Q) What are different **cloud security challenges**?
Ans:
Cloud computing security challenges fall into **three broad** categories:

Data Protection: Securing your data both at rest and in transit

User Authentication: Limiting access to data and monitoring who accesses the data

Disaster and Data Breach: Contingency Planning

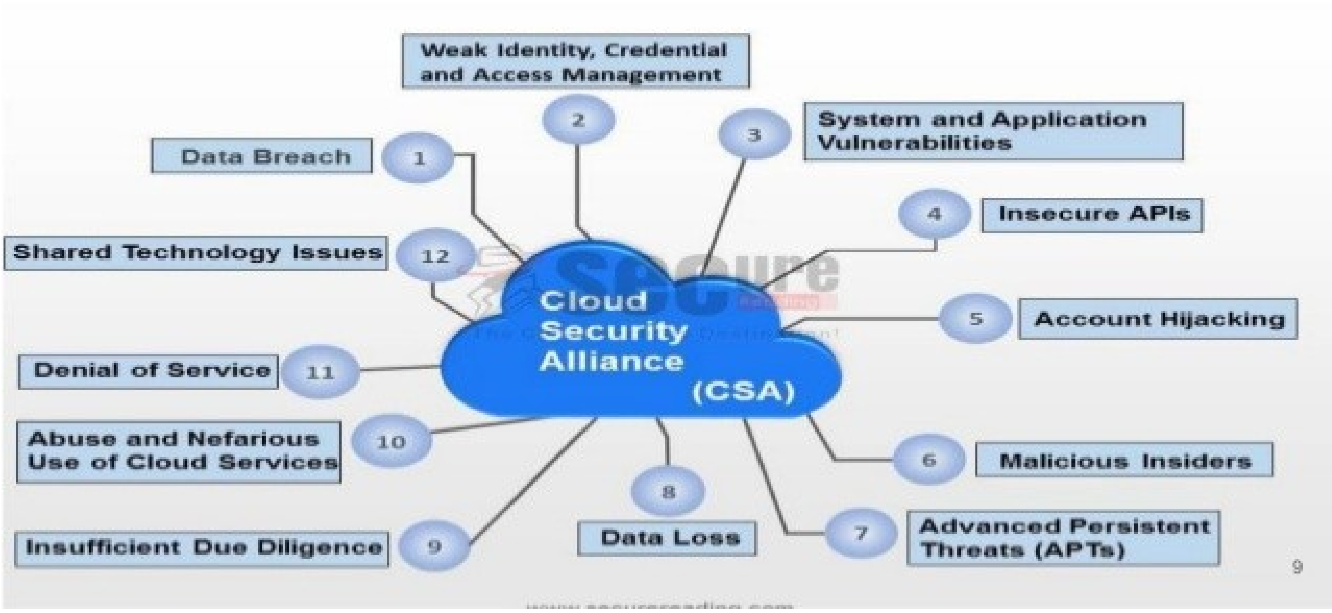
Data Protection
Implementing a cloud computing strategy means placing critical data in the hands of a third party, so ensuring the data remains secure both at rest (data residing on storage media) as well as when in transit is of paramount importance. Data needs to be encrypted at all times, with clearly defined roles when it comes to who will be managing the encryption keys. In most cases, the only way to truly ensure confidentiality of encrypted data that resides on a cloud provider's storage servers is for the client to own and manage the data encryption keys.

User Authentication
Data resting in the cloud needs to be accessible only by those authorized to do so, making it critical to both restrict and monitor who will be accessing the company's data through the cloud. In order to ensure the integrity of user authentication, companies need to be able to view data access logs and audit trails to verify that only authorized users are accessing the data. These access logs and audit trails additionally need to be secured and maintained for as long as the company needs or legal purposes require. As with all cloud computing security challenges, it's the responsibility of the customer to ensure that the cloud provider has taken all necessary security measures to protect the customer's data and the access to that data.

Contingency Planning
With the cloud serving as a single centralized repository for a company's mission-critical data, the risks of having that data compromised due to a data breach or temporarily made unavailable due to a natural disaster are real concerns. Much of the liability for the disruption of data in a cloud ultimately rests with the company whose mission-critical operations depend on that data, although liability can and should be negotiated in a contract with the services provider prior to commitment. A comprehensive security assessment from a neutral third-party is strongly recommended as well.

Companies need to know how their data is being secured and what measures the service provider will be taking to ensure the integrity and availability of that data should the unexpected occur. Additionally, companies should also have contingency plans in place in the event their cloud provider fails or goes bankrupt. Can the data be easily retrieved and migrated to a new service provider or to a non-cloud strategy if this happens? And what happens to the data and the ability to access that data if the provider gets acquired by another company?

Challenges in Cloud Security


Weak Identity, Credential and Access Management
2
System and Application Vulnerabilities
3
Data Breach
1
4
Insecure APIs
Shared Technology Issues
12
Cloud Security Alliance (CSA)
5
Account Hijacking
Denial of Service
11
Abuse and Nefarious Use of Cloud Services
10
6
Malicious Insiders
8
Data Loss
7
Advanced Persistent Threats (APTs)
Insufficient Due Diligence
9
9
www.securereading.com

Insecure Interfaces/APIs

CSPs often provide a number of application programming interfaces (APIs) and interfaces for their customers. In general, these interfaces are well-documented in an attempt to make them easily-usable for a CSP's customers.

However, this creates potential issues if a customer has not properly secured the interfaces for their cloud-based infrastructure. The documentation designed for the customer can also be used by a cybercriminal to identify and exploit potential methods for accessing and exfiltrating sensitive data from an organization's cloud environment.

Hijacking of Accounts

Many people have extremely weak password security, including password reuse and the use of weak passwords. This problem exacerbates the impact of phishing attacks and data breaches since it enables a single stolen password to be used on multiple different accounts.

Account hijacking is one of the more serious cloud security issues as organizations are increasingly reliant on cloud-based infrastructure and applications for core business functions. An attacker with an employee's credentials can access sensitive data or functionality, and compromised customer credentials give full control over their online account. Additionally, in the cloud, organizations often lack the ability to identify and respond to these threats as effectively as for on-premises infrastructure.

Malicious Insiders

Insider threats are a major security issue for any organization. A malicious insider already has authorized access to an organization's network and some of the sensitive resources that it contains. Attempts to gain this level of access are what reveals most attackers to their target, making it hard for an unprepared organization to detect a malicious insider.

On the cloud, detection of a malicious insider is even more difficult. With cloud deployments, companies lack control over their underlying infrastructure, making many traditional security solutions less effective. This, along with the fact that cloud-based infrastructure is directly accessible from the public Internet and often suffers from security misconfigurations, makes it even more difficult to detect malicious insiders.

Data Loss/Leakage

Cloud-based environments make it easy to share the data stored within them. These environments are accessible directly from the public Internet and include the ability to share data easily with other parties via direct email invitations or by sharing a public link to the data.

The ease of data sharing in the cloud – while a major asset and key to collaboration in the cloud – creates serious concerns regarding data loss or leakage. In fact, 69% of organizations point to this as their greatest cloud security concern. Data sharing using public links or setting a cloud-based repository to public makes it accessible to anyone with knowledge of the link, and tools exist specifically for searching the Internet for these unsecured cloud deployments.

DDoS and Denial-of-Service Attacks

As more and more businesses and operations move to the cloud, cloud providers are becoming a bigger target for malicious attacks. Distributed denial of service (DDoS) attacks are more common than ever before. Verisign reported IT services, cloud platforms (PaaS) and SaaS was the most frequently targeted industry during the first quarter of 2015.

A DDoS attack is designed to overwhelm website servers so it can no longer respond to legitimate user requests. If a DDoS attack is successful, it renders a website useless for hours, or even days. This can result in a loss of revenue, customer trust and brand authority.

Complementing cloud services with DDoS protection is no longer just good idea for the enterprise; it's a necessity. Websites and web-based applications are core components of 21st century business and require state-of-the-art cybersecurity.

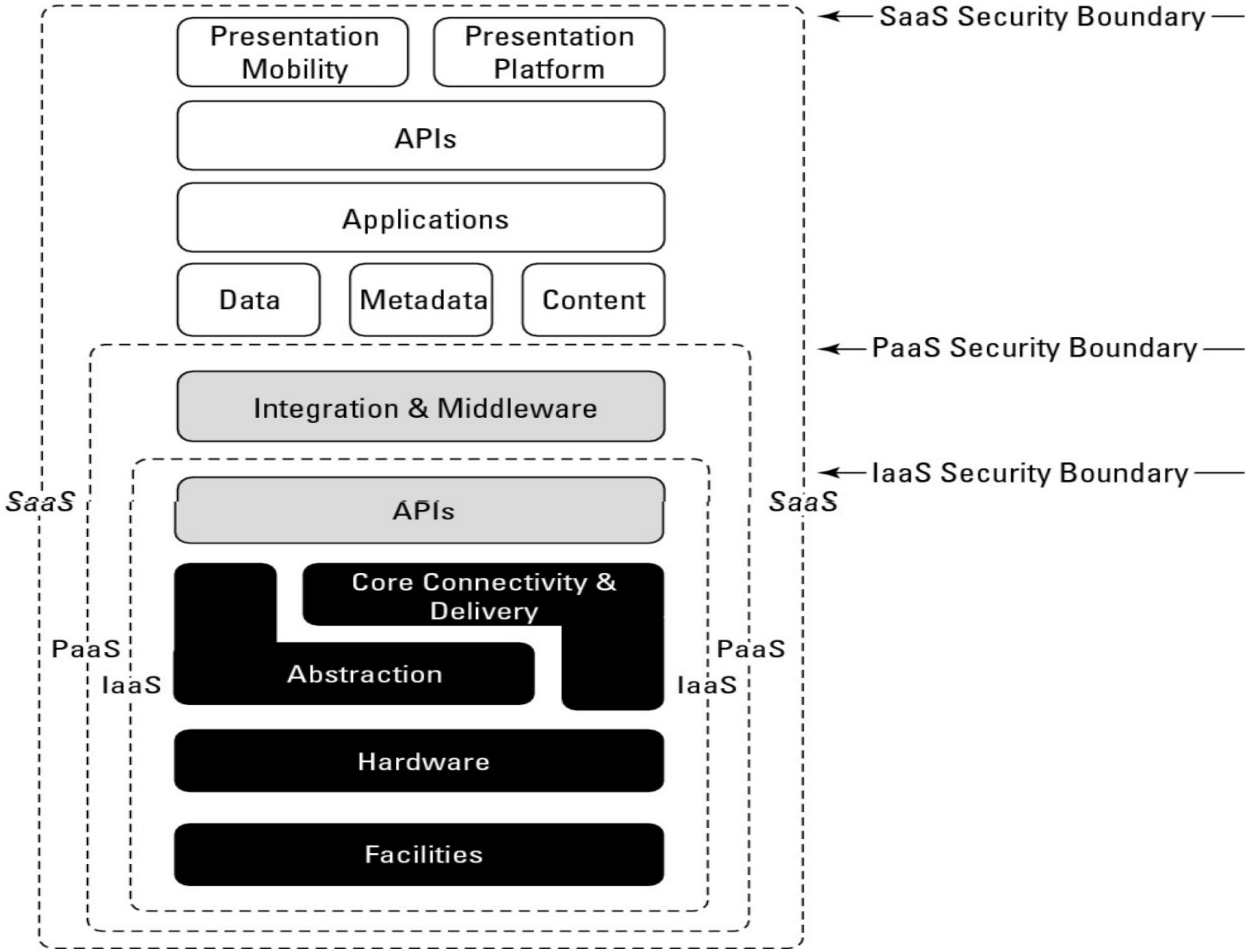
Data breaches

Known data breaches in the U.S. hit a record-high of 738 in 2014, according to the Identity Theft Research Center, and hacking was (by far) the number one cause. That’s an incredible statistic and only emphasizes the growing challenge to secure sensitive data.

Traditionally, IT professionals have had great control over the network infrastructure and physical hardware (firewalls, etc.) securing proprietary data. In the cloud (in all scenarios including private cloud, public cloud, and hybrid cloud situations), some of those security controls are relinquished to a trusted partner meaning cloud infrastructure can increase security risks. Choosing the right vendor, with a strong record of implementing strong security measures, is vital to overcoming this challenge.

Security in Cloud Computing

Cloud Security Reference Model



Security service boundary

The CSA functional cloud computing hardware/software stack is the Cloud Reference Model. IaaS is the lowest level service, with PaaS and SaaS the next two services above. As one move upward in the stack, each service model inherits the capabilities of the model beneath it, as well as all the inherent security concerns and risk factors. IaaS supplies the infrastructure; PaaS adds application development frameworks, transactions, and control structures; and SaaS is an operating environment with applications, management, and the user interface. As one ascend the stack, IaaS has the least levels of integrated functionality and the lowest levels of integrated security, and SaaS has the most. The most important lesson from this discussion of architecture is that each different type of cloud service delivery model creates a security boundary at which the cloud service provider's responsibilities end and the customer's responsibilities begin. Any security mechanism below the security boundary must be built into the system, and any security mechanism above must be maintained by the customer. As one move up the stack, it becomes more important to make sure that the type and level of security is part of your Service Level Agreement.

How Security gets integrated

In the SaaS model, the vendor provides security as part of the Service Level Agreement, with the compliance, governance, and liability levels stipulated under the contract for the entire stack. For the PaaS model, the security boundary may be defined for the vendor to include the software framework and middleware layer. In the PaaS model, the customer would be responsible for the security of the application and UI at the top of the stack. The model with the least built-in security is IaaS, where everything that involves software of any kind is the customer's problem. Numerous definitions of services tend to muddy this picture by adding or removing elements of the various functions from any particular offering, thus blurring which party has responsibility for which features, but the overall analysis is still useful.

Cloud Security

- 80% of enterprises consider security the #1 inhibitor to cloud adoptions
- 48%of enterprises are concerned about the reliability of clouds
- 33%of respondents are concerned with cloud interfering with their ability to comply with regulations.

Security is the top concern for the adoption of cloud services.

Understanding Security Risks

- IT security is a very complicated area of cloud computing for three reasons:

  –Security is trusted to the cloud provider; therefore, if the provider has not done a good job, there may be problems

  –Security is difficult to monitor, so problems may not be apparent until there is a problem

  –Measuring the quality of the cloud provider's security approach may be difficult because many cloud providers do not expose their infrastructure to customers

- Approximately70% of security breaches are caused by insiders, (or people who get help from insiders)*

  –The security approach must deal with internal and external threats

Understanding Security Risks

- Often times with a cloud service agreement (contract), the agreement is crafted to protect the service provider, not the cloud customer

–Cloud customers must have a deep level of understanding the contract

IT security in cloud computing adds at least one critical layer of complexity the consumer, are trusting security to an external source. This trusted relationship may add the challenge of monitoring and validating the security of the cloud provider, especially if the provider does not wish to expose their internal infrastructure to customers.

Network Security

Basic Terminology

Plaintext:

Plaintext is the text to be encrypted.

Ciphertext:

The encrypted output

Encryption:

The process by which plaintext is converted into ciphertext

Encryption Algorithm:

The sequence of data processing steps that go into transforming plaintext into ciphertext. Various parameters used by an encryption algorithm are derived from a secret key. In cryptography for commercial and other civilian applications, the encryption and decryption algorithms are made public.

Secret Key:

A secret key is used to set some or all of the various parameters used by the encryption algorithm. The important thing to note is that, in classical cryptography, the same secret key is used for encryption and decryption. It is for this reason that classical cryptography is also referred to as symmetric key cryptography. On the other hand, in the more modern cryptographic algorithms, the encryption and decryption keys are not only different, but also one of them is placed in the public domain. Such algorithms are commonly referred to as asym- metric key cryptography, public key cryptography, etc.

Basic Terminology

Decryption Algorithm:

The sequence of data processing steps that go into transforming ciphertext back into plaintext. In classical cryptography, the various parameters used by a decryption algorithm are derived from the same secret key that was used in the encryption algorithm.

Cryptography:

The many schemes available today for encryption and decryption

Cryptographic system:

Any single scheme for encryption and decryption

cipher:

A cipher means the same thing as a “cryptographic system”

block cipher:

A block cipher processes a block of input data at a time and produces a cipher text block of the same size.

Basic Terminology

Stream cipher:

A stream cipher encrypts data on the fly, usually one byte at a time.

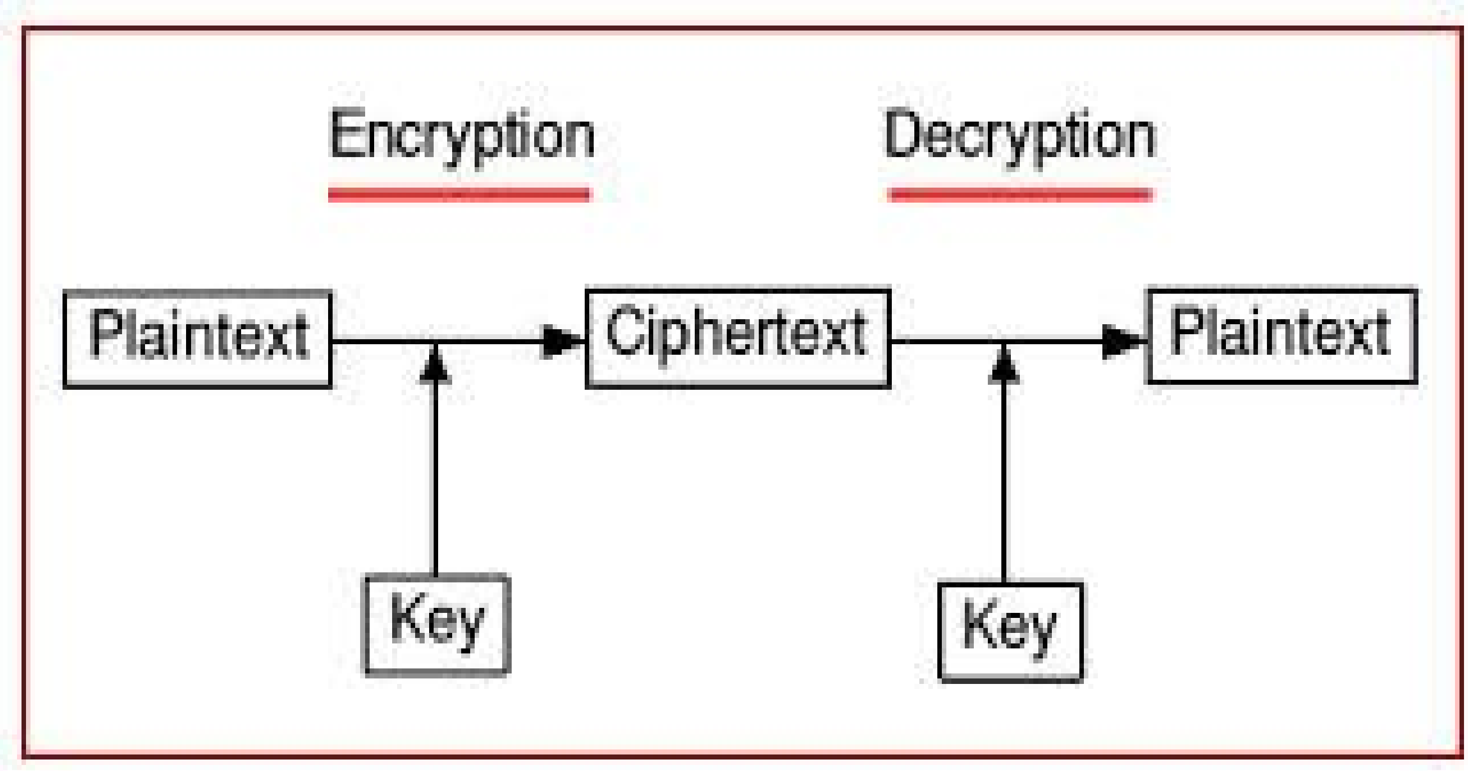
Cryptanalysis:

Means "breaking the code". Cryptanalysis relies on a knowledge of the encryption algorithm (that for civilian applications should be in the public domain) and some knowledge of the possible structure of the plaintext (such as the structure of a typical inter-bank financial transaction) for a partial or full reconstruction of the plaintext from ciphertext. Additionally, the goal is to also infer the key for decryption of future messages. The precise methods used for cryptanalysis depend on whether the "attacker" has just a piece of ciphertext, or pairs of plaintext and ciphertext, how much structure is possessed by the plaintext, and how much of that structure is known to the attacker. All forms of cryptanalysis for classical encryption exploit the fact that some aspect of the structure of plaintext may survive in the ciphertext.

Cryptology:

Cryptography and cryptanalysis together constitute the area of cryptology

Encryption and Decryption



Cryptography

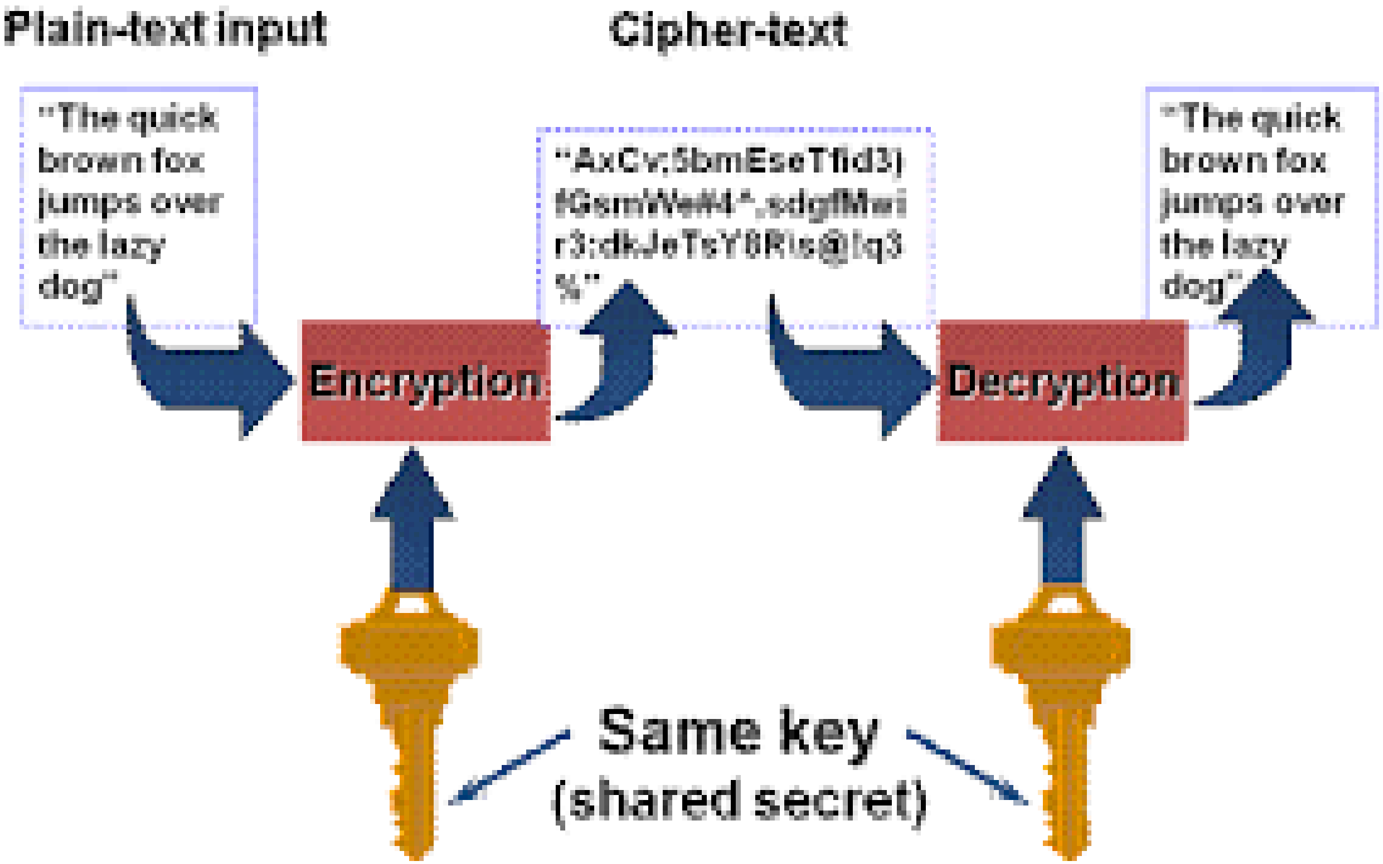
Cryptographic systems are generically classified along three independent dimensions:

1. The type of operations used for transforming plaintext to ciphertext. All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (that is, that all operations be reversible). Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions.

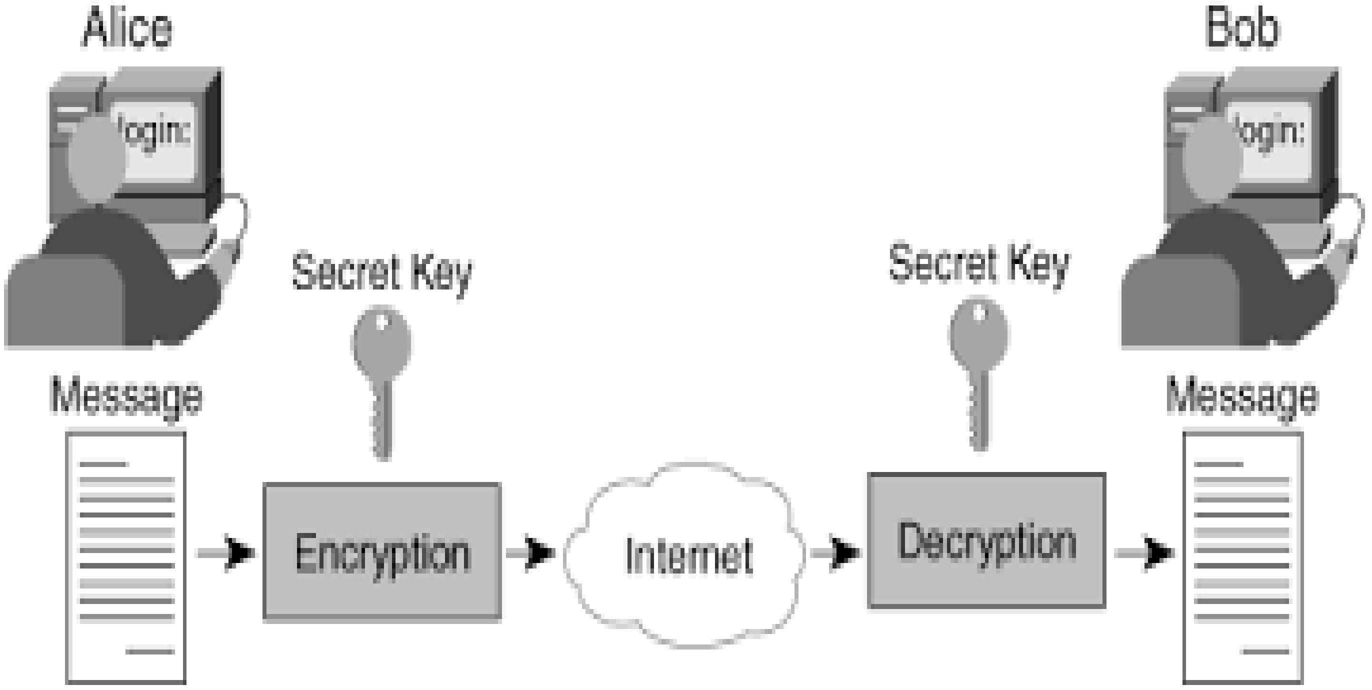
2. The number of keys used. If both sender and receiver use the same key,the system is referred to as symmetric, single-key, secret-key,or conventional encryption. If the sender and receiver each use a different key, the system is referred to as asymmetric, two-key, or public-key encryption.

3. The way in which the plaintext is processed. A block cipher processes the input one block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.

Symmetric Key Cryptography


Plain-text input
Cipher-text
"The quick brown fox jumps over the lazy dog"
"AxCv;5bmEseTfid3) fGsmWe#4^,sdgfMwi r3:dkJeTsY8R\s@!q3 %"
"The quick brown fox jumps over the lazy dog"
Encryption
Decryption
Same key
(shared secret)

Symmetric Key Cryptography

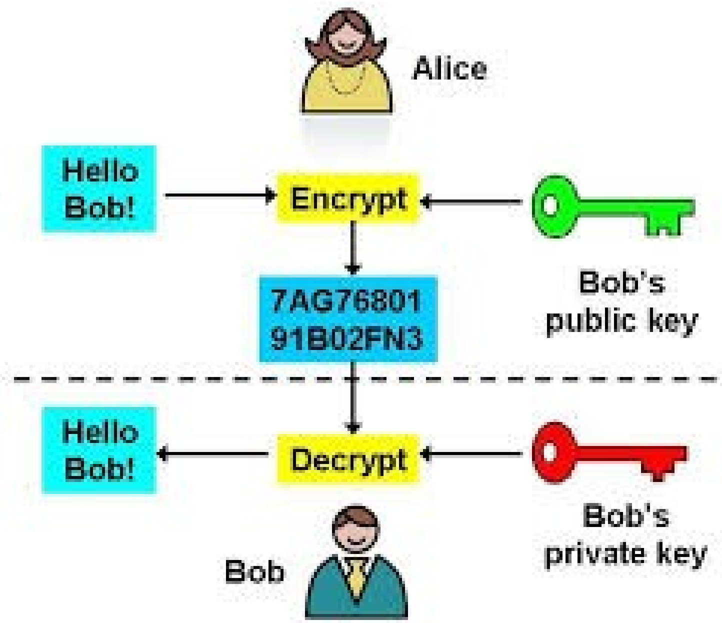


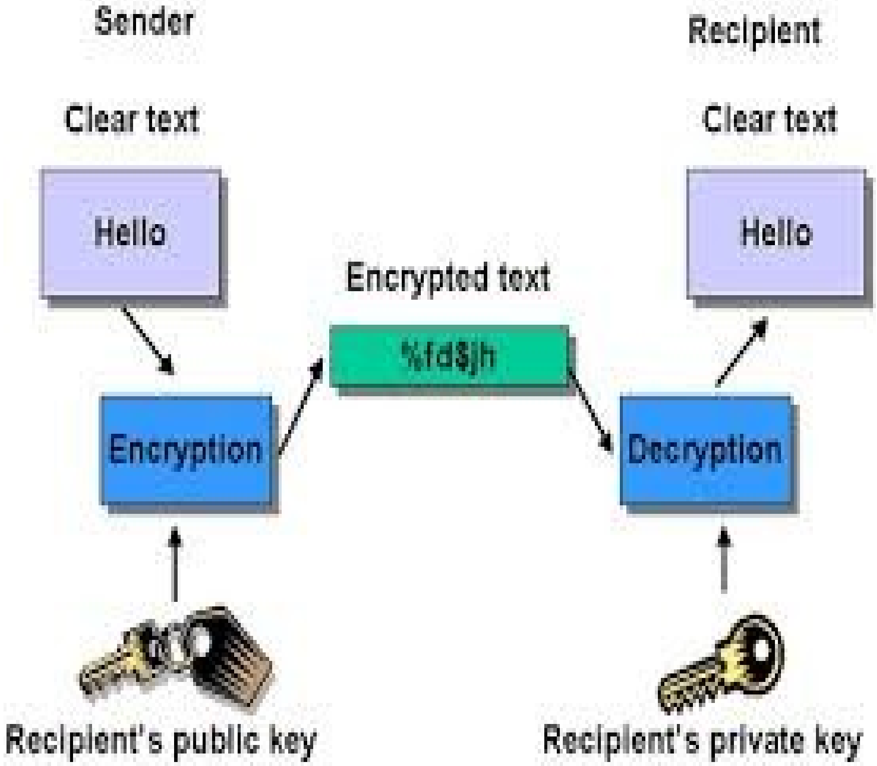
Symmetric Key Cryptography

An encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message. Symmetric-key systems are simpler and faster, but their main drawback is that the two parties must somehow exchange the key in a secure way. Symmetric-key cryptography is sometimes called secret-key cryptography.

The first major symmetric algorithm developed for computers in the United States was the Data Encryption Standard (DES), approved for use in the 1970s. The DES uses a 56-bit key. Because computers have become increasingly faster since the '70s, security experts no longer consider DES secure -- although a 56-bit key offers more than 70 quadrillion possible combinations (70,000,000,000,000,000), an attack of brute force (simply trying every possible combination in order to find the right key) could easily decipher encrypted data in a short while. DES has since been replaced by the Advanced Encryption Standard (AES), which uses 128-, 192- or 256-bit keys. Most people believe that AES will be a sufficient encryption standard for a long time coming.

Asymmetric Key Cryptography





Asymmetric Key Cryptography

Asymmetric-key cryptography, also known as **Public Key cryptography**, is a class of cryptographic algorithms which requires two separate keys, one of which is *secret* (or *private*) and one of which is *public*. Although different, the two parts of this key pair are mathematically linked. The public key is used to encrypt plaintext; whereas the private key is used to decrypt ciphertext. The term "asymmetric" stems from the use of different keys to perform these opposite functions, each the inverse of the other – as contrasted with conventional ("symmetric") cryptography which relies on the same key to perform both).

An important element to the public key system is that the public and private keys are related in such a way that only the public key can be used to encrypt messages and only the corresponding private key can be used to decrypt them. Moreover, it is virtually impossible to deduce the private key if you know the public key. Public-key systems, are becoming popular for transmitting information via the Internet. They are extremely secure and relatively simple to use. Public key cryptography was invented in 1976 by Whitfield Diffie and Martin Hellman. For this reason, it is sometime called *Diffie-Hellman encryption.* It is also called *asymmetric encryption* because it uses two keys instead of one key (*symmetric encryption*).

Identity and key management

An Identity and Access Management (IAM) system defines and manages user identities and access permissions. Users of IAM include customers (customer identity management) and employees (employee identity management). With IAM technologies, IT managers can ensure that users are who they say they are (authentication) and that users access the applications and resources they have permission to use (authorization).

4 Key Benefits of Identity and Access Management Systems

1. **Eliminating weak passwords**—research shows over 80% of data breaches are caused by stolen, default, or weak passwords. IAM systems enforce best practices in credential management, and can practically eliminate the risk that users will use weak or default passwords. They also ensure users frequently change passwords.
2. **Mitigating insider threats**—a growing number of breaches is caused by insiders. IAM can limit the damage caused by malicious insiders, by ensuring users only have access to the systems they work with, and cannot escalate privileges without supervision.
3. **Advanced tracking of anomalies**—modern IAM solutions go beyond simple credential management, and include technologies such as machine learning, artificial intelligence, and risk-based authentication, to identify and block anomalous activity.

4. Multi-factor security—IAM solutions help enterprises progress from two-factor to three-factor authentication, using capabilities like iris scanning, fingerprint sensors, and face recognition.

Repudiate - मुकर जाना

Digital Signature -

⇒ Why?

- It must verify the author & the date & time of the signature
- It must to authenticate the contents at the the time of ~~sign~~ signature
- It must be verifiable by third parties to resolve disputes

⇒ Digital Signature -

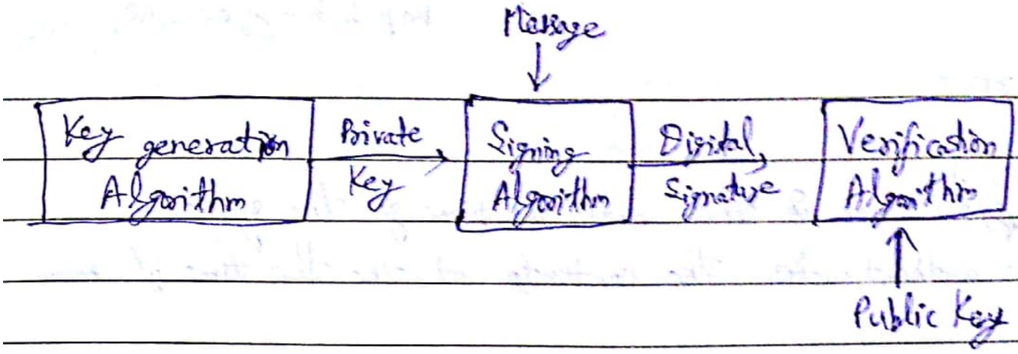
- One of the major application of public key Cryptography
- For message authentication, message integrity & non-repudiation
- A digital signature is an authentication mechanism that enables the creater of a message to attach a code that acts a signature.

Properties of Digital Signature -

- Signature must be a bit pattern that depends on the message being signed. (i.e. depends on message & digital signature should vary with message)
- It must use some ~~private~~ unique information of sender, to prevent both forgery & denial.
- It must be relatively easy to produce the digital signature.
- It must be relatively easy to recognize & verify the digital signature.
- It must be practical to retain a copy of the digital signature in storage.

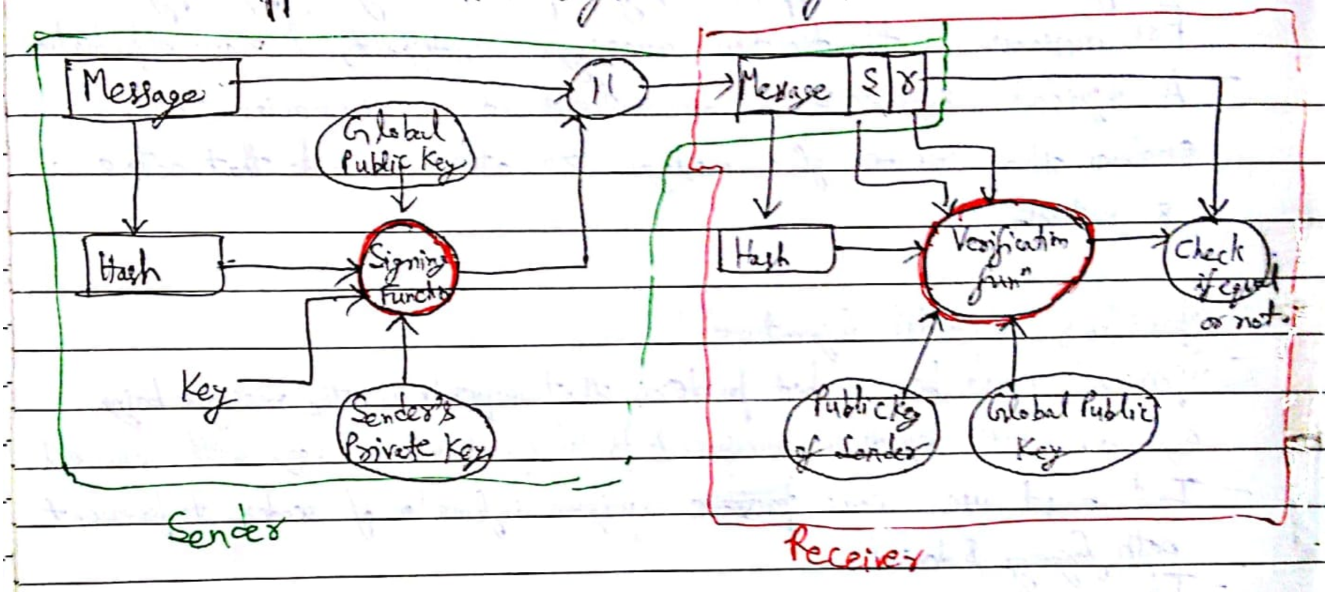
Algorithm -

- There must be a key generation algorithm.
  - To generate private key i.e. the unique information.
- There must be a signing Algorithm.
  - Inputs = Message & private key , Output = Digital Signature
- There must be a verification Algorithm.
  - At the receiver side to verify the signature using the message, public key & sign



⇒ Digital Signature Algorithm (~~DSS~~) -

DSS approach to Digital Signature -



⇒ Global Public elements

→ p : a prime no., a multiple of 64 between 512 & 1024 bits length

→ q : a prime divisor of $p-1$, bit length of 160 bits

→ g : $g = h^{(p-1)/q} \text{ mod } p$, $1 < h < p-1$

Prime divisor means it is a prime no. & q divides $(p-1)$ without leaving remainder

⇒ User's Private Key -

→ x random integer where $0 < x < q$

⇒ User's Public Key -

→ $y = g^x \text{ mod } p$

⇒ User's per message secret no.

→ k = random integer, where $0 < k < q$

Now let's look at Signing & Verification

=> Signing -

$r = (g^k \bmod p) \bmod q$

$s = [k^{-1}(H(M) + xr)] \bmod q$

Signature = (r, s)

=> Verifying

$w = s^{-1} \bmod q$

$t = [H(M) * w] \bmod q$

$u = rw \bmod q$

$v = [(g^t y^u) \bmod p] \bmod q$

Test -

$$\boxed{v = r}$$

SSL –

Provides security to the data that is transferred b/w web browser & server. It encrypts link b/w webserver & web browser. SSL is a method for providing security for web based applications. It is designed to make use of TCP to provide a reliable end-to-end secure service.

⇒ SSL is not a single protocol but rather two layers of protocols –

(i) One layer makes use of TCP directly i.e. SSL Record Protocol. and it provides basic security service to various higher layer protocols.

An independent protocol that makes use of record protocol is HTTP.

(ii) Another three higher level protocols that make use of record protocol are part of the SSL stack. They are used in the management of SSL exchanges.

a) Handshake protocol

b) Change Cipher spec protocol

c) Alert protocol.

<table>
<tr><td>SSL Handshake Protocol</td><td>SSL Change Cipher Spec Protocol</td><td>SSL Alert Protocol</td><td>HTTP</td></tr>
<tr><td colspan="4">SSL Record Protocol</td></tr>
<tr><td colspan="4">TCP</td></tr>
<tr><td colspan="4">IP</td></tr>
</table>

SSL Protocol Stack

(i) SSL Record Protocol –

Provides two services –

a) Confidentiality – using conventional encryption

b) Message Integrity – Using a Message Authentication Code (MAC).

In SSL record protocol

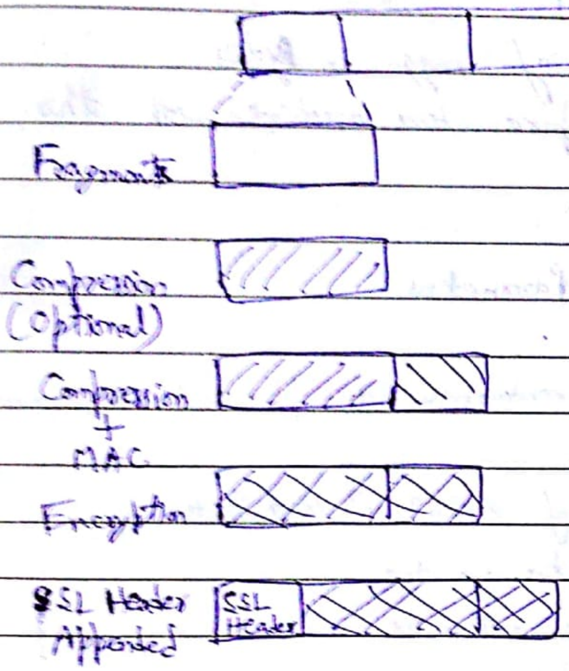
a) Application message to be transmitted gets fragment into manageable blocks. $2^{14} = 16,384$ bytes or less.

b) These blocks then optionally compressed which must be lossless

c) MAC is computed over the compressed data using a shared key. & then appended to compressed Block.

d) The compressed message plus MAC are then encrypted using symmetric encryption.

e) At last SSL header is appended to data.



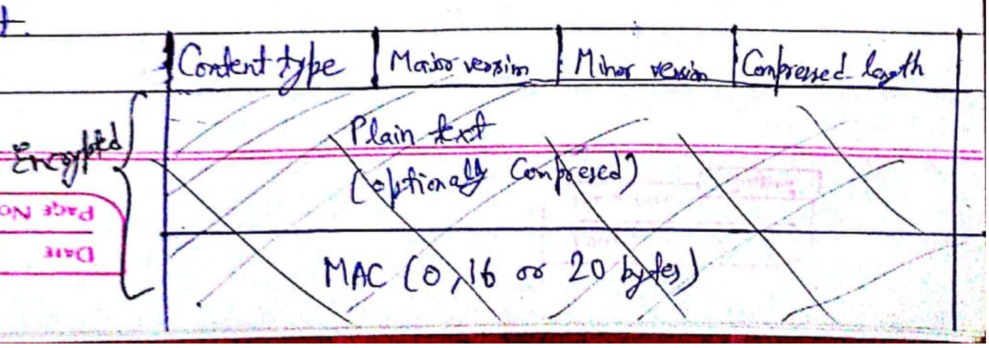
The Header consists of the following fields—

(i) Content type (8 bits) – Higher layer protocol used to process the enclosed fragment.

(ii) Major Version (8 bits) – Indicates major version of SSL in use
eg → For SSL v3, value is 3.

(iii) Minor version (8 bits) – Indicates minor version in use.
eg – For SSL v3, value is 0.

(iv) Compressed length (16 bits) – The length in bytes of the compressed fragment.



(ii) Handshake Protocol

Most complex part of SSL and allows the server & client to authenticate each other and to negotiate an encryption and MAC algo & cryptographic keys to be used to protect data sent in SSL record.

- ~~Et~~ This protocol is used before any application data is sent.
- It consists of series of messages exchanged by client & server.
- Each message has three fields -
  - a) Type (1 Byte) Indicates one of the 10 messages
    eg → 'Hello-request'
  - b) Length (3 Bytes) - Length of msg in Bytes
  - c) Content (≥ 0 byte): The parameters associated with this message.

Message Type	Parameters
① hello_request	NULL .
② client_hello	version, random, session id, cipher suite, compression method
③ server_hello	"
④ certificate	chain of X.509 v3 certificates
⑤ server-key-exchange	parameters, signature
⑥ certificate-request	type, authorities
⑦ server-done	NULL
⑧ certificate verify	signature
⑨ client-key-exchange	parameters, signature
⑩ finished	hash value

It has four phases-

1
1 Byte

(iii) Change Cipher Spec Protocol-

- Consist of single message which consists of a single byte with value 1.
- It is used to cause the pending state to be copied into the current state which updates the cipher suite to be used on this connection.

(iv) Alert Protocol-

- Used to convey SSL related alerts to the peer entity.
- It consist of two bytes the first of which takes the values 1 (warning) or 2 (fatal).
- If the level is fatal SSL immediately terminates the connection.
- The second byte contains a code that indicates the specific alert

Level	Alert
1 byte	1 byte