Unit 4

Public Key Infrastructure

MKK

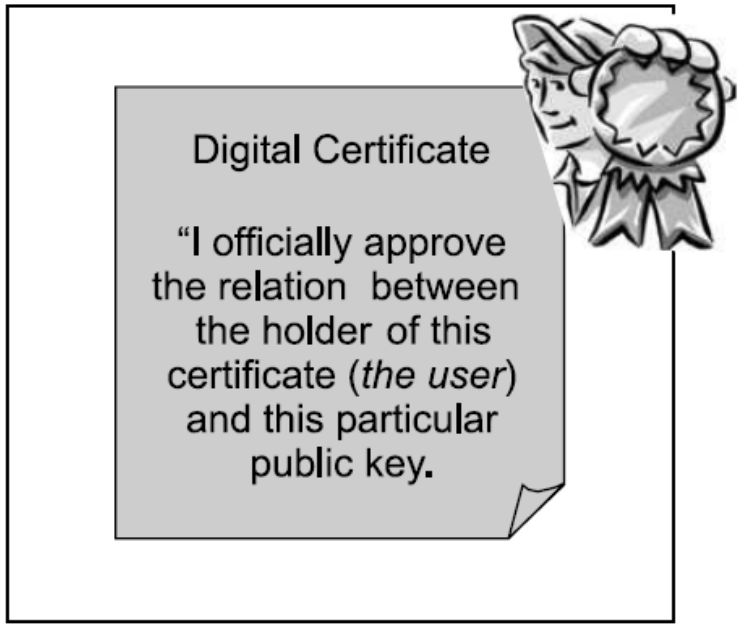
INTRODUCTION

- Public Key Infrastructure (PKI) technology has attracted significant attention, and has become the central focus of modern security mechanisms on the Internet
- PKI is closely related to the ideas of asymmetric-key cryptography mainly including message digests, digital signatures and encryption services.
- Digital certificates are termed passports on the Web.

DIGITAL CERTIFICATES

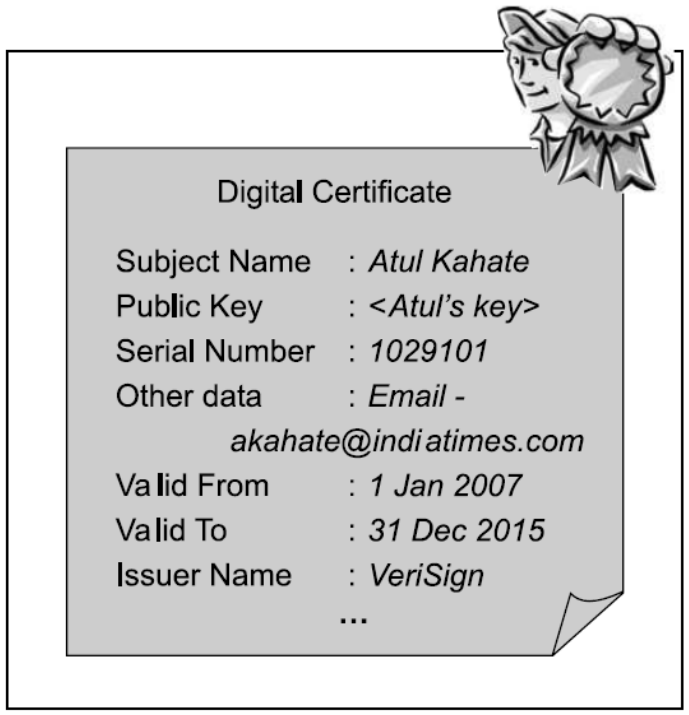
- The asymmetric-key cryptography can be a very good solution for solving the problem of key agreement or key exchange
- But it also has one unresolved issue, which is how the parties/correspondents exchange their public keys with each other.
- Obviously, they cannot exchange them openly—this can very easily lead to a man-in-the-middle attack on the public key itself!
- After a lot of thought, this problem was resolved with a revolutionary idea of using **digital certificates.**

The Concept of Digital Certificates



- A digital certificate is simply a small computer file.
- Just as my passport signifies the association between me and my other characteristics such as my full name, nationality, date and place of birth, photograph and signature, my digital certificate simply signifies the association between my public key and me.
- As we have noted, a digital certificate establishes the relation between a user and his/her public key. Therefore, a digital certificate must contain the user name and the user's public key. This will prove that a particular public key belongs to a particular user. Apart from this, a digital certificate contains
  - subject name
  - Public key
  - serial number
  - Other data
  - Valid from, valid to
  - issuer name

MKK



As the figure shows, the digital certificate is actually quite similar to a passport.

Just as every passport has a unique passport number, every digital certificate has a unique serial number.

As we know, no two passports issued by the same issuer (i.e. government) can have the same passport number.

Similarly, no two digital certificates issued by the same issuer can have the same serial number.

Passport entry	Corresponding digital-certificate entry
Full name	Subject name
Passport number	Serial number
Valid from	Same
Valid to	Same
Issued by	Issuer name
Photograph and signature	Public key

MKK

Certification Authority (CA)

A Certification Authority (CA) is a trusted agency that can issue digital certificates. Who can be a CA?. The authority of acting as a CA has to be with someone who everybody trusts. Consequently, the governments in various countries decide who can and who cannot be a CA.

Usually, a CA is a reputed organization, such as a post office, financial institution, software company, etc. Two of the world's most famous CAs are VeriSign and Entrust.

Safescrypt Limited, a subsidiary of Satyam Infoway Limited, became the first Indian CA in February 2002.

Thus, a CA has the authority to issue digital certificates to individuals and organizations, who want to use those certificates in asymmetric-key cryptographic applications.

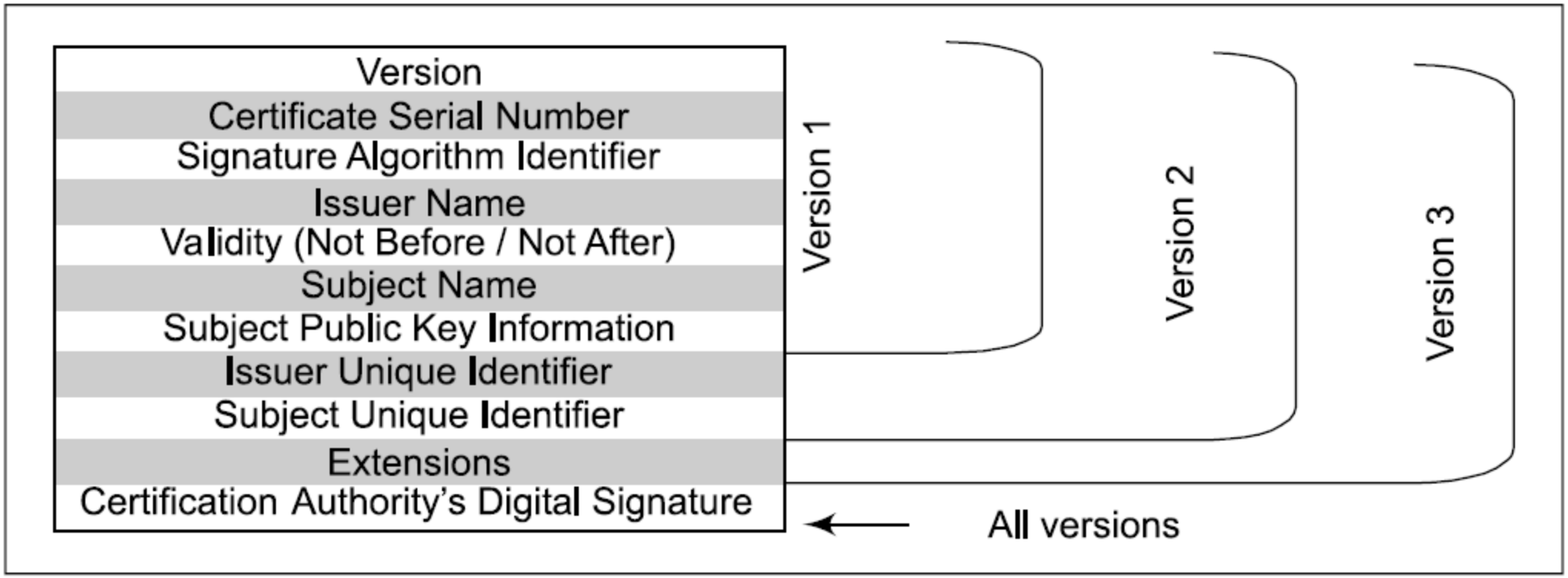
Technical Details of a Digital Certificate

A standard called X.509 defines the structure of a digital certificate. The International Telecommunication Union (ITU) came up with this standard in 1988. At that time, it was a part of another standard called X.500. Since then, X.509 was revised twice

The current version of the standard is Version 3, called X.509V3. The Internet Engineering Task Force (IETF) published the RFC2459 for the X.509 standard in 1999.

MKK

Contents of a digital certificate



MKK

Private Key Management

A private key, also known as a secret key, is a variable in cryptography that is used with an algorithm to encrypt and decrypt data.

Private Key Management includes

- Protecting Private Keys
- Multiple Key Pairs
- Key Update
- Key Archival

Protecting Private Keys

- As we know, a user must hold the private key secretly. It must not be possible for another user to access someone's private key.
- How can a private key be protected? There are several mechanisms, as shown below:

Mechanism	Description
Password protection	This is the simplest and most common mechanism to protect a private key. The private key is stored on the hard disk of the use's computer as a disk file. This file can be accessed only with the help of a password or a *Personal Identification Number (PIN)*. Since anyone who can guess the password correctly can access the private key, this is considered the least secure method of protecting a private key.
PCMCIA cards	The Personal Computer Memory Card International Association (PCMCIA) cards are actually chip cards. The private key is stored on such a card, which means that it need not be on the user's hard disk. This reduces the chances of it being stolen. However, for a cryptographic application such as signing or encryption, the key must travel from the PCMCIA card to the memory of the user's computer. Therefore, there is still scope for it being captured from there by an attacker.
Tokens	A **token** stores the private key in an encrypted format. To decrypt and access it, the user must provide a **one-time password** (which means that the password is valid only for that particular access; next time, this password becomes invalid, and another must be used). We shall later discuss how this works. This is a more secure method.

MKK

Biometrics	The private key is associated with a unique characteristic of an individual (such as fingerprint, retina scan or voice comparison). This is similar in concept to the tokens, but here the user need not carry anything with him/her, unlike the token.
Smart cards	In a smart card, the private key of the user is stored in a tamperproof card. This card also contains a computer chip, which can perform cryptographic functions such as signing and encryption. The biggest benefit of this scheme is that the private key never leaves the smart card. Thus, the scope for its compromise is tremendously reduced. The disadvantage of this scheme is that the user needs to carry the smart card with him/her, and compatible smart-card readers must be available to access it.

- Apart from this In many situations, the private key of the user might be required to be transported from one location to another.
- For instance, suppose the user wants to change his/her PC. To handle these situations, there is a cryptographic standard by the name PKCS#12. This allows a user to export his/her digital certificate and private key in the form of a computer file.
- Obviously, the certificate and the private key must be protected as they are moved to another location. For this, the PKCS#12 standard ensures that they are encrypted using a symmetric key, which is derived from the user's private-key protection password.

MKK

Multiple Key Pairs

- The PKI (Public Key Infrastructure) approach also recommends that in serious business applications, users should possess multiple digital certificates, which also means multiple key pairs.
- The need for this is that one certificate could be strictly used for signing, and another for encryption. This ensures that the loss of one of the private keys does not affect the complete operations of the user. The following guidelines are generally helpful:
  - The private key that is used for digital signing (non-repudiation) must not be backed up or archived after it expires. It must be destroyed.
  - In contrast, the private key used for encryption/decryption must be backed up after its expiry, so that encrypted information can be recovered even at a later date.

Key Update

Good security practices demand that the key pairs should be updated periodically. This is because over time, keys become susceptible to cryptanalysis attacks. Causing a digital certificate to expire after a certain date ensures this. This requires an update to the key pair. The expiry of a certificate can be dealt with in one of the following two ways:

- The CA (certificate authority) reissues a new certificate based on the original key pair
- A fresh key pair is generated, and the CA issues a new certificate based on that new key pair.

The key update process itself can be handled in two ways, as follows:

- In the first approach, the end user has to detect that the certificate is about to expire, and request the CA to issue a new one.
- In the other approach, the expiry date of the certificate is automatically checked every time it is used, and as soon as it is about to expire, its renewal request is sent to the CA. For this, special systems need to be in place.

Key Archival

- The CA must plan for and maintain the history of the certificates and the keys of its users.
- For instance, suppose that someone approaches the CA of Alice, requesting the CA to make Alice's digital certificate available, as was used three years back to sign a legal document for verification purposes. If the CA has not archived the certificates, how can the CA provide this information? This can cause serious legal problems.
- Therefore, key archival is a very significant aspect of any PKI solution.

THE PKIX MODEL

As we know, the X.509 standard defines the digital-certificate structure, format and fields. It also specifies the procedure for distributing the public keys. In order to extend such standards and make them universal, the Internet Engineering Task Force (IETF) formed the Public Key Infrastructure X.509 (PKIX) working group. This extends the basic philosophy of the X.509 standard, and specifies how the digital certificates can be deployed in the world of the Internet.

PKIX Services

(a) Registration It is the process where an end-entity (subject) makes itself known to a CA.

(b) Initialization This deals with the basic problems, such as the methodology of verifying that the end-entity is is talking to the right CA.

(c) Certification In this step, the CA creates a digital certificate for the end-entity and returns it to the end-entity

(d) Key-Pair Recovery Keys used for encryption may be required to be recovered at a later date for decrypting some old documents.

(e) Key Generation PKIX specifies that the end-entity should be able to generate private-and public-key pairs

(f) Key Update This allows a smooth transition from one expiring key pair to a fresh one, by the automatic renewal of digital certificates.

(g) Cross-certification Helps in establishing trust models

(h) Revocation PKIX provides support for the checking of the certificate status in two modes: online (using OCSP-Online Certificate Status Protocol) or offline (using CRL-Certificate Revocation List).

MKK

Public Key Cryptography Standards

- The PKCS model was initially developed by RSA Laboratories with help from representatives of the
- government, industry, and academia.
- The main purpose of PKCS is to standardize Public Key Infrastructure (PKI).
- The standardization is in many respects, such as formatting, algorithms and APIs.
- This would help organizations develop and implement inter-operable PKI solutions, rather than everyone choosing their own standard.

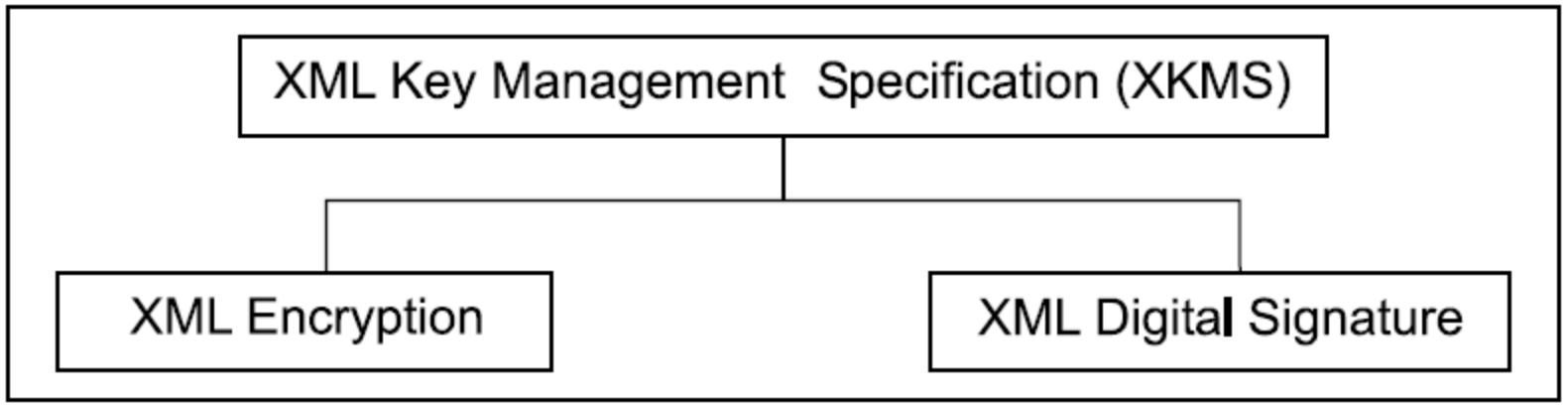
Standard	Purpose	Details
PKCS#1	RSA Encryption Standard	Defines the basic formatting rules for RSA public key functions, more specifically the digital signature. It defines how digital signatures should be calculated, including the structure of the data to be signed as well as the format of the signature. The standard also defines the syntax for RSA private and public keys.
PKCS#2	RSA Encryption Standard for Message Digests	This standard outlined the message-digest calculation. However, this is now merged with PKCS#1, and does not have an independent existence.
PKCS#3	Diffie–Hellman Key Agreement Standard	Defines a mechanism to implement Diffie–Hellman Key Agreement protocol.
PKCS#4	NA	Merged with PKCS#1.
PKCS#5	Password Based Encryption (PBE)	Describes a method for encrypting an octet string with a symmetric key. The symmetric key is derived from a password.

PKCS#6	Extended Certificate Syntax Standard	Defines syntax for extending the basic attributes of an X.509 digital certificate.
PKCS#7	Cryptographic Message Syntax Standard	Specifies a format/syntax for data that is the result of a cryptographic operation. Examples of this are digital signatures and digital envelopes. This standard provides many formatting options, such as messages that are only signed, only enveloped, signed and enveloped, etc.
PKCS#8	Private Key Information Standard	Describes the syntax for private-key information (i.e. the algorithm and attributes used to generate the private key).
PKCS#9	Selected Attribute Types	Defines selected attribute types for use in PKCS#6 extended certificates (e.g. email address, unstructured name and address).
PKCS#10	Certificate Request Syntax Standard	Defines syntax for requesting for digital certificates. A certificate request contains a Distinguished Name (DN) and public key.
PKCS#11	Cryptographic Token Interface Standard	This standard, also called **Cryptoki**, specifies an API for the single-user devices that contain cryptographic information, such as private keys and digital certificates. These devices are also capable of performing cryptographic functions. Smart cards are examples of such devices.
PKCS#12	Personal Information Exchange Syntax Standard	Defines syntax for personal identity information, such as private keys, digital certificates, etc. This allows the users to transfer their certificates and other personal identity information from one device to another, using a standard mechanism.
PKCS#13	Elliptic Curve Cryptography Standard	Currently under development, this standard deals with a new cryptographic mechanism called *Elliptic Curve Cryptography*.
PKCS#14	Pseudo-Random Number Generation Standard	Currently under development, this standard will specify the requirements and process of random number generation. Since random numbers are extensively used in cryptography, standardizing their generation is important.
PKCS#15	Cryptographic Token Information syntax standard	Defines a standard for cryptographic tokens, so that they can interoperate.

MKK

XML

- The overall technology related to XML and security can be summarized as shown in the following Fig



XML Encryption

- The most interesting part about XML encryption is that we can encrypt an entire document, or its selected portions.

The steps involved in XML encryption are quite simple, and are as follows:

1. Select the XML to be encrypted (one of the items listed earlier, i.e. all or part of an XML document).
2. Convert the data to be encrypted in a canonical form (optional).
3. Encrypt the result using public key encryption.
4. Send the encrypted XML document to the intended recipient.

- sample XML document, containing the details of a credit card user.

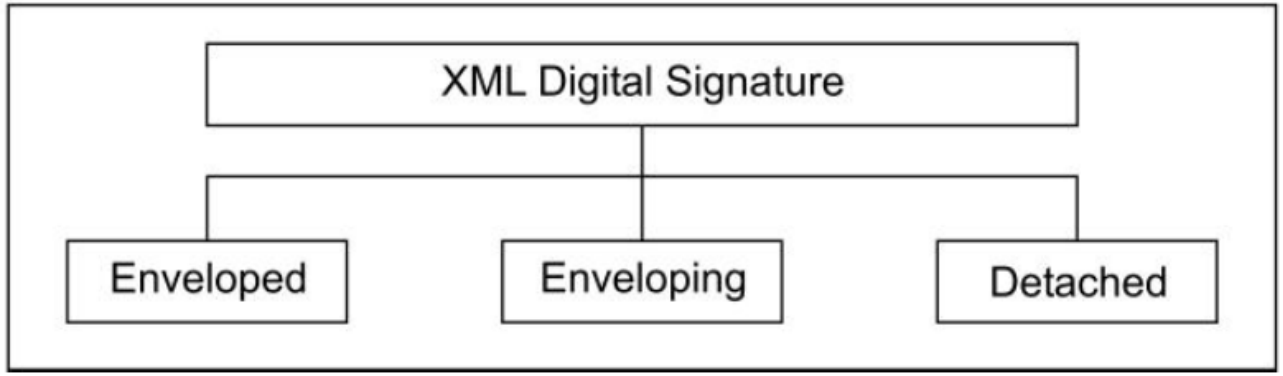
```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://mybank.org'>
        <Name> John Smith <Name/>
        <CreditCard Limit='10000' Currency='USD'>
                <Number> 1617 1718 0181 9910 </Number>
                <Issuer> Master </Issuer>
                <Expires> 05/05 </Expires>
        </CreditCard>
</PaymentInfo>
```

XML Digital Signature

The steps in performing XML digital signatures are as follows.

1. Create a SignedInfo elementwith SignatureMethod, CanonicalizationMethod and References.
2. Canonicalize the XML document.
3. Calculate the SignatureValue, depending on the algorithms specified in the SignedInfo element.
4. Create the digital signature (i.e. Signature element), which also includes the SignedInfo, Key-Info, and SignatureValue elements.

XML digital signatures can be classified into three types: enveloped, enveloping, and detached



The difference between these types of XML digital signatures is easy to understand.

- In the *enveloped XML* digital signatures, the signature is inside the original document (which is being digitally signed).
- In the *enveloping XML* digital signatures, the original document is inside the signature.
- A *detached* digital signature has no enveloping concept at all, it is separate from the original document.

Enveloped Signature

```
<Original_document>
        <Signature> ... </Signature>
</Original_document>
```

Enveloping Signature

```
<Signature>
        <Original_document>
        </Original_document>
</Signature>
```

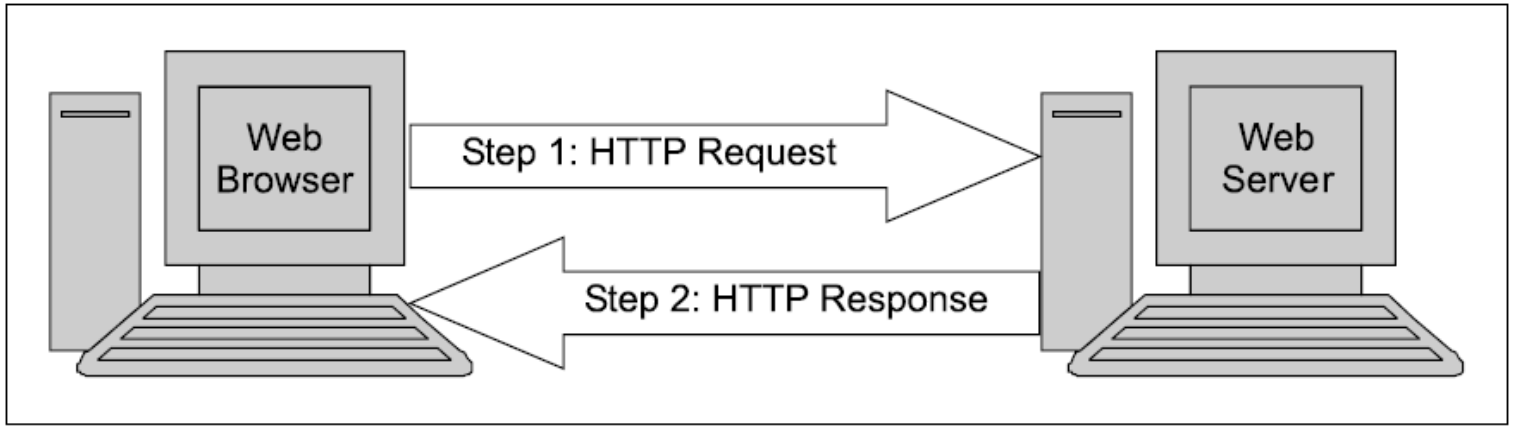
Detached Signature

```
<Original_document>
</Original_document>
<Signature>
</Signature>
```

Internet Security Protocols

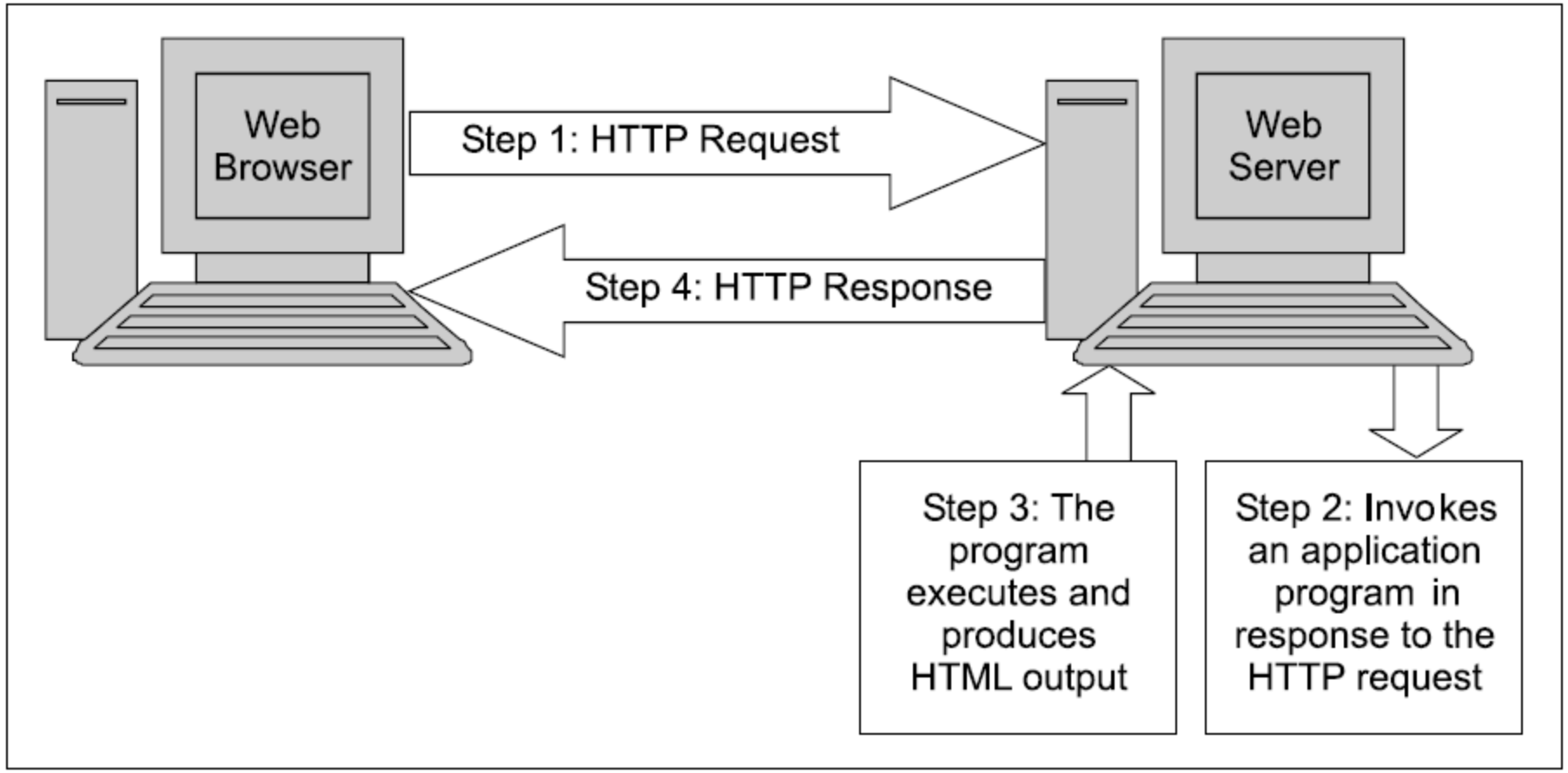
Basic Concepts

Static Web pages: Static Web pages are very simple. It is written in languages such as HTML, JavaScript, CSS, etc. For static web pages when a server receives a request for a web page, then the server sends the response to the client without doing any additional process. And these web pages are seen through a web browser. In static web pages, Pages will remain the same until someone changes it manually.



MKK

Dynamic Web Pages: Dynamic Web Pages are written in languages such as CGI, AJAX, ASP, ASP.NET, etc. In dynamic web pages, the Content of pages is different for different visitors. It takes more time to load than the static web page. Dynamic web pages are used where the information is changed frequently, for example, stock prices, weather information, etc.



MKK

Secure Socket Layer

- Secure Sockets Layer (SSL) is a standard technique for transmitting documents securely across a network. SSL technology, created by Netscape, establishes a secure connection between a Web server and a browser, ensuring private and secure data transmission. SSL communicates using the Transport Control Protocol (TCP).
- The term "socket" in SSL refers to the method of sending data via a network between a client and a server.
- A Web server requires an SSL certificate to establish a secure SSL connection while using SSL for safe Internet transactions. SSL encrypts network connection segments atop the transport layer, a network connection component above the program layer.
- SSL is based on an asymmetric cryptographic process in which a Web browser generates both a public and a private (secret) key. A certificate signing request is a data file that contains the public key (CSR). Only the recipient receives the private key.

Secure Socket Layer Protocols:

- SSL record protocol
- Handshake protocol
- Change-cipher spec protocol
- Alert protocol

MKK

SSL Record Protocol:

- SSL Record provides two services to SSL connection.
- Confidentiality
- Message Integrity
- Handshake Protocol:

Handshake Protocol:

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other.

Change-cipher Protocol:

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After the handshake protocol, the Pending state is converted into the current

Alert Protocol:

This protocol is used to convey SSL-related alerts to the peer entity. state.

SHTTP (SECURE HYPER TEXT TRANSFER PROTOCOL)

- The Secure Hyper Text Transfer Protocol (SHTTP) is a set of security mechanisms defined for protecting the Internet traffic. This includes the data-entry forms and Internet-based transactions.
- Note that an HTTP request sent by using SSL is identified as HTTPS (e.g. HTTPs://www.yahoo.com), whereas this is SHTTP (e.g. sHTTP://www.yahoo.com). The services offered by SHTTP are quite similar to those of SSL. However, SSL has become highly successful—SHTTP has not.
- SHTTP works at the application layer, and is therefore, tightly coupled with HTTP, unlike SSL
- SHTTP supports both authentication and encryption of HTTP traffic between the client and the server.
- The key difference between SSL and SHTTP is that SHTTP works at the level of individual messages. It can encrypt and sign individual messages.
- SHTTP is very rarely used

Secure Electronic Transaction

Secure Electronic Transaction or SET is a system that ensures the security and integrity of electronic transactions done using credit cards in a scenario. SET is not some system that enables payment but it is a security protocol applied to those payments. It uses different encryption and hashing techniques to secure payments over the internet done through credit cards. The SET protocol was supported in development by major organizations like Visa, Mastercard, and Microsoft which provided its Secure Transaction Technology (STT), and Netscape which provided the technology of Secure Socket Layer (SSL).

Participants in SET: In the general scenario of online transactions, SET includes similar participants:

- Cardholder – customer
- Issuer – customer financial institution
- Merchant
- Acquirer – Merchant financial
- Certificate authority

SET functionalities:

- Provide Authentication
  - Merchant Authentication – To prevent theft, SET allows customers to check previous relationships between merchants and financial institutions. Standard X.509V3 certificates are used for this verification.
  - Customer / Cardholder Authentication – SET checks if the use of a credit card is done by an authorized user or not using X.509V3 certificates.
- Provide Message Confidentiality
  - Confidentiality refers to preventing unintended people from reading the message being transferred. SET implements confidentiality by using encryption techniques. Traditionally DES is used for encryption purposes.
- Provide Message Integrity
  - SET doesn't allow message modification with the help of signatures.

SSL versus SET

SSL (Secure Sockets Layer):

- SSL is a cryptographic protocol that provides secure communication over a network. It operates at the transport layer and is typically used to secure connections between a client (such as a web browser) and a server (such as a web server) using encryption and authentication. SSL has been deprecated and replaced by its successor, Transport Layer Security (TLS). TLS is an updated version of the protocol and is commonly used to secure internet communications.

Key Features of SSL/TLS:

- *Encryption:* SSL/TLS protocols provide encryption to protect data in transit, preventing eavesdropping and unauthorized access.
- *Authentication:* SSL/TLS supports server authentication, ensuring the client is communicating with the intended server.
- *Integrity:* SSL/TLS ensures the integrity of data transmitted between client and server, preventing tampering and unauthorized modifications.

SET (Secure Electronic Transaction):

- SET is a cryptographic protocol specifically designed for securing online credit card transactions. It operates at the application layer and provides end-to-end security for online payment processing. SET was primarily developed by Visa and Mastercard and aimed to address security concerns associated with online transactions.

Key Features of SET:

- *End-to-End Security:* SET provides security throughout the entire payment process, including cardholder authentication, merchant verification, and secure communication between all parties involved (cardholder, merchant, and payment gateway).
- *Digital Certificates:* SET utilizes digital certificates to authenticate the various entities involved in a transaction, including the cardholder, merchant, and payment gateway.
- *Confidentiality and Integrity:* SET ensures the confidentiality and integrity of transaction data through encryption and digital signatures.
- *Non-Repudiation*: SET incorporates mechanisms to prevent any party from denying their involvement in a transaction, providing non-repudiation.

3-D Secure Protocol

- The 3-D Secure Protocol is an authentication protocol designed to add an extra layer of security to online credit and debit card transactions. It is commonly used for e-commerce transactions and is implemented by major card networks such as Visa (Verified by Visa), Mastercard (Mastercard SecureCode), and American Express (American Express SafeKey).

Process Flow:

- The typical process flow of a 3-D Secure transaction involves the following steps:
- a. During the checkout process, the cardholder selects their payment method and provides their card details.
- b. The merchant's website sends the transaction details to the acquiring bank or payment gateway.
- c. The acquiring bank or payment gateway checks if the card is enrolled in the 3-D Secure program.
- d. If enrolled, the cardholder is redirected to their card issuer's authentication page.
- e. The cardholder is prompted to provide the necessary authentication information, such as a one-time password (OTP), biometric data, or other authentication factors.
- f. The card issuer validates the authentication information and sends a response back to the merchant. g. Based on the authentication result, the merchant proceeds with the transaction or takes appropriate action.

Crypto Currency and Bitcoin

- Cryptocurrency and Bitcoin are closely intertwined with cryptography. Cryptocurrencies utilize cryptographic techniques to secure transactions, verify the integrity of the data, and ensure the privacy and security of users. Bitcoin, being the first and most well-known cryptocurrency, relies heavily on cryptography for its operation.

key aspects of cryptography in the context of cryptocurrency and Bitcoin:

- Secure Transactions: Cryptocurrencies use cryptographic algorithms to secure transactions.
- Wallet Security: Cryptocurrency wallets, which store the private keys necessary to access and transfer funds, heavily rely on cryptography. Private keys are generated using cryptographic algorithms and are used to sign transactions, providing ownership and control of the associated funds.
- Address Generation: Cryptocurrency addresses, which are used to receive funds, are derived from public keys using cryptographic hashing algorithms.
- Mining and Proof of Work: Bitcoin and many other cryptocurrencies rely on a consensus mechanism called Proof of Work (PoW) for adding new blocks to the blockchain. Miners compete to solve computationally intensive cryptographic puzzles, known as hashing algorithms
- Cryptographic Hash Functions: Cryptocurrencies use cryptographic hash functions extensively. Hash functions play a vital role in verifying the integrity of data and ensuring that blocks of transactions are linked together securely. Bitcoin, for example, uses SHA-256 as its hashing algorithm
- Cryptographic Key Management: Cryptocurrency systems require secure key management practices. Users must generate and securely store their private keys, as losing the keys can result in permanent loss of access to funds.

That's all about

UNIT 4

MKK