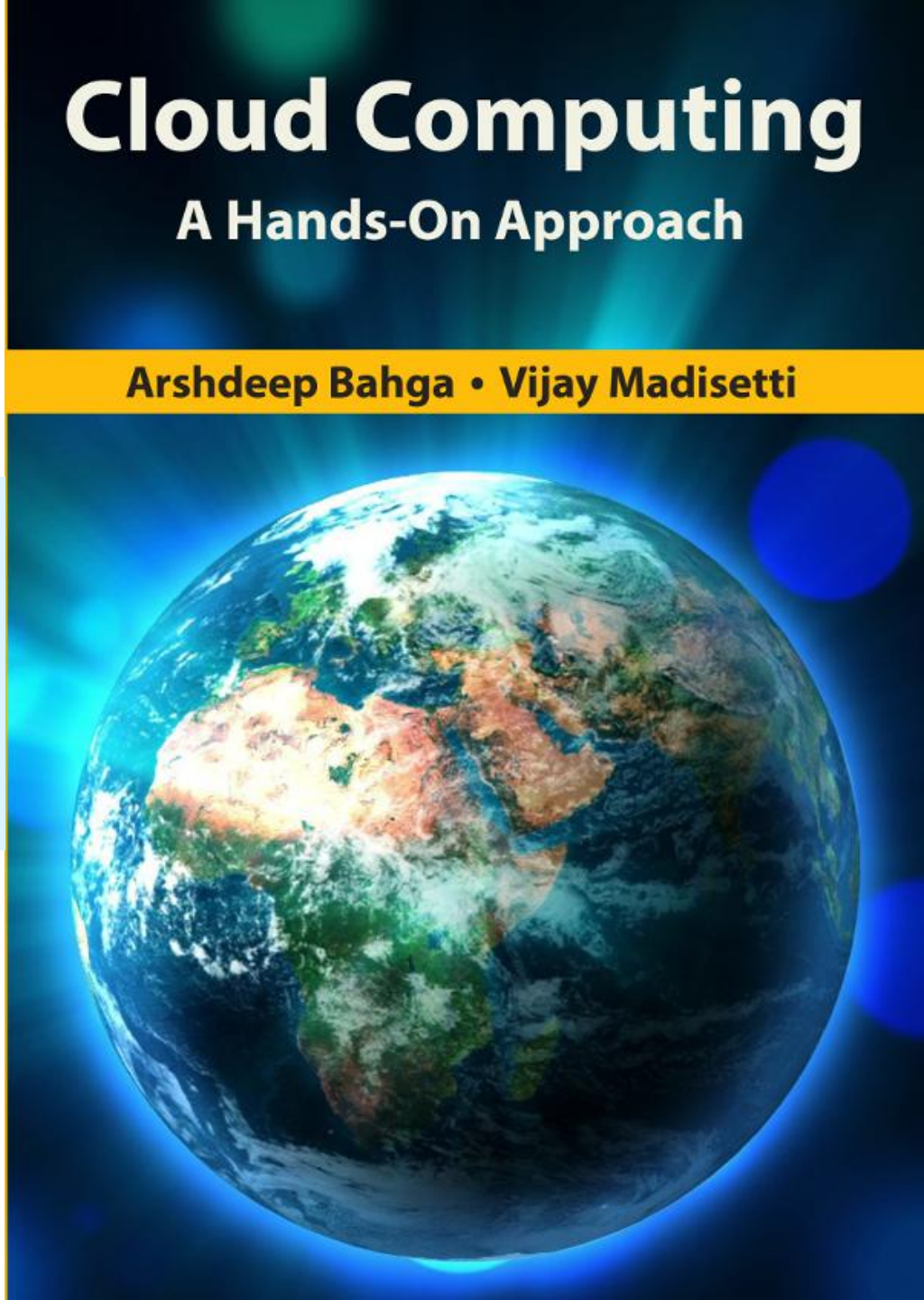
Chapter 12

Cloud Security



Outline

- Cloud security challenges
- Authorization
- Authentication
- Identify & Access Management
- Data Security
- Data Integrity
- Encryption & Key Management

Cloud Security Challenges

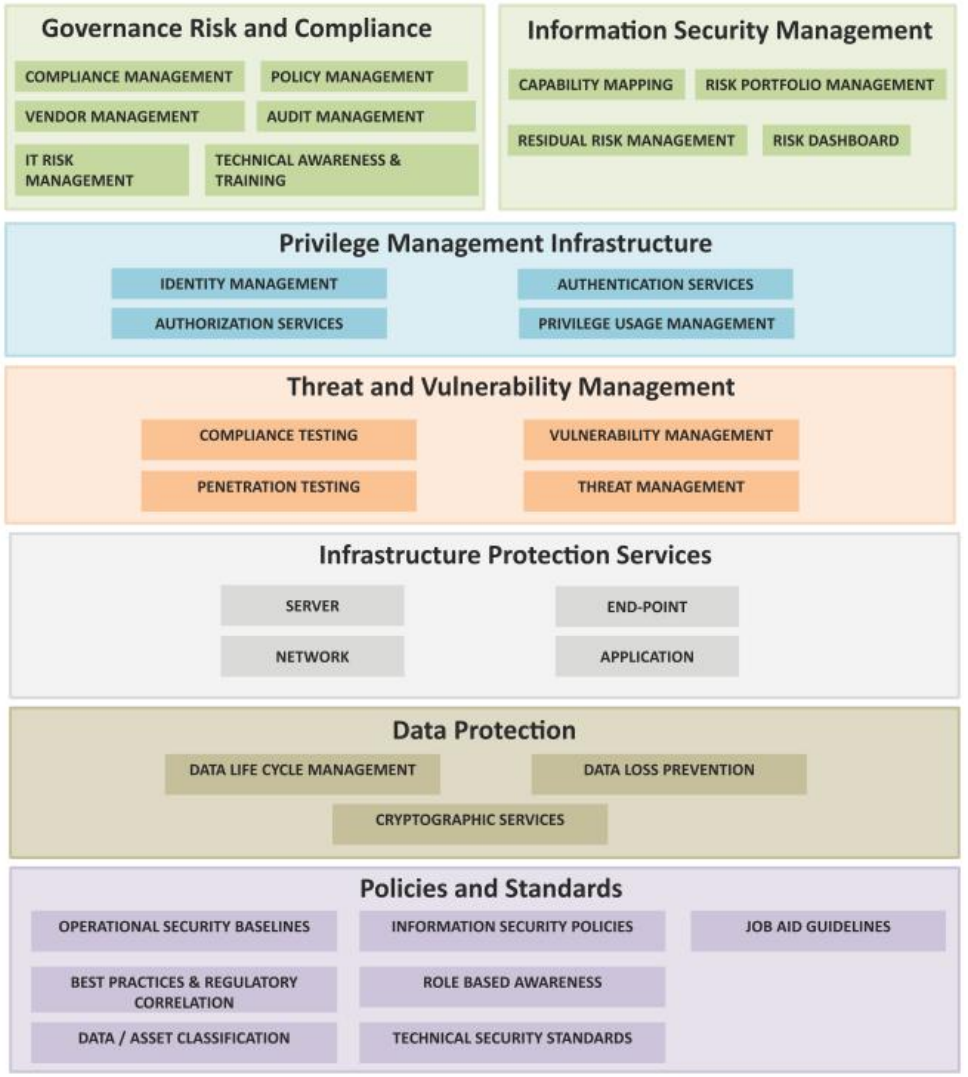
- Authentication
  - Authentication refers to digitally confirming the identity of the entity requesting access to some protected information.
  - In a traditional in-house IT environment authentication polices are under the control of the organization. However, in cloud computing environments, where applications and data are accessed over the internet, the complexity of digital authentication mechanisms increases rapidly.
- Authorization
  - Authorization refers to digitally specifying the access rights to the protected resources using access policies.
  - In a traditional in-house IT environment, the access policies are controlled by the organization and can be altered at their convenience.
  - Authorization in a cloud computing environment requires the use of the cloud service providers services for specifying the access policies.
- Security of data at rest
  - Due to the multi-tenant environments used in the cloud, the application and database servers of different applications belonging to different organizations can be provisioned side-by-side increasing the complexity of securing the data.
  - Appropriate separation mechanisms are required to ensure the isolation between applications and data from different organizations.

Cloud Security Challenges

- Security of data in motion
  - In traditional in-house IT environments all the data exchanged between the applications and users remains within the organization's control and geographical boundaries.
  - With the adoption of the cloud model, the applications and the data are moved out of the in-house IT infrastructure to the cloud provider.
  - Therefore, appropriate security mechanisms are required to ensure the security of data in, and while in, motion.
- Data Integrity
  - Data integrity ensures that the data is not altered in an unauthorized manner after it is created, transmitted or stored. Due to the outsourcing of data storage in cloud computing environments, ensuring integrity of data is important.
- Auditing
  - Auditing is very important for applications deployed in cloud computing environments.
  - In traditional in-house IT environments, organizations have complete visibility of their applications and accesses to the protected information.
  - For cloud applications appropriate auditing mechanisms are required to get visibility into the application, data accesses and actions performed by the application users, including mobile users and devices such as wireless laptops and smartphones.

CSA Cloud Security Architecture

- Cloud Security Alliance (CSA) provides a Trusted Cloud Initiative (TCI) Reference Architecture.
- TCI is a methodology and a set of tools that enable cloud application developers and security architects to assess where their internal IT and their cloud providers are in terms of security capabilities, and to plan a roadmap to meet the security needs of their business.
- Security and Risk Management (SRM) domain within the TCI Reference includes:
  - Governance, Risk Management, and Compliance
  - Information Security Management
  - Privilege Management Infrastructure
  - Threat and Vulnerability Management
  - Infrastructure Protection Services
  - Data Protection
  - Policies and Standards



Authentication

- Authentication refers to confirming the digital identity of the entity requesting access to some protected information.
- The process of authentication involves, but is not limited to, validating the at least one factor of identification of the entity to be authenticated.
- A factor can be something the entity or the user knows (password or pin), something the user has (such as a smart card), or something that can uniquely identify the user (such as fingerprints).
- In multifactor authentication more than one of these factors are used for authentication.
- There are various mechanisms for authentication including:
  - SSO
  - SAML-Token
  - OTP

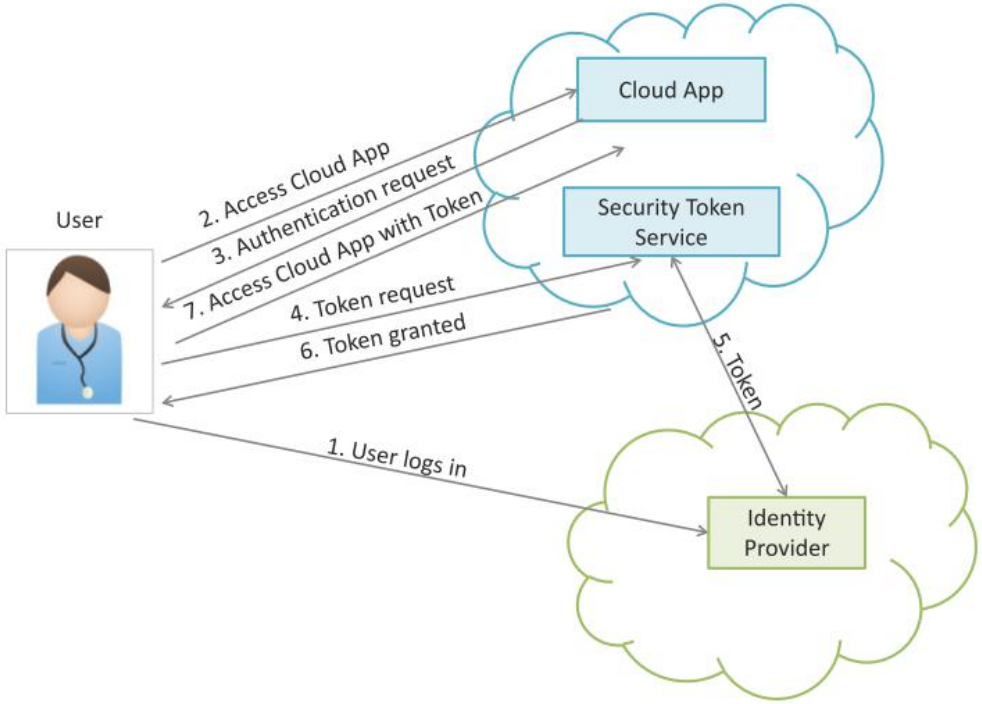
Single Sign-on (SSO)

- Single Sign-on (SSO) enables users to access multiple systems or applications after signing in only once, for the first time.
- When a user signs in, the user identity is recognized and there is no need to sign in again and again to access related systems or applications.
- Since different systems or applications may be internally using different authentication mechanisms, SSO upon receiving initial credential translates to different credentials for different systems or applications.
- The benefit of using SSO is that it reduces human error and saves time spent in authenticating with different systems or applications for the same identity.
- There are different implementation mechanisms:
  - SAML-Token
  - Kerberos

SAML-Token

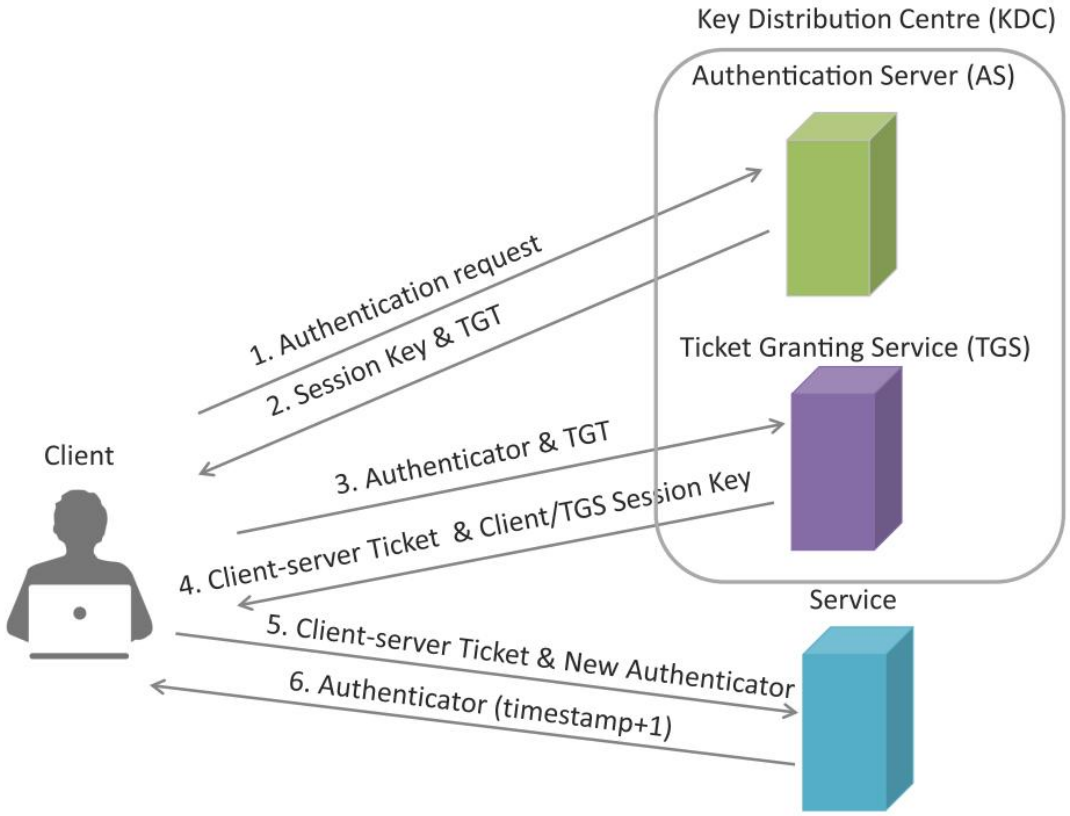
- Security Assertion Markup Language (SAML) is an XML-based open standard data format for exchanging security information (authentication and authorization data) between an identity provider and a service provider.

- SAML-token based SSO authentication
  - When a user tries to access the cloud application, a SAML request is generated and the user is redirected to the identity provider.
  - The identity provider parses the SAML request and authenticates the user. A SAML token is returned to the user, who then accesses the cloud application with the token.
  - SAML prevents man-in-the-middle and replay attacks by requiring the use of SSL encryption when transmitting assertions and messages.
  - SAML also provides a digital signature mechanism that enables the assertion to have a validity time range to prevent replay attacks.



Kerberos

- Kerberos is an open authentication protocol that was developed At MIT.
- Kerberos uses tickets for authenticating client to a service that communicate over an un-secure network.
- Kerberos provides mutual authentication, i.e. both the client and the server authenticate with each other.

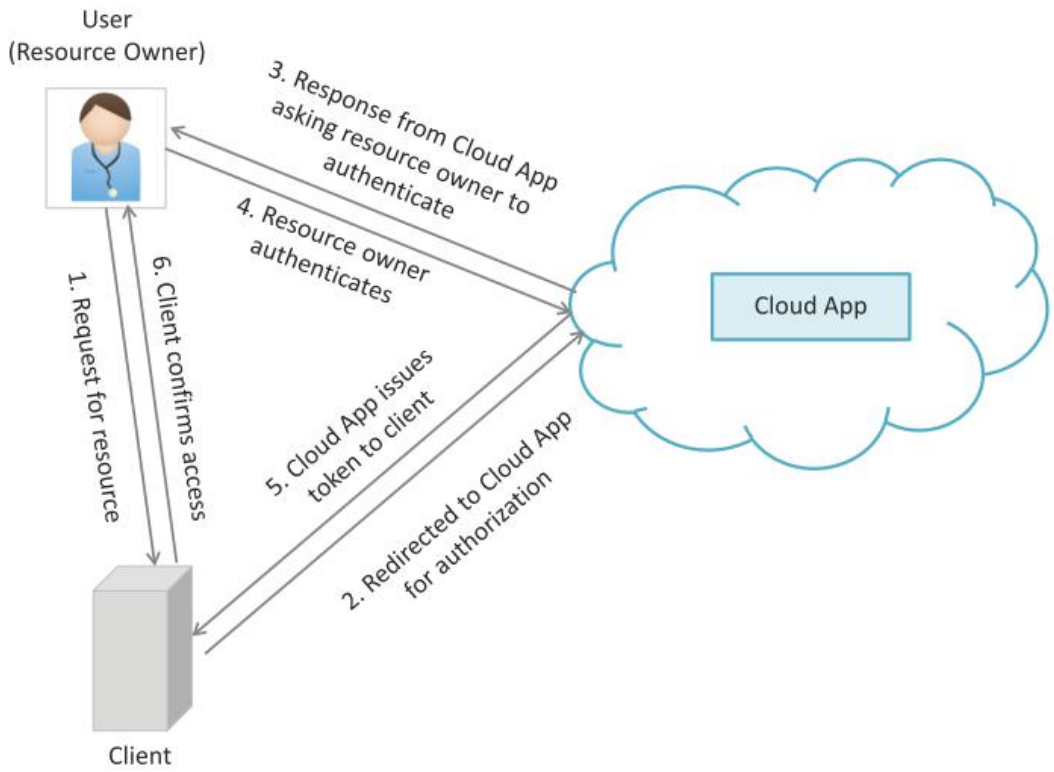


One Time Password (OTP)

- One time password is another authentication mechanism that uses passwords which are valid for single use only for a single transaction or session.
- Authentication mechanism based on OTP tokens are more secure because they are not vulnerable to replay attacks.
- Text messaging (SMS) is the most common delivery mode for OTP tokens.
- The most common approach for generating OTP tokens is time synchronization.
- Time-based OTP algorithm (TOTP) is a popular time synchronization based algorithm for generating OTPs.

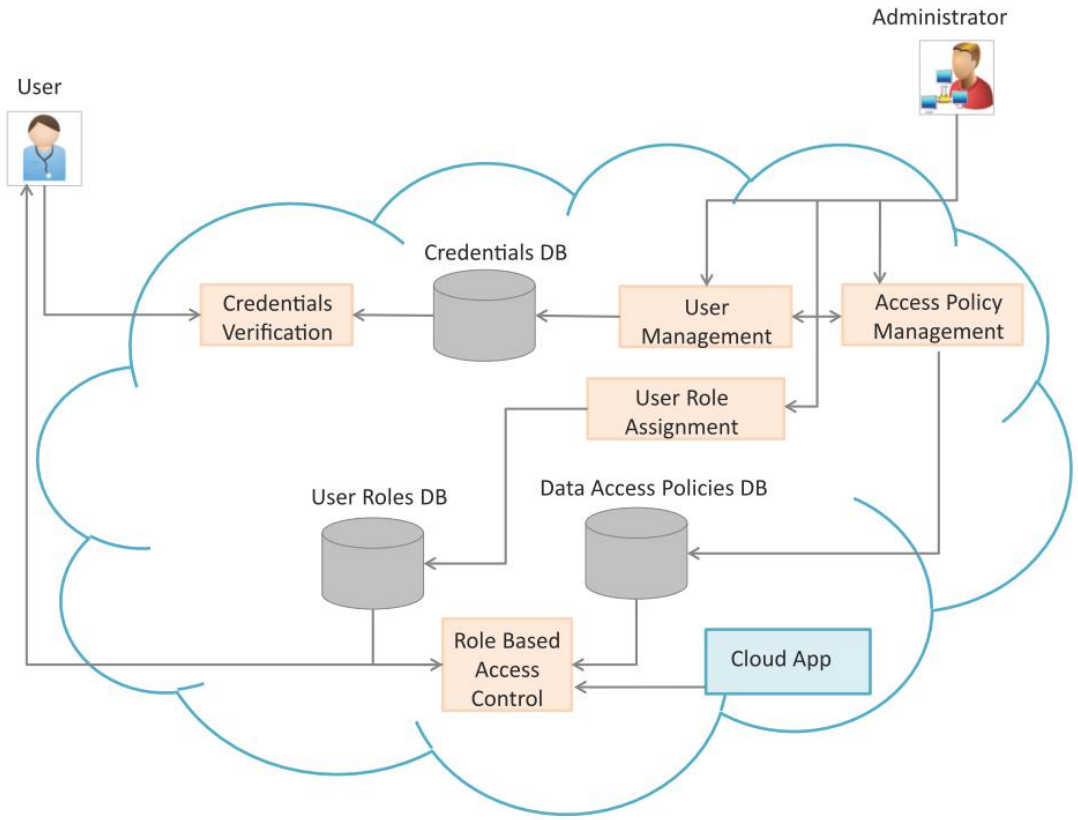
Authorization

- Authorization refers to specifying the access rights to the protected resources using access policies.
-
- OAuth
  - OAuth is an open standard for authorization that allows resource owners to share their private resources stored on one site with another site without handing out the credentials.
  - In the OAuth model, an application (which is not the resource owner) requests access to resources controlled by the resource owner (but hosted by the server).
  - The resource owner grants permission to access the resources in the form of a token and matching shared-secret.
  - Tokens make it unnecessary for the resource owner to share its credentials with the application.
  - Tokens can be issued with a restricted scope and limited lifetime, and revoked independently.



Identity & Access Management

- Identity management provides consistent methods for digitally identifying persons and maintaining associated identity attributes for the users across multiple organizations.
- Access management deals with user privileges.
- Identity and access management deal with user identities, their authentication, authorization and access policies.
- Federated Identity Management
  - Federated identity management allows users of one domain to securely access data or systems of another domain seamlessly without the need for maintaining identity information separately for multiple domains.
  - Federation is enabled through the use single sign-on mechanisms such as SAML token and Kerberos.
- Role-based access control
  - Used for restricting access to confidential information to authorized users.
  - These access control policies allow defining different roles for different users.



Securing Data at Rest

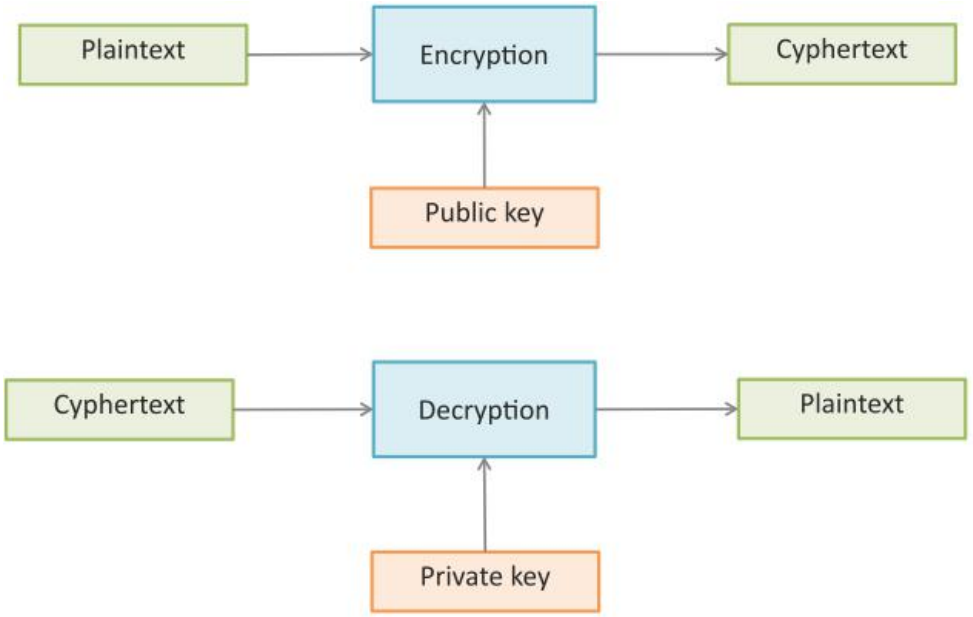
- Data at rest is the data that is stored in database in the form of tables/records, files on a file server or raw data on a distributed storage or storage area network (SAN).
- Data at rest is secured by encryption.
- Encryption is the process of converting data from its original form (i.e., plaintext) to a scrambled form (ciphertext) that is unintelligible. Decryption converts data from ciphertext to plaintext.
- Encryption can be of two types:
  - Symmetric Encryption (symmetric-key algorithms)
  - Asymmetric Encryption (public-key algorithms)

Symmetric Encryption

- Symmetric encryption uses the same secret key for both encryption and decryption.
- The secret key is shared between the sender and the receiver.
- Symmetric encryption is best suited for securing data at rest since the data is accessed by known entities from known locations.
- Popular symmetric encryption algorithms include:
  - Advanced Encryption Standard (AES)
  - Twofish
  - Blowfish
  - Triple Data Encryption Standard (3DES)
  - Serpent
  - RC6
  - MARS

Asymmetric Encryption

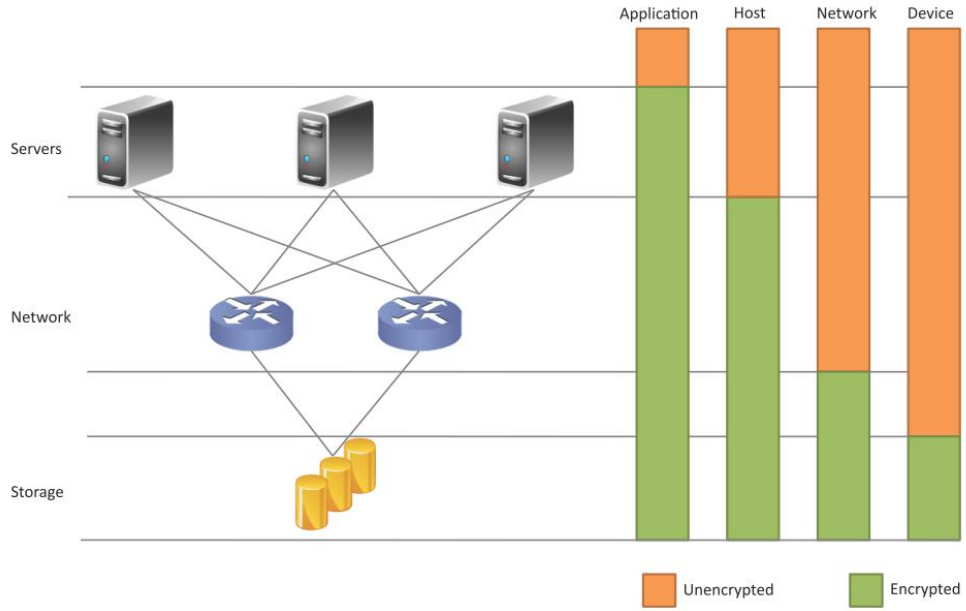
- Asymmetric encryption uses two keys, one for encryption (public key) and other for decryption (private key).
- The two keys are linked to each other such that one key encrypts plaintext to ciphertext and other decrypts ciphertext back to plaintext.
- Public key can be shared or published while the private key is known only to the user.
- Asymmetric encryption is best suited for securing data that is exchanged between two parties where symmetric encryption can be unsafe because the secret key has to be exchanged between the parties and anyone who manages to obtain the secret key can decrypt the data.
- In asymmetric encryption a separate key is used for decryption which is kept private.



Encryption Levels

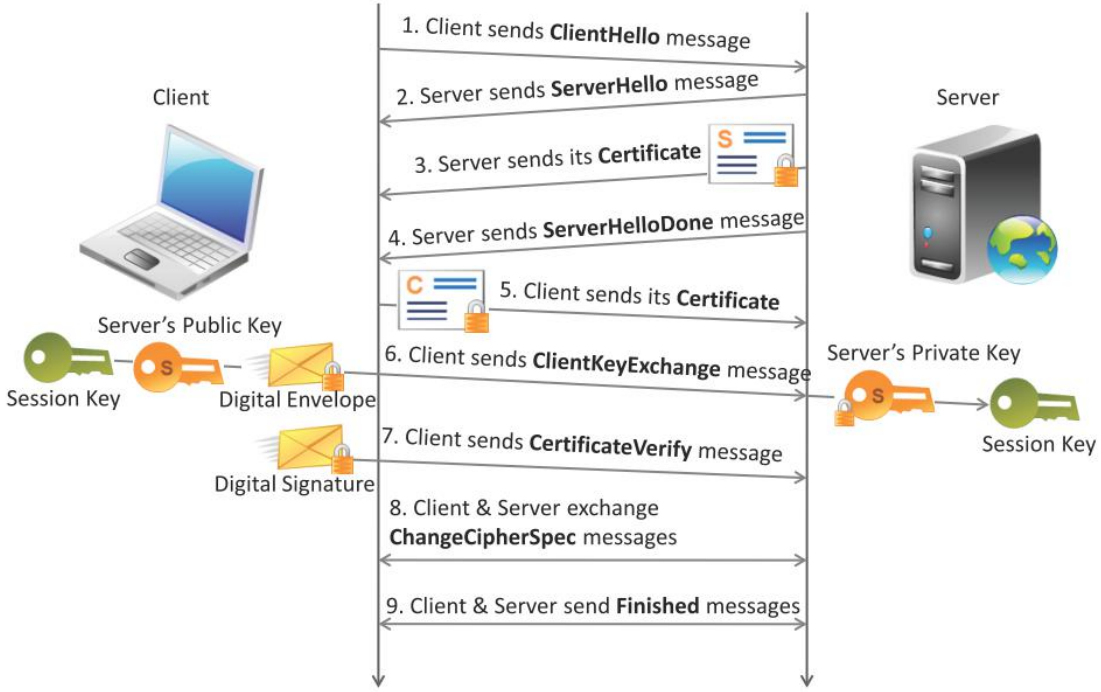
Encryption can be performed at various levels:

- Application
  - Application level encryption involves encrypting application data right at the point where it originates i.e. within the application.
  - Application level encryption provides security at the level of both the operating system and from other applications.
  - An application encrypts all data generated in the application before it flows to the lower levels and presents decrypted data to the user.
- Host
  - In host-level encryption, encryption is performed at the file-level for all applications running on the host.
  - Host level encryption can be done in software in which case additional computational resource is required for encryption or it can be performed with specialized hardware such as a cryptographic accelerator card.
- Network
  - Network-level encryption is best suited for cases where the threats to data are at the network or storage level and not at the application or host level.
  - Network-level encryption is performed when moving the data form a creation point to its destination using a specialized hardware that encrypts all incoming data in real-time.
- Device
  - Device-level encryption is performed on a disk controller or a storage server.
  - Device level encryption is easy to implement and is best suited for cases where the primary concern about data security is to protect data residing on storage media.



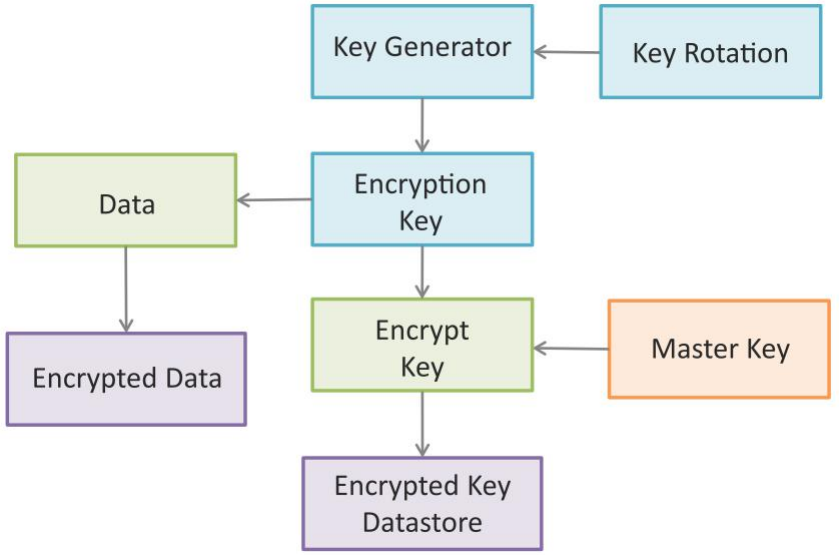
Securing Data in Motion

- Securing data in motion, i.e., when the data flows between a client and a server over a potentially insecure network, is important to ensure data confidentiality and integrity.
- Data confidentiality means limiting the access to data so that only authorized recipients can access it.
- Data integrity means that the data remains unchanged when moving from sender to receiver.
- Data integrity ensures that the data is not altered in an unauthorized manner after it is created, transmitted or stored.
- Transport Layer Security (TLS) and Secure Socket Layer (SSL) are the mechanisms used for securing data in motion.
- TLS and SSL are used to encrypt web traffic using Hypertext Transfer Protocol (HTTP).
- TLS and SSL use asymmetric cryptography for authentication of key exchange, symmetric encryption for confidentiality and message authentication codes for message integrity.



Key Management

- Management of encryption keys is critical to ensure security of encrypted data.
- The key management lifecycle involves different phases including:
  - Creation
  - Backup
  - Deployment
  - Monitoring
  - Rotation
  - Expiration
  - Archival
  - Destruction
- Key Management Approach (example)
  - All keys for encryption must be stored in a data store which is separate and distinct from the actual data store.
  - Additional security features such as key rotation and key encrypting keys can be used.
  - Keys can be automatically or manually rotated.
  - In the automated key change approach, the key is changed after a certain number of transactions.
  - All keys can themselves be encrypted using a master key.



Auditing

- Auditing is mandated by most data security regulations.
- Auditing requires that all read and write accesses to data be logged.
- Logs can include the user involved, type of access, timestamp, actions performed and records accessed.
- The main purpose of auditing is to find security breaches, so that necessary changes can be made in the application and deployment to prevent a further security breach.
- The objectives of auditing include:
  - Verify efficiency and compliance of identity and access management controls as per established access policies.
  - Verifying that authorized users are granted access to data and services based on their roles.
  - Verify whether access policies are updated in a timely manner upon change in the roles of the users.
  - Verify whether the data protection policies are sufficient.
  - Assessment of support activities such as problem management.

Further Reading

- CSA Trusted Cloud Initiative, https://research.cloudsecurityalliance.org/tci/
- Keberos, http://web.mit.edu/kerberos/
- TOTP: Time-Based One-Time Password Algorithm http://tools.ietf.org/html/rfc6238
- OAuth community site, http://oauth.net/
- The OAuth 2.0 Authorization Framework, http://tools.ietf.org/html/rfc6749
- Python OAuth2, https://github.com/simplegeo/python-oauth2