Network Security [2]

Public Key Encryption

Also used in message authentication & key distribution

Based on mathematical algorithms, not only on operations over bit patterns (as conventional) => much overhead

Known also as: Asymmetric

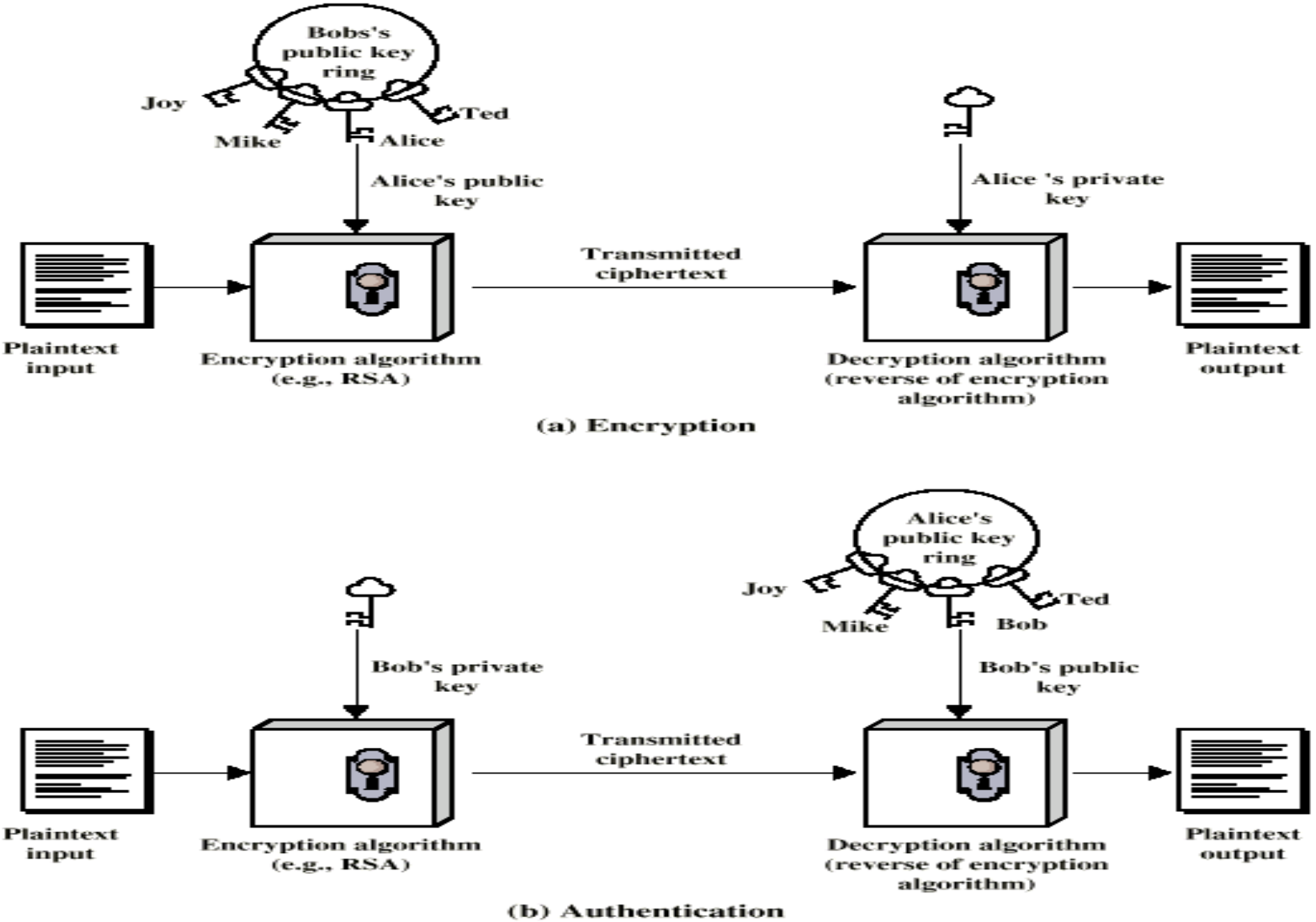
> Use two separate keys: a secrete key (private, not distributed, owned only by one party) and a public key (for all communication parties)

See next slide

Ingredients

> Plain text
>
> Encryption algorithm
>
> Public and private key pair
>
> Cipher text
>
> Decryption algorithm

Public-key encryption & authentication



Public Key Encryption - Operation

One key made public

- Used for encryption

Other kept private

- Used for decryption

Infeasible to determine decryption key, given encryption key and algorithm

Either key can be used for encryption, the other for decryption

Steps

- User generates pair of keys
- User places one key in public domain; he owns a collection of public keys
- To send a message to that user, encrypt using his public key
- User decrypts ciphertext using peer private key

Changing own private key, need for updating correspondent public key

Digital Signature

Public-key encryption used as authentication

Sender encrypts message with its private key; obtain ciphertext as digital signature

Receiver can decrypt, using sender's public key

This authenticates sender, who is only person having the matching key

Does not give privacy of data (confidentiality)

Decrypt key is public

Usually (for storing and speed purposes) the message is transmitted unencrypted, but is accompanied by an 'authenticator', obtained using a hash function depending on the message (if using a piece of the message, a message digest is obtained)

Changing the message => change authenticator

Authenticator encrypted with sender's private key, serving as digital signature

Examples of message digest hash functions:
SHA (Secure Hash Algorithm)
MD5 (used on Internet, cheaper, high speed, less secure than SHA)

RSA (Rivest, Shamir, Adleman) Public-Key Encryption Algorithm

Best public-key encryption algorithm

Use of block cipher, where the plaintext and ciphertext lengths are integers from 0 to n-1, for a given n

p, q primes, typically great values

For calculated n = p*q, calculate Euler totient of n, named $\Phi(n)$, i.e. the number of positive integers less than n and relatively prime (coprime) to n

Select a number e, relatively prime to $\Phi(n)$

Number d is multiplicative inverse of e modulo $\Phi(n)$: $e*d = 1 \bmod \Phi(n)$

See next slide

Key Generation

Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; \quad 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \bmod \phi(n)$
Public key	$\mathrm{KU} = \{e, n\}$
Private key	$\mathrm{KR} = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Decryption

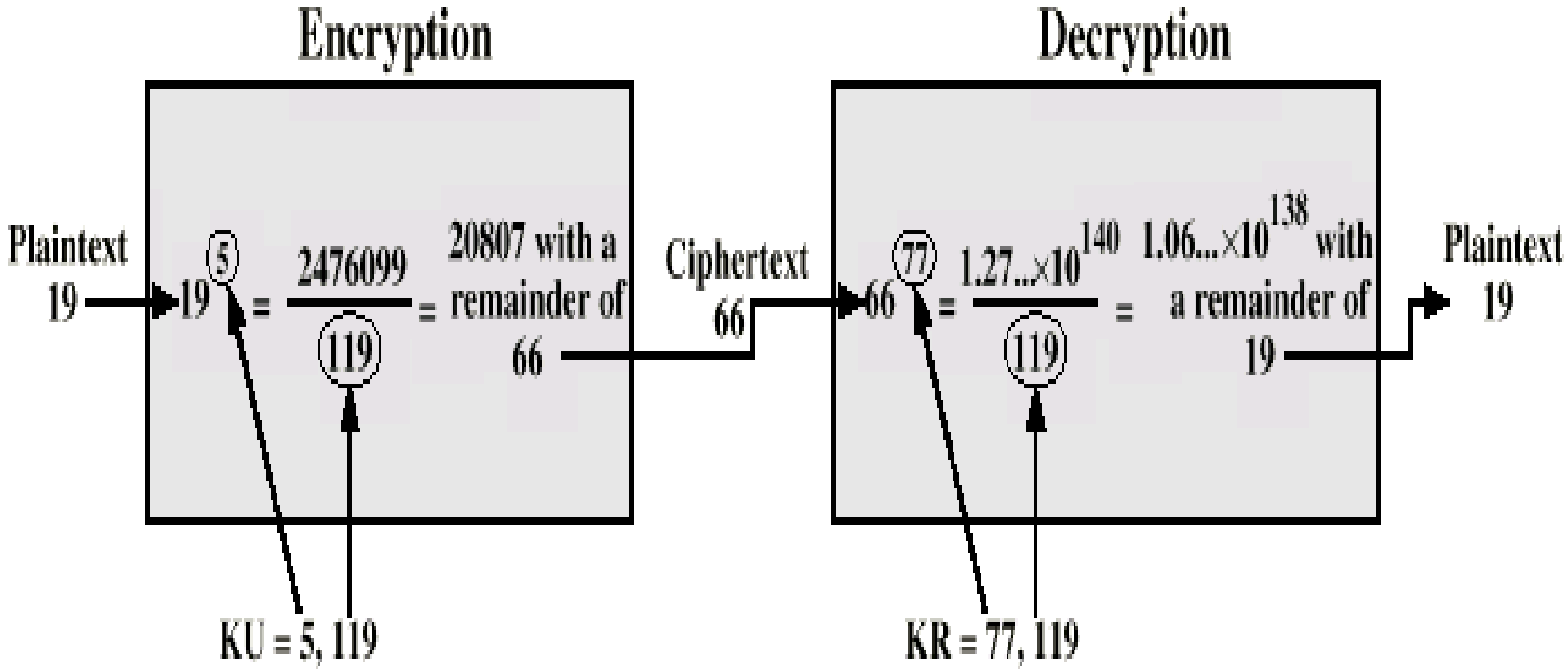
Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

RSA Example

1. Select two prime numbers, $p = 7$ and $q = 17$.
2. Calculate $n = pq = 7 \times 17 = 119$.
3. Calculate $\phi(n) = (p - 1)(q - 1) = 96$.
4. Select e such that e is relatively prime to $\phi(n) = 96$ and less than $\phi(n)$; in this case, $e = 5$.
5. Determine d such that $de = 1 \bmod 96$ and $d < 96$. The correct value is $d = 77$, because $77 \times 5 = 385 = 4 \times 96 + 1$.

The resulting keys are public key KU = {5, 119} and private key KR = {77, 119}. The example shows the use of these keys for a plaintext input of $M = 19$. For encryption, 19 is raised to the fifth power, yielding 2,476,099. Upon division by 119, the remainder is determined to be 66. Hence $19^5 \equiv 66 \bmod 119$, and the ciphertext is 66. For decryption, it is determined that $66^{77} \equiv 19 \bmod 119$.

See next slide


Encryption
Decryption
Plaintext
19
19 5 = 2476099 = 20807 with a remainder of 66
119
Ciphertext
66
66 77 = 1.27...×10^140 = 1.06...×10^138 with a remainder of 19
119
Plaintext
19
KU = 5, 119
KR = 77, 119

Authentication Protocols

Authentication (Is this … ?) is not Authorization (… is allowed to do?)

Authentication based on a **shared secret** key

Is a Challenge-response type protocol:

One party transmits a big random one-time number – number used once (nonce) – unique identification) to

the other, who transforms it in a special

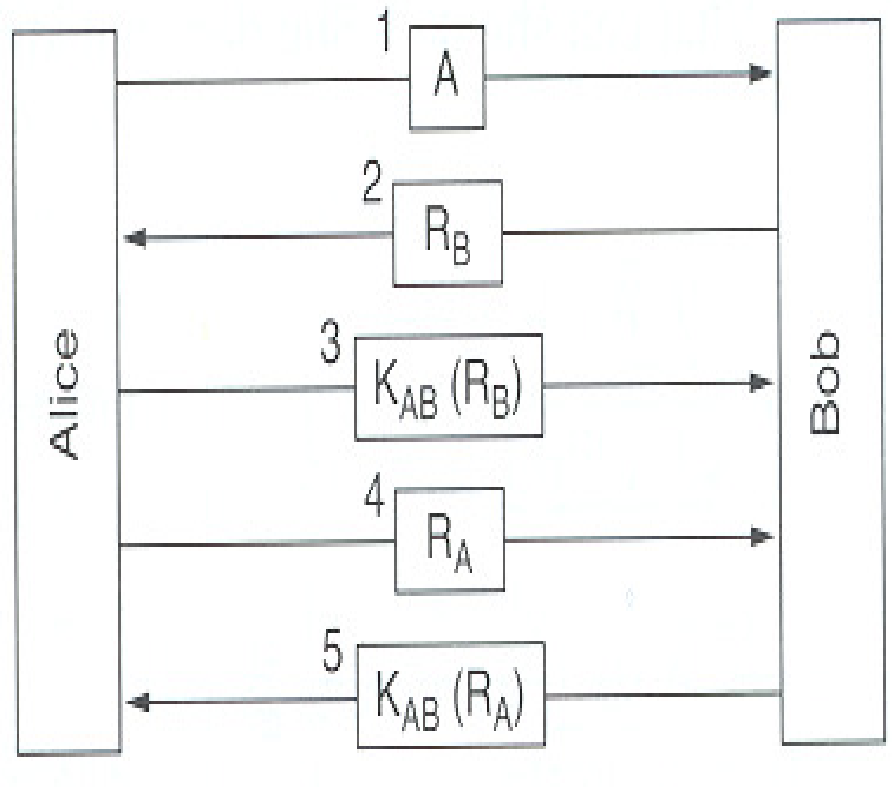
way and returns the result

For a bidirectional authentication: see fig

K_{AB} is the shared secret key

A & B identifies parties

R_A& R_B are nonces



Protocol attacked by reflection attack,

inserting a false session

Diffie-Hellman Key Exchange

Algorithm used in IPSec

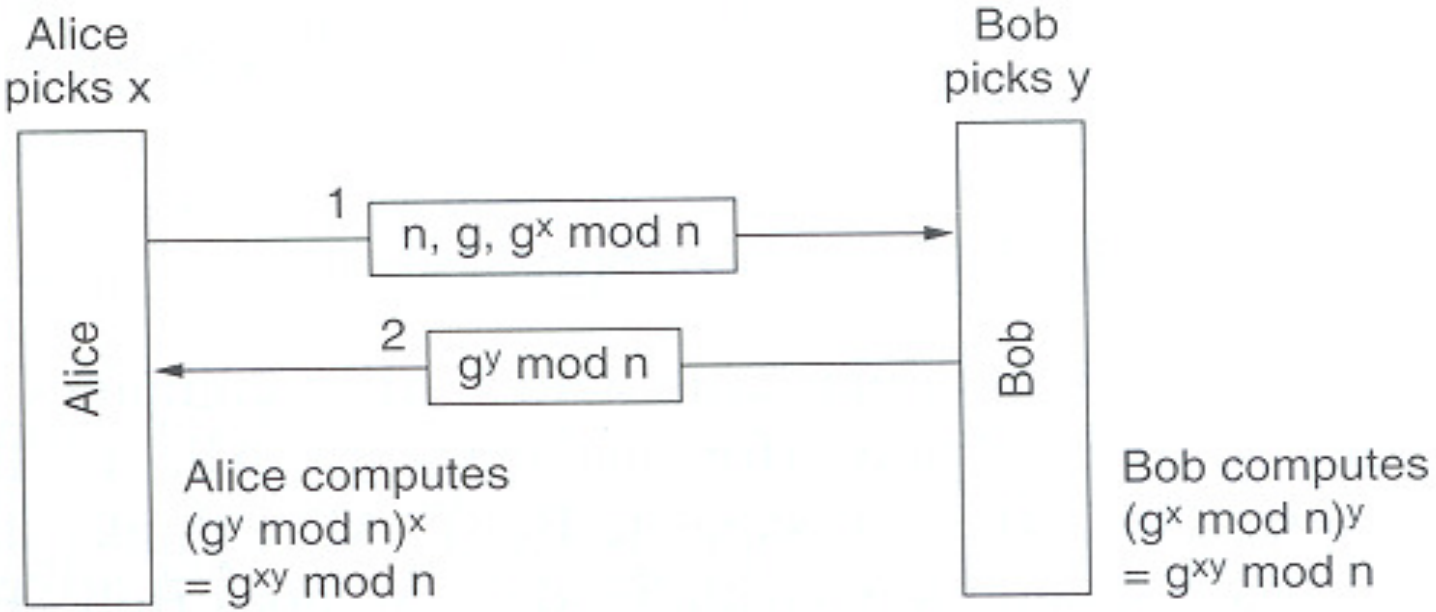
Scope: establishing a *shared secret key* between two 'strangers', on an insecure net

Use of two large prime numbers, n and g with (n-1)/2 also prime; numbers are *public*

A picks and keeps secret a number x

So B part with secret number y

The common secret key K_{AB} : g^{xy} mod n



Example:

Alice & Bob agree to use a prime number n=23 and base g=5.
Alice chooses a secret integer $\boldsymbol{x}$=**6**, then sends Bob A = $g^{\boldsymbol{x}}$ mod n

A = $5^{\mathbf{6}}$ mod 23 = 8.

Bob chooses a secret integer $\boldsymbol{y}$=**15**, then sends Alice : B = $g^{\boldsymbol{y}}$ mod n

B = $5^{\mathbf{15}}$ mod 23 = 19.

Alice computes **s** = $B^{\boldsymbol{x}}$ mod n

$19^{\mathbf{6}}$ mod 23 = **2**.

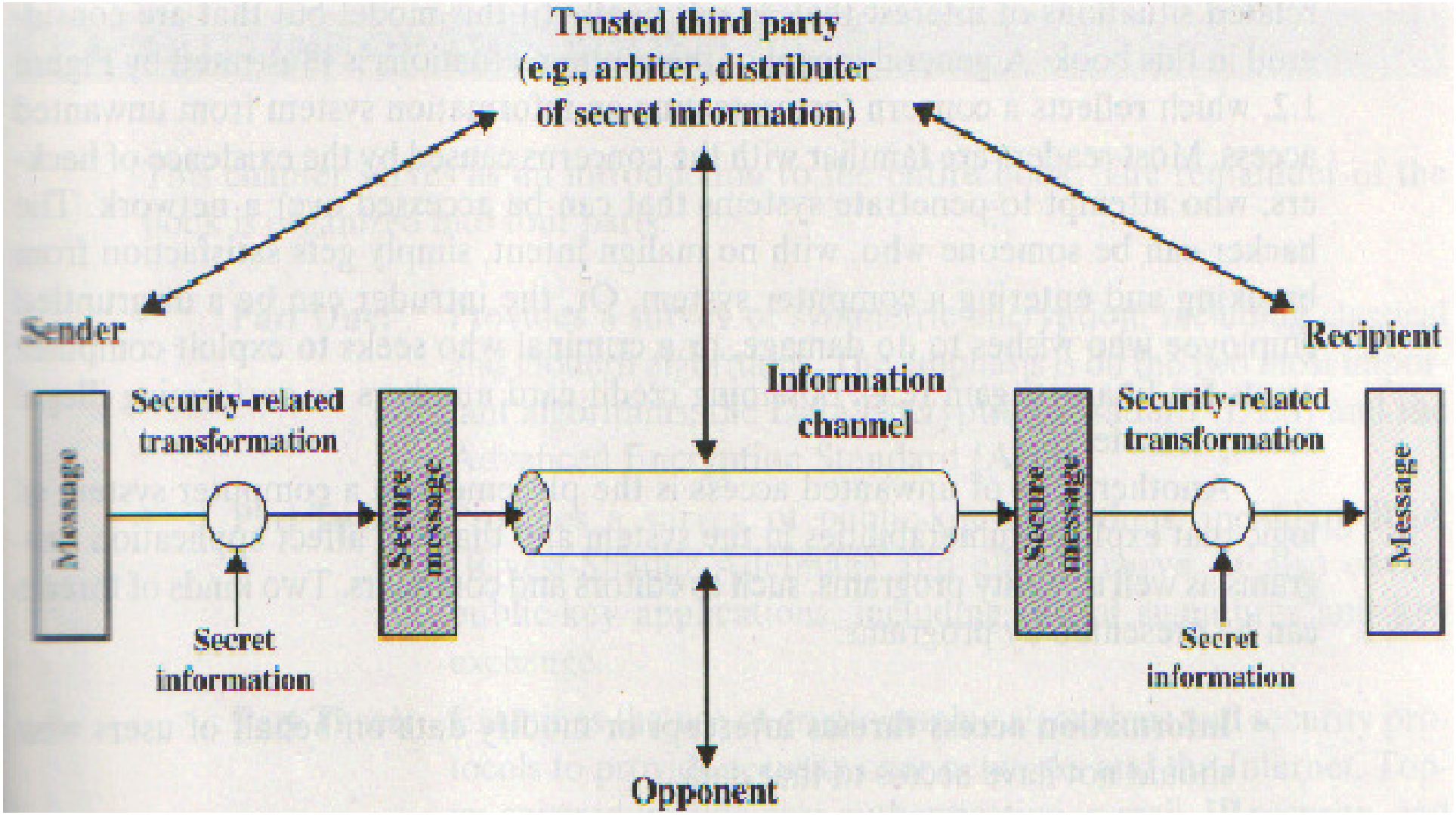
Bob computes **s** = $A^{\boldsymbol{y}}$ mod n

$8^{\mathbf{15}}$ mod 23 = **2**.

Secret key:

s = K_{AB}

A modern model for network security implies the existence of a trusted third party
See figure bellow
Modern authentication algorithms based on KDC (Key Distribution Center)



Session Key

- Used for duration of one logical connection
- Destroyed at end of session
- Used for user data

Permanent key

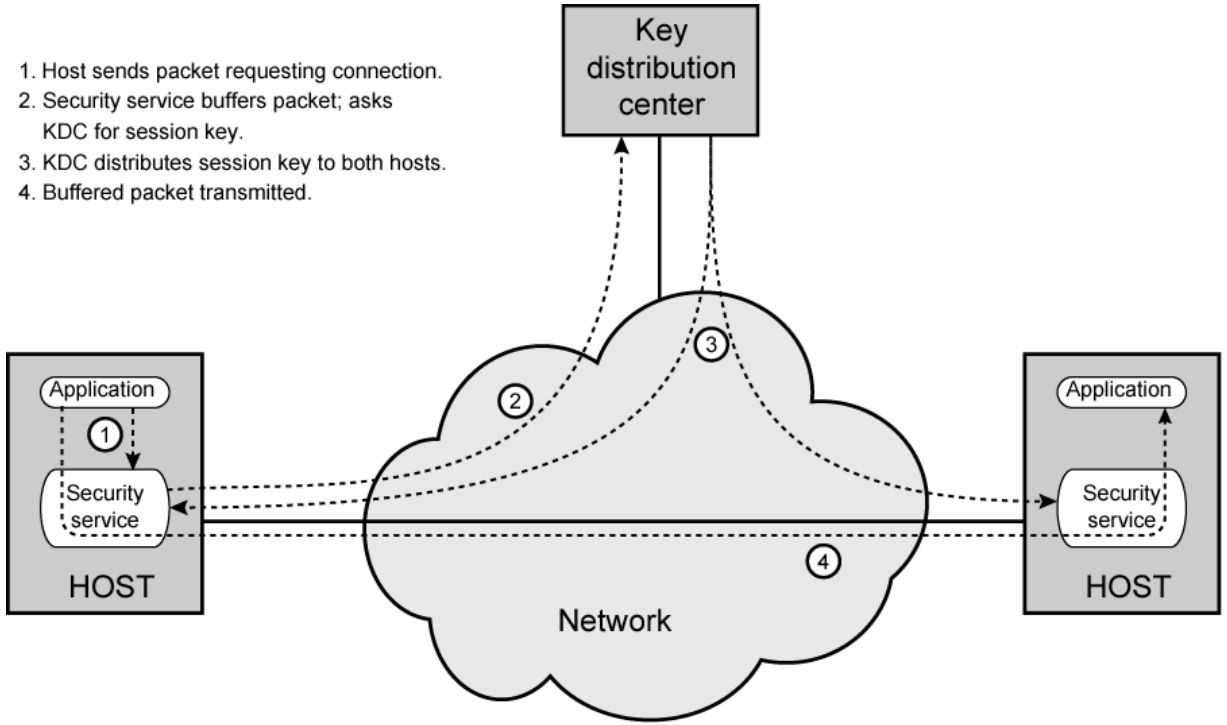
- Used for distribution of keys

Key distribution center

- Determines which systems may communicate
- Provides one session key for that connection

Security service module (SSM)

- Performs end to end encryption
- Obtains keys for host



Multiway challenge-response protocol (Needham-Schroeder)

The *Needham-Schroeder Symmetric Key Protocol* is based on a symetric encription algorithm. It forms the basis for the Kerberos protocol. This protocol aims to establish a session key between two parties on a network, typically to protect further communication.

KDC (denoted here as server S) gives to A a A-B session key K_{AB}

A uses nonce N_A
K_{AS} is a symetric key known by A and S
K_{BS} is a symetric key known by B and S
B will challenge A with nonce N_B

The protocol can be specified as follows:

Alice sends a message to the server identifying herself and Bob, telling the server she wants to communicate with Bob.

$$A \rightarrow S : A, B, N_A$$

The server generates K_{AB} and sends back to Alice a copy encrypted under K_{BS} for Alice to forward to Bob and also a copy for Alice.
The nonce assures Alice that the message is fresh and that the server is replying to that particular message and the inclusion of Bob's name tells Alice who she is to share this key with.

$$S \rightarrow A : \{N_A, K_{AB}, B, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$

Alice forwards the key to Bob who can decrypt it with the key he shares with the server, thus authenticating the data.

$$A \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$$

Bob sends Alice a nonce encrypted under K_{AB} to show that he has the key.

$$B \rightarrow A : \{N_B\}_{K_{AB}}$$

Alice performs a simple operation on the nonce, re-encrypts it
and sends it back verifying that she is still alive and that she holds the key

$$A \rightarrow B : \{N_B - 1\}_{K_{AB}}$$

Authentication using Kerberos

'Real' authentication protocol, based on Needham-Schroeder algorithm

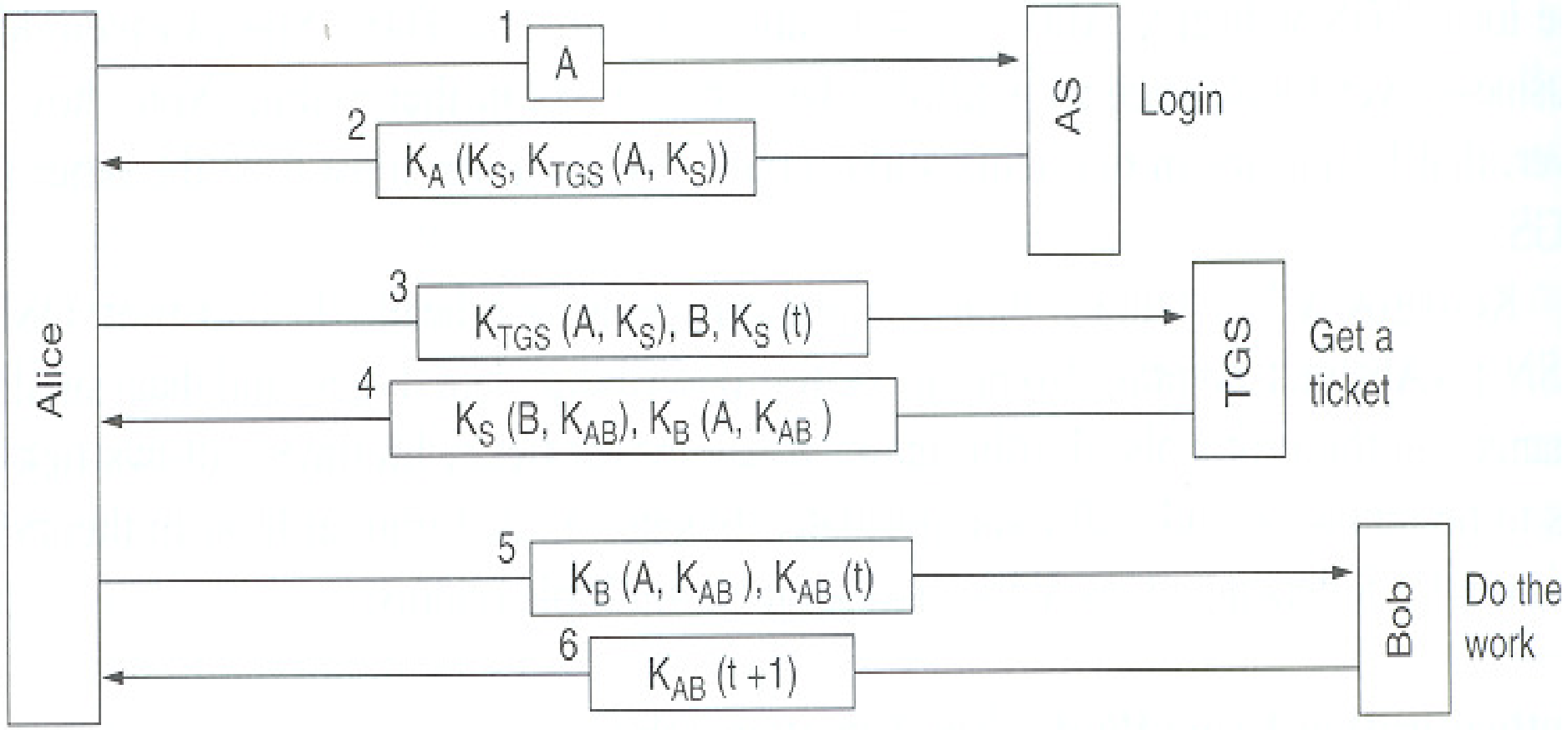
Used by Windows 2000 and other NOS; today use of Kerberos v5

Used in securing access to network resources (server access); A – a user, B – a server

Implies work with extra resources provided by Kerberos:

Authentication Server (AS) – similar to KDC

Ticket-Granting Server (TGS) – issues 'proof of identity' tickets



Kerberos overview

1. User logs on to workstation and requests service on host
2. AS verifies user's access right, creates a ticket-granting ticket and session key; results are encrypted using key derived from user's password and passed to user
3. Workstation, based on user's password, decrypts incoming AS message and sends ticket and authenticator (based on user's name, IP address and time) to TGS
4. TGS decrypts ticket and authenticator, checks request and creates ticket for accessing requested server
5. Workstation sends ticket and authenticator to server
6. Server verifies that ticket and authenticator match, then grants access to service; if mutual authentication is required, server returns an authenticator

Security Protocols in the Internet

IP Level Security

IPv4 and IPv6 Security

IAB (Internet Architecture Board) defined & introduced IPSec specification: necessary security capabilities, over LANs, WANs and across the Internet. Use for:

Secure branch office connectivity over Internet

Secure remote access over Internet

Extranet and intranet connectivity

Enhanced electronic commerce security

IPSec Scope

Three main facilities:

Authentication header (AH) – authentication function

Encapsulated security payload (ESP) – combined authentication/encryption function

Key exchange function

Security Association (SA)

One way relationship between sender

and receiver, affording security services

For two ways, two associations are required

Three SA identification parameters

- Security parameter index – local meaning
- IP destination address (SA destination endpoint) – unicast for moment
- Security protocol identifier: use of AH or ESP

SA implementation - Parameters

Sequence number counter

Sequence counter overflow

Anti-reply windows

AH information

ESP information

Lifetime of this association

IPSec protocol mode

- Tunnel, transport or wildcard

Path MTU

Transport and Tunnel Modes

AH & ESP support two modes of use:

Transport mode

- Protection for upper layer protocols
- Extends only to payload of an IP packet
- End to end between hosts

Tunnel mode

- Protection for entire IP packet
- Entire packet (IP packet + security fields) treated as payload for ‘outer’ IP packet
- No routers examine inner packet (traverses a ‘tunnel’ between IP endpoints)
- Outer packet may have different source and destination address, due to security adds (carries firewall address)
- Tunnel mode may be implemented with firewall or IP router implementing IPSec software

Authentication Header

Support for data integrity and authentication for IP packets

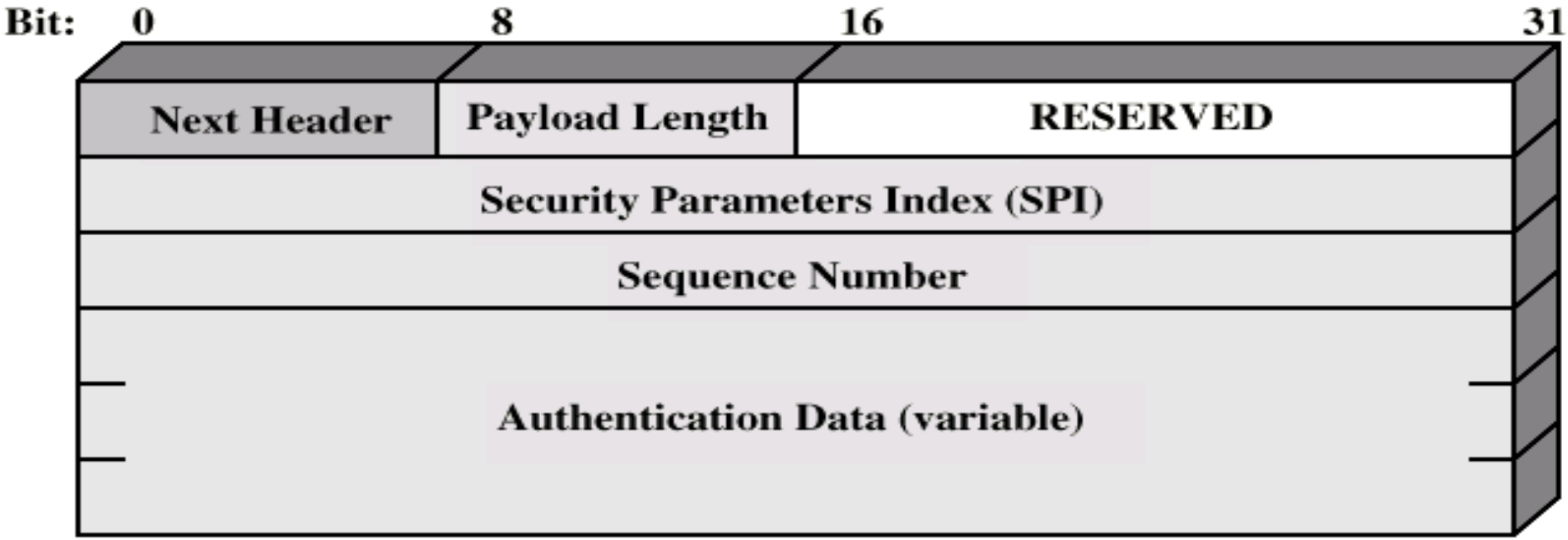
A Security Association identified in *Security Parameters Index* field

Use of Message Authentication Code (result of hash functions) – stored in *Authentication Data* field

Two IP endpoint parts must share a secret key

Authentication calculated over immutable IP fields (during transit) or predictable in value at arrival; the rest are set to 0, for purposes of calculation and over IP payload

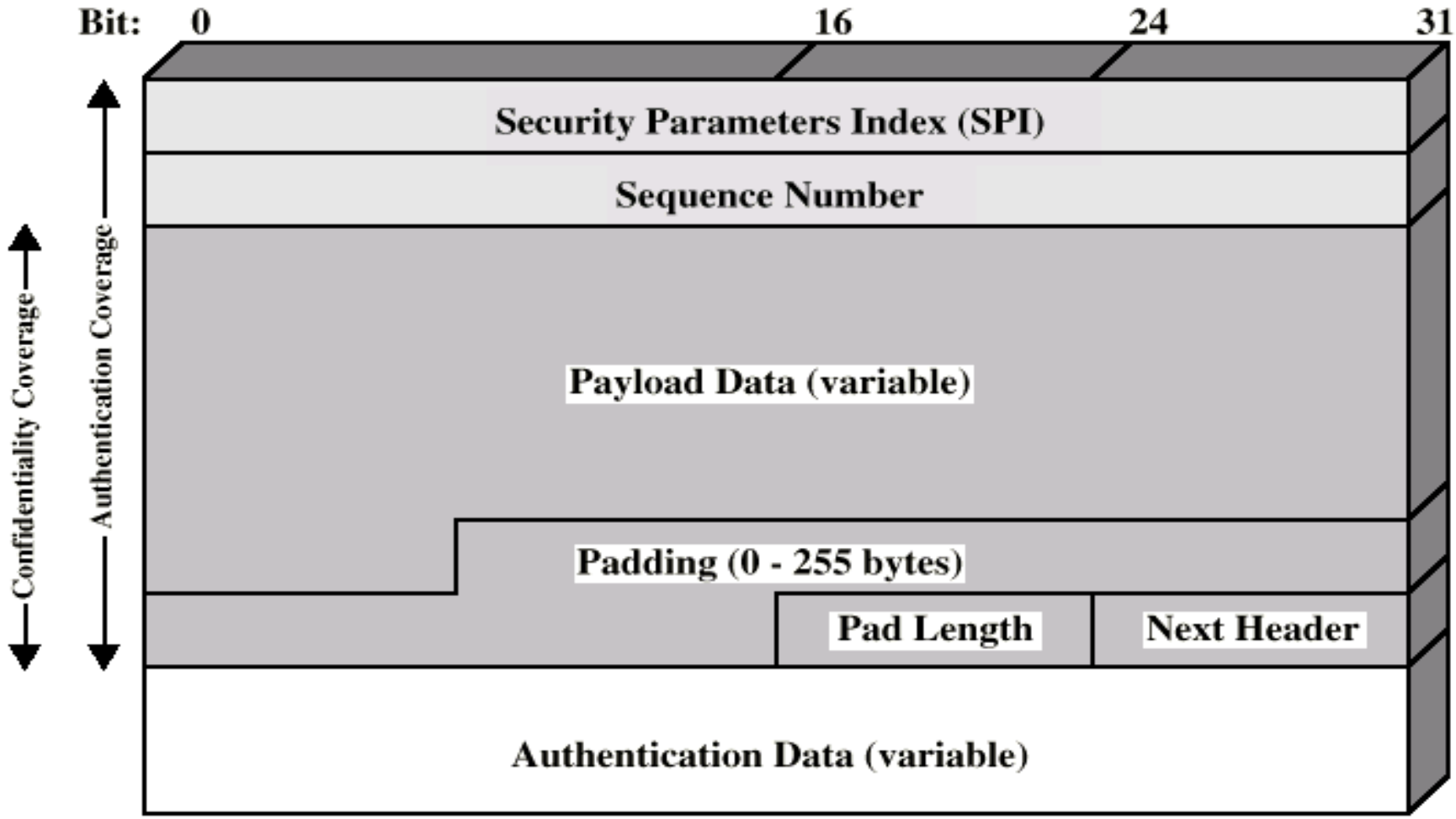
IPSec Authentication Header



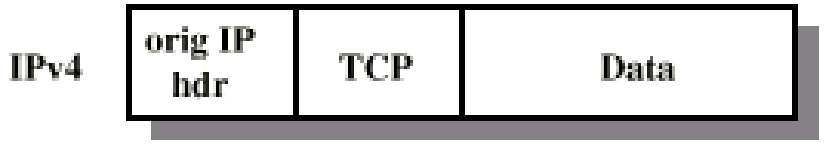
Encapsulating Security Payload

Provides confidentiality services and some authentication

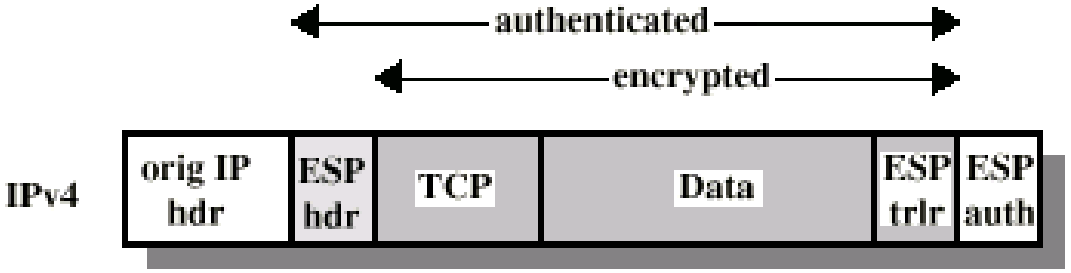
Payload Data field is a transport level segment (transport mode) or IP packet (tunnel mode)



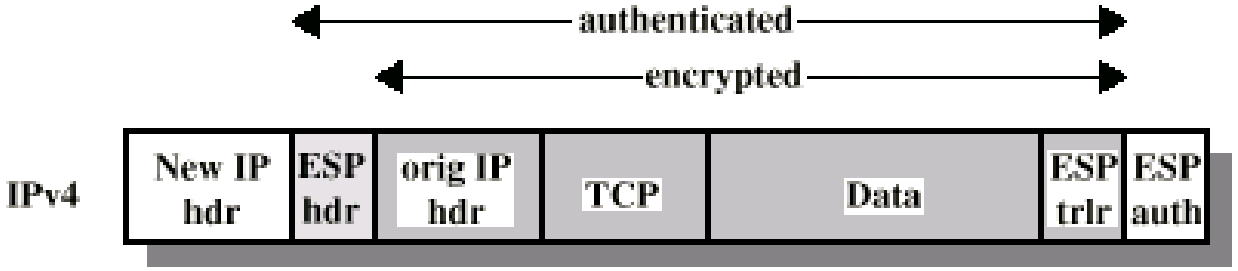
Scope of ESP Encryption & Authentication



(a) Original IP Packet



(b) Transport Mode



(c) Tunnel Mode

Key Management for IPSec

Determination & Distribution of secret keys:

Manual

Network administrator

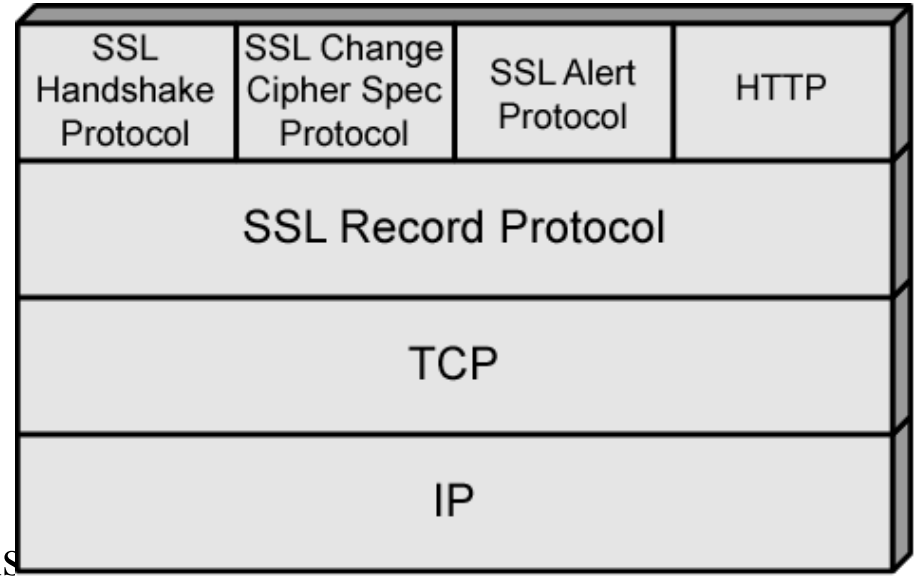
Automatic

Use of ISAKMP/Oakley automated key management protocol

Oakley key determination protocol: Diffie-Hellman key exchange algorithm with extra security

Internet Security Association and Key Management Protocol – gives specific protocol support (message types & formats, negotiation of security attributes)

Secure Sockets Layer (SSL)



SSL general-purpose service

Set of protocols that rely on TCP

Two implementation options

Part of underlying protocol suite

Transparent to applications

Embedded in specific packages

e.g. Netscape and Microsoft Explorer and most Web servers

SSL uses TCP to provide reliable end-to-end secure service

SSL uses two layers of protocols

Record Protocol, on top of TCP provides basic security services to various higher-layer protocols

In particular, HTTP can operate on top of SSL

Three higher-layer protocols , on top os SSL Record Protocol

Handshake Protocol

Change Cipher Spec Protocol

Alert Protocol

Used in management of SSL exchanges

SSL Record Protocol

Confidentiality (Handshake Protocol defines shared secret key)

Message Integrity (Handshake Protocol defines shared secret key; used to form message authentication code (MAC))

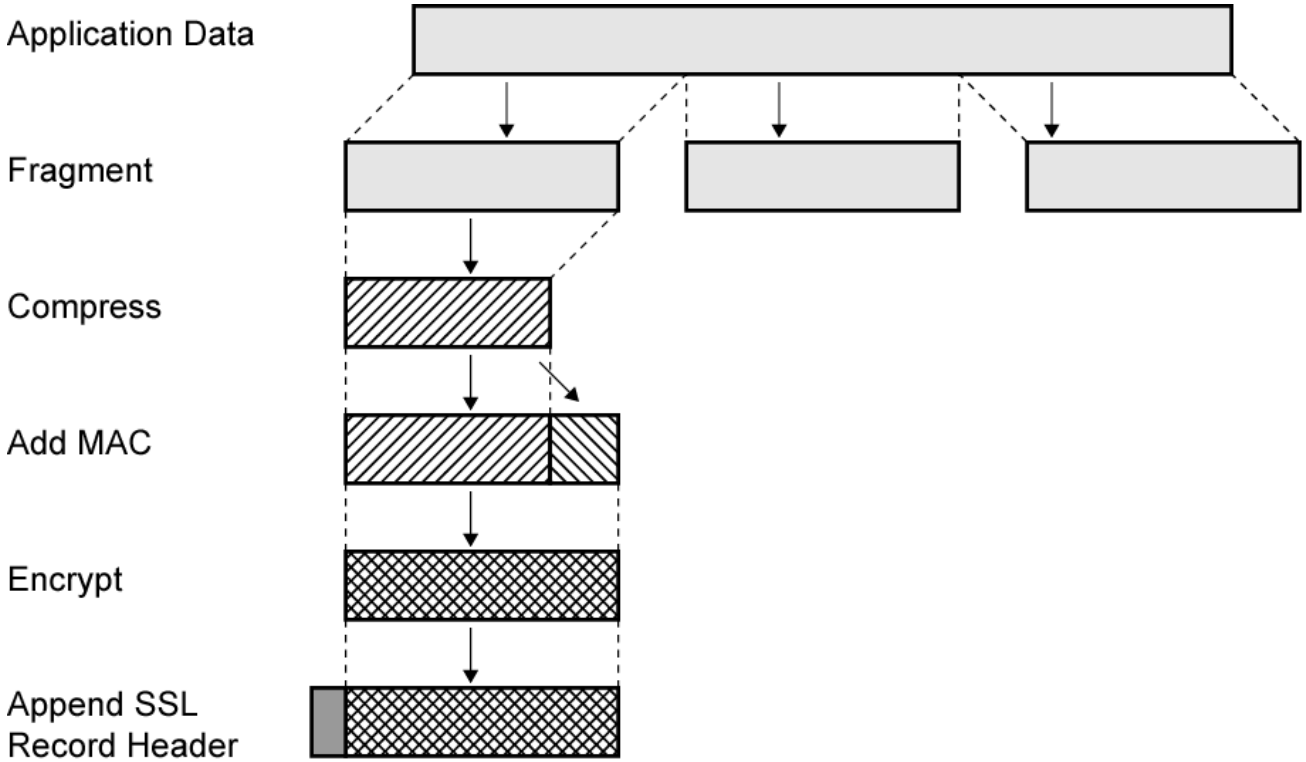
Each upper-layer message fragmented

2^14 bytes (16384 bytes) or less

Compression optionally applied

Compute message authentication code (MAC)

Compressed message plus MAC encrypted using symmetric encryption



Alert Protocol

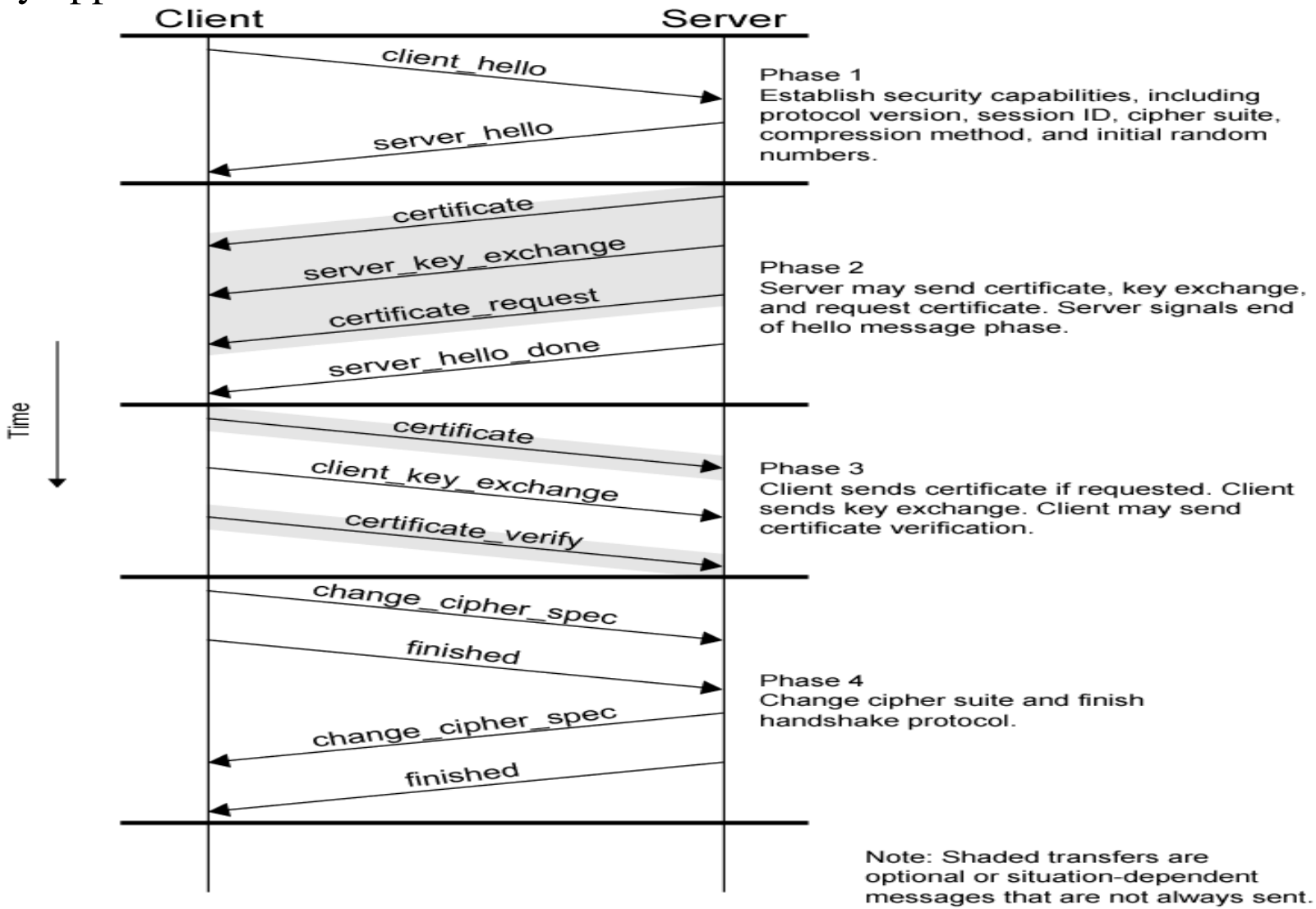
Convey SSL-related alerts to peer entity
Alert messages compressed and encrypted
- Two bytes
  - First byte warning(1) or fatal(2)
    - If fatal, SSL immediately terminates connection
    - Other connections on session may continue
    - No new connections on session
  - Second byte indicates specific alert
    - E.g. fatal alert is an incorrect MAC
    - E.g. nonfatal alert is close_notify message

Change Cipher Spec Protocol

Uses Record Protocol
Single message
- Updates cipher suite to be used on this connection

Handshake Protocol

Authenticate
Negotiate encryption and MAC algorithm and cryptographic keys
Used before any application data sent



Transport Layer Security (TLS)

Security protocol lying between an application (HTTP) and TCP levels

TLS derived from SSL (Secure Sockets Layer – developed for web-security)

TLS composed from:

Handshake protocol

security negotiation, server authentication to the browser, exchange of client's secret key

Data Exchange protocol

data exchange between client & server, using accepted secret key

Application Layer Security Protocols

Pretty Good Privacy (PGP)

Used extensively for email applications

Advantage of using the best protocols for public key encryption (RSA, Diffie-Hellman), or symmetric encryption (IDEA, TDES); SHA for hash coding

Used by any individual user, also by companies

S/MIME (Secure/ Multipurpose Internet Mail Extension)

Based on technology from RSA

Emerging as standard for commercial & organizational use

Other Network Security Issues

Firewalls

Comms device (router, computer) installed between organization's internal network and rest of the Internet

Normally used as a packet filter, based on info from network or transport levels headers: IP source/destination addresses, type of protocol (TCP/UDP)

At Application level, use of Proxy Firewall

For filtering, use of info from Application level

Most common: HTTP proxy, acting between external clients and the corporation HTTP server

Virtual Private Networks (VPN)

Private: guarantees privacy inside organization

Virtual: doesn't use real private WANs, is a public network physically

Use of Internet for both private and public communications

Use of IPSec in the tunnel mode for authentication, privacy, integrity functions