Digital Signature and Authentication Protocols

Unit-4 [LH 5]

Outlines

- Authentication Basic
- Password (Attacking a password system, countering password, Password Aging)
- Challenge Response, Biometrics, Location, Multiple Methods,
- Mutual (Symmetric, Public Key)
- One way (Symmetric, Public Key)
- Digital Signature
- Direct Digital Signature,
- Arbitrated Digital Signature
- Digital Certificate
- x.509 Certificate
- Authentication Protocols
- Authenticate Services,
- Kerberos V4
- Digital Signature Standards (DSS)
- DSS Approach Vs RSA Approach

Authentication Basic

- **Authentication** is the process of verifying the identity of user or information. User authentication is the process of verifying the identity of user when that user logs into a computer system.
- *Authentication is the binding of an identity to a subject*.
- This process consists of sending the credentials from the remote access client to the remote access server in an either plaintext or encrypted form by using an authentication protocol.

Authentication Basic

- The external entity must provide information to enable to the subject to confirm its identity. This information comes from one (or more) of the following:
  - What the entity knows (such as passwords or secret information)
  - What the entity has (such as a badge or card)
  - What the entity is (such as fingerprints or retinal characteristics)
  - Where the entity is (such as in front of a particular terminal)
- The authentication process consists of obtaining the authentication information from an entity, analyzing the data, and determining if it is associated with that entity. This means the the computer must store some information about the entity.

Authentication Basic

- We represent these requirements in an authentication system consisting of five components:

1. The set A of *authentication information* is the set of specific information with which entities prove their identities.
2. The set C of *complementary information* is the set of information that the system stores and uses to validate the authentication information.
3. The set F of *complementation function* that generate the complementary information from the authentication information . That is for $f \varepsilon F$, $f:A \rightarrow C$
4. The set L of *authentication functions* that verity identity. That is. For $l \varepsilon L$, $l: A \; X \; C \rightarrow \{\textbf{true}, \textbf{false}\}$
5. The set S of *selection functions* that enable an entity to create or alter the authentication and complementary information.

Passwords

- A password is information associated with an entity that confirms the entity's identity.
- Passwords are an example of an authentication mechanism based on what people know: the user supplies a password, and the computer validates it.
- If the password is the one associated with the user, that user's identity is authenticated. If not, the password is rejected and the authentication fails.

Passwords

- The goal of ***authentication system*** is to ensure that entities are correctly identified.
- The entity can guess another password, then the guesser can impersonate the other.
- The authentication model provides a systematic way to analyze this problem.
- There are two approaches for protecting the passwords:
  i. Hide enough information so that one of *a,c*, or *f* cannot be found.
  ii. Prevent access to the authentication functions *L.*

Attacking a Password System

- The simplest attack against a password-based system is to guess passwords.
- A ***dictionary attack*** is the guessing a password by repeated trail and error.
- A hacker uses a program or script to try to login by cycling through combinations of common words.
- Dictionary attacks work on the assumption that most passwords consists of whole words, dates, or numbers taken from a dictionary.

Attacking a Password System

- ***Hybrid password guessing*** attacks assume that network administrators push users to make their passwords at least slightly different from a word that appears in a dictionary.
- Hybrid guessing rules vary from tool to tool, but most mix uppercase and lowercase characters, add numbers at the end of the password, spell the password backward or slightly misspell it, and include characters such as @!# in the mix.

Attacking a Password System

- ***Keystroke logging***, often referred to as key logging or keyboard capturing, is the action of recording (logging) the keys struck on a keyboard secretly so that the person using the keyboard is unaware that their actions are being monitored.
- Data can be retrieved by the person operating the logging program.
- Attackers often find it much easier to reset password that to guess them.
- Many password cracking programs are actually ***password resetters***.

Countering password guessing

- Password guessing requires either the set of complementation functions and complementary information or access to the authentication functions.
- In both approaches, the goal of the defenders is to maximize the time needed to guess the password.
- Some common password guessing are:

1. Random selection of passwords
2. Pronounceable passwords
   - eg. The password “helgoret” and “juttelon” are pronounceable passwords; “przbqxdf” and “zxrptflfn” are not.

Countering password guessing

3. User Selection of passwords

- Password based on account and user names.
- Dictionary words
- Reversed dictionary words
- Patterns from keyboard
- Passwords shorter than six characters
- Passwords containing only digits.
- Passwords containing only uppercase or lowercase letters, or letters and numbers, or letters and punctuation
- Passwords used in past
- Passwords that look like license plate numbers
- Acryonyms

Countering password guessing

4. Reusable passwords and Dictionary Attacks
   - Password reuse is a problem where people try to remember multiple passwords for everything they interact with on a regular basis, but instead use the same password on multiple systems, tiers of applications, or even social sites.
5. Guessing through Authentication Functions.

Password Aging

- Password aging is the requirement that a password be changed after some period of time or after some event has occurred.
- Guessing of password requires that access to the complement, the complementation functions, and the authentication function to be obtained.
- If none of these have changed by the time password is guessed, then the attackers can use the password to access the system.

Password Aging

- Assume that the expected time to guess a password is 180 days.
- Then changing the password more frequently than every 180 days will, in theory, reduce the probability that an attacker can guess a password that is still being used.
- In practice, aging by itself ensures little, because the estimated time to guess a password is an average; it balances those passwords that can be easily guessed against those that cannot.
- If users can choose passwords that are easy to guess, the estimation of the expected time must look for a minimum, not an average.

Password Aging

- There are problems involved in implementing password aging.
  - The *first* is forcing users to change to a different password.
  - The *second* is is providing notice to the need to change and a user friendly method of changing passwords.
- Password aging is useless if a user can simply change the current password to the same thing. One technique is prevent this to record the *n* previous passwords.
  - When a user changes a password, the proposed password is compared with these *n* previous ones. If there is a match, the proposed password is rejected.
  - Problem with this technique, user can change n times, and then change back to the original passwords. This defeat the goal of password aging.
- An alternative approach is based on time. In this implementation, the user must change the password to one other than the current password.

Challenge Response

- Passwords have the fundamental problem that they are usable.
- If an attacker sees a password, she later replay the password.
- The system cannot distinguish between the attacker and the legitimate user, and allows access.
- An alternative is to authenticate in such a way that the transmitted password changes each time.
- Then, if an attacker replays a previously used password, the system will reject it.

Challenge Response

- Let user ***U*** desire to authenticate himself to system ***S***.
- Let ***U*** and ***S*** have an agreed on secret function *f*.
- A challenge response authentication system is one in which ***S*** sends a random message m (the challenge) to ***U***, and ***U*** replies with the transformation $r = f(m)$ (the response). ***S*** validate *r* by computing it separately.

Challenge Response

Pass algorithms

- Let there be a challenge-response authentication system in which the function f is the secret. Then f is called a *pass algorithm*.
- Under this definition, no cryptographic keys or other secret information may be input to f.
- The algorithm computing f is itself the secret.

Challenge Response

One Time Passwords

- A one-time password is a password that is invalidated as soon as it is used.
- The ultimate form of password aging occurs when a password is valid for exactly one use.
- A mechanism that uses one-time passwords is also a challenge-response mechanism.
- The challenge is the number of the authentication attempt, the response is the one-time password.
- The number of any one-time password scheme are the generation of random passwords and the synchronization of the user and the system.

Hardware-Supported Challenge-Response Procedures

- Hardware support comes in two forms: a program for a general-purpose computer and special-purpose hardware support. Both perform the same functions.
- The first type of hardware device informally called a *token*, provides mechanisms for hashing or enciphering information. With this type of device, the system sends a challenge. The user enters it into the device. The device returns an appropriate response.
- Some devices require the user to enter a personal identification number of password, which is used as cryptographic key or is combined with the challenge to produce the response.

Hardware-Supported Challenge-Response Procedures

- The second type of device is temporarily based. Every 60 seconds, it displays a different number. The numbers range from 0 to 10n-1, inclusive.
- A similar device is attached to the computer. It knows what number the device for each registered user should display.
- To authenticate, the user provides his login name. The system requests a password. The user then enters the number shown on the hardware device, followed by a fixed (reusable) password.
- The system validates that the number is the one expected for the user at that time that reusable portion of the password is correct.

Biometrics

- Biometrics is the measurement and statistical analysis of people’s unique physical and behavioral characteristics
- Recognizing people by their voices or apperance, and impersonating people by assuming their apperance, was widely known in classical times.
- Efforts to find physical characteristics that uniquely identify people include the fingerprints, and DNA sampling.
- Using such a feature to identify people for a computer would ideally eliminate errors in authentication.
- Biometrics is the automated measurement of biological or behavioral features that identify a person.

Biometrics

- When a user is given an account, the system administration takes a set of measurements that identify that user to an acceptable degree of error.
- Whenever the user accesses the system, the biometrics authentication mechanism verifies the identity.
- It is considerably easy to identify the user by biometrics since no searching is required.
- A comparison to the known data for the claimed user’s identity will either verify or reject the claim.
- Common characteristics are fingerprints, voice characteristics, eyes, facial features, and keystroke dynamics.

Biometrics

- Fingerprints
- Voices
- Eyes
- Faces
- Keystrokes
- Combinations
- Caution

See: introduction of Computer Security by Matt Bishop, chapter 5 for detail expalanation.

Location

- Denning and MacDoran suggested an innovative approach to authentication.
- The reason that if a user claims to be Anna, who is at that moment working in a bank of California but is also logging in from Russia at the same time, the user is impersonating Anna.
- Their scheme is based on the Global Positioning System (GPS), which can pinpoint a location to within a few meters.
- The physical location (to within a few meters) and time (to within a few milliseconds) is unique, and hence form a *location signature*. This signature is transmitted to authenticate the user
- The host also has a ***location signature sensor*** (LSS) and obtains a similar signature for the user. If the signature disagree, the authentication fails.

Multiple Methods

- Authentication methods can be combined or multiple methods can be used.
- Authentication by location generally uses special-purpose hardware.
- Although the key feature of this technique is physical location, without the LSS it will not work.
- Techniques using multiple methods assign one or more authentication methods to each entity.
- The entity must authenticate using the specific method, or method, chosen.

Multiple Methods

- The specific authentication methods vary from system to system, but in all cases the multiple layers of authentication require an attacker to know more, or posses more, than is required to spoof a single layer.
- Some versions of the UNIX operating system provide a mechanism called *pluggable authentication modules* (PAM).
-

Mutual (Symmetric, Public Key)

- Mutual authentication, also known as ***two-way authentication***, is a security process in which entities authenticate each other before actual communication occurs.
- In a network environment, this requires that both the client and the server must provide ***digital certificates*** to prove their identities.
- In a mutual authentication process, a connection can occur only if the client and the server exchange, verify, and trust each other's certificates.
- The certificate exchange occurs by means of the ***Transport Layer Security (TLS) protocol***.
- The core of this process is to make sure that clients communicate with ***legitimate*** servers, and servers cooperate only with clients who attempt access for legitimate purposes.

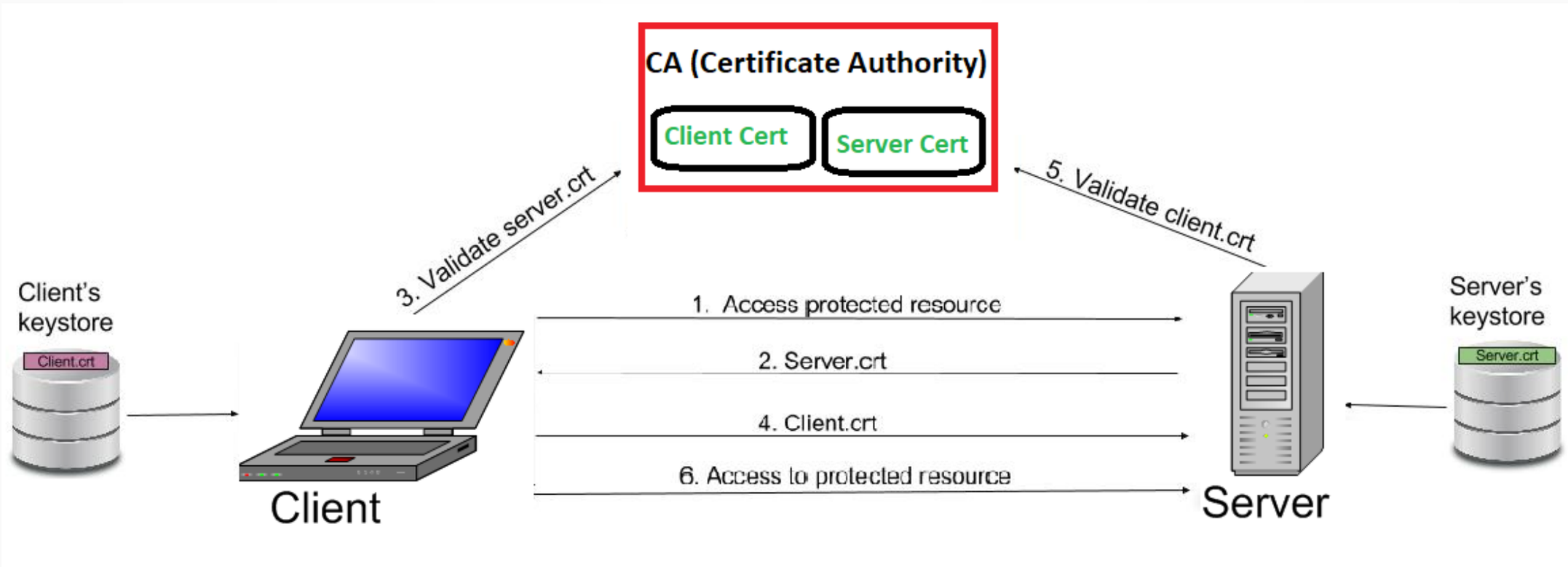
Mutual Authentication

- Mutual authentication is a desired characteristic in ***verification schemes*** that transmit sensitive data, in order to ensure data security.
- Mutual authentication can be accomplished with two types of credentials: *usernames* and *passwords*, and *public key certificates*.
- Mutual authentication is often employed in the *Internet of Things* (IoT).
- Mutual authentication protect communication against adversarial attacks such as man in the middle attack, replay attacks, spoofing attacks, impersonation attacks.

Mutual Authentication

- Establishing the authentication using certificate based 2-way SSL involves:
  - A client request access to a protected resource.
  - The sever presents its certificate to the client.
  - The client verifies the server’s certificate.
  - If successful, the client sends its certificate to the server.
  - The server verifies the clients credentials.
  - If successful, the server grants access to the protected resource requested by the client.

Mutual Authentication



Mutual Authentication

Below is the high level description of the steps involved in establishment of connection and transfer of data between a client and server in case of two-way SSL:

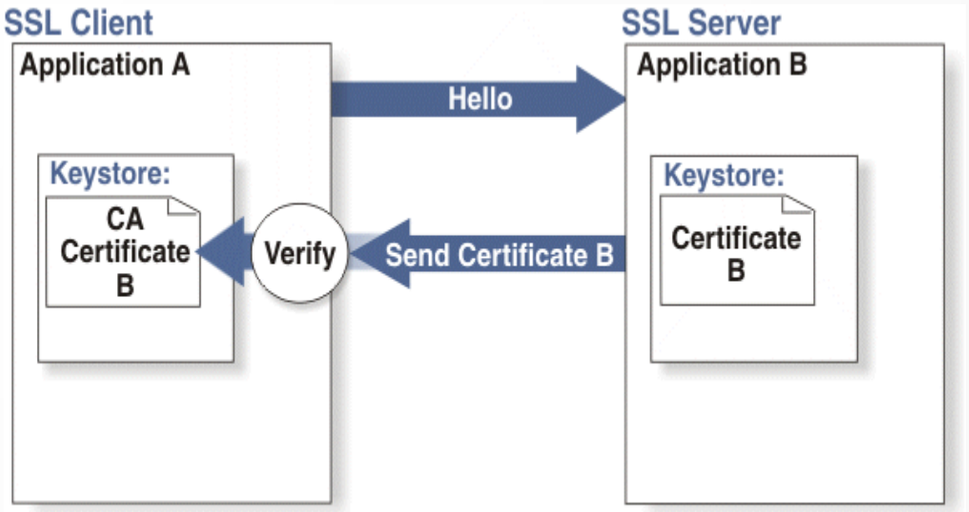
1. Client requests a protected resource over HTTPS protocol and the SSL/TSL handshake process begin.
2. Server returns its public certificate to the client along with server hello.
3. Client validates/verifies the received certificate. Client verifies the certificate through certification authority (CA) for CA signed certificates.
4. If Server certificate was validated successfully, client will provide its public certificate to the server.
5. Server validates/verifies the received certificate. Server verifies the certificate through certification authority (CA) for CA signed certificates.
6. After completion of handshake process, client and server communicate and transfer data with each other encrypted with the secret keys shared between the two during handshake.

One-way (Symmetric, Public key)

- One-way authentication is a process or technology *in which only client authenticates server's identity before actual communication occurs*.
- This is to ensure that clients are communicating exclusively with legitimate servers.
- For implementing *one-way SSL*, server shares its public certificate with the clients.

One-way (Symmetric, Public key)

- Establishing the authentication using certificate-based-1-way SSL involves:
  - A client requests access to a protected resource.
  - The server presents its certificate to the client.
  - The client verifies the server's certificate.
  - If successful, the client authenticates the server as legitimate.

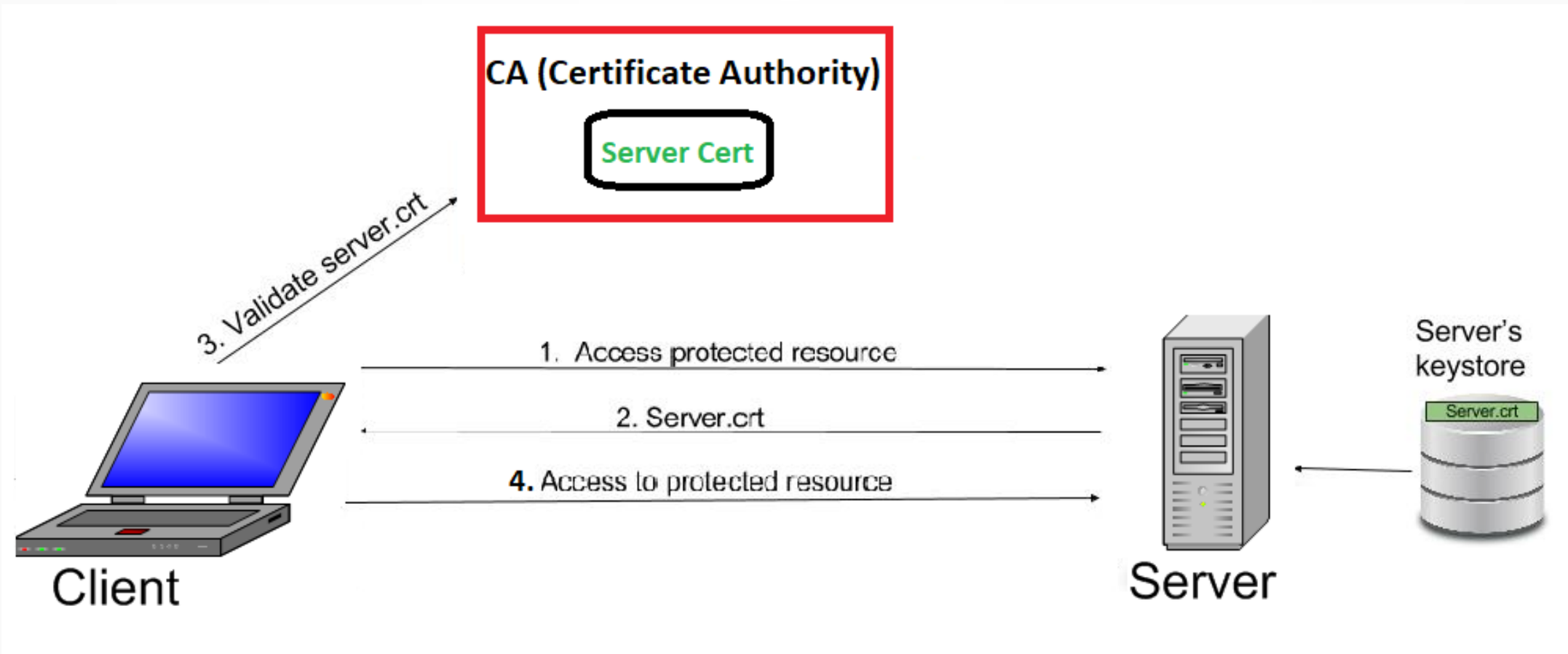


How one-way SSL Works?

Below is the high level description of the steps involved in establishment of connection and transfer of data between a client and server in case of one-way SSL:

1. Client requests for some protected data from the server on HTTPS protocol. This initiates SSL/TLS handshake process.
2. Server returns its public certificate to the client along with server hello message.
3. Client validates/verifies the received certificate. Client verifies the certificate through certification authority (CA) for CA signed certificates.
4. SSL/TLS client sends the random byte string that enables both the client and the server to compute the secret key to be used for encrypting subsequent message data. The random byte string itself is encrypted with the server's public key.
5. After agreeing on this secret key, client and server communicate further for actual data transfer by encrypting/decrypting data using this key.

One-way (Symmetric, Public key)



Digital Signatures

- A digital signature is *an authentication mechanism that enables the creator of a message to attach a code that acts as a signature*.
- A digital code (generate and authenticated by public key encryption) which is *attached to an electronically transmitted document to verify its contents and the sender's identity.*
- The signature guarantees the source and the integrity of the message.
- It is the one of the most important work of ***public key cryptography***. Typically the signature is formed by taking the hash of the message and encrypting the message with the creator's private key.

Digital Signature

- The digital signature must have the following properties:
  i. It must verify the author and the date and time of the signature.
  ii. It must authenticate the contents at the time of the signature.
  iii. It must be verifiable by third parties, to resolve disputes.
- Thus, the digital signatures function includes the authentication function.

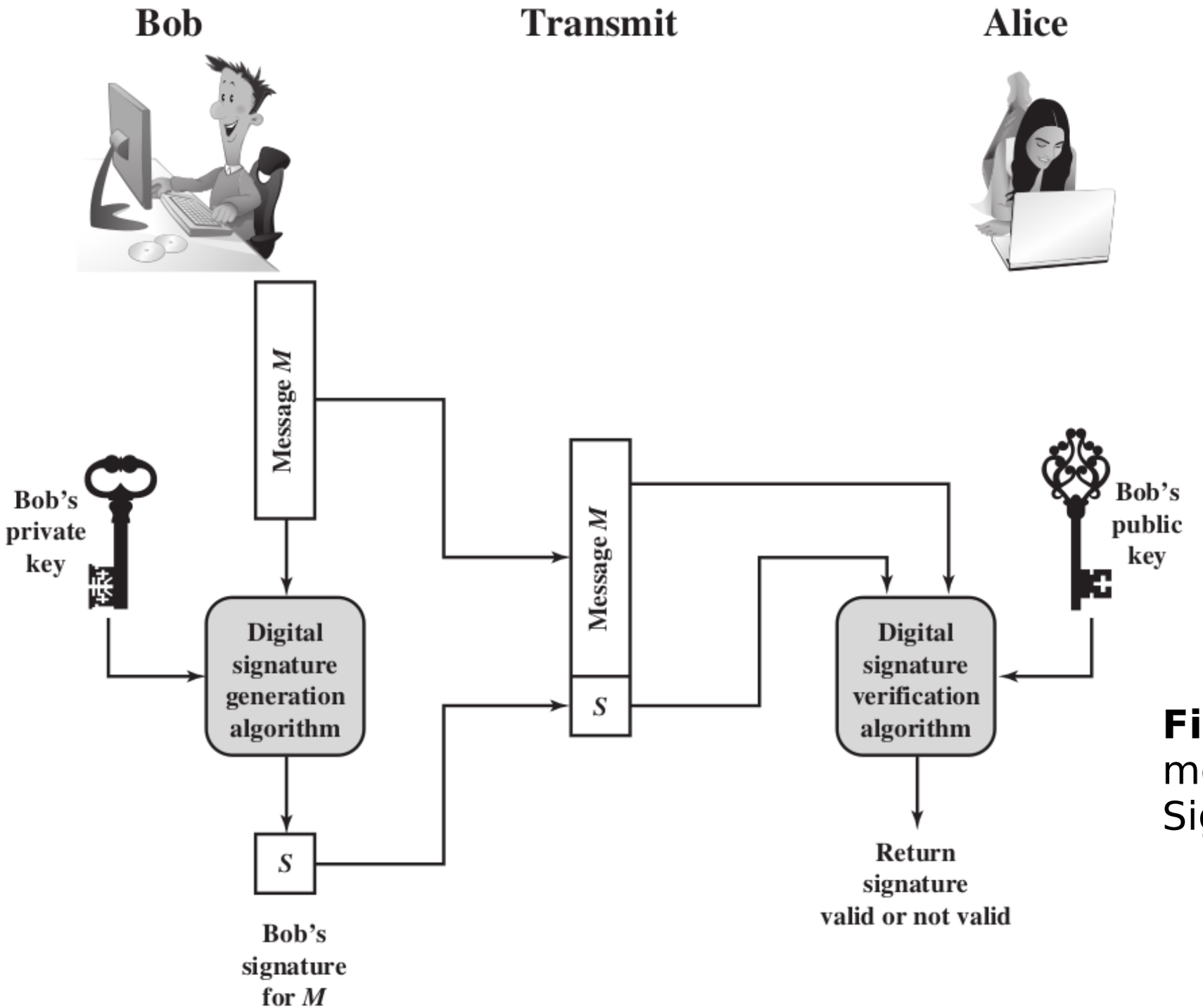


Figure: Generic model of Digital Signature Process

- To create a digital signature, signing software (such as an email program) creates a one-way hash of the electronic data to be signed.
- The private key is then used to encrypt the hash. The encrypted hash is the digital signature.
- The reason for encrypting the hash instead of the entire message or document is that a hash function can convert an arbitrary input into a fixed length value, which is usually much shorter.
- This saves time since hashing is much faster.
-

Digital Signature

- The value of the hash is unique to the hashed data.
- Any change in the data, even changing or deleting a single character, results in a different value.
- This attributes enables others to validate the integrity of the data by using the signer’s public key to decrypt the hash.
- If the decrypted hash matches a second computed hash of the same data, it proves that the data hasn’t changed since it was signed.
- Of the two hashes don’t match, the data has either been tempered with in some way (integrity) or the signature was created with a private key that doesn’t correspond to the public key presented by the signer (authentication)

Direct Digital Signature

- The term direct digital signature refers to a digital signature scheme that involves only the communicating parties (source, destination).
- It is assumed that the destination knows the public key of the source.
- Confidentiality can be provided by encrypting the entire message plus signature with a shared secret key (symmetric encryption)
- Note that it is important to perform the signature functions first and then an outer confidentiality function.

Direct Digital Signature

- In case of dispute, some third party must view the message and its signature.
- If the signature is calculated on an encrypted message, then the third party also needs access to the decryption key to read the original message.
- However, if the signature is the inner operation, then the recipient can store the plain text message and its signature for later use in dispute resolution.

Arbitrated Digital Signature

- Implementing an arbitrated digital signature invites a third party into the process called a “trusted arbiter.”
- The role of the trusted arbiter is usually two folds:
- first this independent third party verifies the integrity of the signed message or date.
- Second the trusted arbiter dates or time-stamps the documents, verifying receipt and the passing on of the signed document to its indented final destiantion.

Arbitrated Digital Signature

- This approach requires suitable level of trust in arbiter to ensure that the arbiter is not biased and unauthorized modification won’t be done.
- This an be implemented with either private or public-key algorithms.

Differences:

Direct Digital Signature	Arbitrated Digital Signature
1. It only require the communicating parties.	1. It requires arbiter along with communicating parties to send or receive messages.
2. In this the digital signature encrypts the whole plain text with the sending party's private key.	2. The encrypted message is send by X to arbiter Z with Y's id, timestamp and some random number PQ.
3. The message is directly transmitted between both parties without any help of a intermediate	3. Arbiter is needed to transmit the message.
4. Timestamp is not maintained by both side.	4. Timestamp is maintained by all three members by default.
5. In case the confidentiality is needed the message will be encrypt with receiver's public key or a shared key.	5. The arbiter provides confidentiality of the message.
6. Vulnerable to any kind of replay attack.	6. The timestamp is used to protect the message from any kind of replay attack.
7. It is implemented using public key.	7. It is implemented using private key.

Digital Certificates

- Also called ***public-key certificate or identity certificate.***
- IT is an electronic file that typically contains identification information about the holder, including the person's public key (used for encrypting and decrypting messages), along with the authority's digital signature, so that the recipient can verify with the authority that the certificate is authentic.
- It is an electronic "passport" that allows a person, computer or organization to exchange information securely over the Internet using the *public key infrastructure*. (PKI).
- A digital certificate is issued by a ***certification authority*** (CA).

Digital Certificates

- **Websites** usually also have digital certificates, to enable a person intending to buy its products to confirm that it is an authenticated site. Such certificates serve as the security basis for HTTPS (Hypertext Transfer Protocol Secure).
- The most common examples of digital signature is X.509.

X.509 Certificate

- X.509 is an *International Telecommunication Union (ITU) standard* defining the format of public key certificates.
- An X.509 certificate *binds an identity to a public key using a digital signature*.
- A certificate *contains an identity* (a hostname, or an organization, or an individual) and a *public key* (RSA, DSA, ECDSA, ed25519, etc.).
- It is either signed by a ***certificate authority*** or is self-signed.
- X.509 certificates are used in many Internet protocols, including **TLS/SSL**, which is the basis for **HTTPS**, the secure protocol for browsing the web

Authentication Protocols

- An authentication protocols is a type of cryptographic protocol specifically designed for transfer of authentication data between two entities.
- It allows the receiving entity to authenticate the connecting entity (e.g. Client connecting to a Server) as well as authenticate itself to the connecting entity (Server to a client) by declaring the type of information needed for authentication.
- It is the most important layer of protection needed for secure communication within computer networks
- The tasks of the authentication protocol is to specify the exact series of steps needed for execution of the authentication.

Authentication Protocolss

- It has to comply with the main protocol principles:
  1. A Protocol has to involve two or more parties and everyone involved in the protocol must know the protocol in advance.
  2. All the included parties have to follow the protocol.
  3. A protocol has to be unambiguous - each step must be defined precisely.
  4. A protocol must be complete - must include a specified action for every possible situation.

Authentication Protocol

- An illustration of password-based authentication using simple authentication protocol:
- Alice (an entity wishing to be verified) and Bob (an entity verifying Alice's identity) are both aware of the protocol they agreed on using. Bob has Alice's password stored in a database for comparison.
  i. Alice sends Bob her password in a packet complying with the protocol rules.
  ii. Bob checks the received password against the one stored in his database. Then he sends a packet saying "Authentication successful" or "Authentication failed" based on the result.

Types of Authentication Protocols

- PAP: Password Authentication Protocol
- CHAP: Challenge Handshake Authentication Protocol
- EAP: Extensible Authentication Protocol

PAP: Password Authentication Protocol

- Password Authentication Protocol (PAP) is a password-based authentication protocol used by Point to Point Protocol (PPP) to validate users.
- Almost all network operating system remote servers support PAP.

Working Cycle:

- PAP authentication is only done at the time of the initial link establishment, and verifies the identity of the client using a two-way handshake.
  1. Client sends username and password. This is sent repeatedly until a response is received from the server.
  2. Server sends authentication-ack (if credentials are OK) or authentication-nak (otherwise)
- It is highly insecure because credentials are sent "ASCII" and repeatedly, making it vulnerable even to the most simple attacks like eavesdropping and man-in-the-middle based attacks.

PAP: Password Authentication Protocol

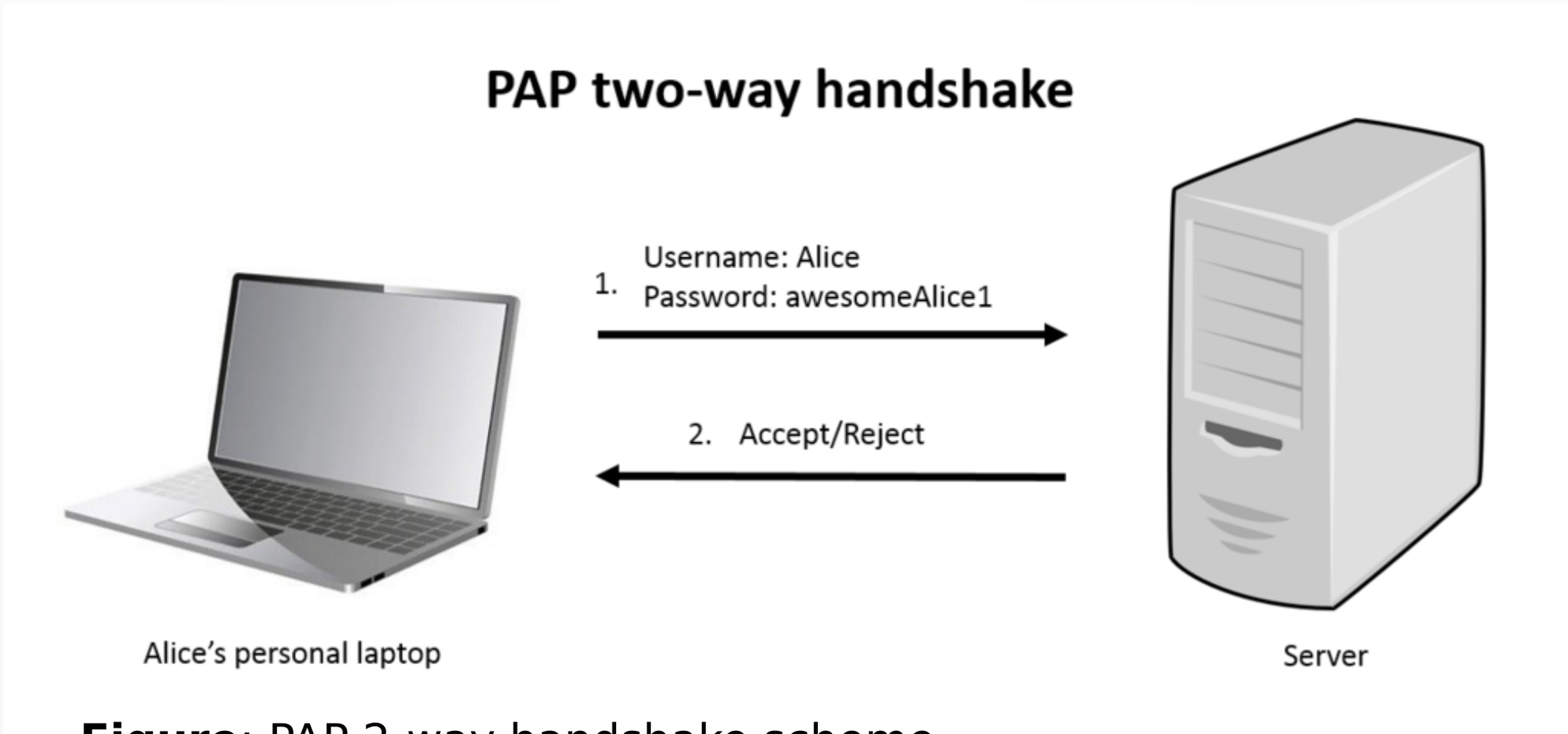


Figure: PAP 2-way handshake scheme.

CHAP: Challenge-Handshake Authentication Protocol

- The authentication process in this protocol is always initialized by the server/host and can be performed anytime during the session, even repeatedly.
- CHAP periodically verifies the identity of the client by using a three-way handshake.
  - Server sends a random string (usually 128B long).
  - The client uses password and the string received as parameters for MD5 hash function and then sends the result together with username in plain text.
  - Server uses the username to apply the same function and compares the calculated and received hash. An authentication is successful or unsuccessful.
- CHAP provides better security as compared to Password Authentication Protocol (PAP)

EAP: Extensible Authentication Protocol

- EAP was originally developed for PPP(Point-to-Point Protocol) but today is widely used in IEEE 802.3, IEEE 802.11(WiFi) or IEEE 802.16 as a part of IEEE 802.1x authentication framework.
- The advantage of EAP is that it is only a general authentication framework for client-server authentication - the specific way of authentication is defined in its many versions called EAP-methods.
- More than 40 EAP-methods exist

Authentication Service: Kerberos v4

- Kerberos allows two users (or client and server) to authenticate each other over an insecure network, such as internet.
- Named after the Greek mythological character Kerberos (or Cerberus), known in Greek mythology as being the monstrous three-headed guard dog of Hades.
- The three heads of the Kerberos protocol represent a client, a server and a Key Distribution Center (KDC), which acts as a trusted third party authentication service.
- Designed originally for Project Athena at M.I.T.
- Windows 2000/XP/Server 20003/Vista, Apple's MAC and Linux use Kerberos as their default authentication mechanism.
- Protects against eavesdropping and replay attacks.
- Uses a trusted third party **(key distribution center**) and symmetric key cryptography.
- Each user and service on the network is principal.
- First 3 versions are no longer in use.

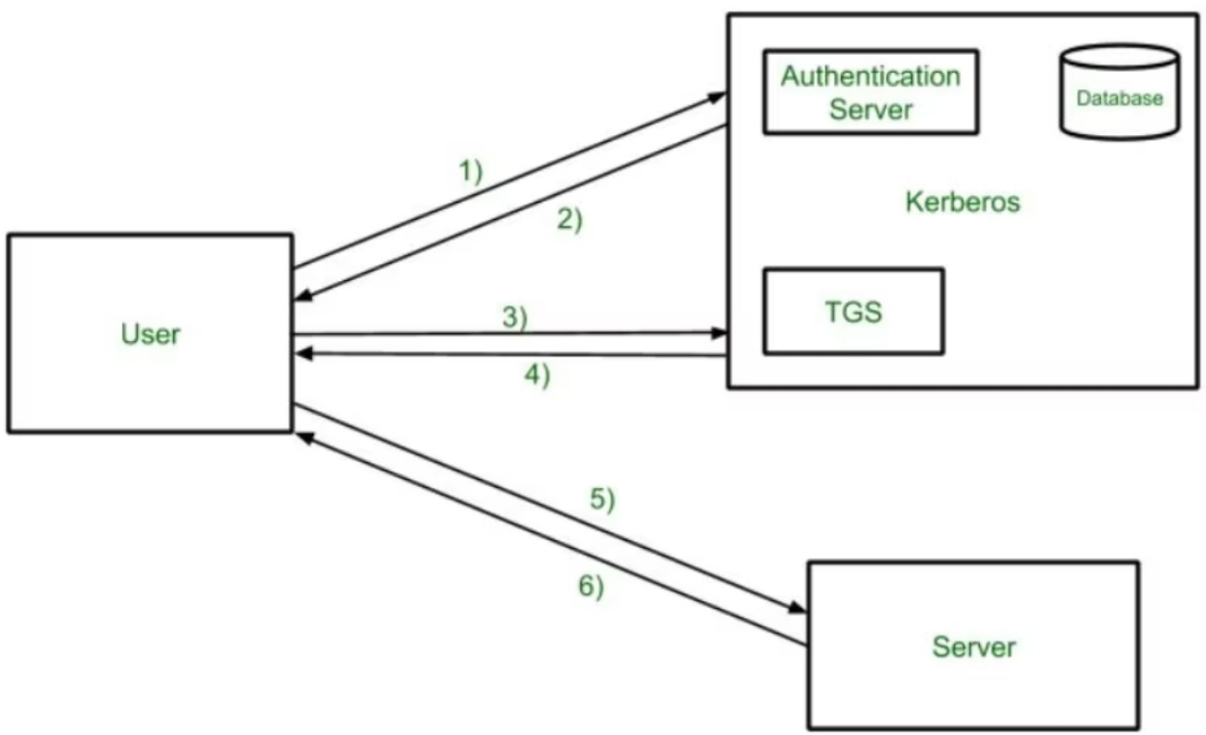
Authentication Service: Kerberos v4

- KDC provides two services:
  - An *authentication service* and
  - A *ticket granting service*.
- KDC "**tickets**" provide mutual authentication allowing nodes to prove their identity to one another in a secure manner.
- Kerberos authentication used **DES** cryptography to prevent packets traveling across the network from being read or changed and to protect messaged from evesdropping and replay attacks.

Kerberos Principals

- The principal entities involved in the typical Kerberos workflow:
- **Client**: The client acts on behalf of the user and initiates communication for a service request
- **Server**: The server hosts the service the user wants to access
- **Authentication Server (AS)**: The AS performs the desired client authentication. If the authentication happens successfully, the AS issues the client a ticket called TGT (Ticket Granting Ticket). This ticket assures the other servers that the client is authenticated
- **Key Distribution Center (KDC)**: In a Kerberos environment, the authentication server logically separated into three parts: A database (Bb), the Authentication Server (AS), and the Ticket Granting Server (TGS). These three parts, in turn, exist in a single server called the Key Distribution Center
- **Ticket Granting Server (TGS)**: The TGS is an application server that issues service tickets as a service

Kerberos Protocol flow



Kerberos Protocol flow

- **Step 1**:
  - Initial client authentication request. The user asks for a Ticket Granting Ticket (TGT) from the authentication server (AS). This request includes the client ID..
- **Step 2**:
  - KDC verifies the client's credentials. The AS checks the database for the client and TGS's availability. If the AS finds both values, it generates a client/user secret key, employing the user's password hash.
  - The AS then computes the TGS secret key and creates a session key (SK1) encrypted by the client/user secret key. The AS then generates a TGT containing the client ID, client network address, timestamp, lifetime, and SK1. The TGS secret key then encrypts the ticket.
- **Step 3**:
  - The client decrypts the message. The client uses the client/user secret key to decrypt the message and extract the SK1 and TGT, generating the authenticator that validates the client's TGS.

Kerberos protocol flow

- **Step 4**:
  - The client uses TGT to request access. The client requests a ticket from the server for requesting services from the server.
- **Step 5**:
  - The user sends the Ticket and Autheticator then generate access to the service. After this User can access the services.

Kerberos protection

Kerberos protects against eavesdropping:

- If someone else sends TGT, they get back a ticket and can’t decrypt the service key unless they know the client’s secret key.

- Kerberos protects against replay attacks.
  - If someone sends TGT or ticket later, it is rejected.
- All clients, servers should have time synchronized within a specified limit.

Digital Signature Standard (DSS)

- The Digital Signature Standard (DSS) is a **digital signature algorithm** developed by the U.S. National Security Agency as a means of authentication for electronic documents.
- The algorithm makes use of two large numbers which are calculated based on a unique algorithm.
- The digital signature can be generated only by the authorized person using their private keys and the users or public can verify the signature with the help of the public keys provided to them.

DSA Approach vs RSA Approach

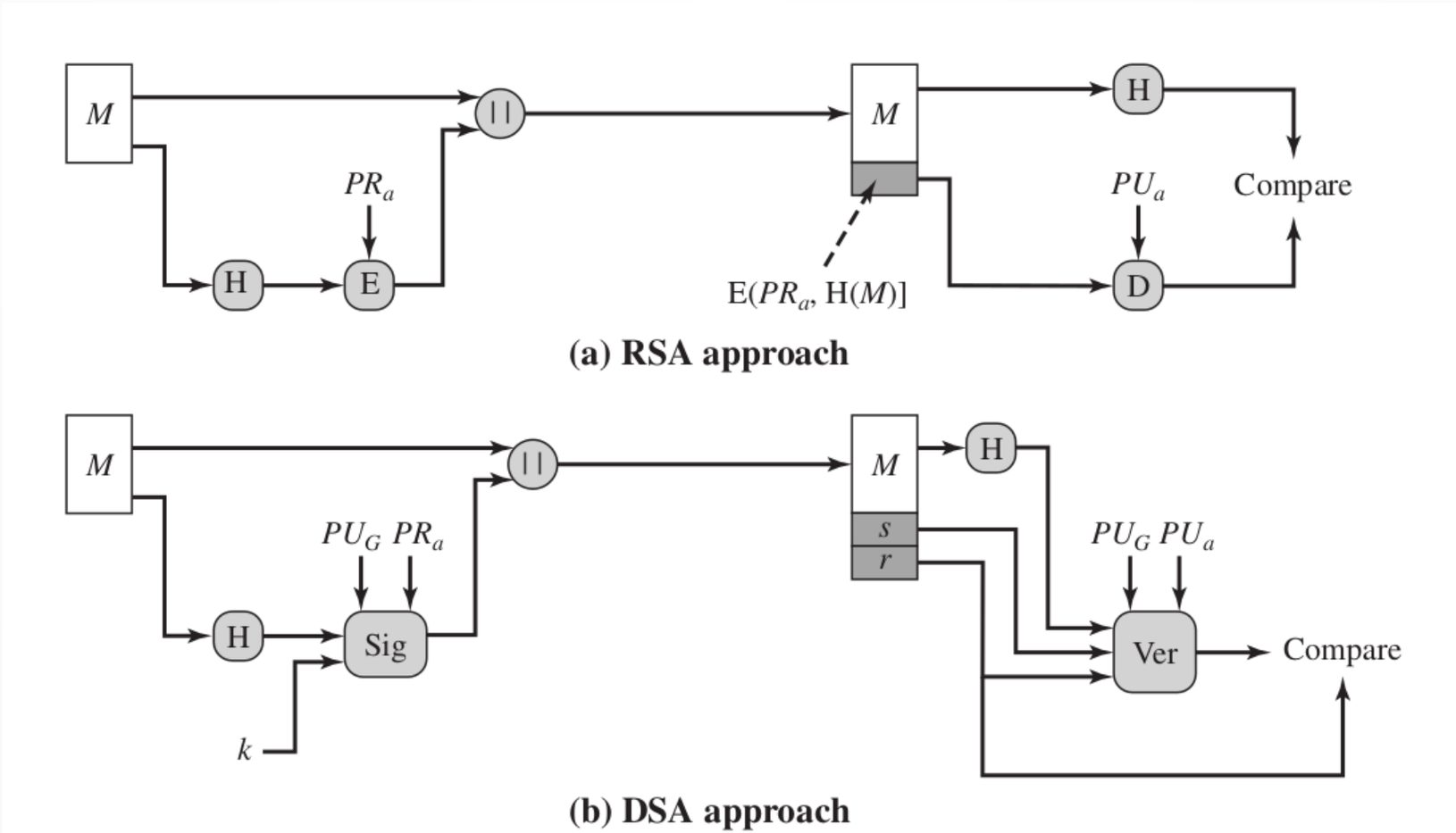
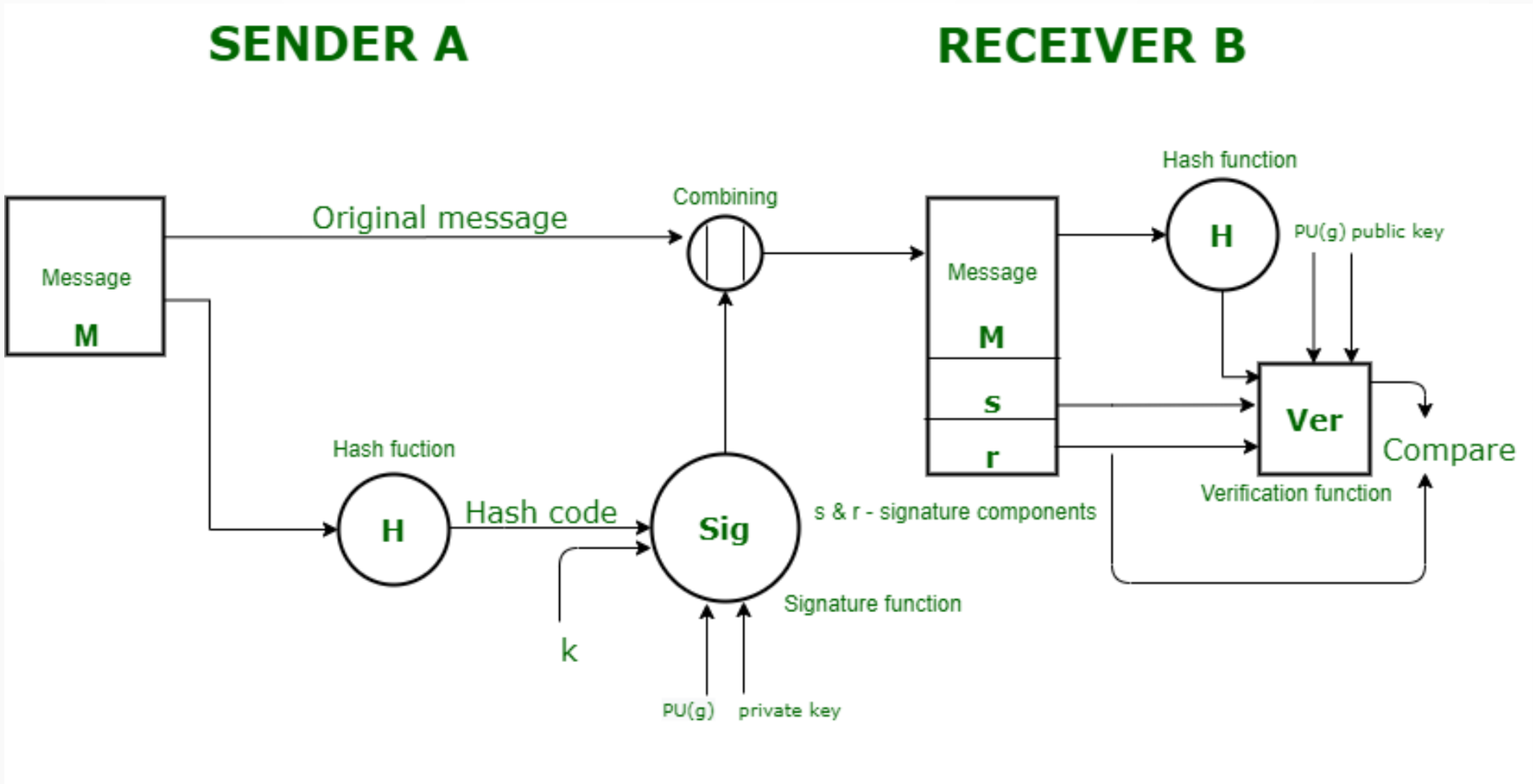


Figure: Two approaches of Digital Signatures

DSA Approach



DSA Approach

Sender Side :

- In DSS Approach, a hash code is generated out of the message and following inputs are given to the signature function
  1. The hash code.
  2. The random number ‘k’ generated for that particular signature.
  3. The private key of the sender i.e., PR(a).
  4. A global public key(which is a set of parameters for the communicating principles) i.e., PU(g).
- These input to the function will provide us with the output signature containing two components – ‘s’ and ‘r’. Therefore, the original message concatenated with the signature is sent to the receiver.

DSA Appraoch

Receiver Side :

- At the receiver end, verification of the sender is done. The hash code of the sent message is generated. There is a verification function which takes the following inputs
  1. The hash code generated by the receiver.
  2. Signature components ‘s’ and ‘r’.
  3. Public key of the sender.
  4. Global public key.
- The output of the verification function is compared with the signature component ‘r’. Both the values will match if the sent signature is valid because only the sender with the help of it private key can generate a valid signature.

DSS vs encryption vs others

- One key difference between encryption and signature operation in the digital signature operation in the Digital Signature Standard is that encryption is reversible, whereas the digital signature operation is not.
- Another fact about the digital signature standard is that it does not provide any capability with regards to key distribution or exchange of keys.
- In other words, security of the digital signature largely depends on the secrecy of the private keys of the signatory.

DSS Approach vs RSA Approach

- The DSA uses an algorithm that is designed to provide only the digital signature function. Unlike RSA, it cannot be used for encryption or key exchange.
- Nevertheless, it is a public-key technique.

RSA Approach

- In the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length.
- This hash code is then encrypted using the sender’s private key to form the signature.
- Both the message and the signature are then transmitted. The recipient takes the message and produces a hash code.
- The recipient also decrypts the signature using the sender’s public key.
- If the calculated hash code matches the decrypted signature, the signature is accepted as valid. Because only the sender knows the private key, only the sender could have produced a valid signature.

DSA Approach

- The DSA approach also makes use of a hash function. The hash code is provided as input to a signature function along with a random number k generated for this particular signature. The signature function also depends on the sender's private key (PR_a) and a set of parameters known to a group of communicating principals. We can consider this set to constitute a global public key (PU_G). The result is a signature consisting of two components, labeled s and r.
- At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function. The verification function also depends on the global public key as well as the sender's public key (PU_a), which is paired with the sender's private key. The output of the verification function is a value that is equal to the signature component r if the signature is valid. The signature function is such that only the sender, with knowledge of the private key, could have produced the valid signature.