19CSE311 - COMPUTER SECURITY
UNIT-3- PART 2
Transport Level Security

March 15, 2022

Dr.Remya S/ Mr. Sarath R
Assistant Professor
Department of Computer Science
AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY
School of Engineering

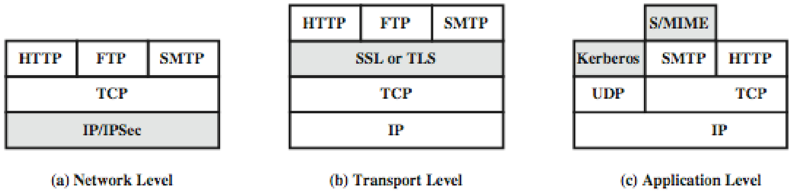
Agenda

- Web security
- Secure Socket Layer(SSL)
- Secure Electronic Transaction(SET)

Web Security

- Web now widely used by business, government, individuals
- but Internet Web are vulnerable
- have a variety of threats
  - Integrity
  - Confidentiality
  - Denial of service
  - Authentication
- need added security mechanisms

Web Traffic Security Approaches



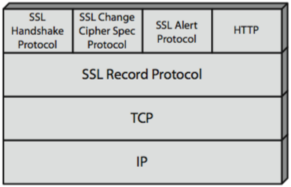
- A number of different approaches for providing Web security are possible.
- The various approaches that have been considered are similar in the services they provide and, to some extent, in the mechanisms that they use, but they differ with respect to their scope of applicability and their relative location within the TCP/IP protocol stack.
- One way to provide Web security is to use IP Security(Fig.a)

- The advantage of using IPSec is that it is transparent to end users and applications and provides a general-purpose solution.
- Further, IPSec includes a filtering capability ,so only selected traffic need incur the IPSec processing overhead.
- Another relatively general-purpose solution is to implement security just above TCP (Figure b).
- Example: Secure Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS).
- At this level, there are two implementation choices. For full generality, SSL (or TLS) could be provided as part of the underlying protocol suite and therefore be transparent to applications. Alternatively, SSL can be embedded in specific packages, e.g. both the Netscape and Microsoft Explorer browsers come with SSL, most Web servers have implemented this.
- Application-specific security services are embedded within the particular application. Figure c shows examples of this architecture.
- The advantage of this approach is that the service can be tailored to the specific needs of a given application.

SSL (Secure Socket Layer)

- Transport layer security service
- Originally developed by Netscape
- Version 3 designed with public input
- Subsequently became Internet standard known as TLS (Transport Layer Security)
- Uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

SSL Architecture



- The SSL Record Protocol provides basic security services to various higher-layer protocols.
- Three higher-layer protocols are also defined as part of SSL: The Handshake Protocol, Change Cipher Spec Protocol, and Alert Protocol. These SSL-specific protocols are used in the management of SSL exchanges.
- In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL.

1. SSL connection
    - A connection is a transport layer model that provides a suitable type of service.
    - A transient, peer-to-peer, communications link
    - Associated with 1 SSL session
2. SSL session
    - An association between client server
    - Created by the Handshake Protocol
    - Define a set of cryptographic parameters
    - May be shared by multiple SSL connections
    - Between any pair of parties (applications such as HTTP on client and server), there may be multiple secure connections. There are a number of states associated with each session. Once a session is established, there is a current operating state for both read and write (i.e., receive and send).
    - In addition, during the Handshake Protocol, pending read and write states are created.
    - Upon successful conclusion of the Handshake Protocol, the pending states become the current states.

A session state is defined by the following parameters

1. Session identifier: An arbitrary byte sequence chosen by the server to identify an active or resumable session state.
2. Peer certificate: An X509.v3 certificate of the peer. This element of the state may be null.
3. Compression method: The algorithm used to compress data prior to encryption.
4. Cipher spec: Specifies the bulk data encryption algorithm (such as null,AES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation.It also defines cryptographic attributes such as the hash size.
5. Master secret: 48-byte secret shared between the client and server.
6. Is resumable: A flag indicating whether the session can be used to initiate new connections.

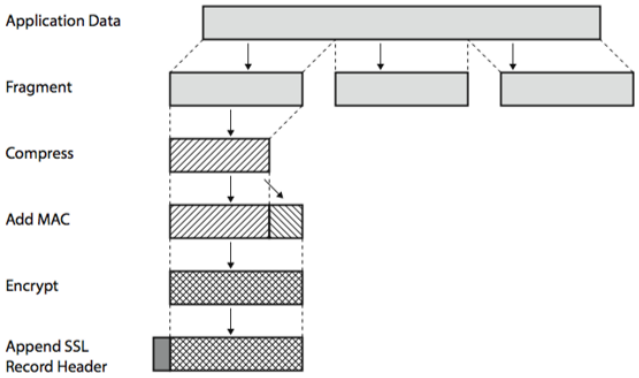
A connection state is defined by the following parameters

1. Server and client random: Byte sequences that are chosen by the server and client for each connection.
2. Server write MAC secret: The secret key used in MAC operations on data sent by the server.
3. Client write MAC secret: The secret key used in MAC operations on data sent by the client.
4. Server write key: The secret encryption key for data encrypted by the server and decrypted by the client.
5. Client write key: The symmetric encryption key for data encrypted by the client and decrypted by the server.

6. Initialization vectors: When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key.This field is first initialized by the SSL Handshake Protocol. Thereafter, the final ciphertext block from each record is preserved for use as the IV with the following record.
7. Sequence numbers: Each party maintains separate sequence numbers for transmitted and received messages for each connection.When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed $2^{64}-1$

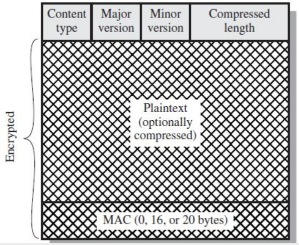
SSL Record Protocol Services

- Confidentiality
- Message Integrity



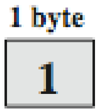
SSL-Record-Header

- Content Type (8 bits): The higher-layer protocol used to process the enclosed fragment.
- Major Version (8 bits): Indicates major version of SSL in use. For SSLv3, the value is 3.
- Minor Version (8 bits): Indicates minor version in use. For SSLv3, the value is 0.
- Compressed Length (16 bits): The length in bytes of the plaintext fragment (or compressed fragment if compression is used).The maximum value is $2^{14} + 2048$



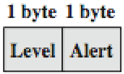
SSL Change Cipher Spec Protocol

- One of 3 SSL specific protocols which use the SSL Record protocol
- A single message which consists of a single byte with the value 1.
- The sole purpose of this message is to cause the pending state to be copied into the current state
- causes pending state to become current and hence updating the cipher suite in use



(a) Change Cipher Spec Protocol

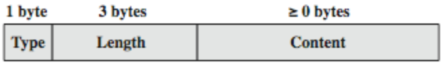
SSL Alert Protocol



(b) Alert Protocol

- Conveys SSL-related alerts to peer entity
- Severity
  1. Warning(1) or fatal(2): If the level is fatal, SSL immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established
- Specific alert
  1. Fatal: unexpected message, bad record mac, Decompression failure, handshake failure, illegal parameter
  2. Warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- Compressed encrypted like all SSL data

SSL Handshake Protocol

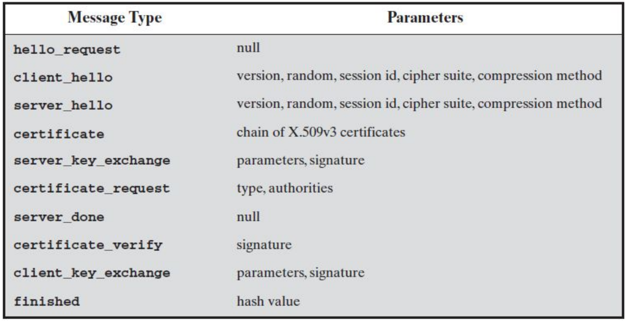


(c) Handshake Protocol

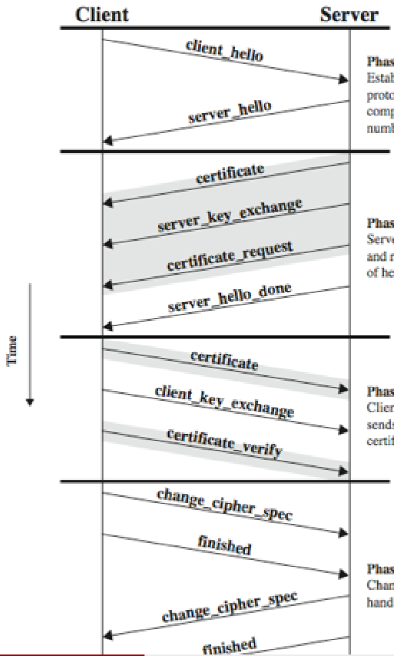
- allows server & client to:
  1. authenticate each other
  2. to negotiate encryption & MAC algorithms
  3. to negotiate cryptographic keys to be used
- comprises a series of messages in phases
  1. Establish Security Capabilities
  2. Server Authentication and Key Exchange
  3. Client Authentication and Key Exchange
  4. Finish

Type indicates one of 10 messages

Table 16.2 SSL Handshake Protocol Message Types



Message Type	Parameters
`hello_request`	null
`client_hello`	version, random, session id, cipher suite, compression method
`server_hello`	version, random, session id, cipher suite, compression method
`certificate`	chain of X.509v3 certificates
`server_key_exchange`	parameters, signature
`certificate_request`	type, authorities
`server_done`	null
`certificate_verify`	signature
`client_key_exchange`	parameters, signature
`finished`	hash value



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Phase 4
Change cipher suite and finish handshake protocol.

Secure Electronic Transactions(SET)

- open encryption security specification
- to protect Internet credit card transactions
- developed in 1996 by Mastercard, Visa etc
- not a payment system
- rather a set of security protocols formats
    - secure communications amongst parties
    - trust from use of X.509v3 certificates
    - privacy by restricted info to those who need it

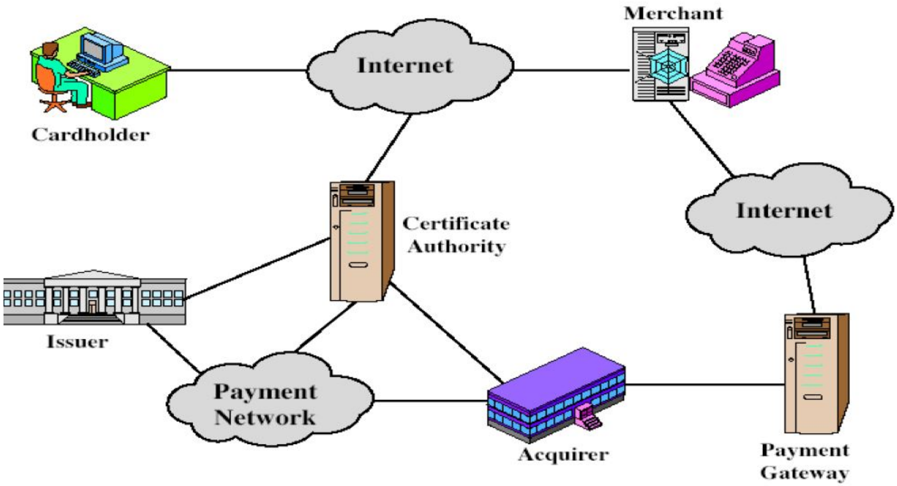
SET Requirements- secure payment processing with credit cards over the Internet and other networks

- Provide confidentiality of payment and ordering information
- Ensure the integrity of all transmitted data
- Provide authentication that a cardholder is a legitimate user of a credit card account
- Provide authentication that a merchant can accept credit card transactions through its relationship with a financial institution
- Ensure the use of the best security practices and system design techniques to protect all legitimate parties in an electronic commerce transaction
- Create a protocol that neither depends on transport security mechanisms nor prevents their use
- Facilitate and encourage inter operability among software and network providers

Key Features-SET

- Confidentiality of information: Cardholder account and payment information is secured as it travels across the network
- Integrity of data: Payment information sent from cardholders to merchants includes order information, personal data, and payment instructions. SET guarantees that these message contents are not altered in transit
- Cardholder account authentication: SET enables merchants to verify that a cardholder is a legitimate user of a valid card account number. SET uses X.509v3 digital certificates with RSA signatures for this purpose.
- Merchant authentication: SET enables cardholders to verify that a merchant has a relationship with a financial institution allowing it to accept payment cards. SET uses X.509v3 digital certificates with RSA signatures for this purpose

SET Components



SET Transaction-Steps

1. Customer opens account:
   - The customer obtains a credit card account, such as MasterCard or Visa, with a bank that supports electronic payment and SET.
2. Customer receives a certificate:
   - After suitable verification of identity, the customer receives an X.509v3 digital certificate, which is signed by the bank.
   - The certificate verifies theccustomer's RSA public key and its expiration date.
   - It also establishes a relationship, guaranteed by the bank, between the customer's key pair and his or her credit card.
3. Merchants have their own certificates:
   - A merchant who accepts a certain brand of card must be in possession of two certificates for two public keys owned by the merchant: one for signing messages, and one for key exchange.
   - The merchant also needs a copy of the payment gateway's public-key certificate

4. Customer places an order:
    - This is a process that may involve the customer first browsing through the merchant's Web site to select items and determine the price.
    - The customer then sends a list of the items to be purchased to the merchant, who returns an order form containing the list of items, their price, a total price, and an order number.
5. Merchant is verified:
    - In addition to the order form, the merchant sends a copy of its certificate, so that the customer can verify that he or she is dealing with a valid store
6. Order and payment are sent
    - The customer sends both order and payment information to the merchant, along with the customer's certificate.
    - The order confirms the purchase of the items in the order form.
    - The payment contains credit card details. The payment information is encrypted in such a way that it cannot be read by the merchant.
    - The customer's certificate enables the merchant to verify the customer

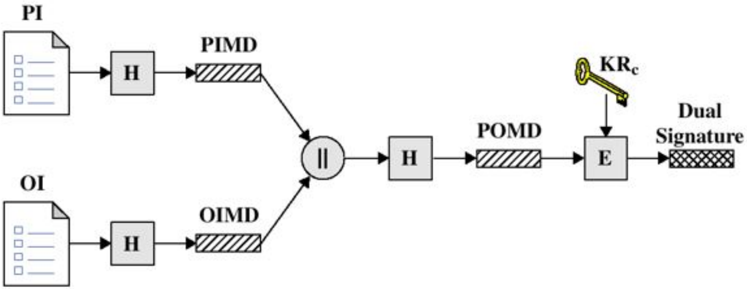
7. Merchant requests payment authorization
    - The merchant sends the payment information to the payment gateway, requesting authorization that the customer's available credit is sufficient for this purchase.
8. Merchant confirms order:
    - The merchant sends confirmation of the order to the customer
9. Merchant provides goods or service
    - The merchant ships the goods or provides the service to the customer
10. Merchant requests payment:
    - This request is sent to the payment gateway, which handles all of the payment processing.

Dual Signature

- The purpose of the dual signature is to link two messages that are intended for two different recipients.
- In this case, the customer wants to send the order information (OI) to the merchant and the payment information (PI) to the bank.
- The merchant does not need to know the customer's credit card number, and the bank does not need to know the details of the customer's order
- use a dual signature for this- signed concatenated hashes of OI & PI

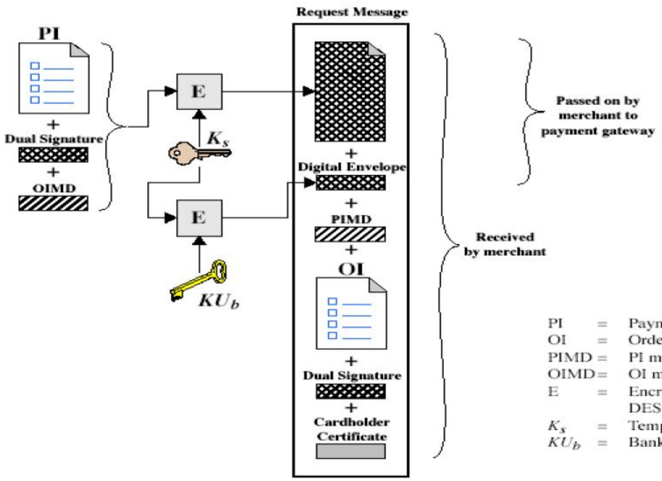
Construction of Dual Signature

$$\boxed{DS = E_{KR_c}[H(H(PI) \| H(OI))]}$$



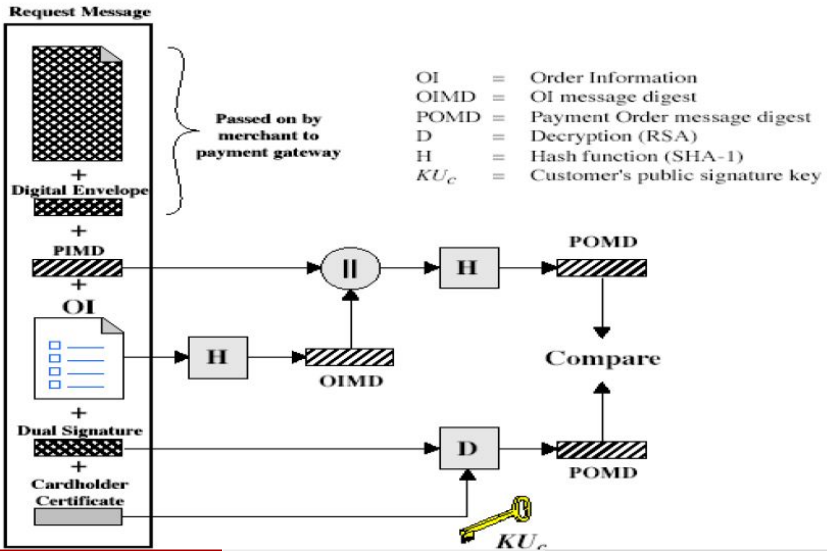
PI = Payment Information		PIMD	= PI message digest
OI = Order Information		OIMD	= OI message digest
H = Hash function (SHA-1)		POMD	= Payment Order message digest
|| = Concatenation		E	= Encryption (RSA)
		KR_c	= Customer's private signature key

Purchase Request-Customer



PI	=	Payment Information
OI	=	Order Information
PIMD	=	PI message digest
OIMD	=	OI message digest
E	=	Encryption (RSA for asymmetric; DES for symmetric)
K_s	=	Temporary symmetric key
KU_b	=	Bank's public key-exchange key

Purchase Request Merchant



1. Verifies cardholder certificates using CA sigs
2. Verifies dual signature using customer's public signature key to ensure order has not been tampered with in transit that it was signed using cardholder's private signature key
3. Processes order and forwards the payment information to the payment gateway for authorization
4. Sends a purchase response to cardholder

Payment Gateway Authorization

1. Verifies all certificates
2. Decrypts digital envelope of authorization block to obtain symmetric key then decrypts authorization block
3. Verifies merchant's signature on authorization block
4. Decrypts digital envelope of payment block to obtain symmetric key then decrypts payment block
5. Verifies dual signature on payment block
6. Verifies that transaction ID received from merchant matches that in PI received (indirectly) from customer
7. Requests receives an authorization from issuer
8. Sends authorization response back to merchant

Payment Capture

- Merchant sends payment gateway a payment capture request
- Gateway checks request
- Then causes funds to be transferred to merchants account
- Notifies merchant using capture response