Network Security

Chapter 8

Network Security

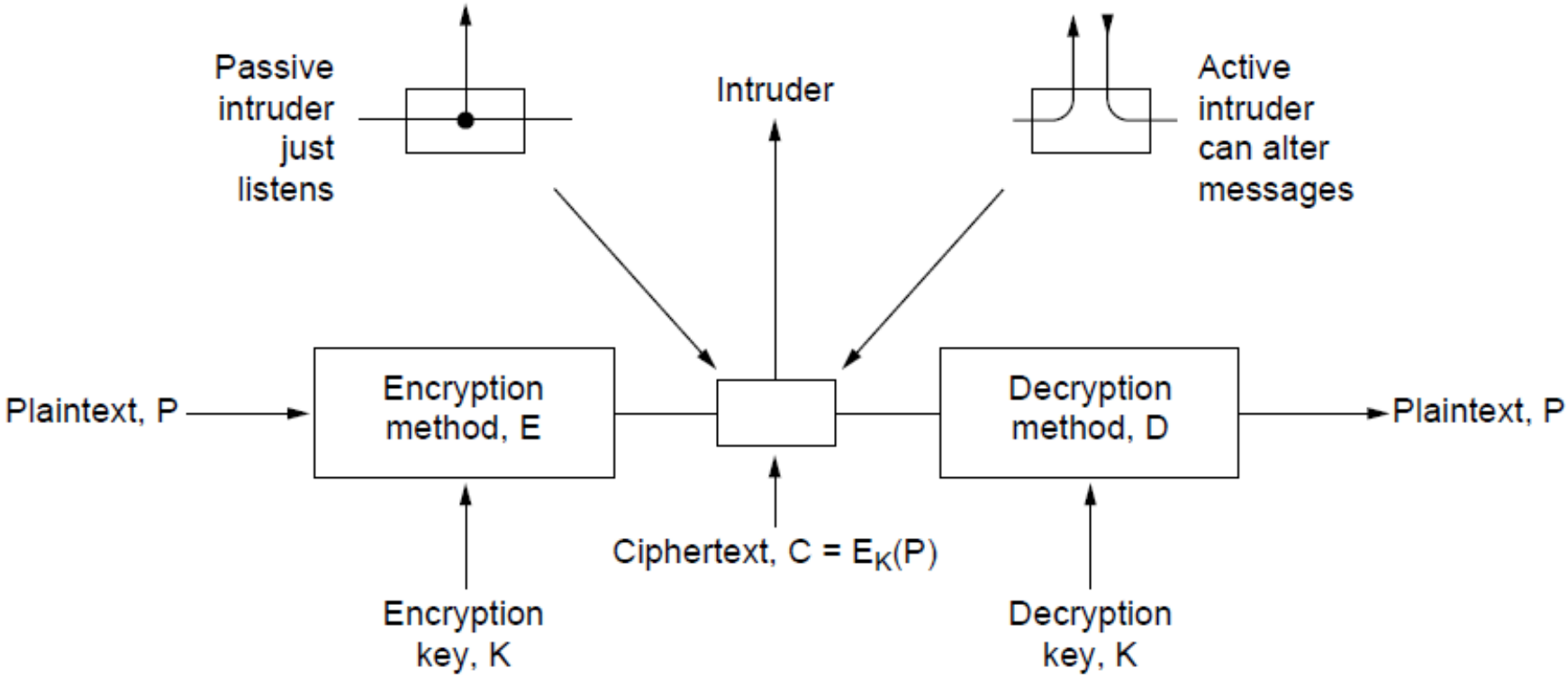
Adversary	Goal
Student	To have fun snooping on people's email
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by email
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military or industrial secrets
Terrorist	To steal germ warfare secrets

Some people who cause security problems and why.

Cryptography

- Introduction
- Substitution ciphers
- Transposition ciphers
- One-time pads
- Fundamental cryptographic principles

Introduction



The encryption model (for a symmetric-key cipher).

Substitution Ciphers

plaintext:	a b c d e f g h i j k l m n o p q r s t u v w x y z
ciphertext:	Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

Monoalphabetic substitution

Transposition Ciphers

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

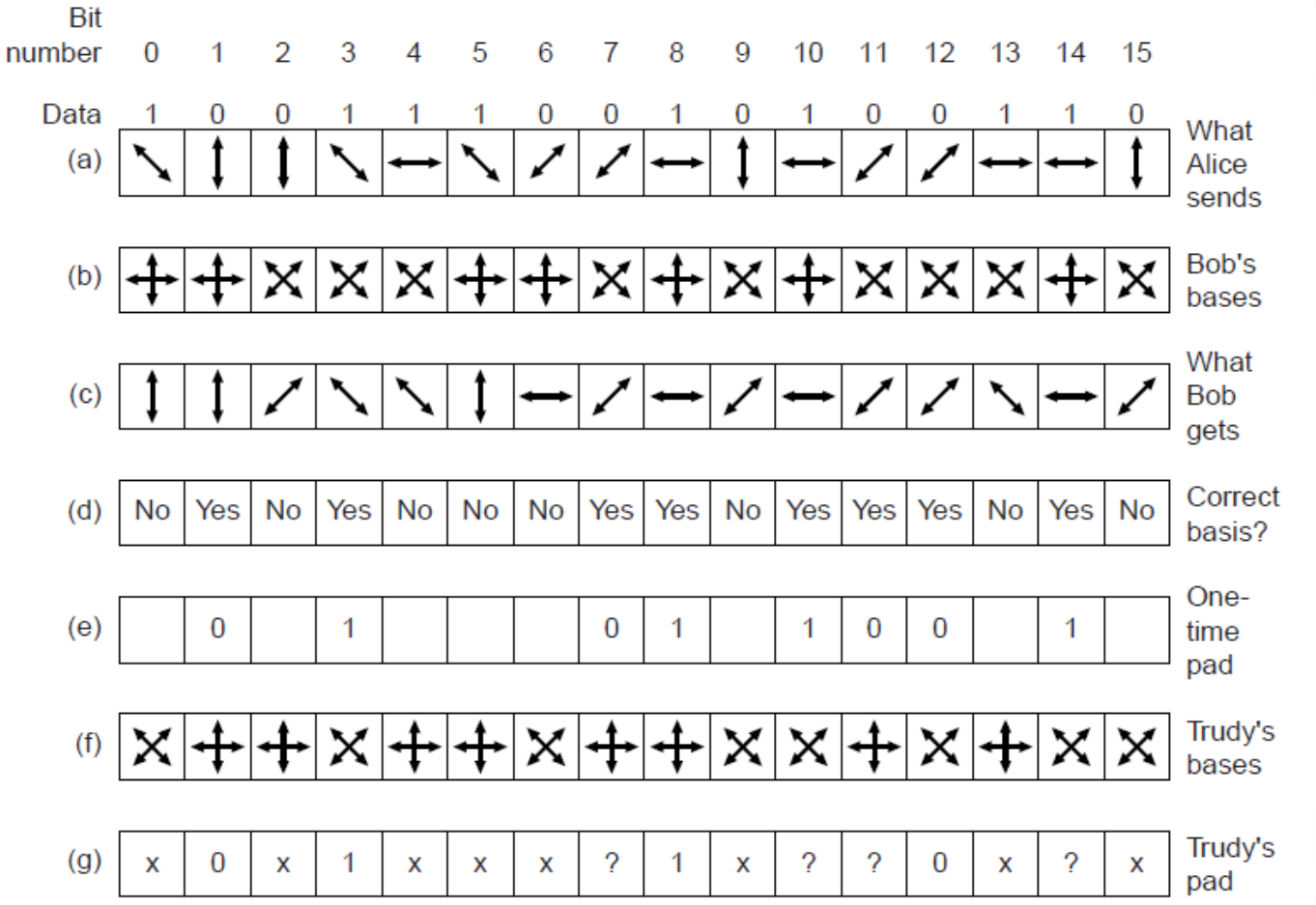
AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

One-Time Pads (1)

Message 1:	1001001	0100000	1101100	1101111	1110110	1100101	0100000	1111001	1101111	1110101	0101110
Pad 1:	1010010	1001011	1110010	1010101	1010010	1100011	0001011	0101010	1010111	1100110	0101011
Ciphertext:	0011011	1101011	0011110	0111010	0100100	0000110	0101011	1010011	0111000	0010011	0000101
Pad 2:	1011110	0000111	1101000	1010011	1010111	0100110	1000111	0111010	1001110	1110110	1110110
Plaintext 2:	1000101	1101100	1110110	1101001	1110011	0100000	1101100	1101001	1110110	1100101	1110011

The use of a one-time pad for encryption and the possibility of getting any possible plaintext from the ciphertext by the use of some other pad.

One-Time Pads (2)

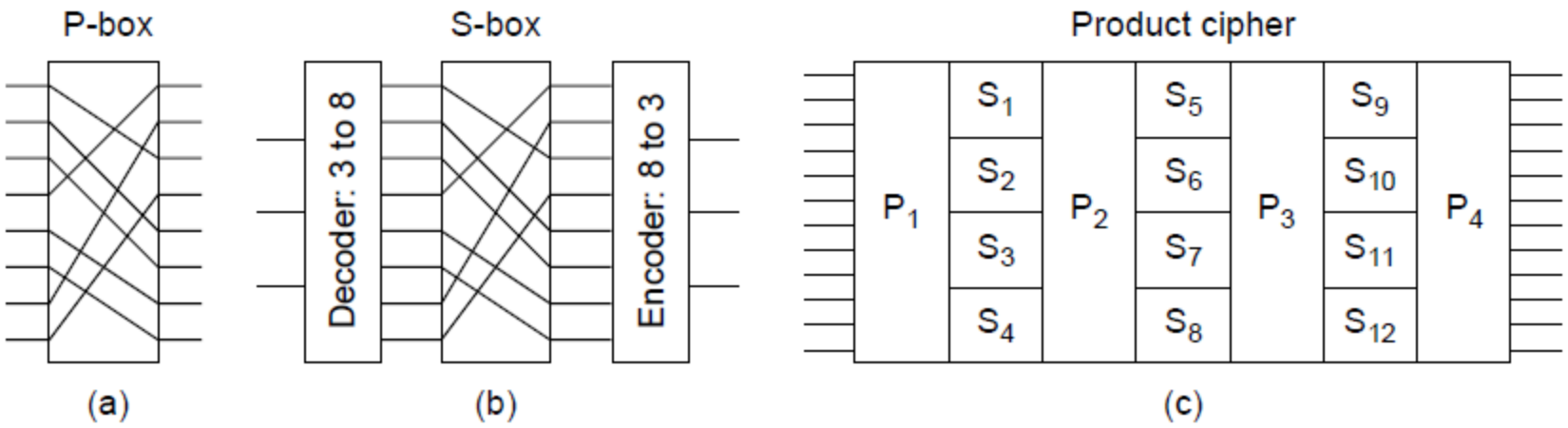


An example of quantum cryptography

Fundamental Cryptographic Principles

1. Messages must contain some redundancy
2. Some method is needed to foil replay attacks

Symmetric-key Algorithms (1)

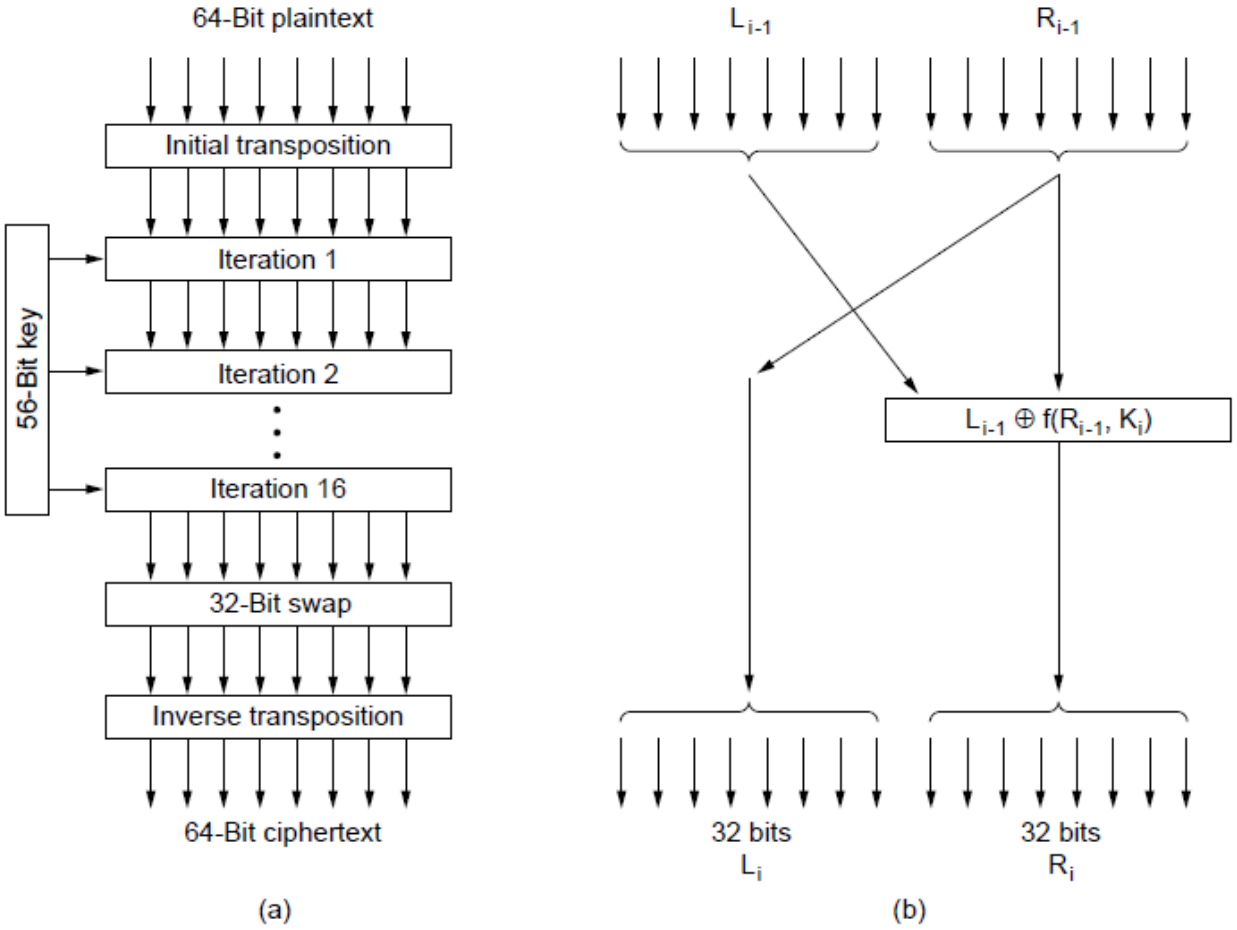


Basic elements of product ciphers.
(a) P-box. (b) S-box. (c) Product.

Symmetric-key Algorithms (2)

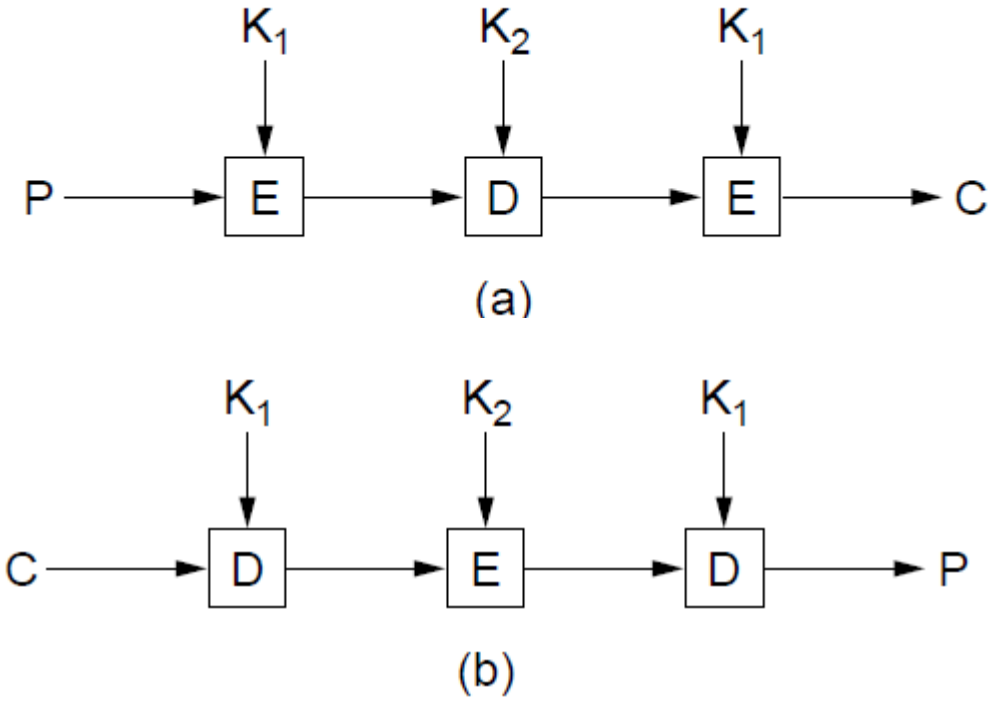
- Data encryption standard
- Advanced encryption standard
- Cipher modes
- Other ciphers
- Cryptanalysis

Data Encryption Standard (1)



The data encryption standard. (a) General outline. (b) Detail of one iteration. The circled + means exclusive OR.

Data Encryption Standard (2)



(a) Triple encryption using DES. (b) Decryption

Advanced Encryption Standard (1)

1. Algorithm symmetric block cipher.
2. Full design must be public.
3. Key lengths of 128, 192, and 256 bits supported.
4. Software and hardware implementations possible.
5. Algorithm public or licensed on nondiscriminatory terms.

Advanced Encryption Standard (2)

```
#define LENGTH 16                                   /* # bytes in data block or key */
#define NROWS 4                                     /* number of rows in state */
#define NCOLS 4                                     /* number of columns in state */
#define ROUNDS 10                                   /* number of iterations */
typedef unsigned char byte;                         /* unsigned 8-bit integer */

rijndael(byte plaintext[LENGTH], byte ciphertext[LENGTH], byte key[LENGTH])
{
  int r;                                            /* loop index */
  byte state[NROWS][NCOLS];                         /* current state */
  struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1];    /* round keys */

  expand_key(key, rk);                              /* construct the round keys */
  copy_plaintext_to_state(state, plaintext);        /* init current state */
  xor_roundkey_into_state(state, rk[0]);            /* XOR key into state */

     . . .
```

An outline of Rijndael

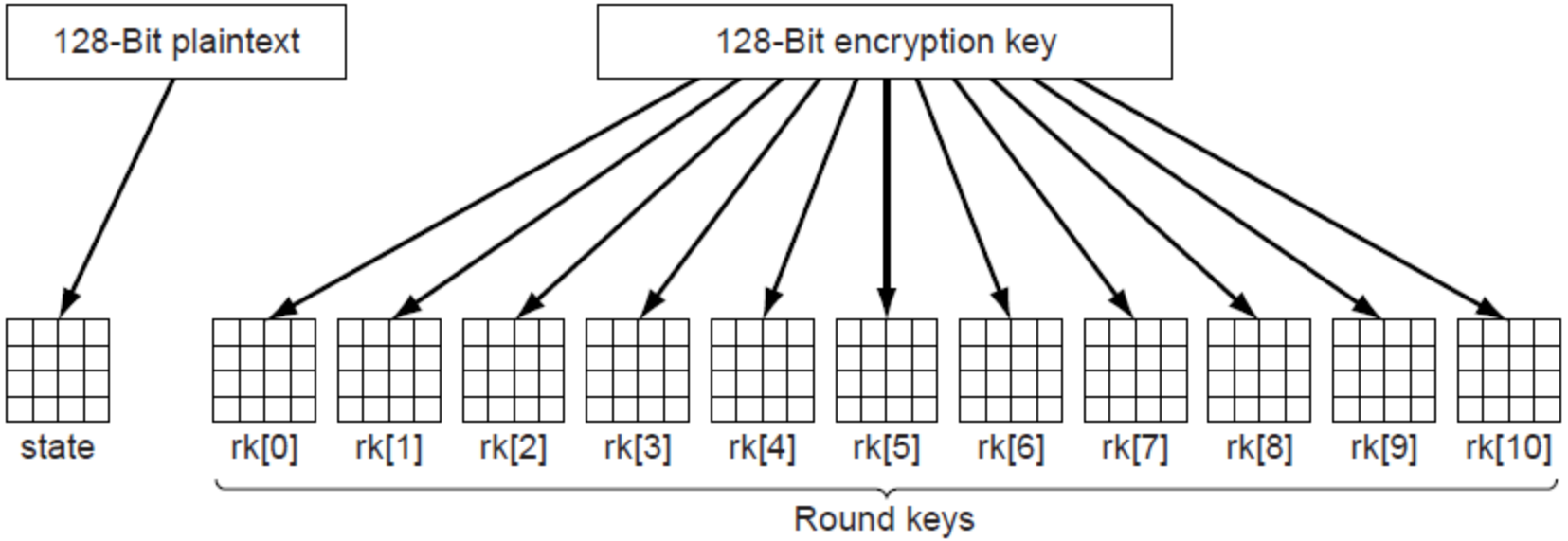
Advanced Encryption Standard (3)

```
  expand_key(key, rk);                                  /* construct the round keys */
  copy_plaintext_to_state(state, plaintext);            /* init current state */
  xor_roundkey_into_state(state, rk[0]);                /* XOR key into state */

  for (r = 1; r <= ROUNDS; r++) {
       substitute(state);                               /* apply S-box to each byte */
       rotate_rows(state);                              /* rotate row i by i bytes */
       if (r < ROUNDS) mix_columns(state);              /* mix function */
       xor_roundkey_into_state(state, rk[r]);           /* XOR key into state */
  }
  copy_state_to_ciphertext(ciphertext, state);          /* return result */
}
```

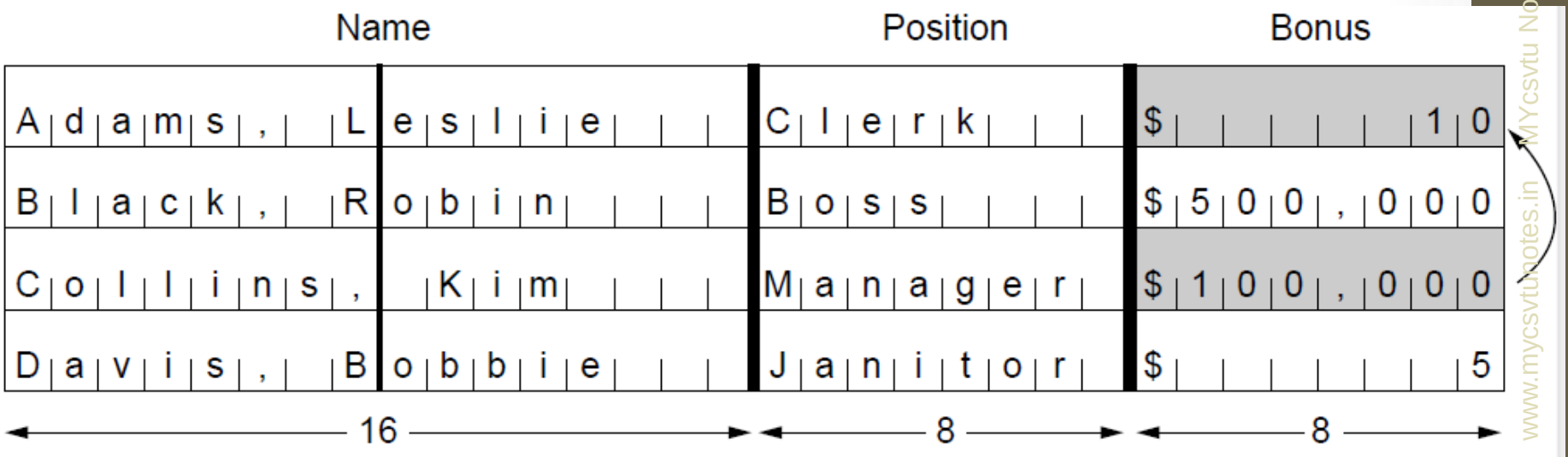

An outline of Rijndael

Advanced Encryption Standard (4)



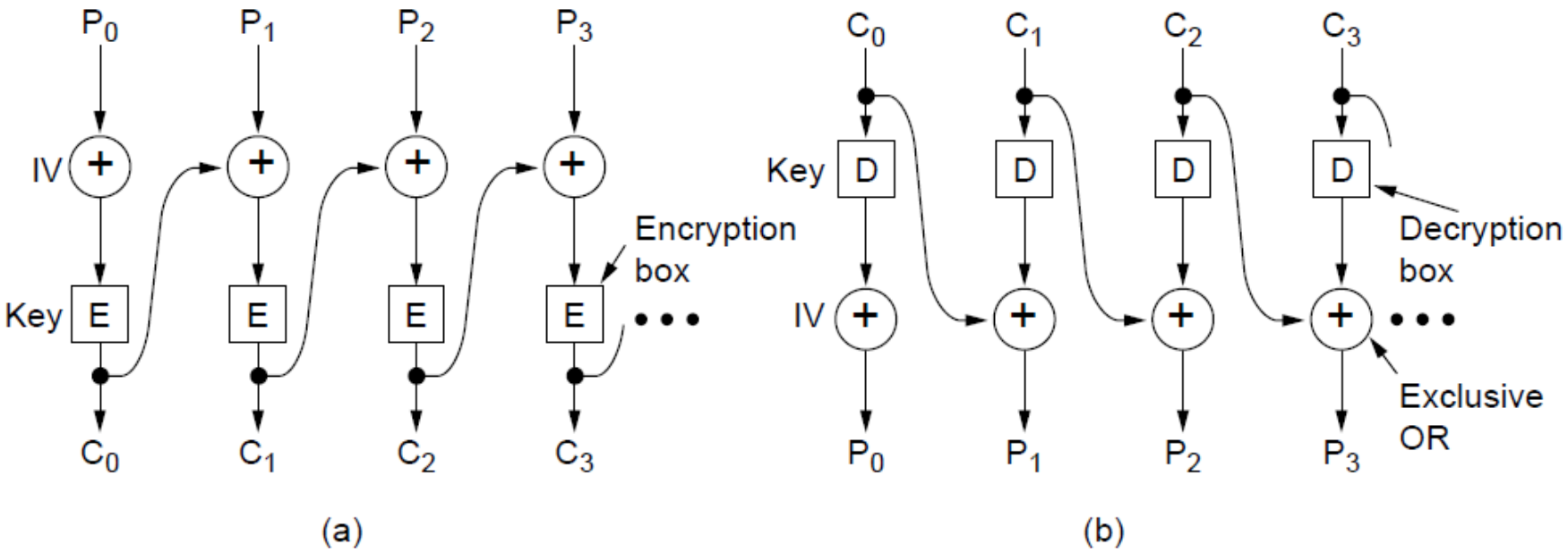
Creating of the *state* and *rk* arrays

Cipher Modes (1)



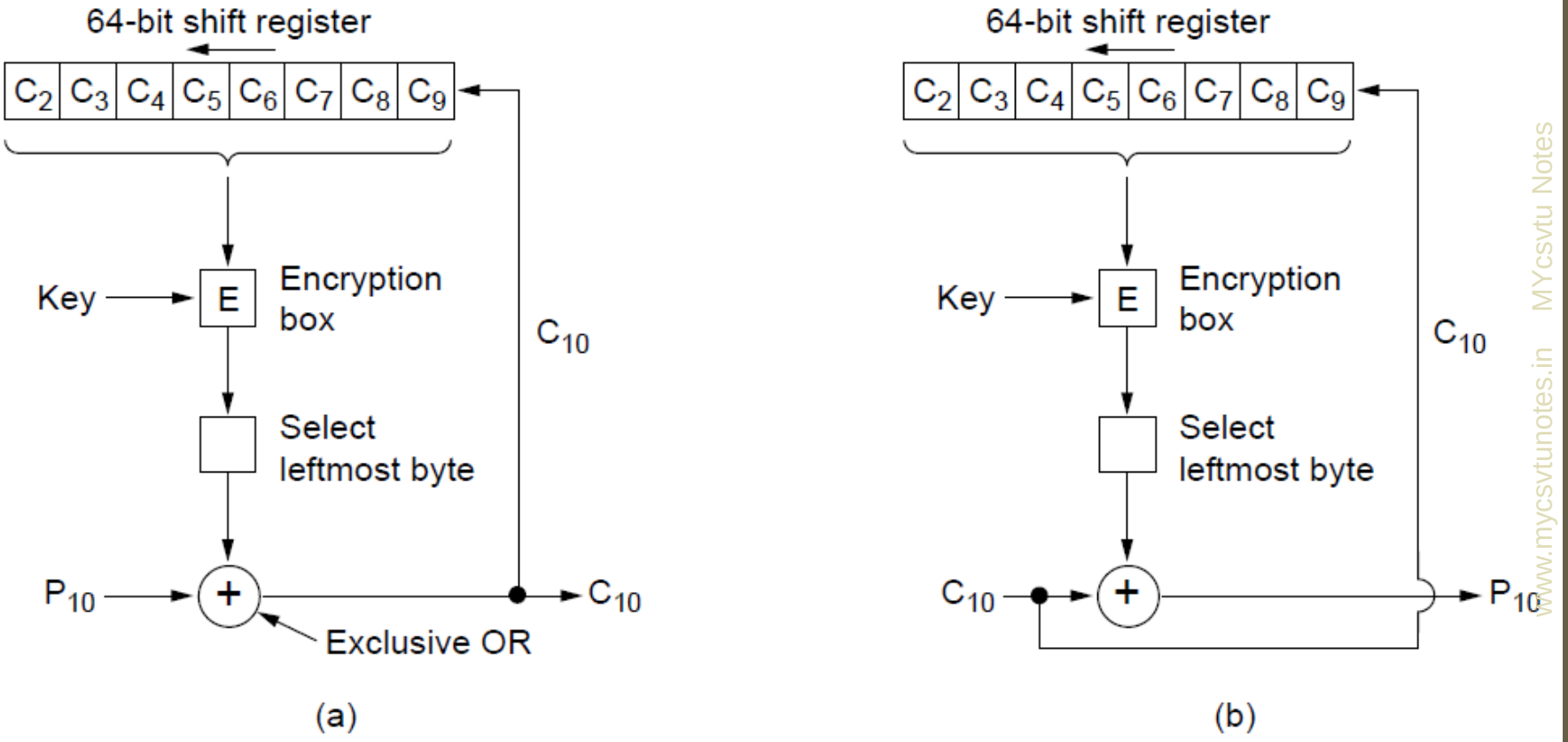
The plaintext of a file encrypted as 16 DES blocks.

Cipher Modes (2)



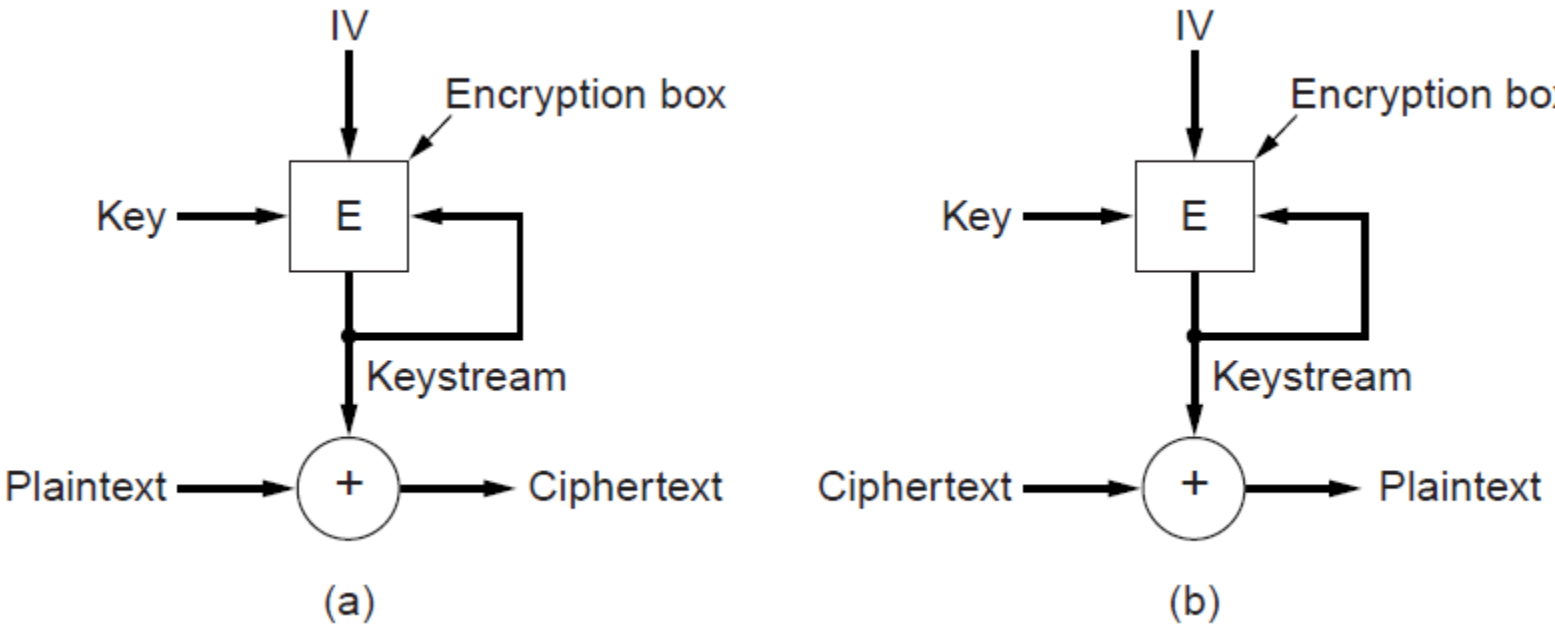
Cipher block chaining. (a) Encryption. (b) Decryption

Cipher Modes (3)



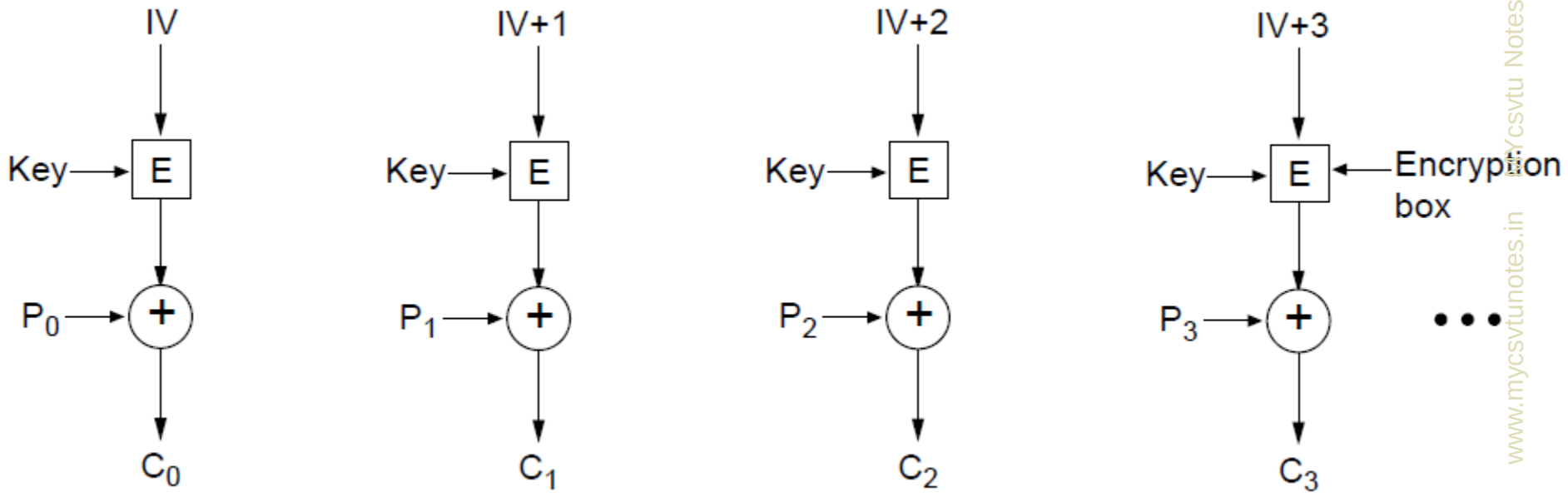
Cipher feedback mode. (a) Encryption. (b) Decryption

Cipher Modes (4)



A stream cipher. (a) Encryption. (b) Decryption

Cipher Modes (5)



Encryption using counter mode

Other Ciphers

Cipher	Author	Key length	Comments
Blowfish	Bruce Schneier	1–448 bits	Old and slow
DES	IBM	56 bits	Too weak to use now
IDEA	Massey and Xuejia	128 bits	Good, but patented
RC4	Ronald Rivest	1–2048 bits	Caution: some keys are weak
RC5	Ronald Rivest	128–256 bits	Good, but patented
Rijndael	Daemen and Rijmen	128–256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	128–256 bits	Very strong
Triple DES	IBM	168 bits	Second best choice
Twofish	Bruce Schneier	128–256 bits	Very strong; widely used

Some common symmetric-key cryptographic algorithms

Public-key Algorithms

- RSA
  - Authors: *R*ivest, *S*hamir, *A*dleman
- Other Public-Key Algorithms

RSA (1)

Method Summary

1. Choose two large primes, *p* and *q*
2. Compute
   $n = p \times q$ and $z = (p - 1) \times (q - 1)$.
3. Choose number relatively prime to *z* call it *d.*
4. Find *e* such that $e \times d = 1 \bmod z$.

RSA (2)

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E
		Sender's computation		Receiver's computation		

An example of the RSA algorithm

Digital Signatures (1)

The authenticity of many legal, financial, and other documents is determined by the presence or absence of an authorized handwritten signature and photocopies do not count.

For computerized message systems to replace the physical transport of paper and ink documents, a method must be found to allow documents to be signed in an unforgeable way.

Required Conditions:

1. Receiver can verify claimed identity of sender.
2. Sender cannot later repudiate contents of message.
3. Receiver cannot have concocted message himself.

Digital Signatures (2)

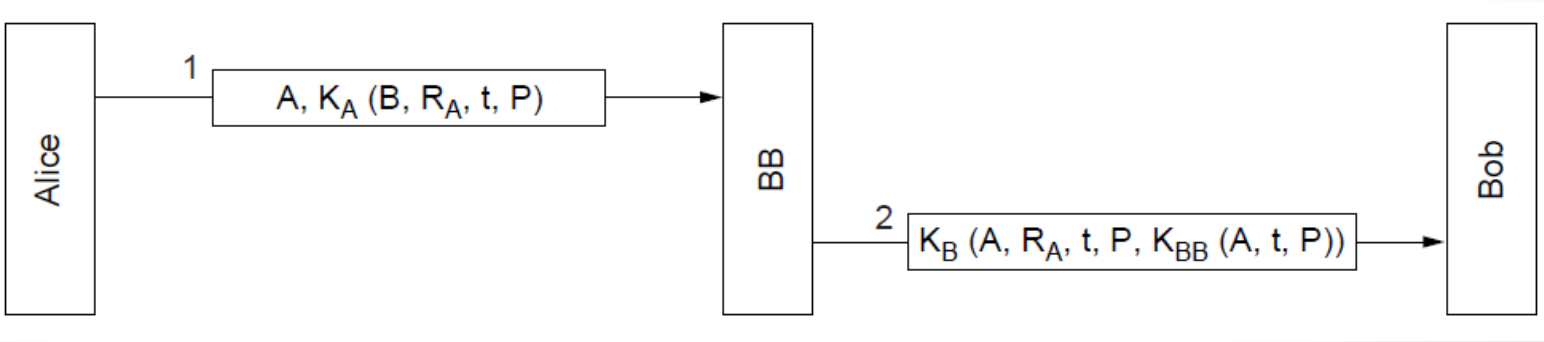
- Symmetric-key signatures
- Public-key signatures
- Message digests
- The birthday attack

Symmetric-key Signatures

One approach to digital signatures is to have a central authority that knows everything and whom everyone trusts, say Big Brother (BB).
Each user then chooses a secret key and carries it by hand to BB's office. Thus, only Alice and BB know Alice's secret key, K_A, and so on.

When Alice wants to send a signed plaintext message, P, to her banker, Bob, she generates $K_A(B, R_A, t, P)$, where B is Bob's identity, R_A is a random number chosen by Alice, t is a timestamp to ensure freshness, and $K_A(B, R_A, t, P)$ is the message encrypted with her key, K_A. Then she sends it as depicted in fig.

BB sees that the message is from Alice, decrypts it, and sends a message to Bob as shown. The message to Bob contains the plaintext of Alice's message and also the signed message K_{BB} (A, t, P). Bob now carries out Alice's request.



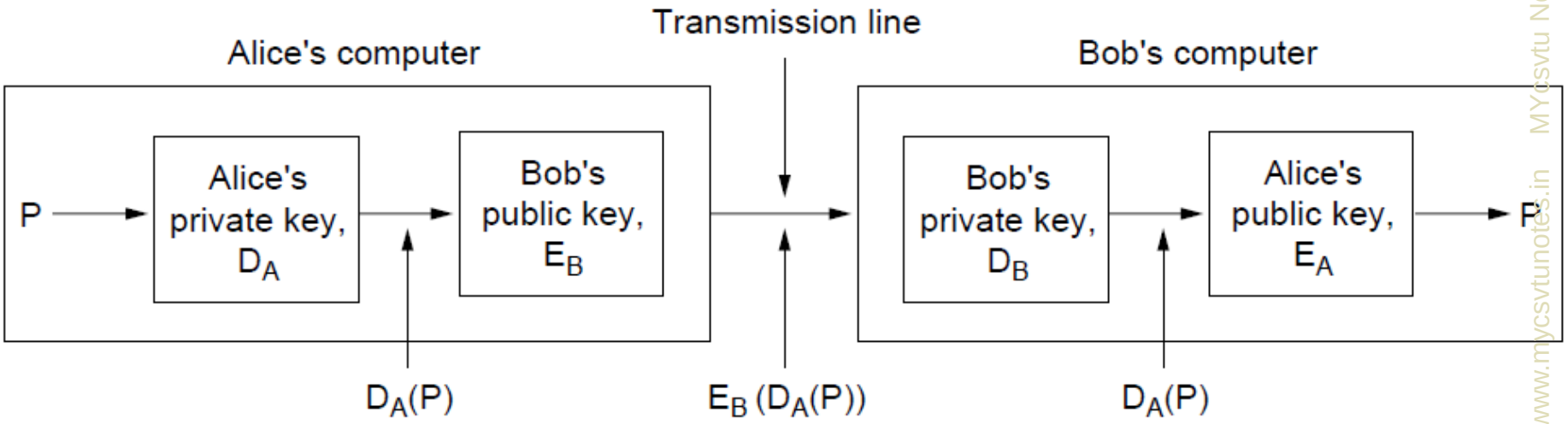
Problem with symmetric key cryptography:

- A structural problem with using symmetric-key cryptography for digital signatures is that everyone has to agree to trust Big Brother.
- Furthermore, Big Brother gets to read all signed messages.
- The most logical candidates for running the Big Brother server are the government, the banks, the accountants, and the lawyers.
- Unfortunately, none of these organizations inspire total confidence in all citizens. Hence, it would be nice if signing documents did not require a trusted authority.

Public-Key Signatures

- Public-key cryptography can make an important contribution in this area.
- The public-key encryption and decryption algorithms have the property that E(D(P)) = P in addition, of course, to the usual property that D(E(P)) = P.
- Alice can send a signed plaintext message, P, to Bob by transmitting $E_B(D_A(P))$.
- Note carefully that Alice knows her own (private) key, D_A, as well as Bob's public key, E_B, so constructing this message is something Alice can do.
- When Bob receives the message, he transforms it using his private key, as usual, yielding $D_A(P)$,

Public-Key Signatures



Digital signatures using public-key cryptography.

Problem with public key signature

- Although using public-key cryptography for digital signatures is an elegant scheme, there are problems that are related to the environment in which they operate rather than with the basic algorithm.
- For one thing, Bob can prove that a message was sent by Alice only as long as D_A remains secret.
- If Alice discloses her secret key, the argument no longer holds, because anyone could have sent the message, including Bob himself.

- The problem might arise, for example, if Bob is Alice's stockbroker.
- Alice tells Bob to buy a certain stock or bond. Immediately thereafter, the price drops sharply.
- To repudiate her message to Bob, Alice runs to the police claiming that her home was burglarized and the PC holding her key was stolen.
- Depending on the laws in her state or country, she may or may not be legally liable, especially if she claims not to have discovered the break-in until getting home from work, several hours later.

- Another problem with the signature scheme is what happens if Alice decides to change her key.
- Doing so is clearly legal, and it is probably a good idea to do so periodically.
- If a court case later arises, as described above, the judge will apply the current E_A to $D_A(P)$ and discover that it does not produce P.
- Bob will look pretty stupid at this point.

Public-Key Signatures (2)

Criticisms of DSS(Digital Signature Standard)

1) Too secret
2) Too slow
3) Too new
4) Too insecure

Message Digests (1)

One criticism of signature methods is that they often couple two distinct functions: authentication and secrecy.

Often, authentication is needed but secrecy is not.

An authentication scheme that does not require encrypting the entire message is more useful.

This scheme is based on the idea of a one-way hash function that takes an arbitrarily long piece of plaintext and from it computes a fixed-length bit string. This hash function, MD, often called a message digest, has four important properties.

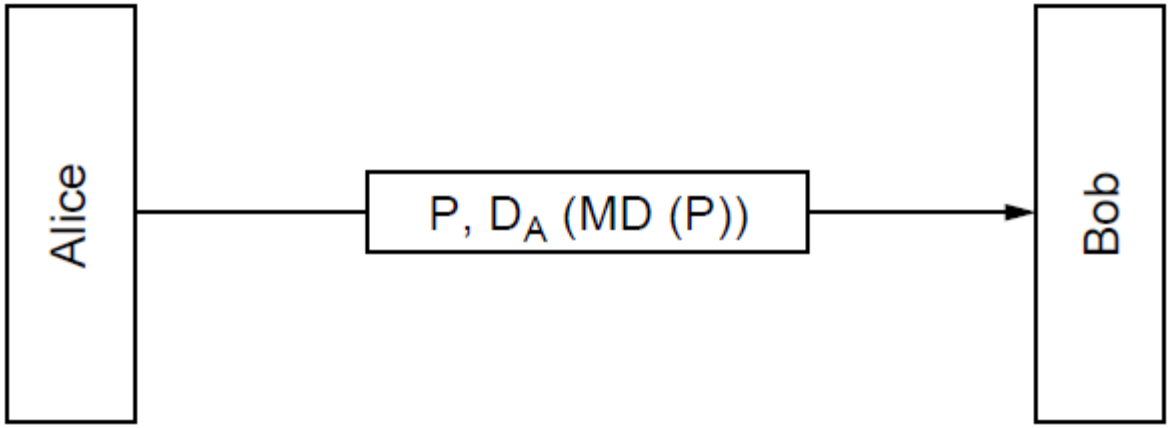
Message Digest properties:

1. Given *P,* easy to compute *MD(P).*
2. Given *MD(P),* effectively impossible to find *P.*
3. Given *P* no one can find *P'* such that *MD(P') = MD(P).*
4. Change to input of even 1 bit produces very different output.

To meet criterion 3, the hash should be at least 128 bits long, preferably more.

To meet criterion 4, the hash must mangle the bits very thoroughly

Message Digests (2)

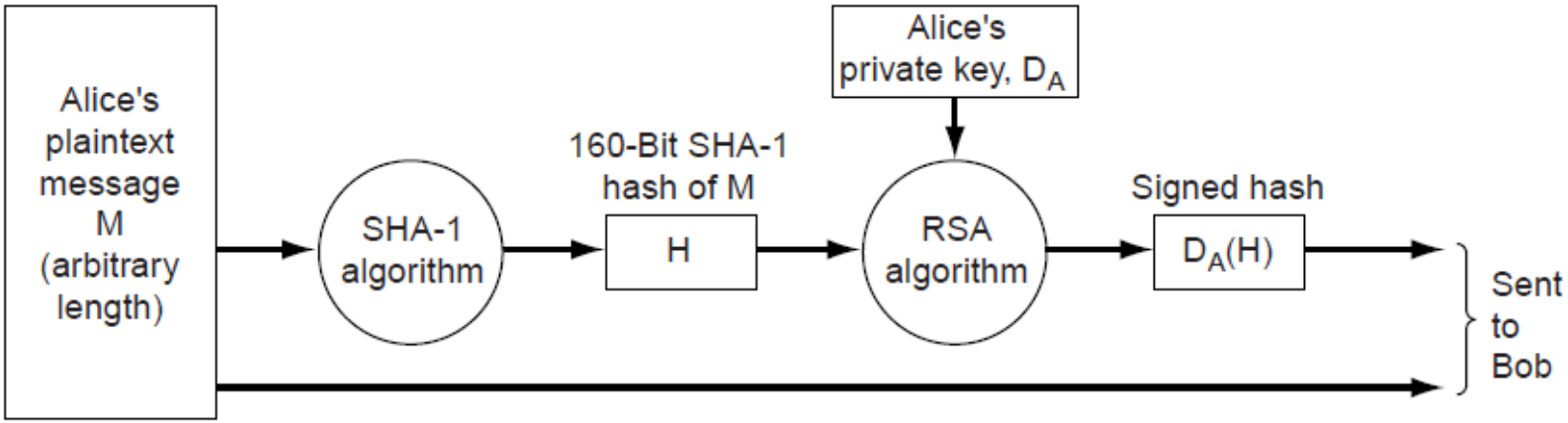


Digital signatures using message digests

SHA-1 (Secure Hash Algorithm 1)

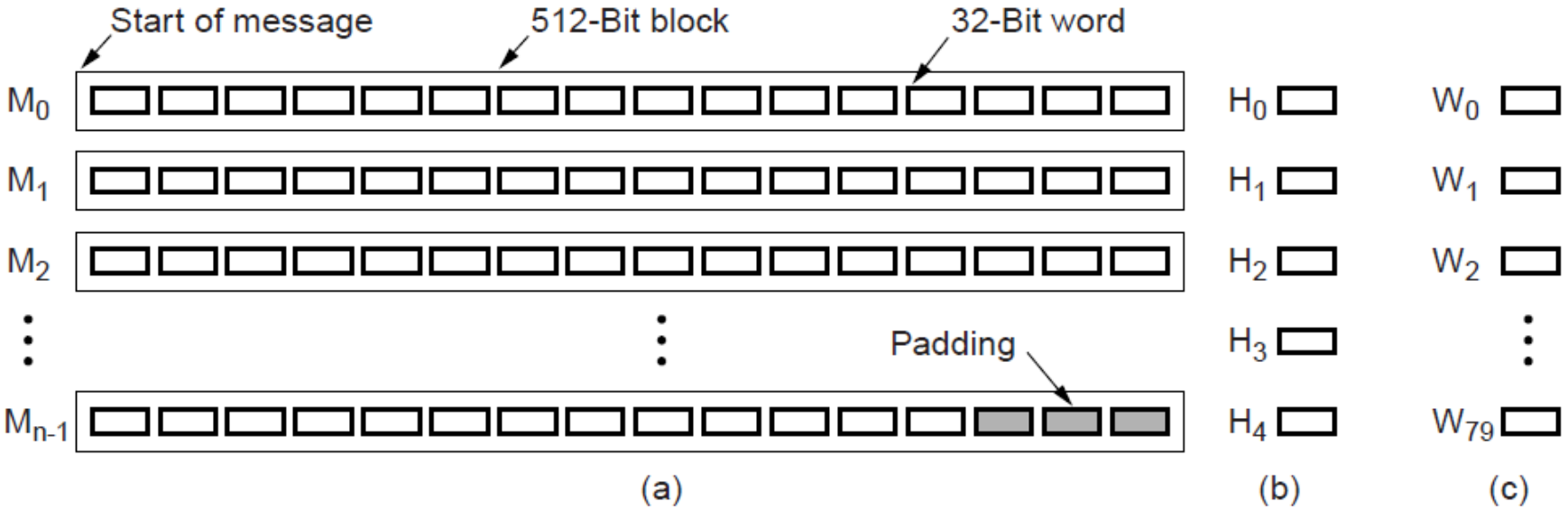
- The major message digest function is SHA-1 (Secure Hash Algorithm 1), developed by NSA and blessed by NIST in FIPS 180-1. Like MD5, SHA-1 processes input data in 512-bit blocks, only unlike MD5, it generates a 160-bit message digest.
- A typical way for Alice to send a non-secret but signed message to Bob is illustrated in fig.
- Here her plaintext message is fed into the SHA-1 algorithm to get a 160-bit SHA-1 hash. Alice then signs the hash with her RSA private key and sends both the plaintext message and the signed hash to Bob.

Message Digests (3)



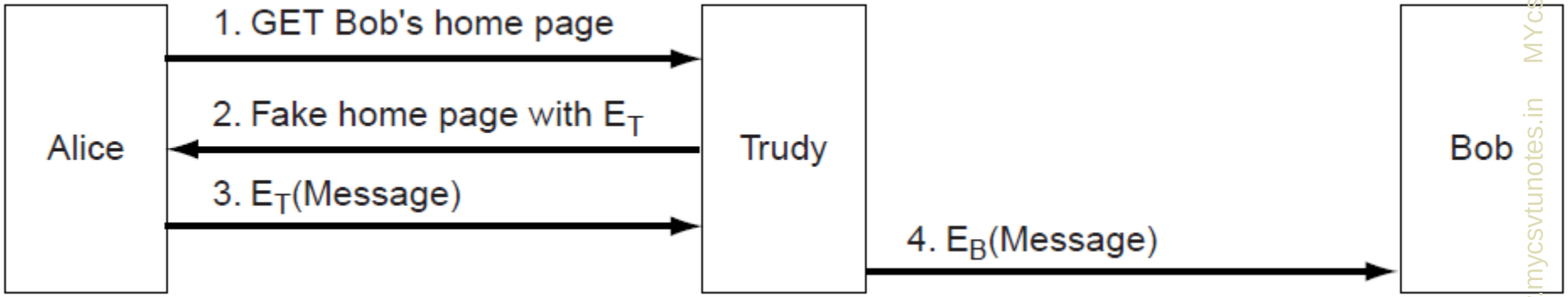
Use of SHA-1 and RSA for signing non-secret messages

Message Digests (4)



(a) A message padded out to a multiple of 512 bits.
(b) The output variables.
(c) The word array.

Management of Public Keys (1)



A way for Trudy to subvert public-key encryption

Management of Public Keys (2)

- Certificates
- X.509
- Public key infrastructures

Certificates

I hereby certify that the public key
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
Robert John Smith
12345 University Avenue
Berkeley, CA 94702
Birthday: July 4, 1958
Email: bob@superdupernet.com

SHA-1 hash of the above certificate signed with the CA's private key

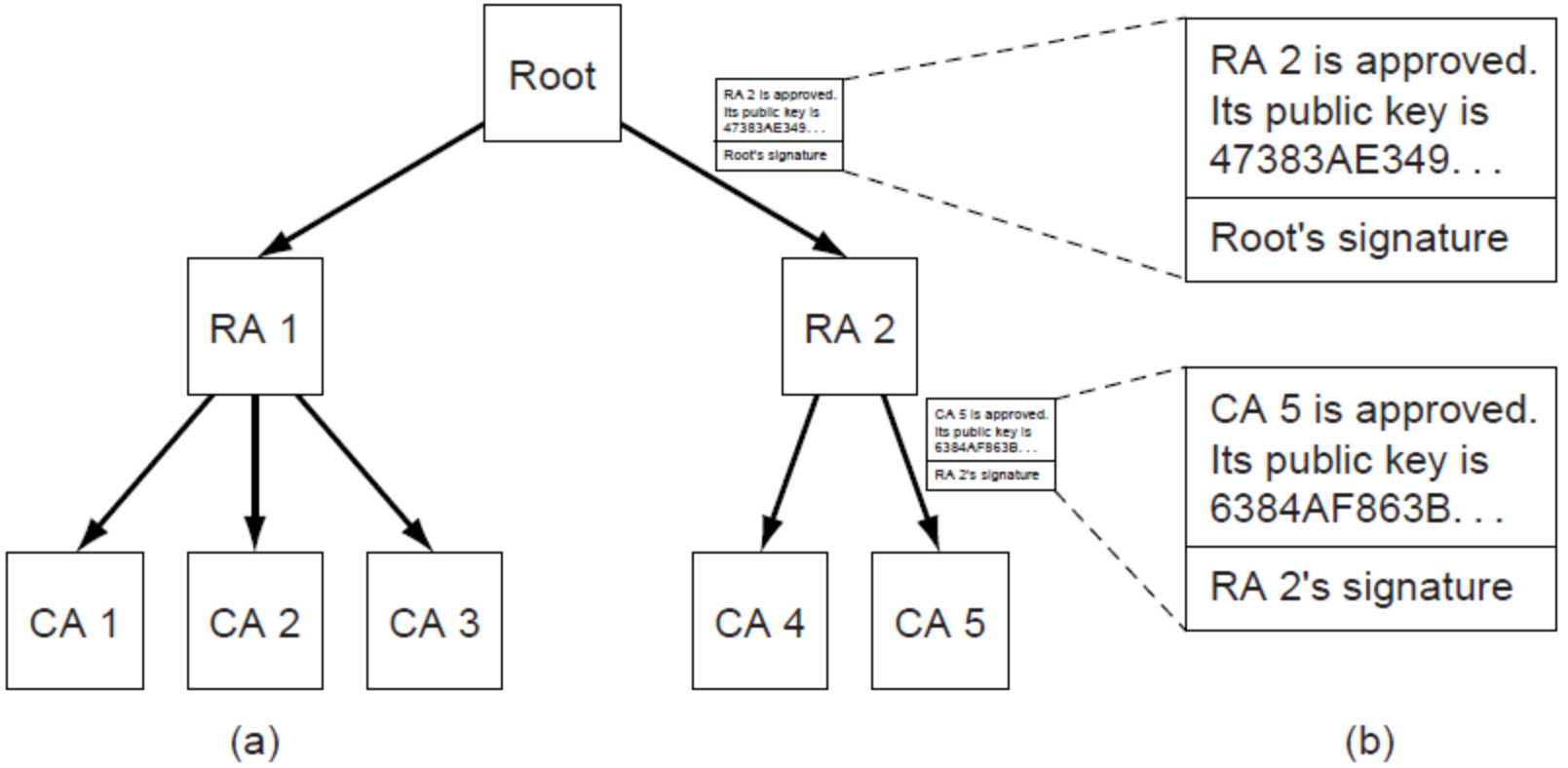
A possible certificate and its signed hash

X.509

Field	Meaning
Version	Which version of X.509
Serial number	This number plus the CA's name uniquely identifies the certificate
Signature algorithm	The algorithm used to sign the certificate
Issuer	X.500 name of the CA
Validity period	The starting and ending times of the validity period
Subject name	The entity whose key is being certified
Public key	The subject's public key and the ID of the algorithm using it
Issuer ID	An optional ID uniquely identifying the certificate's issuer
Subject ID	An optional ID uniquely identifying the certificate's subject
Extensions	Many extensions have been defined
Signature	The certificate's signature (signed by the CA's private key)

The basic fields of an X.509 certificate

Public Key Infrastructures

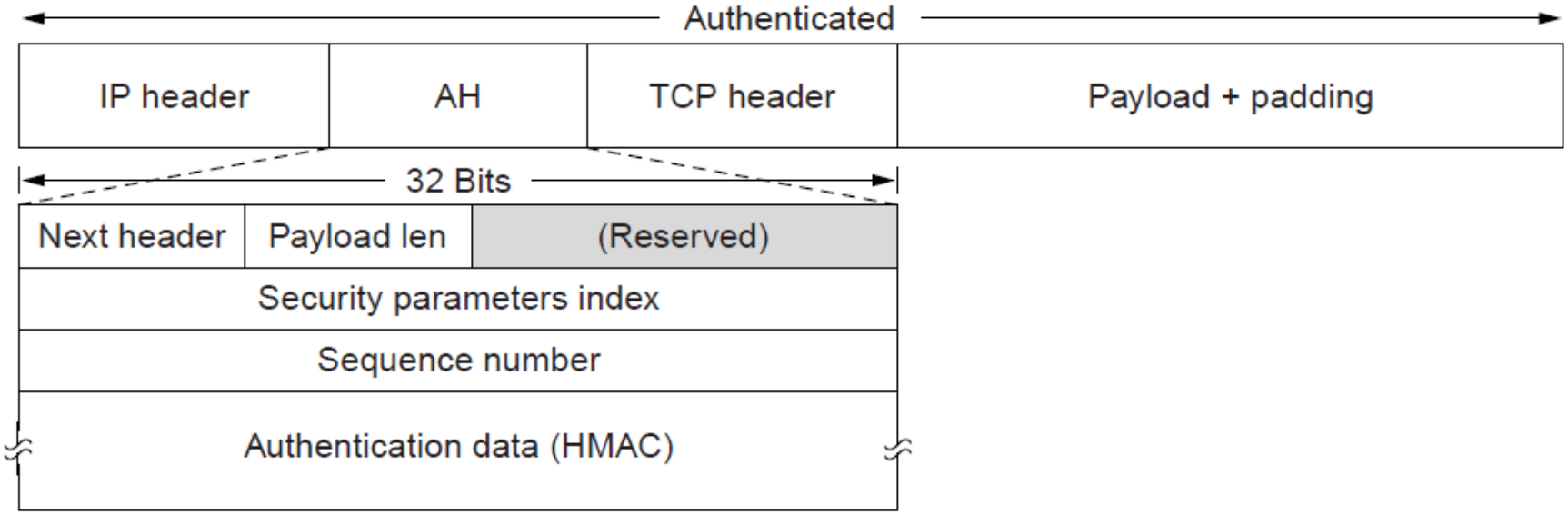


(a) A hierarchical PKI. (b) A chain of certificates.

Communication Security

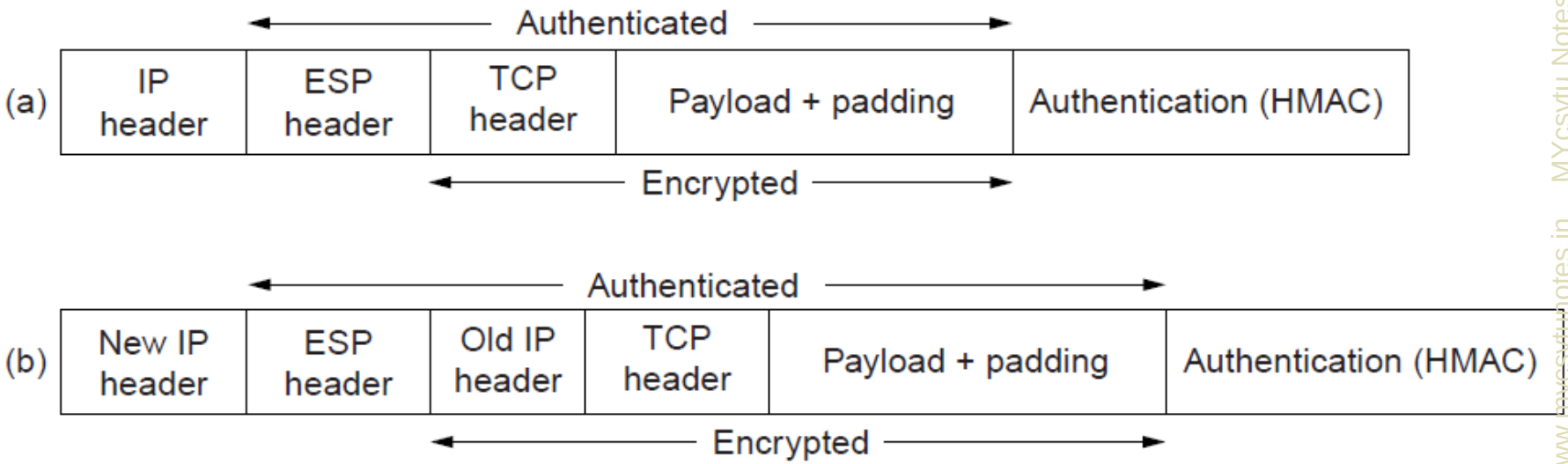
- IPsec
- Firewalls
- Virtual private networks
- Wireless security

IPsec (1)



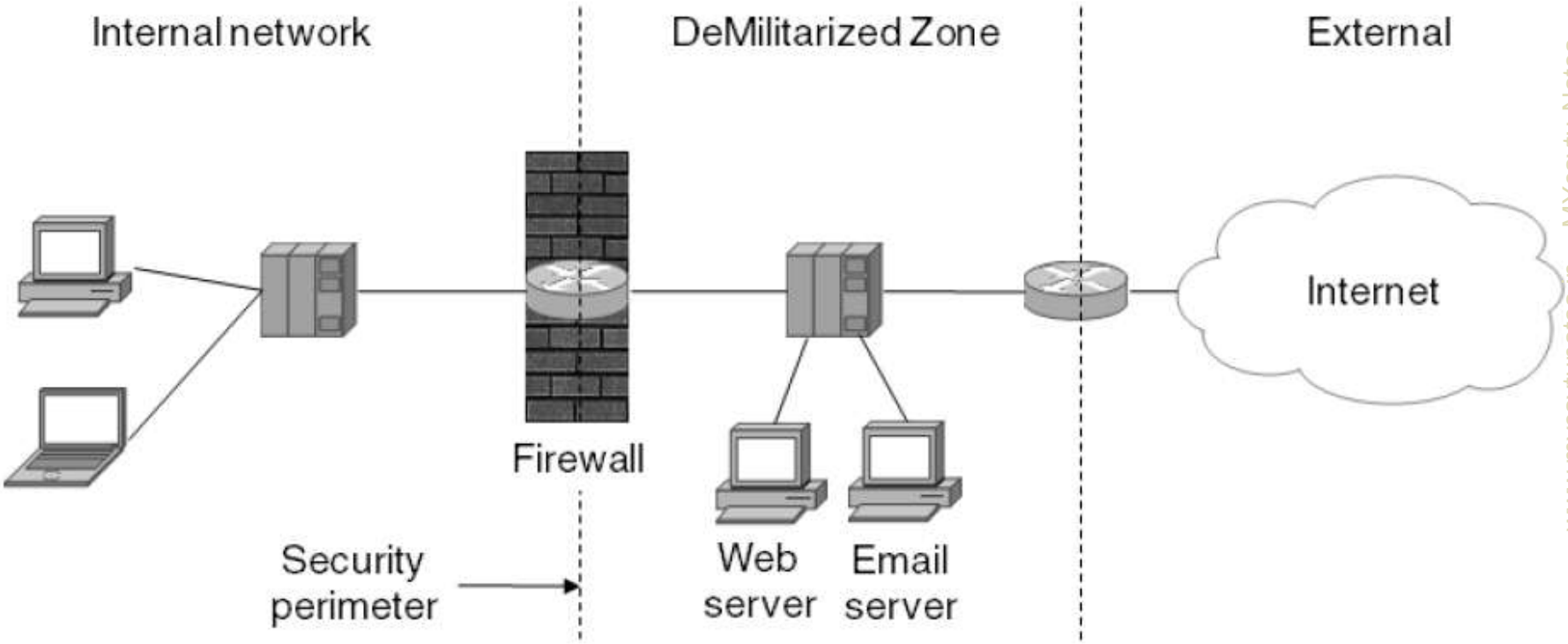
The IPsec authentication header in transport mode for IPv4.

IPsec (2)

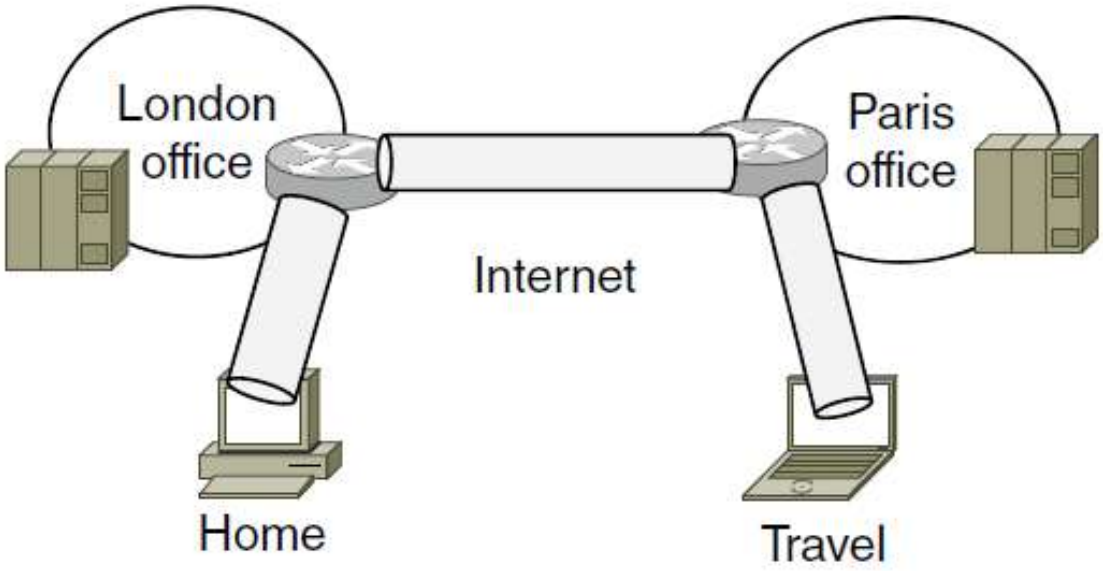


(a) ESP in transport mode. (b) ESP in tunnel mode.

IPsec (3)

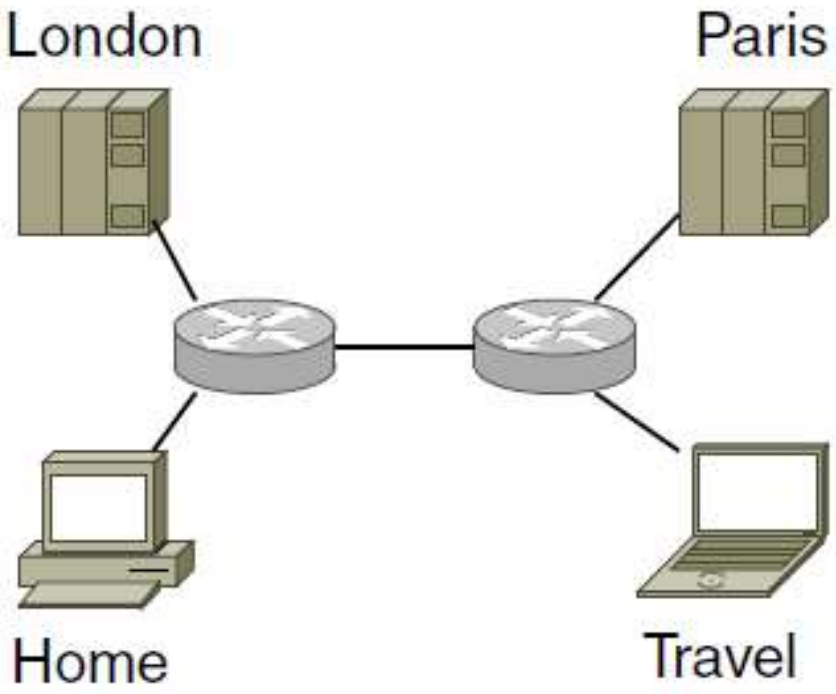


Virtual Private Networks (1)



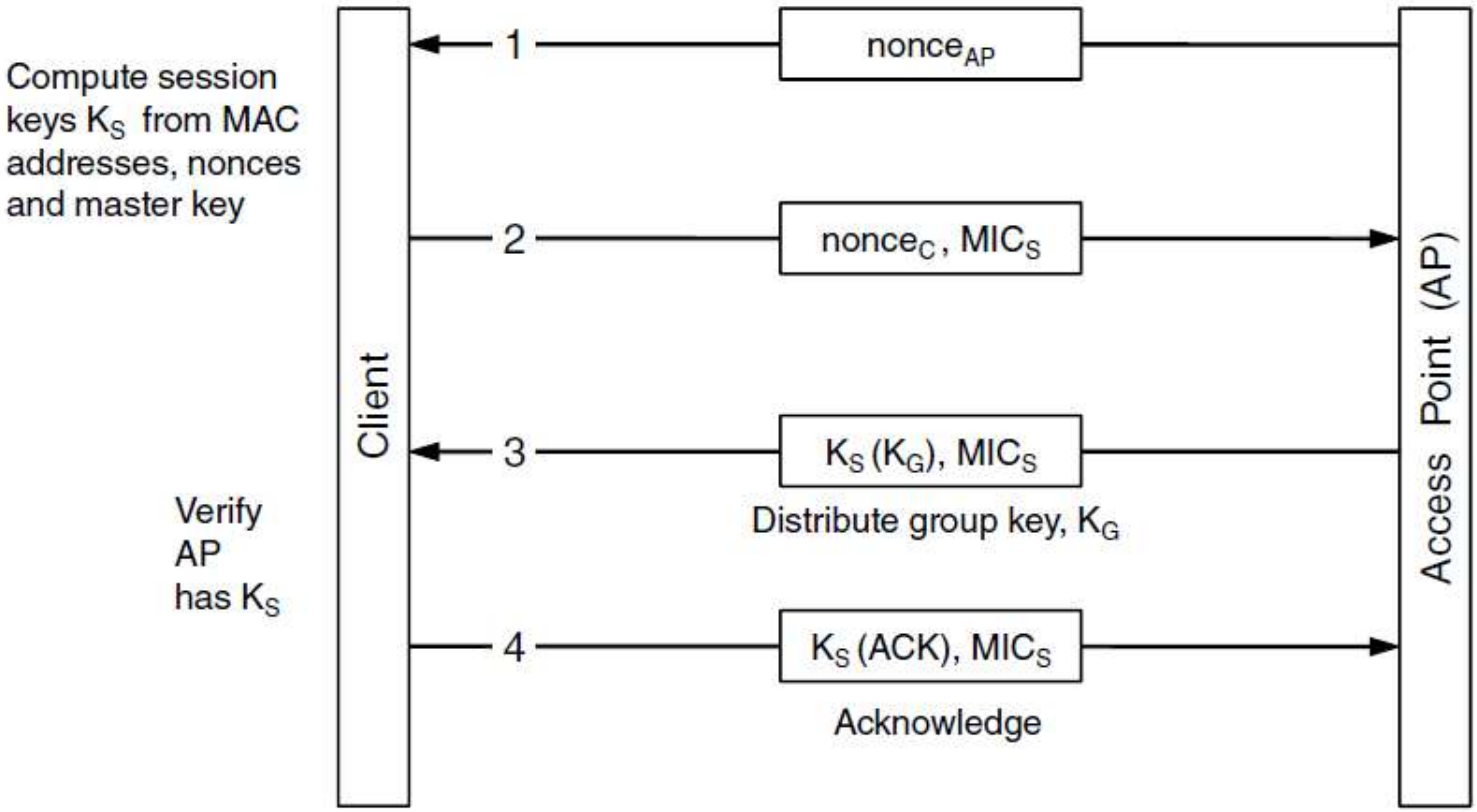
A virtual private network

Virtual Private Networks (2)



Topology as seen from the inside

Wireless Security



The 802.11i key setup handshake

Authentication Protocols

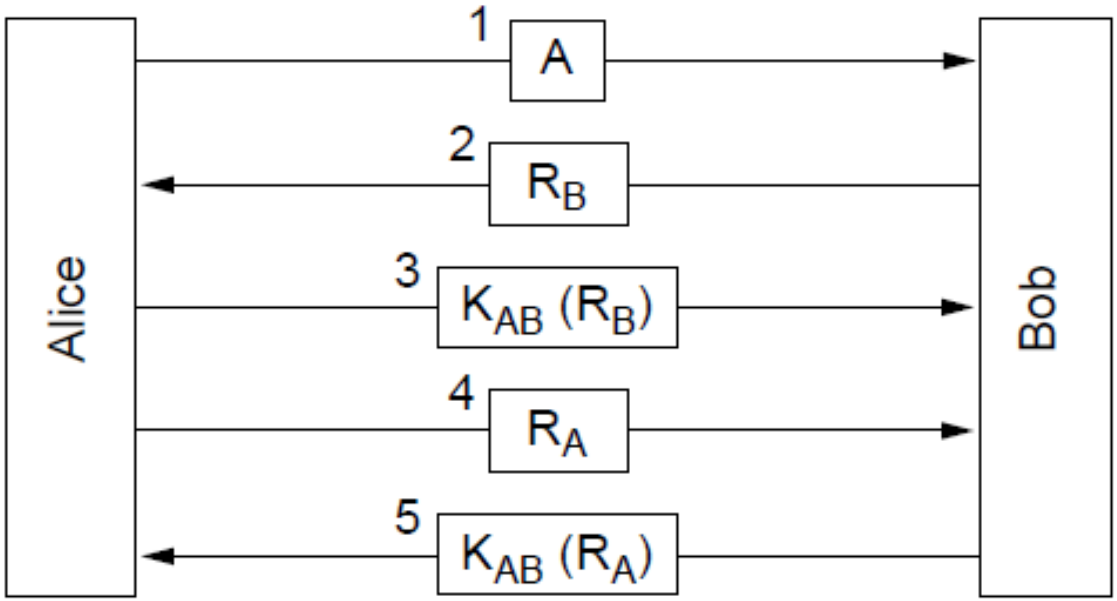
- Shared secret key
- Establishing a shared key: the Diffie-Hellman key exchange
- Key distribution center
- Kerberos
- Public-key cryptography

Shared Secret Key (1)

Notation for discussing protocols

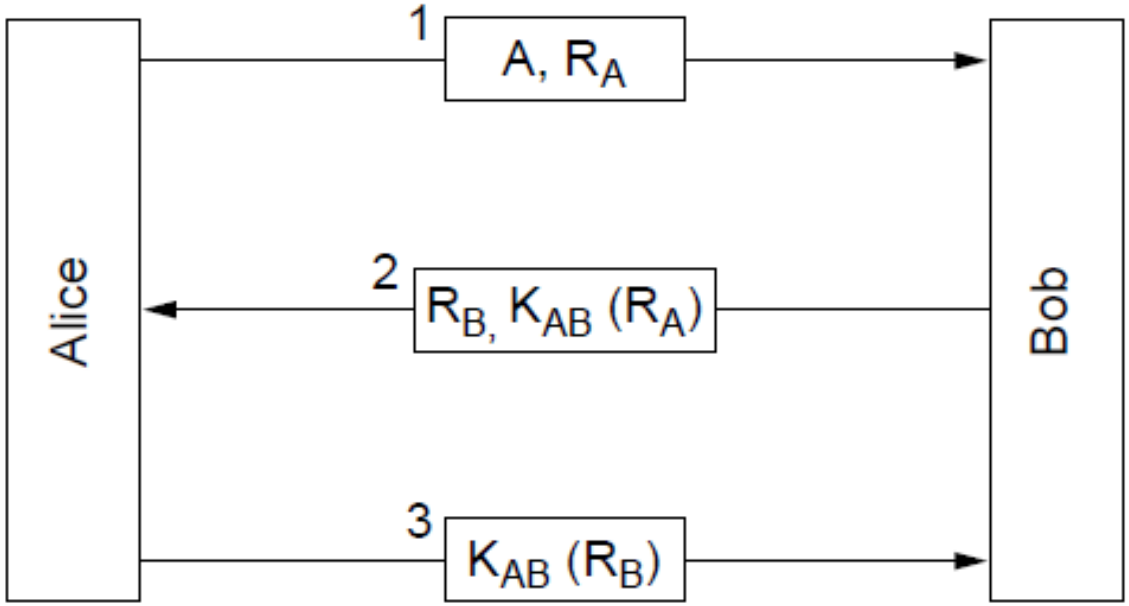
- *A, B* are the identities of Alice and Bob.
- R_i*'s* are the challenges, where the subscript identifies the challenger.
- K_i are keys, where *i* indicates the owner.
- K_S is the session key.

Shared Secret Key (2)



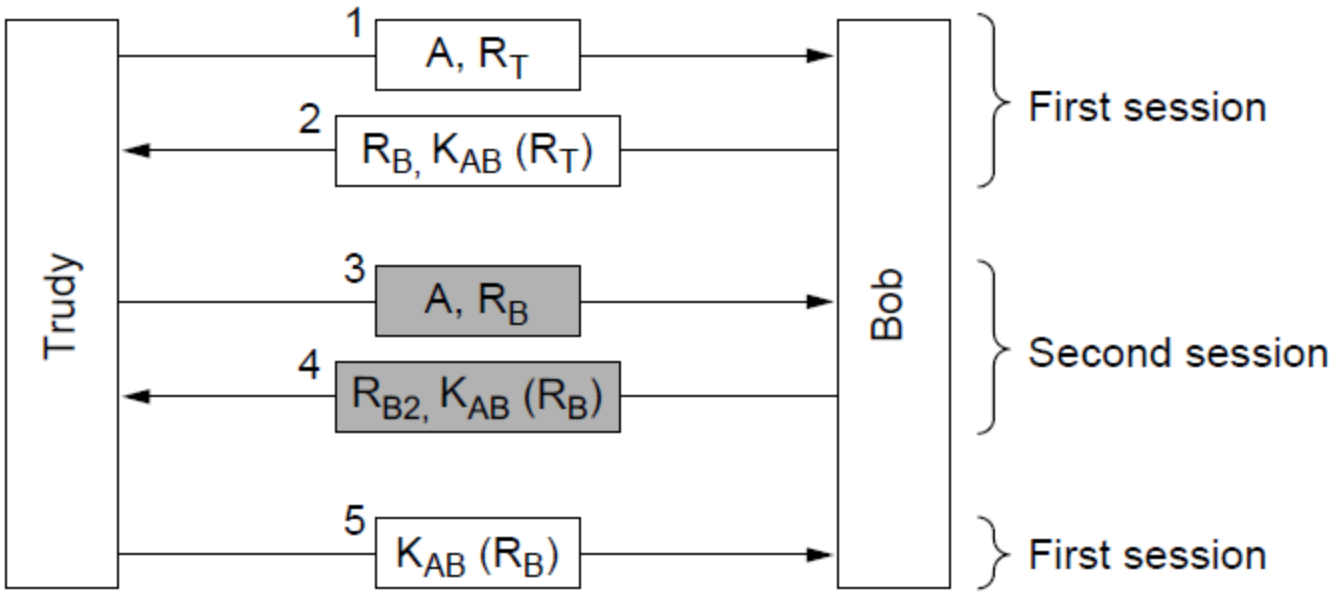
Two-way authentication using a challenge-response protocol.

Shared Secret Key (3)



A shortened two-way authentication protocol

Shared Secret Key (4)



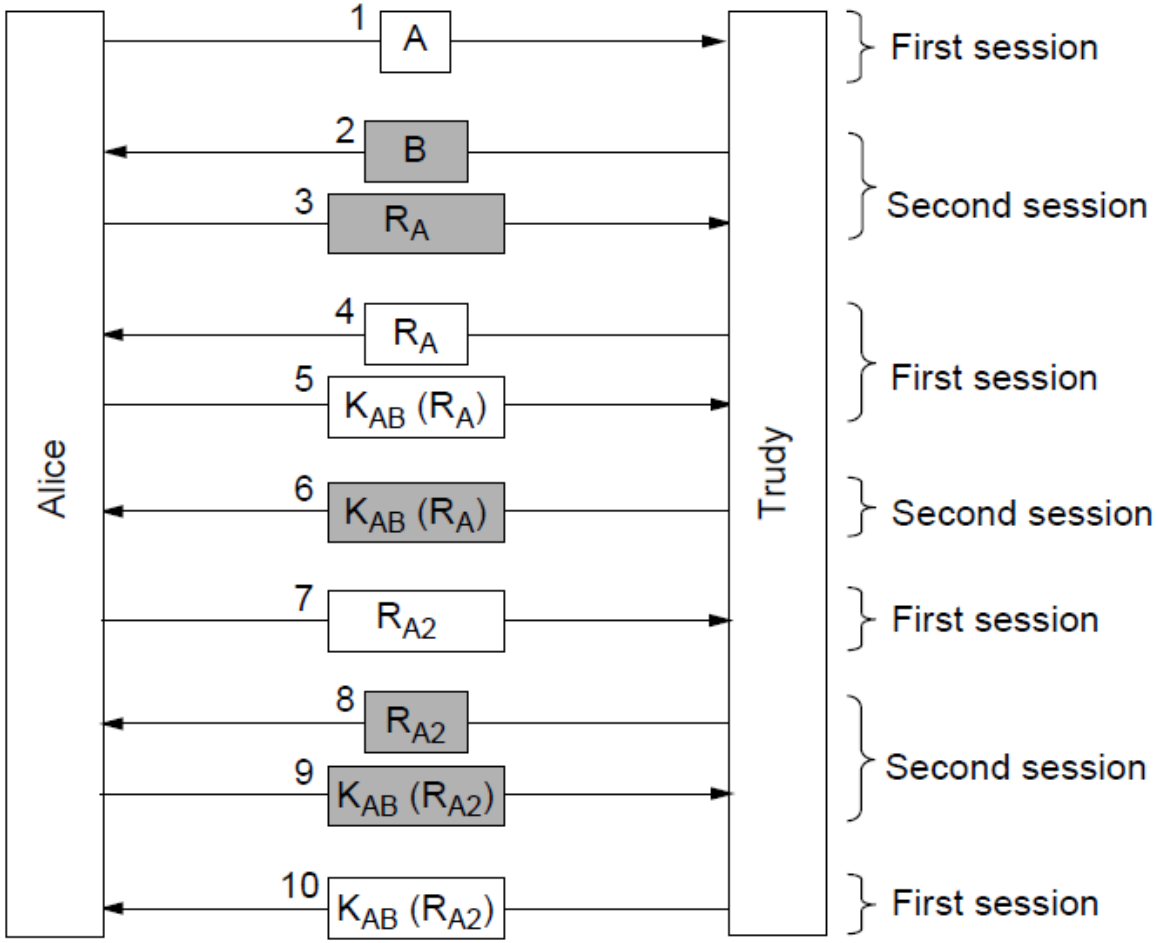
The reflection attack.

Shared Secret Key (5)

General design rules

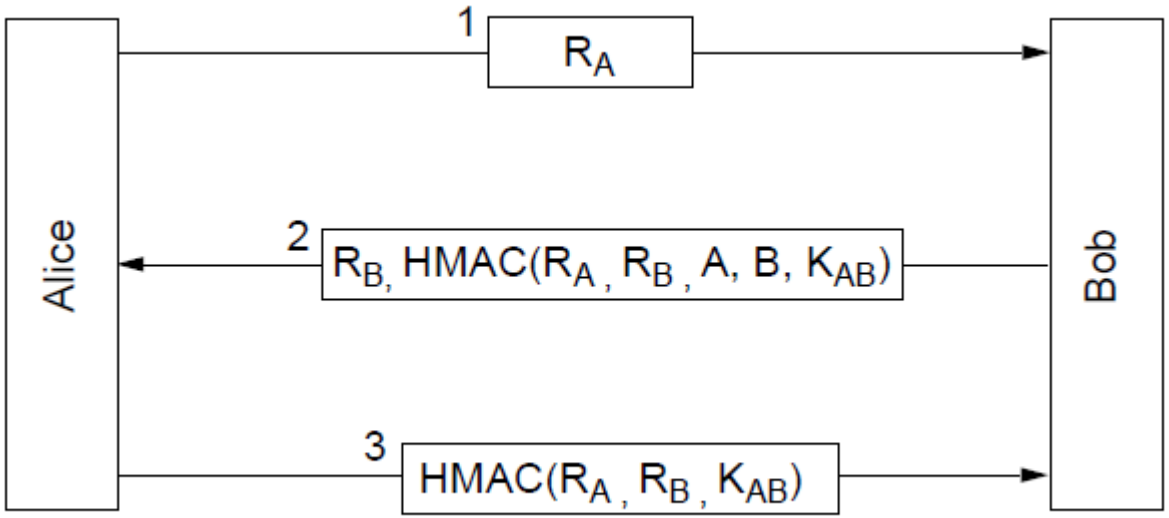
1. Have initiator prove who she is before responder
2. Initiator, responder use different keys
3. Draw challenges from different sets
4. Make protocol resistant to attacks involving second parallel session

Shared Secret Key (6)



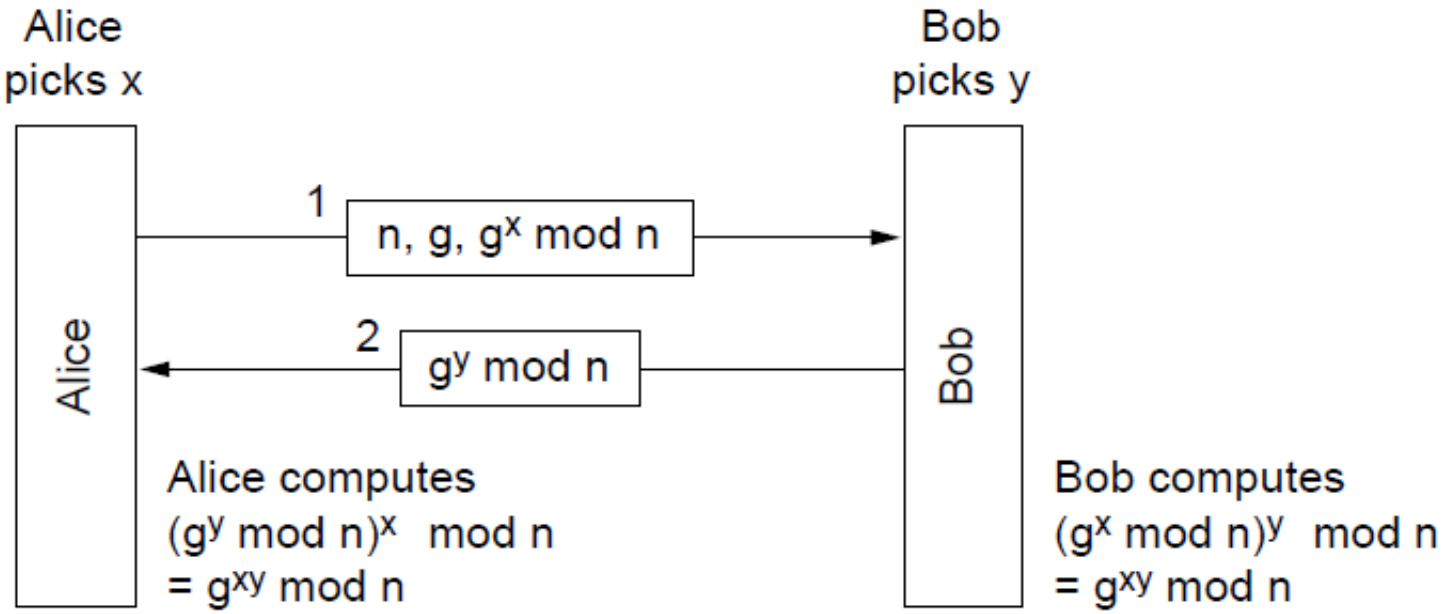
A reflection attack on the protocol of Fig. 8-32

Shared Secret Key (7)



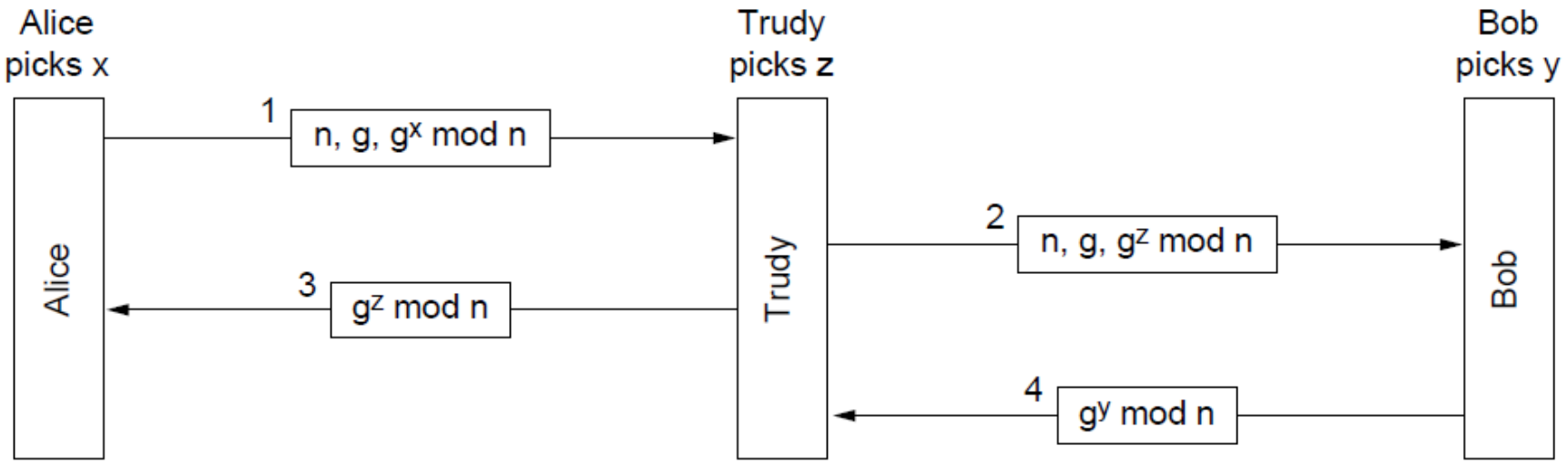
Authentication using HMACs

The Diffie-Hellman Key Exchange (1)



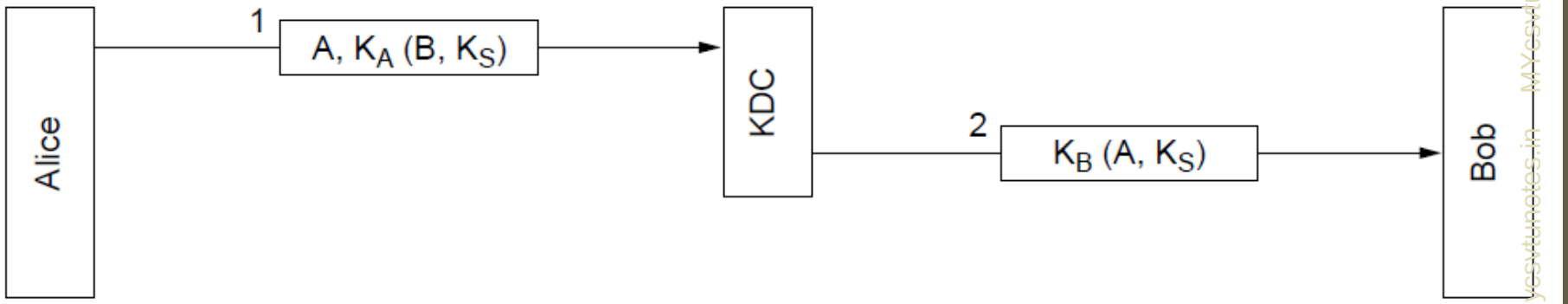
The Diffie-Hellman key exchange

The Diffie-Hellman Key Exchange (2)



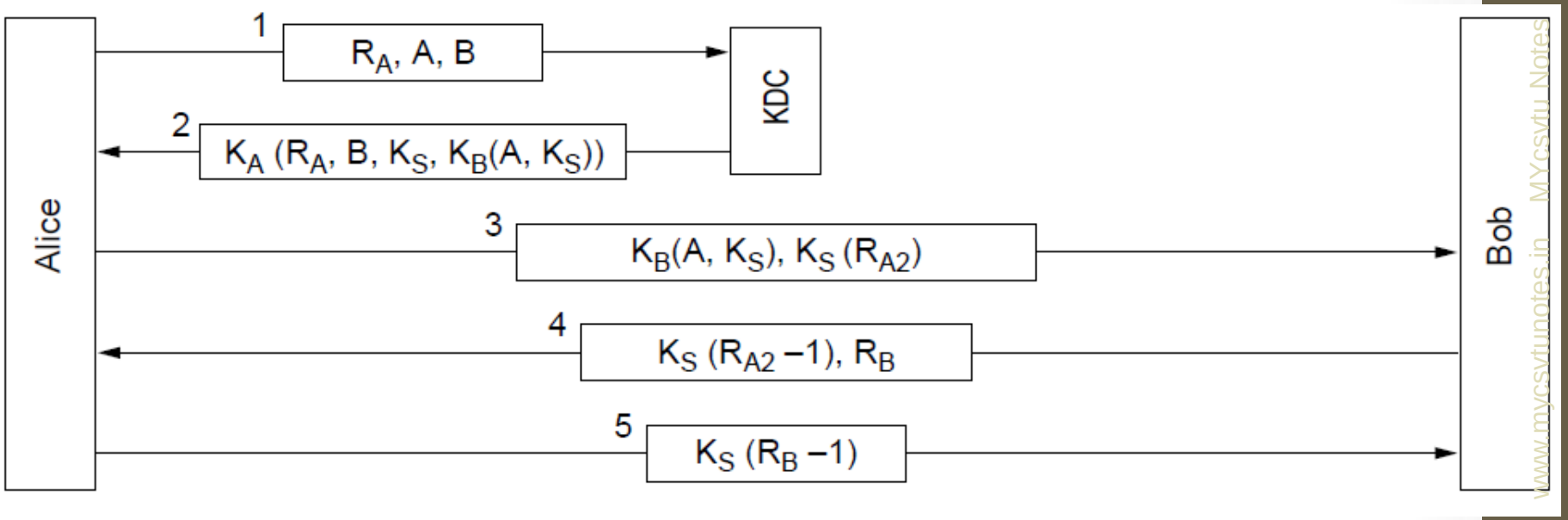
The man-in-the-middle attack

Key Distribution Center (1)



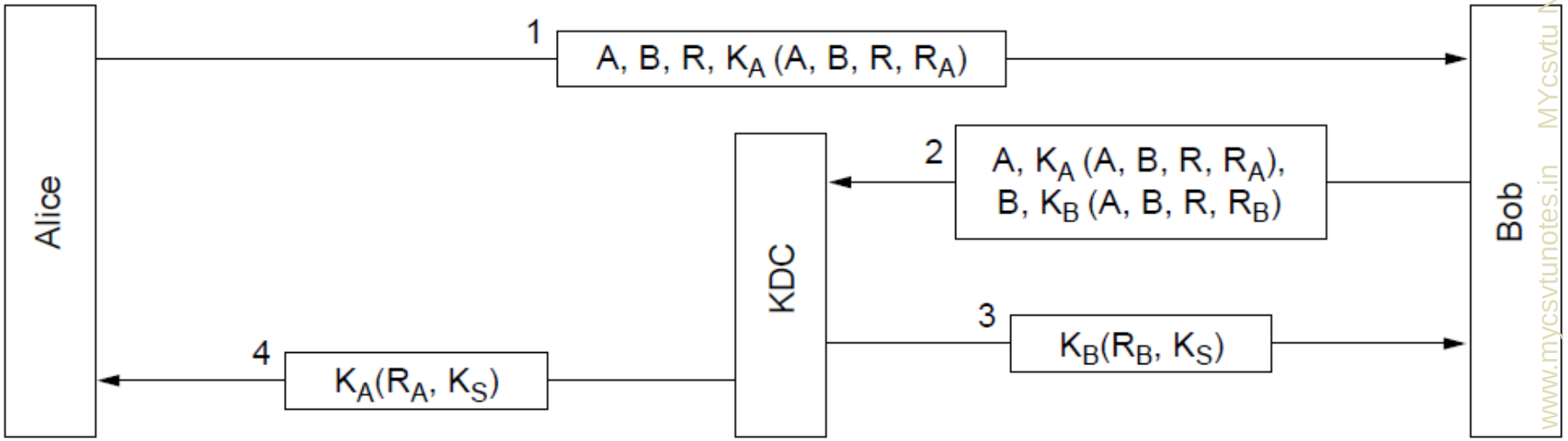
A first attempt at an authentication protocol using a KDC.

Key Distribution Center (2)



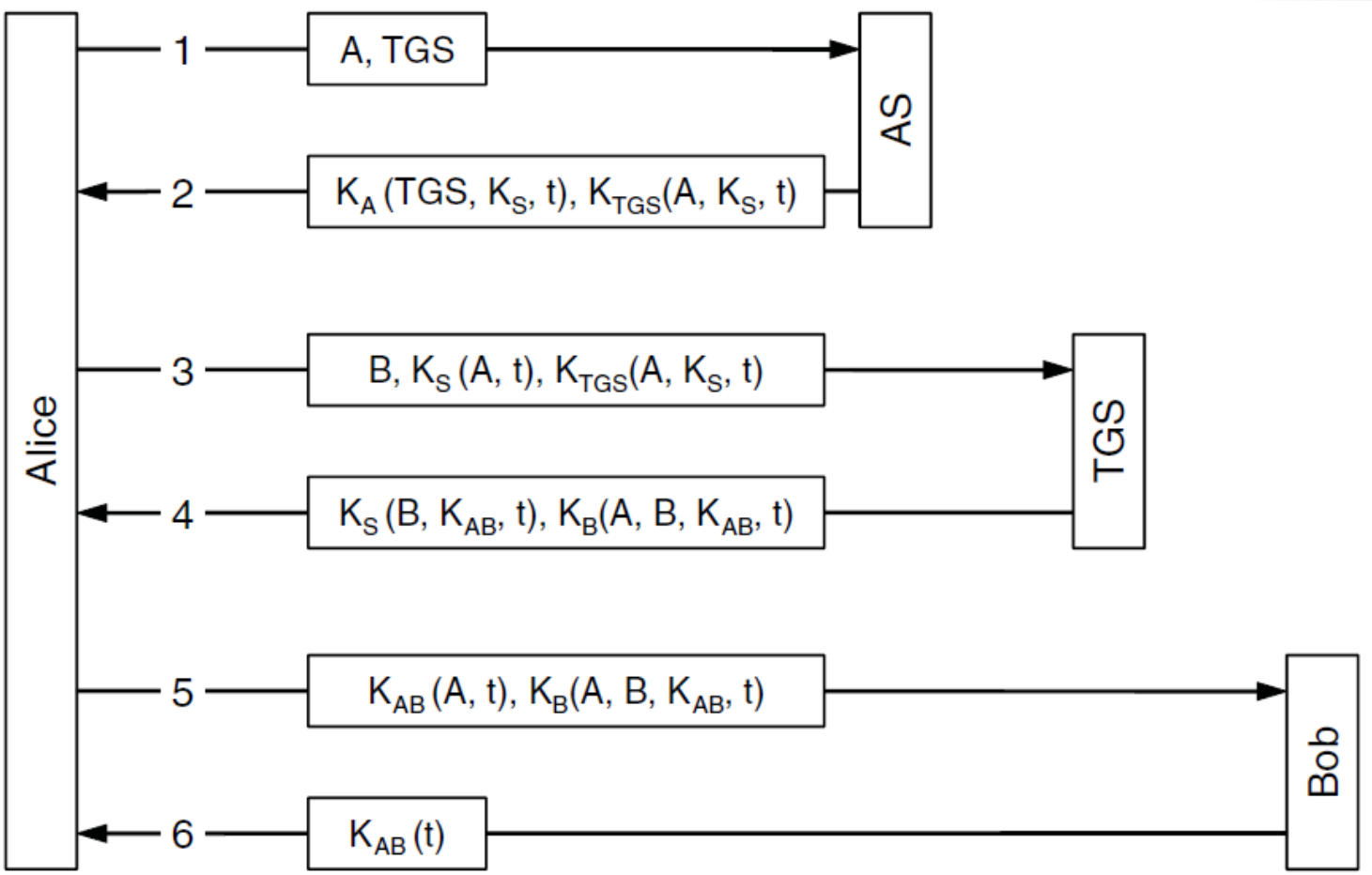
The Needham-Schroeder authentication protocol

Key Distribution Center (3)



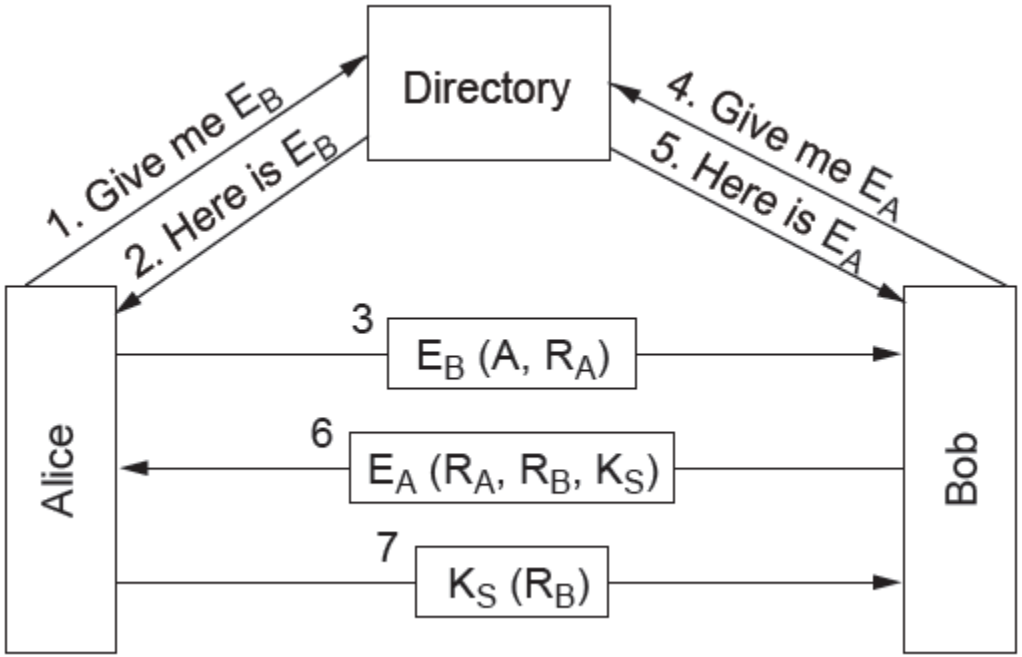
The Otway-Rees authentication protocol (slightly simplified).

Kerberos



The operation of Kerberos V5

Public-Key Cryptography

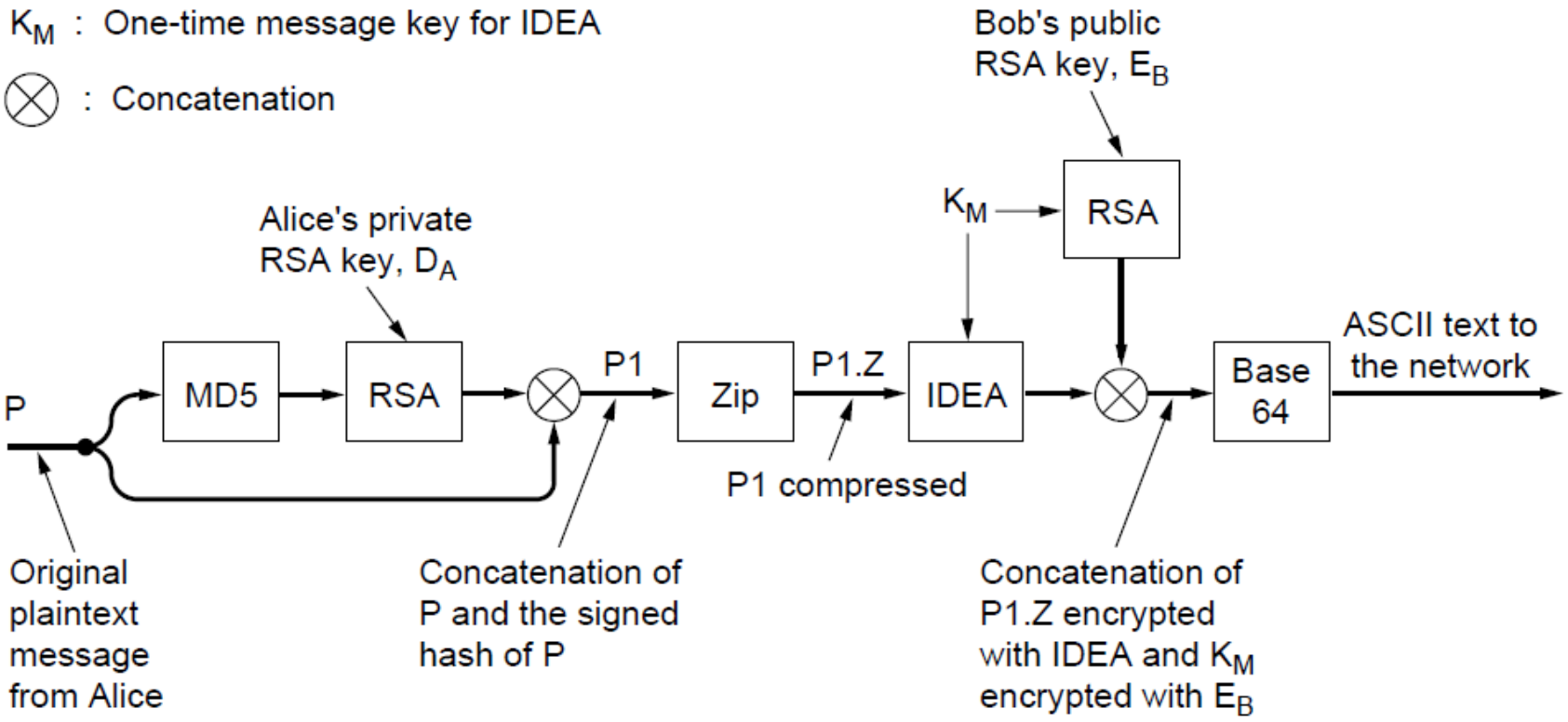


Mutual authentication using public-key cryptography

Email Security

- PGP—Pretty Good Privacy
- S/MIME

PGP—Pretty Good Privacy (1)

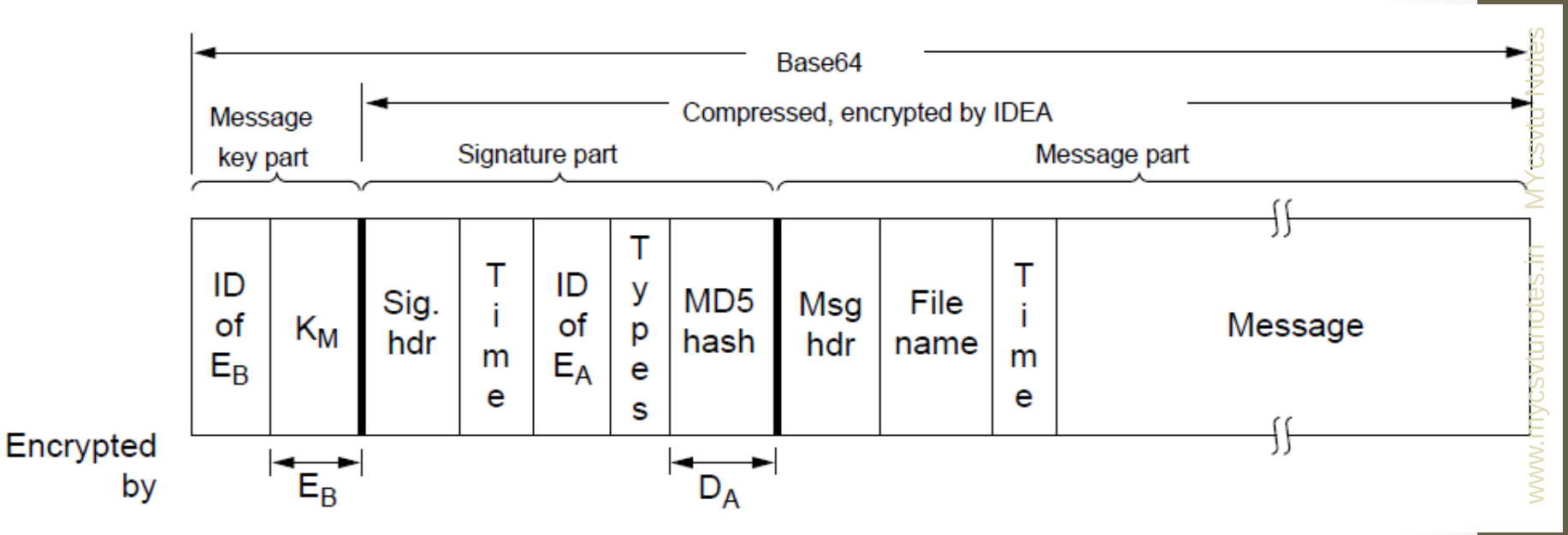


PGP in operation for sending a message

PGP—Pretty Good Privacy (2)

- Casual (384 bits):
  - Can be broken easily today.
- Commercial (512 bits): b
  - Breakable by three-letter organizations.
- Military (1024 bits):
  - Not breakable by anyone on earth.
- Alien (2048 bits):
  - Unbreakable by anyone on other planets

PGP—Pretty Good Privacy (3)

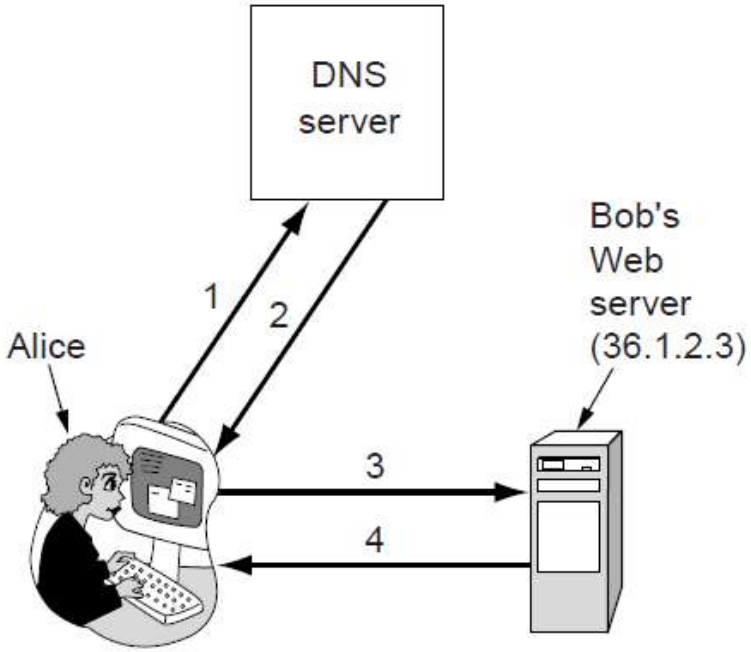


A PGP message

Web Security

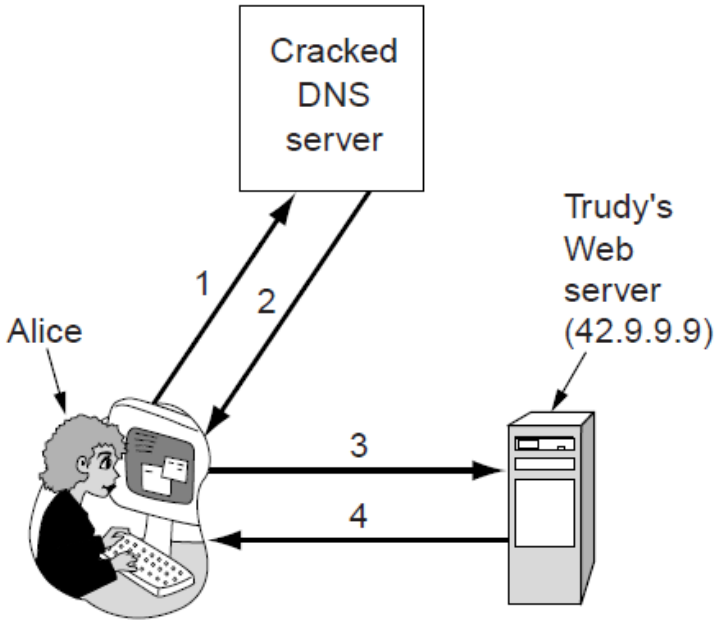
- Threats
- Secure naming
- SSL—the Secure Sockets Layer
- Mobile code security

Secure Naming (1)



Normal situation

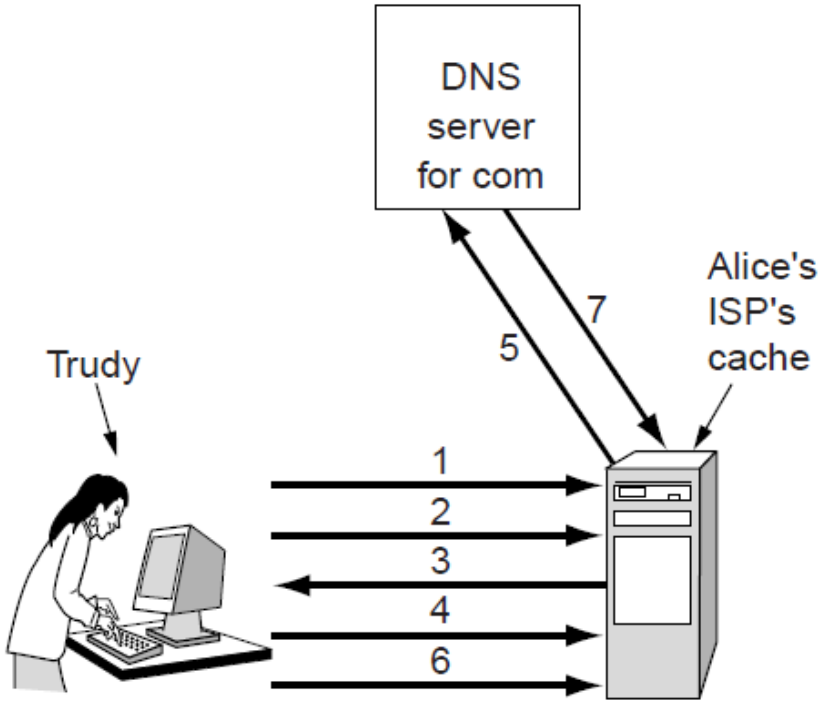
Secure Naming (2)



1. Give me Bob's IP address
2. 42.9.9.9 (Trudy's IP address)
3. GET index.html
4. Trudy's fake of Bob's home page

An attack based on breaking into DNS and modifying Bob's record.

Secure Naming (3)



1. Look up foobar.trudy-the-intruder.com
   (to force it into the ISP's cache)
2. Look up www.trudy-the-intruder.com
   (to get the ISP's next sequence number)
3. Request for www.trudy-the-intruder.com
   (Carrying the ISP's next sequence number, n)
4. Quick like a bunny, look up bob.com
   (to force the ISP to query the com server in step 5)
5. Legitimate query for bob.com with seq = n+1
6. Trudy's forged answer: Bob is 42.9.9.9, seq = n+1
7. Real answer (rejected, too late)

How Trudy spoofs Alice's ISP.

Secure Naming (4)

DNSsec fundamental services:

- Proof of where the data originated.
- Public key distribution.
- Transaction and request authentication.

Secure Naming (5)

Domain name	Time to live	Class	Type	Value
bob.com.	86400	IN	A	36.1.2.3
bob.com.	86400	IN	KEY	3682793A7B73F731029CE2737D...
bob.com.	86400	IN	SIG	86947503A8B848F5272E53930C...

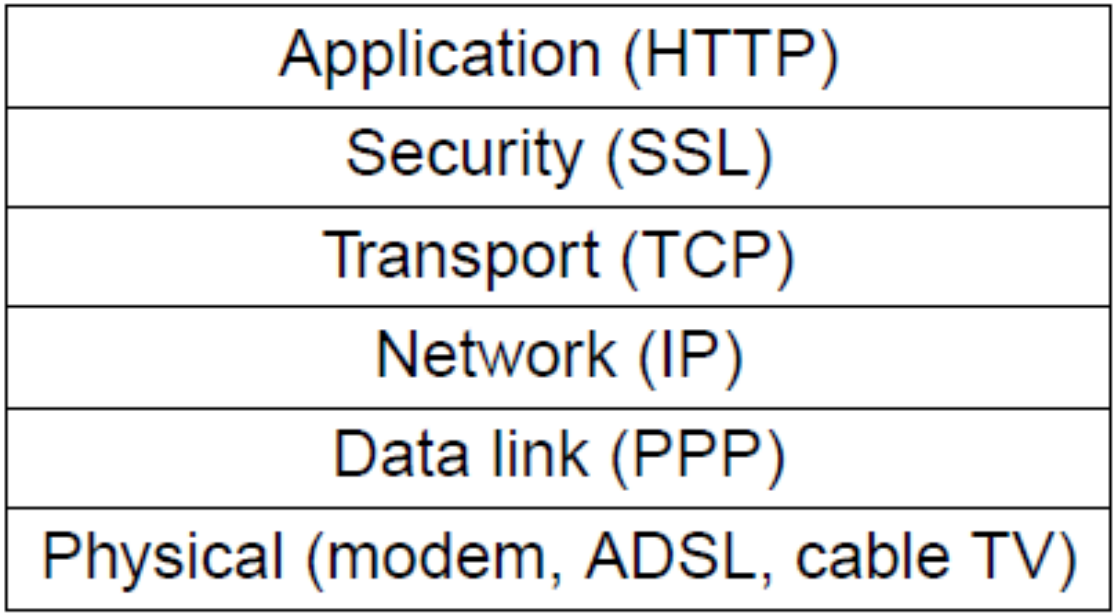
An example RRSet for *bob.com.* The KEY record is Bob's public key. The *SIG* record is the top-level *com* server's signed hash of the *A* and *KEY* records to verify their authenticity.

SSL—The Secure Sockets Layer (1)

Secure connection includes …

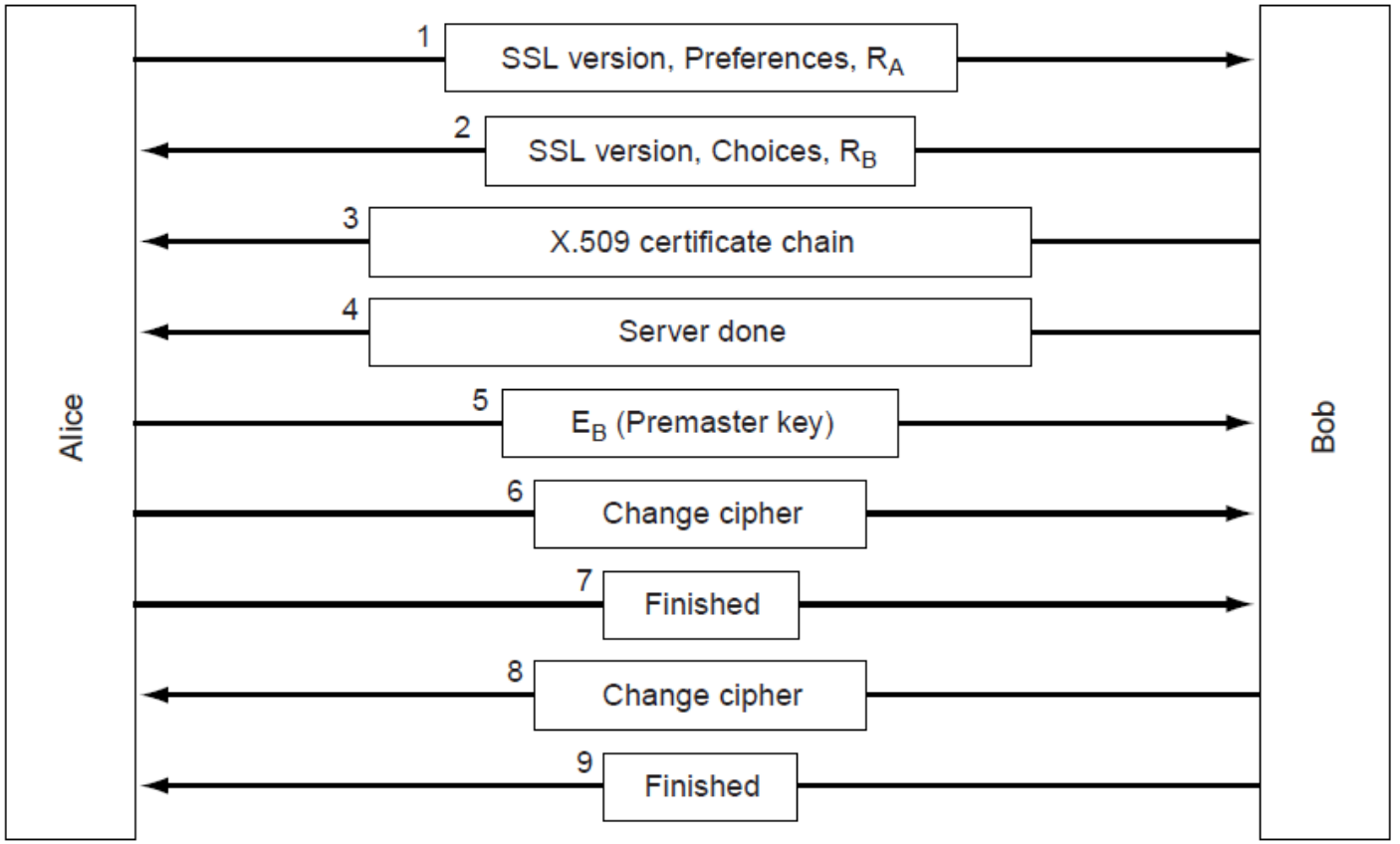
- Parameter negotiation between client and server.
- Authentication of the server by client.
- Secret communication.
- Data integrity protection.

SSL—The Secure Sockets Layer (2)



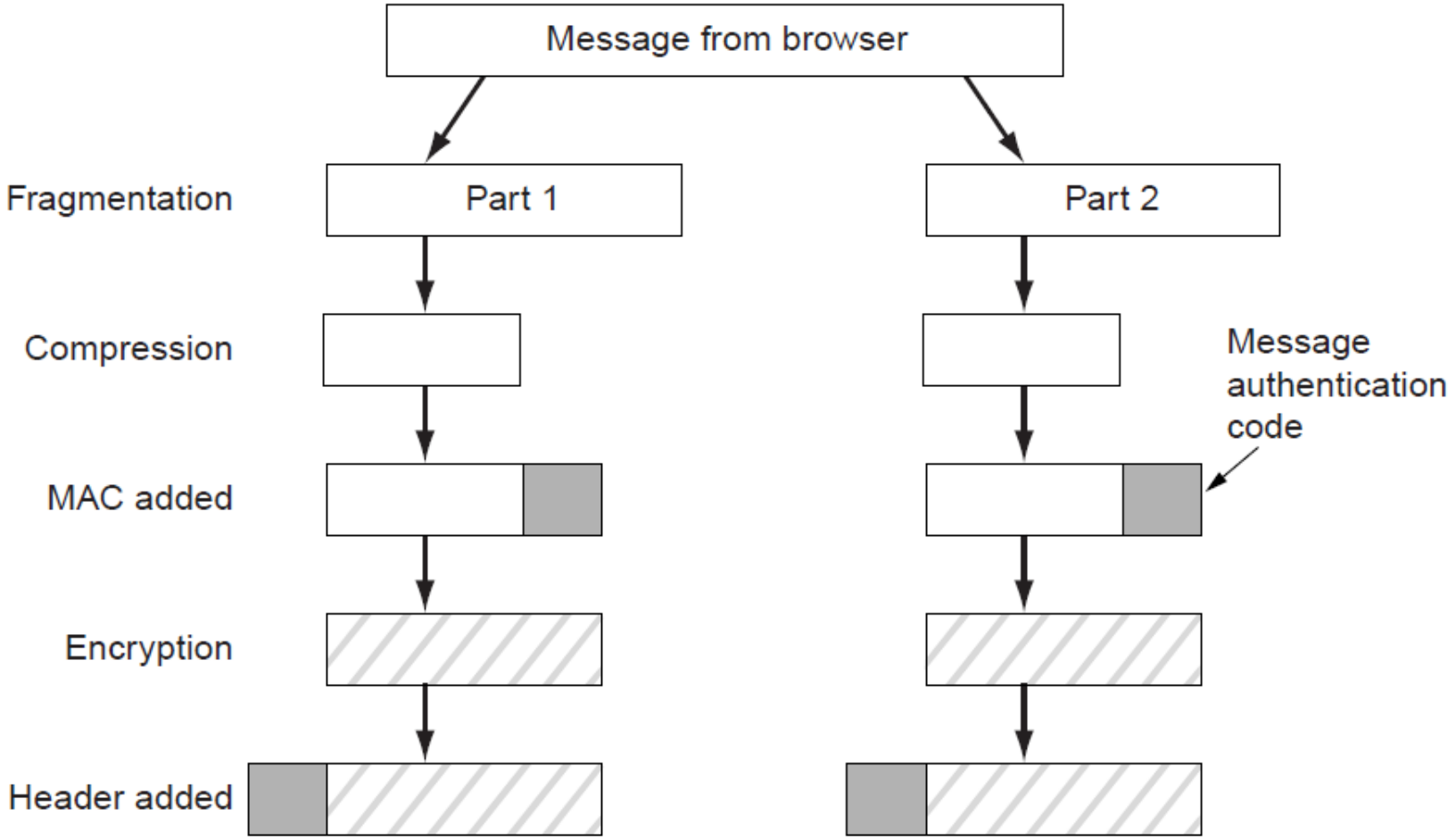
Layers (and protocols) for a home user browsing with SSL.

SSL—The Secure Sockets Layer (3)



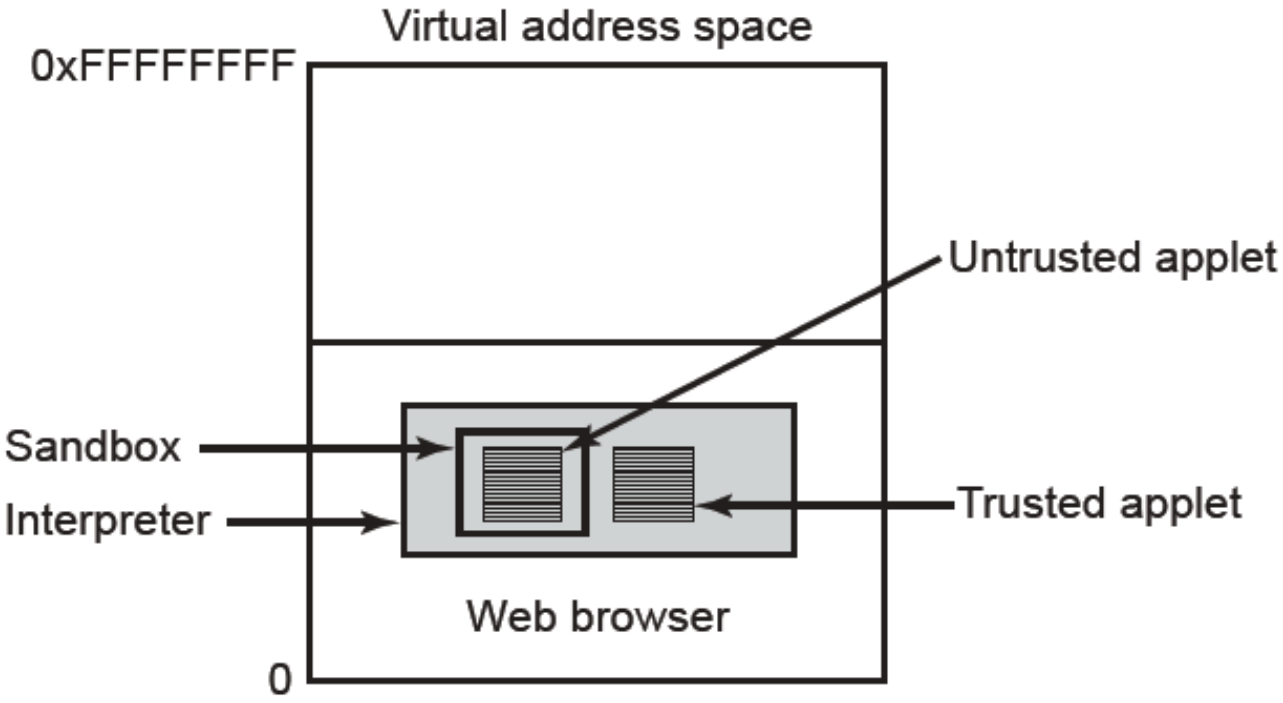
A simplified version of the SSL connection establishment subprotocol.

SSL—The Secure Sockets Layer (4)



Data transmission using SSL

Mobile Code Security

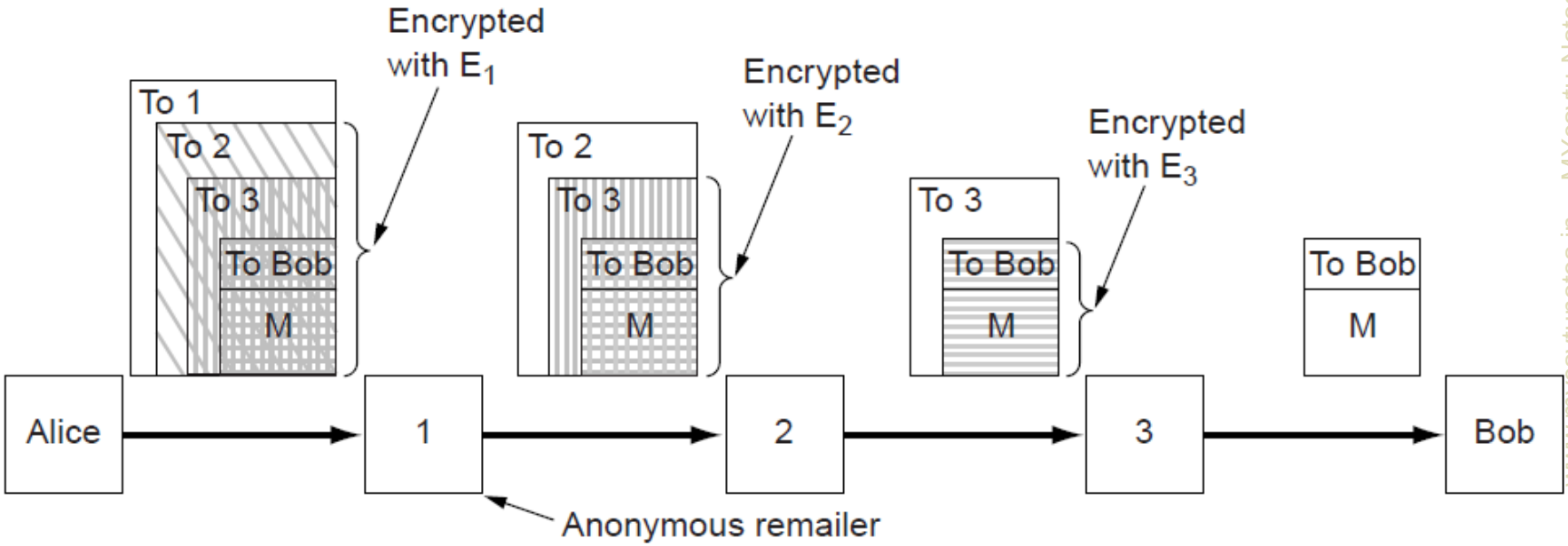


Applets can be interpreted by a Web browser

Social Issues

- Privacy
- Freedom of speech
- Copyright

Privacy



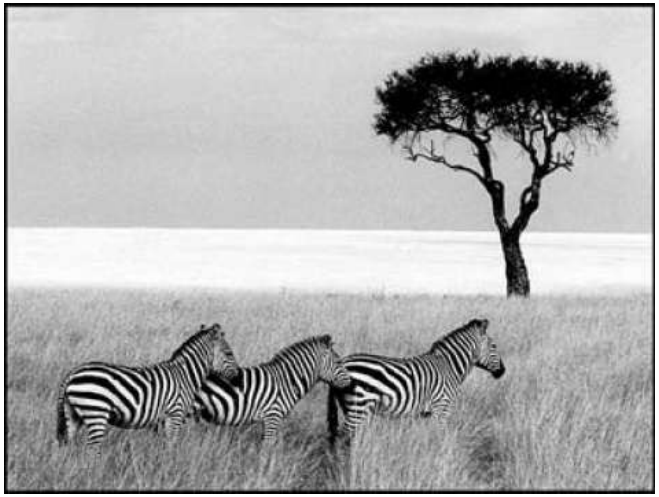
How Alice uses 3 remailers to send Bob a message

Freedom of Speech (1)

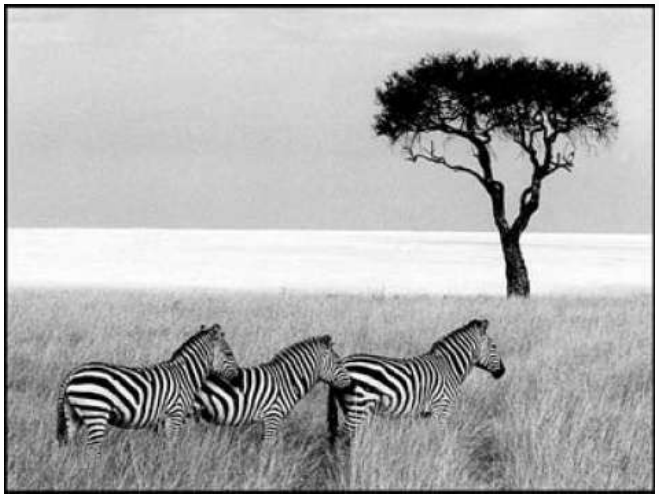
Possible banned material:

- Inappropriate for children
- Hate aimed at various groups
- Information about democracy
- History that contradicts government position
- Manuals for potentially illegal activities

Freedom of Speech (2)



(a)



(b)

(a) Three zebras and a tree.

(b) Three zebras, a tree, and the complete text of five plays by William Shakespeare.

End

Chapter 8