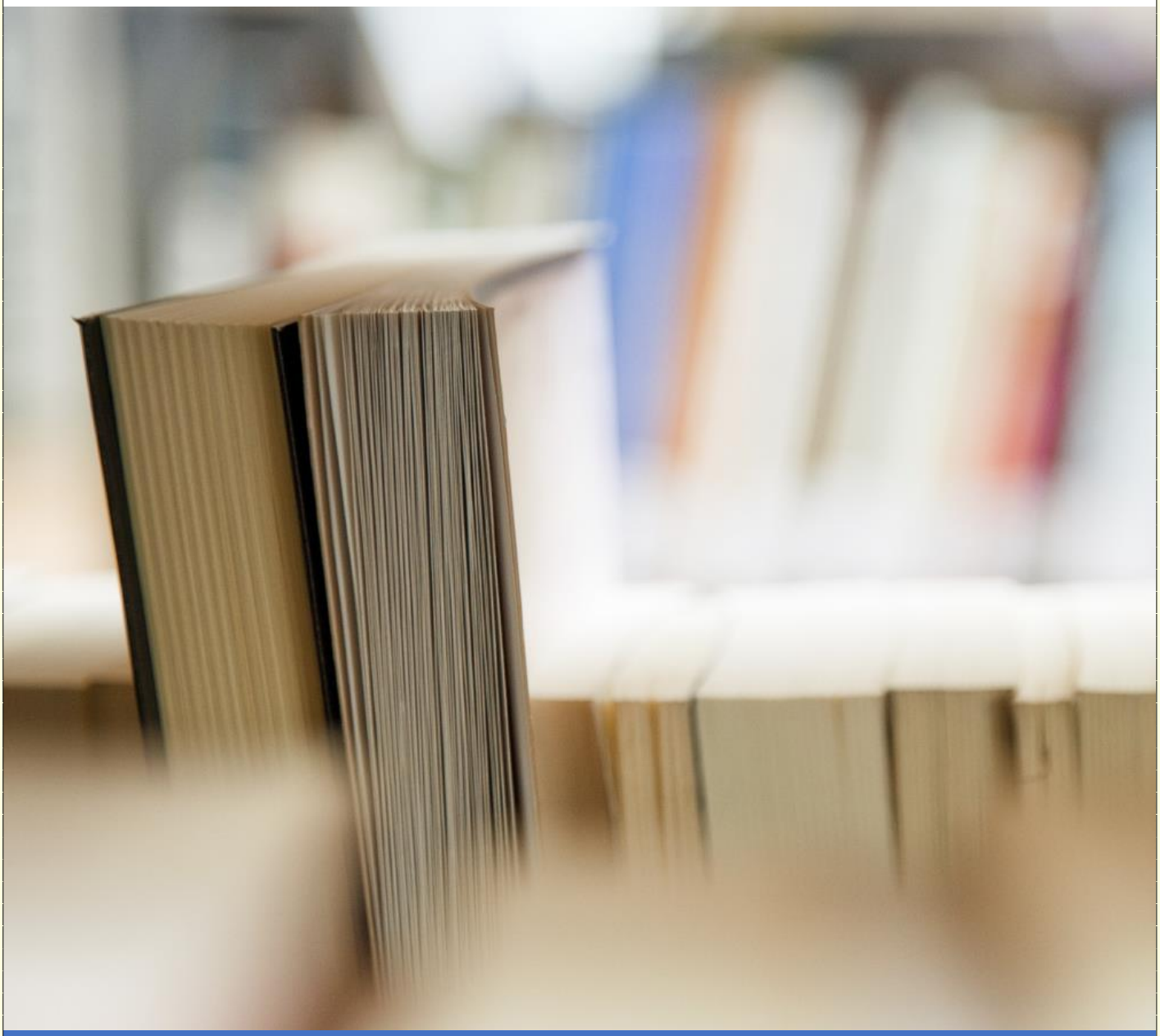


Learn Cryptography in Just 3 Hrs.

Encryption, Hashing, SSL/TLS, PKI

HadoopExam Learning Resources

Contents

About book

Cryptography, SSL/TLS, HTTPS, Public Key, Private Key, Digital certificates, PKI (Public Key Infrastructure is a very confusing subject and ideally if you are a developer you need to know basics of its workings and this book is covering that. You don't need to hours to understand this concept this is the book we have created for you and make sure that your fundamental understanding of SSL/TLS certificate, public-private key, PKI, Cryptography, Digital certificate all are covered in just few hours and you should not spend many days to understand the same. In fact, to validate your understanding this book cover 45+ interview questions. If you see all the big publisher publishes books with 300 and 400 pages which requires many days to complete a single book and most of the reader reads first 2-3 chapters of the book and leave it for many days and they start again from first chapter. We don't want you to face the similar problem, just read this book in one sitting and also give you fundamental learning in just couple of hours and spend another hour for question and answer-based learning. Anytime, you want to revise this concepts comeback to book finish the same in couple of hours.

Cryptography is the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents. The term is derived from the Greek word kryptos, which means hidden. It is closely associated to encryption, which is the act of scrambling ordinary text into what's known as ciphertext and then back again upon arrival. In addition, cryptography also covers the obfuscation of information in images using techniques such as microdots or merging. Ancient Egyptians were known to use these methods in complex hieroglyphics, and Roman Emperor Julius Caesar is credited with using one of the first modern ciphers.

When transmitting electronic data, the most common use of cryptography is to encrypt and decrypt email and other plain-text messages. The simplest method uses the symmetric or "secret key" system. Here, data is encrypted using a secret key, and then both the encoded message and secret key are sent to the recipient for decryption. The problem? If the message is intercepted, a third party has everything they need to decrypt and read the message. To address this issue, cryptologists devised the asymmetric or "public key" system. In this case, every user has two keys: one public and one private. Senders request the public key of their intended recipient, encrypt the message and send it along. When the message arrives, only the recipient's private key will decode it — meaning theft is of no use without the corresponding private key.

About Audio book

This book is also available in audio book format, so you can listen in your leisure time. This way you don't have to ruin your eyes all the time on the computer, mobile or laptop. Rather just put headphone on your ear and start listening the contents. Also, this is highly recommended that you regularly listen this book on your car speakers while travelling to office and if you are using public transport then use the earphone to listen the contents.

To get or subscribe the audio book content, please visit http://hadoopexam.com

Provide Feedback

This is a full-length book from http://hadoopexam.com and we love the feedback so that we can improve the quality of the book. Please send your feedback on hadoopexam@gmail.com or

admin@hadoopexam.com and new content you want us to add then also please drop an email so our team will keep the same thing in mind while preparing for the next edition or updating this book edition. We love and appreciate your feedback for sure.

Restrictions

Copyright© Material

Publication Information

First Version Published: Oct 2022

Edition : 1.0

Piracy

We highly discourage the piracy of copyright material especially it happened online on the internet. Piracy causes the damages to all first of all it damages yourself by not honestly using the correct material, generally pirated material is edited and wrong information is presented which can make big damage as part of your learning process. As well as when you become author and honestly write similar material, piracy will damage your material as well. Hence, don't encourage piracy. If piracy is reduced cost of material will automatically decreases. It also makes damages to author, publisher, dealer and distributors. If you come across any illegal copies of this works in any form on the Internet, then please share the detail with the URL, location or website name immediately on email id hadoopexam@gmail.com we really appreciate your help in protecting author's hard work and also help in reducing the cost of material.

Author/Trainer required

Corporate Trainer: We have many requirements, where our corporate partners need their team to be trained on particular skill sets. If you are already providing corporate trainings for any skills set, then please become our onsite training partner and fill in the form mentioned above and our respective team will contact you soon. You will get very good revenue for sure. However, what we want, you must be able to train our corporate partner resources. What matters to us? Your proficiency in a particular domain/skill and good oral communication skills. You must be able to accessible to learners as well.

Online Trainer: If you are a working professional and master or proficient in any particular skills and feel that, you are capable of giving online virtual trainings e.g. 2 hrs. a day until course contents are completed. Please fill in above form and our respective team will contact you or send an email at admin@hadoopexam.com . You will get a very good revenue share for sure. What matters to us? Your proficiency in a particular domain/skill and good oral communication skills. It will certainly not impact your daily work.

Self-Paced Trainings: Ok, you want to work as per your comfortable time and at the same time sharpen your skills. You can consider this option. You can create self-paced trainings on particular domain/skills. Please fill in above form to connect with us as soon as possible. Before somebody else connect with us for the same skill set. Your commitment is very important for us. We respect your work and we will not sell your work in just $10 to acquire more resources. As we know, it takes a good amount of time and you will provide quality material, so we charge reasonable on that so, you will feel motivated with your work and effort. We respect you and your skill.

Certification Material: You may be already certified professional or preparing for particular certification in a specific domain/skill. So why not use this to make money as well as sharing your effort with other learners globally. Please connect with us by filling form or send email at admin@hadoopexam.com and our respective team will contact you soon.

Author: Yes, we are also looking for authors. Who can write books on a particular technology and what you can get certainly a very good revenue sharing and you can bring the same on your resume or linked in profile to show your excellence? Yes, we are not in need of very good oral communication skills, but good writing skill. However, team will also help you to get work done. Author can be more than one for a particular book. However, we wanted you to be in long relationship. So that you don't just write a single e book, but can create an entire series for a particular domain or skill. Good royalty for sure...

Chapter-1: Security Fundamental Concepts

Data security is the most critical part in Information Technology solution and you don't find any organization, which uses IT solution does not have data security. However, the security strength varies organization or organization as well as application to application.

There are three different stages your data can have.

1. **Data in Transit**: It means data moves from one point to another point or location. This can be sending an email, your WhatsApp messages, fetching data from Web Server (i.e. Tomcat, Apache etc.) to your client agent (Usually Browser). If data travel through internet, then this is highly vulnerable, if proper security is not in place. Data in transit can be read by any middleman and misused and we need proper protection for the same.
2. **Data at Rest**: This data is not moving from one location to another location or networks. However, there are stored on either your Hard Disk, in Databases (i.e. MySQL, Oracle, Cassandra). This information or data also needs to be protected.
3. **Data in Use:** Data in use accessed and consumed by applications or employees or users. This can be read, processed or updated. Since, this data is directly accessible to end user it is highly vulnerable.

To protect your data various mechanism followed with below one or all together.

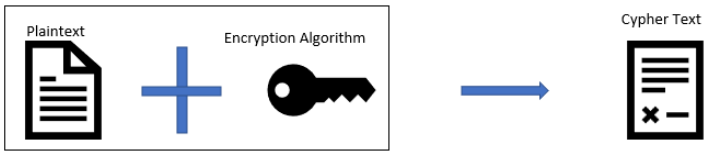
- Data Encryption
- Authentication
- Data Access permissions

Plaintext and Ciphertext

Plaintext refers to anything which human can easily read and interpret its meaning. Like your username and password are below considered plaintext

UserName	admin@hadoopexam.com
Passoword	Pass@12344

Ciphertext is any text which is encrypted. Cyphertext are randomized letter and numbers which human and even machine cannot make sense until it is decrypted or converted back to plaintext. To create a cyphertext you need encryption algorithm. If you process your plaintext with encryption algorithm a ciphertext is generated.



Encryption

Encryption is a method of converting your plaintext into cyphertext, and only the person having relevant decryption keys can generate plaintext from ciphertext.

Encryption Key

Encryption key is a set of mathematical rules and value that both the sender and the receiver know. In case encryption key is compromised then your ciphertext is no more secure. Hence, keeping encryption key safe is critical in overall encryption process.

Decryption

Decryption is a reverse process of encryption. As discussed previously ciphertext can be converted into plaintext using decryption key. Please note that decryption key **cannot** be same as encryption key, it depends what type of encryption algorithm you are using.

Data Integrity

Data Integrity means original data is not altered by anybody whether its in transit or at rest.

Encryption Algorithm

Encryption algorithms are a mathematical formula which, with the help of a key. Changes the plaintext into ciphertext. Common example of encryption algorithms

- AES
- DES and Triple DES
- RSA

Chapter-2: Cryptography

Cryptography is a study of securing communications from outside observers. There are encryption algorithms which takes the plaintext and generate ciphertext, and without decryption nobody can understand it or find the original text. To decrypt the ciphertext you need a key (either same or different) depend on the algorithm used. There are following four objectives of the Cryptography.

1. **Confidentiality**: Only intended recipient can decrypt the message and read its contents.
2. **Non-repudiation**: Sender of the message cannot deny he has not sent the message.
3. **Integrity**: Information or data in the messages are not altered.
4. **Authenticity**: Both sender and received can verify each other's identity.

Most of the ciphers and algorithms used in early days of the cryptography have been deciphered, but it would require years and sometimes decades to decipher of just one message. Hence, there is always advancement going on this field. There are following types of cryptography's

- Secret Key Cryptography (Symmetric Cryptography)
- Public Key Cryptography (Asymmetric Cryptography)
- Hash Functions

Cipher Algorithm

A Cypher algorithm is a mathematical formula, which can be used to obscure the value and content of data. The most important part of algorithms is to use a key(encryption or decryption key) as part of the mathematical formula.

You can use Key to encrypt the data and either same of different (sometime referred as complementary key) is needed to decrypt the data back to a useful form.

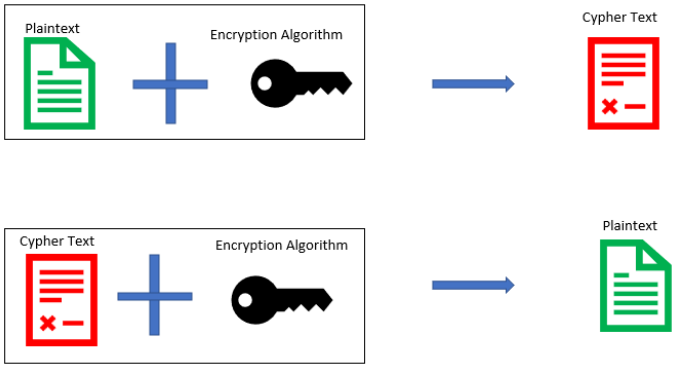
Encryption Key Size

Many Ciphers algorithm increase their protection by increasing the size of the keys they use. If you have key with larger size more computing time and power is needed to encrypt and decrypt the data. Hence, it is very important to choose a cipher which is a best fit for both data protection and computation cost.

Chapter-3: Symmetric Cryptography & encryption

Symmetric Cryptography is sometime referred as a symmetric key encryption or Secret Key Cryptography. It means for both encryption and decryption you are using the same key. As you can see below, same key is used. Secret key cryptography can be used on both in-transit and at-rest data. But it is commonly used on at-rest data, as sending the secret to the recipient of the message can lead to compromise. Following are the Secret Key algorithm

- AES
- DES



Benefits if Symmetric key

- In this case, to generate a strong key is relatively less expensive.
- The key size is small (as we discussed, key size matters).
- Algorithm is relatively less expensive to encrypt and decrypt the data.

This symmetric encryption is effective since, it does not have much time delay as part of encryption and decryption process.

Authentication & Symmetric Key

If you see symmetric key is also providing authentication as well. Since, both the parties (source & destination) have the same key to encrypt /decrypt the data. Until and unless key is stolen. This key is also referred as a **secret-key**. If anyone else finds this key, which he should not know then both data confidentiality and authentication are affected, hence it is very critical to protect this key.

Drawback of Symmetric Key

The major drawback of sharing this secret key with other party or exchanging the secret key. Because it may be possible while exchanging the secret key somebody steal the key. To solve this problem, IT industry use the mechanism to encrypt the secret key itself using a different key (It simple don't get confused).

And to decrypt that “encrypted-secret-key” another party needs respective decryption key. This may lead to a never-ending situation. But there is a solution to this problem and that is done using asymmetric key cryptography. Which we will discuss in later chapters.

Bulk Cyphers

As name suggest a bulk cipher is a symmetric encryption algorithm which is used for encrypting and decrypting of large amounts of data. There are two different types of bulk cyphers are possible.

1. *Stream Cipher*

This cipher works on entire data but one byte at a time. RC4 (128 bit) is the most common stream cipher.

2. *Block Cipher*

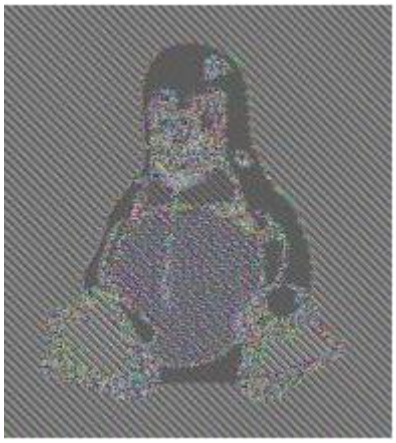
As name suggests it operates on data in groups (blocks) of bytes. However, stream ciphers perform better than block ciphers. But if you consider block cipher then it provides a better security then stream cipher. If you have heard the below ciphers, they are block ciphers

- DES (56-bit)
- TDES (Triple Data Encryption Standards – 168 bit)
- AES (Advanced Encryption standard)

Both DES and TDES operate on block of 8 bytes at a time. And AES operates on blocks of 16 bytes (128-bit) at a time and AES is much popular. Blocks ciphers are designed to encrypt a single fixed size chunk of data, and if your text size does not exactly fit in a block. Block Cipher modes of operations can solve this problem. A block cipher mode of operation defines how the different blocks of a multi-block plaintext should be encrypted and decrypted. There are two block cipher mode operations are possible ECB and CBC.

ECB (Electronic Block Code) Mode

ECB mode is the simple one and it treats each block separately for the multi-block plaintexts. As you can see in the below image (taken from Wikipedia) which is encrypted using ECB, you can still interpret it’s a penguin.



Using ECB mode, encrypting identical plaintext blocks produces identical ciphertext blocks. In this case, each pixel is an independent block of plaintext containing the colour of that pixel. Since, in this image, many of the pixels have the same colour, those identical blocks encrypt to identical ciphertext blocks. As a result, Tux is still visible in the ciphertext. While ECB mode is faster, easier, and more parallelizable to implement, it leaks data about the underlying message being encrypted.

Simply by observing the ciphertexts (which are public information), an eavesdropper can identify identical blocks and make guesses about the original plaintext.

Cipher Block Chaining

Block ciphers can operate in cipher block chaining (CBC) mode. CBC mode means that each block depends on the proper encryption of the block before it. This interdependence ensures that a change to any of the plaintext bits will cause the final encrypted block to change in a way that cannot be predicted or counteracted without knowing the key to the block cipher.

The main feature of CBC mode is that it requires Initial chaining vector, that helps preventing tow identical text sequences from producing the same ciphertext when encrypted.

Encryption usually works by taking a number of text blocks from entire texts and then apply a key on this to produce a cipher text. The typical size you can have is 128 bit or 256 bit. Assume for same text as below, if it generates the same cipher text then intruder can guess your text (by doing some mapping method), which is not good as shown below.

This can be very-very risky and at least computer can easily find the original text without encryption key. And this is known as Electronic Code Block. As you can see above it is always generating same cipher text (xcbjbexx) for the plaintext (HdoopExam). DES is the algorithm which is a 56-bit version and easily crackable.

Adding Salt

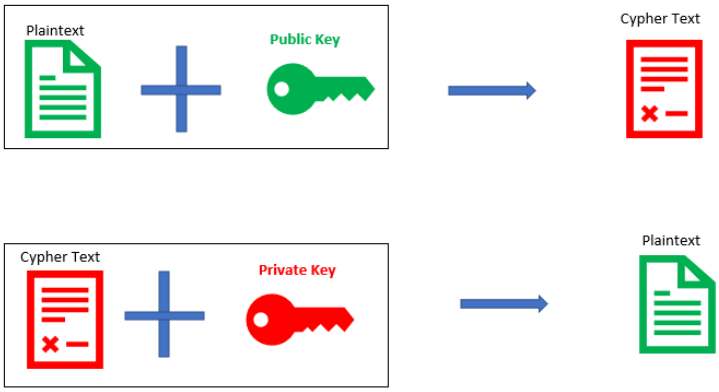
To overcome this problem, where we always get the same cipher text for plain text. We can add a salt and resulted cipher text is always changed. This is usually done in the shared-key encryption (also, known as symmetric encryption) and in hashing (which is known as one-way encryption). And that method which is usually used are CPC (Cipher Block Chaining), where you can start with a rando, seed, known as an Initialization Vector (IV). Which is used in the first block. And next the output from the first block is then used to chain into the next block by Exclusive-OR (XOR) the output of the first block with the output of second block, and it goes on for rest of the blocks.

CBC is an advancement on ECB. You can think here that parallel encryption is not possible with CBC since, every encryption is depended on the previous cipher. A major advantage of CBC mode is that, while encryption must be performed sequentially, decryption can be parallelized.

The recipient of the final encrypted data needs to original ICV (Initial Chaining Vector or IV). In most of the case there is no need for ICV to be secret, but it should never be re-used with the same key. The size of the ICV is the same as the cipher block size (8 bytes of DEST and 16 bytes for AES).

Chapter-4: Public Key Cryptography

Public Key Cryptography is also known as asymmetric encryption. It uses two keys. One key is used for encryption, while the other key can be used decrypt the message. If key-1 is used for encryption then same Key-1 can not be used to decrypt the data rather other key i.e., Key-2 should be used for decryption.



In this scenario, you have to always keep one key as a private and that key is called private key. And another key is called public key and known to everybody or available as a public. Not there is a mathematical relation between these two keys is such that the private key cannot be derived from the public key, however public key can be derived from the private key. This is very important that private key is not distributed and remain with the owner only. The public key can be given to anybody without any security issue. Following is the algorithm which is used for public key cryptography

- Diffie-Hellman (DH)
- DSS
- ECC

The most commonly used implementations of public key cryptography are based on algorithm created by RSA data security.

Public key cryptography involves a pair of keys known as a public key and a private key (a public key pair), which are associated with an entity that needs to authenticate its identity electronically or to sign or encrypt data. Each public key is published and the corresponding private key is kept secret. Data that is encrypted with the public key can be decrypted only with the corresponding private key.

RSA public key pairs can be any size, but they are usually

- 1024 bytes
- 2048 bytes

If you compare symmetric-key encryption with public-key encryption, you will find that public-key encryption requires more calculations. Therefore, public-key encryption is not always appropriate for large amounts of data. However, it is possible to use public-key encryption to send a symmetric key, which you can then use to encrypt additional data.

Data encrypted with your private key can be decrypted only with your public key. However, this is not a used for encrypting sensitive or classified data because it means that anyone (which usually,

anybody can have it) with your public key, which is by definition published, could decrypt the sensitive data. Despite this, private-key encryption is useful in Digital signature, which we will discuss later, because it enables you to use your private key to sign data with your digital signature, anyone with your public key can be assured that only you sent the data.

Features of Public Key

Public authentication has following critical things

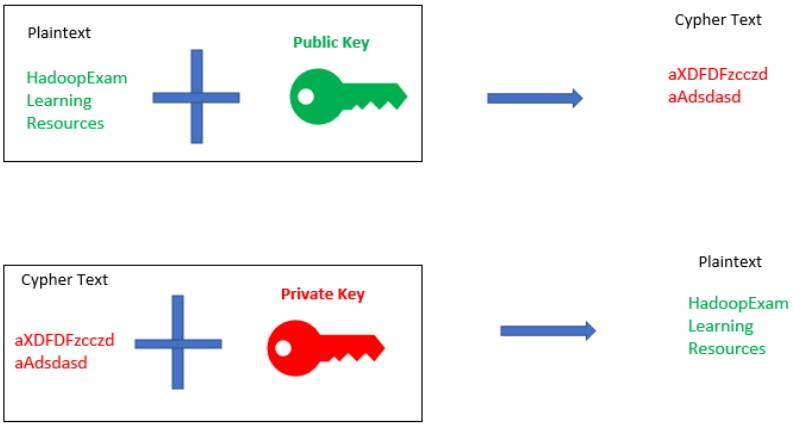
- Using public key, you can protect your communication over the wire usually public domain like Internet.
- Entire security depends, how you protect your private key.

One of the very common use cases of public key encryption is exchanging of the symmetric key. For example, if you (Amit) want to exchange a symmetric key with Rakesh, you can generate a public key pair and publicly broadcast your public key to Rakesh. Suppose somebody else get the public key, it does not affect your security. On receipt, Rakesh could then generate a symmetric key, encrypt that key using your public key, and send the encrypted result back to you. You can then decrypt that result with your private key to reveal the symmetric key that Rakesh generated. Rakesh and you can then begin to use the symmetric key to exchange any other data.

Important: A public key is a cryptographic key (Asymmetric) that can be distributed to the public and does not require secure storage. Messages encrypted by the public key can only be decrypted by the corresponding private key.

As you are aware Asymmetric key is slow and not good for bulk data. Hence, we have to use symmetric key to encrypt the bulk volume of data. Asymmetric encryption is good for following two things

1. Digital Signature
2. Transferring Symmetric key



Important conclusion:

1. Both public and private keys are not identical.
2. Private key is exclusively kept with owner and public key is publicly available.
3. Both public key and private key are mathematically related.
4. To sign only public key shared with Certificate Authority.
5. Public key is available on X.509 certificate.

6. X.509 is certificate format and you(owner) embed that public key in it.
7. X.509 is sent to CA (Certificate Authority) to sign your public key.
8. After signing, you (owner) can always be verified that, public key in certificate is owned by him
9. X.509 certificate is used for Identity and Authentication.
10. Public & Private key combination can be used for Digital Signature as well.

Public Key applications

Asymmetric or public key encryption is used in following application.

- SSH algorithms
- SSL/TLS
- S/MIME encrypted email
- Code signing
- Bitcoin/Blockchain
- Signal private messenger
- Digital signatures

Public Key Algorithms

There are below four public key algorithms in use today, and more can be developed.

- Rivest-Shamir-Adleman (RSA)
- Elliptic Curve Digital Signature Algorithm (ECDSA)
- Digital Signature Algorithm (DSA)
- Diffie-Hellman key agreement protocol

RSA is one of the most popular algorithms, ECC is a newer algorithm that offers shorter keys that achieve comparable strengths were compared with longer RSA keys. DSA implements the Digital Signature Standard (DSS) published by the National Institute of Standards and Technology (NIST) and is used for digital signatures only.

AES 256 Algorithm

The AES 256 certificate is an algorithm and the current encryption standard. The previous standard was AES 128. AES 256 keeps track of vulnerabilities and when the encryption has been breached, a higher standard of encryption will be implemented. The higher the standard encryption, the better cryptic the public/private key pair is. An AES 256 certificate is a long length key that causes brute force attacks by would-be credential thieves virtually impossible.

DIFFIE HELLMAN Algorithm

Diffie Hellman, also known as exponential key exchange, is a method of encryption that uses numbers raised to specific powers that produce decryption keys on the basis of components that are never directly shared, making it overwhelming for potential threats to penetrate. The algorithm creates a mathematically complex encryption that is shared between two parties over a secret communication over a public network so that they can allow an exchange of a private encryption key.

RSA Key Exchange

RSA, named after its inventors Ron Rivest, Adi Shamir, and Leonard Aldeman, is much like the Diffie Hellman algorithm and factors large integers that are the product of two large prime numbers. RSA key exchange uses public and private keys, while the public key can be shared with everyone, the private key must be kept secret. However, in RSA cryptography either of the public or private key can be used to encrypt a message while the other is used to decrypt.

DSA

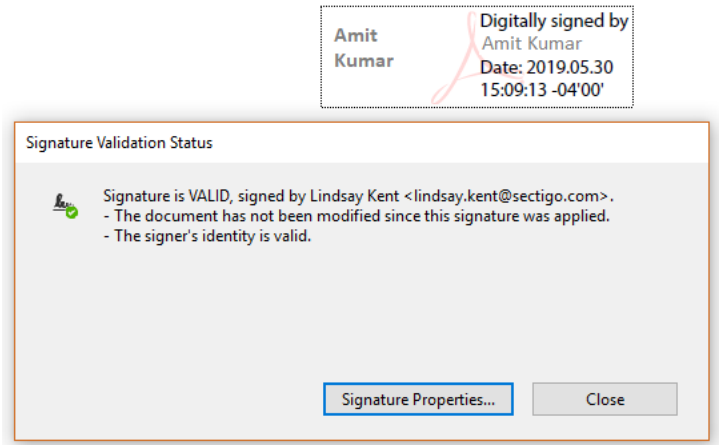
The DSA, or digital signature algorithm, is used to create digital signatures. DSA was created in 1991 by the National Institute of Standards and Technology and is the standard for government agencies.

The pitfall of the DSA algorithm is that it can only do digital signatures and not public key encryption. However, the advantage lies within the algorithms speed of producing a digital signature.

Where all places you use PKI

It is very important to understand where the PKI is used, before we understand what is the PKI. Below are the few examples which will clarify your concept

- **HTTPS (TLS protocol on the HTTP protocol):** If you need secure communication between server and client browser. You have to install SSL certificate on the server. SSL Certificate works as an identity card for that website. When you purchase an SSL certificate, you get the public and private keys. The private key is stored secretly on the web server, and the server uses it to prove its legitimacy. And when you use SSL certificate it
    - protects the integrity of the message.
    - protects the information from man-in-the-middle attacks.
    - authenticates both the communicating parties (browser and server).
- **SSH**: Secure Shell (SSH) is a cryptographic network protocol that provides secure network services in between hosts and users over an unsecured network. It's used to facilitate a secure remote login from one computer to another. SSH also has the public key infrastructure (PKI) at its heart.
- **Secure Messaging**: Whether it's WhatsApp, iMessage, Facebook Messenger, or other such messaging services, we all use communications services or apps. Many of these services are encrypted using PKI and protect against attempts of data interception and tampering.
- **Document Signing**: Like physical signatures to authenticate physical documents, there's a need to sign documents digitally. For one, it helps the recipient ensure that the message is coming from a verified entity; it also allows them to ensure it's not been tampered with. This is also done by PKI using document signing certificates. As you can see on many pdf files are issued by your employer you see below signature



- **Code/Application Signing**: You download many applications on computers and mobile devices most of the time. Company or individuals who created these programs or software. But what's the guarantee that the software you're installing is from the company/individual that you think it's from? Can't someone just change the name of the file and disguise malicious software as the software? Yes. And that's a very real concern for everyone. Well, that's where code signing certificates come in. On Windows you see message like

A code signing certificate authenticates the identity of the developer or publisher and the integrity of the file. This enables browsers to verify that the software itself hasn't been altered in any way using public key encryption. It does this by applying a digital signature and a one-way hash.

- **VPN Authentication**: Certificates can be used to authenticate users for VPN access. Since VPNs can grant access to critical information, certificates are a preferred method of authentication over passwords. Usually the Root/Intermediate CA is stored on the Firewall and once the user is authenticated, a secure tunnel is created to access the network the user is trying to access.

Private Key

Private Keys are used by the recipient to decrypt a message that is encrypted using a public key. Since the message is encrypted using a given public key, it can only be decrypted by the matching private key. This establishes the ownership of the private and public key, ensuring the message is only read by the approved parties.

Public Key Infrastructure

The ultimate goal of a PKI is identity and access management for a secure network.

PKI is a system that is used for encryption and authentication purposes and is not a specific product/software. Therefore, it needs some material basis to implement encryption. And that comes in variety of technologies (software and hardware), entities (Certificate Authority), processes, policies and procedures.

This PKI system facilitates and governs encryption and everything that makes it possible – including basics of public keys and digital certificates to the management of them and understanding the CAs which issues and revoke the certificate as well. Hence, we can say that PKI is made up of following three things

- Public-Private Key Pairs
- X.509 Digital Certificate
- Certificate Issuing Authority

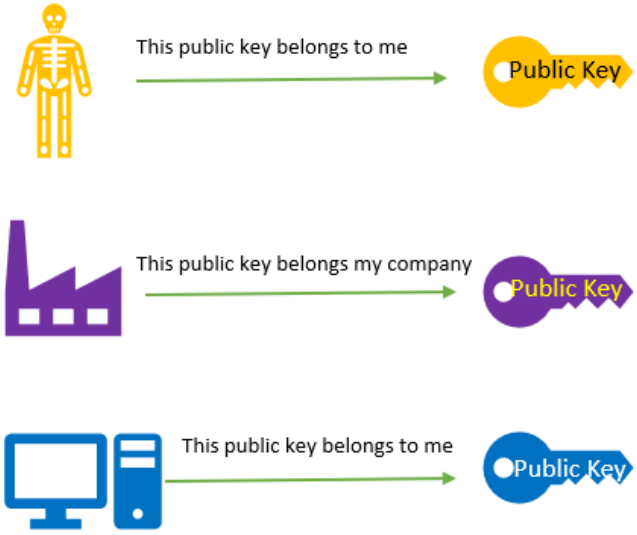
The public Key Infrastructure (PKI) is the set of hardware, software, policies, processes and procedures which is required to create, manage, distribute, use, store and revoke digital certificates and public-key.

You can say that PKI is foundation so that digital signatures and encryption can be used across large user base. PKIs deliver the elements essential for a secure and trusted business environment for e-commerce and the growing Internet of Things (IoT).

You can use PKI to establish the identity of the machine, person, device and software service. For example, to bind a public key with their associated user (who owns private key), PKI use the digital certificate. Digital certificates are the credentials that facilitate the verification of identities between users in a transaction. In security eco-system Digital certificates are used to identify the users to whom encrypted data is sent, or to verify the identity of the signer of information, protecting the authenticity and integrity of the certificate is imperative to maintain the trustworthiness of the system. Hence, CA issues the certificate. PKI assigns identities to keys so that recipients can accurately verify the owners.

Overall purpose of the PKI is to facilitate the secure data transfer on network, because simple password-based authentication is not enough and helps in getting/implementing rigorous proof is required to confirm the identity of the parties involved in the communication and to validate the information being transferred.

PKI is part of Cryptography and provides an arrangement which binds public keys with respective identities of entities like person or organization.



However, there is process of registration and issuance of certificate by a certificate authority (CA) to binding and confirm who owns this public key. Also, what level of assurance is provided on each public key is also matter.

Certification Authority

You have your public key, but what is the guarantee you that you are the owner of that public key. In general term, how do I make sure that ID Card (public key) you are having is really belongs to you. There should be somebody to confirm that by looking the contents (Student Name, Student Enrolment number) can confirm yes this is a valid ID Card (Public Key) of this student. Here, school will work as CA (Certification Authority) to confirm that it's a valid ID card (public key).

Certificate authorities are the trusted third-party entities that issue and manage digital certificates. They're the most crucial entity in PKI since millions of users — knowingly or unknowingly — are going to rely on them. Before issuing a digital certificate, a CA is supposed to conduct a vetting process to make sure that it issues the certificate to a legitimate entity. Even just a small mistake in the vetting process could result in a mis-issuance and cause a disaster not just in terms of the damage, but also in terms of the overall trust in the system that is PKI.

Therefore, to be a trusted CA, you must fulfil super-stringent criteria formed by an independent body of browsers, operating systems, and mobile devices that's known as the CA/Browser Forum (or CA/B Forum for short). On top of that, you need a multi-million-dollar infrastructure that includes sizeable operational elements, hardware, software, policy frameworks, practice statements, auditing, security infrastructure, and personnel.

The CA generally handles all aspects of the certificate management for a PKI, including the phases of certificate lifecycle management. In a PKI system, the client generates a public-private key pair. The public key and information to be imprinted on the certificate are sent to the CA. The CA then creates a digital certificate consisting of the user's public key and certificate attributes. The certificate is signed by the CA with its private key.

Once the certificate is distributed to the user, they can present the signed certificate and the receiver can trust that it belongs to the client because of the matching public-private key pair.

- **Public CA (Certification Authority)**

  A public CA is a third-party entity that issues certificates for a fee after doing the necessary checks on the organization requesting a certificate.

  The checks by default include domain validation, and Third-party CAs have their own public-private key pairs with which they sign the certificates. Most of the well-known CAs are recognized by servers and clients; therefore, certificates signed by them are immediately validated by the entity initiating a secure connection. Publicly-signed certificates offer a higher level of assurance since they are issued by a recognized CA, and are generally used for securing websites and other endpoints involving direct user interaction.

- **Private CA**

  When an organization creates its own local CA without going for an external one, it's called a private CA. In this case, the certificates are signed with the private key of the organization's root certificate. Private CAs can be created to issue certificates for an organization's internal network where discretion is required, and only a select group of users are involved.Private CAs can be created to issue certificates for an organization's internal network where discretion is required, and only a select group of users are involved.

Root Certification Authority

A Root CA is a trusted CA that is entitled to verify the identity of a person and signs the root certificate that is distributed to a user. The certificate is considered valid because it has been verified and signed by a trusted root CA.

Intermediate Certification Authority

An Intermediate CA is also a trusted CA, and is used as a chain between the root CA and the client certificate that the user enrols for. Since the root CA has signed and trusts the intermediate CA, certificates that are generated from the intermediate CA are trusted as if they were signed by the root CA.

Registration Authority

This authority performs the following functionality,

- Identification and authentication of certificate applicants

- The approval or rejection of certificate applications
- Initiating certificate revocations or suspensions under certain circumstances
- Processing subscriber requests to revoke or suspend their certificates
- Approving or rejecting requests by subscribers to renew or re-key their certificates

Important: Please note that RA does not sign or issue the certificate. Most of the time CA assign one or more of the above tasks by CA (Certification Authority).

- RAs do not have the signing authority of a CA and only manage the vetting and provisioning of certificates.
- Most commercial PKI solutions offer a stand-alone RA component.

A registration authority (RA) is an authority in a network which verifies that end users' requests for a digital certificate and tells the CA (Certification Authority) to issue it. RA's is the part of PKI. As we know RA cannot create or issue a certificate because that is only done by CA. Hence, RA works as an intermediary for the CA and perform functionalities mentioned above.

The main purpose of an RA is to ensure that a user or device is allowed to request a digital certificate from a specific website or application. If the request is allowed, the RA forwards the certificate request to the CA, which completes the digital certificate request process.

RA is also known as a subordinate certificate authority. A registration authority can be thought of as a gatekeeper to a certificate authority. In order to be issued a certificate, the requesting user or device must first register with the RA and fulfil the necessary requirements, including identity and authentication checks. This comes in the form of a certificate signing request.

Requests that are successfully registered by the RA are then forwarded to the CA, whose responsibility is to issue an electronic document called a certificate. This certificate is issued to the requesting user or device. The issued certificate can be validated against the CA's public key to ensure that the certificate is indeed valid and that the connection to the remote resource is truste

A successful digital certificate request process happens in the following order:

- A user attempting to access a certificate-backed website requests the certificate from the CA. This request is sent to the web server.
- The web server forwards the certificate request to the RA. The RA ensures the user is allowed to receive a certificate.
- If the RA grants the request, it is passed to the CA, which generates the digital certificate.
- The CA sends the digital certificate directly to the user to complete the process.

Registration Authority (RA), which is authorized by the Certificate Authority to provide digital certificates to users on a case-by-case basis. All of the certificates that are requested, received, and revoked by both the Certificate Authority and the Registration Authority are stored in an encrypted certificate database.

Certificate history and information is also kept on what is called a certificate store, which is usually grounded on a specific computer and acts as a storage space for all memory relevant to the certificate history, including issued certificates and private encryption keys. Google Wallet is a great example of this.

Certification Revocation List

A CRL is a list of certificates that have been revoked by the CA that issued them before they were set to expire. This is a helpful security feature if a device is stolen that contains a certificate. A Base CRL is a large file containing all revoked certificates. This file is published and updated at infrequent intervals. A Delta CRL is a small file containing the certificates that have been revoked since the last base CRL was published. Typically, the Base CRL is updated on a weekly basis, and the Delta CRL is updated on a daily basis. When the base CRL is updated, the list of revoked certificates listed in the Delta CRL are appended to the Base CRL.

Certification Store

A Certificate Store is used to store certificates and can potentially contain certificates from multiple CAs. For example, different Windows certificates are stored in the certificate store and can be viewed using MMC snap-in, while in macOS, certificates are stored in the keychain.

Hardware Security Module (HSM)

A Hardware Security Module isn't a mandatory component of a PKI, but it improves the security of the PKI as a whole when implemented. This device protects and manages digital keys and serves as the groundwork for building a secure enterprise PKI infrastructure. The HSM contributes to managing the complete lifecycle of cryptographic keys, which includes creation, rotation, deletion, auditing, and support for APIs to integrate with various applications.

Certificate Lifecycle

1. **Certificate Enrolment** – An entity submits a request for a certificate to the Certificate Authority (CA). An entity can be a person, a device, or even just a few lines of code.
2. **Certificate Issuance** – The CA needs to validate the identity of the applicant, which is typically done through credentials or by trusting another CA that has already validated the applicant.
3. **Certificate Validation** – Every time the certificate is used to authenticate, the server checks with the CA to confirm that the certificate is still valid and hasn't expired or been revoked.
4. **Certificate Revocation** – Certificates contain an expiration date that's specified when they are first issued, usually for a duration of several years. When that date is reached, the certificate will automatically be considered invalid for any authentication attempt.
5. **Certificate Renewal** – Instead of automatically being shunted to a CRL, some CA's have settings that renew certificates upon expiration date, though typically they re-verify identity. At this time, you can choose whether or not to generate a new key pair – effectively making it a totally new certificate.

Trust Store

A trust store is a list of root certificates (sometimes called trust anchors) that comes pre-installed on a device. It's composed of more than a hundred of the largest and most trusted CAs such as Digicert, Apple, Microsoft, Symantec, Mozilla, Lets Encrypt, and more.

It serves a couple of very important purposes. First, they sign (validate) the identity of the device for other certificate authorities. The root CAs know the public key of the device and can confirm to any third parties.

Certificate authorities rarely sign certificates using the root CA directly. Instead, they put one or more levels of separation between themselves and the client by creating intermediate certificate authorities. Intermediate CAs are functionally identical, but they have less "authority" because they

are responsible for signing fewer certificates. Theoretically, they are just as trustworthy, but in the case that they are compromised, it limits the damage that can be caused.

What is a Microsoft PKI?

Microsoft offers a commonly used PKI called "Active Directory Certificate Services" (ADCS). It was designed to work with the Microsoft environments (AD, NPS, GPO) that historically dominated IT infrastructures.

While some still use AD CS, many organizations are moving away from it due to the limitations that come with being designed for legacy infrastructure. It requires a lot of human resources to deploy and maintain, along with the fact that it requires everything to be on-premise, which can prevent organizations from moving to an all-cloud environment which is where the industry is heading.

Advantages of Public Key Infrastructure

There are multiple advantages of using public key encryption as part of your public key infrastructure.

Authentication

In the age when fraudsters and scamsters are trying every trick in the book to fool users, authentication/validation becomes an out and out necessity. When you're transmitting information through a website, an email, or text messages, making sure that you're communicating to the intended entity is a must. The vetting process conducted by certificate authority and the use of the private and public key; PKI facilitates authentication in an unprecedented, smooth way.

Privacy

One of the essential security elements when it comes to online communication is privacy. After all, nobody wants to disclose their passwords, credit card information, or cute cat photos. By encrypting the data between the sender and the recipient, PKI keeps the original data secure so that only the intended recipient can see the data in its original format.

Data Integrity

When you send sensitive information online, it's imperative for both the parties to have the recipient receive the data in the unaltered form. Through a technique called "hashing," PKI allows the recipient to check whether the message/document/data has remained in the same form or not.

Non-Repudiation

PKI provides a mechanism to digitally sign online transactions (files, emails, documents, etc.), the way we physically sign documents and stuff. This way, it acts as proof that the person who signed it is the originator of the data. And, therefore, it also makes it impossible for the sender to deny that he/she was the one who signed and sent it. This is called "non-repudiation."

Disadvantages of Public Key Infrastructure

While there are many advantages of public key infrastructure and the encryption it provides, we'll admit — it's not perfect. There are some specific disadvantages as well that are worth noting:

Speed

PKI is an extremely secure process that delivers what it's supposed to. A large part of the credit goes to the key pair and super-complex mathematical algorithms. However, this complexity brings computational overhead when it comes to encrypting data in large volumes. As a result, it slows down the data transfer process to a minor degree, 5 milliseconds (ms) to be precise. In terms of CPU usage, the difference between encrypted connections and unencrypted connections is found to be 2%. Although this difference is quite small, you could consider migrating to HTTP/2 to speed up the data transfer process.

Private Key Compromise

The mathematics behind PKI is supposed to be so strong that even super-computers (let alone hackers) aren't able to crack it within a practical time. However, the entire PKI security doesn't depend on the unconquerable mathematics; it also depends on the security of the private key as it can decrypt the data encrypted by the public key.

Therefore, if the private key gets compromised, a cybercriminal doesn't need to crack the super-complex mathematical algorithms. They can decrypt the data (of the past as well as of the future) with the private key and can also imitate the server to fool clients. This could result in organizational secrets, passwords, financial information, etc. being compromised. In other words, it could cause unprecedented disasters. This is a significant cause for concern while using PKI.

Chapter-5: Hash Function

Hash functions are irreversible, one-way functions which protect the data, at the cost of not being able to recover the original message. Using hashing you can transform a given string into a fixed length string. A good hashing algorithm will produce unique outputs for each input given. The only way to crack a hash is by trying every input possible, until you get the exact same hash. A hash can be used for hashing data (such as passwords) and in certificates. Following are the some examples of hashing algorithms

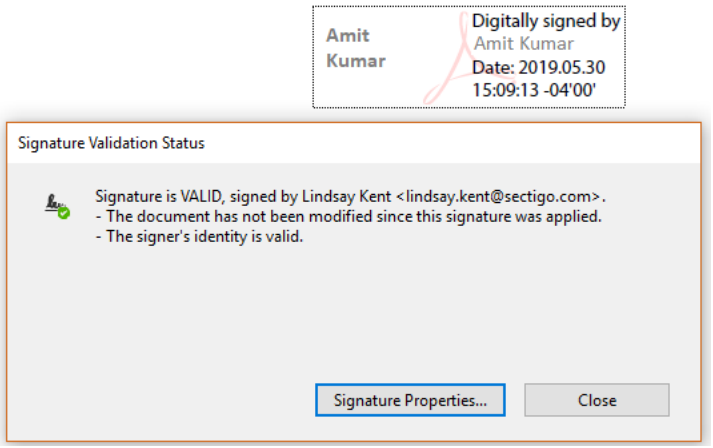
- MD5
- SHA-1
- SHA-2 (It is a family and include SHA-224, SHA-256, SHA-384, and SHA-512)
- SHA-3
- Whirlpool

Chapter-6: Digital Signature

Individuals and organizations are using digital documents (i.e., pdfs) instead of paper documents to conduct day-to-day transactions. By reducing dependency on paper documents, we are protecting the environment and saving the planet's resources. Digital signatures support this change by providing assurances about the validity and authenticity of a digital document.

Digital signature

A digital signature is an electronic, encrypted, stamp of authentication on digital information such as email messages, macros, or electronic documents. A signature confirms that the information originated from the signer and has not been altered.



Signing certificate and certificate authority

To create a digital signature, you need a signing certificate, which proves identity. When you send a digitally-signed macro or document, you also send your certificate and public key. Certificates are issued by a certification authority, and like a driver's license, can be revoked. A certificate is usually valid for a year, after which, the signer must renew, or get a new, signing certificate to establish identity.

A certificate authority is an entity similar to a notary public. It issues digital certificates, signs certificates to verify their validity and tracks which certificates have been revoked or have expired.

Digital signature assurances

The following terms and definitions show what assurances are provided by digital signatures.

- **Authenticity:** The signer is confirmed as the signer.
- **Integrity:** The content has not been changed or tampered with since it was digitally signed.
- **Non-repudiation:** Proves to all parties the origin of the signed content. Repudiation refers to the act of a signer denying any association with the signed content.
- **Notarization:** Signatures in Microsoft Word, Microsoft Excel, or Microsoft PowerPoint files, which are time stamped by a secure time-stamp server, under certain circumstances, have the validity of a notarization.

To make these assurances, the content creator must digitally sign the content by using a signature that satisfies the following criteria:

- The digital signature is valid.

- The certificate associated with the digital signature is current (not expired).
- The signing person or organization, known as the publisher, is trusted.
- **Important:** Signed documents, which have a valid time stamp, are considered to have valid signatures, regardless of the age of the signing certificate.
- The certificate associated with the digital signature is issued to the signing publisher by a reputable certificate authority (CA).

A digital signature scheme typically consists of three algorithms:

- A key generation algorithm that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
- A signing algorithm that, given a message and a private key, produces a signature.
- A signature verifying algorithm that, given the message, public key and signature, either accepts or rejects the message's claim to authenticity.

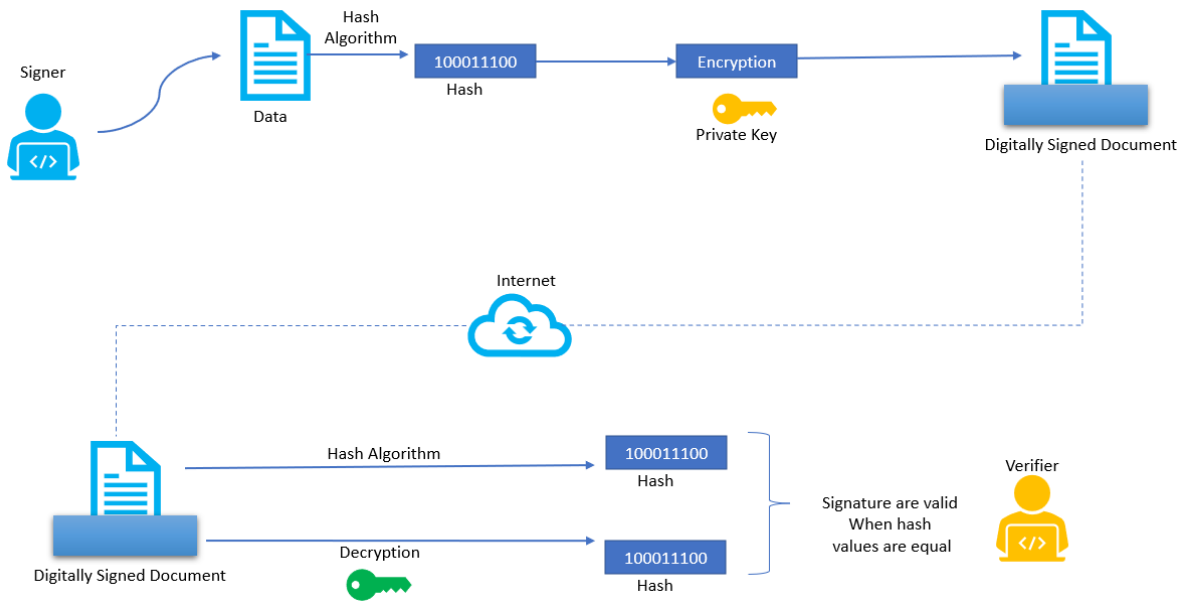
Two main properties are required. First, the authenticity of a signature generated from a fixed message and fixed private key can be verified by using the corresponding public key. Secondly, it should be computationally infeasible to generate a valid signature for a party without knowing that party's private key. A digital signature is an authentication mechanism that enables the creator of the message to attach a code that acts as a signature. The Digital Signature Algorithm (DSA), developed by the National Institute of Standards and Technology, is one of many examples of a signing algorithm.

Class of Digital Certificates

This may vary country to country, and in India. The different types of Digital Signature Certificates are:

- **Class 2:** Here, the identity of a person is verified against a trusted, pre-verified database.
- **Class 3:** This is the highest level where the person needs to present himself or herself in front of a Registration Authority (RA) and prove his/ her identity.

How digital signature works?



Digital signatures, are unique to each signer. Digital signature solution providers, such as DocuSign, follow a specific protocol, called PKI. PKI requires the provider to use a mathematical algorithm to generate two long numbers, called keys. One key is public, and one key is private.

When a signer electronically signs a document, the signature is created using the signer's private key, which is always securely kept by the signer. The mathematical algorithm acts like a cipher, creating data matching the signed document, called a hash, and encrypting that data. The resulting encrypted data is the digital signature. The signature is also marked with the time that the document was signed. If the document changes after signing, the digital signature is invalidated.

To protect the integrity of the signature, PKI requires that the keys be created, conducted, and saved in a secure manner, and often requires the services of a reliable Certificate Authority (CA). Digital signature providers, like DocuSign, meet PKI requirements for safe digital signing.

Java Code Signing: Generate CSR

To request a code signing certificate, you have to provide us a certificate signing request (CSR) generated from the machine you'll use to sign the code. We'll use the information in this file to validate your request and provide the information to anyone downloading your code.

Important: You must generate the CSR from your local machine and not from the web server you're using to host the file.

1. **Windows-only preparation:** If you use Windows, you must complete the following steps before generating your CSR. Run cmd as an administrator. Move to your JDK installation's bin directory:

cd C:\Program Files\java\<jdkversion number>\bin

2. **Create a keystore:** To store your public and private keys (used to sign your code), you must create a keystore. You'll use this repeatedly throughout the code signing certificate request and installation process.

- Create a keystore called codesignstore:

```
keytool -genkey -alias codesigncert -keyalg RSA -keysize 2048 -keystore codesignstore
```

- Complete the information requested from you at the command prompt. There are a few important things to note:
    - It's important that this information is consistent with everything else you plan to use in your request.
    - Note your keystore's password because you must have it to sign your Java code.

3. **Generate a CSR:** Now, use your keystore to create your certificate signing request (which you will use to request the certificate you purchased from GoDaddy).
    1. Generate your CSR:

    keytool -certreq -v -alias codesigncert -file mycsr.pem -keystore codesignstore

    2. Enter your keystore's password and press enter.
    3. Open your CSR. How you do that depends on your OS:

    start notepad "mycsr.pem"

4. Copy your CSR, including the entirety of the lines containing BEGIN NEW CERTIFICATE REQUEST and END NEW CERTIFICATE REQUEST.

Chapter-7: X.509 Certificate

Digital Certificate

You can use digital certificate as a proof of users, servers' authenticity. When you go to https://hadoopexam.com how do you make sure that this page is returned is really coming from hadoopexam.com and not attacked by man in the middle. That is you can find using Digital Certificate X.509 and you can trust is that is signed by CA (Certificate Authority).

X.509 is a standard format for public key certificates, it was first introduced in 1988 with X.500 standards. Following are the common applications of X.509 certificates

- **SSL/TLS and HTTPs**: for authenticated and encrypted web browsing.
- **Signed and encrypted email**: via S/MIME protocol.
- **Code Signing**
- **Client Authentication**
- **Government Issued electronic ID**

X.509 certificates are the perfect mechanism to become a new authentication standard, you can replace password-based authentication using X.509 certificate.

Features of digital certificate

Digital certificate and X.509 Certificate and PKI Certificate all are same thing. And they have following qualities

- You can image its user or entity or company identity like a passport or Aadhar Card (India) or Driving License.
- It contains information about individual user or any entity who owns this.
- If certificate is not self-signed and signed by CA then you can assume they are trusted.
- It contains information that can prove its authenticity
- Using certificate information, you can track back the issue of certificate.
- Each certificate has expiration date.
- Certification can be presented to someone for authenticity.

Public Key Encryption and Certificate

X.509 is based in asymmetric cryptography. Each certificate uses the pair of encryption keys known as the public and private key. The private key is kept by the certificate holder while the public key can be freely distributed.

Certificate can be used as an authentication method for numerous different resources. Wi-Fi is a common application for certificate-based authentication, but certificates can also be applied to VPN and to a web application.

Certificates are issued by trustworthy sources called Certificate Authorities (Cas). A CA is responsible for verifying the identity of the person or device requesting a certificate, as well as ensuring that they are only distributed to approval entities.

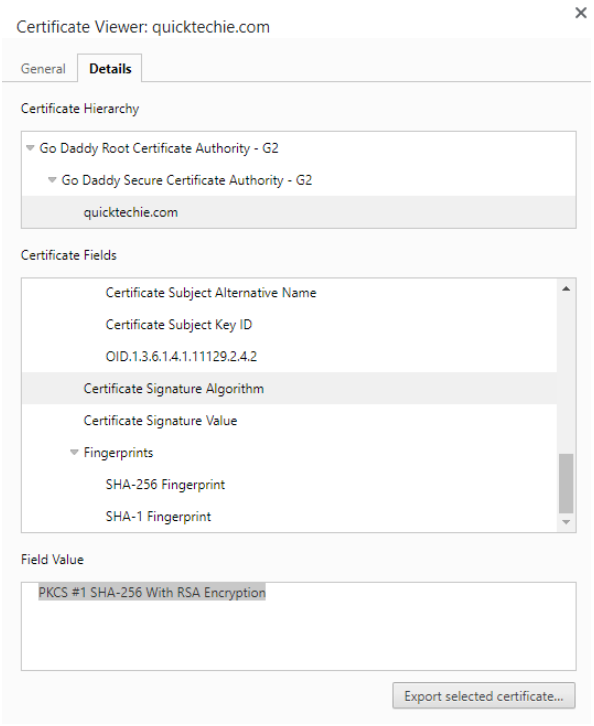
Certificate Authority

As we have been discussing, issuing the digital certificate is responsibility of the Certificate Authority and they do following things.

- Carefully check the methods for certificate recipients
- What type of certificate it is?
- Parameters in the certificate.
- Security and operations procedures

X.509 Certificate Attributes

Each certificate has a number of attributes and fields that provide some information about the certificate user, certificate issuer and cryptographic detail.



- **Subject:** The name of the entity whose public key the certificate identifies. This name uses the X.500 standard, so it is intended to be unique across the Internet. This is the Distinguished Name (DN) of the entity. The name of the user or device the certificate is being issued to.

  CN = quicktechie.com

- **Serial Number:** An identifying number that the CA assigns to each certificate it issues.

  36:35:80:7A:37:59:7E:D3

- **Signature Algorithm:** The private key's algorithm, which is usually RSA 2048.

  PKCS #1 SHA-256 With RSA Encryption

- **Validity:** A date range in which the certificate is considered valid.
- **Issuer:** The issuing CA's name.
- **DNS:** Used to imprint the certificate with the device's information.
- **Other Name:** User principal name. This field is usually used to indicate the user's identity for Wi-Fi connections specifically.
- **RFC822:** An email address associated with the user.

In addition to these required fields, an X.509 certificate might contain one or more extensions that hold information about how the key is to be used (a KeyUsage extension) or how the certificate authority conducts its business (a CertificatePolicies extension).

So you can think of a digital certificate is a binding between named entity (a person or device) and a public key.

Java Tool to for X.509 Certificate

keytool is a key and certificate management utility and can be used to create public/private key pairs and self-signed X.509 v3 certificates, and to manage keystores. Keys and certificates are used to digitally sign your Java applications and applets using jarsigner tool.

A keystore is a protected database that holds keys and certificates. Access to a keystore is guarded by a password (defined at the time the keystore is created, by the person who creates the keystore, and changeable only when providing the current password). In addition, each private key in a keystore can be guarded by its own password.

Using keytool, it is possible to display, import, and export X.509 v1, v2, and v3 certificates stored as files, and to generate new self-signed v3 certificates.

keytool allows users to administer their own public/private key pairs. The keytool command stores the keys and certificates in a keystore.

TLS/SSL and X.509

X.509 certificate can be used in SSL/TLS certificate, which is a basis of the HTTPS (secure protocol for web browsing). An X.509 (also called digital) certificate contains a public key and an identity (a hostname, or an organization, or an individual), and is either signed by a certificate authority or self-signed. When a certificate is signed by a trusted certificate authority, or validated by other means, someone holding that certificate can rely on the public key it contains to establish secure communications with another party, or validate documents digitally signed by the corresponding private key.

Signed X.509 Certificate

As we know that certificate has to be signed by trusted authority to use it in production environment. If certificate is signed by yourself than it is called self-signed certificate. Certificate authorities (CAs) are often well-known commercial organizations or they can be local or internal organizations. Please note that self-signed certificate will not be trusted for public-facing applications. Because of this, self-signed certificate are mainly used to encrypt and authenticate data within an organization's network.

Step-1: Suppose, you generated a certificate by using any tool i.e. Keytool is the one option in Java and there are many other tools available for the same. When you generate a public-private key pair, you should always save private key at secured place and never-ever share, not even with the certificate authority.

Step-2: Share private key at secured place.

Step-3: Public key you can share to public. But why people or recipient believe this public key is a valid and you really own it or it had not come from any intruder. For that you give your generated public key to a certificate authority (i.e., Verisign).

Step-4: Now Verisign will take you public key and encrypt your public key (sign it) using its own (Verisign) private key and your public key is now signed by a trusted authority (Verisign).

Step-5: After that you can use this signed certificate on your website or application.

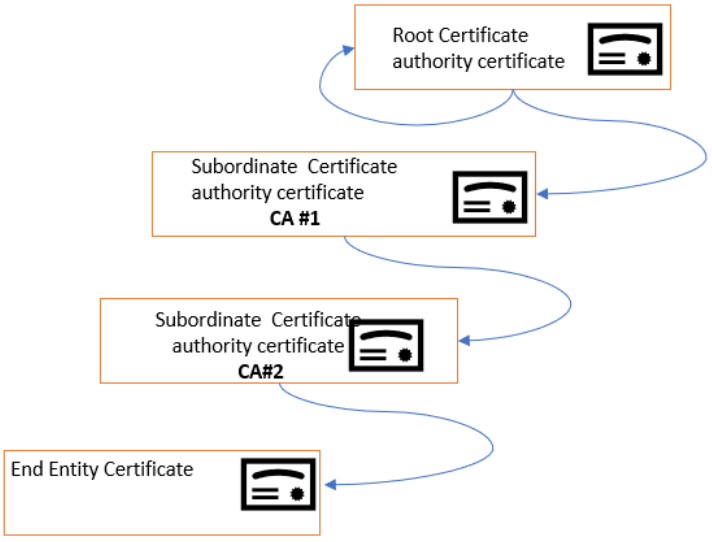
Step-6: Your application first shares this certificate with the recipient before starting any communication with you. Now recipient has to make sure (usually browser in case of HTTP/S) that this certificate or public key is valid and owned by you.

Step-7: Recipient agent usually browser first check who signed your certificate (public key) then it finds the respective CA or trusted authority public key. Since, certificate was signed by private key of CA, it can be decrypted using its private key and then confirm its valid public key and owned by you.

Important: This entire process is only for authentication and no encryption involved yet. Hence, X.509 can be used to confirm users' identity as well, without even having any password. When a certificate authority uses its private key to sign and issue a certificate, it makes the declaration that binds the entity (subject) to its public key.

Certificate Hierarchy

As we have seen above certificate authority sign the certificate using their own private key. Hence, another party can verify in a certificate including its extension. The other party gets the certificate authority's public key from a certificate issued to the certificate authority and does a signature check that might involve the public key from yet another certificate. The chain of verification can be quite long, depending on the certificate hierarchy.



As you can see in above image.

- Representation of a certificate hierarchy containing four entities where the end-entity certificate is issued by subordinate certificate authority (CA #2).
- The certificate of CA #2 is issued by subordinate certificate authority (CA #1).
- The certificate of CA #1 is issued by the root certificate authority.
- The certificate of the root certificate authority is self-issued.
- The chain of signature verification begins with the end-entity certificate.
- The public key of CA #2 is used to verify the signature of the end-entity certificate.
- If the signature is valid, the public key of CA #1 is used to verify the signature of the CA #2 certificate.
- If the signature is valid, the public key of the root certificate is used to verify the signature of the CA #1 certificate.

- Finally, the signature of the root certificate is verified using its own public key.

Since each CA has a certificate of its own, layers of trust get created through CA hierarchies and it means CAs issue certificates for other CAs. Please note that this process is not a circular, because there is a root certificate. Normally, certificates have an issuer and a subject as two separate parties. However, in root certificate both parties are same, and that root certificate is self-signed. Usually CA hierarchy involves two tiers as below

- Root CA
- Subordinate Certificate
- End-Entity Certificate

As you can see below

Two tier is absolutely necessary at a minimum. Subordinate CAs do the best they can to protect their certificates, they carry a much higher security risk than root CAs. Subordinate CA.

This multi-leveled hierarchy of trust is called a certificate chain. You can trace the chain from the client's certificate all the way back to a single root CA, and every chain ends with a person (or company) from which all the trust is ultimately derived.

In practice, these chains tend to interlink with other chains – often from other CAs. And those CAs often choose to implicitly trust each other, accepting a signed certificate from another CA without validating it themselves. That's called federation, and while it makes things easier, it means the trust store is only as secure as the weakest link.

More than one CA can sign a certificate, which increases the trust you have that it is accurate because more than one CA has validated it. When more than one CA signs a certificate, it's called cross-signing.

Root CA Security

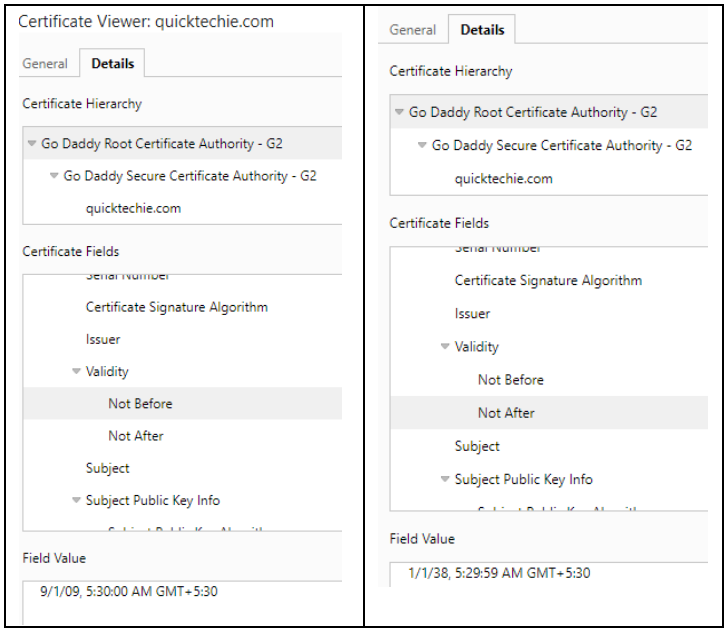
Assume if private key of CAs (issuing authority) is falls under wrong hand then it's a devastating scenario and someone can issue certificates fraudulently. It is even more dangerous because chain in CA hierarchy and there is no way to revoke the root certificate. If root CA is compromised then

organization needs to make that security breach public. That is the reason root CAs have the most tighten security measures.

And that's the reason root CA never be available online and should be stored respective private keys in highly safe and kept in Data center which has all the cameras and 24/7 security guards.

Root CA should be offline almost all the time, However, there are times where Root CA need to come online. Like, root CAs need to come online for the creation of public keys, private keys and new certificates as well as to ensure that its own key material is still legitimate and hasn't been damaged or compromised in any way. Ideally, root CAs should run these tests about 2-4 times a year.

Root CA also expires and lasts for 15-20 years, check below a certificate on https://quicktechie.com Because building a trust in a new root is not easy.



X.509 Certificate formats

X.509 is a digital certificate which comes in various formats as below

- **Single binary certificate**: In this case certificate is a binary data structure containing all the X.509 fields and it is encoded using DER.
- **PKCS #7 binary certificate package**: The PKCS #7 binary certificate package, based on the Public Key Cryptographic Standards (PKCS) published by RSA Laboratories, is a package used to distribute one or more certificates, or an entire chain of certificates. Most commercial certificate authorities return multiple requested certificates using the PKCS #7 format as a convenience rather than distributing certificates individually. When used for distribution purposes, the PKCS #7 package as a whole is neither signed nor encrypted. As with the single binary certificate, the PKCS #7 package does not contain any private keys.
- **PKCS #12 binary certificate package**: The PKCS #12 binary certificate package is a password-encrypted package that can contain nearly any type of data.
- **Base64-encoded certificates**: The binary certificate and the PKCS #7 and #12 binary packages can be additionally encoded using the base64 algorithm. Base64 encoding is a mechanism to convert binary data into text so that it can be easily transported as text, such

as within an e-mail. When converting from binary to text, each three bytes of binary are converted into four characters from the following set: a–z, A–Z, 0–9, \, and +. When you peek at a base64-encoded certificate on any platform, it looks similar to the following:

```
-----BEGIN CERTIFICATE-----
MIIKnzCCCYegAwIBAgIQDZnsm3pSfYyFBQHnkWja5jANBgkqhkiG9w0BAQsFADBP
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMSkwJwYDVQQDEyBE
aWdpQ2VydCBUTFMgUlNBIFNIQTI1NiAyMDIwIENBMTAeFw0yMjA4MjUwMDAwMDBa
Fw0yMzA4MjUyMzU5NTlaMH0xCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhORVcgWU9S
SzEPMA0GA1UEBxMGQXJtb25rMTQwMgYDVQQKEytJbnRlcm5hdGlvbmFsIEJ1c2lu
ZXNzIE1hY2hpbmVzIENvcnBvcmF0aW9uMRQwEgYDVQQDEwt3d3cuaWJtLmNvbTCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALsfuoyq2opTi6EL2mJqDHX3
DAuH3PADDnk2pZoAAAGC1abi4QAABAMARjBEAiB3efNJOT1b6ZbJN8umSkgHwGj2
rXFfaWHKtn8KhhVS3AIgWb1gEZdy4Zsh3D/bEYMwa6dEGiQ0T+Qvh2XEKQ/dCNEw
DQYJKoZIhvcNAQELBQADggEBAJwYCjC5QoOfCi/c84QqlAfs7V6LtHdO8a9kOdHb
9my+sJOBGGPwMZ829hBLYjV9wPysWq5lWT0fG6skovI/KonNnu8Ip6/4BqRNVIwT
NffzJ/IywmGmPJfDfP8cjxaVcyXtPvsuKIMIVg4zBWo2djJYV02UpTyQsFjo4CsP
TVb73VXVgUtwSidUnprKjp9KXReHHF4W1NHWD9ERLePo4hpSW+0YPqKL5gE9lp1s
0pVwY/AVixmcYGTVaEscgFv2cxnDQueaTbMMesaxZhgtTRnaJlqspNi5qklWBY5x
c34w8sFsvPWknApAMPXjTd7OhIMLv/n7NBus+AVz4UYpX1M=
-----END CERTIFICATE-----
```

Types of Digital Certificates

There are mainly three types of certificates

1. TLS/SSL Certificate
2. Code Signing Certificate
3. Client Certificate

TLS/SSL Certificate

As we have discussed this is installed on the server. Server can be Web Server, App Server, Mail Server, LDAP Server.

Code signing certificate

Code signing Certificates are used to sign software or files that are downloaded over the internet. Usually, they are signed by software publisher. The purpose is that whatever software library I gave is genuine. These are useful when you distribute your software via the third-party website.

Client Certificate

Client certificate or Digital IDs are used to identify one user to another, a user to a machine, or a machine to another machine. For example, emails, where sender digitally signs the communication and recipient verifies the signature.

Certificate Signing request

Let's see steps by step process of certification creation for asymmetric encryption key

1. You as an individual or organization create a private and public key pair.
2. You use public key and identifying attributes which can be encoded into a certificate signing request (CSR)
3. Now you as an individual or company signs the public key as a key owner to prove possession of that private key.
4. Now the issuing CA (like Verisign) validates the request and signs the certificate with the CA's own private key.

5. Now anyone can use the public key portion of a certificate to verify that it was issued by a trusted CA and confirming who owns the certificate and sign the certificate.

Important: One critical point you should note is that CA itself uses its own private key and corresponding public key, which creates the need for CA hierarchies.

Cross Signing

Cross-signing expands trust within your network. When a certificate is signed by two CAs, it allows the certificate to verify trust by more than one CA without the need to distribute a separate certificate for each CA. Cross-signing is still effective when one CAs private key is leaked, as you can issue a revocation for all the public keys for that CA, but the certificates that were cross-signed can still maintain a level of trust with the other CA without the need of reissuing certificates for the CA that was revoked.

Chapter-8: Secure Sockets Layer/Transport Layer Security

SSL and TLS are industry standards and used for secure communication over unprotected network, it can be internet or even inside your organization network as well. Both SSL and TLS are cryptographic protocols. SSL was developed by Netscape and was first released on SSL 2.0 in 1995 and then its improved version SSL 3.0 in 1996.

TLS was created by Internet Engineering Task Force (IETF), which is a successor to SSL. TLS is updated and more secure version compare to SSL. Both SSL and TLS can be used for securing network communication. However, the difference between TLS 1.0 and SSL 3.0 are significant and they did not interoperate. Then later TLS 1.1 and TLS 1.2 was released.

TLS or SSL provides the following functionality

- Data Encryption
- Data Origin Authentication
- Message Integrity
- Server and Client Authentication using X.509 certificates.

SSL/TLS X.509 certificates

SSL/TLS X.509 certificates are digital files that are used for secure socket layer or Transport Layer Security (TLS). As we have discussed X.509 certificate contains a public key and the identity of a hostname, organization or individual. This certificate is a fundamental concept of PKI (Public Key Infrastructure).

SSL/TLS certificate does two functions as below.

1. **Authentication**: Certificate can assist with authentication and verifying the identity of a host or site. The SSL certificate has information about the authenticity of details around the identity of a host or site.
2. **Data Encryption**: It enables the encryption of information exchanged via a website.

How TLS/SSL and X.509 Certificate works

As SSL/TLS certificates enable encryption, they are integral to Hyper Text Transfer Protocol Secure (HTTPS), a protocol that encrypts all communication exchanged between a website and your browser.

Step-1: HTTPS starts when a browser requests a secure page.

Step-2: The web server responds with its public key and its certificate.

Step-3: The browser then verifies a trusted authority or CA issued this digital file.

Step-4: Assuming that's the case, the browser uses the web server's public key to encrypt a random symmetric encryption key (session key) and sends it to the server with an encrypted URL and other encrypted HTTP data.

Step-5: If the public key is valid, the web server uses its private key to decrypt the symmetric encryption key, URL, and HTTP data before sending over the HTML document and HTTP data now encrypted with the symmetric key.

Step-6: This symmetric key, in turn, allows the browser to decrypt the HTTP data and display it to the user.

SSL/TLS Protocol

Both SSL and TLS are cryptographic protocol and used for authentication and encryption over the network. There are many similarities between these protocols hence many applications configure their implementation together “SSL/TLS”.

Public key infrastructure uses the TLS protocol to establish secure connections between clients and servers over the internet. TLS was introduced in 1999, since then people are moving away from SSL. Currently we are on TLS 1.3. Usually, TLS is primarily used to secure client-server connection, it is also used to protect emails, VOIP calls, and other connections.

TLS Handshake

Now your web browser wants to make a secure connection with the server in case of HTTPs, there has to be a handshake between both the parties and that happens using asymmetric encryption. First server will send digital certificate to the client and then check if there is any session key already exists for this server and public key (which is in the certificate) if not it creates one and encrypt using the same public key and send it to the server. Now server will decrypt this encrypted session key using its own private key. Now both server and client have the same session key and key is used to encrypt and decrypt all messages which are exchanged in this particular session.

Chapter-9: Key Management

As we have been learning cryptography and encryption algorithms and most important part of all the algorithm is the Cryptography keys. If these keys are compromised then it can collapse the entire security system and attacker can decrypt all the encrypted data and get the sensitive and classified information. Hence, there is a need of the proper management of keys and their related components can ensure the safety of confidential information. Key management is the process of putting certain standards in place to ensure the security of cryptography keys in a company. Key management involves

- Creation of keys
- Exchange of the Keys
- Storage of the keys
- Deletion of the keys
- Refreshing of keys
- Access management of the keys

Key management is a fundamental thing any organization has to do for its data security. Because data are encrypted and decrypted using these keys. In fact, keys also help in transmission of data across an internet or wire. Poorly protected keys can be stolen and misused. Also, keys provide the compliance with certain standards and regulations to ensure companies are using best practices when protecting cryptographic keys.

Asymmetric and Symmetric Keys

As we have been learning that there are two types of cryptographic keys, symmetric and asymmetric keys. Usually, data at rest can use symmetric key for example data in database, in S3 bucket. For example, where you are storing your data in S3 bucket or in a database you can encrypt the data using same key.

Asymmetric keys are relatively complicate to understand. Because you are not using the same key for both encryption and decryption. They always come in pair public key and private key and used for encryption and decryption of the data. But both the keys are related to each other. The public key from the pair is usually used to encrypt the data. And public key you can share with anybody because it usually encrypts the data and do not decrypt the data. And private key is used for the decryption of the data.

Asymmetric keys are mostly used when data are in-transit or being sent over the wire or over the network, whether it's a public or private connection. However, please note that when you are sending highly sensitive data, you are using both symmetric and asymmetric keys to encrypt the data. Let's see the step-by-step process

Step-1: Data on your system is at rest. Encrypt this data using symmetric encryption key. Name of the symmetric key is like SYM-KEY-1.

Step-2: Now, you need to encrypt the symmetric key (SYM-KEY-1) as well before sending to the recipient. And that can be encrypted using recipient public key name it (ASYM-PUBLIC-KEY2) once it is encrypted it can only be decrypted using recipient private key (ASYM-PRIVATE-KEY-2) only.

Step-3: Now, you have both your data encrypted (Cipher Text) and Symmetric Key (SYM-KEY-1) is encrypted and both you can send over the network.

Step-4: Now both ciphertext and encrypted symmetric key (SYM-KEY-1) reaches the recipient.

Step-5: Recipient first decrypt the encrypted symmetric key (SYM-KEY-1) using its own private key (ASYM-PRIVATE-KEY2).

Step-6: Now the recipient has the symmetric key (SYM-KEY-1) as decrypted and use it to decrypt the cipher text.

Working of Key Management

For key management there is a lifecycle of operations to make sure that the key is created, stored, used, and rotated securely. And following are possible lifecycle involved.

- Generation
- Distribution
- Use
- Storage
- Rotation
- Backup & Recovery
- Revocation
- Destruction

Step-1: Generation: The generation is the first step and which algorithm is used to generate that key is very important. If you use the weak algorithm than attacked can find the encryption key. Usually, key generation algorithm like AES encryption or random number generator are used for secure key generations. Now if generated key is placed at in-secure location then, key can be compromised as soon as it is created.

Step-2: Distribution: Now you should be able to safely distribute the keys. Usually, keys are distributed to the required user using secure TLS or SSL connection so that security of the key is maintained while being distributed. If in-secure channel is used then man-in-middle can attack the key and steal it.

Step-3: Use the Key: Now use the key for cryptographic operations. And we have to make sure that key is only used by the authorized users, which make sures that the key is not misused.

Step-4: Storage: When the key is used to encrypt the data, it must be stored for later decryption. The most secure way to store the key is HSM (Hardware Security Module) or CloudHSM. If you are not using HSM then keys can either be securely stored on the client's side, or, if the keys are used on the Cloud, then the Cloud Service Provider's key Management Service can be used.

Step-5: Rotation: Once a Key's cryptoperiod, or time period of the key is usable, passes, the key must be rotated. When the key of an encrypted set of data expires, the key is retired and replaced with a new key. First the data is decrypted by the old key or old key pair and encrypt again with the new key pair. Rotation is necessary because the longer a key is in rotation, the more chance there is

someone to steam or find out the key. You can even do the rotation of the keys before the cryptoperiod expires in cased where the key is suspected to be compromised.

Step-6: Revocation and Destruction: These are another way to deal with the compromised key. If you revoke a key, it means it can not be used to encrypt or decrypt the data, even its cryptoperiod is still valid.

Step-7: Destruction: You can destroy a key any time and it will delete the key permanently from key manager database or storage. If you destroy the key, it makes impossible to recreate the key, unless a backup image is used. If you are following NIST standard then it requires that deactivated keys be kept in an archive, so that it can be reconstructed, if data encrypted in the past must now be decrypted by that key pair.

Best Practice in Key Management

It is not by choice, it is a mandate if you are following NIST standards and regulations like PCI DSS, FIPS and HIPAA. And they expect you to follow the best practices to maintain the security of cryptographic keys used to protect sensitive data. Few of them are follow, which can keep changing over the period.

- **Least Privilege Principal**: It means users should only have access to key that are absolutely necessary for him/her work. It means only authorized users can access important cryptographic keys, and providing better tracking of key usage. Suppose, if key is compromised, only a handful users may have access to the key and suspect pool can be narrowed down if breach is withing the known users.
- **Hard-Coding:** Till now I have not seen this kind of mistakes. However, you should never-ever hard code keys. Specially, if you are distributing your code and you have hard-coded a key in it, then it would be instantly compromised.
- **HSMs:** HSMs are physical device which stores cryptographic keys and performs cryptographic operations on-premises. For an attacker to steam the keys from an HSM, they would need to physically remove the device from the premises, steam a quorum of access cards needed to access the HSM, and bypass the encryption algorithm used to keep the keys secure. HSMs on the Cloud are also a viable key management storage method, but there is always the change that the Cloud Provider's security fails, attached to access the keys stored therein.
- **Automation:** Automation is widely used method to make sure that keys do not go past their cryptoperiod and become over used. In fact, you can automate the other portion of the key as well, for example creating a new key, backing up keys regularly, distributing keys, revoking keys and destroying keys.
- **Split Keys:** You can protect you key by splitting it into multiple parts. And by that not a single person can ever have the full key access. Rather multiple people have to come together to use the key. This assures that others can be held responsible by their peers, if their portion of the key is compromised.
- **Separate Duties:** Separating duties related to key management is another important practice for any organization. An example of separation of duties is that one person is assigned to authorize new user's access to keys, while another distributes the keys, and a third person creates the keys. With this method, the first person cannot steal the key during the distribution phase, or learn the value of the key during the generation phase of the key lifecycle.

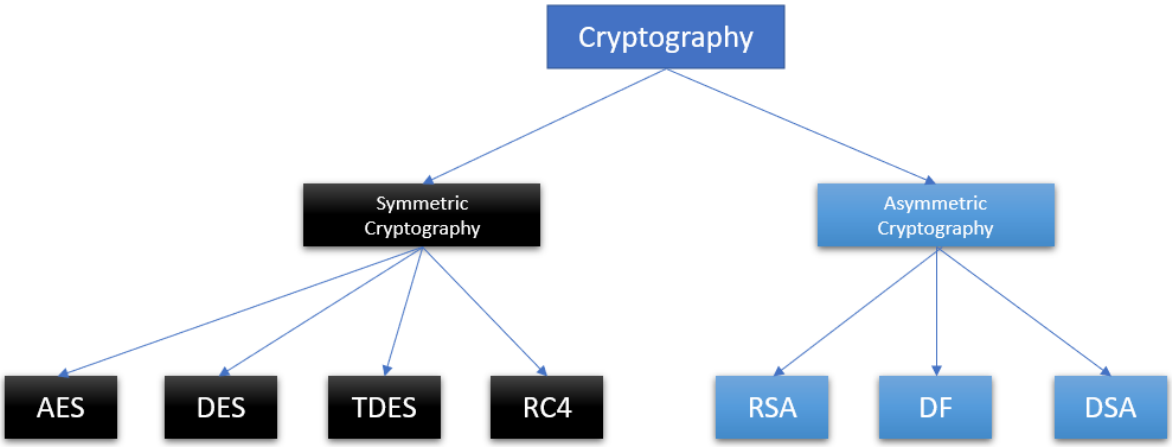
- **Create and Enforce Policies:** Creating and enforcing security policies relating to encryption keys is another way many organizations ensure the safety and compliance of their key management system. Security policies provide the methods everyone within an organization follows, and creates another method of tracking who can and has accessed certain keys.

Chapter-10: Interview Questions & Answer on Cryptography

I have always preferred questions answer approach to make sure my or someone learning is correct or not and the same questions and answer can help you clear your interviews as well. Below are the all-interview questions and answer related to cryptography.

Question-1: What is a Cryptography?

Answer: This is a process of converting your plain text into cipher text and then converting back cipher text to plain text is a cryptography. Purpose of cryptography is to authenticate and protect data from theft, misuse and change. Following diagram show you the hierarchy and study of cryptography.



Question-2: What is a Digital Signature?

Answer: Digital signatures are like electronic "fingerprints." In the form of a coded message, the digital signature securely associates a signer with a document in a recorded transaction.

Question-3: What is a digital certificate?

Answer: A digital certificate is an electronic document issued by a Certificate Authority (CA). It contains **the public key for a digital signature** and specifies the identity associated with the key, such as the name of an organization. The certificate is used to confirm that the public key belongs to the specific organization. The CA acts as the guarantor. Digital certificates must be issued by a trusted authority and are only valid for a specified time. They are required in order to create a digital signature.

Question-4: What is a Certificate Authority (CA)?

Answer: Digital signatures rely on public and private keys. Those keys have to be protected in order to ensure safety and to avoid forgery or malicious use. When you send or sign a document, you need assurance that the documents and the keys are created securely and that they are using valid keys. CAs, a type of Trust Service Provider, are third-party organizations that have been widely accepted as reliable for ensuring key security and that can provide the necessary digital certificates. Both the entity sending the document and the recipient signing it must agree to use a given CA.

For example, DocuSign is also a CA when signers sign using the DocuSign Express Digital Signature. That means you can always send a document with a digital signature by using DocuSign as the Certificate Authority. Alternatively, you can securely establish your own CA using the DocuSign Signature Appliance and still access the rich features of DocuSign cloud services for transaction management. Some organizations or regions rely on other prominent CAs, and the DocuSign platform supports them, as well. These include OpenTrust, which is widely used in European Union countries, and SAFE-BioPharma, which is an identity credential that life science organizations may elect to use.

Question-5: What is Public Key Infrastructure (PKI)?

Answer: Public Key Infrastructure (PKI) is a set of requirements that allow (among other things) the creation of digital signatures. Through PKI, each digital signature transaction includes a pair of keys: a private key and a public key. The private key, as the name implies, is not shared and is used only by the signer to electronically sign documents. The public key is openly available and used by those who need to validate the signer's electronic signature. PKI enforces additional requirements, such as the Certificate Authority (CA), a digital certificate, end-user enrolment software, and tools for managing, renewing, and revoking keys and certificates.

Question-6: Are electronic signatures legal?

Answer: In the US, electronic signatures have been used since the passage of the US Federal ESIGN Act in 2000. Electronic signatures are used for electronically signing offer letters, sales contracts, permission slips, rental/lease agreements, liability waivers, financial documents, etc. They are legally enforceable in most business and personal transactions in almost every country in the world.

Question-7: What is a CSR?

Answer: A certificate signing request (CSR) is one of the first steps towards getting your own SSL/TLS certificate. Generated on the same server you plan to install the certificate on, the CSR contains information (e.g., common name, organization, country) the Certificate Authority (CA) will use to create your certificate. It also contains the public key that will be included in your certificate and is signed with the corresponding private key. We'll go into more details on the roles of these keys below.

Question-8: What information is included in a CSR?

Answer: Following information is included in certification signing request

1. Information about your business and the website you're trying to equip with SSL, including:
- Common Name (CN)(e.g. *.hadoopexam.com, www.hadoopexam.com, mail.hadoopexam.com): This is a fully qualified name of your server.
- Organization (O): The legal name of your organization. Do not abbreviate and include any suffixes, such as Inc., Corp., or LLC.
- Organizational Unit (OU): The division of your organization handling the certificate.
- City/Locality (L)/ Country/ State: The city where your organization is located. This shouldn't be abbreviated.
- Email Address: Email address used to contact your organization.
2. The public key that will be included in the certificate.
3. Information about the key type and length. The most common key size is RSA 2048, but some CAs, including GlobalSign, support larger key sizes (e.g. RSA 4096+) or ECC keys.

Question-9: Why do I need to re-key my certificate?

Answer: You should re-key your SSL certificate when:

- You change the domain name (common name) of your SSL.
- You want to change your SSL certificate's issuing organization. For more information, see Using the Right Issuing Organization for Your SSL.
- Your server crashes.
- You lose your private key.
- You move your website to a new server.

Question-10: In case of web server, Where's my private key?

Answer: When you generate a CSR to request an SSL certificate, a private key is created on your local server. You shouldn't ever send your private key to someone else (including, your web hosting provider like Godaddy), but you'll need to know where it is to complete the installation of your SSL certificate. Below are some common places to find the private key on your server.

Windows/IIS: On Windows servers, the OS manages the certificate for you in a hidden file, but you can export a .PFX file that contains both the certificate and the private key.

1. Open the Microsoft Management Console (MMC).
2. In the Console Root, expand Certificates (Local Computer). Your certificate will be located in the Personal or Web Server folder.
3. Locate and right click the certificate, click Export and follow the guided wizard.

Once you have the .pfx file, you can keep it as a backup of the key, or use it to install the certificate on another Windows server. If you're trying to use the certificate on a different OS, you'll need to split the .pfx into the certificate and the private key.

Apache: You can find the location of your private key in your Apache configuration file, which is named .httpd.conf or apache2.conf. The line SSLCertificateKeyFile shows you the file path to your private key.

NGINX: The path to your private key is listed in your site's virtual host file. Navigate to the server block for your site (by default, it's located in the /var/www directory). Open the configuration file for your site and search for ssl_certificate_key which will show the path to your private key.

More info

- Still can't find your private key? Try searching for a ".key" file, or following the installation steps for your server type. The installation steps should include where your private key is located.
- If your private key is nowhere to be found, or your site isn't serving HTTPS connections, you will need to rekey your certificate, and save your private key in a location where you can find it.

Question-11: What is a Crypter and cryptographer?

Answer: A cryptographer is person who does cryptography, and that implies planning or dissecting any part of encryption. A crypter, then again, is an individual attempting to guilefully camouflage some malware as anything different like a helpful program, so that it can be spread undetected.

Question-12: What is a Ciphertext?

Answer: In cryptography, ciphertext or cyphertext is the result of encryption performed on plaintext using an algorithm, called a cipher.[1] Ciphertext is also known as encrypted or encoded information because it contains a form of the original plaintext that is unreadable by a human or computer without the proper cipher to decrypt it. This process prevents the loss of sensitive information via hacking. Decryption, the inverse of encryption, is the process of turning ciphertext into readable plaintext.

Question-12: What are Ciphers?

Answer: Cipher is a process of creating data in a non-readable form. In other words, you can say it is an algorithm responsible for the encryption and decryption of data.

Question-13: What is RSA in the field of Cryptography?

Answer: RSA (Rivest–Shamir–Adleman) is an asymmetric cryptographic algorithm. It consists of two keys: Public and Private keys. The Private key holds only by the owner of that key, and the corresponding public key is available to different persons. If encryption is happening with the private key, decryption can be done with the public key, and vice versa depending on the usage of asymmetric encryption.

Question-14: What is the major difference between the Symmetric and Asymmetric Key Algorithm?

Answer: Difference between the Symmetric and Asymmetric Key algorithms is using the same key in the case of the Symmetric Key algorithm while using different keys (public and private key) in the case of the Asymmetric Key Algorithm.

Question-15: What are the advantages of the Symmetric Key Algorithm?

Answer: The main advantage of the Symmetric Key Algorithm is the fast speed of encryption in comparison with the Asymmetric Key Algorithm. Another important advantage of this algorithm is the property of extreme security that makes it unbreakable.

Question-16: What is Block Cipher?

Answer: Block cipher is a method of encrypting data using cryptographic keys and algorithms to apply to a block or chunks of the message simultaneously rather than individually. The transposition cipher, AES, DES, 3DES, Twofish are examples of a Block cipher.

Question-17: What is Stream Cipher?

Answer: In this cipher, the cryptographic algorithm is used to encrypt or decrypt a message one bit or character at a time. Here, stream ciphers encrypt data byte by byte. The Caesar cipher is an example of the stream cipher.

Question-18: What is the Data Encryption Standard (DES)?

Answer: DES or Data Encryption Standard is a symmetric-key algorithm to encrypt data into a non-readable form. DES uses the same key of size 56 bits to encrypt and decrypt data.

Question-19: What is Triple DES (3DES)?

Answer: Triple-DES is a type of symmetric-key algorithm and uses 168 bits keys (three 56 bits keys) to encrypt or decrypt a message. It is considered a strong algorithm than DES.

Question-20: What is the Advanced Encryption Standard (AES)?

Answer: Advanced Encryption Standard (AES) is a symmetric key block cipher used to encrypt and decrypt messages.

Question-21: What is ECB mode?

Answer: ECB stands for Electronic CodeBook. It is the simplest mode of operation in the block cipher. Here, each plain block is encrypted independently to produce a ciphertext block. For the same plain block, the ciphertext block will remain the same.

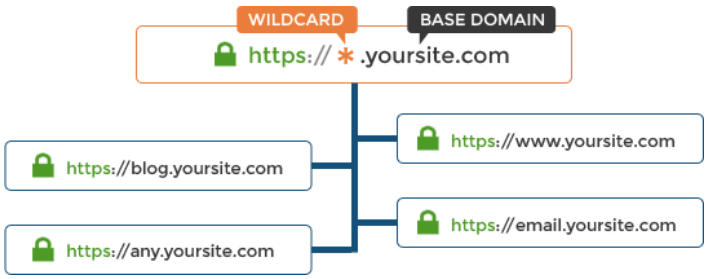
Question-22: What is the difference between Encryption and Hashing?

Answer: Both Encryption and Hashing are used to convert readable data into an unreadable format. The difference is that the encrypted data can be converted back to original data by the process of decryption but the hashed data cannot be converted back to original data.

Question-23: What is the SAN SSL Certificate Vs. Wildcard SSL certificate?

Answer: There was a time when we needed to purchase separate SSL certificates for all our websites and subdomains. This used to result in a significant amount of administrative work and cost. Fortunately, that's not the case anymore. Today we have products like wildcard SSL certificates and Multi-Domain (SAN/UCC) certificates to help to secure all sub-domains and multiple different domains with a single certificate. However, the thing is that many people don't understand the difference between these both products and where each of these products should be used.

- **Wildcard SSL certificates**: allow you to protect multiple sub-domains with a single certificate. For example, you can purchase one wildcard certificate for your website hadoopexam.com and that certificate will protect login.hadoopexam.com, blog.hadoopexam.com and so on.



- **Multi-Domain SSL certificates:** also known as SAN or UCC certificates that allow you to protect multiple different domain names with a single certificate. For example, you can get a SAN certificate for hadoopexam.com and that certificate can be used to protect quicktechie.com.au, training4exam.com and so on. You just need to add the SAN entries of all your domains to the SAN certificate while generating it, and the certificate generated with those entries will protect all your desired domains.

Question-24: What are the "Limitations of Wildcard Certificates"?

Answer: A major limitation of wildcard certificates is that they can protect sub-domains at one specific domain level only. For example, you can protect sub.hadoopexam.com, domain. hadoopexam.com or other similar sub-domains with a wildcard certificate generated for example.com, but that certificate won't work for sub.quicktechie.com, login.sub.hadoopexam.com or any other sub-domain that's more than one level away from the root domain.

Question-25: What are the "Limitations of Multi-Domain SAN certificates"?

Answer: There's a limit on how many domains you can protect with these certificates. Majority certificate authorities in the market won't allow you to protect more than 250 domains. Some of them come with even more restrictions and allow protecting up to 25 domains only. That's not too much of a problem though, because unless you're running a huge enterprise, you won't need to create more than 250 different domains.

Most Multi-Domain SSL certificate prices include the limited number of SAN entries, you need to pay extra for additional SAN licenses.

While Multi-Domain SAN certificates allow us to protect multiple domains as well as subdomains with a single certificate, wildcard certificates help us achieve that with subdomains only. However, wildcards do not put a limit on how many subdomains we can protect with them, while multi-Domain SAN certificates do. But that doesn't make any of these two certificate types superior from each other – each of them has its own use cases, and that should be understood carefully before choosing either of these products for website protection. You should choose either of these products based on your current and future requirements.

Question-26: What is the difference between encryption and decryption?

To begin answering the difference between encryption and decryption in cryptography questions, scrambling and descrambling are sometimes known as "encryption" and "decryption" in terms of decryption, respectively. For instance, when the sending party's composed message "Take Care" is mixed, it becomes what is known as the "scrambled message." This implies that the compiled idea has been concealed so that it would be unimportant, or in terms of cryptography, it would be undecipherable.

Moreover, in cryptography questions, it can be detailed that encryption can also be shown as the change of data from a logical state to an obvious vague state. When the receiving party gets this encoded compiled idea, it should be unscrambled into a coherent and understandable condition of the unique situation. This course of unscrambling is otherwise called decoding or decryption.

Question-27: What is Salting?

Answer: Salting is the process of obfuscating passwords by introducing an arbitrary value to them after they've been hashed. Since individuals will generally reuse passwords, a hacker with a rundown of normal passwords or taken ones secures his position simpler. A Salt is an arbitrarily created number, yet it decreases the chance of being in a predetermined table.

Question-28: Why is SSL encrypting are considered insufficient?

Answer: SSL only secures your information while it is in transit. It does not safeguard data after it reaches the recipient. SSL also does not encrypt your information. When it comes to processing metadata, this might be a flaw. This enables your prospective hacker to create a client profile and consider network layouts.

Question-29: What is the purpose of a Key Distribution Center (KDC)?

Answer: The KDC comprises a system that contains the names and passwords of all end-users at the business or organization and many other authorized servers and networking devices.

Question-30: What is secret-key cryptography?

Answer: This is the same as symmetric encryption, both parties need to have the key and need to agree on the cryptographic algorithm. Issues include

- How to get the key from 1 party to another securely, typically fixed with asymmetric encryption to transport key.
- Compromised key means the whole session is compromised.
- Compromised key can be used to send data to one of the parties and impersonate one of the participants.

Question-31: What is a session key?

Answer: Generally speaking, a session key is used to secure communications between two computers or a user and a client or a client and a server. A session key is a single use type of key that can be transmitted along with each message and is encrypted with the recipient's public key.

Single use symmetric keys are faster, but they will need to be generated randomly. Session keys are used in Open PGP. There can be a public key encrypted session key or a passphrase-based encryption.

Question-32: What is RSA?

Answer: RSA is a public key algorithm for key management digital signatures, and is used for asymmetric encryption. There is 1024-bit RSA and 2048-bit RSA. It uses large prime factors to calculate a modulus that can be used as a key, along with the recipient's public key to send an encrypted message.

Question-33: Is RSA currently in use today?

Answer: Yes, its everywhere- both by private and governments.

Question-34: What is difference between DSA and RSA?

Answer: DSA is faster in signing, but slower in verifying. DSA uses 1024 bits. RSA can go up to 2048 bits. RSA can be used for encryption and authentication, but DSA can only be used for encryption.

Question-35: What is one-way hash function?

Answer: Hash you can't undo. Password hashes. 2-way hash is reversible.

Question-36: What are the applications of a hash function?

Answer: A hash function can be used in a verity of ways when a function is needed to map data of arbitrary size to data of a fixed size. It is used to store password hashes. In other words, it is used to protect data in a way that uniquely identifies each data set since they are one-way functions.

Question-37: What's the difference between Diffie-Hellman and RSA?

Answer: RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman.

Diffie-Hellman key exchange (D-H) is a specific method of exchanging cryptographic keys. It is one of the earliest practical examples of key exchange implemented within the field of cryptography. The Diffie-Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel.

Question-38: What is SSL and why is it not enough when it comes to encryption?

Answer: SSL is used to establish an encrypted link between a server and a client that allows secure transmission of sensitive data. This is a security protocol so it describes how algorithms are used and it determines variables of the encryption for both the link and data being transmitted. When a browser tries to connect to a web server, a SSL handshake occurs. Three keys are used to set up the SSL connection: public, private, and session keys. The public and private keys are used to create a symmetric session key. After that, the session key is used to encrypt the transmitted data. Browser connects to web server. Server sends SSL Cert, and public key. Browser checks cert root against trusted CAs. If it's okay, then it creates, encrypts and sends back a symmetric session key encrypted with the server's public key. Server decrypts the symmetric session key with its private key and sends back an acknowledgement encrypted with the session key. After that, the server and browser encrypt the data with the session key. SSL (application layer) works on top of TCP (transport layer). The order goes: HTTPS/SSL (app), TCP (transport), IPSEC (internet).

SSL only helps protect your data while it is in transit, but once it has reached its destination, then it cannot do much. SSL does not encrypt all of your data. One example is the metadata that is created and not encrypted. One real weak point is the handling of metadata. Metadata can help attackers build user profiles, and can help them map out a network. Examples of metadata is: IP and MAC addresses, protocol types, other network information. Certificate authorities that are trusted could also be compromised. Certificates could be outdated, they could be duplicates if an attacker has compromised it. Encryption and data protection while data is in transit and while data is at rest are very important.

Question-39: What is salting, and why is it used?

Answer: Salting is when a random value is added to passwords in order to obfuscate the password after hashing.

People tend to use the same passwords and not randomly, so it makes it easier for an attacker who has access to a list of commonly used passwords or stolen passwords. Rainbow tables have passwords and their hashes listed out, so this makes it even easier for an attacker. A salt is generated at random and can be small, but the main purpose is to lower the probability of the hash value to be found in a pre-calculated table. You can concatenate the password and the salt prior to hashing it. Salts can be stored in the clear in the DB, next to the hashed value.

Another defence against stolen passwords is to hash the passwords multiple times. A pepper is similar to a salt, and is used by concatenating it with the salt and the password, but the pepper is not stored at all.

Question-40: What are salted hashes?

Answer: Salted hashes are passwords that have been hashed using a proven hashing algorithm, like SHA2, but before they are hashed, a salt value has been added.

Question-41: How is a Key Distribution Center (KDC) used?

Answer: The Key Distribution Center consists of a database of all of the end users at the place of business or corporation and their respective passwords, as well other trusted servers and computers along the network.

If an end user wishes to communicate with another end user on a different computer system, the sending party enters their password into the KDC using a specialized software called “Kerberos.” When the password is received by the KDC, the Kerberos then uses a special mathematical algorithm which adds the receiving party’s information and converts it over to a cryptographic key.

Once this encrypted key has been established, the KDC then sets up and establishes other keys for the encryption of the communication session between the sending and the receiving party. These other keys are also referred to as tickets. These tickets will actually expire at a predetermined point in time in order to prevent unauthorized use, and it would also be rendered useless if it is stolen, hijacked or intercepted by a third party.

Question-42: What are the specific components of the Public Key Infrastructure (PKI)?

Answer: The PKI consists of the following components:

1. The Certificate Authority (CA): This is the party who issues the digital certificates
2. The Digital Certificate: This serves to verify the identity of the certificate holder and is issued by the CA. These digital certificates are typically kept in the local computer of the employee, or even the central server at the place of business or organization
3. The LDAP or X.500 Directories: These are the databases which collect and distribute the digital certificates from the CA
4. The Registration Authority (RA): If the place of business or organization is very large (such as a multinational corporation), this entity usually handles and processes the requests for the required digital certificates and then transmits those requests to the CA to process and create the required digital certificates

Question-43: What are the technical specifications of the Certificate Authority, which provides details in a certificate?

Answer: The Certificate Authority consists of the following technical specifications:

1. The digital certificate version number
2. The serial number
3. The signature algorithm identifier
4. The issuer names
5. The validity periods
6. The public key
7. The subject distinguished name
8. The subject alternate name email
9. The subject name URL

Question-44: What is the LDAP protocol and how is it used in a Public Key Infrastructure (PKI)?

Answer: LDAP is an acronym which stands for Lightweight Directory Access Protocol. This is a database protocol used for the updating and searching of the directories which run over the TCP/IP network protocol (this is the network protocol which is primarily used by the PKI infrastructure).

It is the job of the LDAP server of the Public Key Infrastructure to contain information and data as it relates to the digital certificates and the public and the private key storage locations, as well as the matching public and private key labels.

The Certificate Authority uses a combination of the end user name and the matching tags to specifically locate the digital certificates on the LDAP server. From that point onwards, the LDAP server checks to see if the requested digital certificate is valid or not, and it if it is valid, it then retrieves a digital certificate which can then be sent to the end user.

Although all digital certificates have a finite lifespan when they are first issued, they can also be revoked for any reason at any time by the Public Key Infrastructure Administrator.

Question-45: What is a General-Purpose Hardware Security Module (HSM)?

Answer: Hardware Security Modules (HSMs) are hardened, tamper-resistant hardware devices that strengthen encryption practices by generating keys, encrypting and decrypting data, and creating and verifying digital signatures. Some hardware security modules (HSMs) are certified at various FIPS 140-2 Levels. Hardware security modules (HSMs) are frequently used to:

1. Meet and exceed established and emerging regulatory standards for cybersecurity
2. Achieve higher levels of data security and trust
3. Maintain high service levels and business agility

Question-46: What is a Certificate Authority?

Answer: A Certificate Authority (CA) is the core component of a public key infrastructure (PKI) responsible for establishing a hierarchical chain of trust. CAs issue the digital credentials used to certify the identity of users. CAs underpin the security of a PKI and the services they support, and therefore can be the focus of sophisticated targeted attacks. In order to mitigate the risk of attacks against Certificate Authorities, physical and logical controls as well as hardening mechanisms, such as hardware security modules (HSMs) have become necessary to ensure the integrity of a PKI.

Question-47: What is Code Signing?

Answer: In public key cryptography, code signing is a specific use of certificate-based digital signatures that enables an organization to verify the identity of the software publisher and certify the software has not been changed since it was published.

Digital signatures provide a proven cryptographic process for software publishers and in-house development teams to protect their end users from cybersecurity dangers, including advanced persistent threats (APTs), such as Duqu 2.0. Digital signatures ensure software integrity and authenticity. Digital signatures enable end users to verify publisher identities while simultaneously validating that the installation package has not been changed since it was signed. All modern operating systems look for and validate digital signatures during installation, and warnings about unsigned code can cause end users to abandon installation.