山东大学
Shandong University

Software School of Shandong University

Security Protocols and Standards

--- PKI Principles and technology sd03031340

(Chapter 1 Review to Cryptography)

Instructor: Hou Mengbo 侯孟波
Email: houmb AT sdu.edu.cn
Office: Information Security Research Office.
Office Rm: Office Building 421

Course Info. (1)

Semester: 2019-2020-2 **Credit Hour**：2

Course Name: Security Protocols and Standard 安全协议与标准

ClassTime: 1-16 total 16 weeks Experiments 16 hours

Course No	sd03031340
audience	NS 17 & JS17 1-8
Total students	22+47
ClassTime	1. Weds. 5-6(13:30:-15:30PM)
Classroom	5-107d

Course Info.(2)

- Chinese textbook (for reference)
  - PKI技术，荆继武，林璟锵,冯登国　　科学出版社
  - PKI原理与技术,　　谢冬青等编著　　清华大学出版社
  - 精通PKI网络安全认证技术与编程实现　马臣云等编　　人民邮电出版社
  - PKI/CA与数字证书技术大全　张明德，刘伟 著　电子工业出版社
- Useful English Texts

(1) Public Key Infrastructure Implementation and Design, by Suranjan etc, Published by M&T Books, 2003(*)

(2) 公钥基础设施（PKI）实现和管理电子安全, PKI Implementing and Managing E-Security， Andrew Nash等著，张玉清等译　清华大学出版社

(3) Deploying a Public Key Infrastructure, by *Heinz etc,2000,* IBM Corporation, International Technical Support Organization

(4) PKI Security Solutions for the Enterprise: Solving HIPAA, E-Paper Act, and Other Compliance Issues, by Kapil Raina, Published by Wiley Publishing, Inc., 2003(*)

(5) Understanding PKI: Concepts, Standards, Deployment and Considerations Addison Wesley, 2002 (*)

(6) RSA Security's Official Guide to Cryptography, by Steve Burnett, The McGraw-Hill Companies,2001

(7) Digital Signature, By Mohan Atreya etc, The McGraw-Hill Companies,2001, 贺军等译,清华大学出版社,2003

(8) Internet Security Cryptographic Principles, Algorithms and Protocols, Man Young Rhee, John Wiley & Sons Ltd,2003

（E-BOOKs are available…）

Course Theme

- PKI concepts / content / significance
- Core PKI Services
- Certificate and CRL
- Trust Model
- PKCS
- CA Standards
- PKI Security Protocols SMIME/ SSL / SET / VPN…
- PKI / CA Design and implementation
- Related Laws
- * Bitcoin & Blockchain

- First, We should review cryptography …

Security Requirements

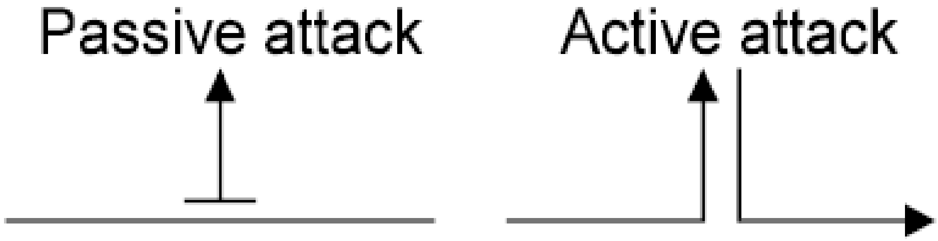
- Confidentiality
  - Protection from disclosure to unauthorised persons
- Integrity
  - Maintaining data consistency
- Authentication
  - Assurance of identity of person or originator of data
- Non-repudiation
  - Originator of communications can’t deny it late
- Access control
  - Un-authorised users are kept out

Security Requirements (ctd.)

- Availability
  - Legitimate users have access when they need it
- ……

- These are often combined
  - User authentication used for access control purposes
  - Non-repudiation combined with authentication

Attack Types



Passive attack can only observe communications or data

Active attack can actively modify communications or data

- Often difficult to perform, but very powerful
  - Mail forgery/modification
  - TCP/IP spoofing/session hijacking

Security Services

From the OSI definition:

- Authentication: Provides assurance of someone's identity
  - Often confused with authorization
- Confidentiality: Protects against disclosure to unauthorized identities
- Integrity: Protects from unauthorized data alteration
- Non-repudiation: Protects against the originator of communications later denying it
- Access control: Protects against unauthorized use

Security Mechanisms

Three basic building blocks are used:

- **Encryption** is used to provide confidentiality, can provide authentication and integrity protection
- **Digital signatures** are used to provide authentication, integrity protection, and non-repudiation
- **Checksums/hash algorithms** are used to provide integrity protection, can provide authentication

One or more security mechanisms are combined to provide a security service.

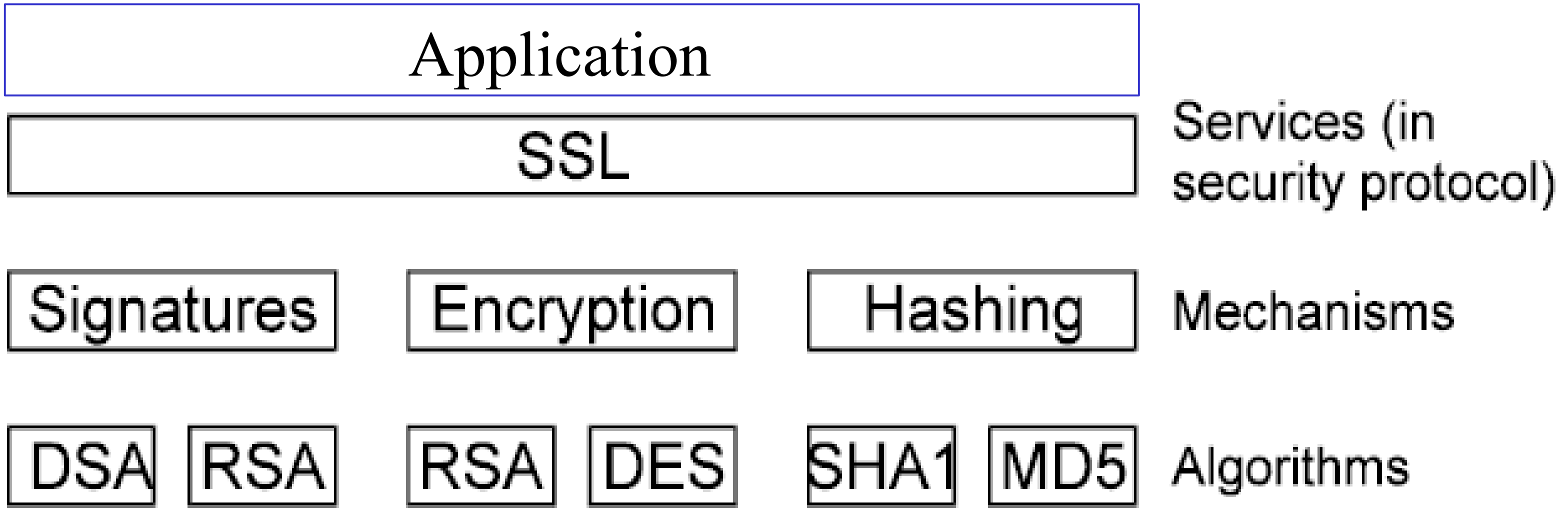
Security Protocols

- What’s **protocols** ？

- What is **security protocol**?

Algorithms, Mechanisms, Services

A typical security protocol provides one or more services



Application

SSL | Services (in security protocol)

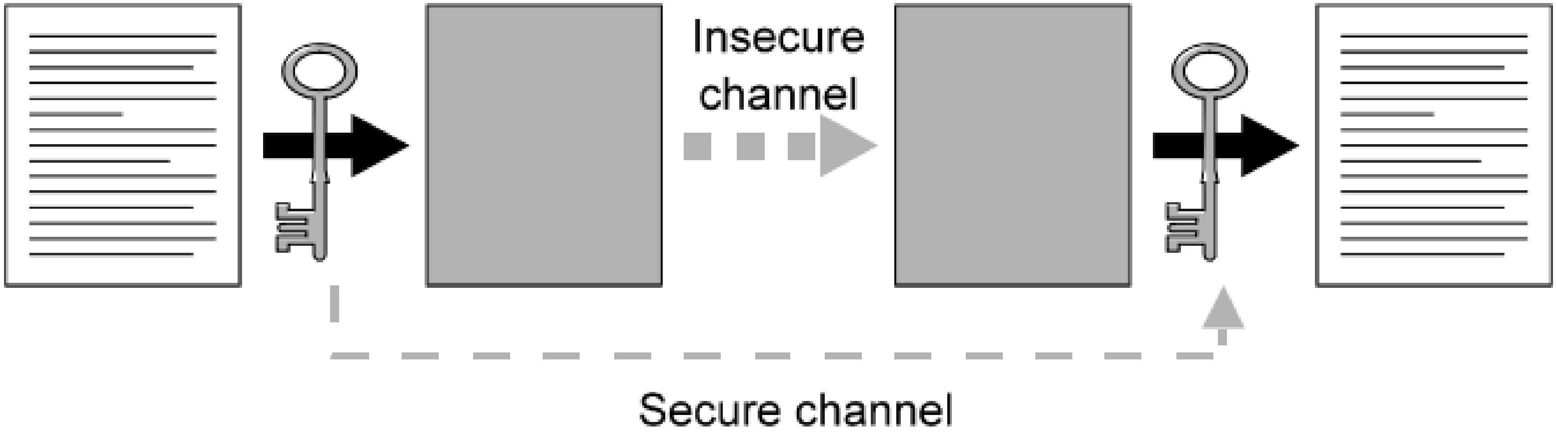
Signatures | Encryption | Hashing | Mechanisms

DSA | RSA | RSA | DES | SHA1 | MD5 | Algorithms

- Services are built from mechanisms
- Mechanisms are implemented using algorithms

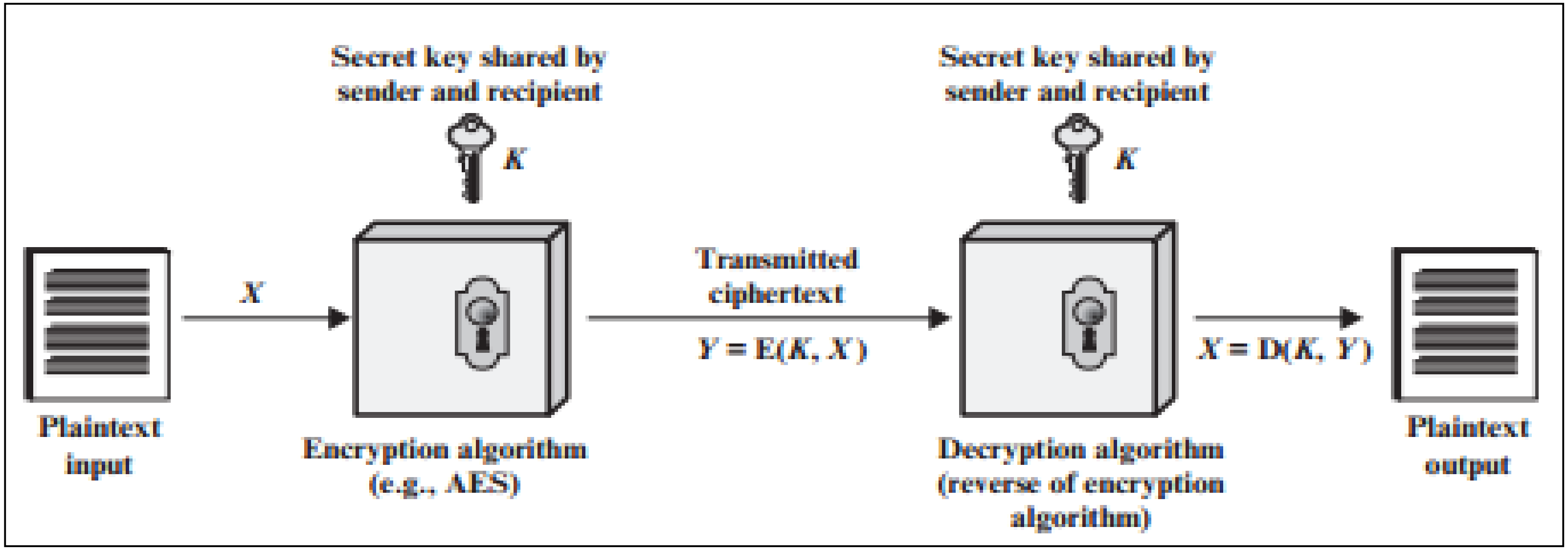
Algorithms : Conventional Encryption

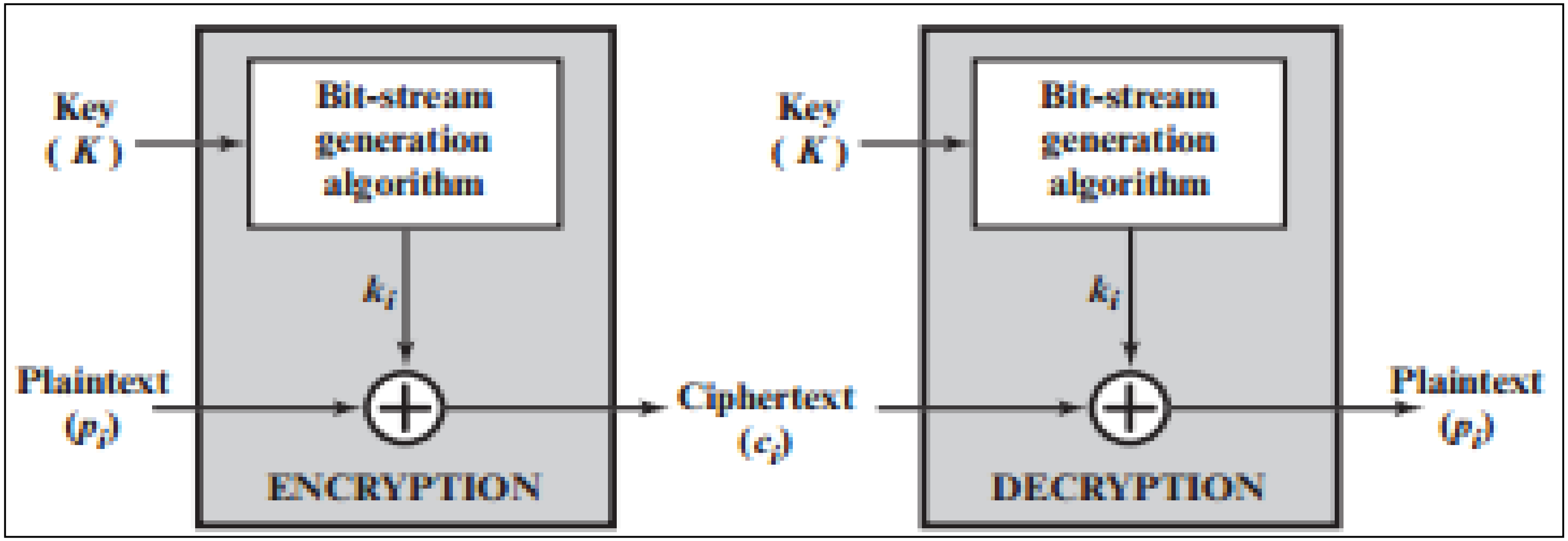
Uses a shared key



Problem of communicating a large message in secret reduced to communicating a small key in secret

Two Types


Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
$Y = E(K, X)$
$X = D(K, Y)$
Plaintext input
Encryption algorithm (e.g., AES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output


Key (K)
Bit-stream generation algorithm
k_i
Plaintext (p_i)
ENCRYPTION
Ciphertext (c_i)
Key (K)
Bit-stream generation algorithm
k_i
DECRYPTION
Plaintext (p_i)

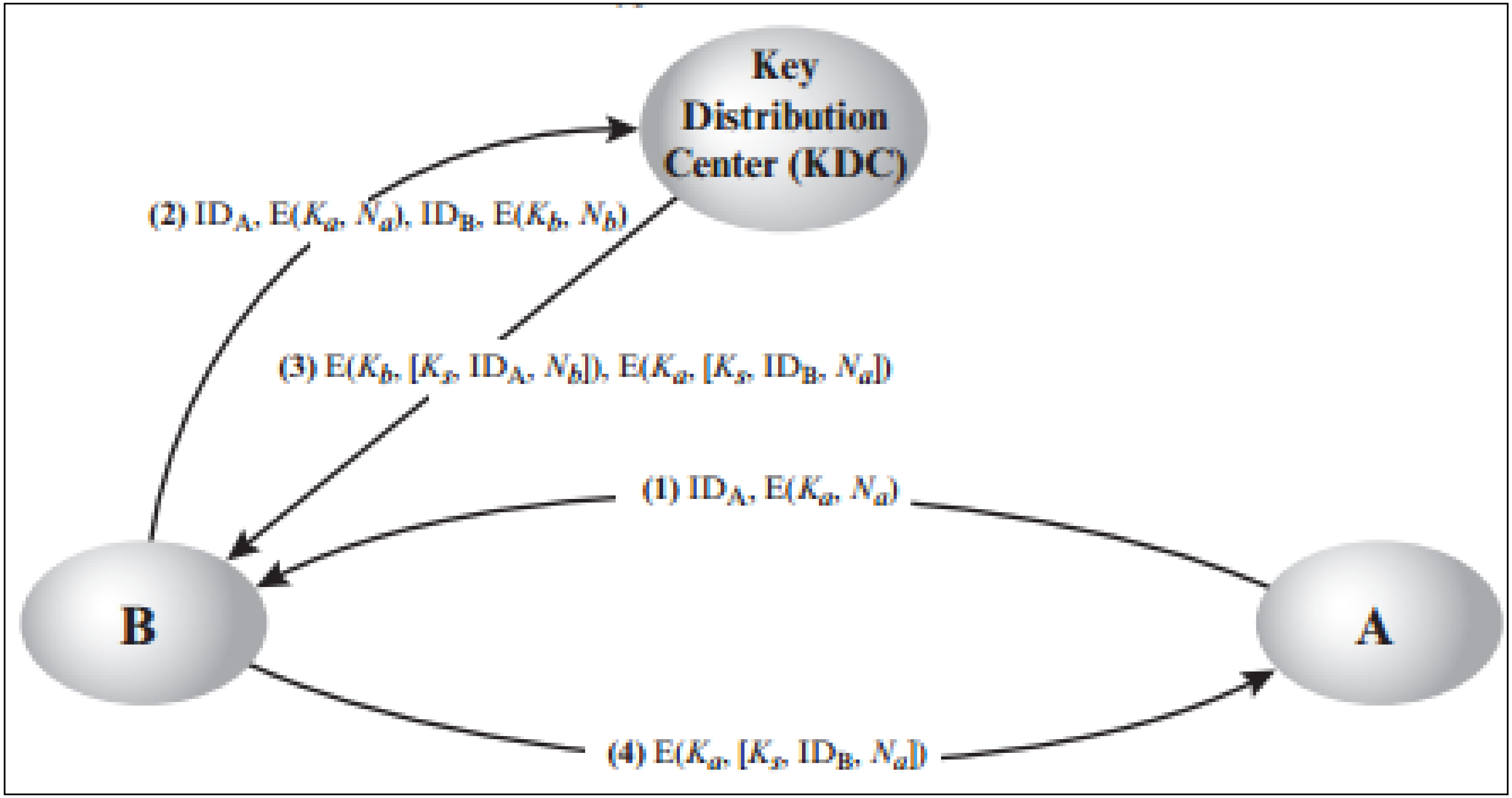
填空题 2分

设置

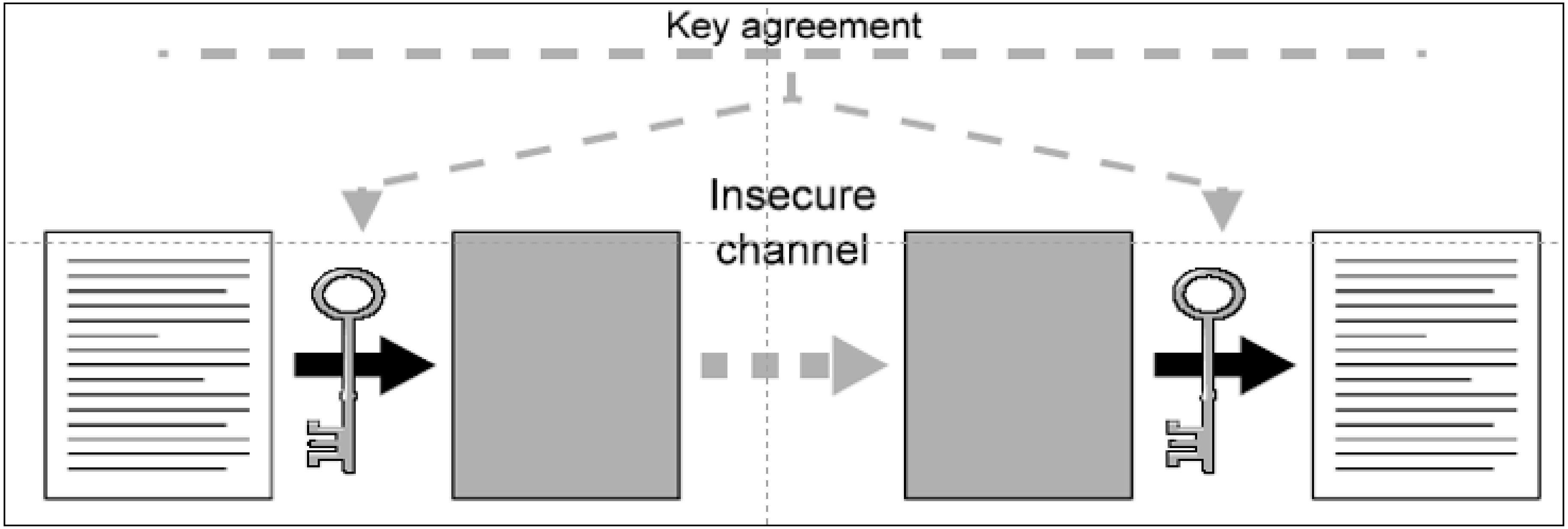
There are two kinds of symmetric ciphers according to the processing procedure and the key form, they are : [填空1] [填空2]

作答

Key Distribution


Key Distribution Center (KDC)
(2) ID_A, $E(K_a, N_a)$, ID_B, $E(K_b, N_b)$
(3) $E(K_b, [K_s, \mathrm{ID}_A, N_b])$, $E(K_a, [K_s, \mathrm{ID}_B, N_a])$
(1) ID_A, $E(K_a, N_a)$
B
A
(4) $E(K_a, [K_s, \mathrm{ID}_B, N_a])$

Key Agreement



主观题 2分

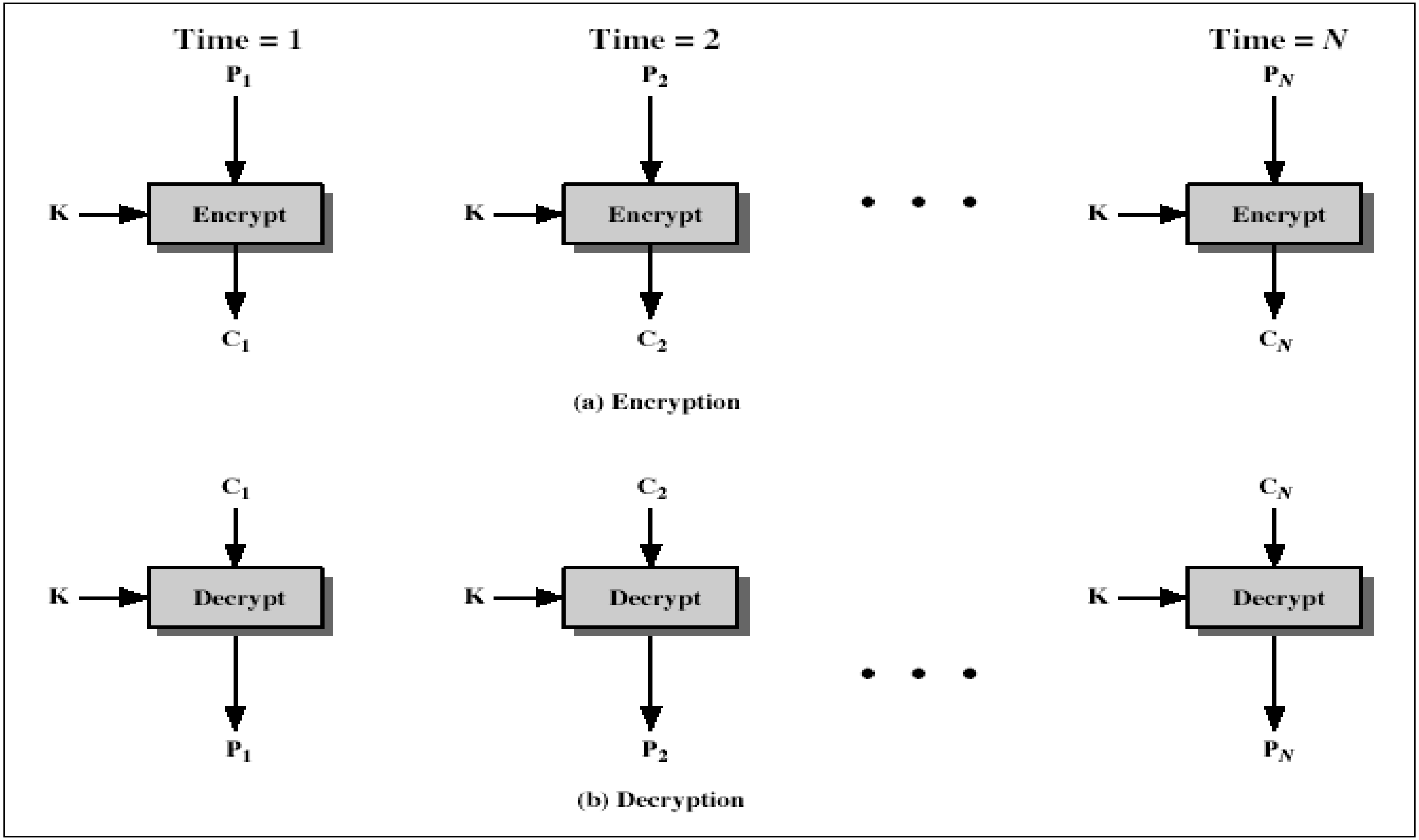
设置

For Block cipher algorithm, in order to process messages with any length, we should consider how to use it, that means the **operation mode**.

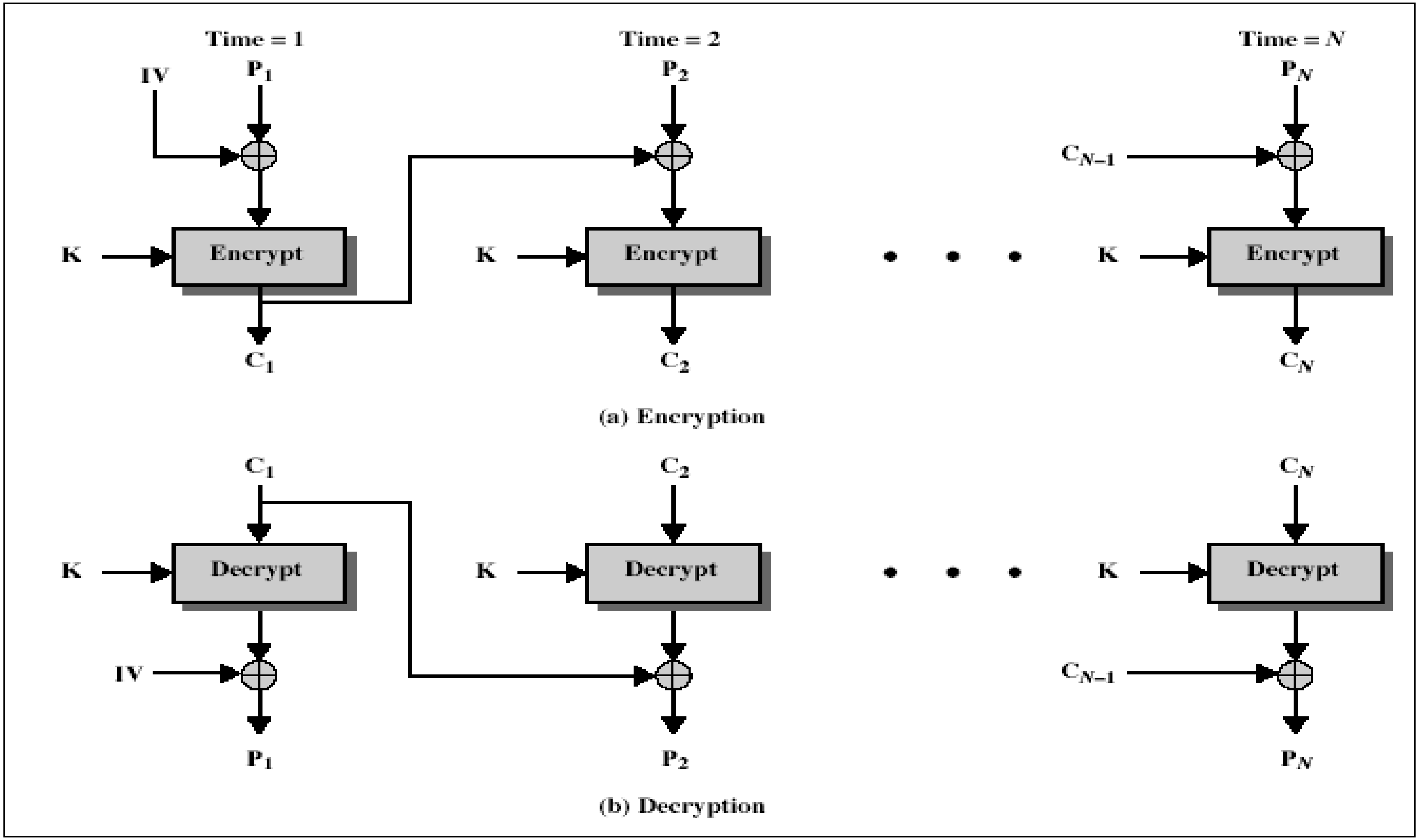
Could you tell some of them?

作答

Electronic Codebook Book (ECB)



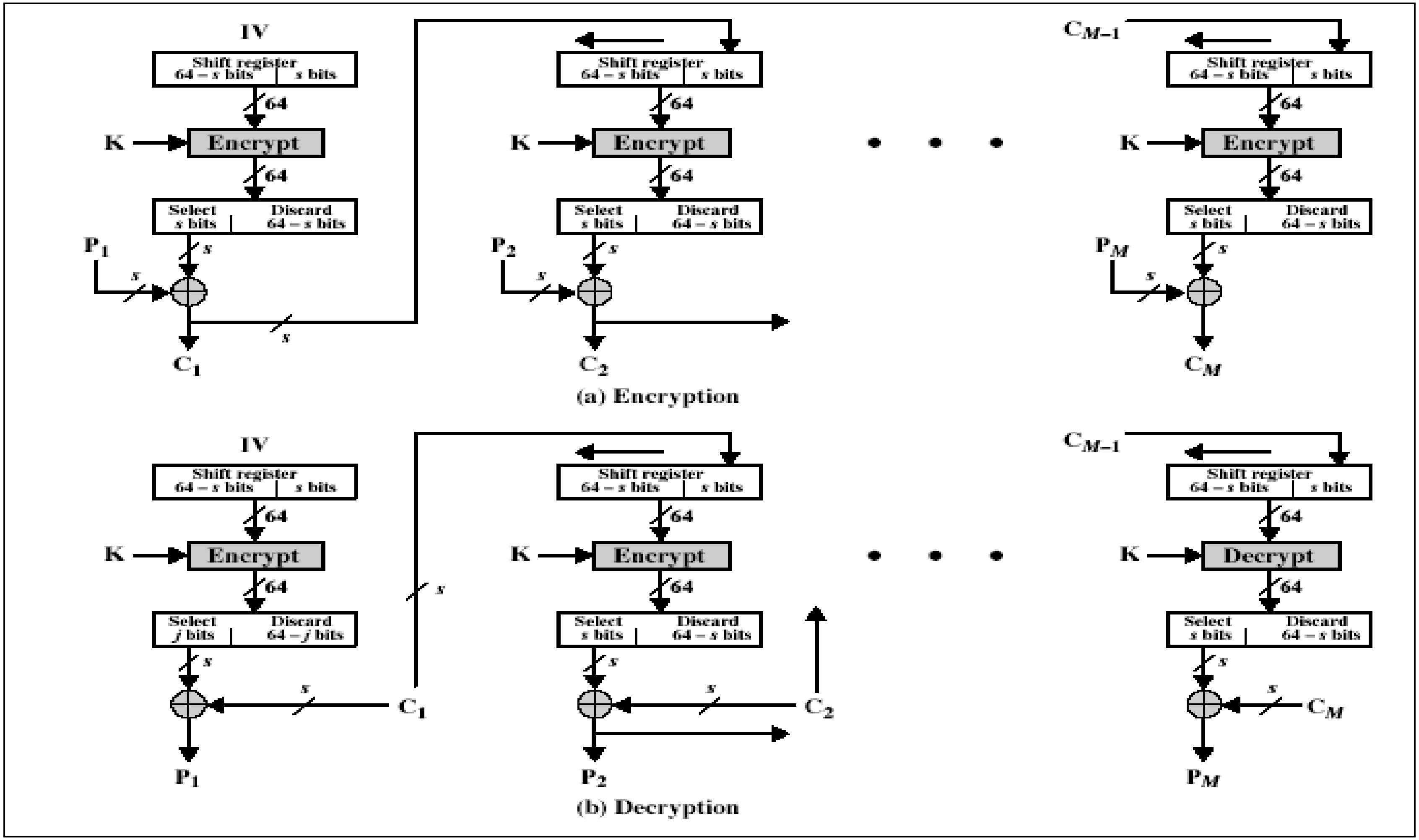
Cipher Block Chaining (CBC)



(a) Encryption

(b) Decryption

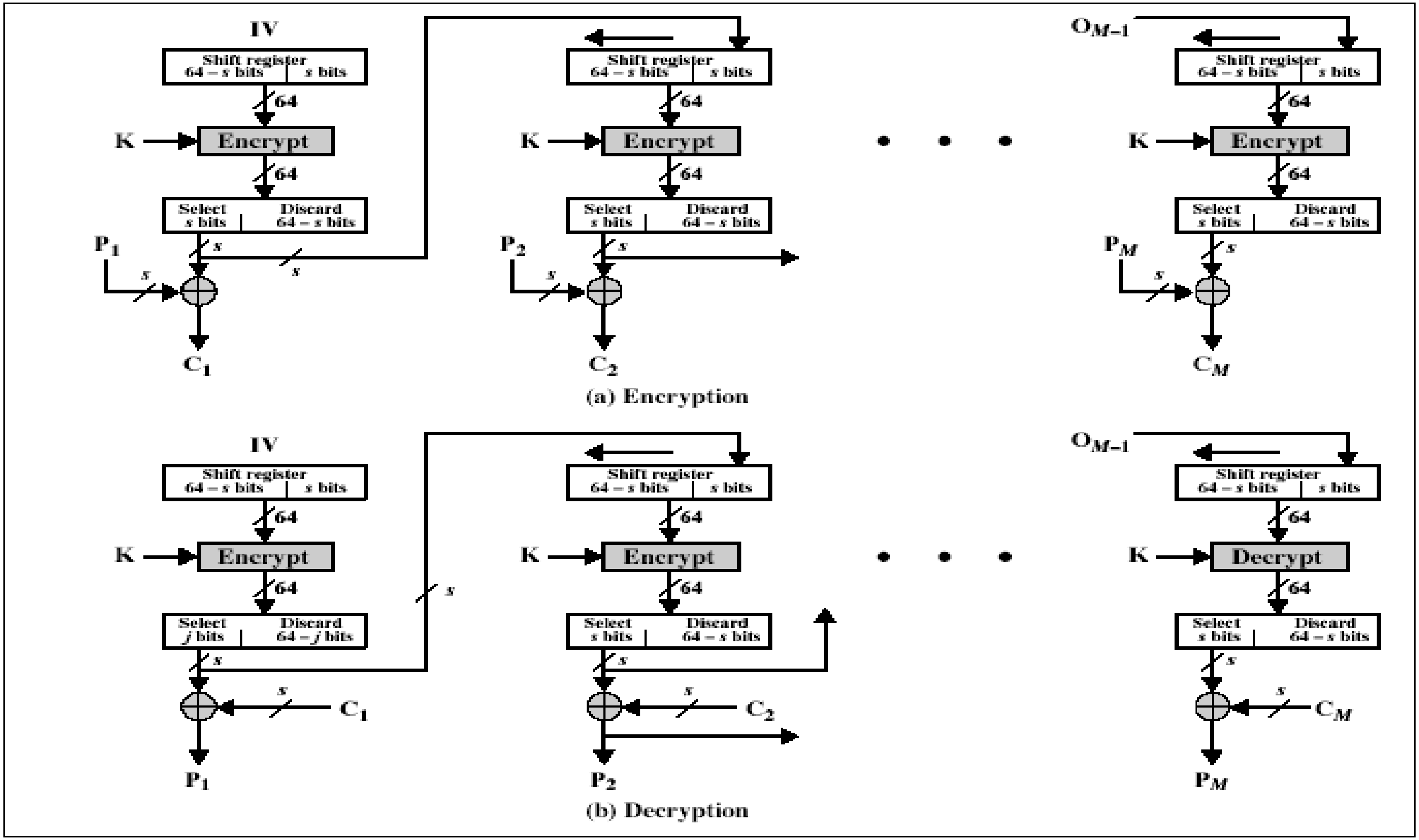
Cipher FeedBack (CFB)



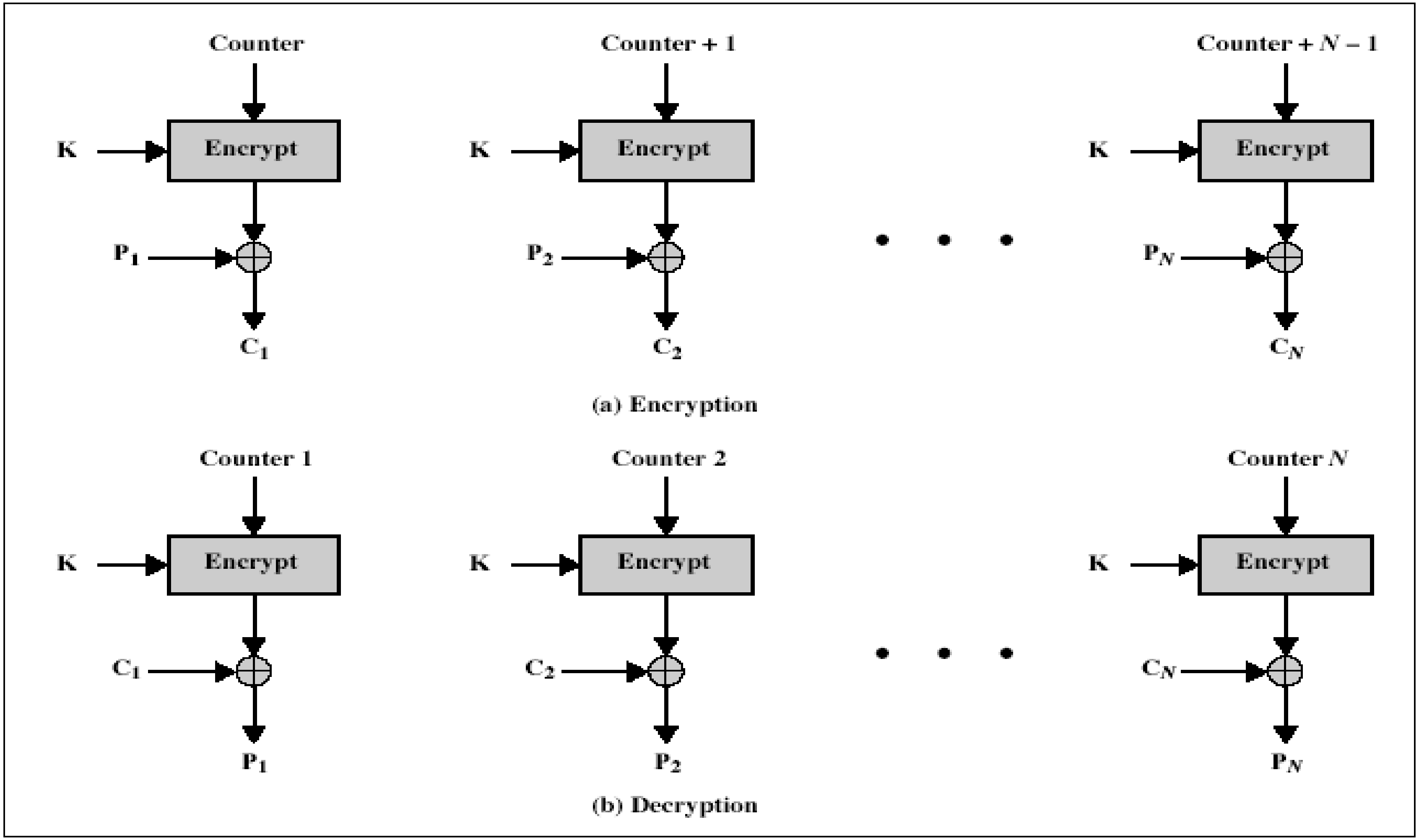
(a) Encryption

(b) Decryption

Output FeedBack (OFB)



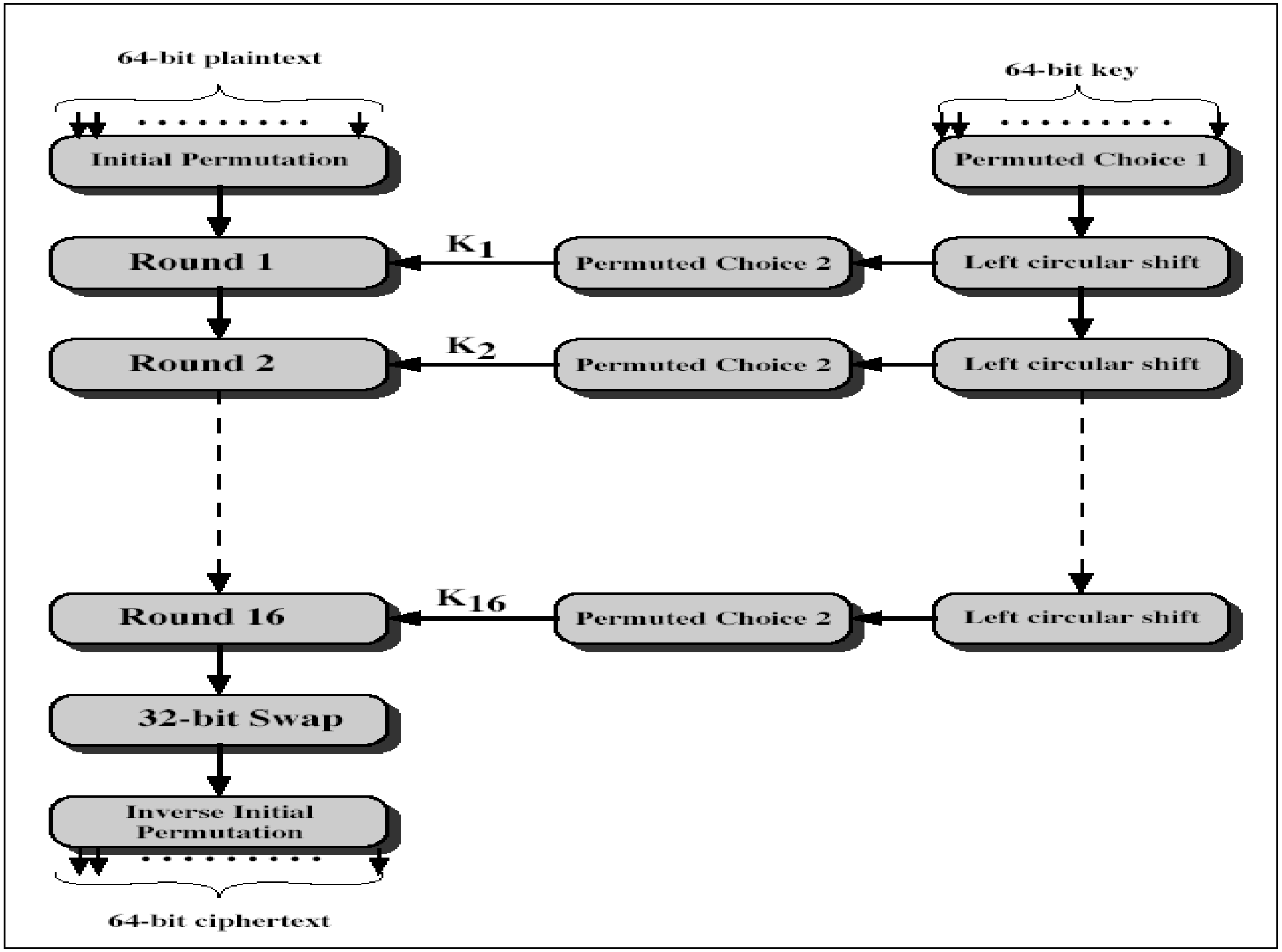
Counter (CTR)



(a) Encryption

(b) Decryption

Algorithm example: DES

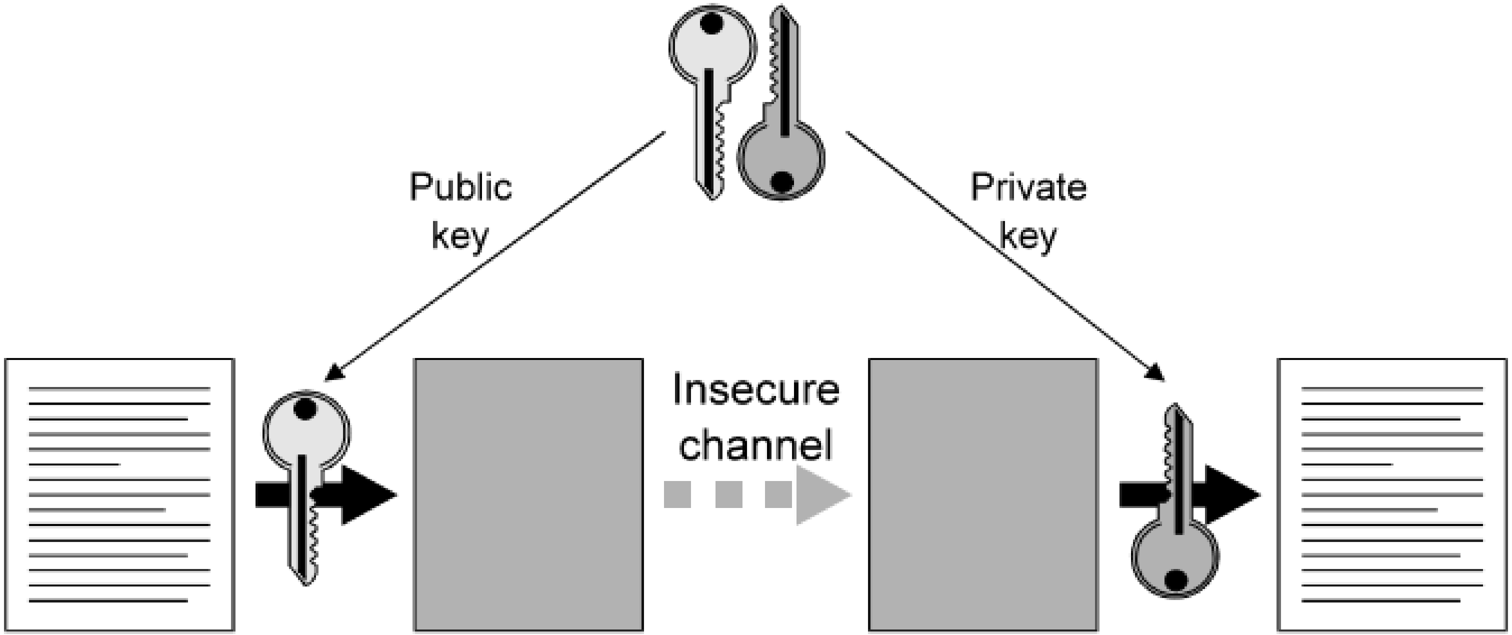


Other Algorithms

- AES
- IDEA RC2 RC4 RC5
- BLOW-FISH CAST128……

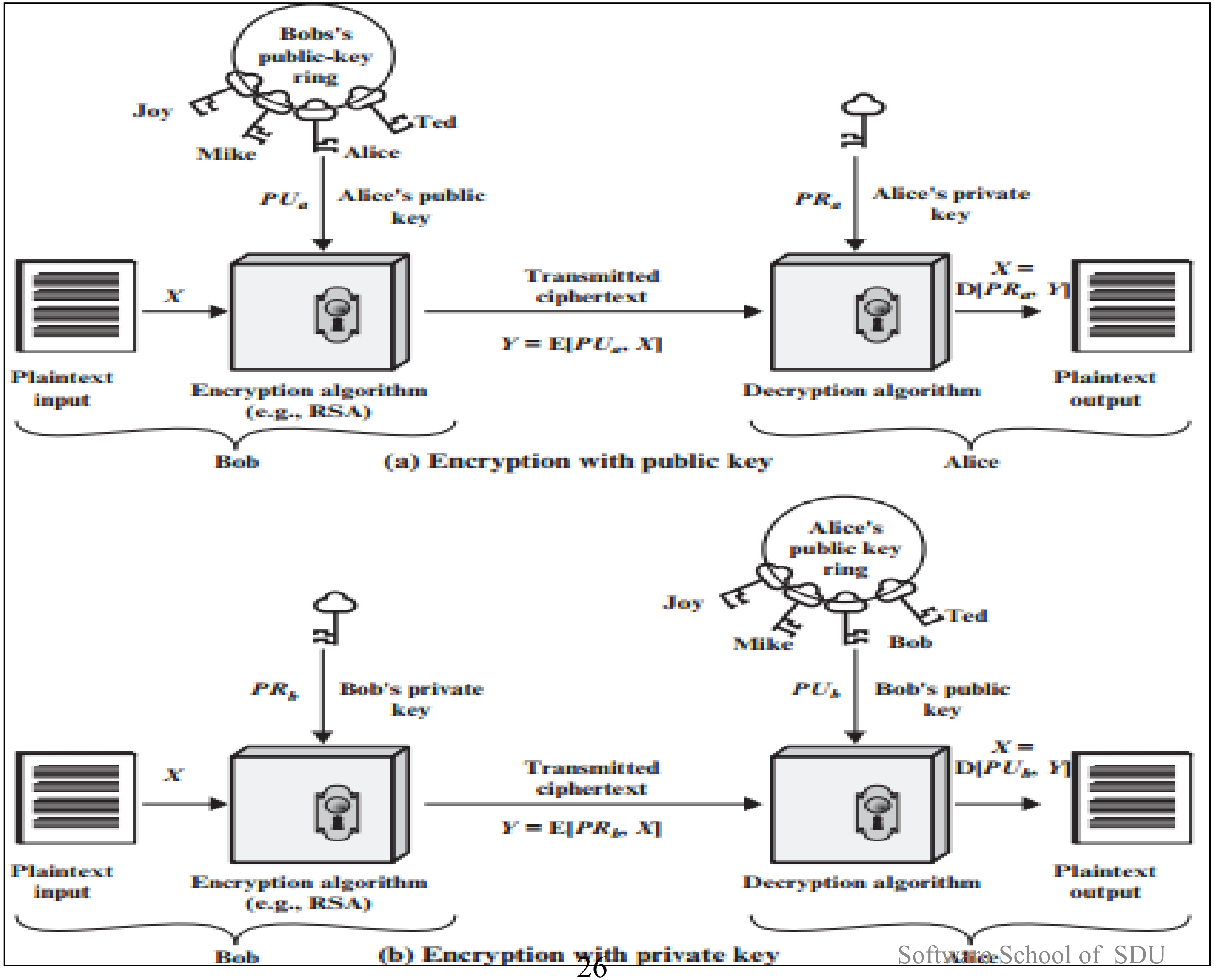
Public-key Encryption

Uses matched public/private key pairs



Anyone can encrypt with the public key, only one person can decrypt with the private key

Algorithms: Asymmetric Cipher



Public-Key Cryptography

- **public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - a **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- is **asymmetric** because
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

Why Public-Key Cryptography?

- developed to address two other key issues:
  - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
  - **digital signatures** – how to verify a message comes intact from the claimed sender
- public invention due to Whitfield Diffie & Martin Hellman at Stanford Uni in 1976
  - known earlier in classified community

Public-Key Characteristics

- Public-Key algorithms rely on two keys with the characteristics that it is:
  - computationally infeasible to find decryption key knowing only algorithm & encryption key
  - computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (in some schemes)

主观题 2分

设置

What can the Public key algorithms be used for?

作答

Public-Key Applications

- can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)
  - **key exchange** (of session keys)
- some algorithms are suitable for all uses, others are specific to one

Public-key Algorithms

RSA (Rivest-Shamir-Adleman), 1977

- Digital signatures and encryption in one algorithm
- Private key = sign and decrypt
- Public key = signature check and encrypt
- Patented, expires September 2000

DH (Diffie-Hellman), 1976

- Key exchange algorithm

Elgamal

- DH variant, one algorithm for encryption, one for signatures
- Non-patented alternative to RSA

Public-key Algorithms (ctd.)

DSA (Digital Signature Algorithm)

- Elgamal signature variant, designed by the NSA as the US government digital signature standard
- Intended for signatures only, but can be adapted for encryption

All have roughly the same strength:

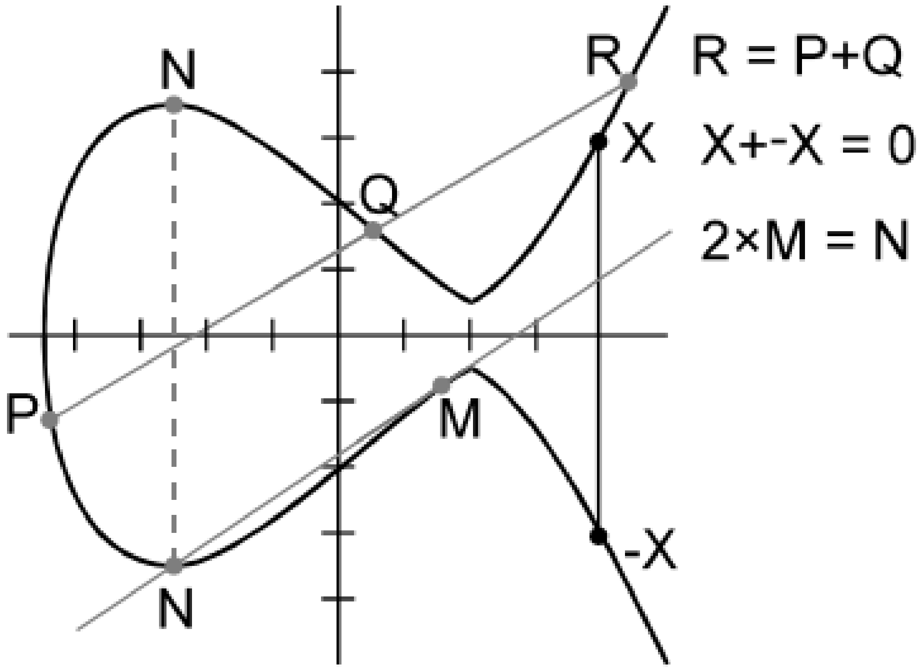
512 bit key is marginal
1024 bit key is recommended minimum size
2048 bit key is better for long-term security

Recommendation

- Anything suitable will do, RSA has wide acceptance but has patent problems in the US

Elliptic Curve Algorithms

Use mathematical trickery to speed up public-key operations



Elliptic Curve Algorithms (ctd)

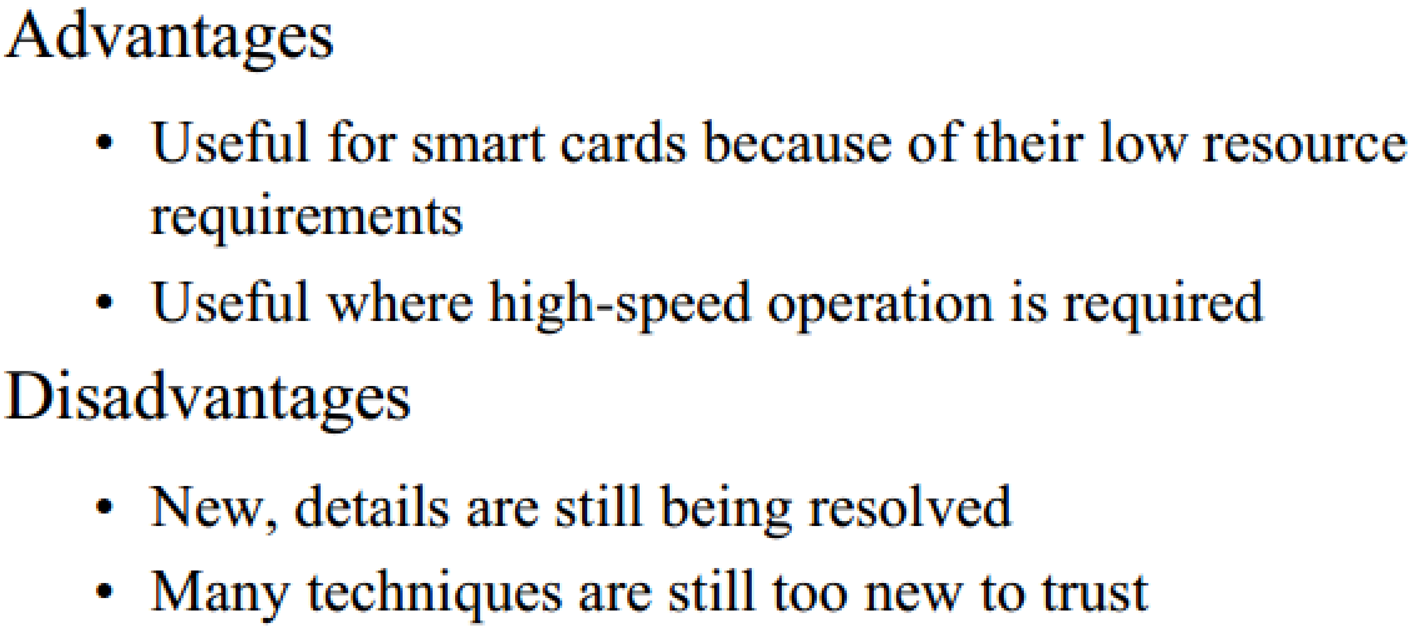
Now we can add, subtract, etc. So what?

- Calling it “addition” is arbitrary, we can just as easily call it multiplication
- We can now move (some) conventional PKC’s over to EC PKC's (DSA → ECDSA)

Now we have a funny way to do PKC’s. So what?

- Breaking PKC’s over elliptic curve groups is much harder than beaking conventional PKC’s
- We can use much shorter keys
- Encryption/decryption is faster since keys are shorter
- Key sizes are much smaller

Advantages/Disadvantages of ECC’s



Advantages

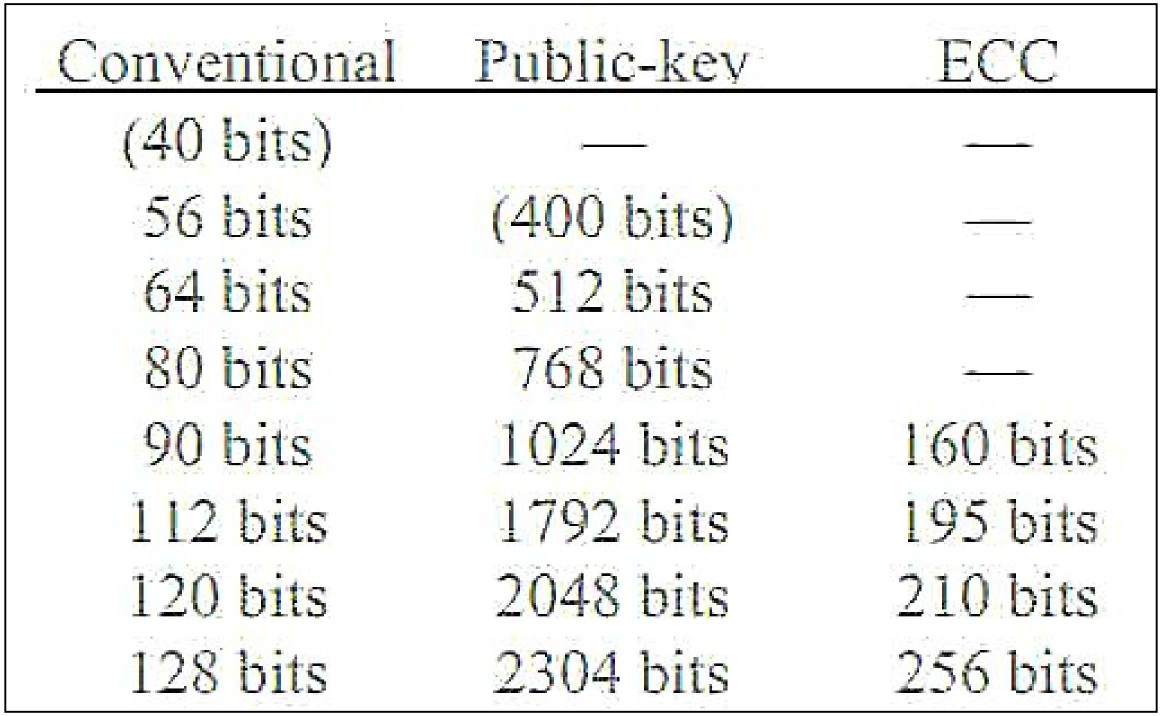
- Useful for smart cards because of their low resource requirements
- Useful where high-speed operation is required

Disadvantages

- New, details are still being resolved
- Many techniques are still too new to trust

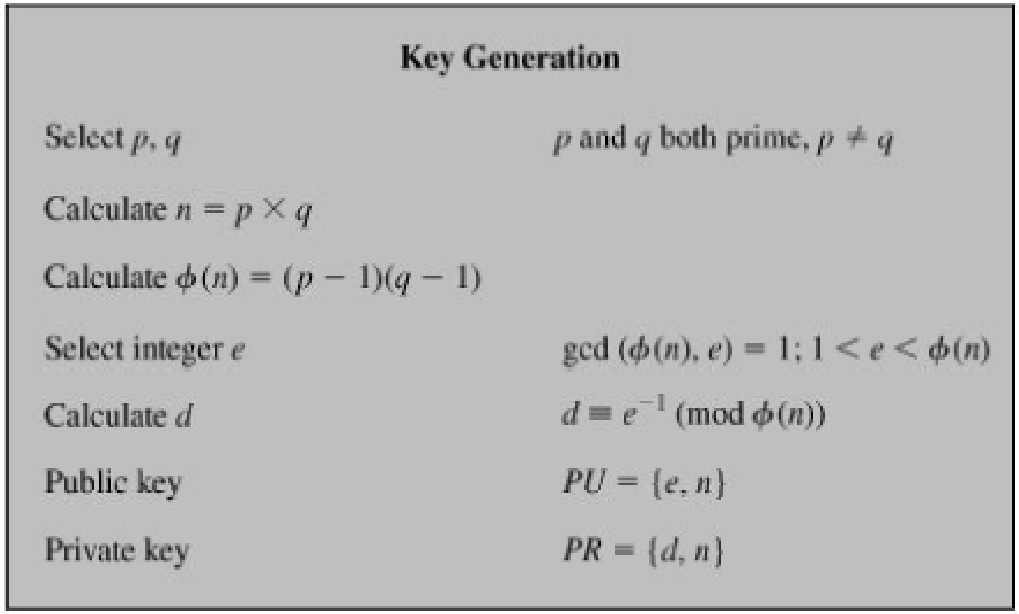
Key Sizes and Algorithms

Conventional vs public-key vs ECC key sizes



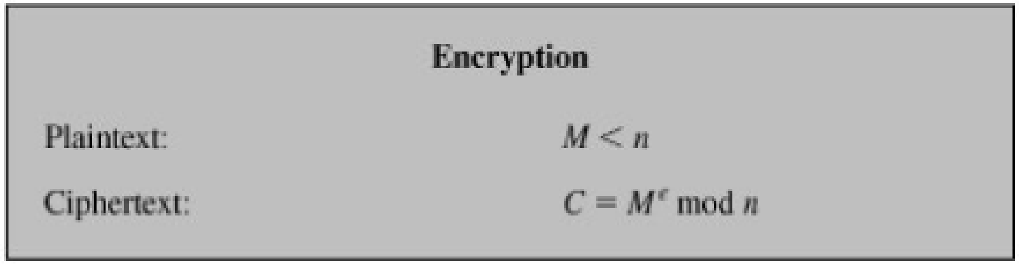
Conventional	Public-key	ECC
(40 bits)	—	—
56 bits	(400 bits)	—
64 bits	512 bits	—
80 bits	768 bits	—
90 bits	1024 bits	160 bits
112 bits	1792 bits	195 bits
120 bits	2048 bits	210 bits
128 bits	2304 bits	256 bits

Public Key Algorithm: RSA



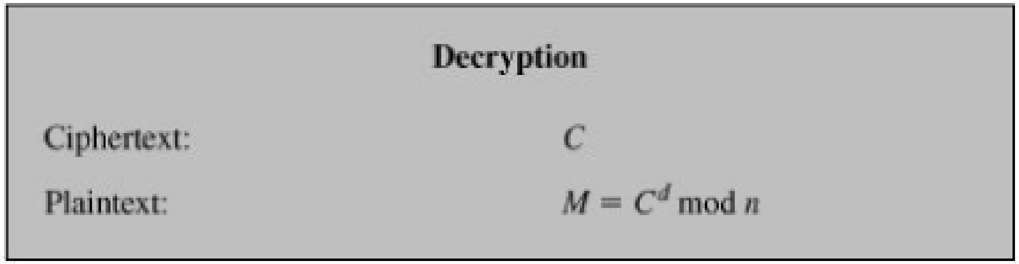
Key Generation

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} (\bmod \phi(n))$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$



Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$



Decryption

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

Decryption：

$$M = C^d \quad mod\ n$$

$$= (M^e)^d \quad mod\ n$$

$$= M^{ed} \quad mod\ n$$

$$= M^{k(p-1)(q-1)+1} \quad mod\ n$$

$$=M.\ M^{k(p-1)(q-1)} \quad mod\ n$$

$$= M\ mod\ n$$

(by Euler Theorem)

Symmetric Cipher VS. Asymmetric Cipher

- Adv. Of Symmetric Cipher
  - 速度快,处理量大，适用于对应用数据的直接加密。
  - 加密密钥长度相对较短,如40比特---128比特。
  - 除了加密，还可构造各种加密体制，如产生伪随机数,HASH函数等。
  - 历史悠久。
- Disadv. of Symmetric Cipher
  - 密钥在双方都要一致、保密，传递较难。
  - 大型网络中密钥量大，难以管理，一般需要TTP。
  - 密钥需要经常更换。

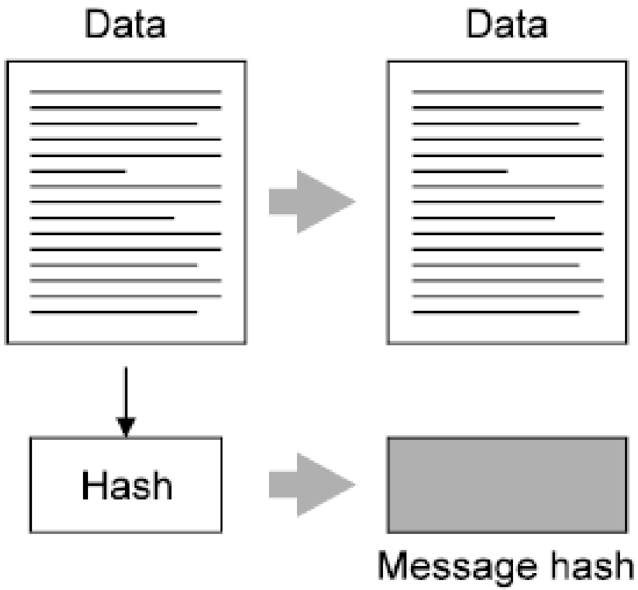
Symmetric Cipher VS. Asymmetric Cipher

- Adv. Of Asymmetric Cipher
  - 只有秘密钥保密，公开钥公开。
  - 网络上密钥管理只需要一个功能性TTP，而不是一个绝对安全的TTP，不需在线，可以离线。
  - 密钥生命周期相对较长.
  - 许多公钥方案可以产生数字签名机制。
  - 在大型网络上，所需的密钥相对较少。

- Disadv. Of Asymmetric Cipher
  - 速度慢，处理量少，适用于密钥交换。
  - 密钥长度相对较长。
  - 安全性没有得到理论证明。
  - 历史较短。

Hash Functions

Creates a unique "fingerprint" for a message



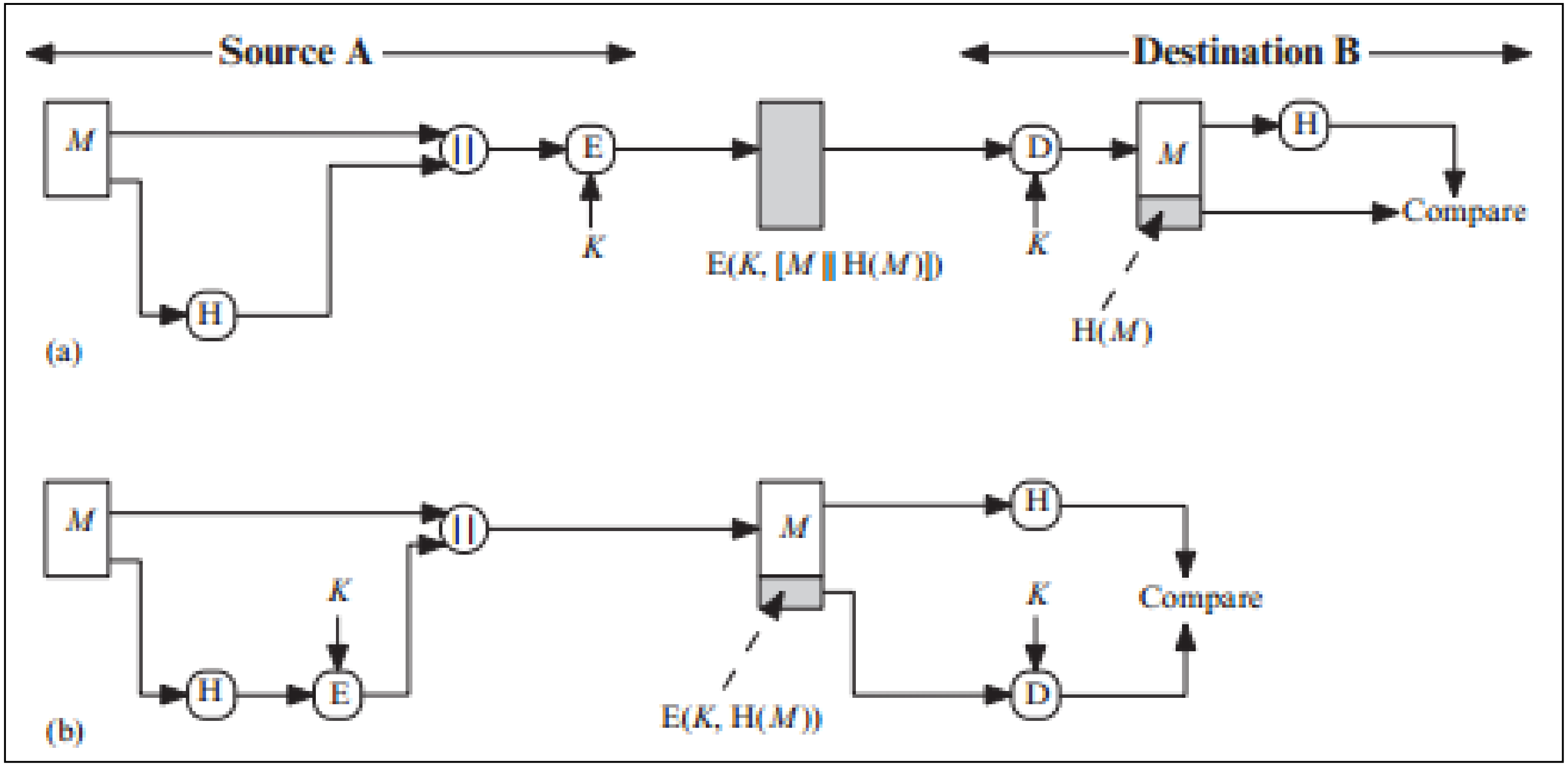
Anyone can alter the data and calculate a new hash value

- Hash has to be protected in some way

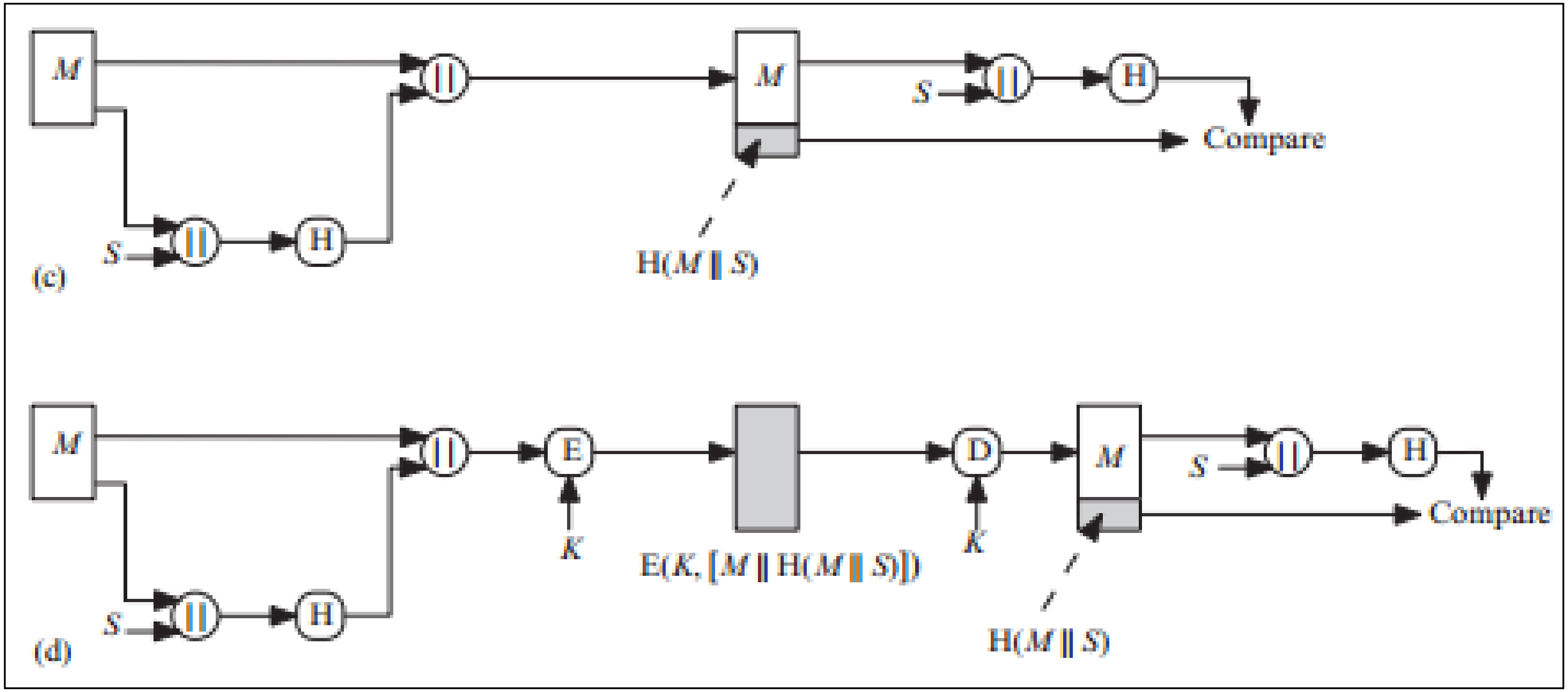
Hash Functions

1. can be applied to any sized message M
2. produces fixed-length output h
3. is easy to compute h=H(M) for any message M
4. given h is infeasible to find x s.t. H(x)=h
   - one-way property
5. given x is infeasible to find y s.t. H(y)=H(x)
   - weak collision resistance
6. is infeasible to find any x,y s.t. H(y)=H(x)
   - strong collision resistance

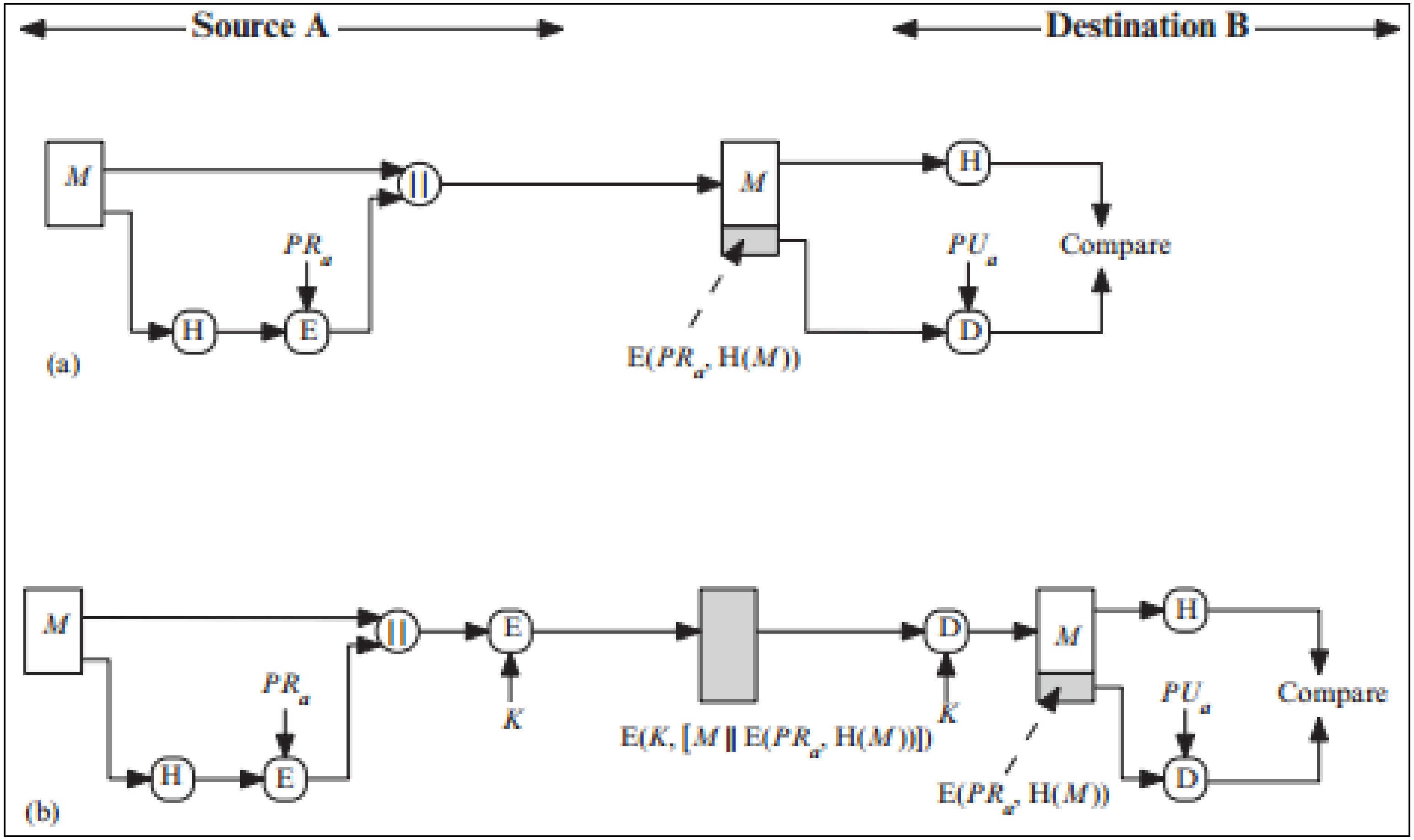
Providing Message Authentication



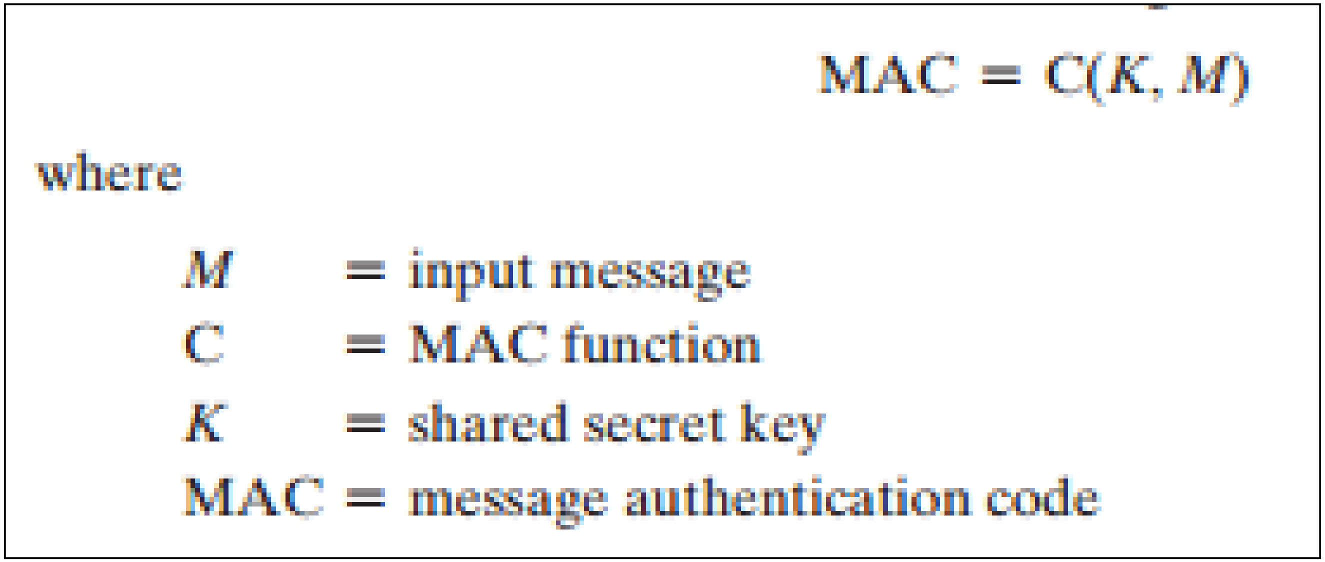
Providing Message Authentication



Providing Help for Message Signature



Message Authentication Code(MAC)



$$\mathrm{MAC} = \mathrm{C}(K, M)$$

where

M = input message

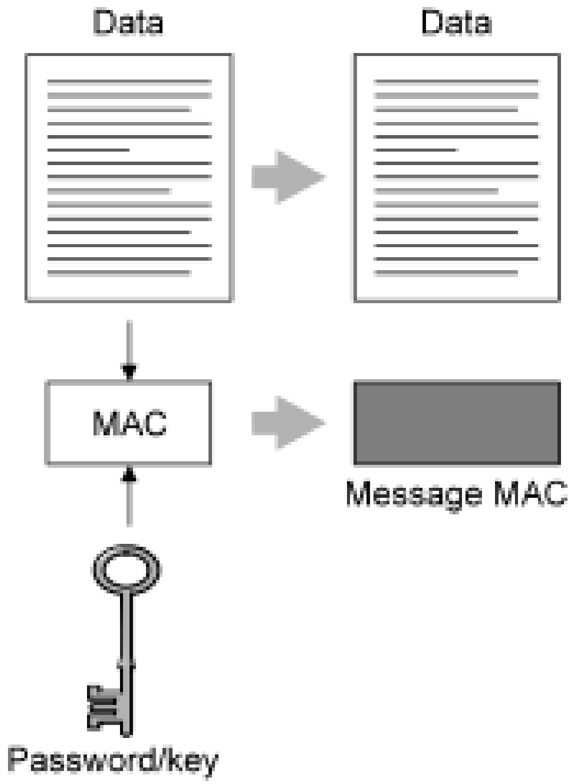
C = MAC function

K = shared secret key

MAC = message authentication code

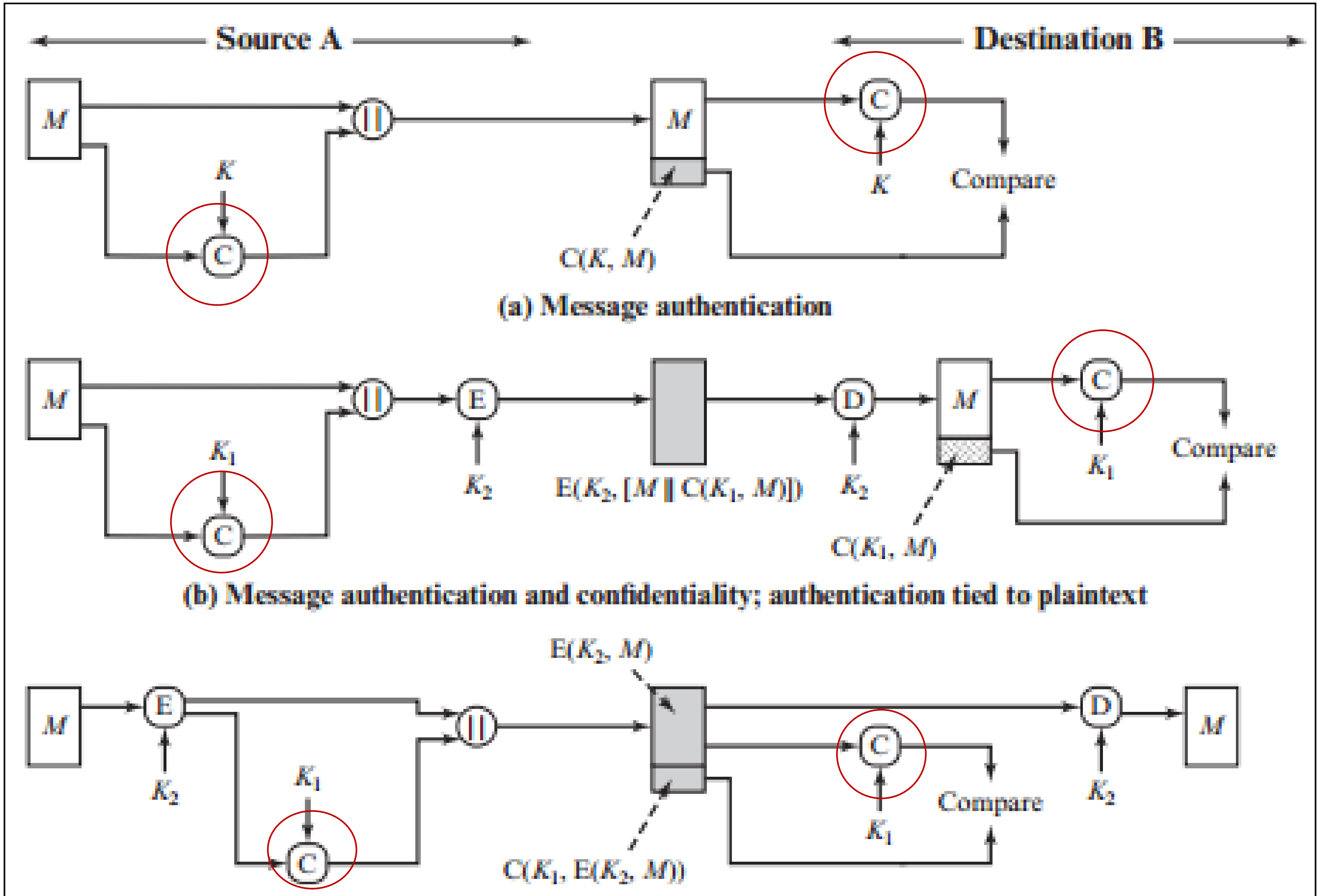
MAC’s

Message Authentication Code, adds a password/key to a hash



Only the password holder(s) can generate the MAC

MAC



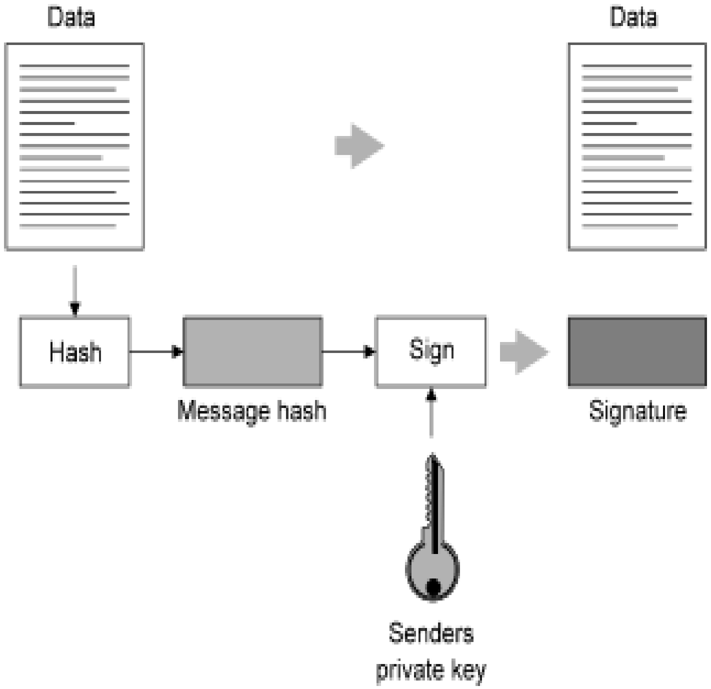
(a) Message authentication

(b) Message authentication and confidentiality; authentication tied to plaintext

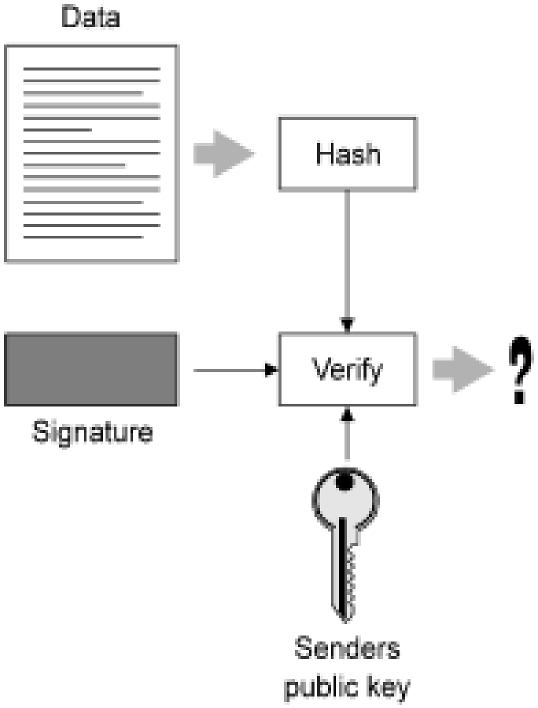
(c) Message authentication and confidentiality; authentication tied to ciphertext

Digital Signatures

Combines a hash with a digital signature algorithm

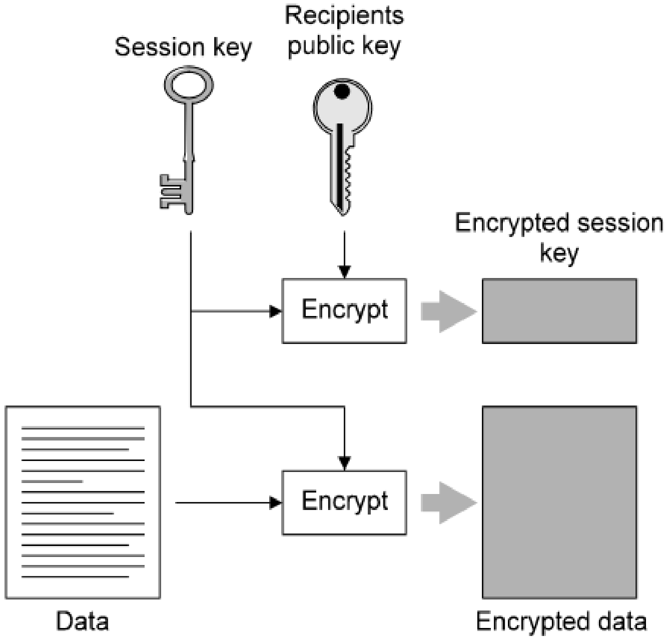


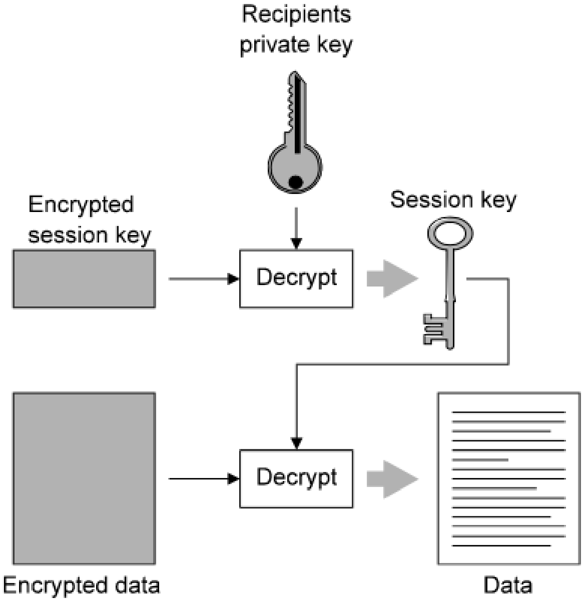
Signature checking:



Message/Data Encryption

Combines conventional and public-key encryption





Public-key encryption provides a secure channel to exchange conventional encryption keys

Key Management

- Key management is the hardest part of cryptography
- Two classes of keys
  - **Short-term session keys** (sometimes called ephemeral keys)
    - Generated automatically and invisibly
    - Used for one message or session and discarded
  - **Long-term keys**
    - Generated explicitly by the user
- Long-term keys are used for two purposes
  - Authentication (including access control, integrity, and nonrepudiation)
  - Confidentiality (encryption)
    - Establish session keys
    - Protect stored data

Key Management Problems

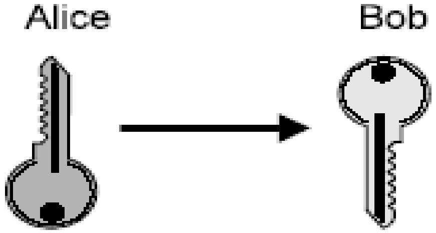
- Distributing keys
  - Obtaining someone else’s public key
  - Distributing your own public key
- Establishing a shared key with another party
  - Confidentiality: Is it really known only to the other party?
  - Authentication: Is it really shared with the intended party?
- Key storage
  - Secure storage of keys
- Revocation
  - Revoking published keys
  - Determining whether a published key is still valid

Key Lifetimes and Key Compromise

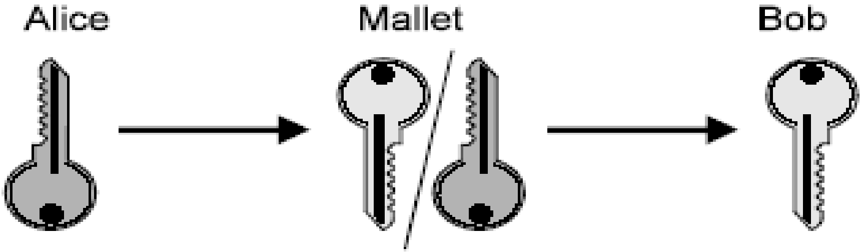
- Authentication keys
  - Public keys may have an extremely long lifetime (decades)
  - Private keys/conventional keys have shorter lifetimes (a year or two)
- Confidentiality keys
  - Should have as short a lifetime as possible
- If the key is compromised
  - Revoke the key
- Effects of compromise
  - Authentication: Signed documents are rendered invalid unless timestamped
  - Confidentiality: All data encrypted with it is compromised

Key Distribution Problem

Alice retains the private key and sends the public key to Bob



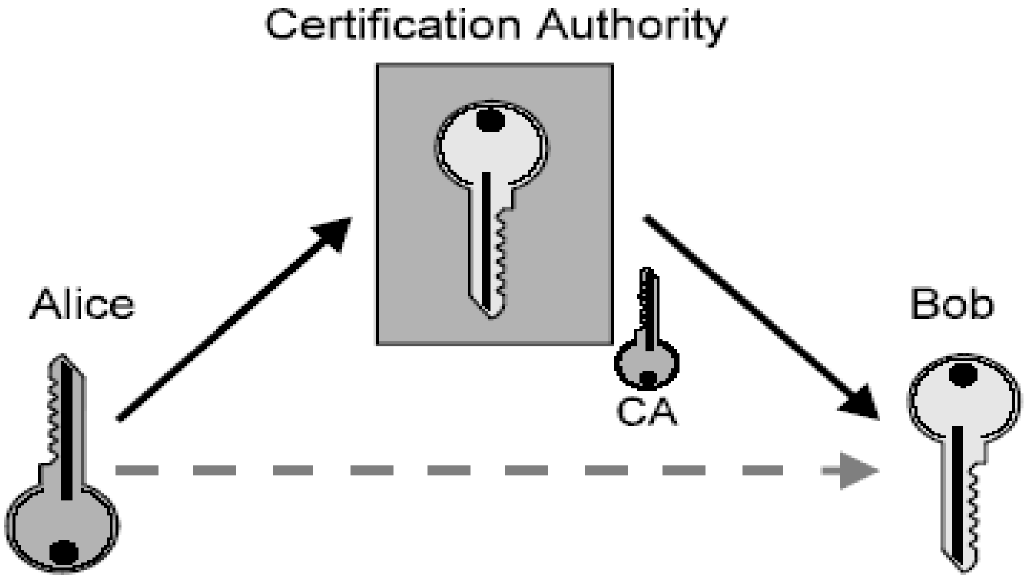
Mallet intercepts the key and substitutes his own key



Mallet can decrypt all traffic and generate fake signed message

Key Distribution Problem(ctd)

A certification authority (CA) solves this problem



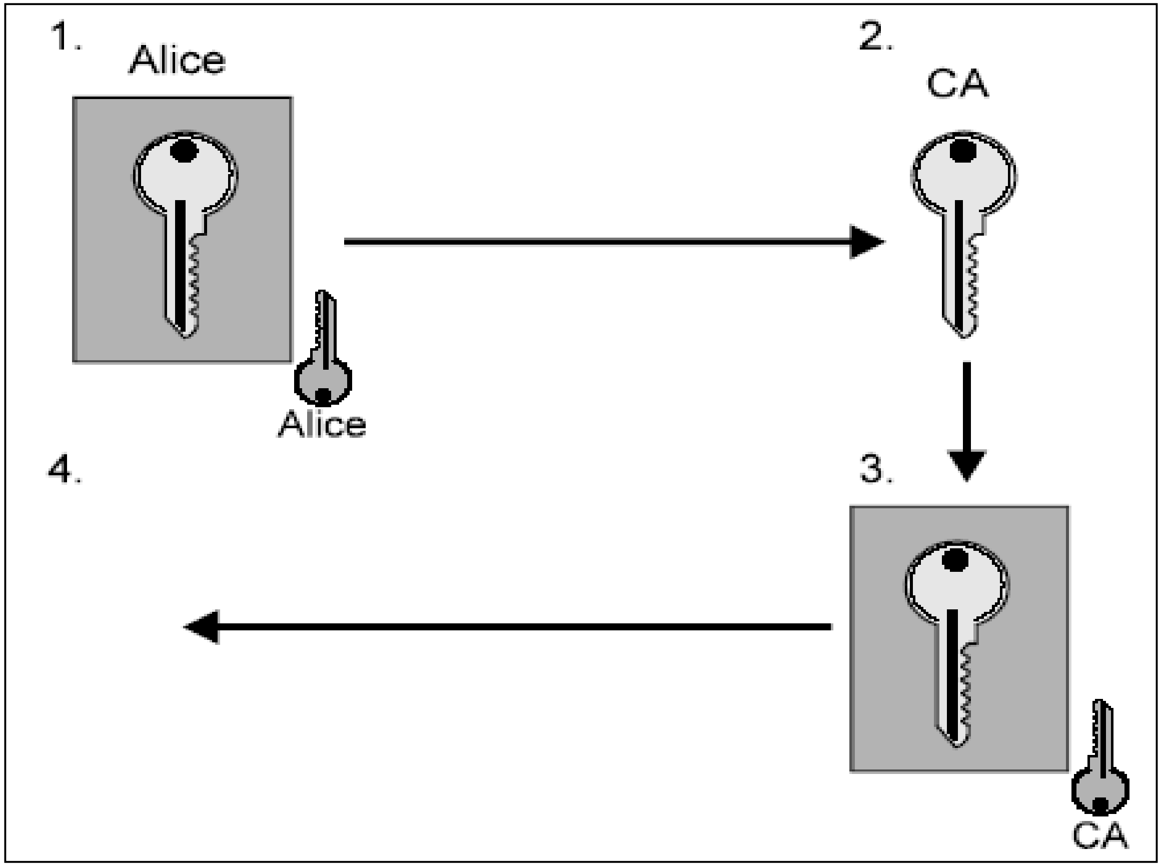
CA signs Alice's key to guarantee its authenticity to Bob

- Mallet can't substitute his key since the CA won't sign it

Certification Authorities

- A certification authority (CA) guarantees the connection between a key and an end entity BY generating the digital certificate for users
- An end entity is
  - A person
  - A role (“Director of marketing”)
  - An organisation
  - A pseudonym
  - A piece of hardware or software
  - An account (bank or credit card)
- Some CA’s only allow a subset of these types

Obtaining a Certificate



Certificate History

Certificates were originally intended to protect access to the X.500 directory

- All-encompassing, global directory run by monopoly telco’s

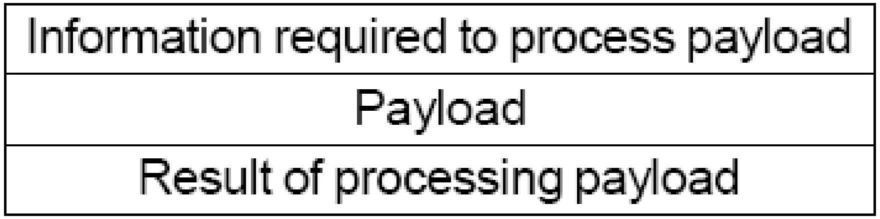
Concerns about misuse of the directory

- Companies don’t like making their internal structure public
  - Directory for corporate headhunters
- Privacy concerns
  - Directory of single women
  - Directory of teenage children

X.509 certificates were developed as part of the directory access control mechanisms

Data Formats

One obviously-correct format for secured content

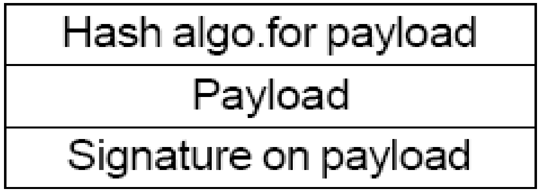


Information required to process payload
Payload
Result of processing payload

- Allows straightforward one-pass processing for encapsulation and decapsulation

Data Formats (ctd)

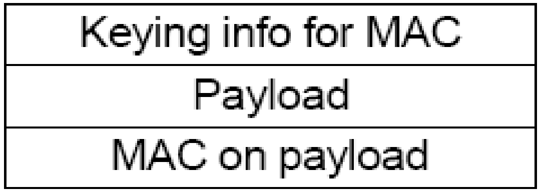
Signed data


Hash algo.for payload

Payload

Signature on payload

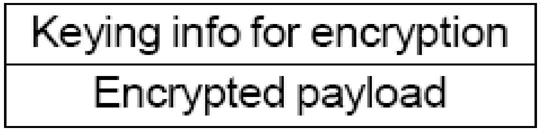
MACd data


Keying info for MAC

Payload

MAC on payload

Encrypted data


Keying info for encryption

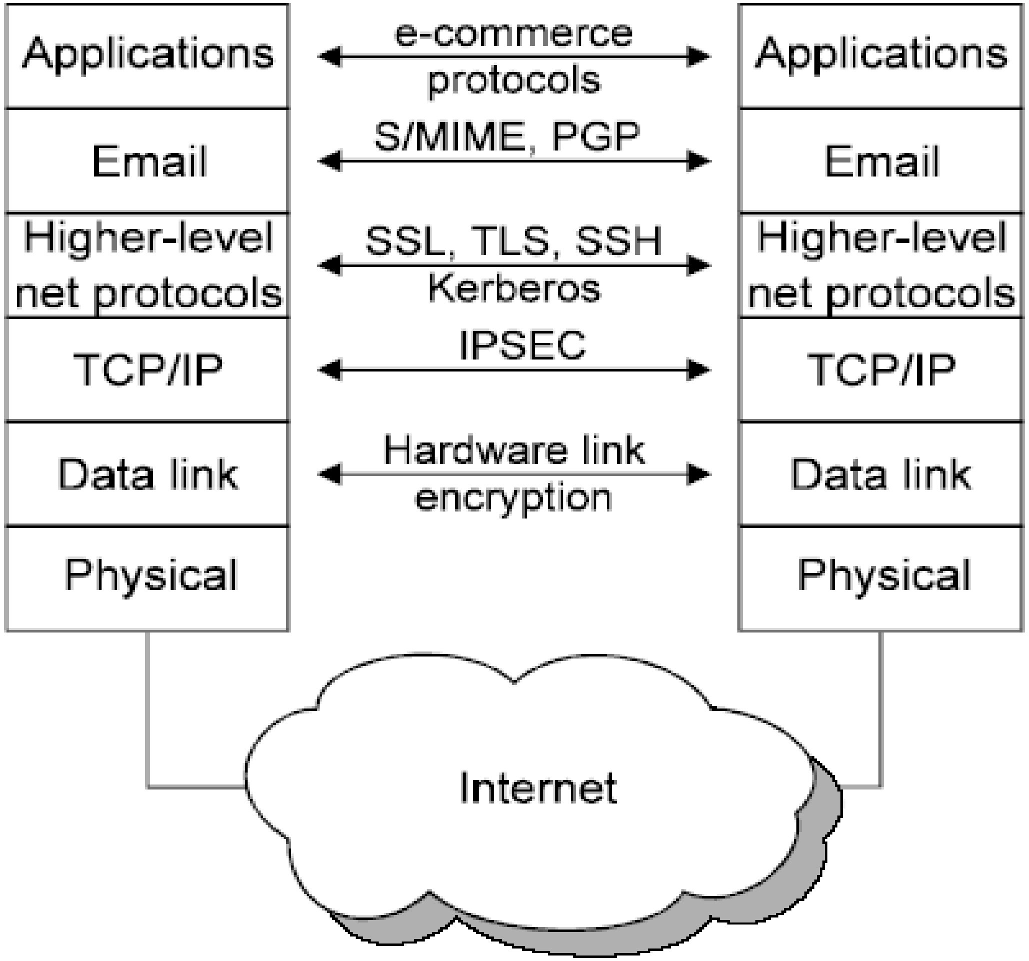
Encrypted payload

Keying info = password derivation info, public-key-encrypted content-encryption key, …

This single obvious format is why PGP and S/MIME, SSL and SSH differ mostly in their bit-bagging formats

- Doesn’t prevent standards groups from coming up with different (broken) versions

Security Protocol: Layers

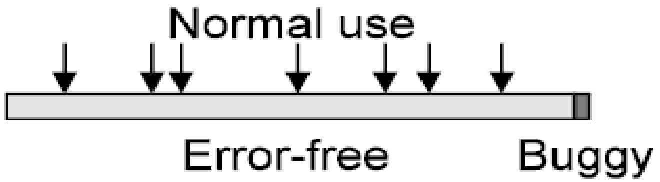


The further down you go, the more transparent it is
The further up you go, the easier it is to deploy

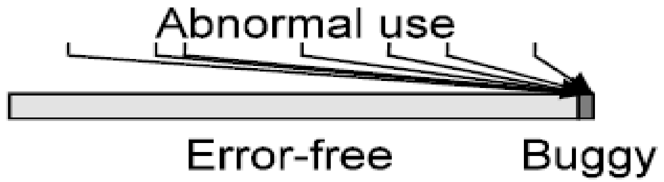
Security is Harder than it Looks

All software has bugs

Under normal usage conditions, a 99.99% bug-free program will rarely cause problems



A 99.99% security-bug-free program can be exploited by ensuring the 0.01% instance is always encountered



This converts the 0.01% failure to 100% failure

单选题 1分

设置

The key used to protect application data is called

A Master key

[✓ B Session key]

提交

填空题 1分

设置

In the pure symmetric encryption environment, we often call the key management authority as [填空1]

作答

填空题 1分

设置

In the Asymmetric encryption environment, we often call the key management authority as [填空1]

作答

主观题 1分

设置

1. 使用对称密钥，我们主要确保它的什么属性？

2. 使用公钥密码机制，我们主要保证公开密钥的什么属性？

作答

填空题 1分

设置

CA is used to issue [填空1] for users to ensure the combinations of identities and their public keys.

作答

PKI (Public Key Infrastructure)

- Internet安全系统解决方案, PKI采用数字证书机制管理公钥，通过第三方可信机构CA，把用户的公开钥和身份信息进行有效捆绑，在Internet上有效验证用户的身份。
- 借助数字证书机制分发密钥，通过各种密码学算法和对要传输的信息进行安全处理，通过构造密码安全协议保证信息传输的机密性、完整性、真实性、不可否认性，实现身份认证，以及访问控制，从而全面保证数据存储安全和传输安全安全。

PKI 发展

- 1976，RSA
- 20世纪80年代，美国学者提出PKI概念
- 美国在1996年成立了联邦PKI指导委员会
- 1999年，国际PKI论坛成立
- 2000年4月，美国国防部宣布要采用PKI安全倡议方案。签署《全球及全国商业电子签名法》
- 2000年10月成立欧洲桥CA指导委员会，2001粘月成立欧洲桥CA
- 2001年6月13日，在亚洲和大洋洲推动PKI进程的国际组织 “亚洲PKI论坛” 宣告成立，宗旨是在亚洲地区推动PKI标准化，为实现全球范围的电子商务奠定基础。

PKI 发展

- 我国PKI技术起步于1998年，独立实体运营的SHECA成立。2001年PKI技术列为“十五”863计划重大项目，2002年成立中国PKI论坛。
- 1999年10月7日，《商用密码管理条例》正式由国务院颁布施行，对商用密码产品的科研、生产、销售和使用实施管理。
- 2002年国家商用密码办公室成立，专门负责全国商用密码管理工作，包括商用密码技术与标准化、产品服务和进出口审批、密码应用、监督管理、检测认证等。
- 2005年，发布实施《**中华人民共和国电子签名法**》（2015年修订）
- 2011年成立了密码行业标准化技术委员会。
- 截止2002年，全国成立全国性和区域性行业性CA机构60余家。
- 2002年成立全国信息安全标准化技术委员会，成立“PKI/PMI（WG4）”工作组

PKI 发展

- 2000年至今，中国PKI技术和应用大发展
  - 技术体系标准化　商密SM系列标准
  - 专业公司蓬勃发展，有2000多款商用密码通用产品，上千家从业单位
  - 产品适应 传统因特网、移动终端、无线网络、云计算、物联网等环境
  - 2017年，国家密码管理局起草了《中华人民共和国密码法（草案征求意见稿）》
- 国际标准 PKIX　系统化 标准化
  - PKIX系列标准(Public Key Infrastructure on X.509)是由因特网网络工程技术小组(Internet Engineering Task Force)的PKI小组制定，PKIX标准化是建立互操作的基础。标准主要定义基于X.509的PKI框架模型，并以RFC形式发布。