Security Layer in Communication Software

Madjid S. Mousavi
Dept of Electrical Engineering
Stevens Institute of Technology

Security Layer

- Security on the Internet has been playing an ever-increasing role of importance in the past few years
- It must provide data privacy for all the data transmitted data between parties and prevent programs that could view the network traffic from reading the sensitive data.

SSL/IPSec

- Security can be implemented at different levels:
  - The security may be implemented at a layer between TCP/IP and application protocols, such as HTTP, LDAP, FTP, and Telnet (SSL).
  - The security May be implemented on top of the IP layer (IPSec)

Security Achieved by the Secure Sockets Layer (SSL)

- *Confidentiality*

  Encrypt data being sent between client and server, so that passive wiretappers cannot read sensitive data.

- *Integrity Protection*

  Protect against modification of messages by an active wiretapper.

- *Authentication*

  Verify that a peer is who they claim to be. Servers are usually authenticated, and clients may be authenticated if requested by servers.

SSL

- Symmetric key algorithms
  - use a **single secret key**, which must be shared and kept private by both the sender and the receiver, for both encryption and decryption. To use a symmetric encryption scheme, the sender and receiver must securely share a key in advance.

- Asymmetric key algorithms
  - A key used to encrypt a message is not the same as the key used to decrypt it.
  - The **public encryption key** is widely distributed, while the **private decryption key**. private decrypting-key is known only to its proprietor.
  - The keys are related mathematically, but the parameters are chosen so that calculating the private key from the public key is either impossible or prohibitively expensive.

TCP/IP Protocol Stack With SSL

TCP/IP Layer	Protocol
Application Layer	HTTP, IMAP, NNTP, Telnet, FTP, etc.
Secure Sockets Layer	SSL
Transport Layer	TCP
Internet Layer	IP

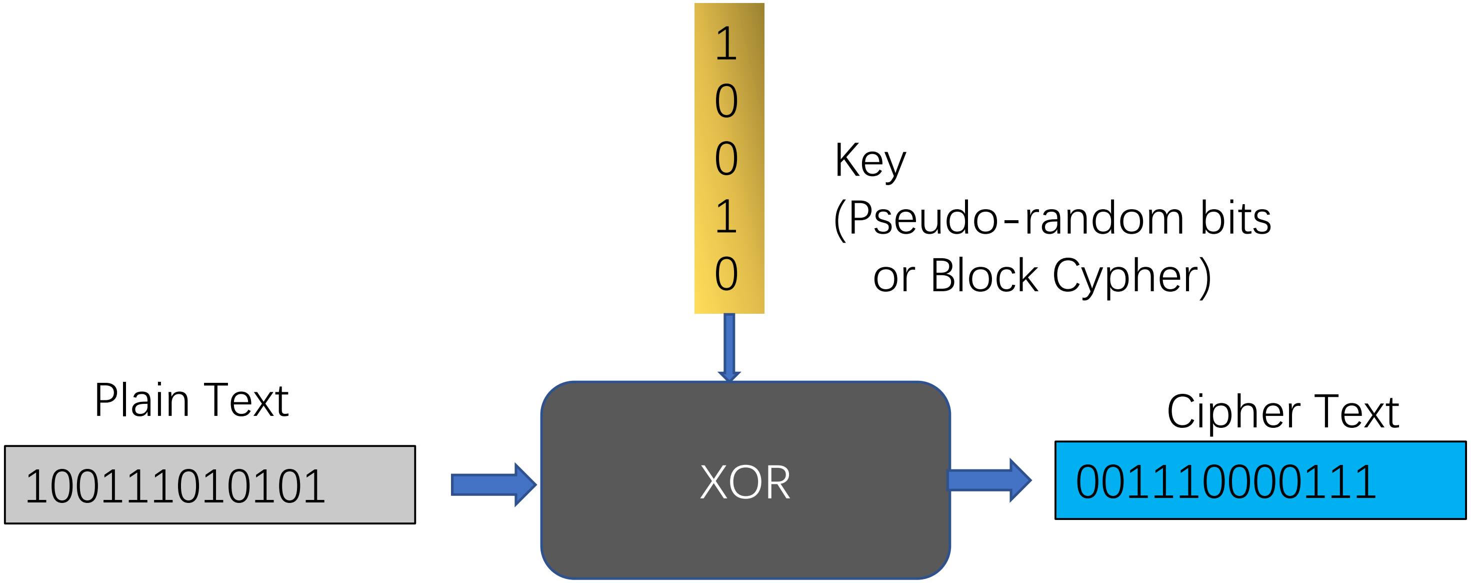
Cryptography

- Cryptography makes it difficult for an unauthorized third party to access and understand private communication between two parties.
- Private data can be made unintelligible to unauthorized parties through the process of encryption. *Encryption* uses complex algorithms to convert the original message, or *cleartext*, to an encoded message, called *ciphertext*. *Decryption* does the reverse.
- A *key* is a bit string that is used by the algorithms for encryption or decryption.
- Two types of Cryptography :
  - Symmetric (same key is used for both encryption and decryption)
  - Asymmetric (Different but mathematically related keys are used for encryption and decryption)

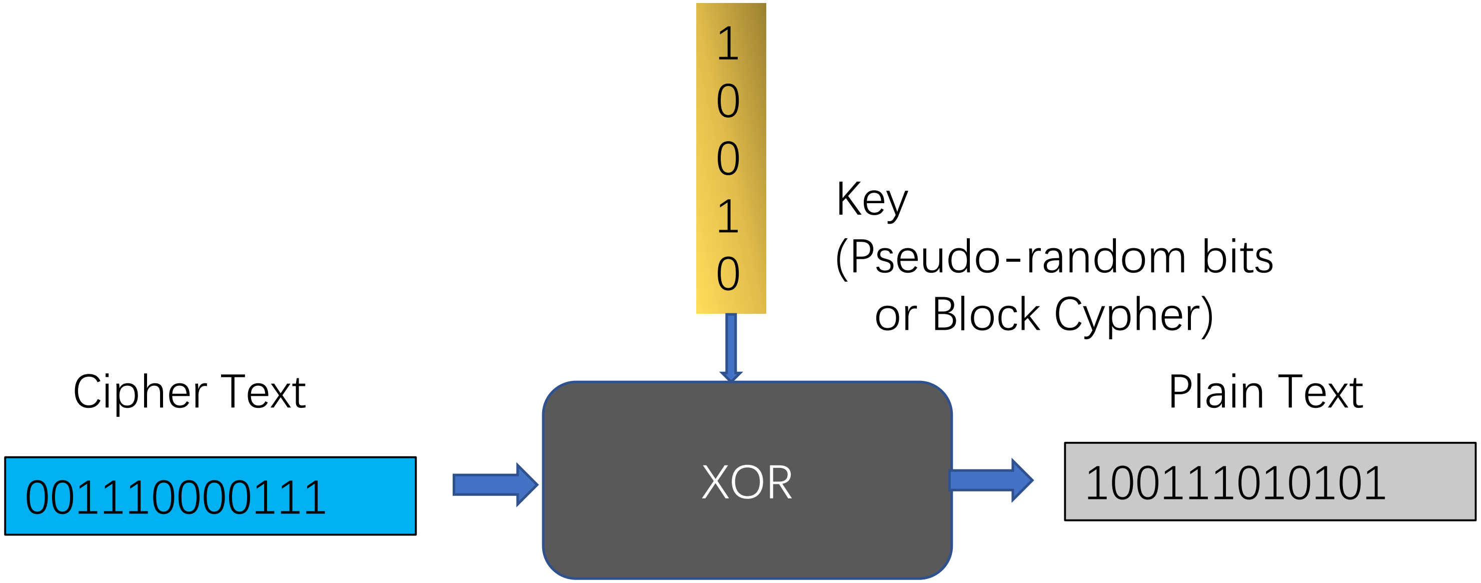
symmetric cryptography

- The same key is used for both encryption and decryption.
  - Stream Ciphers
    - Random stream with a known seed used for both encryption and decryption
  - Block Ciphers
    - Data Encryption Standard (DES) – 64 bits
      - Is no longer considered secure
    - Advanced Encryption Standard (AES)- 128, 192, 256

Symmetric Encryption



Symmetric Decryption



1
0
0
1
0

Key
(Pseudo-random bits
or Block Cypher)

Cipher Text

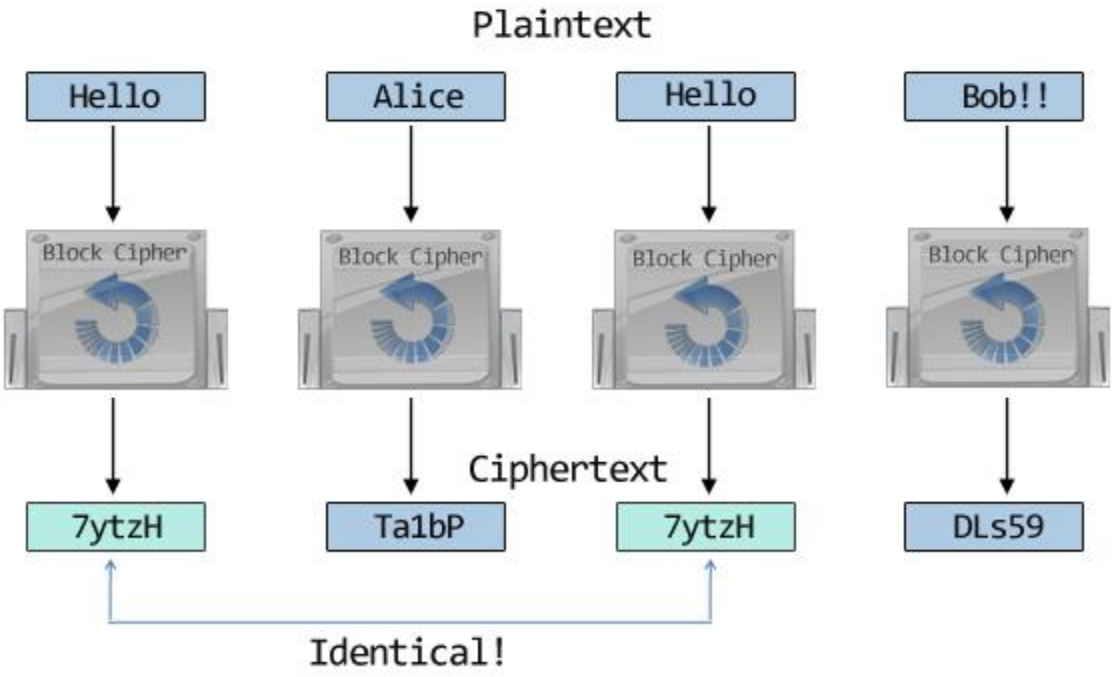
001110000111

XOR

Plain Text

100111010101

Symmetric Cryptographic Cipher



All block ciphers have an inherent problem. If two blocks of plaintext are identical, then the ciphertext for both will also be identical. Why is this a problem?

Cipher block chaining

- Each plaintext block is XORed with the previous ciphertext block before being encrypted. CBC requires an initialization vector to XOR with the first plaintext block. CBC mode is very popular and is considered an industry best practice.
- Examples of Block cipher
  - Data Encryption Standard (DES) – 64 bits
  - Advanced Encryption Standard (AES)- 128, 192, 256

Hash Functions

- A hash function is a reproducible method for turning a long message into a concise fixed-length piece of data, known as hash code.
- A hash is useful because it can serve as a ‘fingerprint’ that uniquely identifies the data from which it originated.

Hash Functions

- The hash function must be one-way.
  - Given a message, it is easy to compute its hash;
  - Given a hash, it is very difficult to find a message that produces this hash.
- The hash function must produce uncorrelated hashes.
  - Two similar messages must produce two unrelated hashes.
  - The good hash function must *minimize* collisions. A collision is when two distinct messages both produce the same hash.

Hash Functions

- Widely used hash functions are
  - Message Digest 5 (MD5)
    - A widely-used hash function that produces 128-bit hashes. Collisions have been found with MD5 and thus this algorithm is no longer considered secure.
  - Secure Hash Algorithm (SHA).
    - SHA-1
      - Another popular hash function that produces 160-bit hashes. Collisions are expected to be found soon with SHA-1 thus its use is no longer recommended.
    - SHA-2
      - A collection of related hash functions that differ mainly in the size of their hashes. Have not undergone as much public scrutiny as SHA-1 or MD5, they are nonetheless considered secure by the industry.

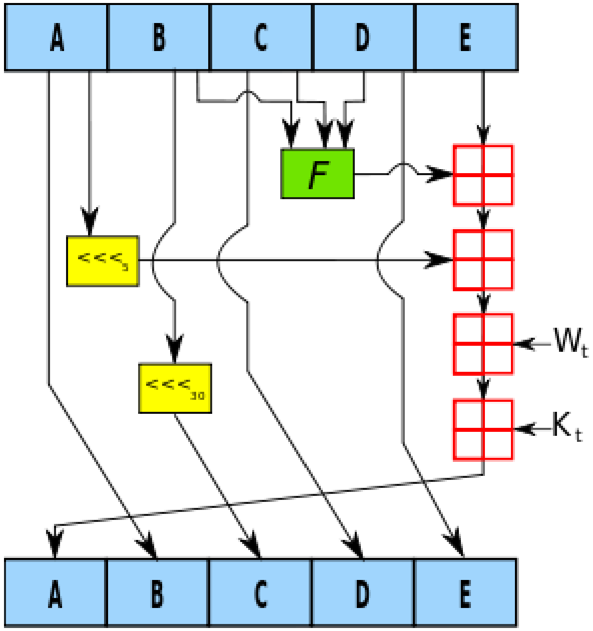
Cryptographic Hash Functions

Q: How can we prevent Charlie from tampering with data that Alice sends to Bob?

A: Make any change in the data detectable.

- A cryptographic hash function is like a checksum.
  - A one way transformation
  - A cryptographic hash function generates, a small string of bits, known as a hash, from a message. Any slight change to the message should make a change in the resulting hash.

Cryptographic Hash Functions



One iteration within the SHA-1 compression function:
A, B, C, D and E are 32-bit words of the state;
F is a nonlinear function that varies;
$_n$ denotes a left bit rotation by *n* places;
n varies for each operation;
W_t is the expanded message word of round t;
K_t is the round constant of round t;
denotes addition modulo 2^{32}.

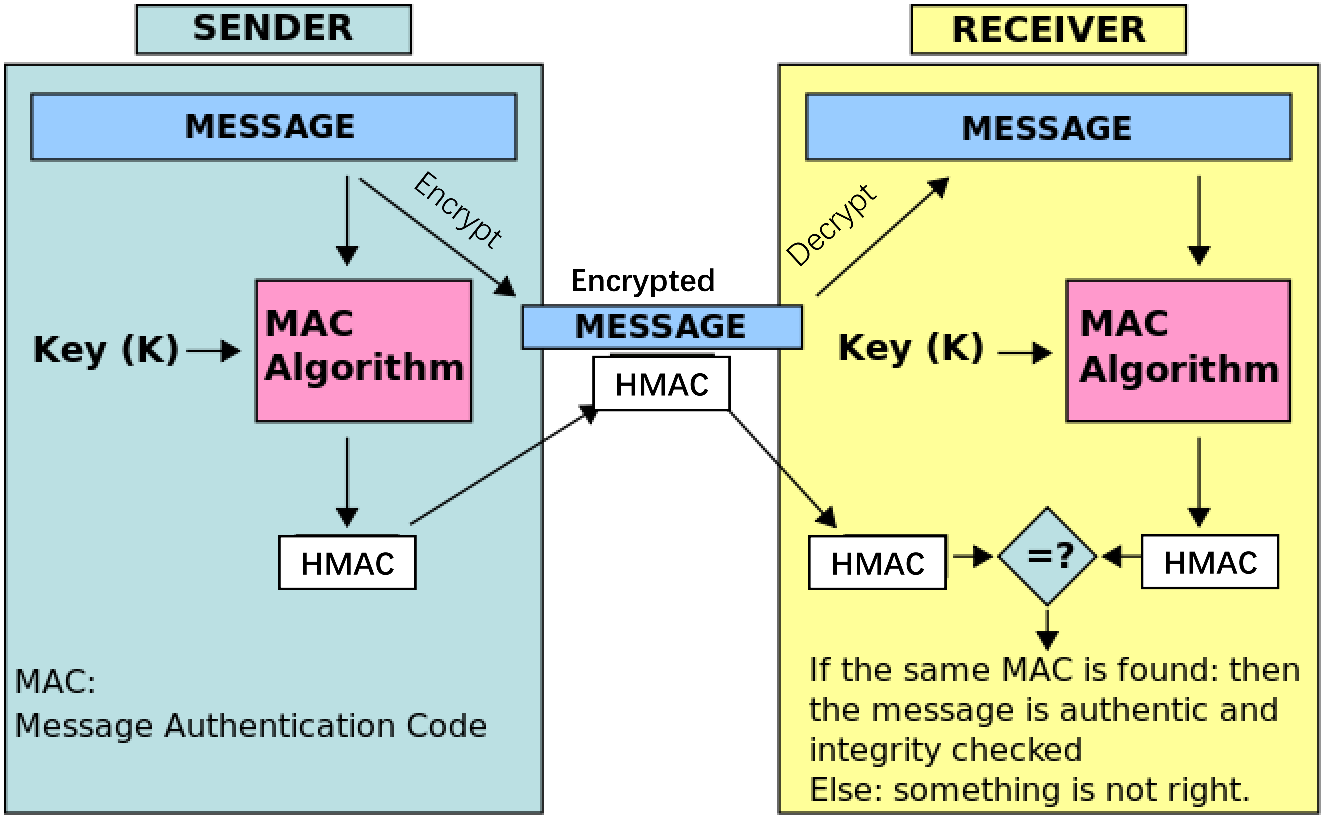
Message Authentication Code

- A message authentication code (MAC) is like a cryptographic hash, but it uses a secret key (symmetric cipher).
- Including a secret key with the data processed by a cryptographic hash produces a hash called an HMAC (hashed message authentication code).
- Shared secret key is also used to encrypt (for sender) and decrypt (for receiver)
- A secure network is needed to share the secret key

Message Authentication Code

- Here’s how we prevent Charlie from tampering with data that Alice sends to Bob.
  - Alice calculates an HMAC for her message and append the HMAC to her original message. She encrypts the message plus the HMAC using a secret key she shares with Bob.
  - Bob decrypts the message and recalculates the HMAC. If his HMAC differs from the one Alice sent then the message was modified in transit.

Message Authentication Code

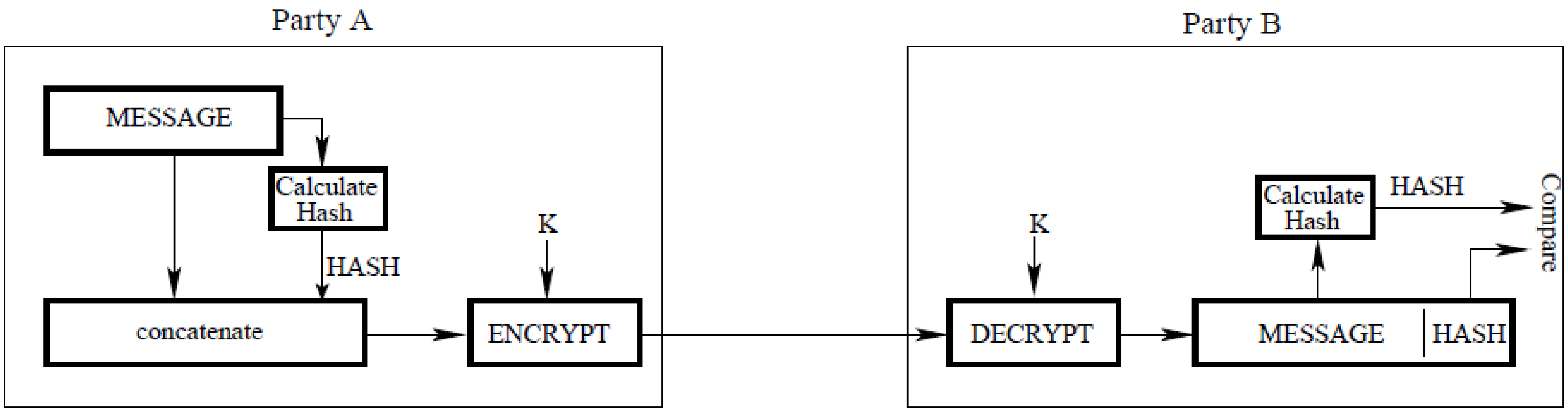


MAC:
Message Authentication Code

If the same MAC is found: then the message is authentic and integrity checked
Else: something is not right.

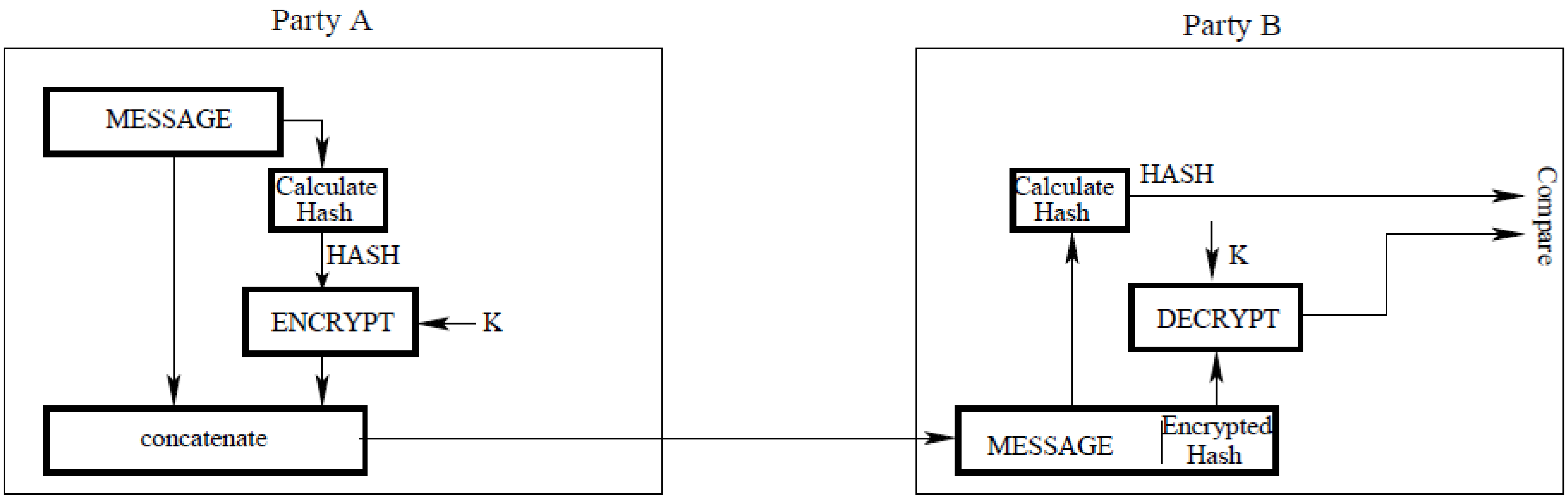
HashFn(message + shared key) = HMAC

Message Authentication Code



(a)

Message Authentication Code

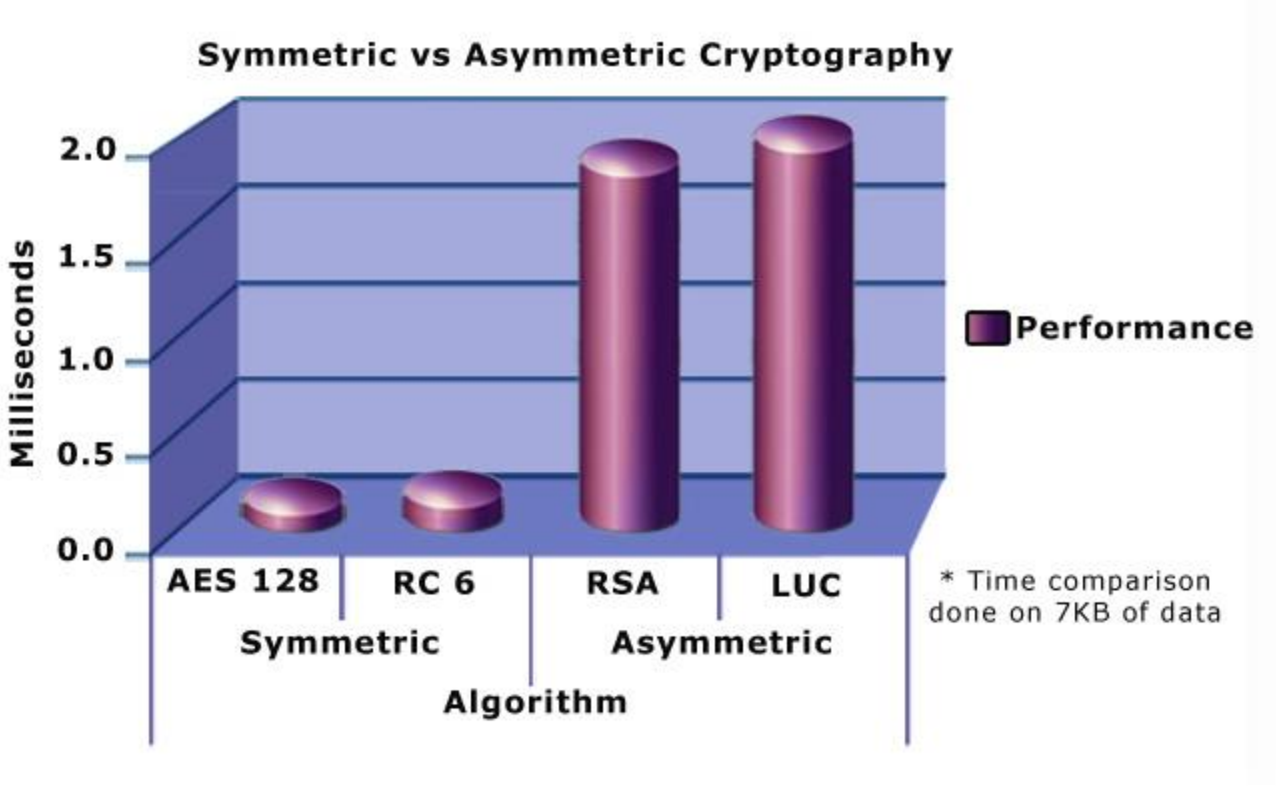


(b)

Symmetric and Asymmetric usage

- Use symmetric ciphers for message encryption.
- Use HMAC (Hash Message Authentication Code) to provide tamper detection.
- Use asymmetric ciphers for message encryption (Mathematically intensive and thus very slow).
- Apply asymmetric ciphers to encrypt cryptographic keys.
- Apply asymmetric cryptography to provide authentication.

Symmetric and Asymmetric performance



Asymetric Cryptography

- *Asymmetric cryptography*, or public-key cryptography, uses two separate keys.
    - One key is called the private key and is not shared with anyone.
    - The other key is called the public key and is publicly available to anyone who wants it.
    - A message encrypted with a public key can only be decrypted with the corresponding private key
    - A message encrypted with a private key can only be decrypted with the corresponding public key.

Mathematical Foundation of Asymmetric cryptography - RSA

- The idea behind it is very simple. Public-key cryptography is possible because there exist mathematical transformations that are very easy to compute, but whose inverse transformations are very difficult to compute.
- A common example is multiplying and factoring. It is very easy to algorithmically multiply two prime numbers together to form a large number, but it is much more difficult to factor that large number back down to the original two prime numbers.

Mathematical Foundation of Asymmetric cryptography - RSA

- Another example: Z=X^Y mod(N).
  - Given X, Y, and N, it is very easy to algorithmically compute Z; this is the exponentiation problem.
  - On the other hand, given Z, Y, and N, it is very difficult to compute X; this is the logarithm problem. However, if the prime factors of N are known, then a shortcut exists to easily compute X.
- So if N is specially chosen to be the product of two large primes, say P and Q, then only someone who knows P and Q will be able to calculate X given Z and Y.

Mathematical Foundation of Asymmetric cryptography - RSA

- <P, Q> A pair of prime numbers are chosen as a "*Private Key*"
  - N = P.Q,
  - P and Q are the prime factors of N and know as "Private Key"
- <N, Y> pair act as a "*Public Key*"
- Only if you can factor N, or can solve the algorithm problem, you can decipher the encrypted text. Both extremely difficult to do without the knowledge of prime factors (private key) P and Q.

SSL

- SSL has a notion of **client** and **server**. The client contacts the server and sends the first message.
- The first message causes the client and server to exchange a few messages to negotiate the encryption algorithm and to pick an encryption key for this connection.
- The server must have an SSL certificate (public key) and the private key associated with that certificate.
- This public key is sent to the client when it connects.
- The client will use the public key to encrypt a value and send it to the server.
- The server must then have the corresponding RSA private key so it can decode the client's message.

    - RSA is a public key cryptography algorithm (Ron Rivest, Adi Shamir and Leonard Adleman)

Encryption Algorithms

- Parties
  - Alice and Bob want to communicate.
  - Charlie, the unauthorized third party, is known as the attacker.
- Secret key
  - Alice and Bob agree on an algorithm, and have the same *secret key*, which they use to encrypt plaintext and decrypt cyphertext.
  - Well-known secret key symmetric cryptographic algorithms include the Data Encryption Standard (DES), triple-strength DES (3DES), Rivest Cipher 2 (RC2), Rivest Cipher 4 (RC4) and the Advanced Encryption Standard (AES).

Encryption Algorithms (cont.)

- Public key
  - Alice and Bob agree on an algorithm (RSA/DH), and Alice creates a pair of keys—public and private—and sends the public key to Bob and other people. Bob (or anyone else) encrypts with the public key, but only Alice can decrypt with the secret private key.
  - Well-known public key (asymmetric) algorithms include Rivest Shamir Adleman (RSA) and Diffie-Hellman (DH).
  - Because they require extensive computations, these algorithms run slowly. Therefore they're only used for encrypting small pieces of data, such as secret keys or signatures.

Shared Key exchange
RSA

- Create a secret key
  - Based on information generated by the client with a secure random number generator
- Use public keys to exchange the secret key (RSA)
  - The server sends its public key to the client
  - The client encrypts the secret key with the server's public key and sends it to the server
  - The server decrypts the secret key information with the server’s private key
- Encrypt and decrypt data with the secret key
  - The client and server use the negotiated algorithm

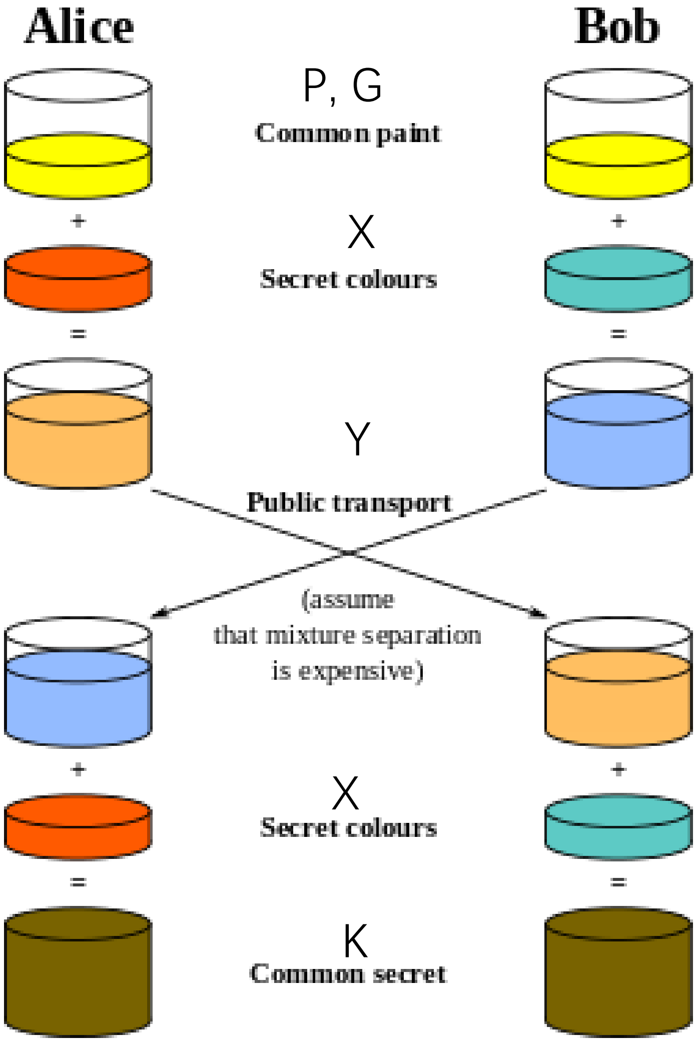
Shared key exchange
Diffie-Hellman (DH)

- Generate the same shared key on both ends (DH)
  - Calculate a public key based on two known prime numbers (base and modulus)
  - Send the public key to the peer
  - Use the peer public key to calculate shared key
  - Note that shared itself key is not exchanged over non-secure channel.

Mathematical Foundation of Asymmetric cryptography- DH

X_a is Alice’s DH private key
Y_a is Alice’s DH public key
P & G are two known primes
K_a is the DH shared key

X_b is Bob’s DH private key
Y_b is Bob’s DH public key
P & G are two known primes
K_b is the DH shared key



Y_a=(G^X_a) mod P

Y_b=(G^X_b) mod P

K_a=(Y_b^X_a) mod P

K_b=(Y_a^X_b) mod P

Mathematical Foundation of Asymmetric cryptography- DH

Prove that Ka= Kb:

```
Ka = (Yb^Xa) mod P

substitute for Yb = (G^Xb) mod P
   = (((G^Xb) mod P)^Xa) mod P
   = (G^XaXb) mod P
   = (((G^Xa) mod P)^Xb) mod P
But Ya = (G^Xa) mod P
so:
   = (Ya^Xb) mod P = Kb
```

Ka = Kb

How SSL Achieves *Integrity Protection*

- Client and server use their secret key, and an agreed-upon cryptographic hash function to attach an HMAC to each message sent.
- The receiver checks that each message has not been altered.

Public Keys and Authentication

Q: How does Alice prove to Bob that she is Alice?

A: Demonstrate that she has her private key.

- Protocol
  - Bob creates a random number, encrypts it with Alice's public key and sends it to Alice.
  - Alice decrypts the random number with her private key, and sends the random number to Bob, proving she's Alice.

Digital Signatures

Q: How does Alice prove to Bob that a message comes from her?

A: Demonstrate that she has her private key.

- Protocol
  - As before, Alice creates her public and private keys, and distributes her public key with her name attached.
  - Alice encrypts a *message (digital signature)* using her private key and sends the message to Bob.
  - If Bob can decrypt the data with Alice's public key, the message must have been encrypted by Alice with her private key, since only Alice has her private key.
  - This is called a *digital signature* (a Certificate encrypted by CA’s private key)

Public Keys and Authentication (cont.)

Q: How do we prevent Charlie from pretending to be Alice by circulating a public key named ‘Alice’?

A: By having someone we trust verify that Alice is Alice.

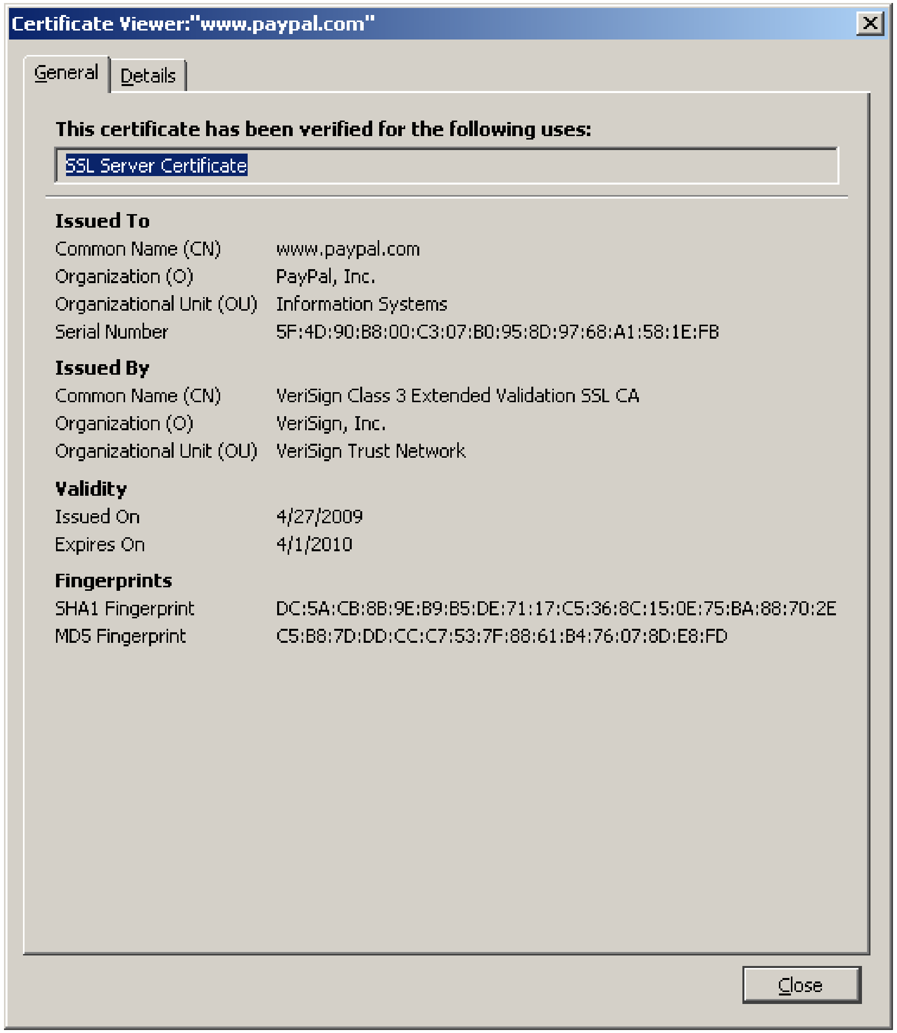
- Public Key Certificate
  - A digital ‘passport’ that is issued by a trusted organization and identifies the bearer.
  - A trusted organization is called a certificate authority (CA).
  - The CA digitally signs the certificate, thereby attesting to the validity of the certificate’s information.

Public Key Certificate

Contains the fields:

- Subject’s public key
- Subject
  - Information about the entity that the certificate represents.
- Issuer
  - The CA that issued the certificate. If a user trusts the CA that issues a certificate, and if the certificate is valid, the user can trust the certificate.
- Signature
  - The signature is created using the CA's private key and ensures the validity of the certificate.
- Period of validity
  - The certificate’s expiration date.

Certificates



Certificate Viewer:"www.paypal.com"

This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN)	www.paypal.com
Organization (O)	PayPal, Inc.
Organizational Unit (OU)	Information Systems
Serial Number	5F:4D:90:B8:00:C3:07:B0:95:8D:97:68:A1:58:1E:FB

Issued By

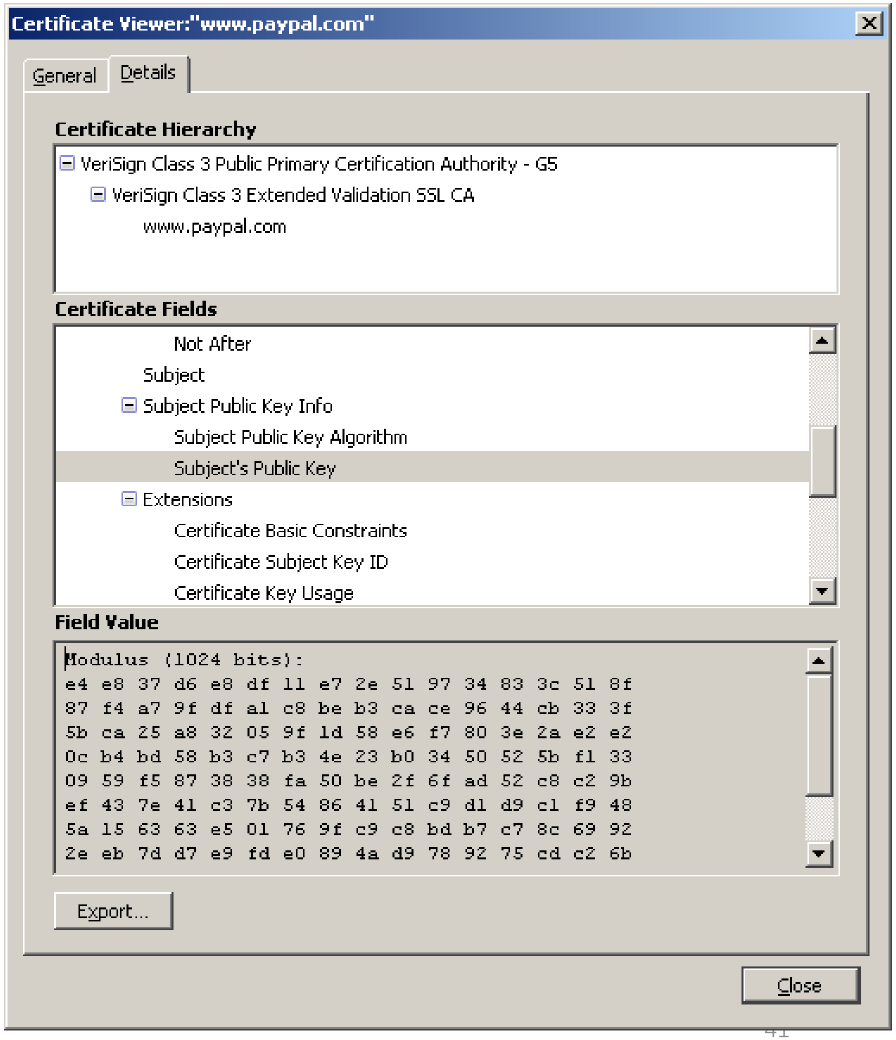
Common Name (CN)	VeriSign Class 3 Extended Validation SSL CA
Organization (O)	VeriSign, Inc.
Organizational Unit (OU)	VeriSign Trust Network

Validity

Issued On	4/27/2009
Expires On	4/1/2010

Fingerprints

SHA1 Fingerprint	DC:5A:CB:8B:9E:B9:B5:DE:71:17:C5:36:8C:15:0E:75:BA:88:70:2E
MD5 Fingerprint	C5:B8:7D:DD:CC:C7:53:7F:88:61:B4:76:07:8D:E8:FD



Certificate Viewer:"www.paypal.com"

Certificate Hierarchy

- VeriSign Class 3 Public Primary Certification Authority - G5
  - VeriSign Class 3 Extended Validation SSL CA
    - www.paypal.com

Certificate Fields

```
        Not After
      Subject
    ⊟ Subject Public Key Info
        Subject Public Key Algorithm
        Subject's Public Key
    ⊟ Extensions
        Certificate Basic Constraints
        Certificate Subject Key ID
        Certificate Key Usage
```

Field Value

```
Modulus (1024 bits):
e4 e8 37 d6 e8 df 11 e7 2e 51 97 34 83 3c 51 8f
87 f4 a7 9f df a1 c8 be b3 ca ce 96 44 cb 33 3f
5b ca 25 a8 32 05 9f 1d 58 e6 f7 80 3e 2a e2 e2
0c b4 bd 58 b3 c7 b3 4e 23 b0 34 50 52 5b f1 33
09 59 f5 87 38 38 fa 50 be 2f 6f ad 52 c8 c2 9b
ef 43 7e 41 c3 7b 54 86 41 51 c9 d1 d9 c1 f9 48
5a 15 63 63 e5 01 76 9f c9 c8 bd b7 c7 8c 69 92
2e eb 7d d7 e9 fd e0 89 4a d9 78 92 75 cd c2 6b
```

Authentication with a Public Key Certificate

Q: How is a public key certificate used to help Alice prove to Bob that she is Alice?

- Protocol
  - Bob obtains Alice’s public key certificate.
  - Bob also has a certificate for a trusted CA that supposedly signed Alice’s public key certificate.
  - Bob checks that the trusted CA signed Alice’s public key certificate by using the CA’s public key to decrypt the signature in Alice’s public key certificate.
  - Run the protocol for “Q: How does Alice prove to Bob that she is Alice?”

Certificate Chains

- Multiple certificates may be linked in a certificate chain.
  - The first certificate is that of the sender.
  - The next is the certificate of the entity that issued the sender’s certificate.
  - If there are more certificates in the chain, each is that of the authority that signed the previous certificate.
  - The final certificate in the chain is the certificate for a root CA, a certificate authority that is widely trusted.
  - Well-known public CAs include VeriSign, Entrust, and GTE CyberTrust.

How SSL Achieves *Authentication*

- Protocol
  - If the client wants to authenticate the server then they follow the protocol in “Authentication with a Public Key Certificate” with the client acting as Bob.
  - If the server wants to authenticate the client then they follow the protocol in “Authentication with a Public Key Certificate” with the server acting as Bob.

How SSL Works - Summary

1. *Handshake*
   - a negotiation process that creates or rejoins a *session*
2. If (Handshake succeeds) then

      Encrypted data can be exchanged

   Else

      The connection is aborted

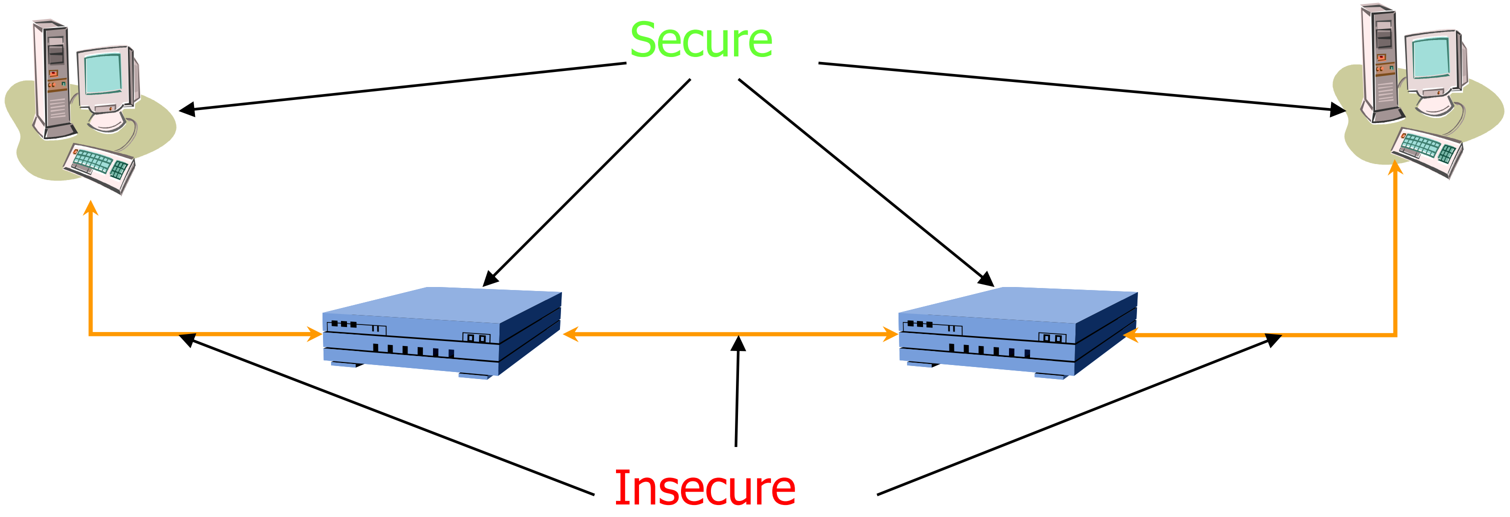
How SSL Works - Summary the *Handshake*

- Negotiate the cipher suite
- Authenticate identities (optional)
- Exchange secret key

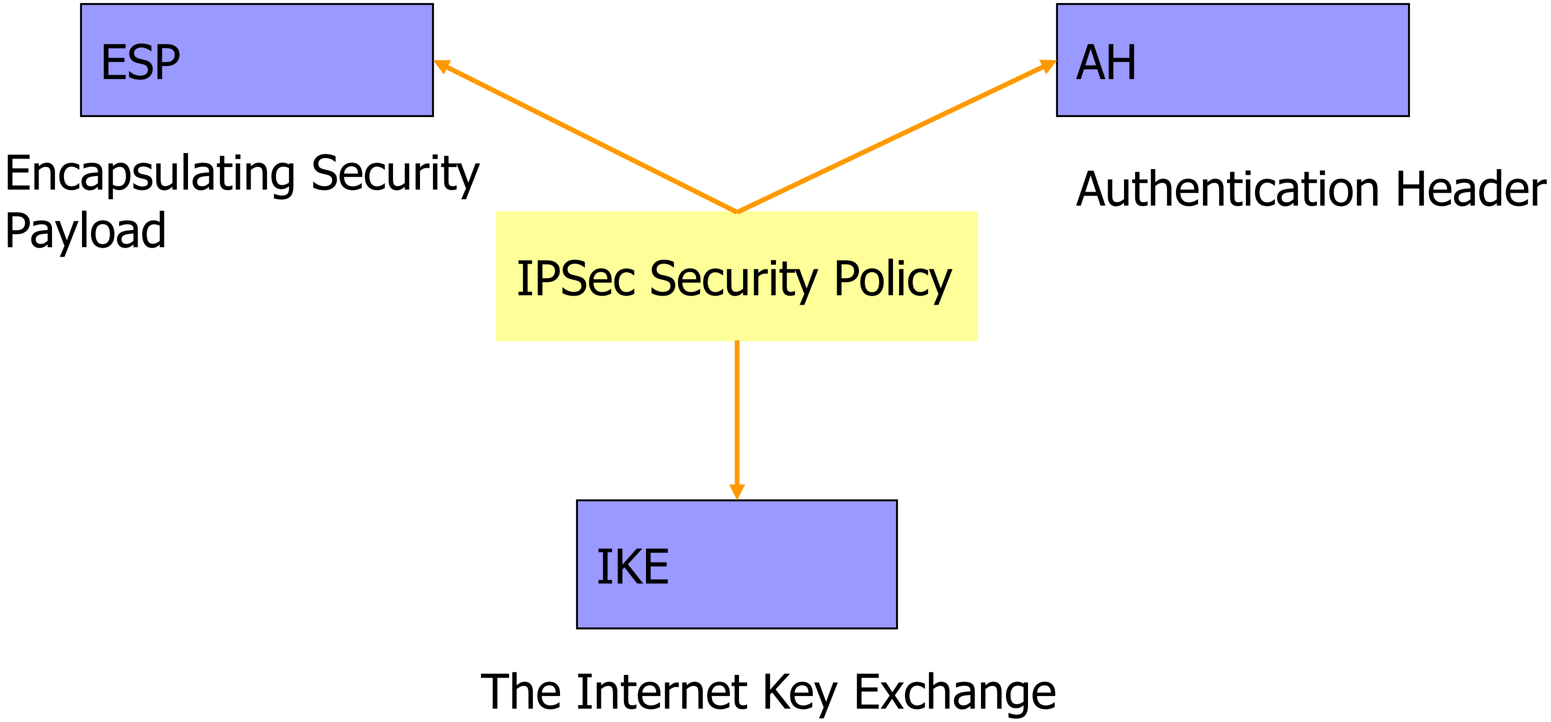
How SSL Works - Summary
Negotiate the Cipher Suite

- A cipher suite
  - A set of cryptographic algorithms
    - An algorithm for exchanging a secret key (asymmetric)
    - A secret key encryption algorithm and key length (symmetric)
    - A cryptographic hash function (HMAC)
- The client tells the server which cipher suites it has available, and the server chooses the best mutually acceptable cipher suite.

The IPSec Security Model



IPSec Architecture



ESP

Encapsulating Security Payload

AH

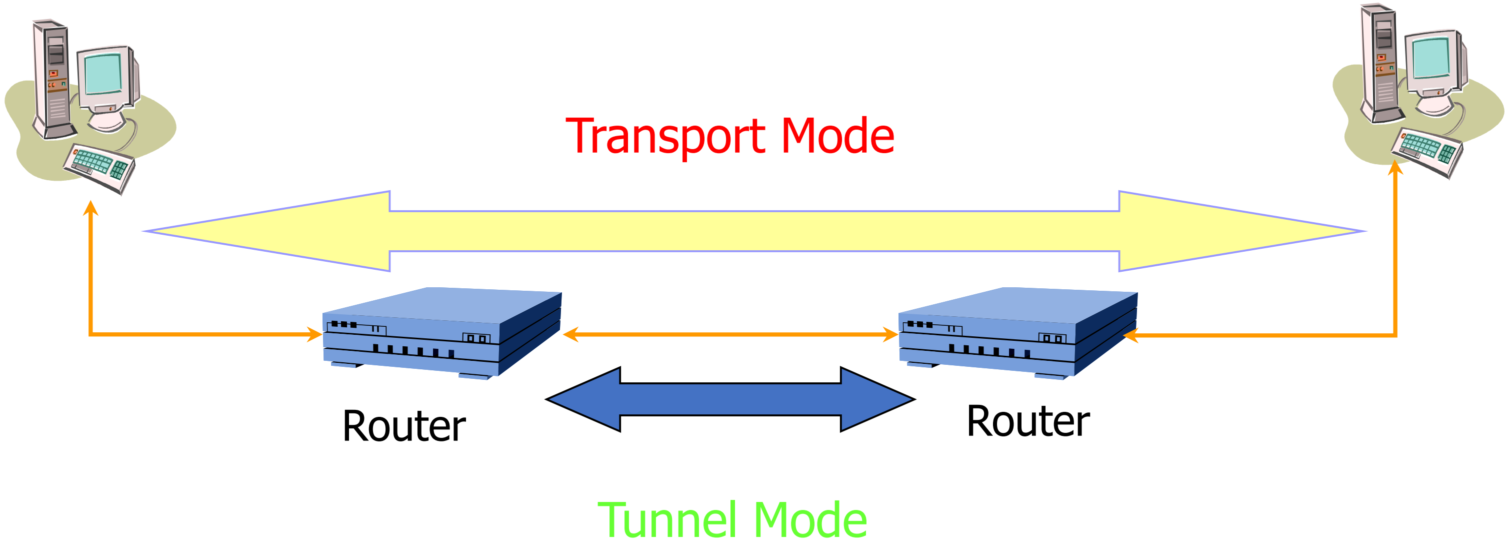
Authentication Header

IPSec Security Policy

IKE

The Internet Key Exchange

IPsec Architecture



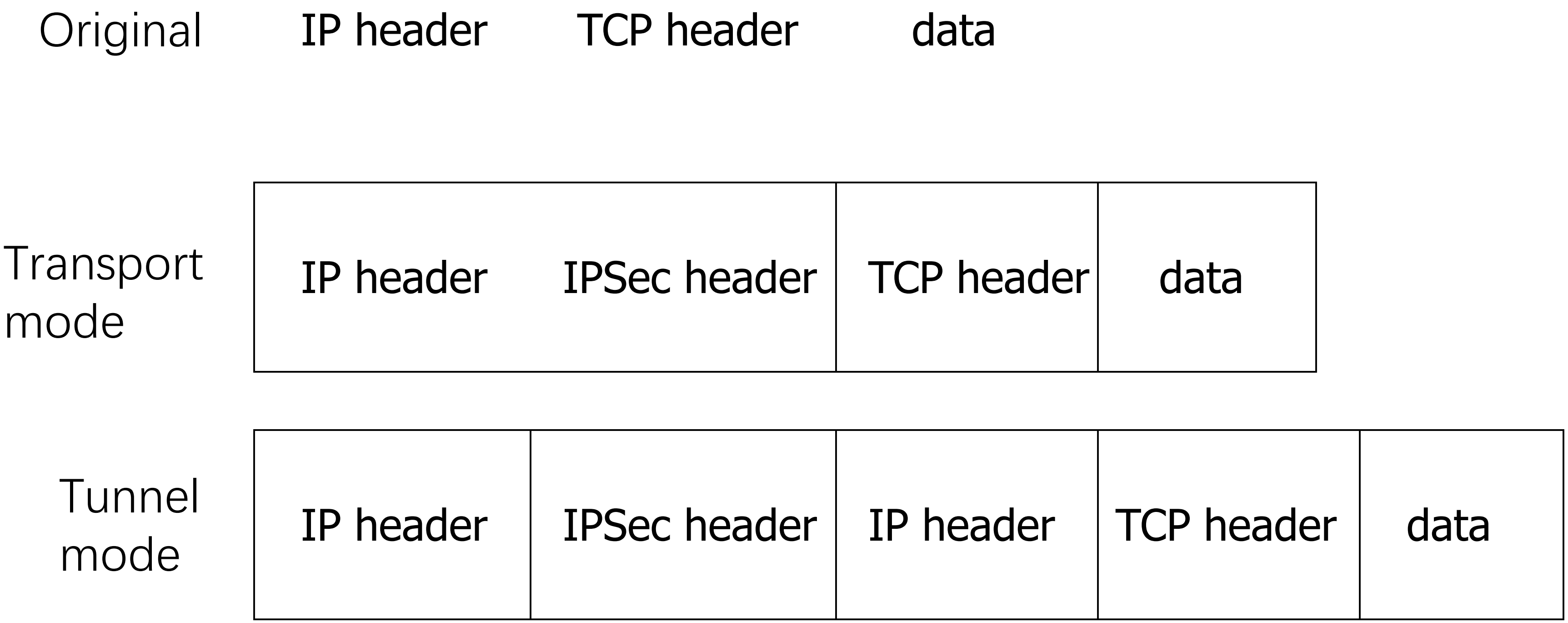
Transport Mode

Router

Router

Tunnel Mode

Various Packets



Original: IP header | TCP header | data

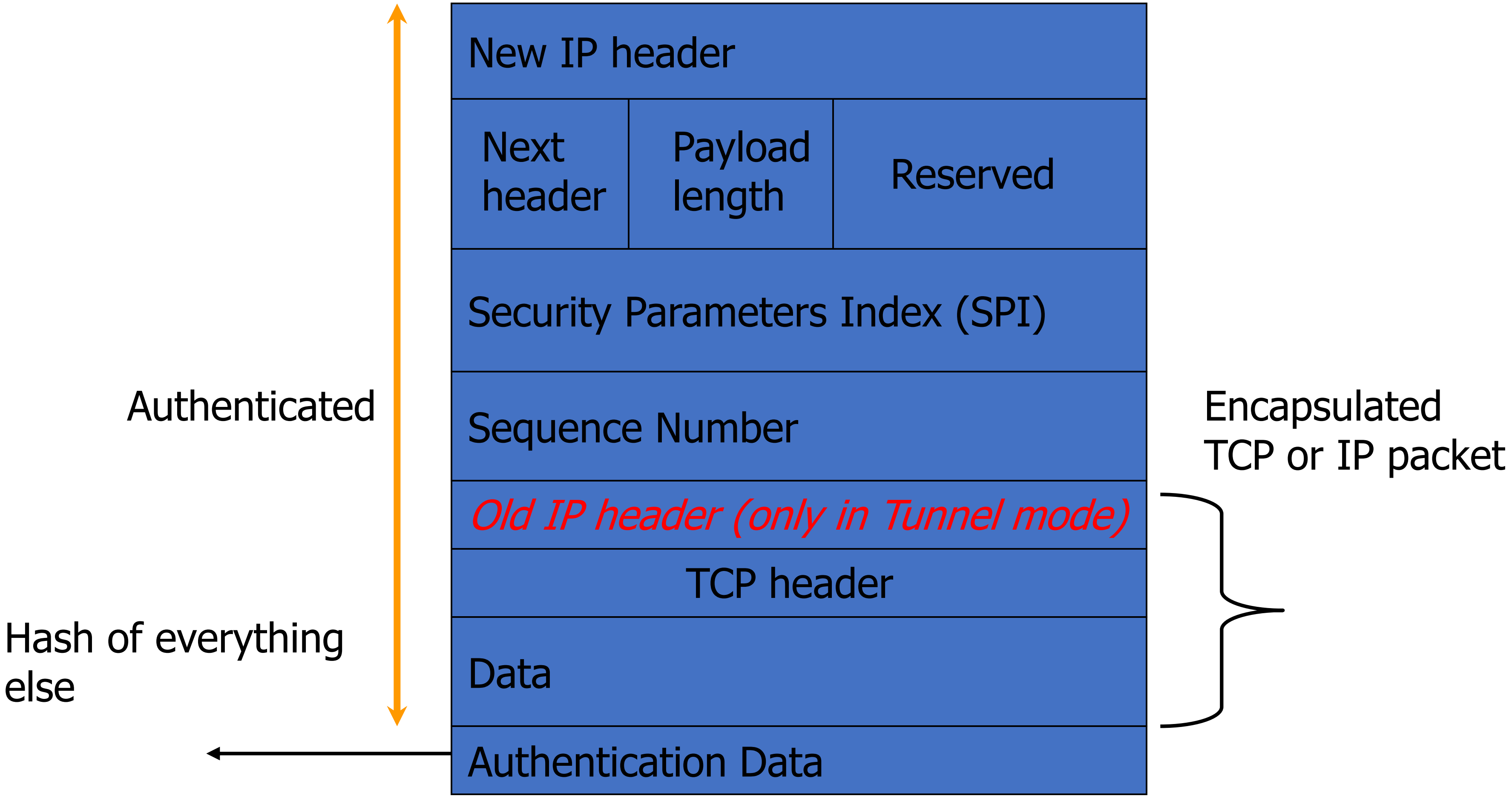
Transport mode: IP header IPSec header | TCP header | data

Tunnel mode: IP header | IPSec header | IP header | TCP header | data

Authentication Header (AH)

- Provides source authentication
  - Protects against source spoofing
- Provides data integrity
- Protects against replay attacks- (valid data transmission is maliciously or fraudulently repeated or delayed)
  - Use monotonically increasing sequence numbers
  - Protects against denial of service attacks
- NO protection for confidentiality!

AH Packet Details



Authenticated

Hash of everything else

New IP header

Next header | Payload length | Reserved

Security Parameters Index (SPI)

Sequence Number

Old IP header (only in Tunnel mode)

TCP header

Data

Authentication Data

Encapsulated TCP or IP packet

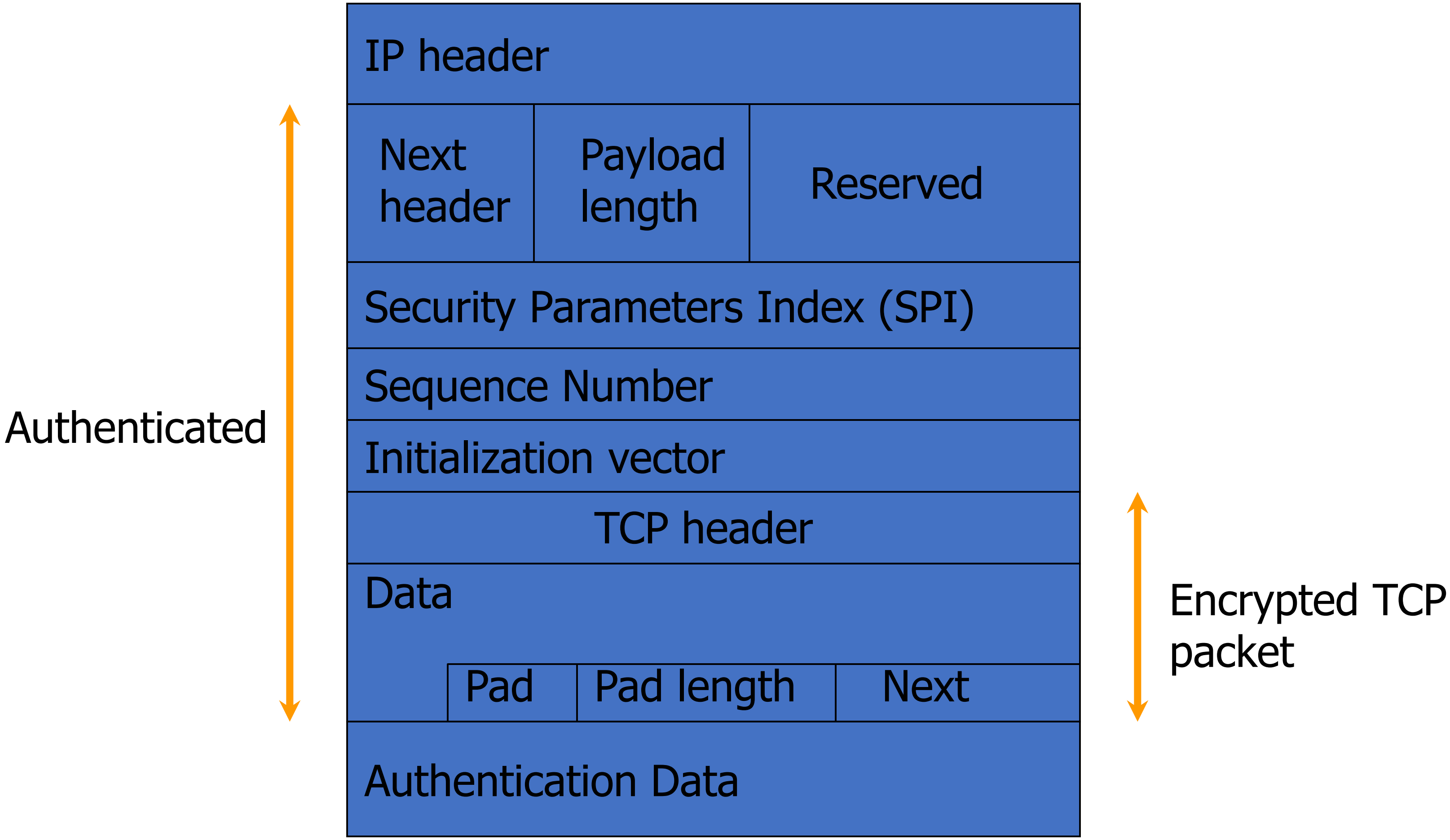
Encapsulating Security Payload (ESP)

- Provides all that AH offers, and
- in addition provides data confidentiality
  - Uses symmetric key encryption

ESP Details

- Same as AH:
  - Use 32-bit sequence number to counter replaying attacks
  - Use integrity check algorithms
- Only in ESP:
  - Data confidentiality:
    - Uses symmetric key encryption algorithms to encrypt packets

ESP Packet Details



IP header

Next header

Payload length

Reserved

Security Parameters Index (SPI)

Sequence Number

Initialization vector

TCP header

Data

Pad

Pad length

Next

Authentication Data

Authenticated

Encrypted TCP packet

Internet Key Exchange (IKE)

- Exchange and negotiate security policies
- Establish security sessions
  - Identified as *Security Associations*
- Key exchange (use Diffie-Hellman key exchange to establish a shared key)
- Key management (RSA)
- Can be used outside IPsec as well