For example, you may be first required to give your fingerprint, after which you can access an application where you are required to provide username and password. For carrying out a transaction on that application, you might require an OTP.

> **Note :** If two authentication techniques from the same type of authentication are used, it is not considered two-factor authentication. It would be considered a single factor authentication. For example, you cannot consider consecutive requirement of two passwords or a password and a PIN to be two-factor authentication. Two-factor authentication essentially requires two different ways to authenticate.

4.3 One-way and Mutual Authentication Schemes

4.3.1 Needham Schroeder Authentication Protocol (Challenge Response Based Authentication)

Needham Schroeder proposed two authentication protocols – one using symmetric keys and one using asymmetric keys. Let's learn about both of them.

4.3.1(A) The Needham–Schroeder Symmetric Key Based Authentication Protocol

In symmetric key based authentication protocol, there are 3 entities :

- 2 users – let's call them Alice (A) and Bob (B)
- 1 Server (S)

> ***Definition :*** *The goal of this protocol to generate and share a key that can be used for securing communication between the two users - A and B.*

Note here that the Needham–Schroeder Symmetric Key Based Authentication Protocol forms a basis for Kerberos based authentication. It solves the key distribution problem.

Assume the following primitives :

- A and B are identities of Alice and Bob respectively,
- K_{AS} is a symmetric key known only to A and S,

- K_{BS} is a symmetric key known only to B and S,
- N_A and N_B are nonce (random number used once) generated by A and B respectively,
- K_{AB} is the symmetric key that needs to be generated and shared between A and B for secure communication.

Protocol Operation

Following is the sequence of activities that the protocol follows.

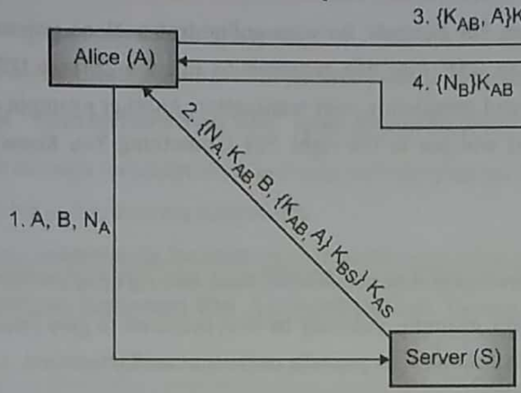


Fig. 4.3.1 : Protocol Workflow

1. Alice sends a message to the server identifying herself and telling the server that she wants to communicate with Bob.
2. The server generates K_{AB} and sends it to Alice encrypting the entire response with K_{AS}.
   (a) A copy of K_{AB} is encrypted using K_{BS} for Alice to forward to Bob and also identify herself and,
   (b) A copy for Alice herself.
3. Alice forwards the key K_{AB} to Bob. Bob decrypts it with the key K_{BS}.
4. Bob sends Alice a nonce N_B encrypted using K_{AB} to show that he has the key.
5. Alice performs a simple operation on the nonce N_B, re-encrypts it and sends it back verifying that she is still alive and that she holds the key as well.

Attack on the protocol

This protocol is vulnerable to replay attack. The attacker can grab the older and compromised value for K_{AB}. He can then replay the message $\{K_{AB}, A\}K_{BS}$ to Bob, who will accept it. Kerberos solves this problem by adding timestamp to avoid replaying older communication.

4.3.1(B) The Needham–Schroeder Asymmetric Key Based Authentication Protocol

In asymmetric key based authentication protocol, there are 3 entities as well

- 2 users – let's call them Alice (A) and Bob (B)
- 1 Server (S)

✍ ***Definition :*** *The goal of this protocol is to share the respective public keys between the two users - A and B.*

Assume the following primitives:

- 3 key pairs : P stands for Public Key, Q stands for Private Key,
- K_{PA} and K_{QA} : Public and Private Keys of A respectively,
- K_{PB} and K_{QB} : Public and Private Keys of B respectively,

- K_{PS} and K_{QS}: Public and Private Keys of S respectively,
- K_{PS} is known to both A and B and is trusted.

Protocol Operation

Following is the sequence of activities that the protocol follows.

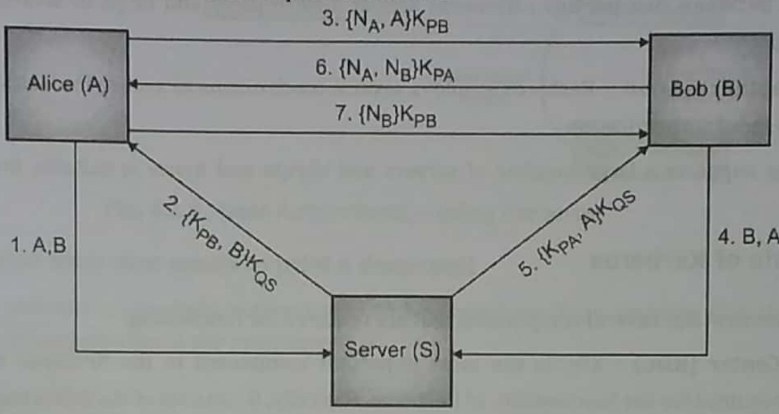


Fig. 4.3.2 : Protocol Workflow

1. A requests B's public keys from S.
2. S responds with B's public key K_{PB} alongside B's identity, encrypted by the server's private key K_{QS}.
3. A chooses a random nonce N_A and sends it to B by encrypting it using B's public Key K_{PB} received in step 2.
4. B now knows that A wants to communicate. So, B requests A's public keys from S.
5. S responds with A's public key K_{PA} alongside A's identity, encrypted by the server's private key K_{QS}.
6. B chooses a random nonce N_B and sends it to A along with N_A to prove his ability to decrypt with K_{QB}.
7. A confirms N_B to B, to prove her ability to decrypt with K_{QA}.

Attack on the protocol

This protocol is vulnerable to Man-in-the-Middle attack. The attacker can make A and B believe that they are communicating. This could be fixed by updating the step 6 by passing along the identity - $\{N_A, N_B, B\}K_{PA}$. So, A would know who she is actually communicating with instead of being attacked by the middle-man.

4.3.2 Kerberos Authentication Protocol

Q.	Explain working of Kerberos.	MU - May 16, 10 Marks
Q.	Explain Kerberos protocol that supports authentication in distributed system.	MU - May 18, 10 Marks
Q.	Write short notes on Kerberos.	MU - May 19, 4 Marks

✍ ***Definition :** Kerberos is a network-based authentication protocol.*

It was developed around mid-1980s at MIT. It works on the client/server model and uses symmetric key cryptography. Today, Kerberos is extensively used for authentication in Microsoft Windows, Unix, Linux and Apple OS.

4.3.2(A) Problems Addressed by Kerberos

1. **Sharing passwords over the network :** With Kerberos, you need not share passwords over the network for authentication.
2. **Establishing trust between two parties :** Kerberos acts as a third party and helps to establish trust between two non-trusting parties.
3. **Difficult to spoof authentication :** Kerberos employs several mechanisms to secure the authentication process and makes it difficult to spoof authentication.
4. **Scalable :** Kerberos supports a large number of servers and clients and hence is suitable for distributed network architectures.

4.3.2(B) Components of Kerberos

The Kerberos environment has several components that are required for functioning.

1. **Key Distribution Center (KDC) :** KDC is the most important component in the Kerberos environment. It holds all the information required for the functionality of Kerberos. Basically, it consists of the following sub-components.
   - (a) **Authentication Service (AS) :** The Authentication Service (AS) issues TicketGranting Tickets (TGTs) that is used to connect to the Ticket Granting Service (TGS). It also verifies principals that require authentication.
   - (b) **Ticket Granting Service (TGS) :** The TGS issues the tickets to the clients using which the clients can connect to the desired server.
   - (c) **Principals :** In the Kerberos terminology, Principals can be users, clients, servers, applications or network services that require authentication. The KDC has the information about each principal account and its secret key. For example, a user could be a principal requiring access to a print server that could be another principal.
   - (d) **KDC Database :** All the principal related information is stored in the KDC database.
   - (e) **Realm :** In Kerberos terminology, a realm is the set of principals who can authenticate to each other. It is a logical grouping of principals. One KDC can have one realm or several realms.
2. **Client :** Typically, a client is the principal that requires to authenticate to another principal (server).
3. **Server :** Typically, a server is the principal that holds resources that the client is interested in and provides the resources that can be consumed after successful authentication.

4.3.2(C) How does it work?

Kerberos heavily uses the concept of tickets that works very much like your train ticket. A ticket is just a temporary proof. Let's understand the working in detail.

Scenario 1 - User authenticating to a computer

1. The user enters the username and password on the computer.
2. The Kerberos software running on the computer sends the username to the Authentication Service (AS).
3. The Authentication Service checks if the username is present. If yes, it sends back a Ticket Granting Ticket (TGT) which is encrypted with the user's pre-shared secret key (password).
4. If the user entered the correct password, she can decrypt the TGT and then is granted access to the computer.

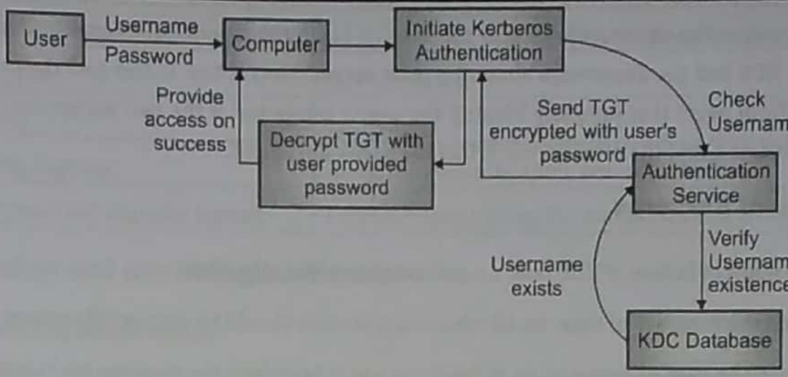

User
Username
Password
Computer
Initiate Kerberos Authentication
Provide access on success
Decrypt TGT with user provided password
Send TGT encrypted with user's password
Check Username
Authentication Service
Verify Username existence
Username exists
KDC Database

Fig. 4.3.3 : User Authentication using Kerberos

Scenario 2 - Authenticated User now wants to print a document

From Scenario 1, the user has successfully authenticated to her computer. Now, suppose that she wants to print a document and hence needs to authenticate to the print server.

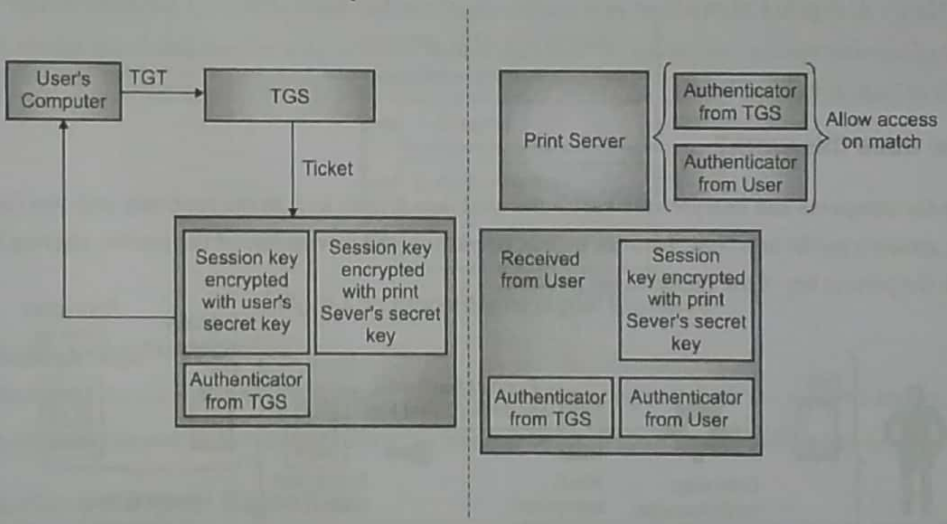


Fig. 4.3.4 : User Authenticating to a Service using Kerberos

The following steps are taken :

1. The computer sends the TGT it received earlier (when the user wanted to authenticate) to the TGS. The TGT is a proof for the TGS that the user has already been successfully authenticated (as in scenario 1).
2. TGS creates a new ticket and puts two copies of the same session key (temporary secret key) in it. It encrypts the first copy of the session key with the user's secret key and the second copy with print server's secret key. This ticket also contains an authenticator information that holds the value of the user's computer's IP Address, sequence number and timestamp from where the TGT came. It then sends the ticket to the user's computer.
3. The user decrypts the ticket created by the TGS with her secret key and obtains the session key. It adds another set of authenticator information (computer's IP Address, sequence number and timestamp) to it and sends the ticket to the print server.

4. The print server receives the ticket and extracts the session key by decrypting it. It knows that the KDC created the ticket because only KDC had the knowledge about the print server's secret key. It also gets the two authenticators (one from TGS and one from user) that uniquely identify the user's computer. If the two authenticators match, the user is successfully authenticated, and the print server prints the user's document.

4.3.2(D) Limitations of Kerberos

1. KDC can be a single point of failure. If KDC fails, no authentication can take place.
2. Kerberos depends on the accuracy of time. So, all clients and servers should be time synchronized.
3. Session keys are stored on user's computer. So, if the computer is breached, the sessions key might be stolen.

4.4 Digital Signature

Q. What is a digital signature? **MU - May 16, Dec. 16, 4 Marks**

✍ ***Definition :*** *A digital signature is a hash value that has been encrypted with the sender's private key. The act of signing means encrypting the message's hash value with a private key (since no one else knows the sender's private key).*

4.4.1 How does this work?

So, the sender computes and encrypts the hash value with her private key. At the receiving end, you decrypt the hash value with the sender's public key. Now, because no one else knows the private key of the sender, altering hash value and re-signing with the private key of the sender is not possible.

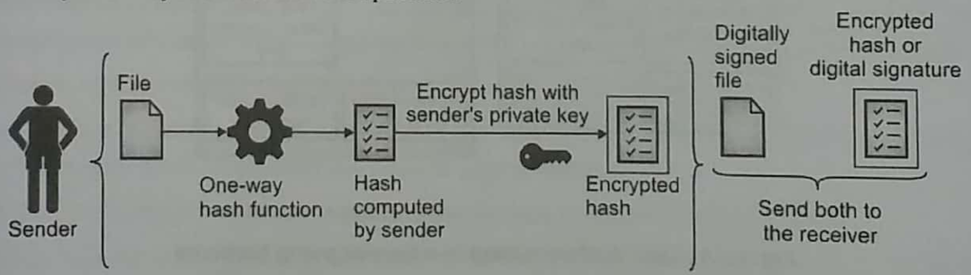

Sender
File
One-way hash function
Hash computed by sender
Encrypt hash with sender's private key
Encrypted hash
Digitally signed file
Encrypted hash or digital signature
Send both to the receiver

Fig. 4.4.1 : Digital Signature Generation at the Sender's end

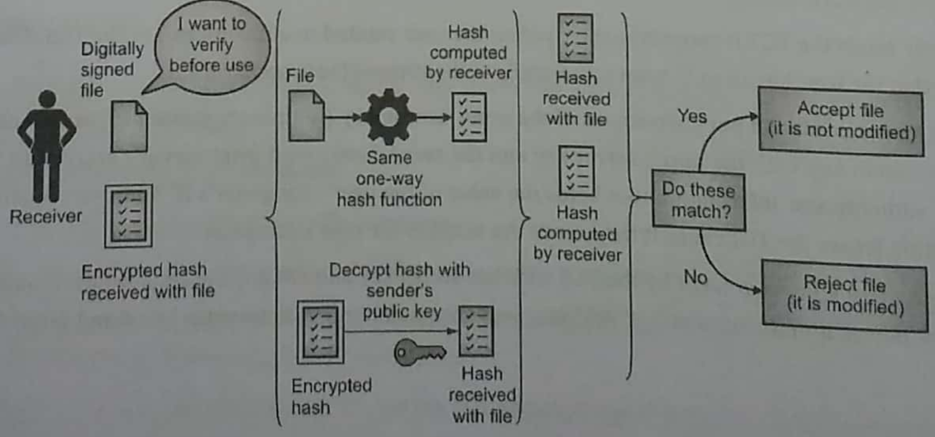

I want to verify before use
Digitally signed file
Receiver
Encrypted hash received with file
File
Same one-way hash function
Hash computed by receiver
Decrypt hash with sender's public key
Encrypted hash
Hash received with file
Hash received with file
Hash computed by receiver
Do these match?
Yes
Accept file (it is not modified)
No
Reject file (it is modified)

Fig. 4.4.2 : Digital Signature verification at the Receiver's end

Processing applied on a message	Security Property achieved
Encryption	Confidentiality
Hashing	Integrity
Digitally Signing	Integrity, authentication, non-repudiation
Encryption and digitally signing	Confidentiality, Integrity, authentication, non-repudiation

4.4.2 Application and use of Digital Signature

1. Sending and receiving secure emails.
2. Signing documents. For example, you can sign income tax returns using digital signature.
3. Sending and receiving important files. For example, insurance policy documents, Aadhar card e-letter, etc.

4.4.3 Properties of Digital Signature

Digital signature provides three security properties.

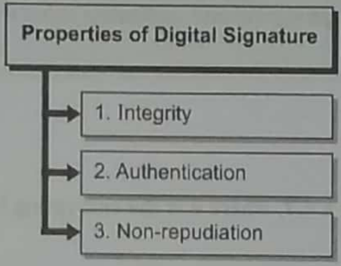


Fig. 4.4.3 : Properties of Digital Signature

1. **Integrity :** Via hash value calculation.
2. **Authentication :** Via the ability to prove sender's identity by decrypting hash with the sender's known public key.
3. **Non-repudiation :** Sender cannot deny sending the message because she used her private key to encrypt the hash.

4.5 Attacks on Digital Signature

Attacks on digital signature are a combination of attacks on cryptosystems and hashing. Please refer to those sections for a refresher.

4.6 RSA Digital Signature Scheme

Q. Explain any digital signature algorithm in detail. **MU - May 16, Dec. 16, 6 Marks**

✍ ***Definition :** RSA signatures are based on public key cryptography.*

RSA uses public key cryptography for creating and verifying digital signatures.

Key Generation

RSA digital signatures work on public and private key pairs. They can be generated by the regular key pair generating method by a Certificate Authority (CA) or on the user's system by herself. Recall from your reading on RSA algorithm that the keys are as following :

- The public key = *(n, e)*
- The private key = *(n, d)*

Message Signing

To sign a message, M

- Calculate the hash value of the message M at sender's end

  h = hash(M)

- Encrypt *h* using RSA private key

  Signature S = $(h)^d$ mod n

Signature Verification

- Decrypt Signature *S* using public key

  h' = $(S)^e$ mod n

- Calculate the hash value of the message M at receiver's end

  h = hash(M)

- If *h* = *h'*, the signature is valid else the signature is invalid

Ex. 4.6.1 : Alice chooses public key as (7, 33) and B chooses public key as (13, 221). Calculate their private keys. A wishes to send message m = 5 to B. Show the message signing and verification using RSA digital signature.

MU - May 19, 10 Marks

Soln. :

For Alice

Public key is (7, 33). Hence, n = 33 and e = 7, where e is the encrypting key. Let's calculate Alice's private key which would give us d, decrypting key.

Assume that 3 and 11 were the chosen prime factors for n = 33.

n = p * q = 3 * 11 = 33

r = (p – 1) * (q –1) = (3 –1) * (11– 1)

r = 2 * 10 = 20

d = e^{-1} mod r

ed ≡ 1 mod 20

7d ≡ 1 mod 20

Let's calculate modulo inverse using extended Euclidean algorithm (swapping 7 and 20)

Index i	quotient q for i	Remainder r for i	s for i	t for i
0		20	1	0
1		7	0	1
2	20 / 7 = 2	20 – 7 *2 = 6	1 – 0*2 = 1	0 – 1*2 = –2
3	7 / 6 = 1	7 – 6 * 1 = 1	0 – 1 * 1 = –1	1 – (–2)*1 = 3
4	6 / 1 = 6	6 – 1 * 6 = 0	Do not calculate	Do not calculate

- Let's re-swap the values.

  x = 3

  y = –1

- Putting the values in the extended Euclidean algorithm, you get

  ax + by = 1

3*7 + 20*(–1) = 1

- Since, you have to find multiplicative inverse in mod 20, divide both sides by mod 20.

3*(7) mod 20 + 20*(–1) mod 20 = 1 mod 20

3*(7) mod 20 + 0 = 1 mod 20

3*(7) mod 20 = 1 mod 20

- Hence, multiplicative inverse is 3.
- Therefore, the value of decrypting key, *d* = 3.
- Hence, the Private Key of user Alice is (3, 33).

For User B

Public key is (13, 221). Hence, n = 221 and e = 13, where e is the encrypting key. Let's calculate B's private key which would give us *d*, decrypting key.

Assume that 13 and 17 were the chosen prime factors for n = 221.

n = p * q = 13 * 17 = 221

r = (p – 1) * (q –1) = (13 –1) * (17– 1)

r = 12 * 16 = 192

d = e^{-1} mod r

ed ≡ 1 mod 192

13d ≡ 1 mod 192

Let's calculate modulo inverse using extended Euclidean algorithm (swapping 13 and 192)

Index i	quotient q for i	Remainder r for i	s for i	t for i
0		192	1	0
1		13	0	1
2	192 / 13 = 14	192 – 13 * 14 = 10	1 – 0*14 = 1	0 – 1*14 = –14
3	13 / 10 = 1	13 – 10 * 1 = 3	0 – 1 *1 = –1	1 – (–14)*1 = 15
4	10 / 3 = 3	10 – 3 * 3 = 1	1 – (–1) * 3 = 4	–14 – 15 * 3 = –59
5	3 / 1 = 3	3 – 1* 3 = 0	Do not calculate	Do not calculate

Let's re-swap the values.

x = –59

y = 4

- Putting the values in the extended Euclidean algorithm, you get

ax + by = 1

13*(–59) + 192*(4) = 1

- Since, you have to find multiplicative inverse in mod 192, divide both sides by mod 192.

13*(–59) mod 192 + 192*(4) mod 192 = 1 mod 192

13*(–59) mod 192 + 0 = 1 mod 192

13*(–59) mod 192 = 1 mod 192

Recall our discussion on calculating mod for negative numbers. You need to keep adding mod until the number turns positive and then calculate mod on the positive number you got.

–59 + 192 = 136

136 mod 192 = 136

- Hence, multiplicative inverse is 136.
- Therefore, the value of decrypting key, d = 136.
- Hence, the Private Key of user B is (136, 221).

Now, the question isn't asking you to encrypt and decrypt the plaintext message 5. It is asking you to show the message signing and verification using RSA digital signature. Hence, do not encrypt the plaintext message 5. It is not required.

- As per RSA digital signature,
- Message Signing
- To sign a message, M
- Calculate the hash value of the message M at sender's end
- $h = hash(M)$
- Encrypt h using RSA private key
- Signature $S = (h)^d \bmod n$
- Signature Verification
- Decrypt Signature S using public key
- $h' = (S)^e \bmod n$
- Calculate the hash value of the message M at receiver's end
- $h = hash(M)$
- If $h = h'$, the signature is valid else the signature is invalid

Since, hash algorithm or the hash value of the message to be digitally signed is not given, let's assume that the hash value h of the plaintext message 5 is 12. Let's create and verify the digital signature using the RSA digital signature scheme.

To sign a message, M

- Encrypt h using RSA private key
- Signature S = $(h)^d$ mod n

User A wishes to send the message to B. Hence, you would use A's private key (3, 33) for encrypting the hash value of the message. The encryption key is 3 from the private key. We have assumed that 12 is the hash value of the plaintext message 5.

S = 12^3 mod 33

S = 12

User A would send the message and its signature S = 12 to user B.

Now, user B can verify the signature as

- Decrypt Signature S using public key
- h' = $(S)^e$ mod n
- The public key of the user A is (7, 33).

h' = 12^7 mod 33

h' = 12

- You find that h = h' = 12. Hence, the digital signature is verified.

Review Questions

Here are a few review questions to help you gauge your understanding of this chapter. Try to attempt these questions and ensure that you can recall the points mentioned in the chapter.

[A] Computer Security Concepts

Q. 1 Describe the various categories of authentication methods with examples. **(8 Marks)**

Q. 2 Describe "Something You Know" as an authentication method category and list its advantages and disadvantages. **(8 Marks)**

Q. 3 Describe "Something You Have" as an authentication method category and list its advantages and disadvantages. **(8 Marks)**

Q. 4 Describe "Someone You Are" as an authentication method category and list its advantages and disadvantages. **(8 Marks)**

Q. 5 Describe "Something You Do" as an authentication method category and list its advantages and disadvantages. **(8 Marks)**

Q. 6 Describe "Somewhere You Are" as an authentication method category and list its advantages and disadvantages. **(8 Marks)**

Q. 7 Provide a comparison between the various authentication method types. **(8 Marks)**

Q. 8 Describe the term "Factors of Authentication" and give examples. **(6 Marks)**

[B] Needham Schroeder Authentication Protocol

Q. 9 With a block diagram, describe the Needham–Schroeder Symmetric Key Based Authentication Protocol. **(8 Marks)**

Q. 10 With a block diagram, describe the Needham–Schroeder Asymmetric Key Based Authentication Protocol. **(8 Marks)**

[C] Kerberos Authentication Protocol

Q. 11 What is Kerberos? List the problems it addresses. **(8 Marks)**

Q. 12 Describe the various components of Kerberos and explain how it works. **(8 Marks)**

Q. 13 With a block diagram, explain how a user can authenticate to a computer using Kerberos. **(8 Marks)**

Q. 14 With a block diagram, explain how an authenticated user can authenticate to a print server using Kerberos for printing a document. **(8 Marks)**

Q. 15 What is Kerberos? List its limitation. **(4 Marks)**

[D] Digital Signature

Q. 16 What is a digital signature? How does it work? **(8 Marks)**

Q. 17 What is a digital signature? What are its applications? **(4 Marks)**

Q. 18 What is a digital signature? List its properties. **(4 Marks)**

□□□