Security

Cryptography is the science of keeping messages private. A cryptographic system uses encryption keys between two trusted communication partners. These keys encrypt and decrypt information so that the information is known only to those who have the keys.

Truststore is used for the storage of certificates from the trusted Certificate Authority (CA), which is used in the verification of the certificate provided by the server in an SSL connection.

Public key cryptography is a method of encrypting or signing data with two different keys and making one of the keys, the public key, available for anyone to use.

The other key is known as the private key. Data encrypted with the public key can only be decrypted with the private key.

Transport Layer Security (TLS) encrypts data sent over the Internet to ensure that eavesdroppers and hackers are unable to see what you transmit which is particularly useful for private and sensitive information such as passwords, credit card numbers, and personal correspondence.

CA:A certificate authority (CA) is a trusted entity that issues Secure Sockets Layer (SSL) certificates.

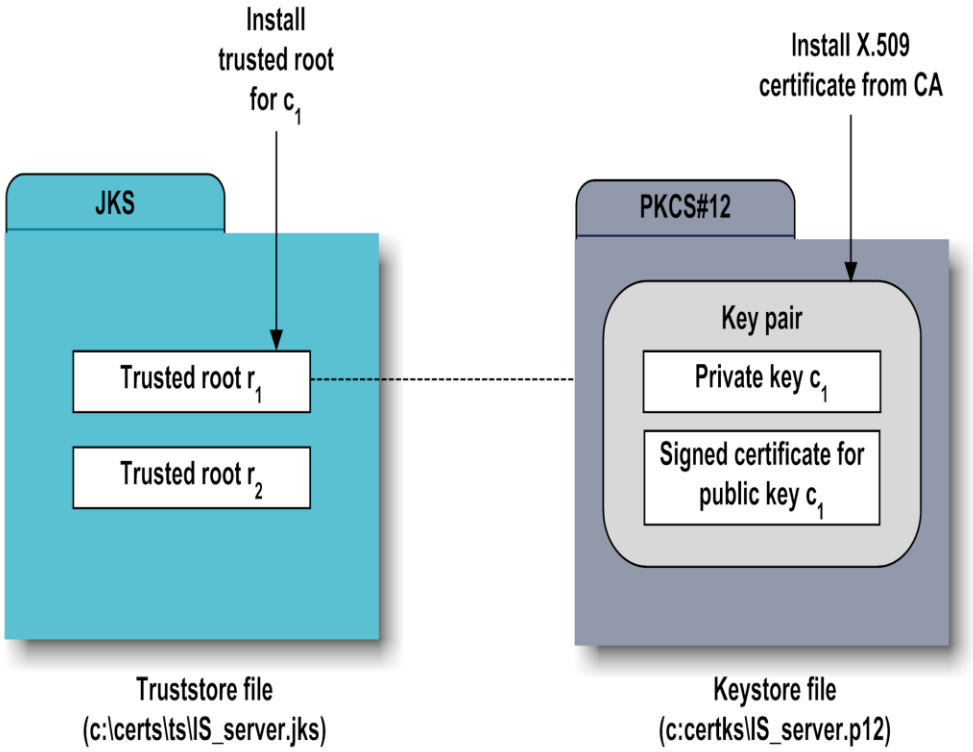
Authentication refers to a user's identity: who they are and whether their identity has been confirmed by a login process.

Authorization refers to a user's privileges or permissions: specifically, what actions they are allowed to perform within a company's systems.

Ciphertext is encrypted text transformed from plaintext using an encryption algorithm. Ciphertext can't be read until it has been converted into plaintext (decrypted) with a key. The decryption cipher is an algorithm that transforms the ciphertext back into plaintext.

Keystore is used to store the private key and own identity certificate to be identified for verification.

In an SSL handshake, the work of Truststore is to verify the credentials, whereas the work of Keystore is to provide those credentials.



TrustStore	KeyStore
TrustStore doesn't contain private and sensitive information.	Keystore contains private and sensitive information.
javax.net.ssl.trustStore is used to specify TrustStore.	javax.net.ssl.keyStore is used to specify Keystore.

TrustStore	KeyStore
TrustStore setup is required for the successful connection at the client side.	Keystore is needed when you are setting up the server-side on SSL.
TrustStore stores other's credentials.	Keystore stores your credential.
A TrustStore holds the certificates of external systems that you trust.	A KeyStore holds your application's certificates.
TrustStore password is given by following extension Djavax.net.ssl.trustStorePassword.	KeyStore password is given by following extension Djavax.net.ssl.keyStorePassword
TrustStore and TrustStore passwords are stored in clear files and is visible to all.	Keystore and key passwords are stored in plain text, in a file that is readable only by members of the appropriate group.

PKI Certificate is a digital certificate used to authenticate users, servers, or devices online. Commonly used for signing code, documents, or email...etc

X.509 Certificates

A *digital certificate*, as specified by the X.509 v3 standard, contains data establishing a principal's authentication and authorization information. A certificate contains:

- A public key, which is used in public key infrastructure (PKI) operations
- Identity information (for example, name, company, and country)
- Optional digital rights, which grant privileges to the owner of the certificate

Each certificate is digitally signed by a *trust point*. The trust point signing the certificate can be a *certificate authority* (CA) such as VeriSign, a corporation, or an individual.

How SSL works?

SSL fundamentally works with the following concepts:

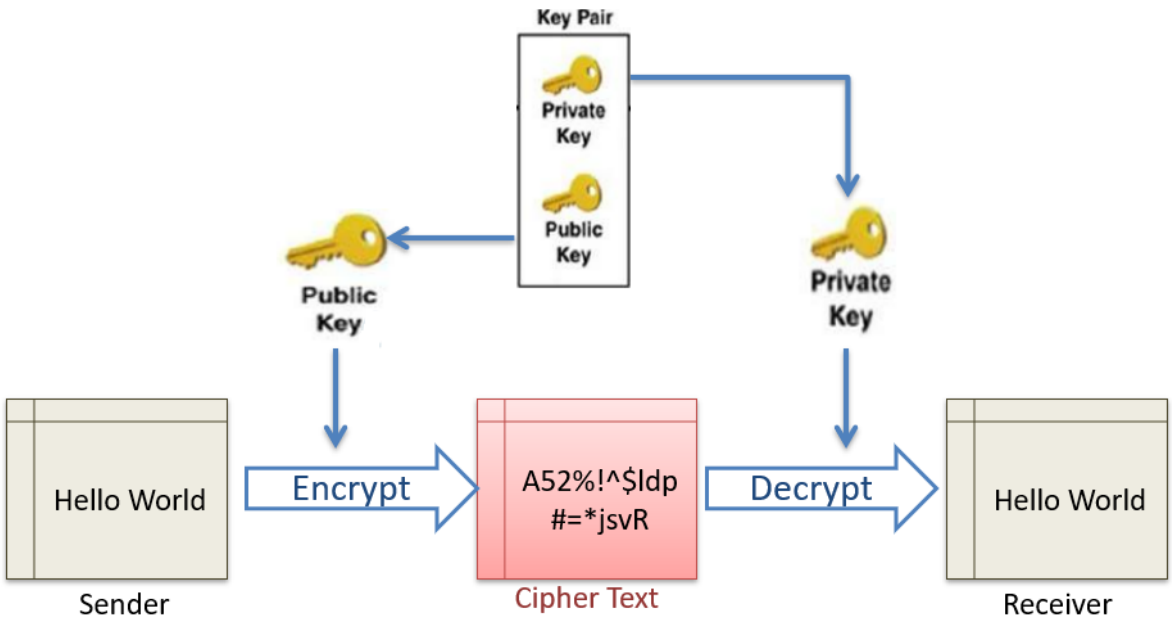
1. Asymmetric Cryptography
2. Symmetric Cryptography

Asymmetric Cryptography

Asymmetric cryptography (also known as Asymmetric Encryption or Public Key Cryptography) uses a mathematically-related key pair to encrypt and decrypt data.

In a key pair, one key is shared with anyone who is interested in a communication. This is called **Public Key**. The other key in the key pair is kept secret and is called **Private Key**.

In the asymmetric cryptography, the sender encrypt data with the receiver's public key and send it to the receiver. The receiver decrypts it using the related private key.

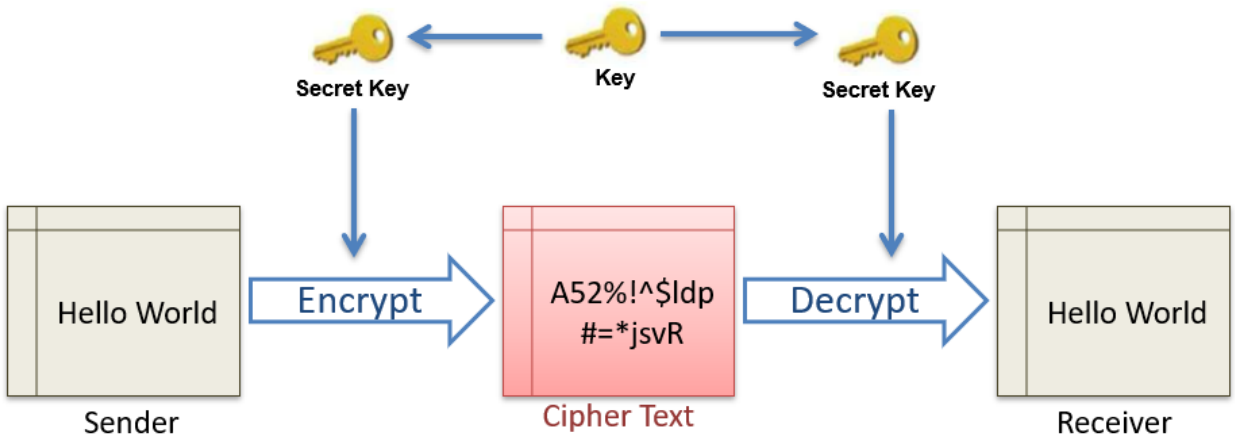


SSL uses asymmetric cryptography to initiate the communication which is known as SSL handshake.

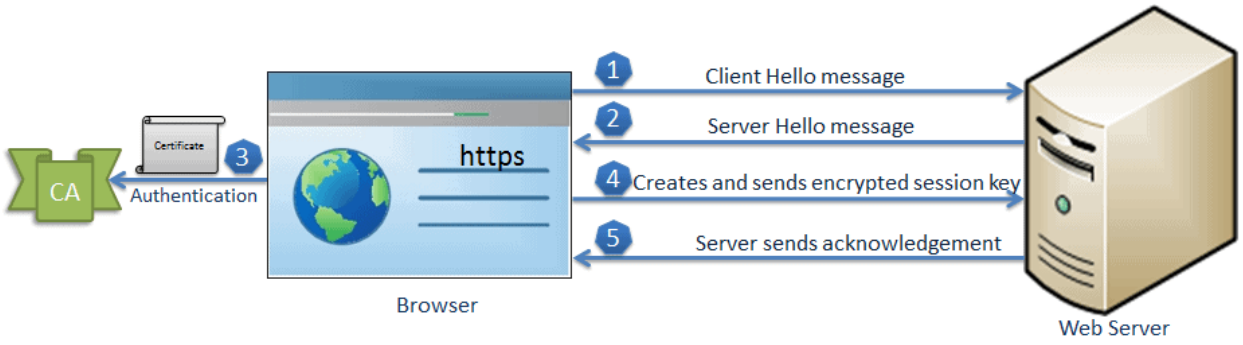
Most commonly used asymmetric key encryption algorithms include ElGamal, RSA, DSA, Elliptic curve techniques and PKCS.

Symmetric Cryptography

In the symmetric cryptography, there is only one key which encrypts and decrypts the data. Both sender and receiver should have this key, which is only known to them.



SSL uses symmetric cryptography using the session key after the initial handshake is done. The most widely used symmetric algorithms are AES-128, AES-192 and AES-256.



1. The client sends a "client hello" message. This includes the client's SSL version number, cipher settings, session-specific data and other information that the server needs to communicate with the client using SSL.

2. The server responds with a "server hello" message. This includes the server's SSL version number, cipher settings, session-specific data, an SSL certificate with a public key and other information that the client needs to communicate with the server over SSL.
3. The client verifies the server's SSL certificate from CA (Certificate Authority) and authenticates the server. If the authentication fails, then the client refuses the SSL connection and throws an exception. If the authentication succeeds, then proceed to step 4.
4. The client creates a session key, encrypts it with the server's public key and sends it to the server. If the server has requested client authentication (mostly in server to server communication), then the client sends his own certificate to the server.
5. The server decrypts the session key with its private key and sends the acknowledgement to the client encrypted with the session key.

Thus, at the end of the SSL handshake, both the client and the server have a valid session key which they will use to encrypt or decrypt actual data. The public key and the private key will not be used any more after this.

Kerberos:

Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users.

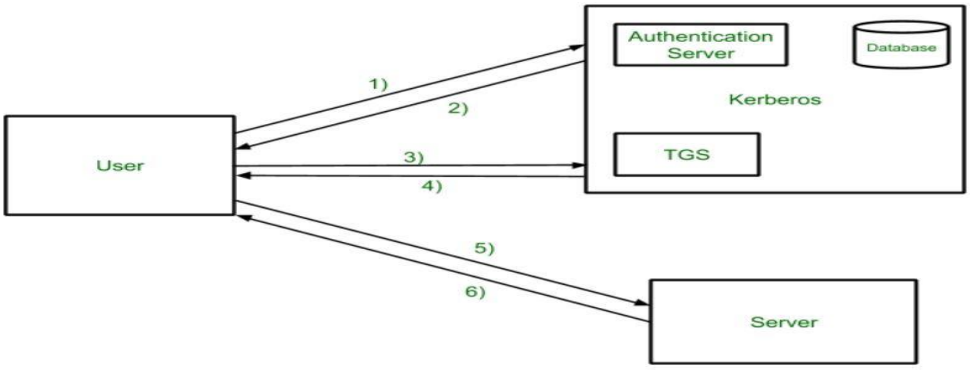
In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as the Key Distribution Center (KDC). Each user and service on the network is a principal.

The main components of Kerberos are:

- **Authentication Server (AS):**
  The Authentication Server performs the initial authentication and ticket for Ticket Granting Service.

- **Database:**
  The Authentication Server verifies the access rights of users in the database.

- **Ticket Granting Server (TGS):**
  The Ticket Granting Server issues the ticket for the Server



- **Step-1:**
  User login and request services on the host. Thus user requests for ticket-granting service.

- **Step-2:**
  Authentication Server verifies user's access right using database and then gives ticket-granting-ticket and session key. Results are encrypted using the Password of the user.

- **Step-3:**
  The decryption of the message is done using the password then send the ticket to Ticket Granting Server. The Ticket contains authenticators like user names and network addresses.

- **Step-4:**
  Ticket Granting Server decrypts the ticket sent by User and authenticator verifies the request then creates the ticket for requesting services from the Server.

- **Step-5:**
  The user sends the Ticket and Authenticator to the Server.

- **Step-6:**
  The server verifies the Ticket and authenticators then generate access to the service. After this User can access the services.

SAML:

Security Assertion Markup Language, or SAML, is a standardized way to tell external applications and services that a user is who they say they are.

SAML makes single sign-on (SSO) technology possible by providing a way to authenticate a user once and then communicate that authentication to multiple applications. The most current version of SAML is SAML 2.0.

Think of SAML authentication as being like an identification card: a short, standardized way to show who someone is. Instead of, say, conducting a series of DNA tests to confirm someone's identity, it is possible to just glance at their ID card.

How does SAML work?

A typical SSO authentication process involves these three parties:

- Principal (also known as the "subject")
- Identity provider
- Service provider

Principal/subject: This is almost always a human user who is trying to access a cloud-hosted application.

Identity provider: An identity provider (IdP) is a cloud software service that stores and confirms user identity, typically through a login process. Essentially, an IdP's role is to say, "I know this person, and here is what they are allowed to do."

An SSO system may in fact be separate from the IdP, but in those cases the SSO essentially acts as a representative for the IdP, so for all intents and purposes they are the same in a SAML workflow.

Service provider: This is the cloud-hosted application or service the user wants to use. Common examples include cloud email platforms such as Gmail and Microsoft Office 365, cloud storage services such as Google Drive and AWS S3, and communications apps such as Slack and Skype. Ordinarily a user would just log in to these services

directly, but when SSO is used, the user logs into the SSO instead, and SAML is used to give them access instead of a direct login.