CHAPTER 8: NETWORK SECURITY

- **Introduction**
- **Cryptography**
- **Symmetric-key algorithms**
- **Public-key algorithms**
- **Digital signatures**
- **Management of public keys**
- **Authentication protocols**
- **Email security**
- Communication security
- * Web Security
- * Social Issues

INTRODUCTION

Some people who cause security problems and why.

Adversary	Goal
Student	To have fun snooping on people's e-mail
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 e-mail
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

Introduction

- Network security problems can be divided roughly into four interwined areas: secrecy, authentication, nonrepudiation, and integrity control.
  - Secrecy（机密性）: to keep information out of the hands of unauthorized users.
  - Authentication（认证）: to determine whom you are talking to before revealing sensitive information or entering into a business deal. (to authenticate people by recognizing their faces, voices, and handwriting).
  - Nonrepudiation（不可否认性）: to deal with signature. (personal signature).
  - Integrity（完整性）: How can you be sure that a message you received was really the one sent and not something that a malicious adversary modified in transit or concocted.

Introduction

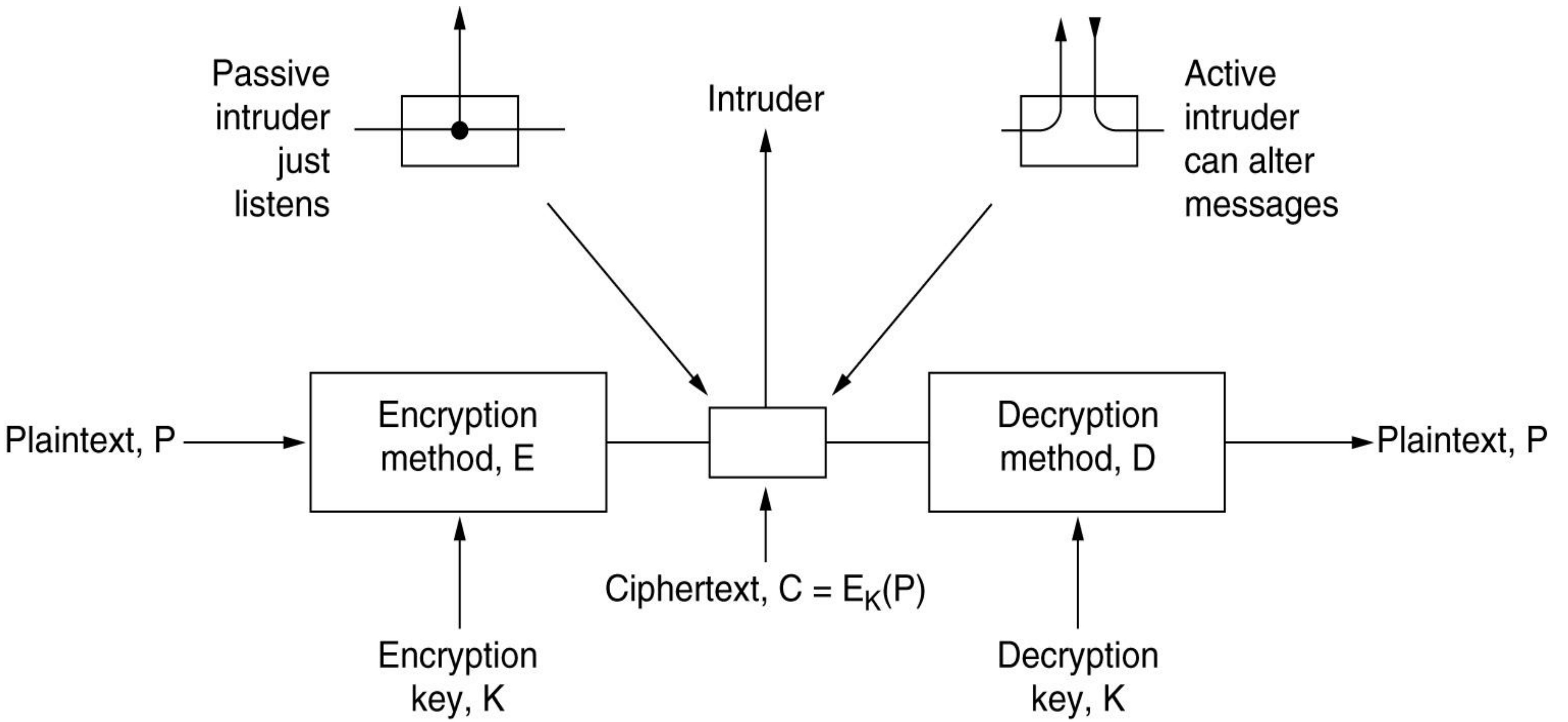
- Every layer has something to contribute:
  - <u>Physical layer</u>: Wiretapping can be foiled by enclosing transmission lines in sealed tubes containing argon gas at high pressure. (not always work)
  - <u>Data link layer</u>: Packets on a point-to-point line can be encoded as they leave one machine and decoded as they enter another. (not routed)
  - <u>Network layer</u>: Firewalls can be installed to keep packets in or keep packets out.
  - <u>Transport layer</u>: Entire connections can be encrypted, end to end, that is, process to process.
  - <u>Application layer</u>: Issues such as authentication and nonrepudiation can only be solved at the application layer.

CRYPTOGRAPHY

- Introduction to Cryptography
- Substitution Ciphers
- Transposition Ciphers
- One-Time Pads
- Two Fundamental Cryptographic Principles

Cryptography: Introduction

The encryption model



Cryptography: Introduction

- Encryption and decryption:
  - Encryption: $E_k(P) = C$.
  - Decryption: $D_k(C) = D_k(\ E_k(P)\ ) = P$.
- The basic model is a stable and publicly known general method parametrized by a secret and easily changed key.
  → **Kerchoff's principle:** all algorithms must be public while the key is secret.
- From the cryptanalyst's point of view, the cryptanalysis problem has three principal variations:
  - ciphertext only, 唯密文
  - known plaintext, and 已知明文
  - chosen plaintext. 选择明文
- To achieve security, the cryptographer should be conservative and make sure that the system is unbreakable even if his opponent can encrypt arbitrary amounts of chosen plaintext.

Cryptography: Substitution Ciphers

- In a substitution cipher, each letter or group of letters is replaced by another letter or group of letters to disguise it.
- The Caesar cipher (Julius Caesar):

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
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

attack (plaintext) ➔ DWWDFN (ciphertext)

- A slight generalization of the Caesar cipher allows the ciphertext alphabet to be shifted by *k* letters, instead of always 3.
  → *k* becomes a key to the general method of circularly shifted alphabets.

Cryptography: Substitution Ciphers

- Monoalphabetic substitution: The next improvement is to have each of the symbols in the plaintext map onto some other letter, with the key being the 26-letter string corresponding to the full alphabet.

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
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

attack (plaintext) ➔ QZZQEA (ciphertext)

- The key combination: $26! = 4 * 10^{26}$. Even at 1 nsec per solution, a computer would take 10^{13} years to try all the keys.

Cryptography: Substitution Ciphers

- *However*, given a surprisingly small amount of ciphertext, the cipher can be broken easily.
  - Solution 1: use statistical properties of natural languages:
    - The most common letters: e, t, o, a, n, i, etc.
    - The most common two letter combinations (digrams): th, in , er, re, an, etc.
    - The most common three letter combinations (trigrams): the, ing, and, ion, etc.
  - Solution 2: guess a probable word or phrase, e.g., the word *financial* from an accounting firm.

Cryptography: Substitution Ciphers

- An example: *financial*
  - Repeated i, 5 other letter in between: 10 hits
  - Repeated n in proper place: 2 hits (* and =)
  - Repeated a in proper place: only 1 hits (*)

```
1           6           11          16          21          26          31          36
C T B M N   B Y C T C   B T J D S   Q X B N S   G S T J C   B T S W X   C T Q T Z   C Q V U J
            -                   -                             -         *

41          46          51          56          61          66          71          76
Q J S G S   T J Q Z Z   M N Q J S   V L N S X   V S Z J U   J D S T S   J Q U U S   J U B X J
  =             -                   -                       -       -   -

81          86          91          96          101         106         111
D S K S U   J S N T K   B G A Q J   Z B G Y Q   T L C T Z   B N Y B N   Q J S W
```

Cryptography: Transposition Ciphers

- Transposition ciphers reorder the letters but do not disguise them. In contrast, substitution ciphers preserve the order of the letter but disguise them.

- Example: A transposition cipher. (See the next slide)

Cryptography: Transposition Ciphers

The cipher is keyed by a word or phrase not containing any repeated letters.

The purpose of the key is to number the columns, column 1 being under the key letter closest to the start of the alphabet

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

- The plaintext is written *horizontally*, in rows.
- The ciphertext is read out by *columns*, starting with the column whose key letter is the lowest.

Cryptography: Transposition Ciphers

- To break a transposition cipher,
  - The cryptanalyst must first be aware that he is dealing with a transposition cipher. By looking at the frequency of E, T, A, O, I, N, etc, it is easy if they fit the normal pattern for plaintext.
  - To make a guess at the number of columns. In many cases, a probable word or phase may be guessed at from the context of the message.
  - To order the columns

Cryptography: One-Time Pads

- Constructing an unbreakable cipher is easy; the technique has been known for decades.
  - Choose a *random bit string* as the key.
  - Convert the plaintext into a bit string.
  - Compute the XOR of these two strings, bit by bit.
- The resulting ciphertext cannot be broken, because every possible plaintext is an equally probable candidate. The ciphertext gives the cryptanalyst no information at all.

Cryptography: One-Time Pads

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.

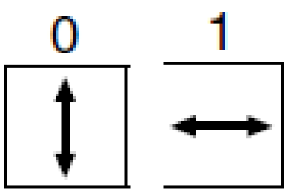
Message 1:	1001001	0100000	1101100	1101111	1110110	1100101	0100000	1111001	1101111	1110101	0101110
Pad 1:	1010010	1001011	1110010	1010101	1010010	1100011	0001011	0101010	1010111	1100110	0101011
Ciphertext:	0011011	1101011	0011110	0111010	0100100	0000110	0101011	1010011	0111000	0010011	0000101
Pad 2: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

Cryptography: One-Time Pads

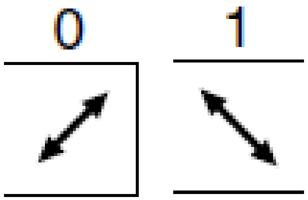
- The one-time pad has a number of practical disadvantages:
  - The key cannot be memorized, so both sender and receiver must carry a written copy with them.
  - The total amount of data that can be transmitted is limited by the amount of key available.
  - The method is sensitive to lost or inserted characters.
- One-time pad can be transmitted over the network via ***quantum cryptography***.

Quantum Cryptography

- Alice, Bob, Trudy
- If a beam of light is passed through a polarizing filter (偏振滤光镜), all photons (光子) will be polasized in the direction of filter's axis.
- Rectilinear basis (直线基): a set of polarizing filters: horizantal and vertical

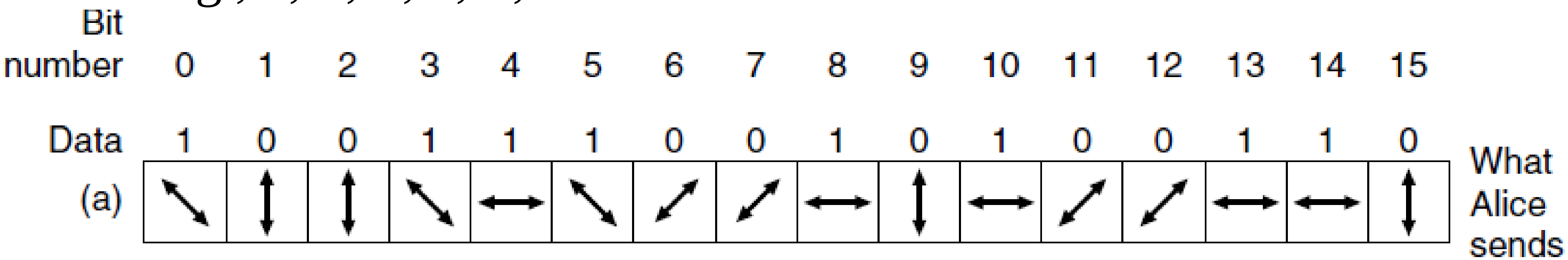


- Diagonal basis (对角基): rectilinear basis rotated by 45 degree.

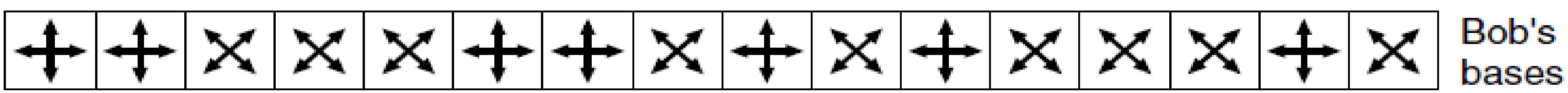


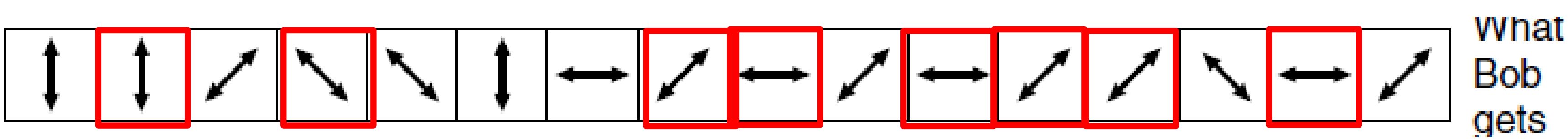
Process of establising one-time pad

- Alice sends one-time pad. Randomly choose the two bases. E.g., x, +, +, x, +, ...



- Bob randomly chooses the two bases
  - If base correct, the bit is correct (**red rectangle**)
  - Otherwise, the bit is random





Process of establising one-time pad (2)

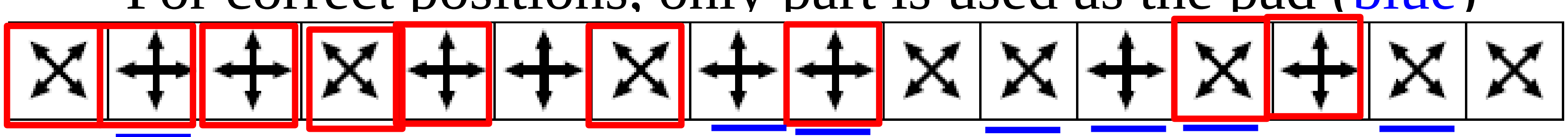
- Bob does not know which bit is correct. So Bob sends his choice of the base. Alice tells which choice is the same to hers.

No	Yes	No	Yes	No	No	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	Correct basis?

- Both use bits at the matching basis.

	0		1				0	1		1	0	0		1		One-time pad

- *Trudy cannot build the one-time pad
  - Suppose he is overhearing. Choose the following base (at random)
  - Know Bob's choice of base & Which bit position is correct (red)
  - For correct positions, only part is used as the pad (blue)



Trudy's bases

  - obtain part of the pad (x: bits not used in pad, ?: bits in pad, but unknown or random)

x	0	x	1	x	x	x	?	1	x	?	?	0	x	?	x	Trudy's pad

Cryptography: Two principles

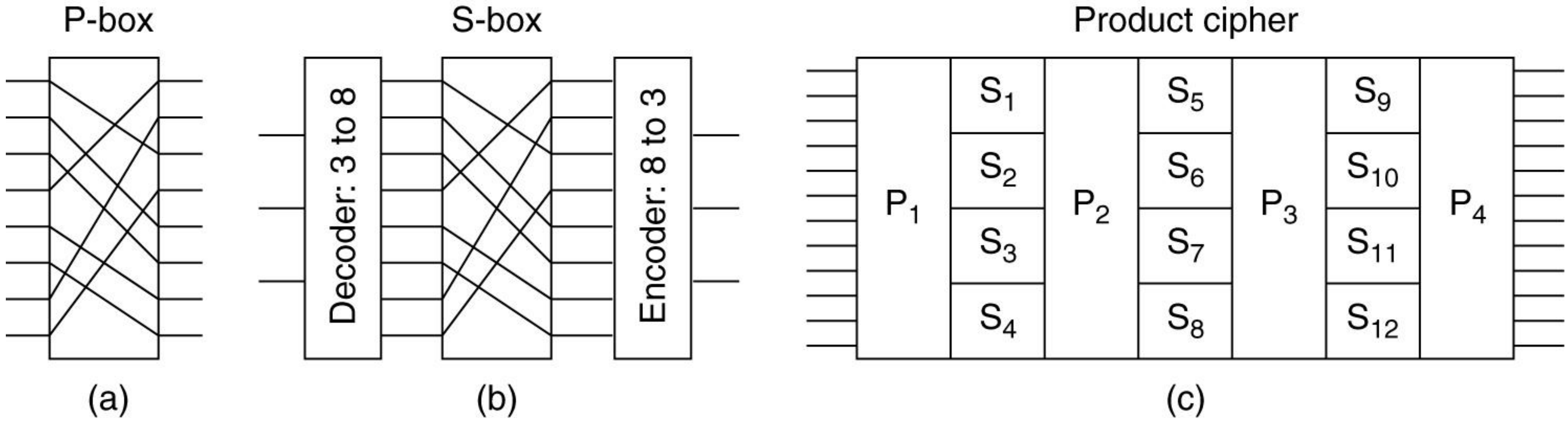
- Redundancy: All encrypted messages must contain some redundancy, that is, information not needed to understand the message. All messages must contain considerable redundancy so that active intruders cannot send random junk and have it be interpreted as a valid message.
- Freshness: Some measures must be taken to prevent active intruders from playing back old messages (replay attack).
- Kerckoff's principle?

SYMMETRIC-KEY ALGORITHMS

- DES – The Data Encryption Standard
- AES – The Advanced Encryption Standard
- Cipher Modes
- Other Ciphers
- Cryptanalysis

Symmetric-Key Algorithms: Introduction

- P(permutation)-box: 01234567 -> 36071245. similar to transposition cipher
- S(substitution)-box: 3-bit plaintext -> 3-bit ciphertext



Basic elements of product ciphers.

(a) P-box. (b) S-box. (c) Product.

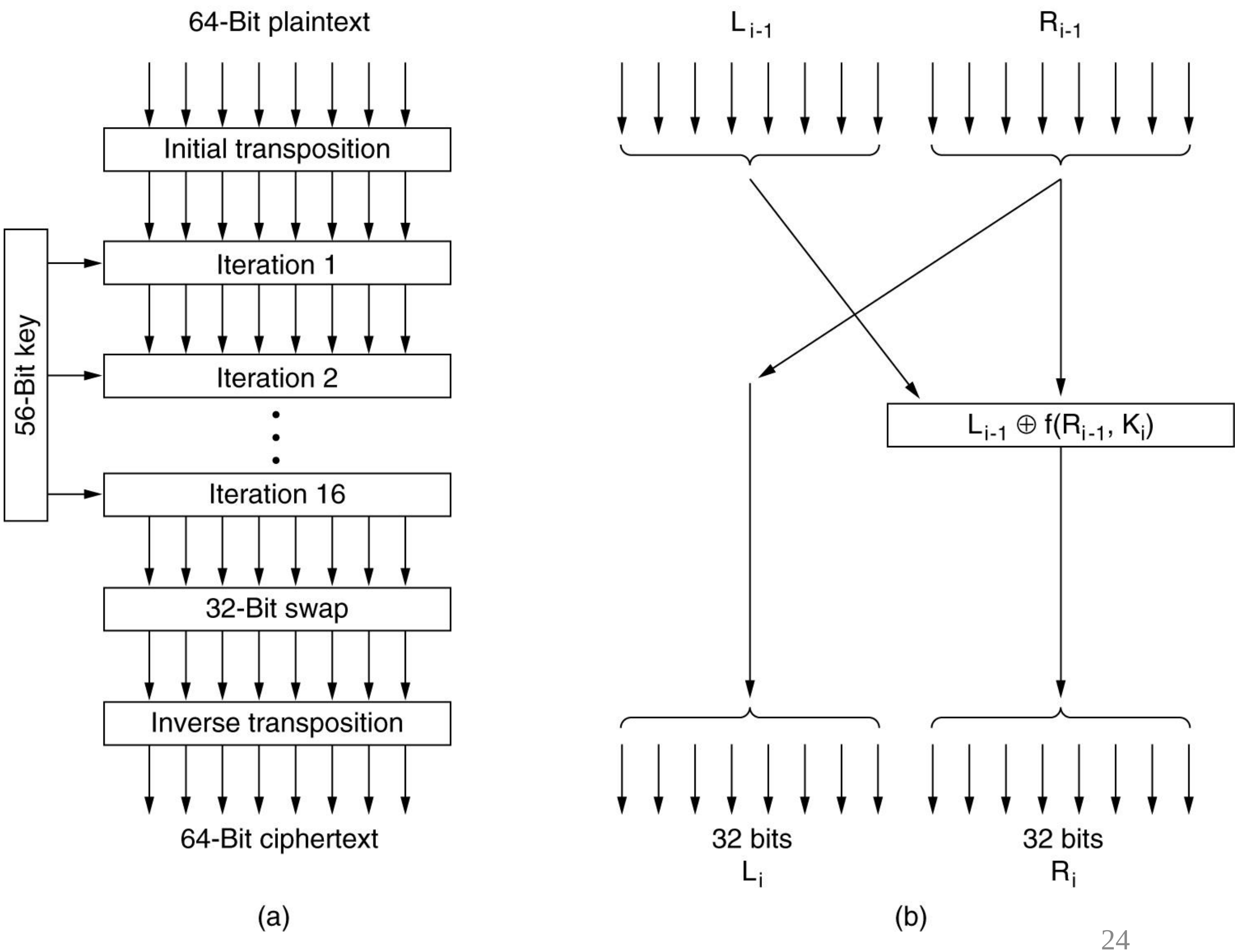
Symmetric-Key Algorithms: DES

The Data Encryption Standard (1977)

(a) outline.

(b) one iteration.

The circled + means XOR.



Symmetric-Key Algorithms: DES

- Function $f(R_{i-1}, K_i)$
  - To expand the 32-bit R_{i-1} according to a fixed transposition and duplication rule → 48-bit number E
  - To XOR E and K_i
  - To partition the XORed result into 8 groups of 6 bits each, each of which is fed into a different S-box. Each of the 64 possible inputs to an S-box is mapped onto a 4-bit output.
  - Finally, these 8x4 bits are passed through a P-box

Symmetric-Key Algorithms: DES

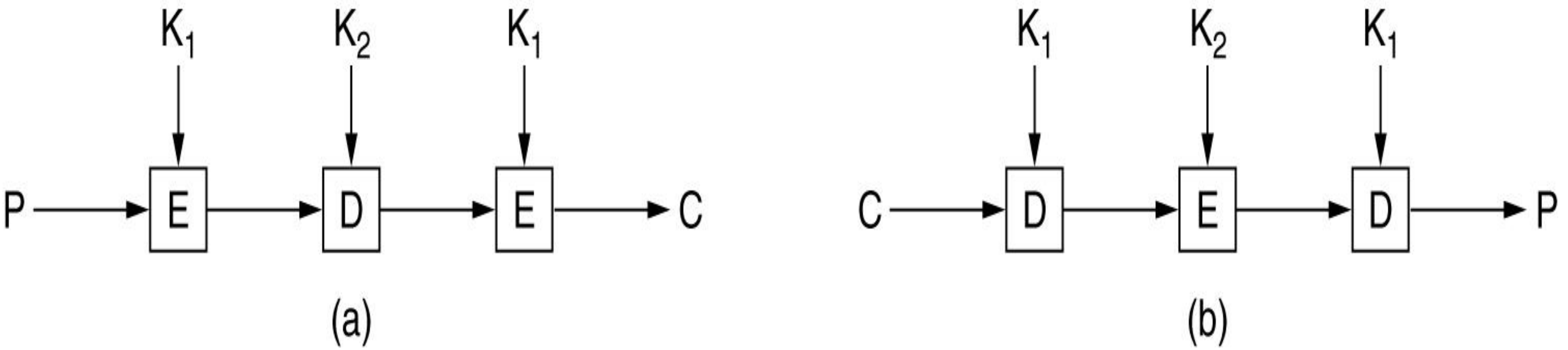
- In each of the 16 iterations, a different key K_i is used. K_i generation:
  - A 56-bit transposition is applied to the key.
  - The key is partitioned into two 28-bit units, each of which is rotated left by a number of bits dependent on the iteration number i.
  - By applying yet another 56-bit transposition to the rotated key, $\rightarrow K_i$
  - A different 48-bit subset of the 56 bits is extracted and permuted on each round.

Symmetric-Key Algorithms: DES

- In 1972, NIST's design requirements (National Institute of Standards and Technology)
- In 1974, IBM submitted the Lucifer algorithm (later called as DES)
- 1976 –1997, DES was used by the NIST
- Problems with DES
  - 128 bit key → 56 bit key (NSA: National Security Agency), Possible backdoor, Nondisclosure of design
- Breaking DES
  - In 1998, 3 days
  - In 1999, 22 hours and 15 minutes (PC networks)
  - 3.5 hours with the dedicated cracker.

Symmetric-Key Algorithms: DES

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



Symmetric-Key Algorithms: DES

- Why only 2 keys?
  - Even the most paranoid cryptographers believe that 112 bits is adequate for routine commercial applications for the time being.
- Why EDE?
  - Backward compatibility with existing single-key DES systems. (K_1=K_2)

Symmetric-Key Algorithms: AES

Rules for AES (Advanced Encryption Standard) proposals (In 1997):

1. The algorithm must be a symmetric block cipher.
2. The full design must be public.
3. Key lengths of 128, 192, and 256 bits supported.
4. Both software and hardware implementations required.
5. The algorithm must be public or licensed on nondiscriminatory terms.

Symmetric-Key Algorithms: AES

- In 1997, 15 serious proposals.
- In August 1998, 5 finalists.
  - Rijndael (from Joan Daemen and Vincent Rijmen, 86 votes).
  - Serpent (from Ross Anderson et al, 59 votes).
  - Twofish (Bruce Schneier, 31 votes).
  - RC6 (RSA Laboratories, 23 votes).
  - MARS (from IBM, 13 votes).
- In October 2000, Rijndael.
- In November 2001, Rijndael became a U.S. government standard published as FIPS 197.

Symmetric-Key Algorithms: Other Cipher

Some common symmetric-key cryptographic algorithms.

Cipher	Author	Key length	Comments
DES	IBM	56 bits	Too weak to use now
RC4	Ronald Rivest	1–2048 bits	Caution: some keys are weak
RC5	Ronald Rivest	128–256 bits	Good, but patented
AES (Rijndael)	Daemen and Rijmen	128–256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	128–256 bits	Very strong
Triple DES	IBM	168 bits	Good, but getting old
Twofish	Bruce Schneier	128–256 bits	Very strong; widely used

Symmetric-Key Algorithms: AES

- Rijndael supports key lengths and block sizes from 128 bits to 256 bits in steps of 32 bits.
- Rijndael is based on Galois (伽罗瓦) field theory
- Rijndael uses substitution and permutations and it also uses multiple rounds.
  - `void rijndael(`

```
  byte plaintext[LENGTH],
  byte ciphertext[LENGTH],
  byte key[LENGTH])
```

Symmetric-Key Algorithms: AES

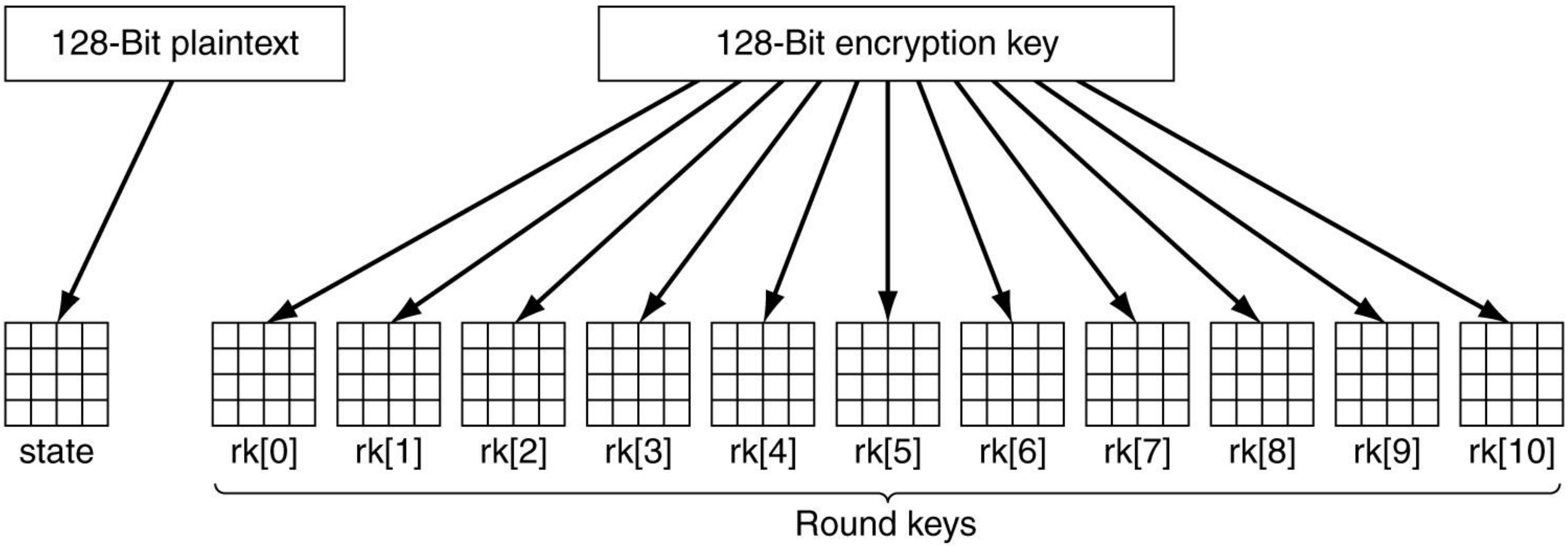
Rijndael outline

128-bit

```
#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 */

  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 */
}
```

Symmetric-Key Algorithms: AES

Creating of the *state* and *rk* arrays.

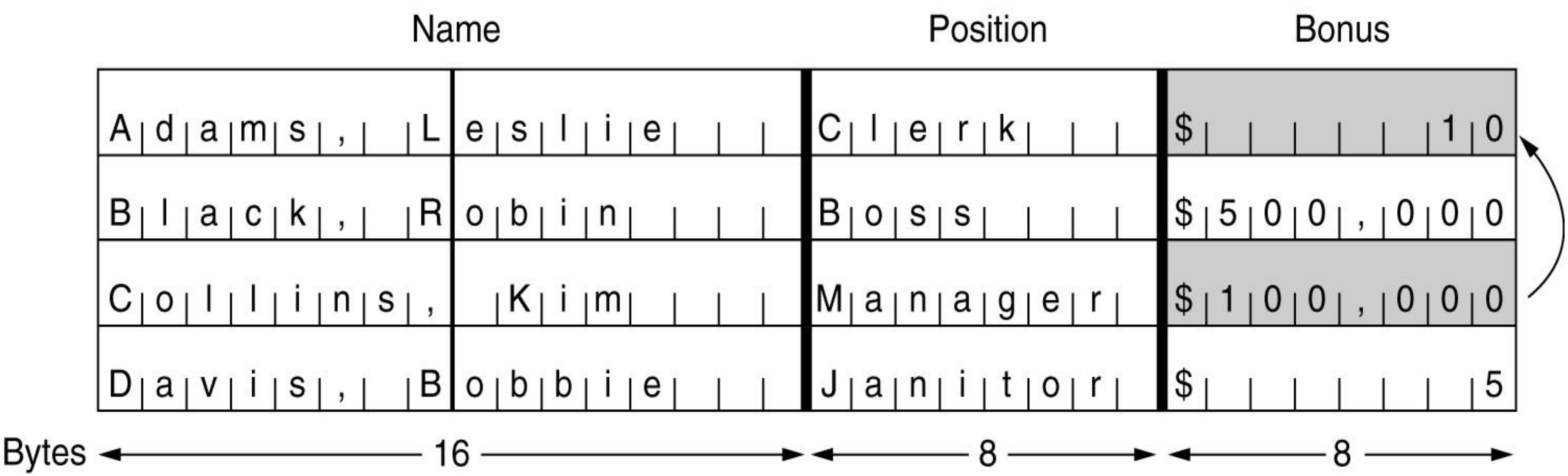


Symmetric-Key Algorithms: Cipher Modes

- Electronic code book mode
- Cipher block chaining mode
- Cipher feedback mode
- Stream cipher mode
- Counter mode

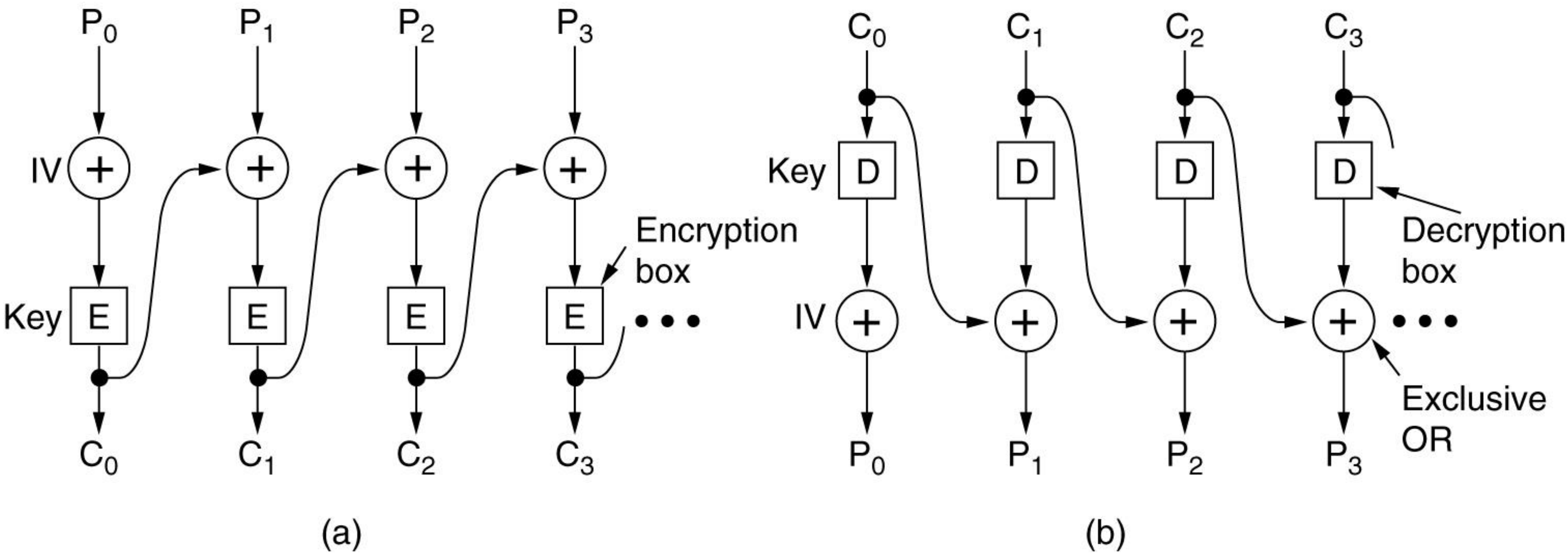
Symmetric-Key Algorithms: Cipher Modes

- **Electronic Code Book Mode (ECB)**
  - The plaintext of a file encrypted as 16 DES blocks.
  - 16 8-byte block encrypted using triple DES
  - Leslie can substitute 4th block with 12th block!



Symmetric-Key Algorithms: Cipher Modes

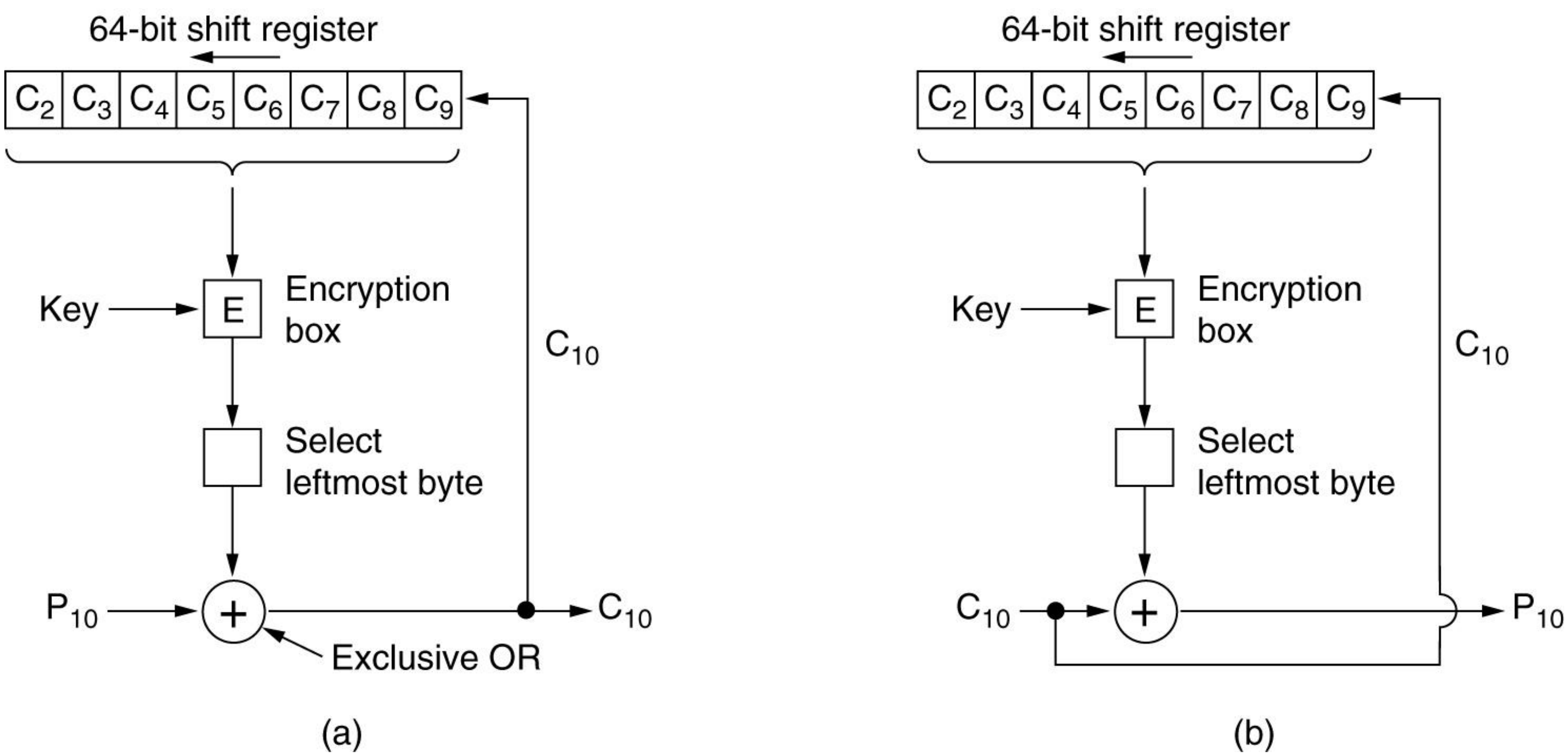
- Problem: each block is encrypted individually. Even under substitution, the cipher is correct
- Solution: make the cipher block dependent on all blocks before. Under substitution, the cipher block would be incorrect.
  - Problem: Decode only after a full reception of block.



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

Symmetric-Key Algorithms: Cipher Modes

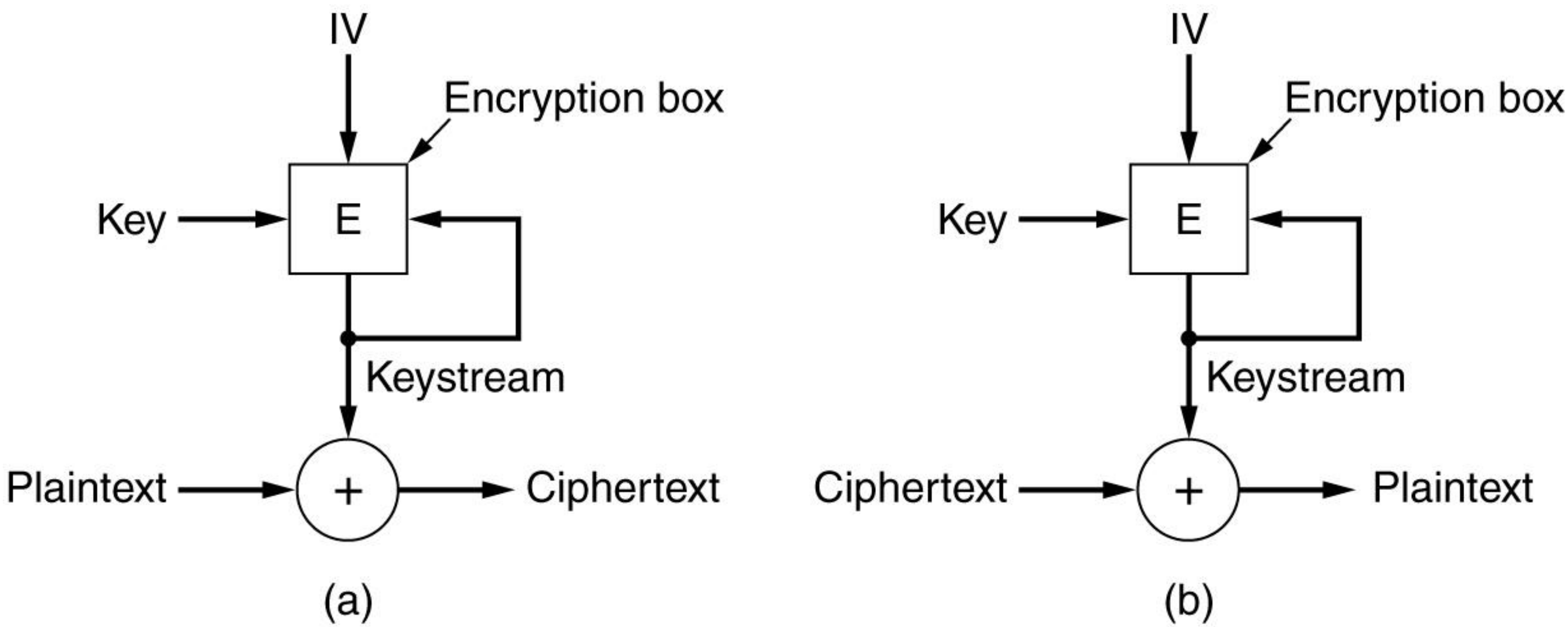
- Solution: Encode and decode on each individual byte.
  - Problem: the decoding of 1 byte depends on previous bytes (cipher) → 1 bit transmission error will impact 8-byte (64-bit) plaintext



Cipher Feedback Mode (a) Encryption. (c) Decryption.

Symmetric-Key Algorithms: Cipher Modes

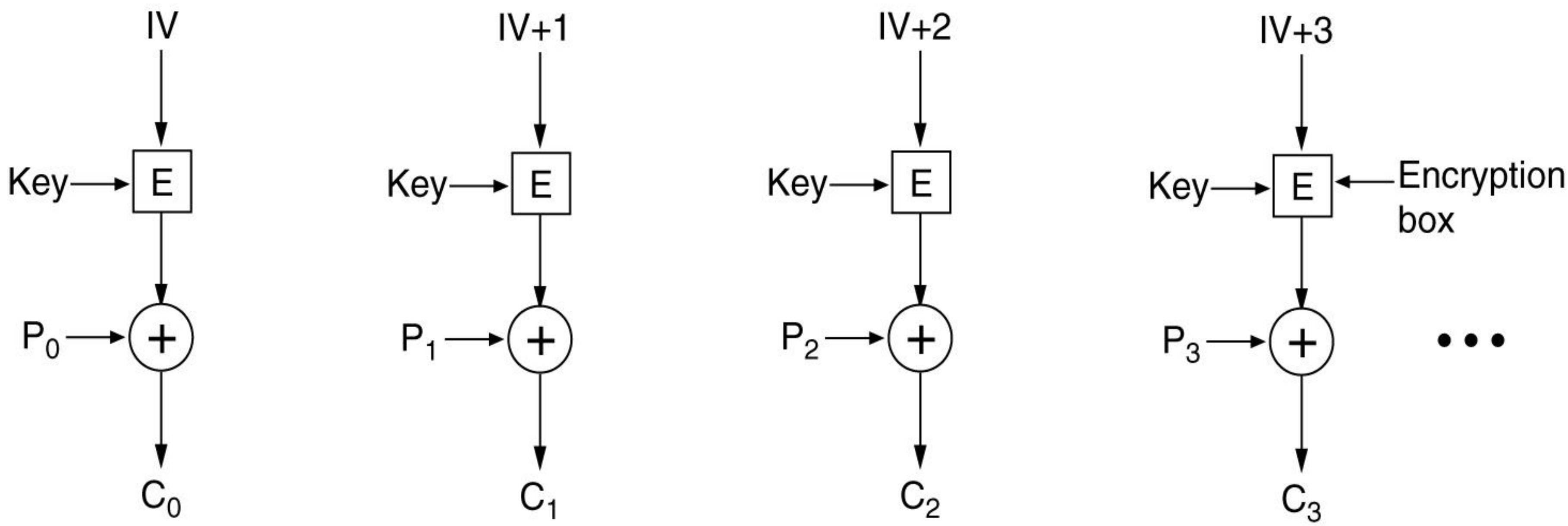
- Solution: XOR plaintext with keystream.
  - Problem: If IV is reused → keystream reuse attack, i.e. P_1 XOR K = C_1, P_2 XOR K = C_2, C_1 XOR C_2 = P_1 XOR P_2!!



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

Symmetric-Key Algorithms: Cipher Modes

- Problem of all other mode (except ECB): random access to cipher block will cause heavy computation overhead.
- Solution: encrypt IVs instead of plaintext.



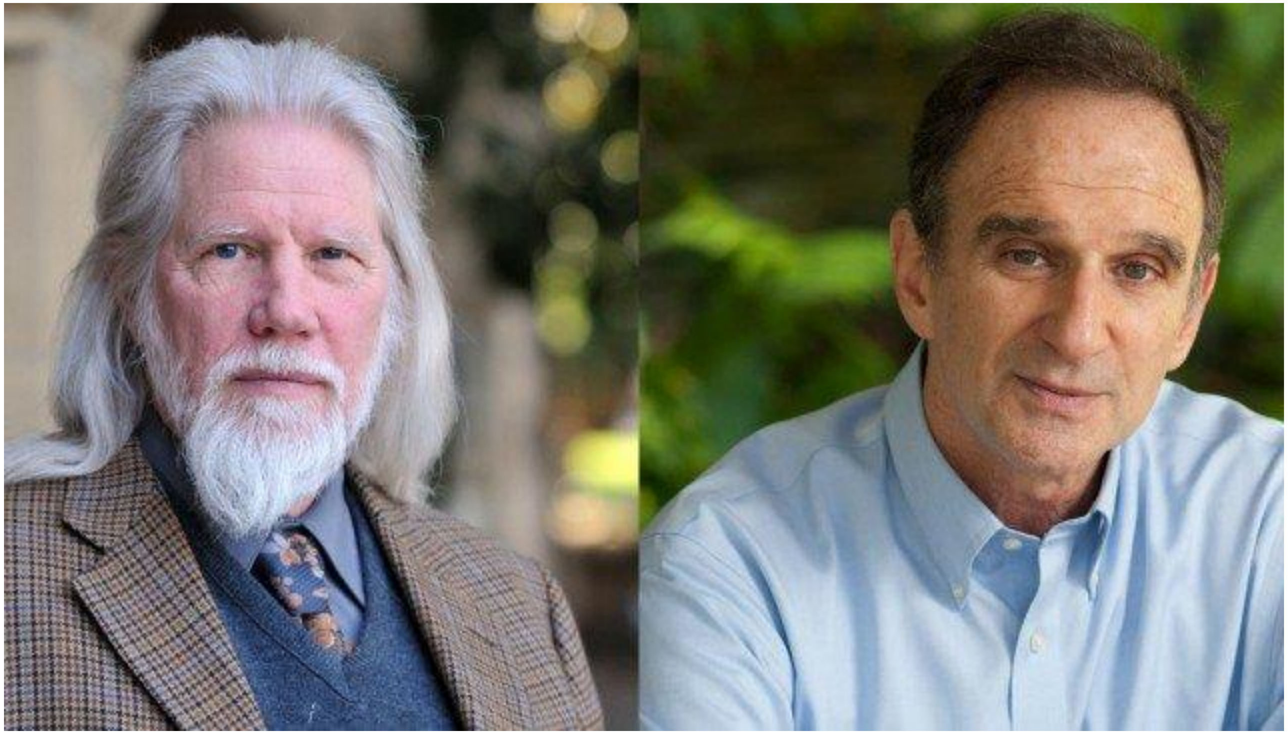
Encryption using **counter mode**.

PUBLIC-KEY ALGORITHMS

- RSA (Rivest, Shamir, Adleman)
- Other Public-Key Algorithms

Public-Key Algorithms: Introduction

- The key distribution has always been the weak link in most cryptosystems.
- In 1976, two researchers at Stanford University, Diffie and Hellman, proposed a radically new kind of cryptosystem, one in which the encryption and decryption keys were different, and the decryption key could not be derived form the encryption key.
- Three requirements for this cryptosystem:
  - D(E(P)) = P
  - It is exceedingly difficult to deduce D from E.
  - E cannot be broken by a chosen plaintext attack.



- The Association for Computing Machinery (**ACM**) named Whitfield Diffie and Martin E. Hellman recipients of the **2015 Turing Award**. They have been honored with this prestigious award for their work in **public-key cryptography** and **digital signatures**. The two computer scientists have given the public the ability to use encrypted software to communicate in a private manner and enabled a way to verify a person's digital identity. Satoshi Nakamoto's **Bitcoin protocol** also borrows from Diffie and Hellman's work and is a significant foundation to the network's operations.

Public-Key Algorithms: Introduction

- The method works like this:
  - A person, say, Alice, wanting to receive secret messages, first devises two algorithms, E_A and D_A, meeting the above requirements.
  - The encryption algorithm and key, E_A, is then made public, hence the name public-key cryptography (to contrast it with traditional secret-key cryptography).
  - Alice publishes the decryption algorithm (to get the free consulting), but keeps the decryption key secret. Thus, E_A is public, but D_A is private.

Public-Key Algorithms: Introduction

- How Alice and Bob establish a secure channel?
  - Both Alice's encryption key, E_A, and Bob's encryption key, E_B, are assumed to be in a publicly readable file.
  - Now Alice takes her first message, P, computes $E_B(P)$,and sends it to Bob.
  - Bob then decrypts it by applying his secret key D_B [i.e. $D_B(E_B(P)) = P$].
  - No one else can read the encrypted message $E_B(P)$, because the encryption system is assumed strong and because it is too difficult to derive D_B from the publicly known E_B.
- Public-key cryptography requires two keys:
  - a public key, used by the entire world for encrypting messages to be sent to that user, and
  - a private key, which the user needs for decrypting messages.

Public-Key Algorithms: RSA

- The RSA algorithm
  - Choose two large primes, p and q (typically 1024 bits)
  - Compute $n = p*q$ and $z = (p-1)*(q-1)$
  - Choose a number relatively prime to z and call it d.
  - Find *e* such that $e*d = 1 \bmod z$
  - To encrypt a message $C = P^e \pmod{n}$
  - To decrypt a message $P = C^d \pmod{n}$
- *Ex:* $P = 3,\ q=11,\ n = 33,\ z = 20$
  $D = 7,\ e = 3.$
- Factoring large numbers is very difficult.

Public-Key Algorithms: RSA

An example of the RSA algorithm.

$P = 3, q=11, n = 33, z = 20$

$D = 7, e = 3.$

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

Public-Key Algorithms: Other Algorithms

- Knapsack
  – Someone owns a large number of objects, each with different weight. The owner encodes the message by secretly selecting a subset of the objects and placing them in the knapsack.
  – The total weight of the objects in the knapsack is made public, as is the list of all possible objects.
  – The list of objects in the knapsack is kept secret.
  – With certain additional restrictions, the problem of figuring out a possible list of objects with the given weight was thought to computational infeasible and then formed the basis of the public-key algorithm.

Public-Key Algorithms: Other Algorithms

- Knapsack (Ralph Merkle)
  - $100 reward → Adi Shamir (the “S” in RSA)
  - $1000 reward for the new strengthened algorithm → Ronald Rivest (the “R” in RSA)
  - $10000 reward? (poor Leonard Adleman)

DIGITAL SIGNATURES

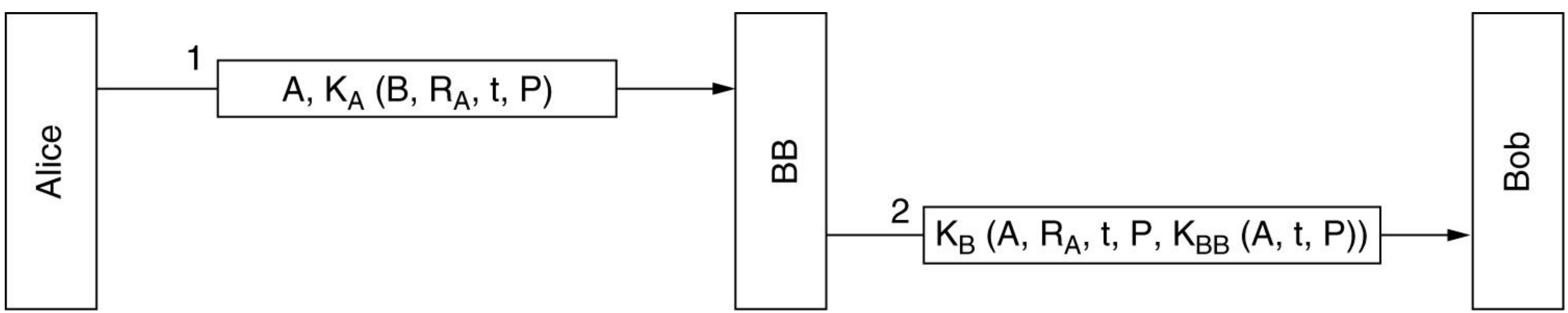
- Symmetric-Key Signatures
- Public-Key Signatures
- Message Digests
- The Birthday Attack

Digital Signatures

- Three conditions for digital signatures
  - The sender cannot later repudiate the contents of the message.
  - The receiver can verify the claimed identity of the sender.
  - The receiver cannot possibly have concocted the message himself.

Digital Signatures: Symmetric-key signatures

- BB (Big Brother): the central authority that knows everything and whom everyone trusts.
- Each user 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.
- Alice sends a signed plaintext message P to Bob.
  - R_A, a random number chosen by Alice.
  - t, timestamp to prevent replay attack.

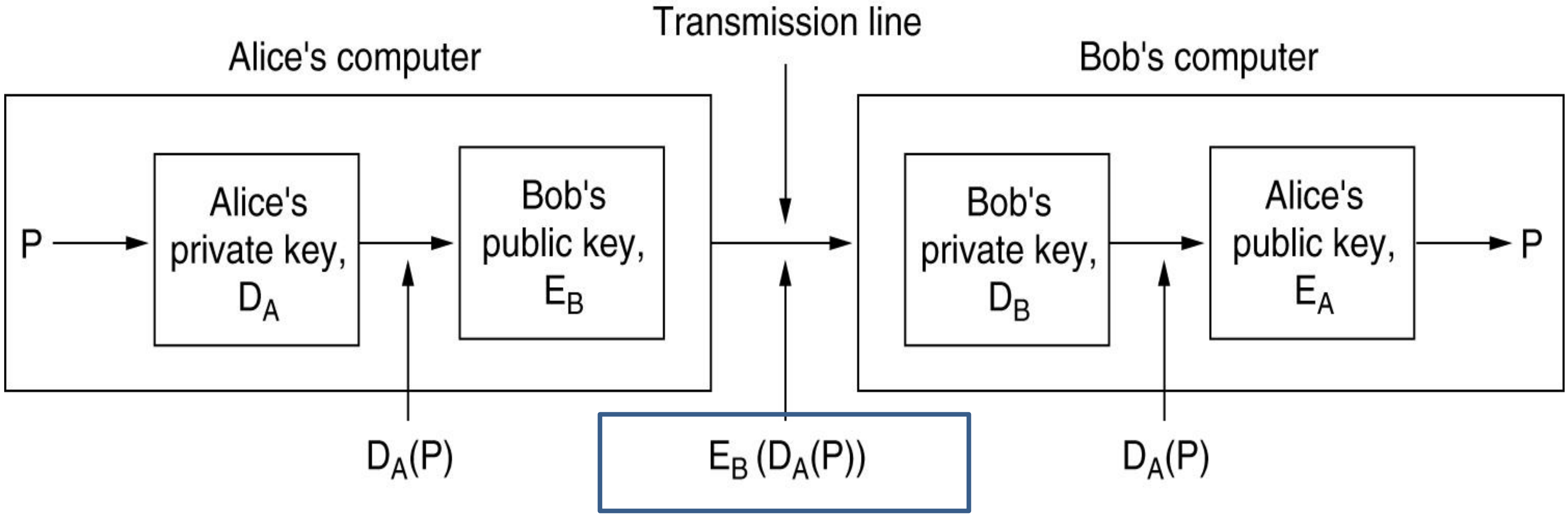


Digital Signatures: Symmetric-key signatures

- What happens if Alice later denies sending the message?
  - Trudy伪装为Alice可能吗？
  - No. Otherwise BB received **A, K_T(…)** → detect mismatch in identity!
  - Bob can show the evidence: $K_{BB}(A, t, P)$
- Trudy replaying either message.
  - Use *t* to reject very old messages, e.g., 1 hour ago
  - Use R_A to check all recent messages, e.g. discard message with the same R.

Digital Signatures: Public key signatures

Digital signatures using public-key cryptography



- Another possibility: send P, $D_A(P)$ (no secrecy)
- Suppose that Alice later denies having sent the message P to Bob
  - When the case comes up in court, Bob can produce both P and $D_A(P)$
  - The judge can verify that Bob has a valid message encrypted by D_A by simply applying E_A to it.

Digital Signatures: Public key signatures

- How about?
  - if Alice discloses her secret key?
  - if Alice modifies her secret key?

Digital Signatures: Message Digests

- Problem: The previous digital signature solutions requires encrypting the entire message, causing large computational overhead.
- MD (message digest) has 4 important properties
  - Given P, it is easy to compute $MD(P)$.
  - Given $MD(P)$, it is effectively impossible to find P.
  - Given P no one can find P' such that

$$MD(P') = MD(P)$$

  - A change to the input of even 1 bit produces a very different output.

Digital signature with public keys

- Without message digest
  - The signed message: $E_B(D_A(P))$ → two encryptions
  - Or $P, D_A(P)$ → one encryption
- With message digest
  - $P, D_A(MD(P))$
  - Trudy cannot modified P to P'. Otherwise the verification fails: $MD(P') \mathrel{!=} E_A (D_A(MD(P)))$

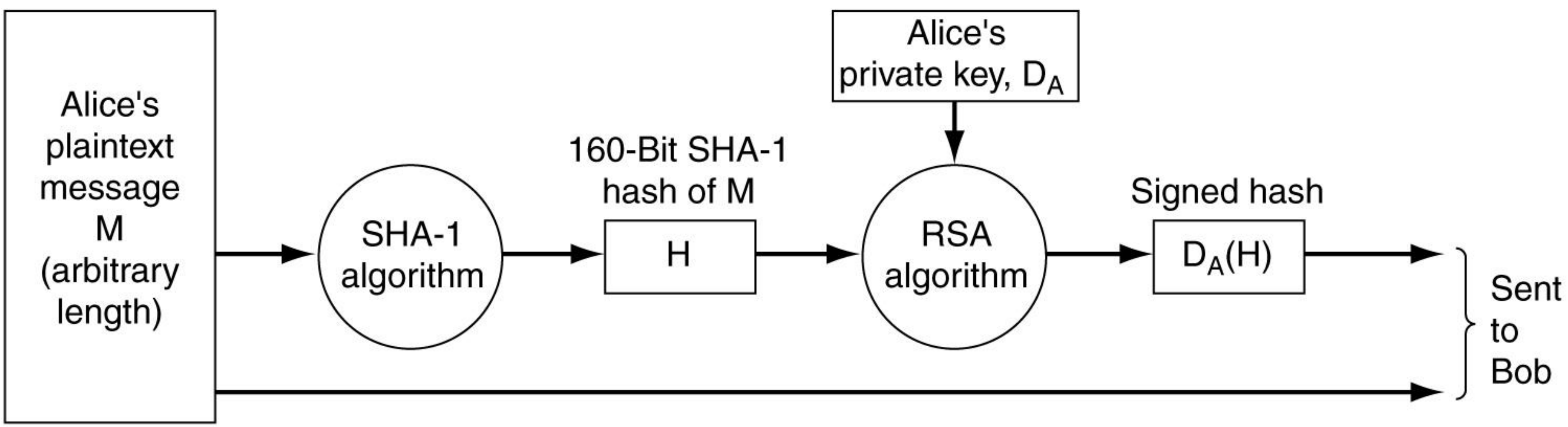
Digital signatures with private keys

- Without message digest
  - BB→ Bob: K_B (…$K_{BB}(A,t,P)$)

- With message digest
  - BB→ Bob: $K_{BB}(A,t,MD(P))$
  - Bob can obtain MD(P) with the help of BB. P cannot be changed to P'. Otherwise MD(P') != MD(P)

Digital Signatures: Secure Hash Algorithm (SHA-1)

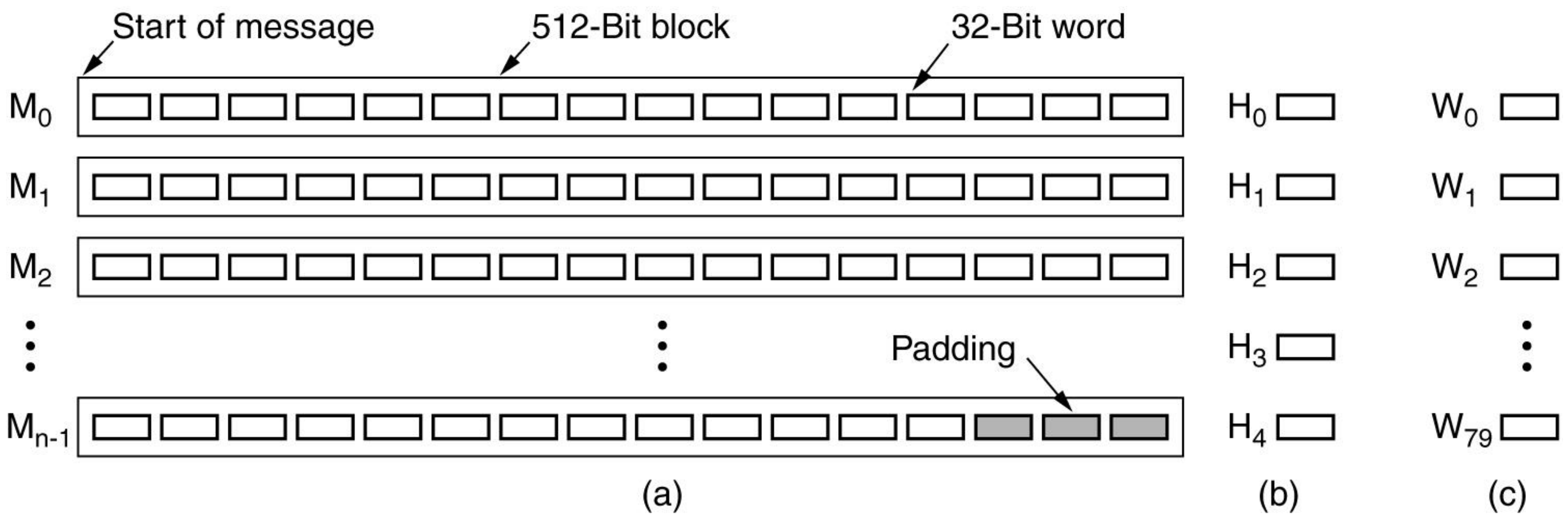
M, D_A(SHA(M))



Use of SHA-1 and RSA for signing nonsecret messages.

Digital Signatures: SHA-1

- SHA-1 (RFC3174)
  - To pad the message by adding a 1 bit to the end, followed by as many 0 bits as needed to make the length a multiple of 512 bits.
  - To OR the lower-order 64-bits with a 64-bit number containing the original message length



(a) A message padded out to a multiple of 512 bits. (b) The output variables (5个32位变量, **160位消息摘要**). (c) The word array.

Digital Signatures: SHA-1

- SHA-1 (Cont'd)
  - Each of the blocks Mi is now processed in turn.
    - Copy 16 words to the W[0..15]
    - Compute the rest W[16..79]
    - Use A through E to initialize H_0 to H_4
    - To do 80 times of a loop (*too much to list*)
  - The result is in H_0 though H_4

Digital Signatures: The Birthday Attack

- How to subvert a *m*-bit message digest?
  - Require 2^m operations to generate a collision, i.e., enumerate this number of messages and their MD
  - In reality, only $2^{m/2}$ operations will be needed using the birthday attack.
- 生日悖论：how many students do you need in a class before the probability of having two people with the same birthday exceeds 1/2? 23
- In general: for *n* inputs (people, messages) and *k* possible outputs (birthdays, MD), there are $n(n-1)/2$ input pairs. If $n(n-1)/2 > k$, the chance of having at least one match is pretty good. A match is likely for $n > \sqrt{k}$. → <u>A 64-bit MD can probably be broken by generating about 2^{32} messages and looking for two with the same message digest.</u>

Digital Signatures: The Birthday Attack

- One university has one tenure position. Tom and Dick are the two candidates. Tom was hired two years before Dick.
- Tom asks Marilyn to write him a letter of recommendation to the Dean. Marilyn tells her secretary, Ellen, to write the Dean a letter, outlining what she wants in it. When it is ready, Marilyn will review it, and compute and sign the 64-bit digest and send it to the Dean. Ellen can send the letter later by e-mail.

Digital Signatures: The Birthday Attack

- Ellen is romantically involved with Dick. So she writes the letter below with **32 bracket options**. (see P805 in textbook)
  - Letter A: … Prof. Wilson is an [*outstanding*|*excellent*] researcher of great [*talent*|*ability*] …
  - Letter B: …Prof. Wilson is an [*poor*|*weak*] researcher of not well known in his [*field*|*area*] …
  - Compute the 2^{32} MD for each letter and find a match

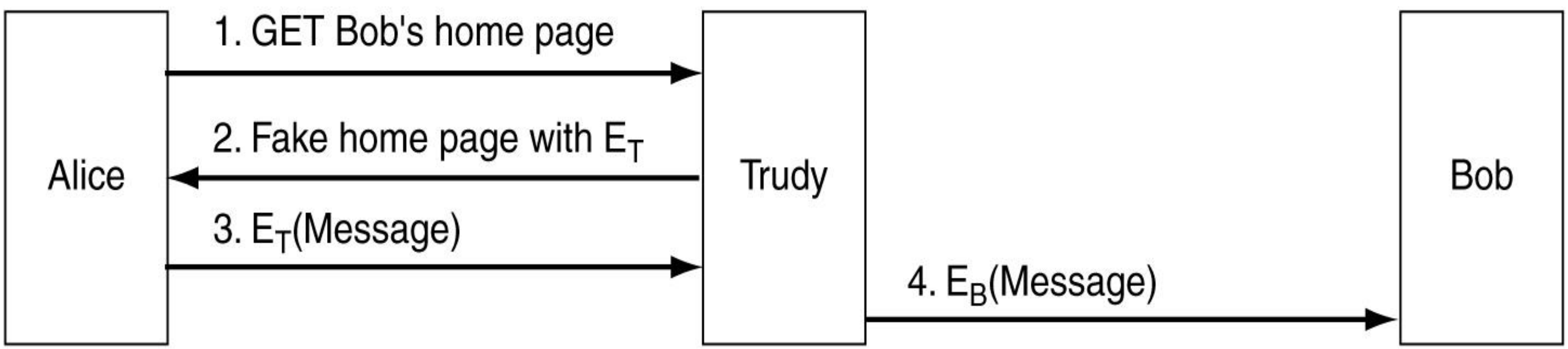
- Marilyn approve the letter A. However Ellen sends the letter B
- Ending:
  - Tom was unlucky.
  - Or other ?

MANAGEMENT OF PUBLIC KEYS

- Certificates
- X.509
- Public Key Infrastructures

Management of Public Keys

- How to obtain public key?
- One solution: Bob puts his public key in the website for others to retrieve.



A way for Trudy to subvert public-key encryption.

Management of Public Keys: Certificates

- **CA** (certification authority): an organization to certify public keys
- Suppose Bob wants to allow Alice and other people to communicate with him securely.
  - He can go to the CA with his public key along with his passport and ask to be certified.
  - The CA issues a certificate and signs its SHA-1 hash with the CA private key.
  - Bob gets a floppy disk containing the certificate and its signed hash (and pays the CA's fee)

Management of Public Keys: Certificates

A possible certificate and its signed hash.

$P_B \,||\, D_{CA}\,(SHA(P_B))$

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

It is assumed that everyone knows the CA's public key

Certificates

- What Trudy can do?
- 1. put **P_T || D_{CA} (SHA(P_T))** on fake page. Alice will find since **P_T** says it is Trudy!
- 2. modify **P_B** to **P_{B2}** by replacing Trudy's public key. But Trudy cannot generate the signed block: **P_{B2} || D_{CA} (SHA(P_B)).** Alice will find mismatch!
  - **SHA(P_{B2}) != E_{CA}(D_{CA} (SHA(P_B))).**

Management of Public Keys: X.509

- X.509 is the standard for managing the certificates
- X.509 describes the certificate fields.

  (See the next slide)
- ASN.1 (Abstract Syntax Notation 1) is used to encode the certificates

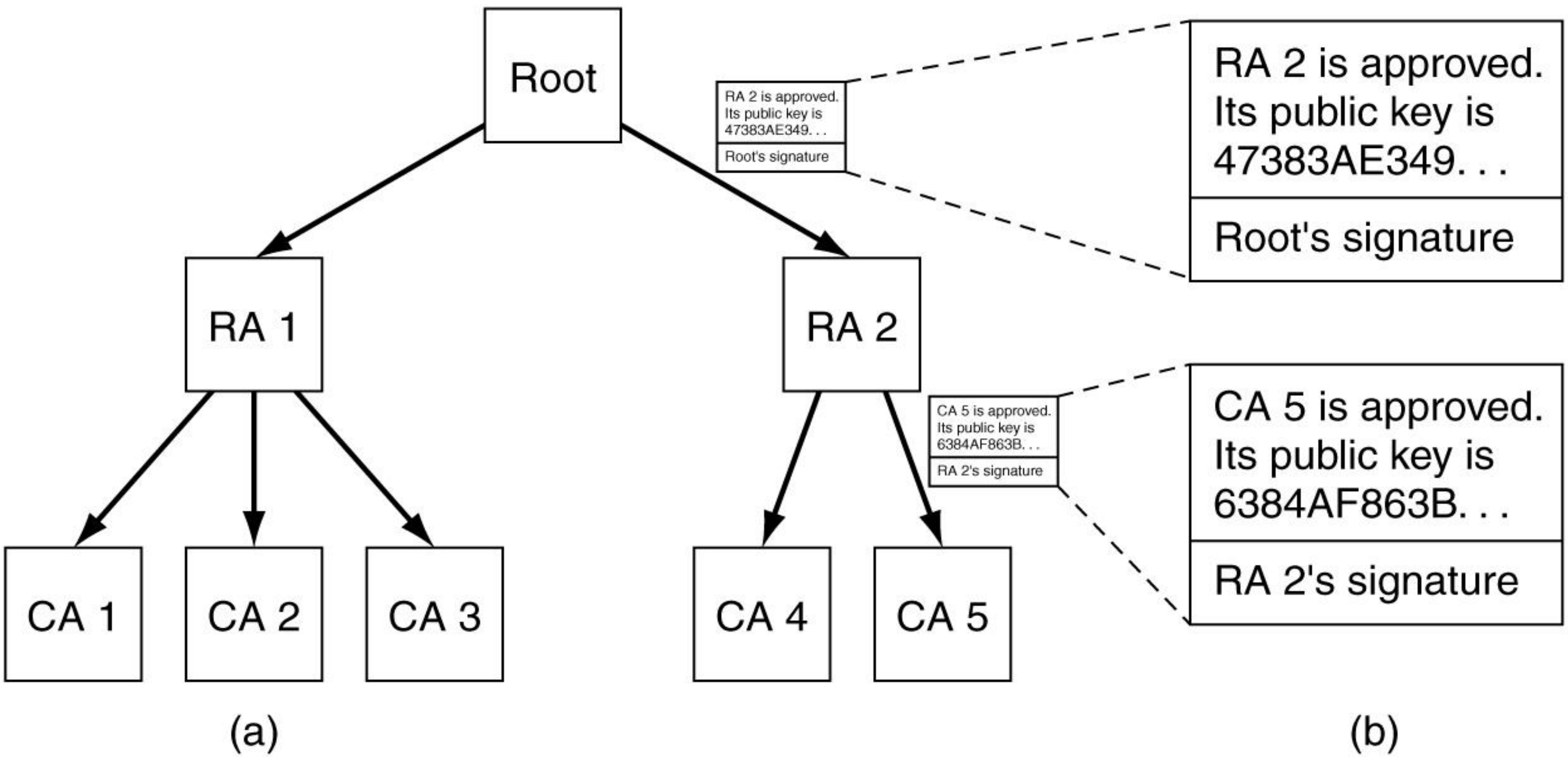
Management of Public Keys: X.509

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

Management of Public Keys: Public Key infrastructure

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



How PKI work?

- Now the certificate becomes: $\mathbf{P_B \mid\mid D_{CA5} (SHA(P_B))}$
- Alice never heard of CA5. Go to CA5 who shows the certificate with RA2's signature (containing CA5's public key).
- Alice never heard of RA2. Go to RA2 who shows the certificate with root's signature.

Management of Public Keys: PKI

- Where certificates are stored?
  - To have each user store his own certificates.
  - DNS
  - dedicated directory servers whose only job is managing X.509 certificates

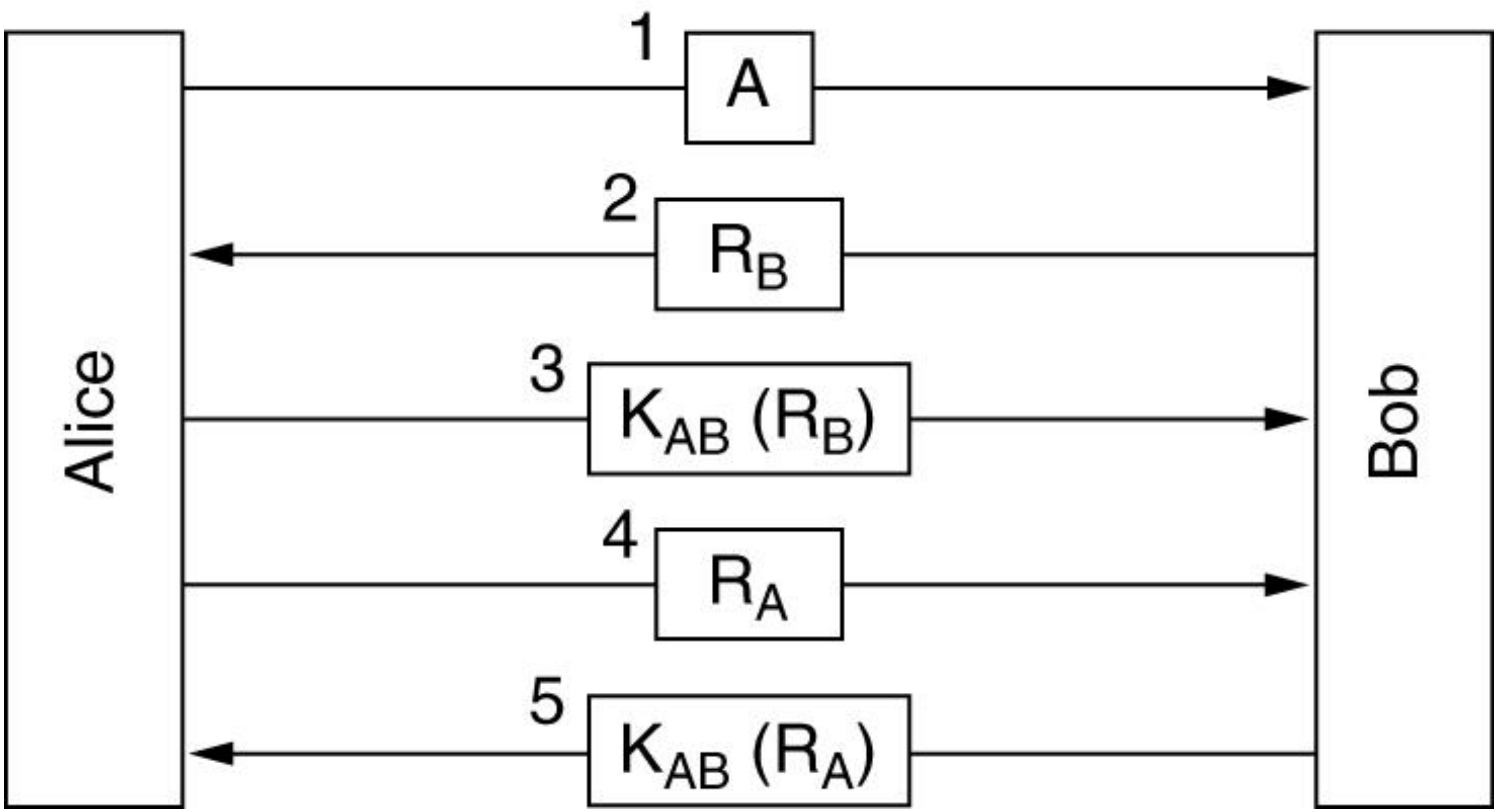
- How to revoke certificates?
  - To have each CA periodically issue a CRL (Certificate Revocation List) giving the serial numbers of all certificates that it has revoked.

AUTHENTICATION PROTOCOLS

- Authentication Based on a Shared Secret Key
- Establishing a Shared Key: Diffie-Hellman
- Authentication Using a Key Distribution Center
- Authentication Using Kerberos
- Authentication Using Public-Key Cryptography

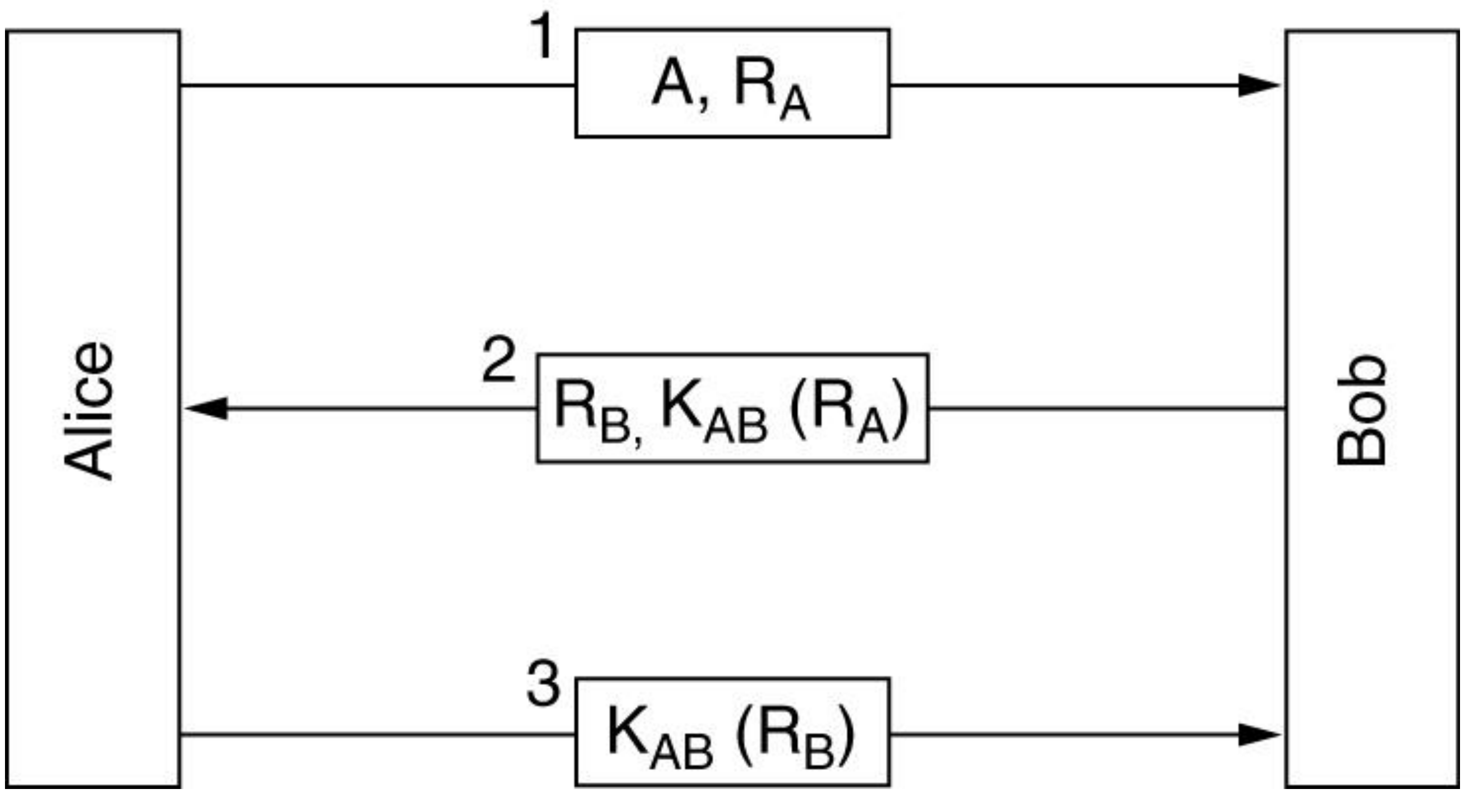
Authentication Protocols: Shared Secret Key

Two-way authentication using a challenge-response protocol. (Incorrect)



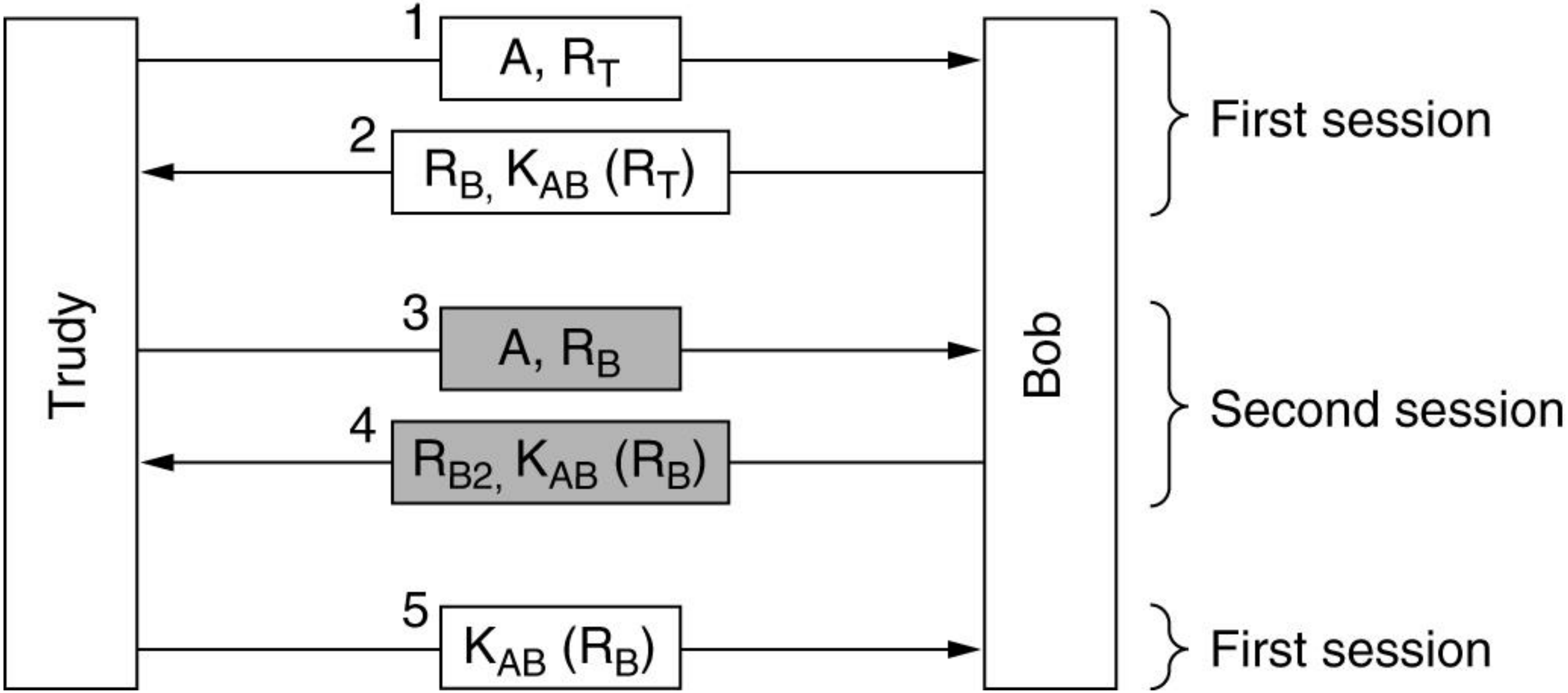
Authentication Protocols: Shred Secret Key

A shortened two-way authentication protocol. (Incorrect)



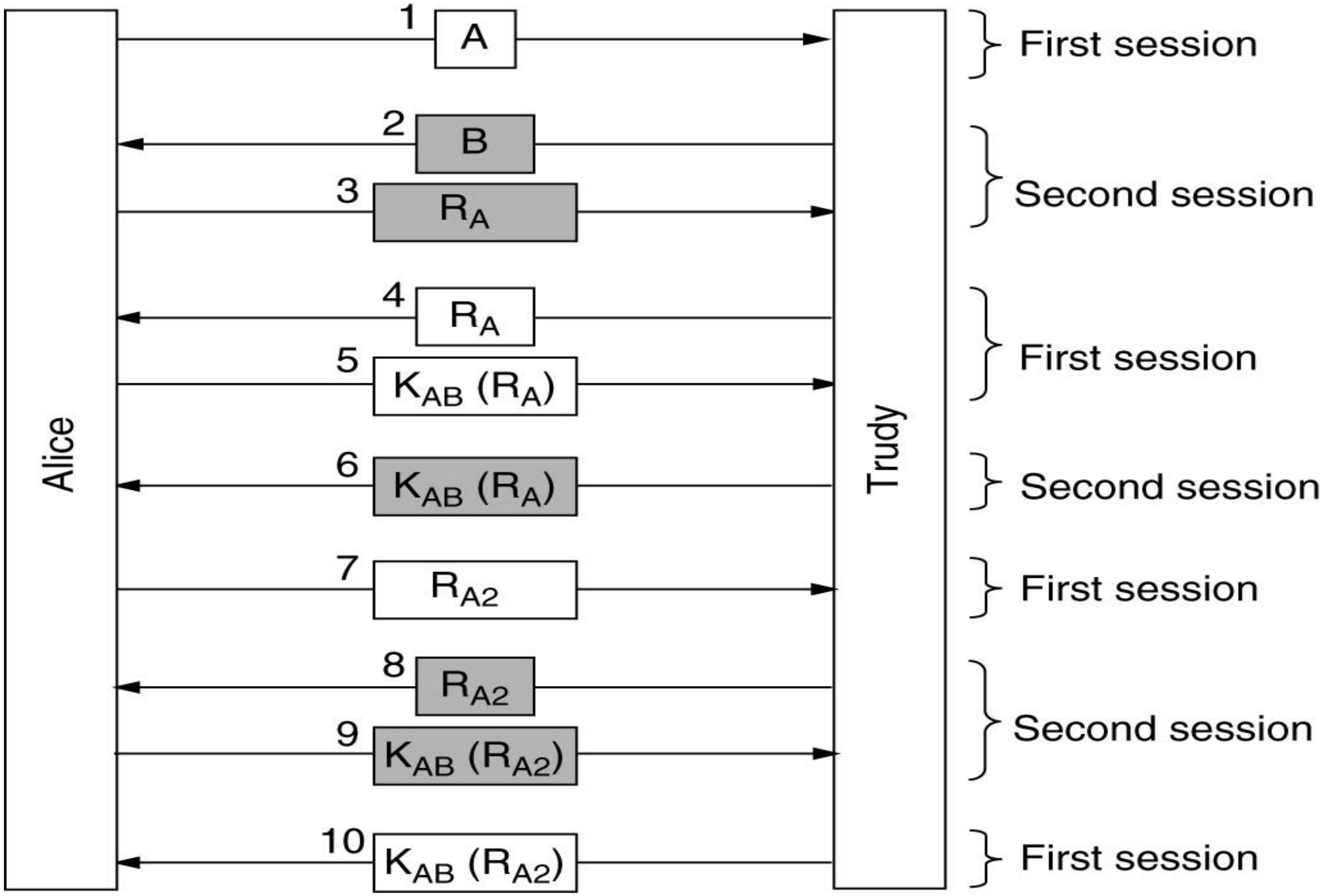
Authentication Protocols: Shared Secret Key

The reflection attack



Authentication Protocols: Shared Secret Key

Two-way authentication using a challenge-response protocol.



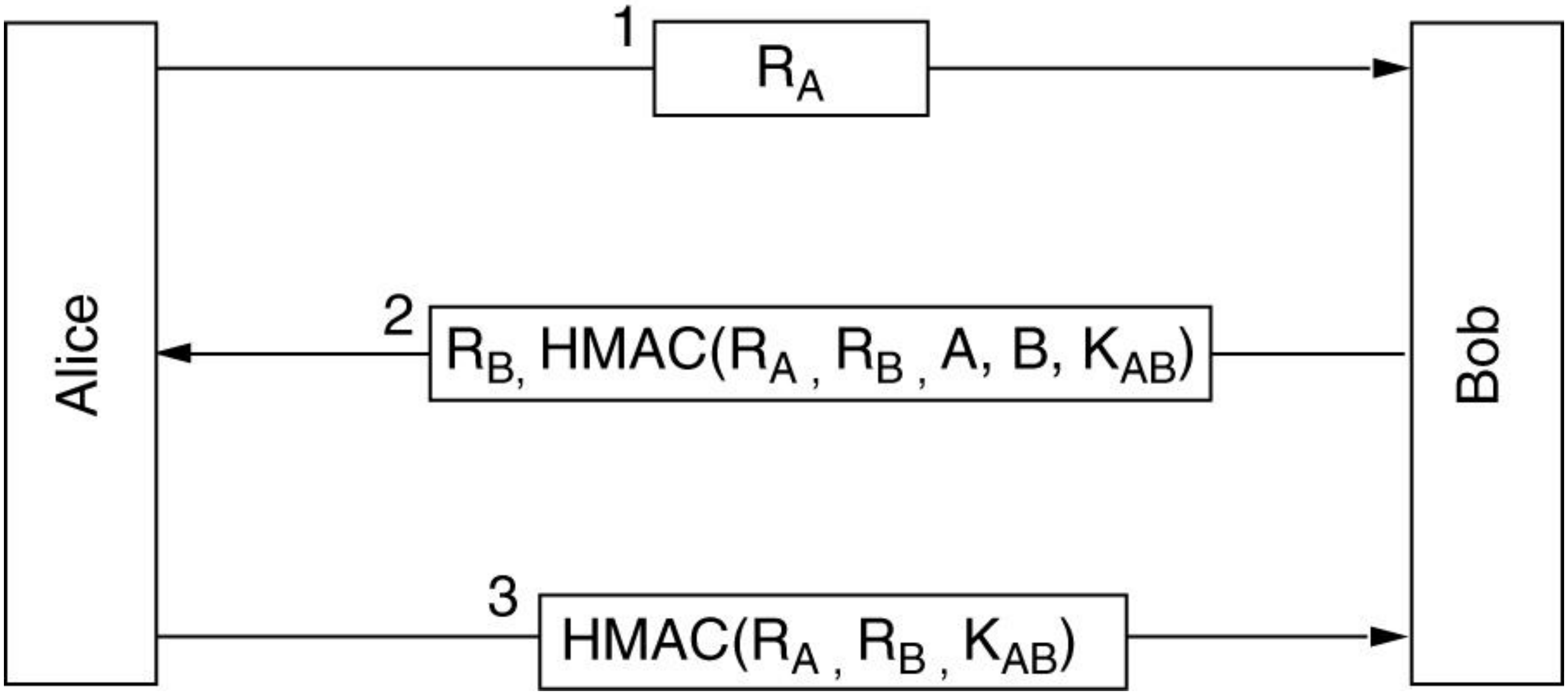
Authentication Protocols: Shared Secret Key

- Designing a correct authentication protocol is harder than it looks.
- Four general useful rules
  - Having the initiator prove what he is before the responder has to.
  - Have the initiator and responder use different keys for proof, even if this means having two shared keys.
  - Have the initiator and responder draw their challenges from different sets.
  - Make the protocol resistant to attacks involving a second parallel session in which information obtained in one session is used in a different one.

Authentication Protocols: Shared Secret Key

Authentication using HMACs. ***<u>(CORRECT)</u>***

(Hashed Message Authentication Code)

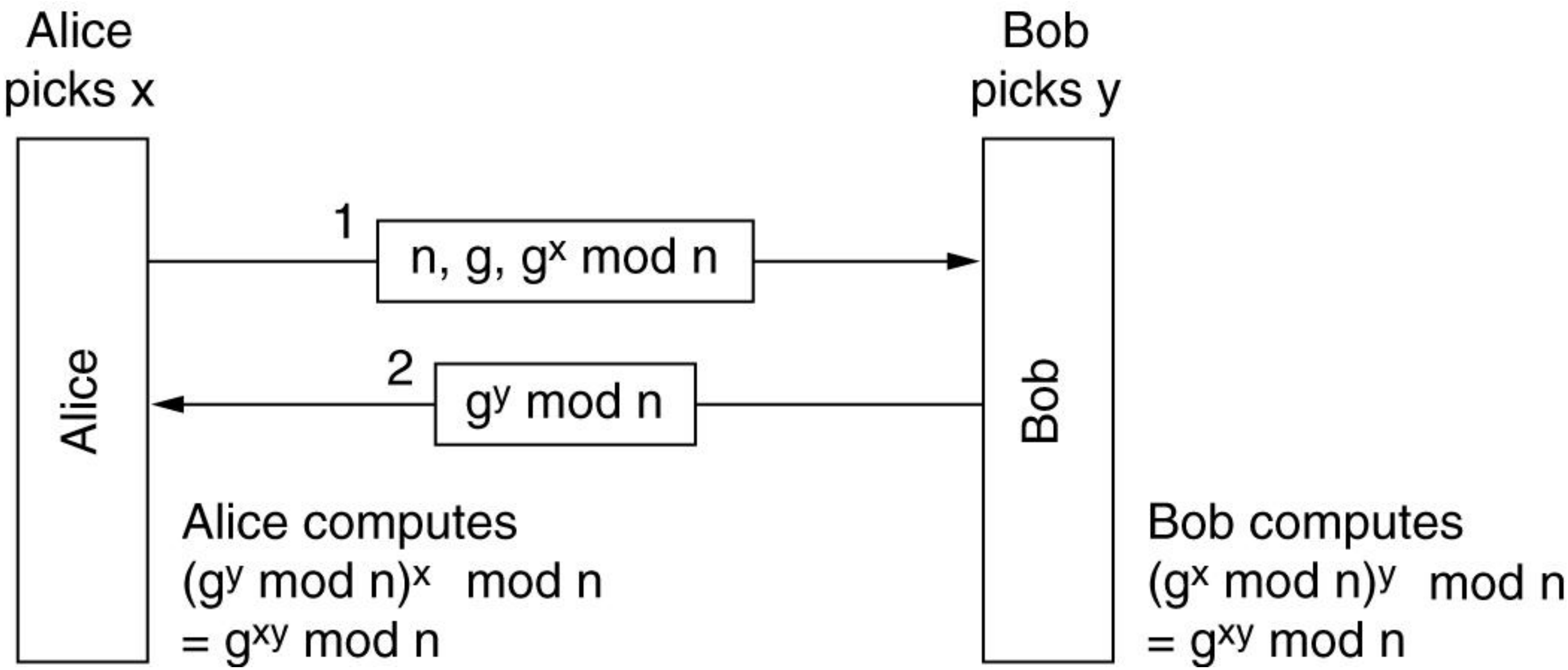


HMAC: hash over the plaintext plus the shared key, i.e. HMAC(P||K)

Authentication Protocols: Establishing a shared key

The Diffie-Hellman key exchange. (Has problems)

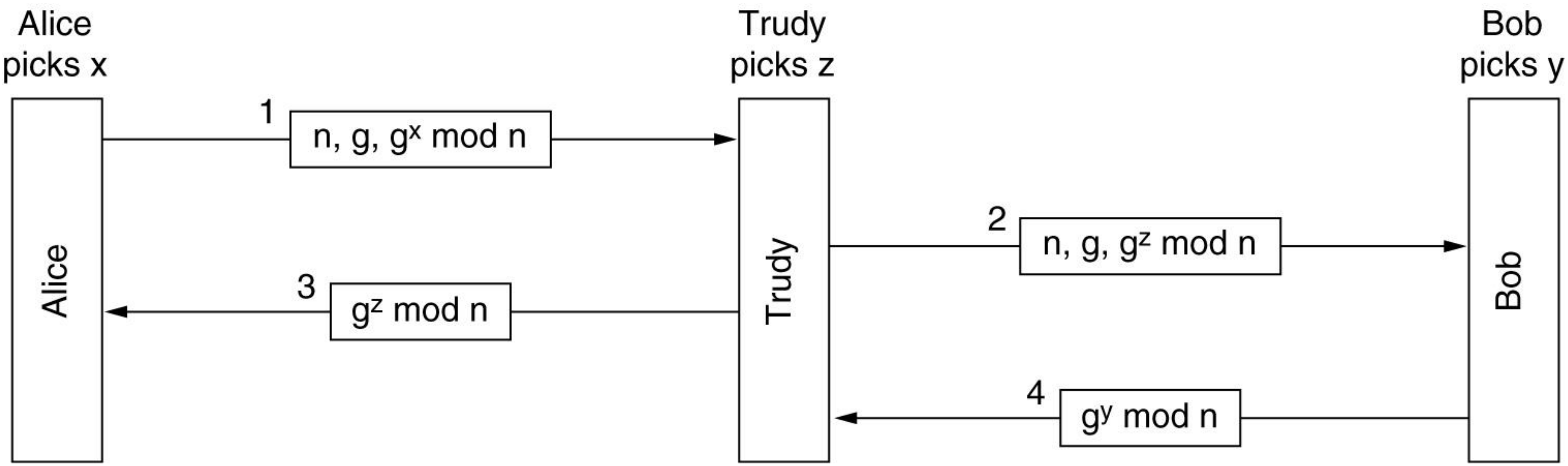
- Agree upon two large numbers: n and g
- n and $(n-1)/2$ are prime.
- Certain conditions for g.



Authentication Protocols: Establishing a shared key

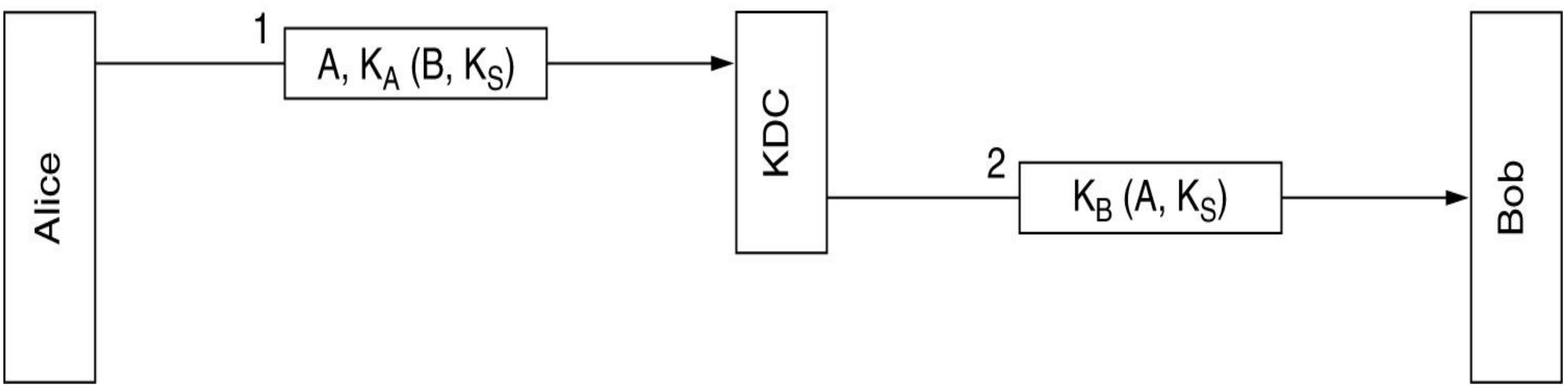
The bucket brigade (水桶队列) or,

man-in-the-middle (中间人) attack.



Authentication Protocols: Key Distribution Center

- Alice wants to establish a session key K_S with Bob
- A first attempt at an authentication protocol using a KDC. (has problems)

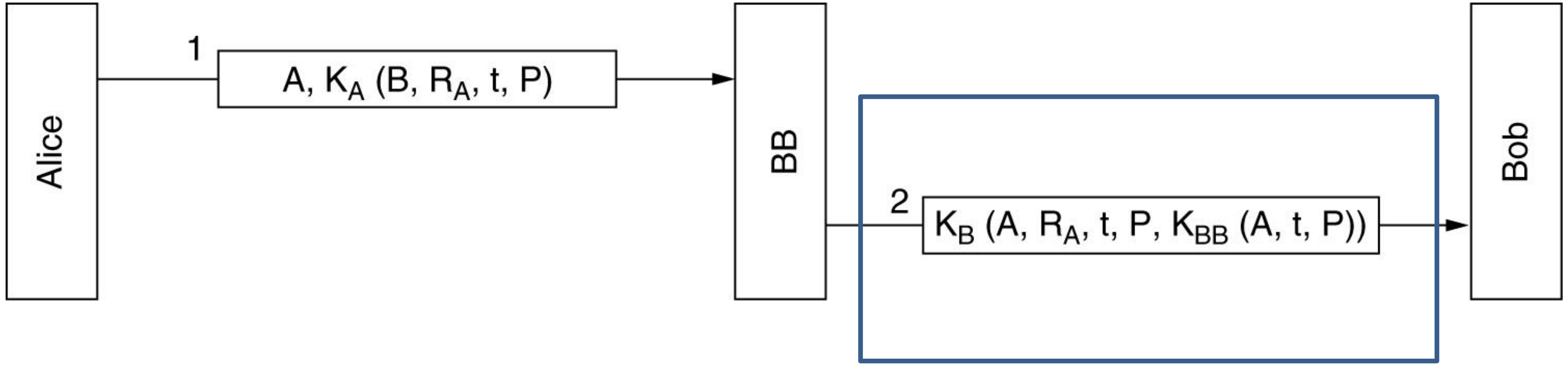


Authentication Protocols: KDC

- Replay attack for the first attempt at an authentication protocol using a KDC:
  - Trudy has done some work for Alice.
  - Alice ask Bob to pay it by bank transfer.
  - Trudy records the messages. (message 2 and following)
  - Trudy replays them again and again.
  - Bob gives Trudy a big loan to expand his “business”.

Comparison?

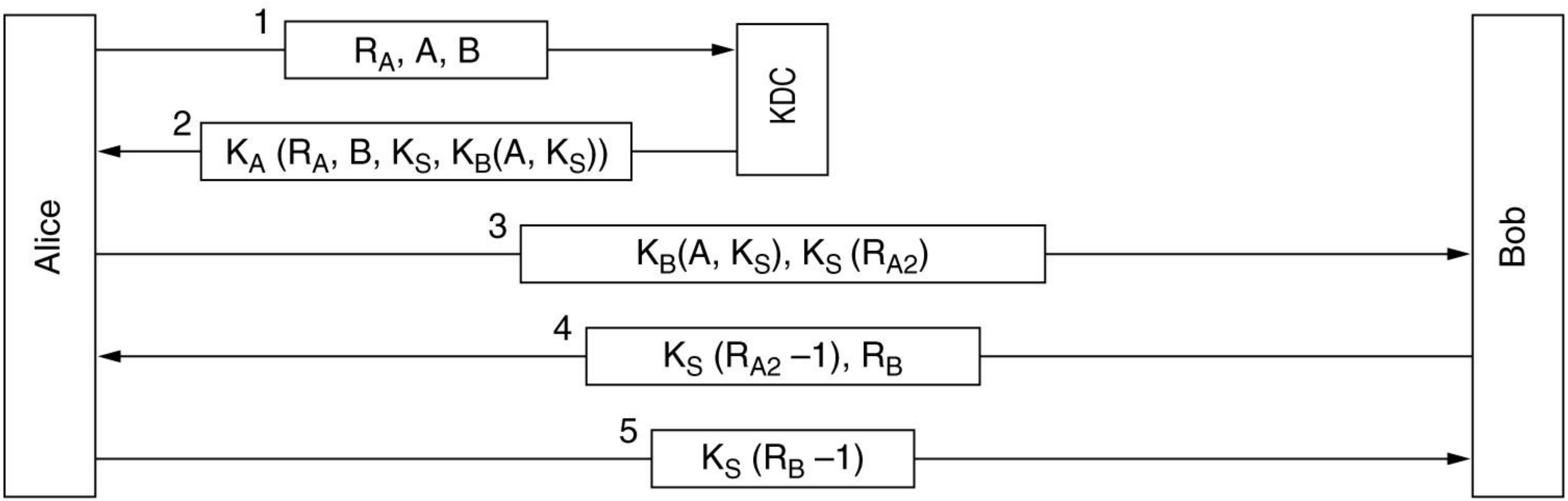
- Digital Signatures using symmetric key
  - ***The importance of t and random numbers to protect against replay attack.***



Authentication Protocols: KDC

The Needham-Schroeder authentication protocol (1978).

(has problems)

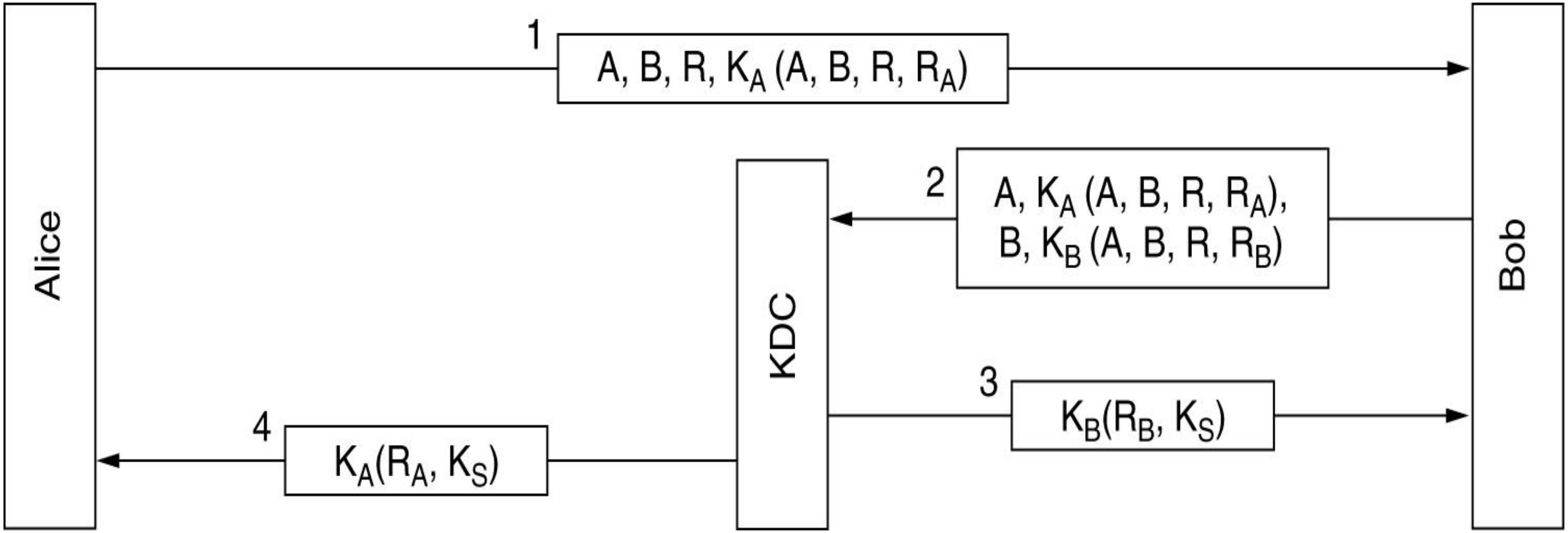


Authentication Protocols: KDC

- The problem for the Needham-Schroeder authentication protocol
  - If Trudy ever manages to obtain an old session key in plaintext,
  - she can initiate a new session with Bob by replaying the message 3 corresponding to the compromised key and convince him that she is Alice.
  - This time she can plunder Alice’s bank account without having to perform the legitimate service even once.

Authentication Protocols: KDC

The Otway-Rees authentication protocol (slightly simplified and Correct).

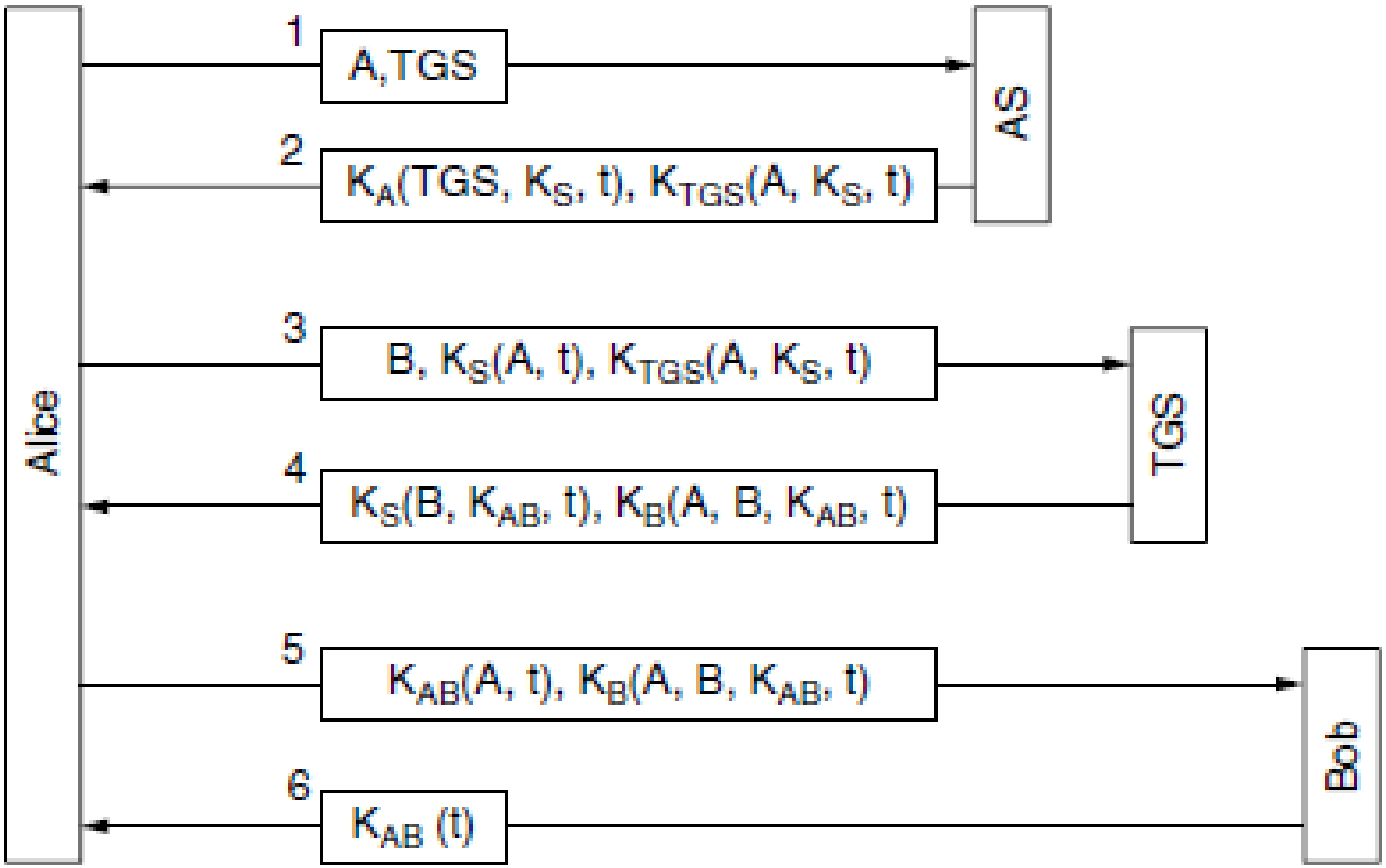


Authentication Protocols: Kerberos

- Kerberos: authentication protocols used by many practical systems.
  - AS (Authentication Server): verifies users during login. Similar to KDC
  - TGS (Ticket-Granting Server): Issues "proof of identity tickets"
  - Bob server: actually does the work Alice wants performed.

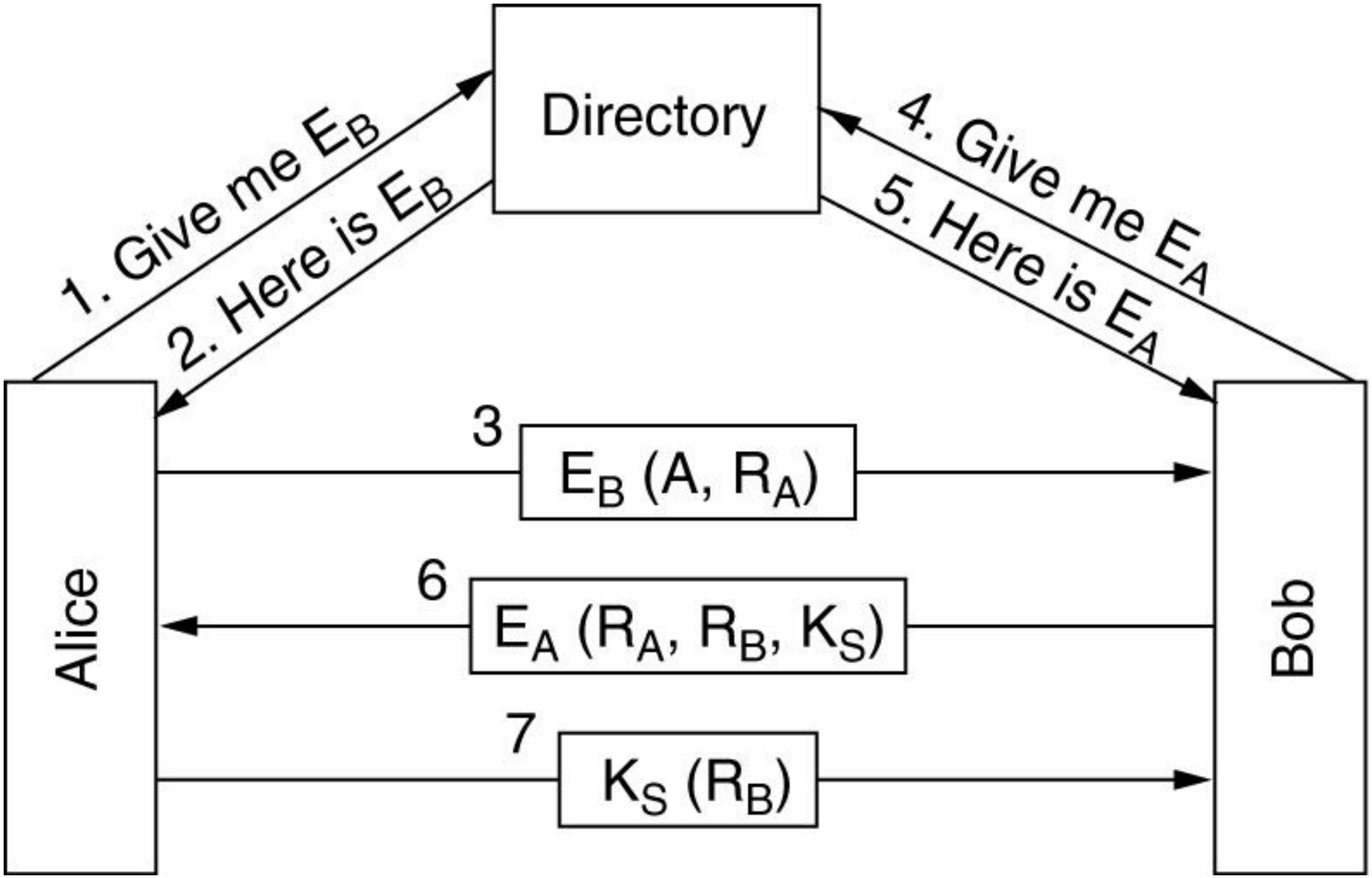
Authentication Protocols: Kerberos

The operation of Kerberos V5.



Authentication Protocols: Public-key

Mutual authentication using public-key cryptography.



Authentication Protocols: Public-key

- If Trudy?
- → $E_B(A, R_T)$
- ← $E_A(R_T, R_B, K_S)$
- → does not know K_S since Trudy has no D_A

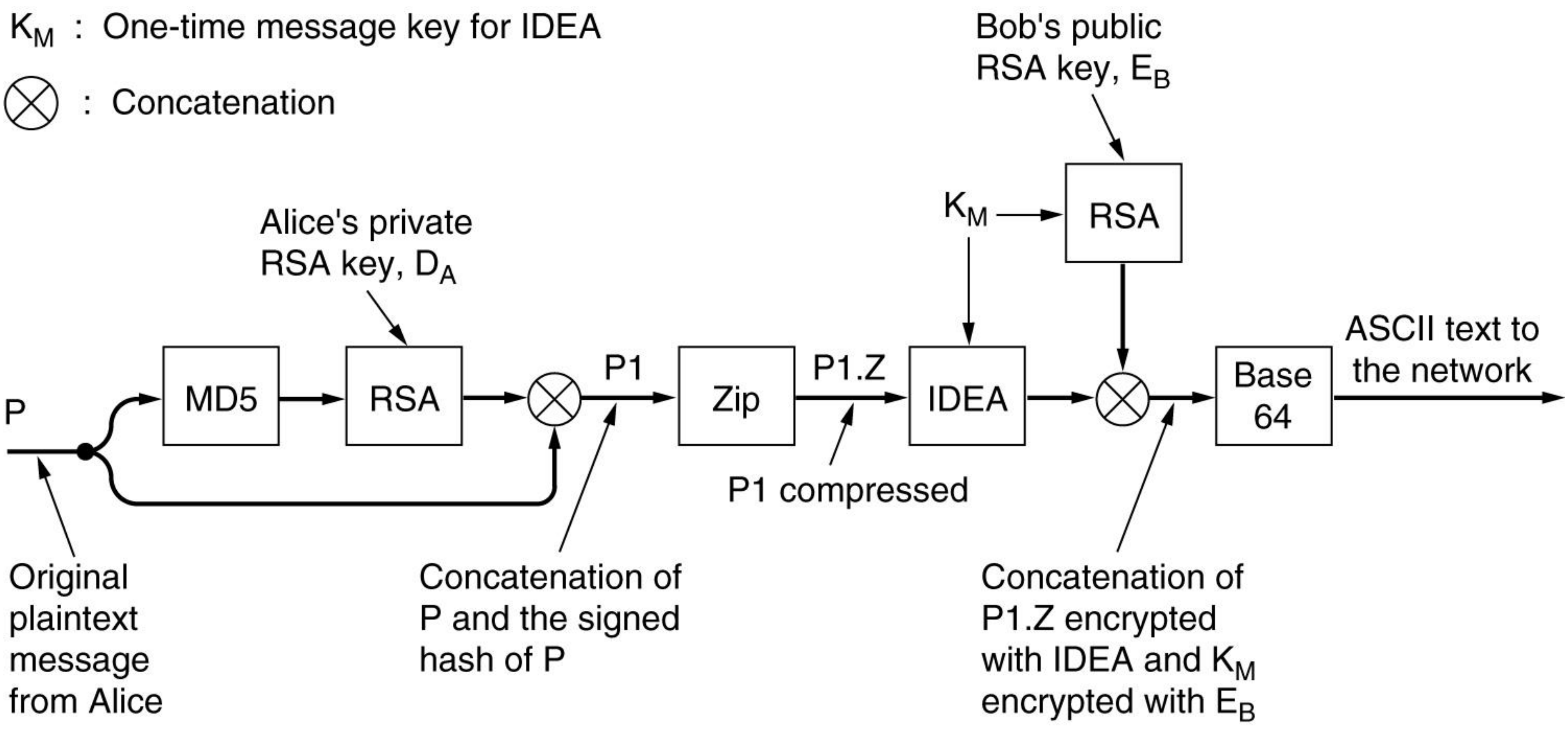
E-Mail Security

- PGP – Pretty Good Privacy
- PEM – Privacy Enhanced Mail
- S/MIME

E-Mail Security: PGP

- PGP is the brainchild of one person, Phil Zimmermann.
- PGP is a complete e-mail package that provides privacy, authentication, digital signatures, and compression, all in an easy-to-use form. Released in 1991.
- PGP
  - Encryption is done by IDEA (International Data Encryption Algorithm)
  - Key management is done by RSA
  - Data integrity uses MD5.

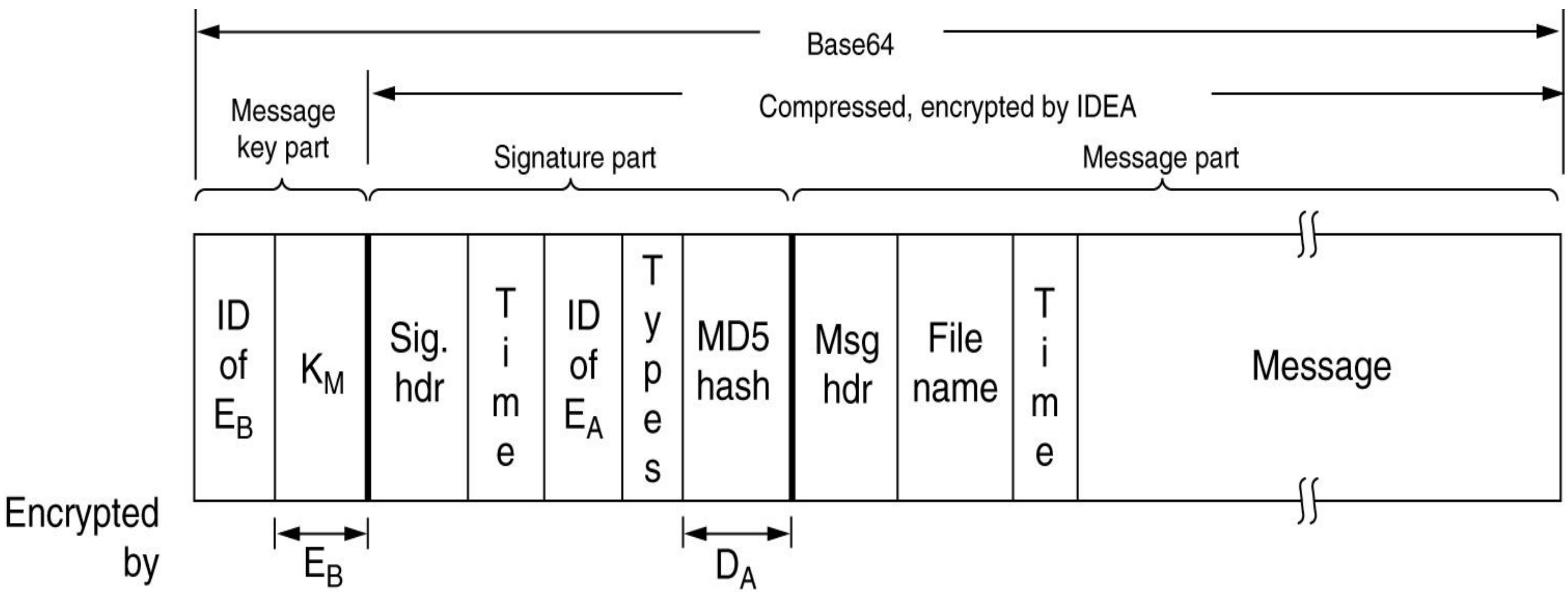
E-Mail Security: PGP



$$K_M\ (\ ZIP(\ P \,||\, D_A\,(MD\ (P))\ )\ ) \,||\, E_B(K_M)$$

E-Mail Security: PGP

A PGP message



E-Mail Security: PEM and S/MIME

- PEM (Privacy Enhanced Email), developed in the late 1980s, is an official Internet standard and described in 4 RFCs: RFC 1421 through RFC 1423.
- PEM has long-since gone to that big bit bin in the sky.
  - Why? Key management problem

- S/MIME (Secure/MIME) is described in RFCs 2632 though 2643.

COMMUNICATION SECURITY

- IPsec
- Firewalls
- Virtual Private Networks
- Wireless Security

Communication Security: IPsec

- IETF has known for years that security was lacking for the Internet.
- Where to add it?
  - Application layer?
    - Total solution.
    - Rewrite many many applications.
  - Transport layer?
    - Help security-unaware users to some extent.
    - No need to rewrite so many applications.
    - → IPsec (IP Security)

Communication Security: IPsec

- The complete IPsec design is a framework for multiple services, algorithms and granularities
  – Multiple services: Not everyone wants to pay the price for having all the services all the time, so the services are available a la carte. The major services are secrecy, data integrity, protection from replay attack.
  – Multiple algorithms: The framework can survive even if some particular algorithm is later broken.
  – Multiple granularities: to make it possible to protect a single TCP connection, all traffic between a pair of hosts, or all traffic between a pair of secure routers, among other possibilities.

Communication Security: IPsec

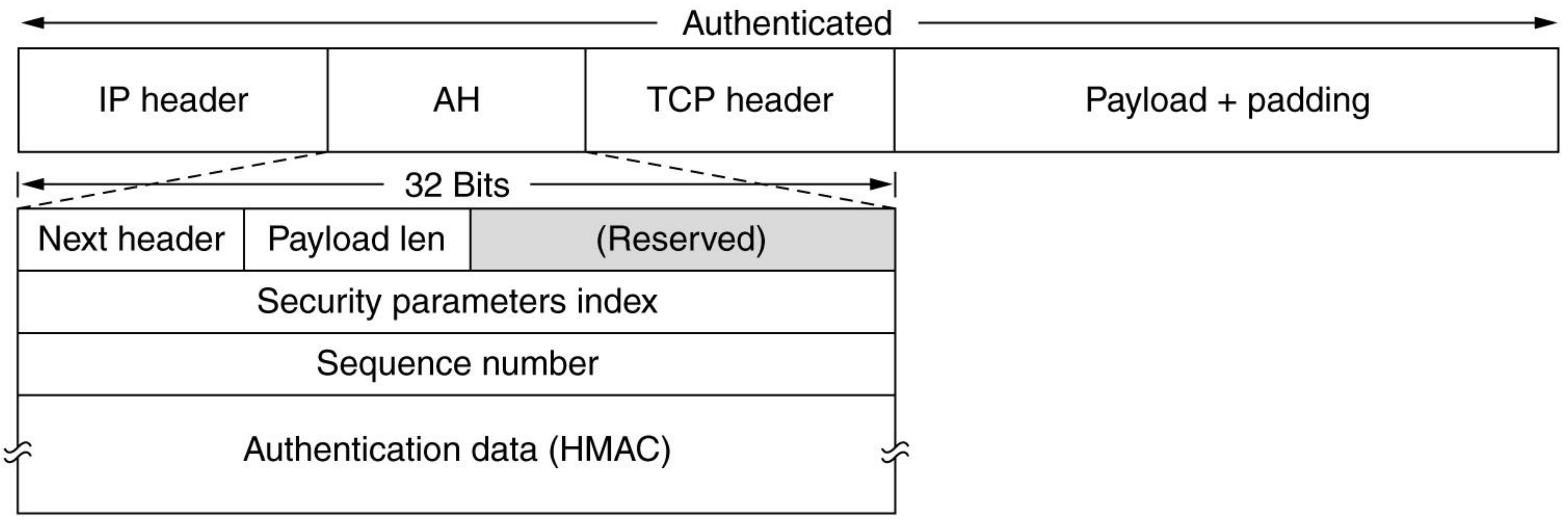
- IPsec is connection oriented.
- A “connection” in the context of IPsec is called an SA (security association).
  - An SA is a simplex connection.
  - An SA has an identifier associated with it.
  - Security identifiers are carried in packets traveling on secure connections and are used to look up keys and other relevant information when a secure packet arrives.
- IPsec has two parts.
  - To establish keys (ISAKMP, Internet Security Association and Key Management protocol)
  - To describe two new headers: AH and ESP.

Communication Security: IPsec

- IPsec can be used in two modes:
  - In transport mode: The IPsec header is inserted just after the IP header. The protocol field in the IP header is changed to indicate that an IPsec header follows the normal IP header (before the TCP header). The IPsec header contains security information, primary the SA identifier, a new sequence number, and possible an integrity check of the payload.
  - In tunnel mode: the entire IP packet, header and all, is encapsulated in the body of a new IP packet with a completely new IP header.

Communication Security: Ipsec: AH

The IPsec **authentication header** in transport mode for IPv4.



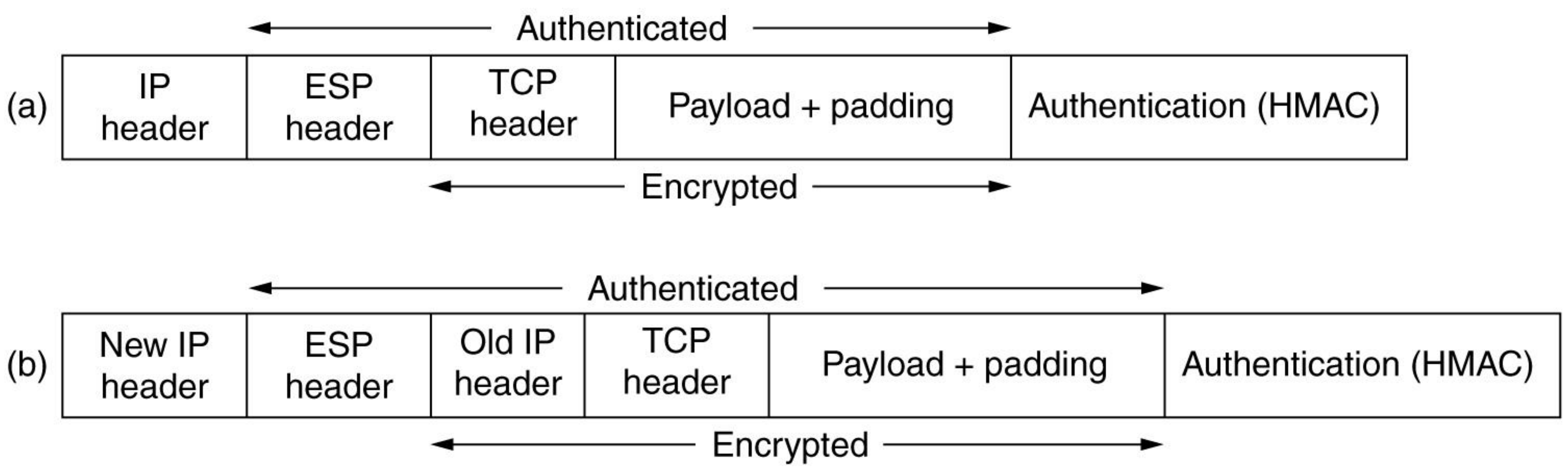
Communication Security: Ipsec: AH

- Authentication Header
  - Next header: to sore the previous value that the IP Protocol field had before it was replaced with 51 to indicated that an AH header follows.
  - Payload length: the number of 32-bit words in the AH header minus 2.
  - Security parameter index or SA: the connection identifier.
  - Sequence number: to number all the packets sent on an SA.
  - Authentication header: to contain the payload’s digital signature.
- Useful when integrity checking is needed but not secrecy is not needed.

Communication Security: Ipsec: ESP

(a) ESP in transport mode.

(b) ESP in tunnel mode.



ESP: ESP header, ESP Payload, ESP trailer, ESP Authentication.

Communication Security: Ipsec: AH or ESP?

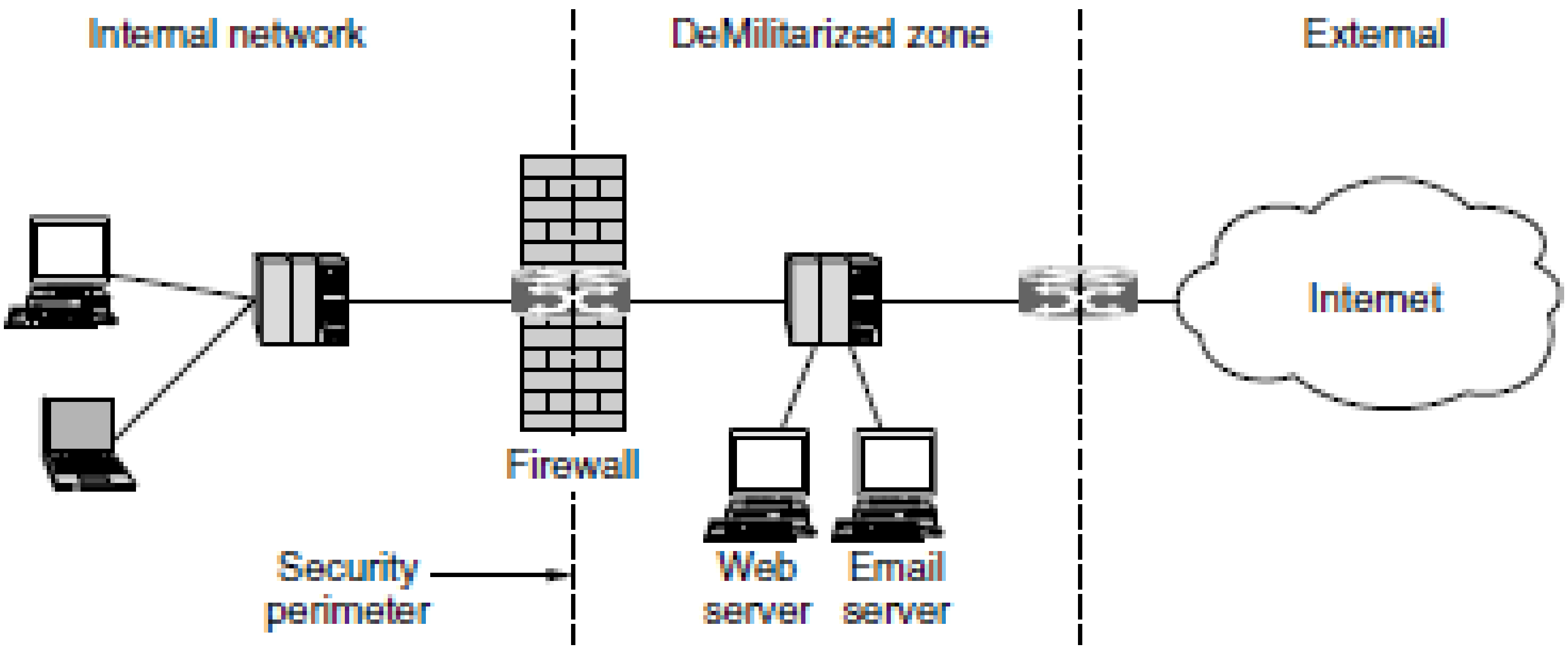
- Originally, AH handled only integrity and ESP handled only secrecy. Now, ESP has added integrity and thus can do everything.
- A product supporting AH but not ESP might have less trouble getting an export license because it cannot do encryption.
- → AH is likely to be phased out in the future.

Communication Security: Firewalls

- The ability to connect any computer, anywhere to any other computer, anywhere is a mixed blessing.
  - For individuals at home, wandering around the Internet is lots of fun.
  - For corporate security managers, it can be a nightmare
    - Let some information leaking out.
    - Let some information leaking in.
- IPsec is good for protecting data in transit. Something is required for disallowing some data transfer.

Communication Security: Firewalls

A firewall protecting an internet network



Communication Security: Firewalls

- The packet filter on the inside
  - Check the outgoing packets.
- The application gateway
  - For further examination of packets.
- The packet filter on the outside:
  - Check the incoming packets.

Communication Security: Firewalls

- Packet filter routers are typically driven by tables configured by the system administrator
  - To list source and destinations that are acceptable.
  - To list source and destinations that are blocked.
  - Default rules about what to do with packets coming from or going to other machines.
- Application gateways (or proxy routers)
  - Header fields,
  - Message size,
  - Content (nuclear, bomb, terror).

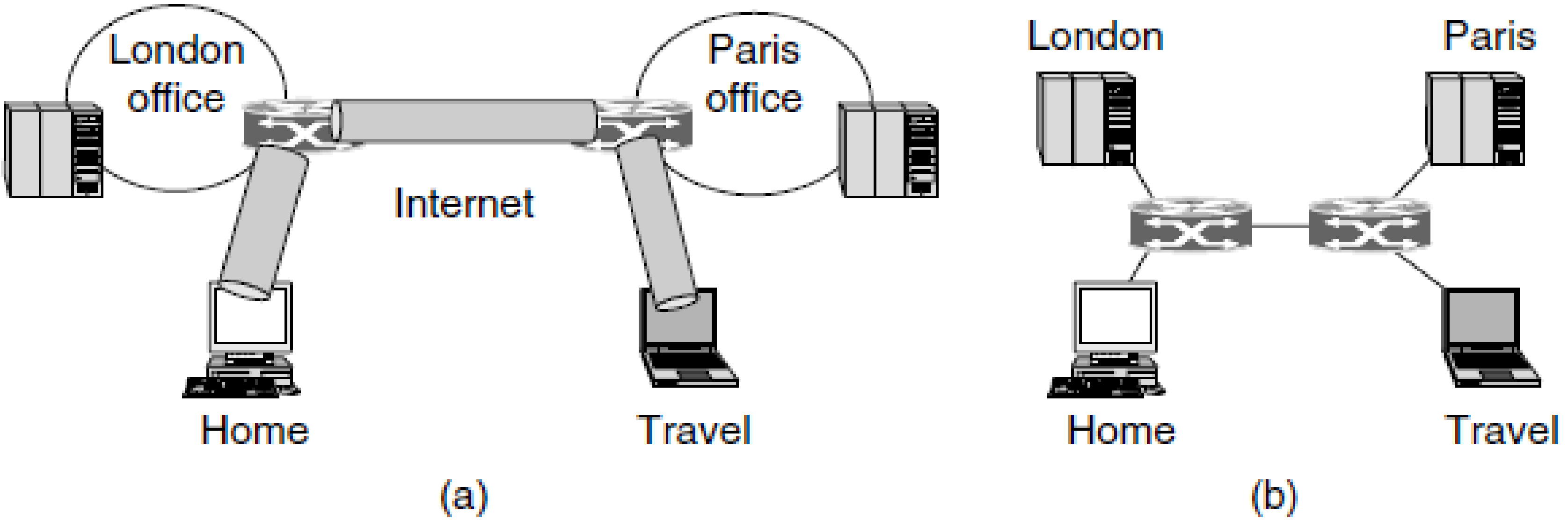
Communication Security: Firewalls

- Even if the firewall is perfectly configured, plenty of security problems still exist.
  - An intruder outside the firewall can put in false source addresses to bypass this check .
  - An insider can encrypt or even photograph documents and then ship them.
  - DoS (Denial of Service): an intruder can send so many TCP SYN packets that the server will send SYN+ACK packets and wait for the respond.
  - DDoS (Distributed Denial of Service)
    - The intruder bring down many computers.
    - And command all of them to attack the same target.

Communication Security: VPN

a) A virtual private network.

b) Topology as seen from the inside



Communication Security: VPN

- VPN
  - Frame relay, ATM, Internet.
  - Firewall + IPsec
    - To equip each office with a firewall and create tunnels through the Internet between all pairs offices.
    - To use IPsec for the tunnelling

Communication Security: Wireless Security

- 802.11 Security:
  - WEP (Wired Equivalent Privacy): data link level security protocol
- Bluetooth Security

Communication Security: Wireless Security

- Breaking the WEP is easy!
- Statements from IEEE
  - We told you that WEB security was no better than Ehternet's
  - A much bigger thread is forgetting to enable security at all.
  - Try using some other security (e.g. transport layer security)
  - The next version, 802.11i, will have better security.
  - Future certification will mandate the use of 802.11i.
  - We will try to figure out what to do until 802.11i arrives.

Communication Security: Wireless Security

- Bluetooth security
  - Three security modes, ranging from nothing at all to full data encryption and integrity control.
  - No security: locking the barn door after the horse has escaped.
  - Physical layer: frequency hopping.
  - Passykeys:
    - Checking for the passkey.
    - Selecting a random 128 bit session key for encryption.
    - Encryption use a stream cipher called E0; integrity control uses SAFER+.

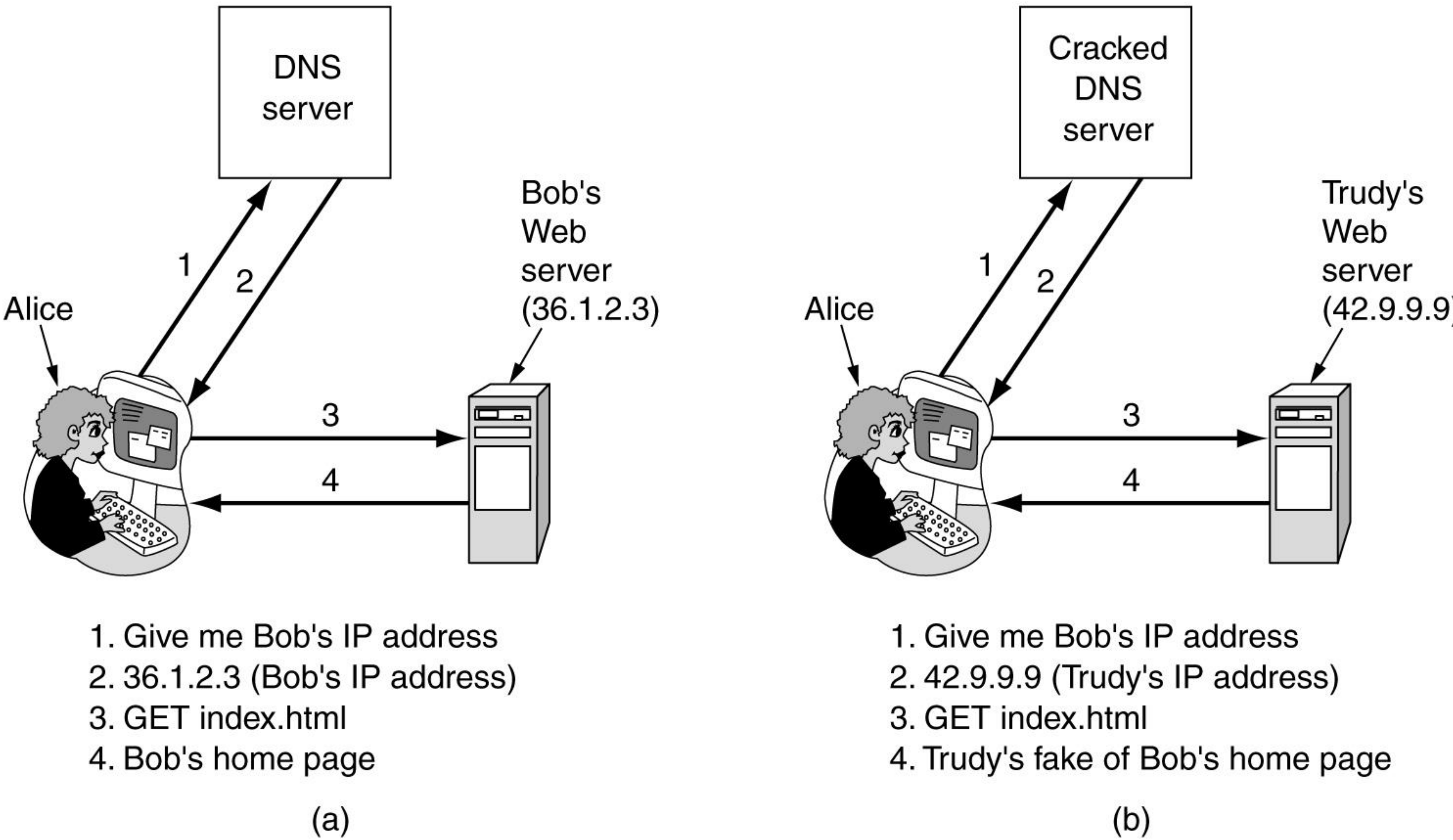
Web Security

- Threats
- Secure Naming
- SSL – The Secure Sockets Layer
- Mobile Code Security

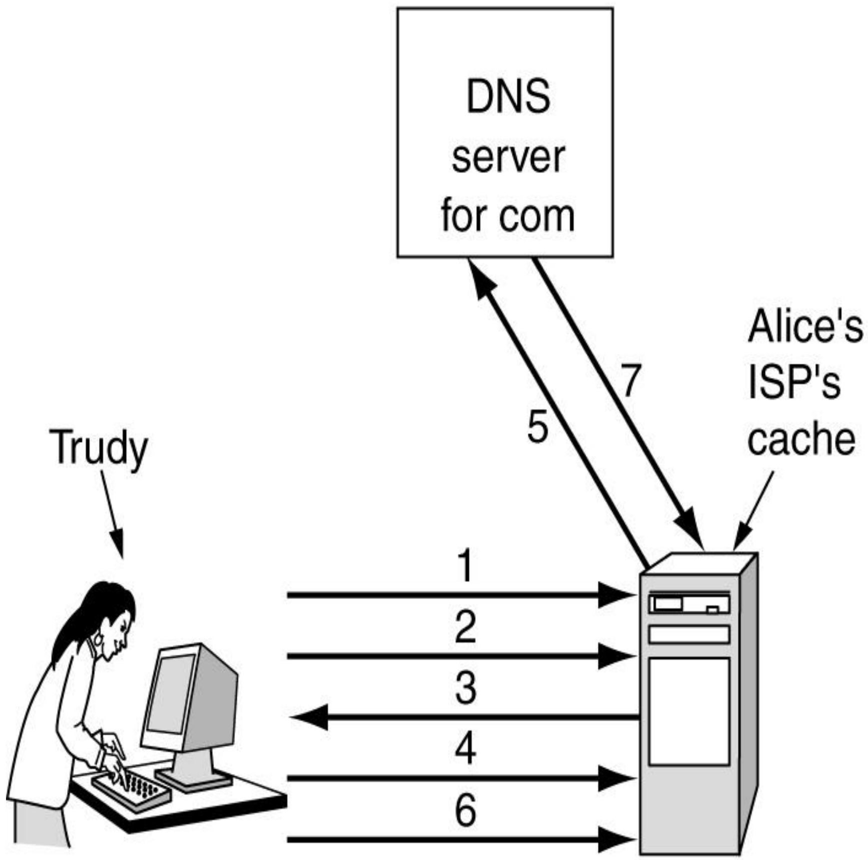
Web Security: Security

- Hacker(电脑高手,电脑黑客) vs cracker(解密高手)
- In 1999, a Swedish cracker broke into Microsoft's Hotmail Web site.
- A 19-year-old Russian cracker named Maxim broke into an e-commerce Web site and stole 300,000 credit card numbers.
- A 23-year-old California student emailed a press release to a news agency falsely stating that the Emulex Corporation was going to post a large quarterly loss and that the C.E.O. was resigning immediately.
- ……

Web Security: Secure Naming



Web Security: Secure Naming



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)

Web Security: Secure DNS

- DNSsec is based on public-key cryptography. Every DNS zone has a public/private key pair. All information sent by a DNS server is signed with the originating zone's private key, so the receiver can verify its authenticity.
- DNSsec offers three fundamental services:
  - Proof where the data originated
  - Public key distribution
  - Transaction and request authentication.

Web Security: Secure DNS

- DNS records are grouped into sets called RRSets (Resource Record Sets)
  - An RRSet may contain multiple records.
  - Each RRSet is crptographically hased and the hash is signed by the zon’es private key (e.g. using RSA)
  - Upon receipt of a signed RRSet, the client can verify whether it was signed by the private key of the originating zone.

Web Security: Secure DNS

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 has of the *A* and *KEY* records to verify their authenticity.

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...

Web Security: Self-Certifying Names

A self-certifying URL containing a hash of server's name and public key.

Web Security: SSL

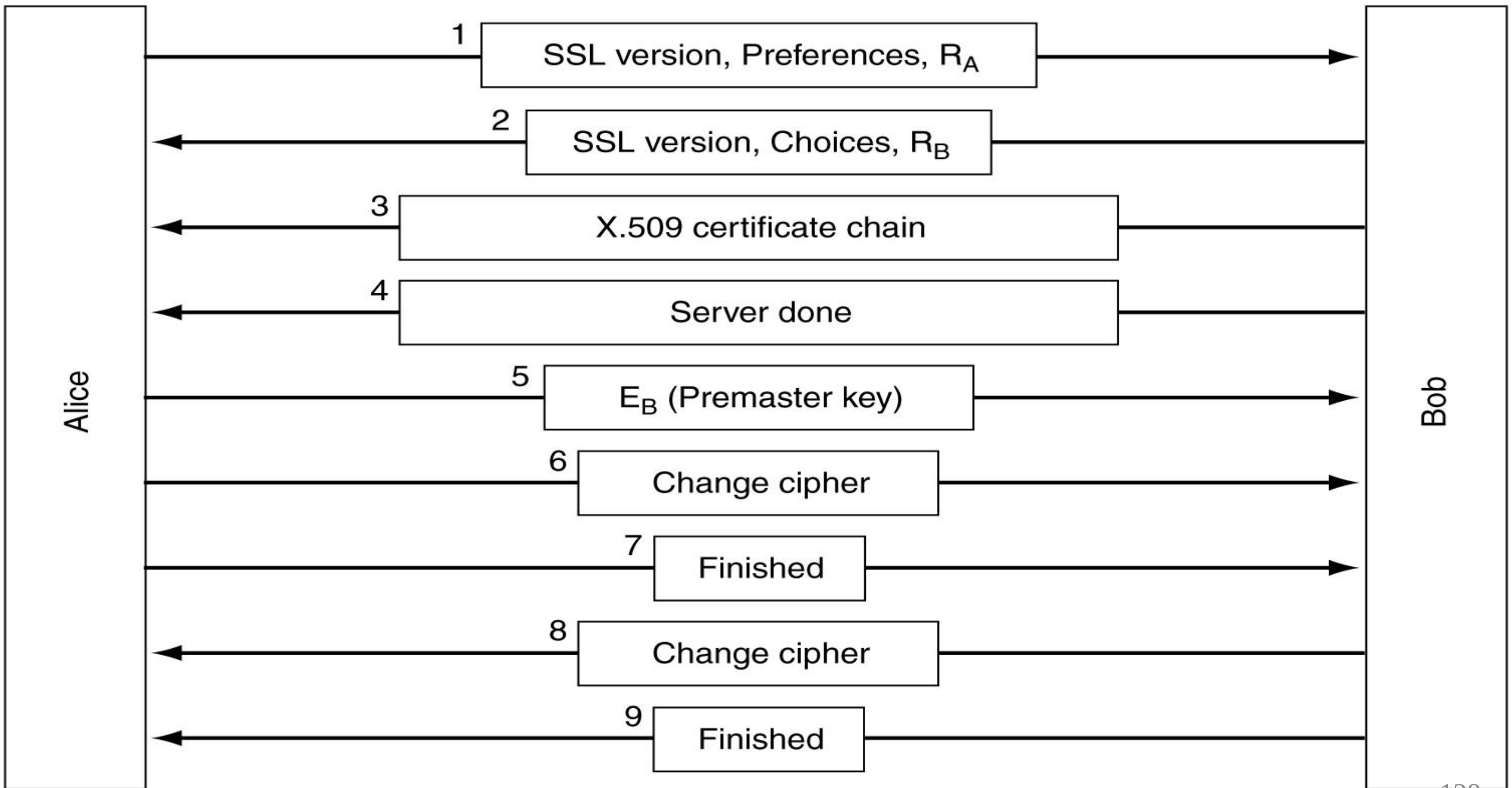
Layers (and protocols) for a home user browsing with SSL.

HTTPS (Secure HTTP) is the HTTP used over SSL.

Application (HTTP)
Security (SSL)
Transport (TCP)
Network (IP)
Data link (PPP)
Physical (modem, ADSL, cable TV)

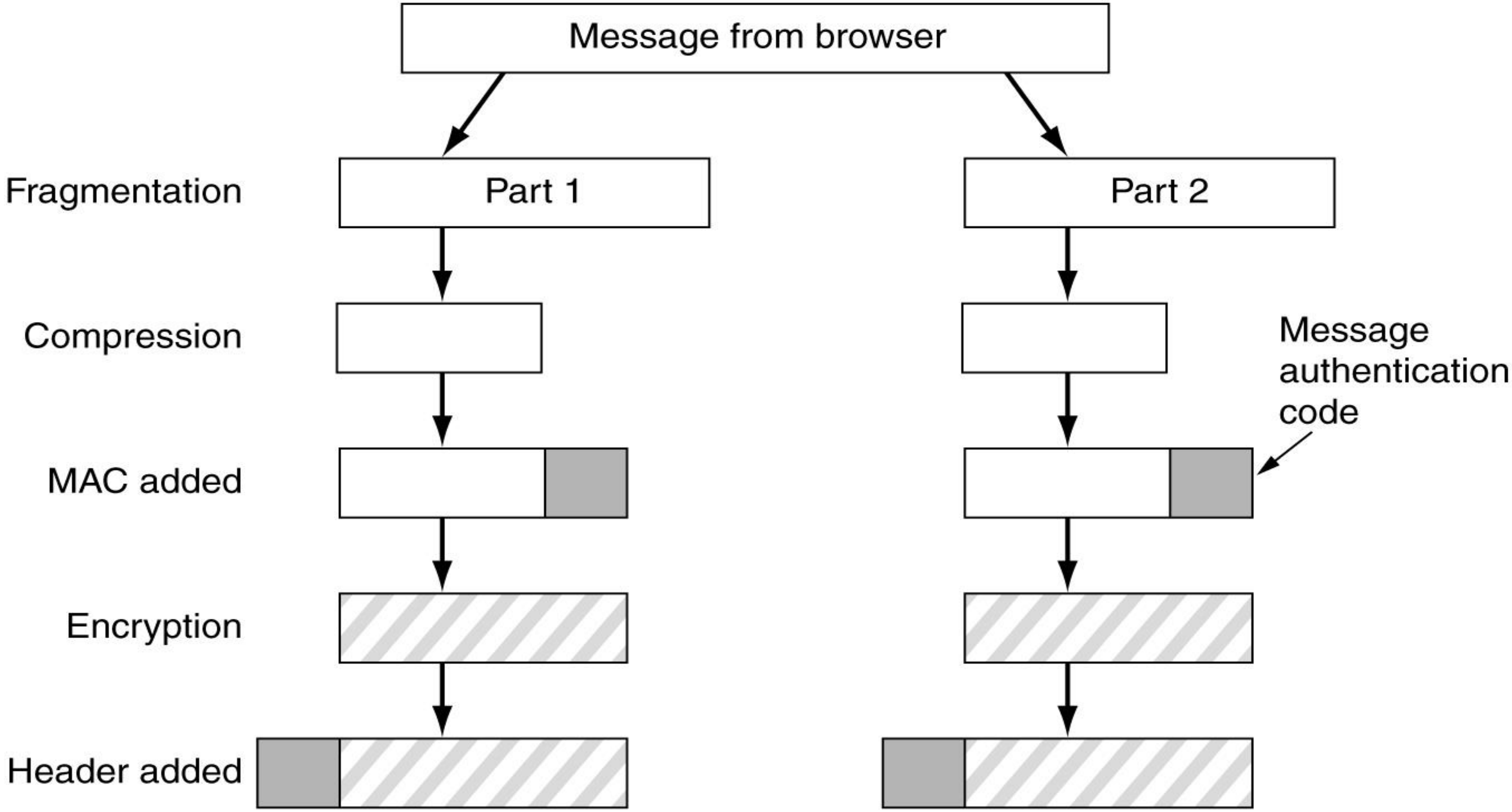
Web Security: SSL

A simplified version of the SSL connection establishment subprotocol.



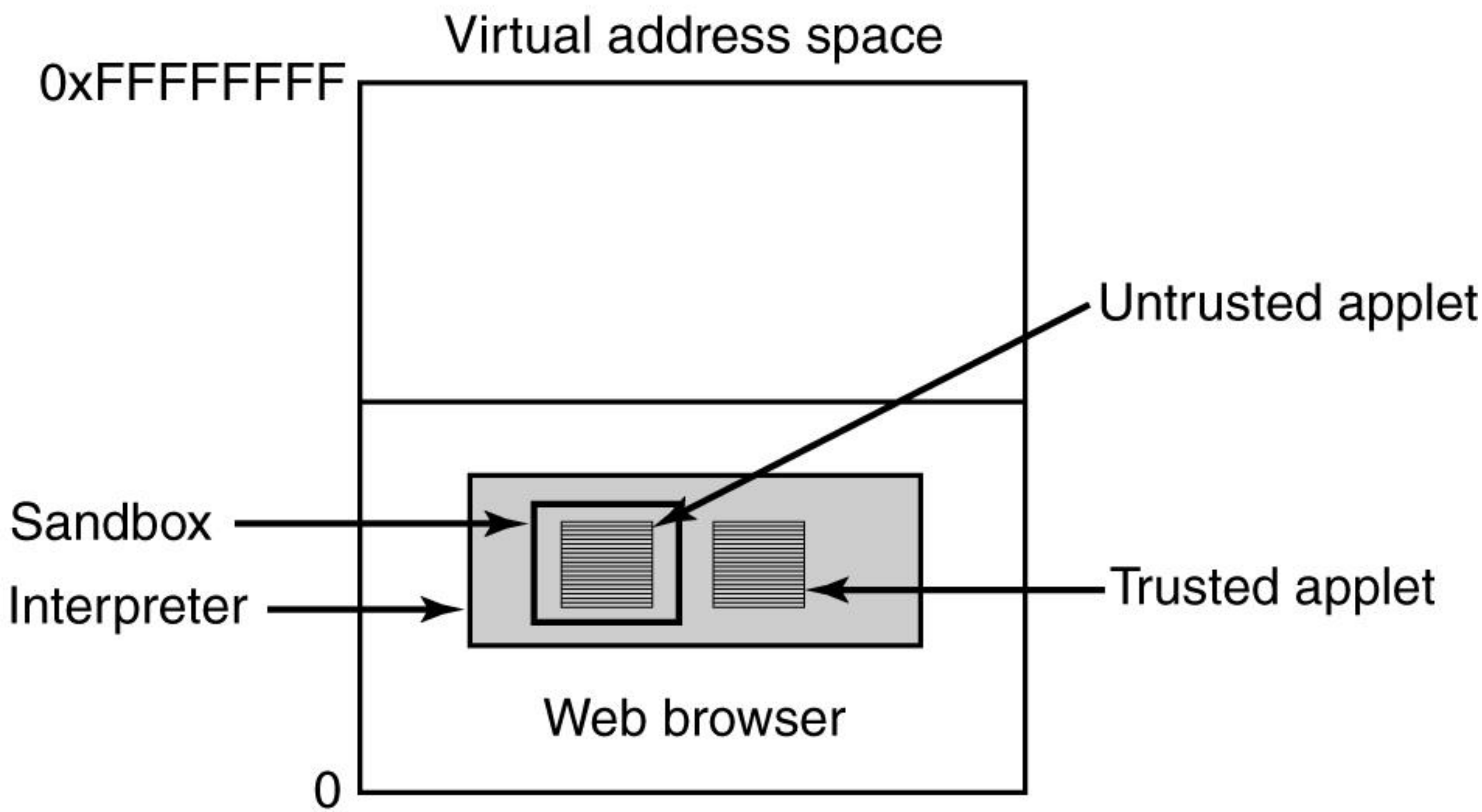
Web Security: SSL

Data transmission using SSL.



Web Security: Mobile Code Security

Applets inserted into a Java Virtual Machine interpreter inside the browser.

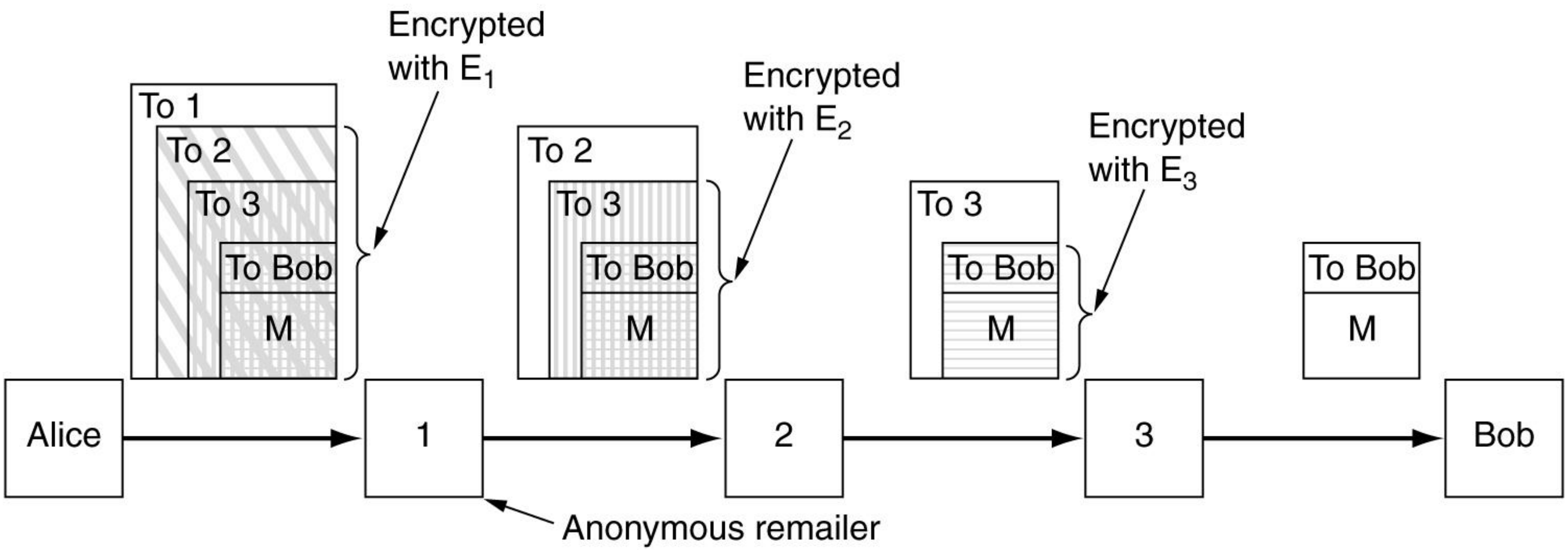


Social Issues

- Privacy
- Freedom of Speech
- Copyright

Social Issues: Privacy

Users who wish anonymity chain requests through multiple anonymous remailers.



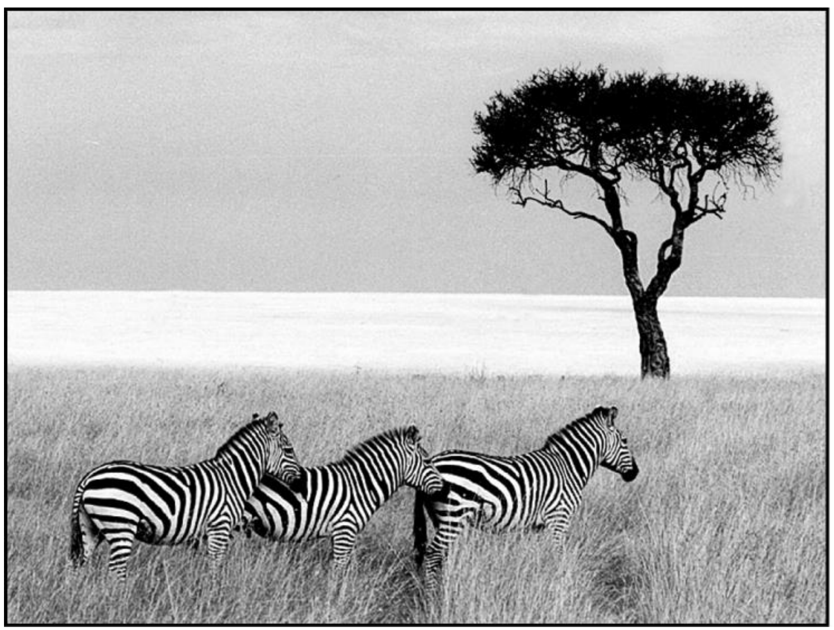
Social Issues: Freedom of Speech

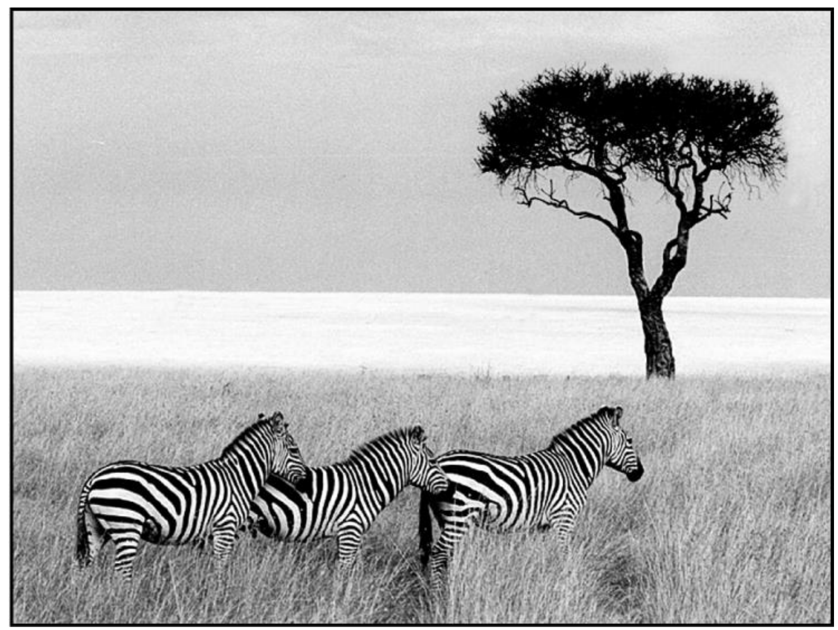
- Possibly banned material:
  1. Material inappropriate for children or teenagers.
  2. Hate aimed at various ethnic, religious, sexual, or other groups.
  3. Information about democracy and democratic values.
  4. Accounts of historical events contradicting the government's version.
  5. Manuals for picking locks, building weapons, encrypting messages, etc.

Social Issues: Freedom of Speech

(a) Three zebras and a tree.

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





Social Issues: Copyright

- Copyright or Copyleft?