Lecture 18:
Security

- ***Operating System and networks***

- Behzad Bordbar

Overview

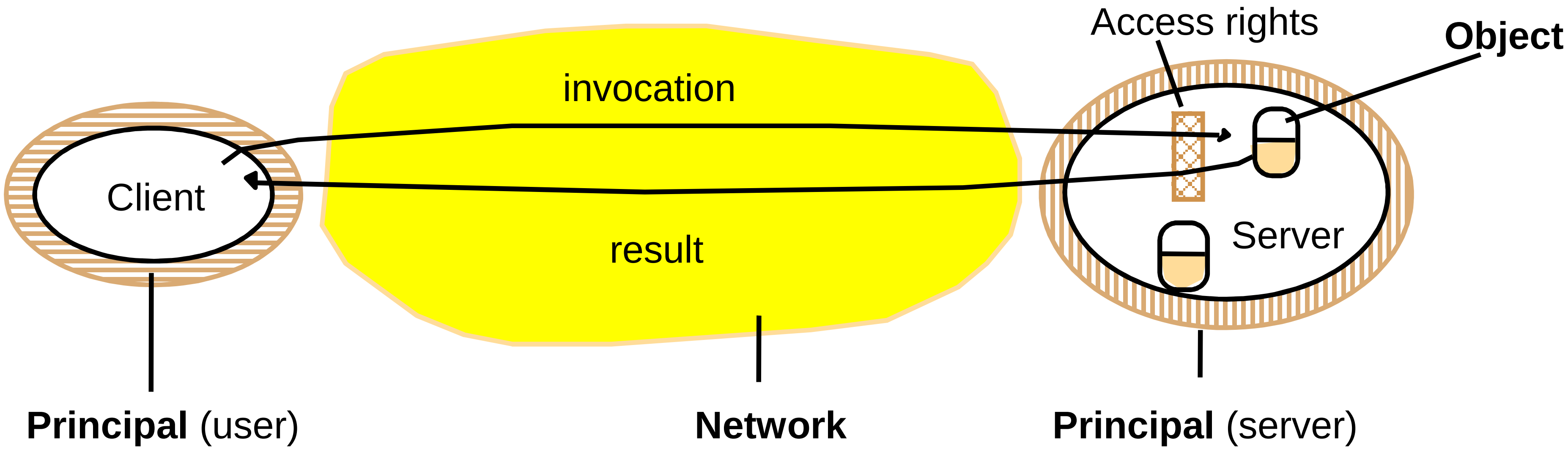
- What is security?
  - policies and mechanisms
  - threats and attacks
- Security of electronic transactions
  - secure channels
  - authentication and cryptography
- Security techniques
  - access control
  - firewalls
  - cryptographic algorithms

Security

- Definition
  - set of measures to guarantee the **privacy, integrity and availability** of resources:
    - ❑objects, databases, servers, processes, channels, etc
  - involves protection of objects and securing processes and communication channels
- Security policies
  - specify who is authorised to access resources (e.g. file ownership)
- Security mechanisms
  - enforce security policy (e.g. file access control)

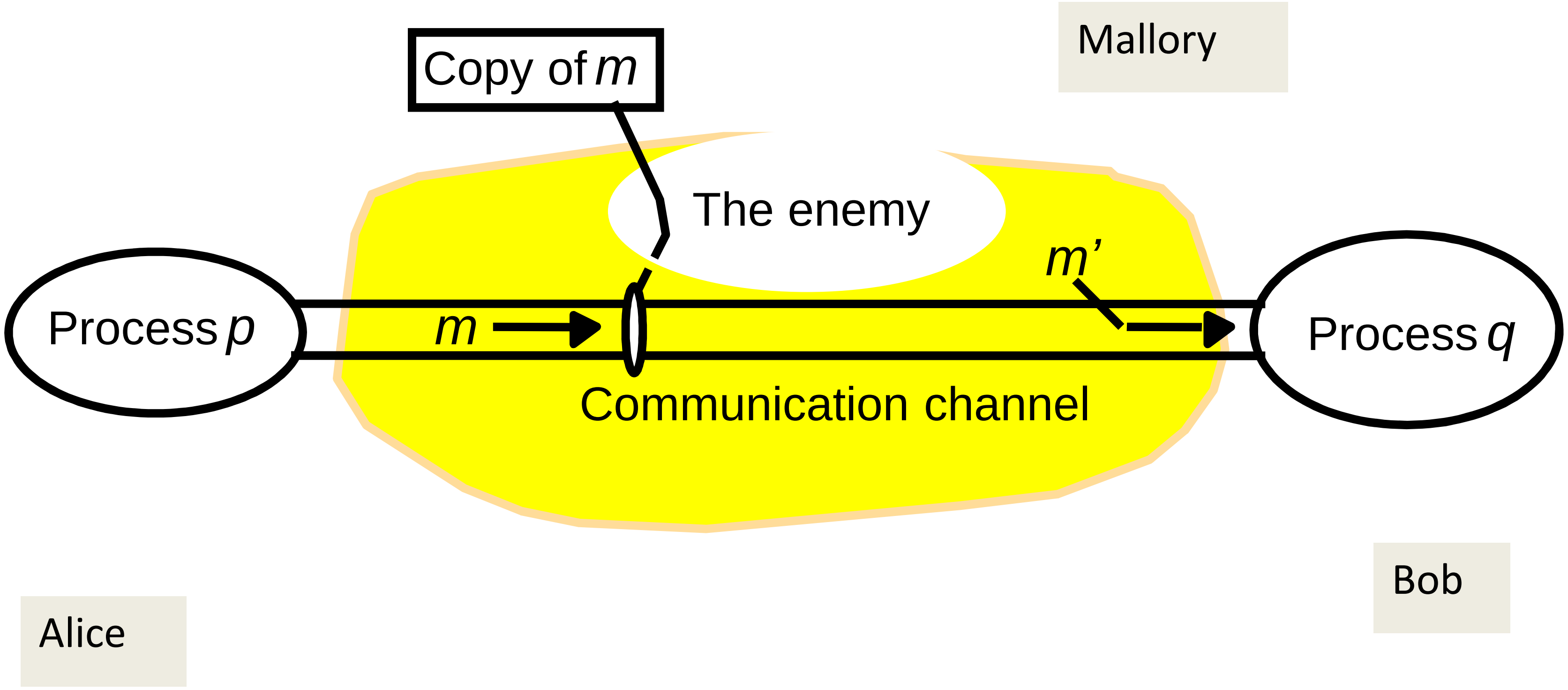
Security model

- Object: intended for use by different clients, via remote invocation
- Principal: authority on whose behalf invocation is issued



The enemy

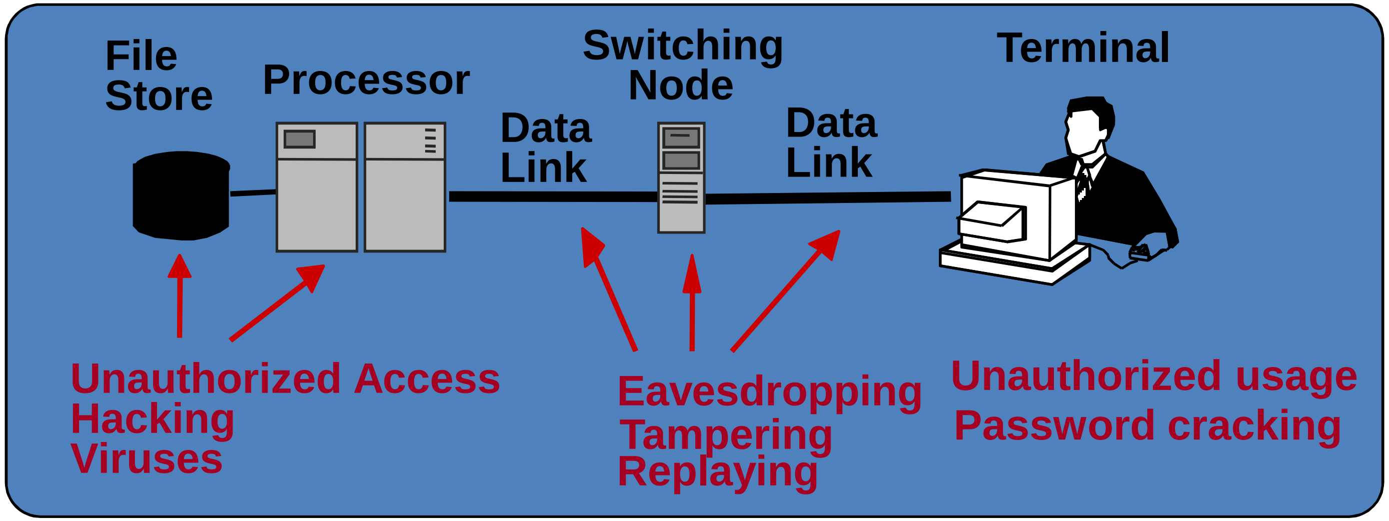
- Processes: encapsulate resources, interact by messages
- Messages: exposed to attack by enemy



Security threats: examples

- Online shopping/banking
  - intercept credit card information
  - purchase goods using stolen credit card details
  - replay bank transaction, e.g. credit an account
- Online stock market information service
  - observe frequency or timing of requests to deduce useful information, e.g. the level of stock
- Website
  - flooding with requests (denial of service)
- My computer
  - receive/download malicious code (virus)

Security threats: what & where



Security threats fall into three categories

Leakage:acquisition of info by unauthorised recipient

Tampering: unauthorised alteration

Vandalism: interference with the property of a system without gain to the perpetrator.

Types of security threats

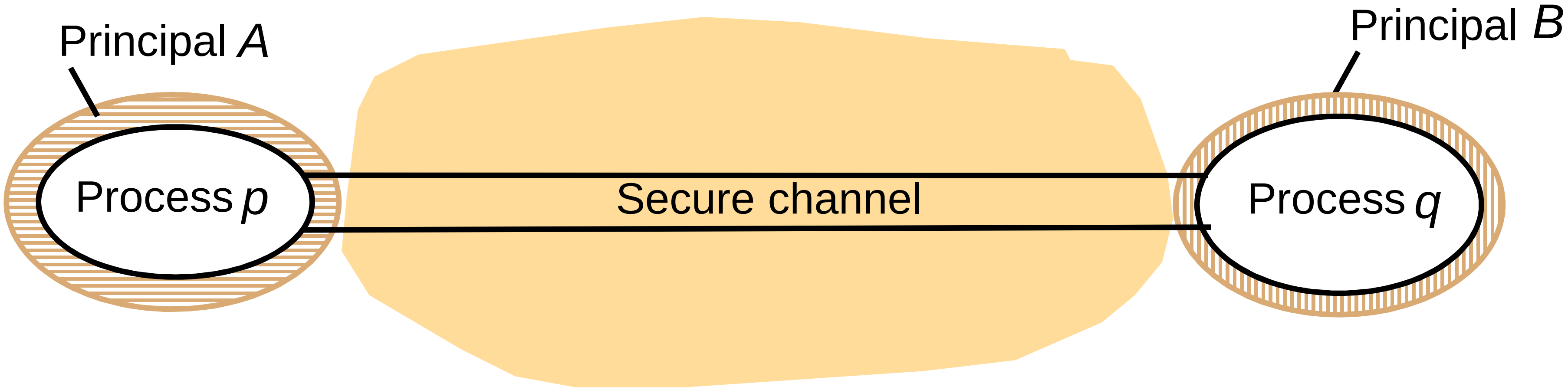
- Eavesdropping
  - obtaining copies of messages without authority
- Masquerading
  - sending/receiving messages using the identity of another principal without their authority
- Message tampering
  - intercepting and altering messages
- Replaying
  - intercepting, storing and replaying messages
- Denial of service
  - flooding a channel with requests to deny access to others

Defeating the enemy: how?

- Encryption (scrambling a message to hide its contents)
  - does not prove identity of sender
- Shared secrets (keys)
  - messages encrypted with the shared key
  - can only be decrypted if the key is known
- Identification (are you who you are?)
  - password protection, etc
- Authentication (are you who you say you are?)
  - include in message identity of principal/data, timestamp
  - encrypt with shared key

Secure channels

- Processes: reliably know identity of principal
- Messages: protected against tampering, timestamped to prevent replaying/reordering.



Threats due to mobility...

- Mobile code (Java JVM)
  - applets, mobile agents (travel collecting information)
  - downloaded from server, run locally
- Security issues: what if the program...
  - illegally writes to a file?
  - writes over another program’s memory?
  - crashes?
- Some solutions
  - stored separately from other classes
  - type-checking and code-validation (instruction subset)
  - still does not guard fully against programming errors...

Designing secure systems

- Basic message
  - networks are insecure
  - interfaces are exposed
- Threat analysis
  - assume worst-case scenario
  - list all threats - complex scenarios!!!
- Design guidelines
  - log at points of entry so that violations detected
  - limit the lifetime and scope of each secret
  - publish algorithms, restrict access to shared keys
  - minimise trusted base

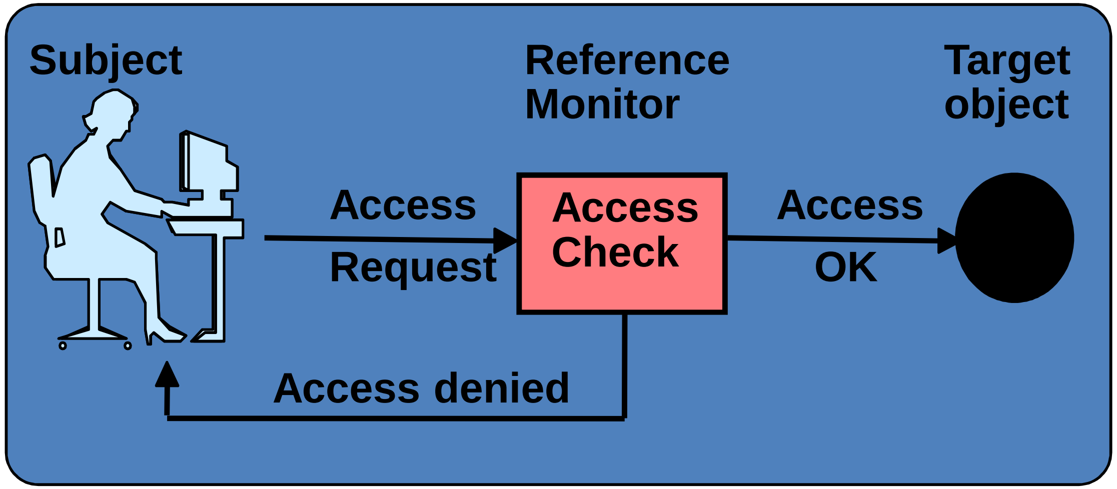
Main security techniques

- Access control
  - implement resource protection, e.g. file protection
  - essential in distributed systems (remote login)
- Firewalls
  - monitor traffic into and out of intranet
- Cryptographic algorithms
  - ciphers
  - authentication
  - digital signatures

Access control

- Definition
  - ensure that users/processes access computer resources in a controlled and authorised manner
- Protection domain
  - is a set of rights for each resource, e.g. Unix files
  - associated with each principal
- Two implementations of protection domains
  - Capabilities
    - ❑request accompanied by key, simple access check
    - ❑open to key theft, or key retained when person left company
  - Access control lists
    - ❑list of rights stored with each resource
    - ❑request requires authentication of principal

Access control



How it works: Reference Monitor

intercepts all access attempts

authenticates request and principal’s credentials

applies access control

- ❑ if Yes, access proceeds
- ❑ if No, access is denied, error message returned to the subject

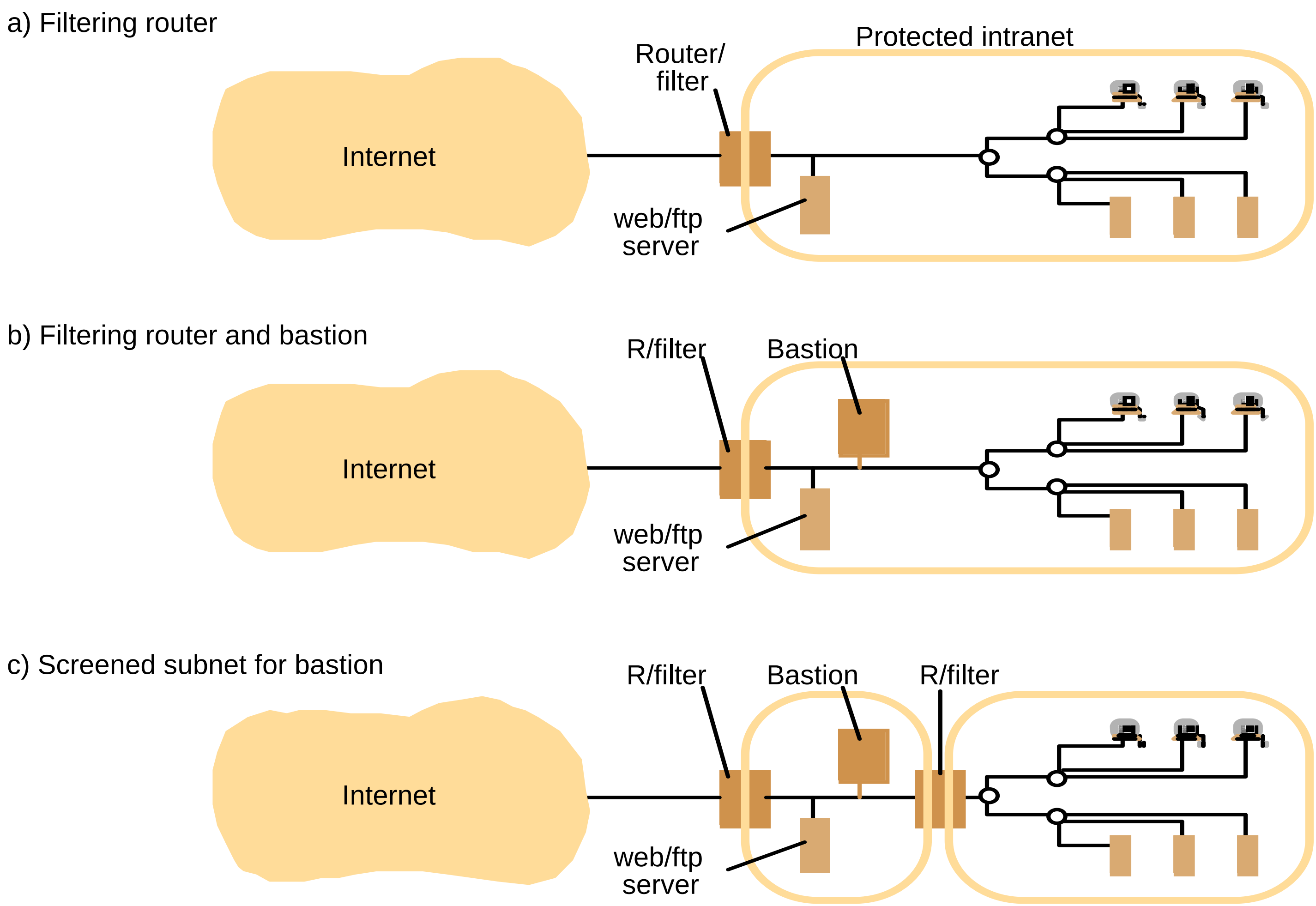
Firewalls

- Monitor and control all communication into and out of an intranet.
- Service control:
  - filter requests for services on internal hosts
  - e.g. reject HTTP request unless to official webserver
- Behaviour control
  - prevent illegal or anti-social behaviour
  - e.g. filter ‘spam’ messages
- User control:
  - allow access to authorised group of users
  - e.g. dial-up services

How does it work...

- A set of processes, at different protocol levels:
- IP packet filtering
  - screening of source & destination, only ‘clean’ packets proceed
  - performed in OS kernel of router
- TCP gateway
  - monitors TCP connection requests
- Application-level gateway
  - runs proxy for an application on TCP gateway, e.g. Telnet
- Bastion
  - separate computer within intranet
  - protected by IP packet filtering, runs TCP/application gateway

Firewall configurations



a) Filtering router

b) Filtering router and bastion

c) Screened subnet for bastion

Cryptographic algorithms

- Encryption
  - apply rules to transform *plaintext* to *ciphertext*
  - defined with a function F and key K
  - denote message M encrypted with K by

$$F_K(M) = \{M\}_K$$

- Decryption
  - uses inverse function

$$F^{-1}_K(\{M\}_K) = M$$

  - can be symmetric (based on secret key known to both parties)
  - or asymmetric (based on public key)

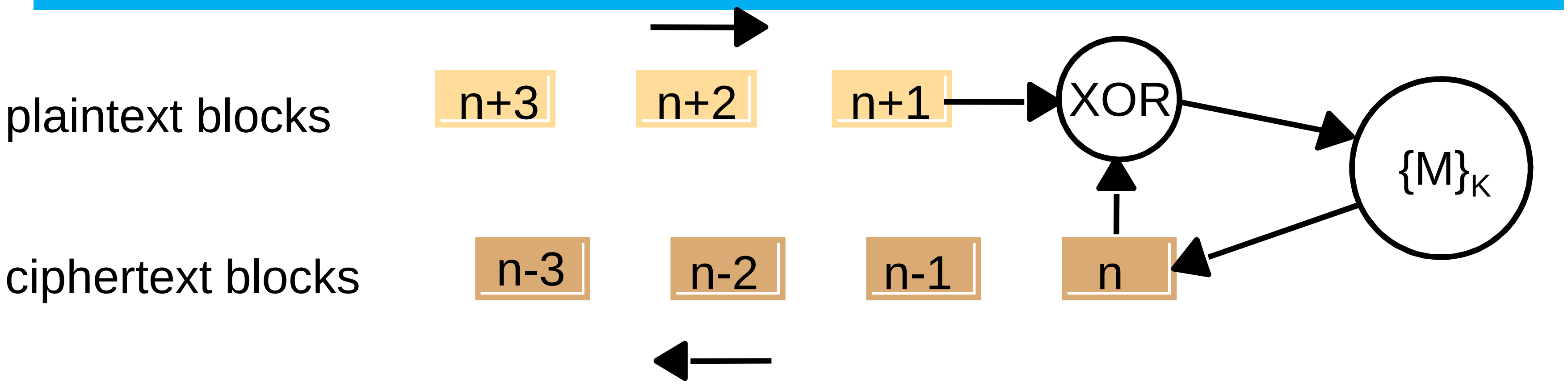
Symmetric cryptography

- One-way functions
  - encryption function F_K easy to compute
  - decryption function F^{-1}_K hard, not feasible in practice
- Idea
  - difficult to discover F_K given $\{M\}_K$
  - difficulty increases with K, to prevent brute-force attack
  - combine blocks of *plaintext* with key through series of XOR, bit shifting, etc, obscuring bit pattern
- Examples
  - several algorithms: TEA, DES
  - secure secret key size 128 bits

Asymmetric cryptography

- Trap-door functions
  - pair of keys (e.g. large numbers)
  - encryption function easy to compute (e.g. multiply keys)
  - decryption function infeasible unless secret known (e.g. factorise the product if one key not known)
- Idea
  - two keys produced: encryption key made public, decryption key kept secret
  - anyone can encrypt messages, only participant with decryption key can operate the trap door
- Examples
  - a few practical schemes: RSA

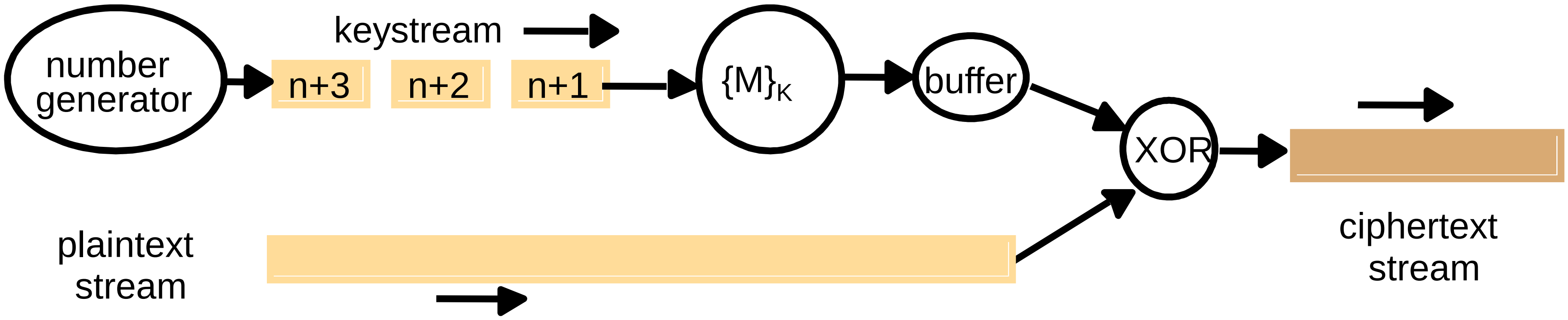
Block ciphers



- How it works
  - message divided into fixed size blocks (64 bits), padded
  - encryption block by block
- Cipher block chaining
  - combine each *plaintext* block with preceding *ciphertext* block using XOR before encryption

Stream ciphers

- How it works
  - used when no block structure (e.g. voice)
  - encryption performed bit by bit
  - generate sequence of numbers, concatenate into secure bit stream



summary

- Security
  - achieves privacy, integrity and availability of distributed systems
- Main techniques
  - access control
  - firewalls
  - cryptography
- Design issues
  - consider worst-case scenario of threats
  - balance cost versus risk
  - log and detect