Lecture 1
Applied Cryptography (Part 1)

ENGG5105/CSCI5470 Computer and Network Security

Spring 2014

Patrick P. C. Lee

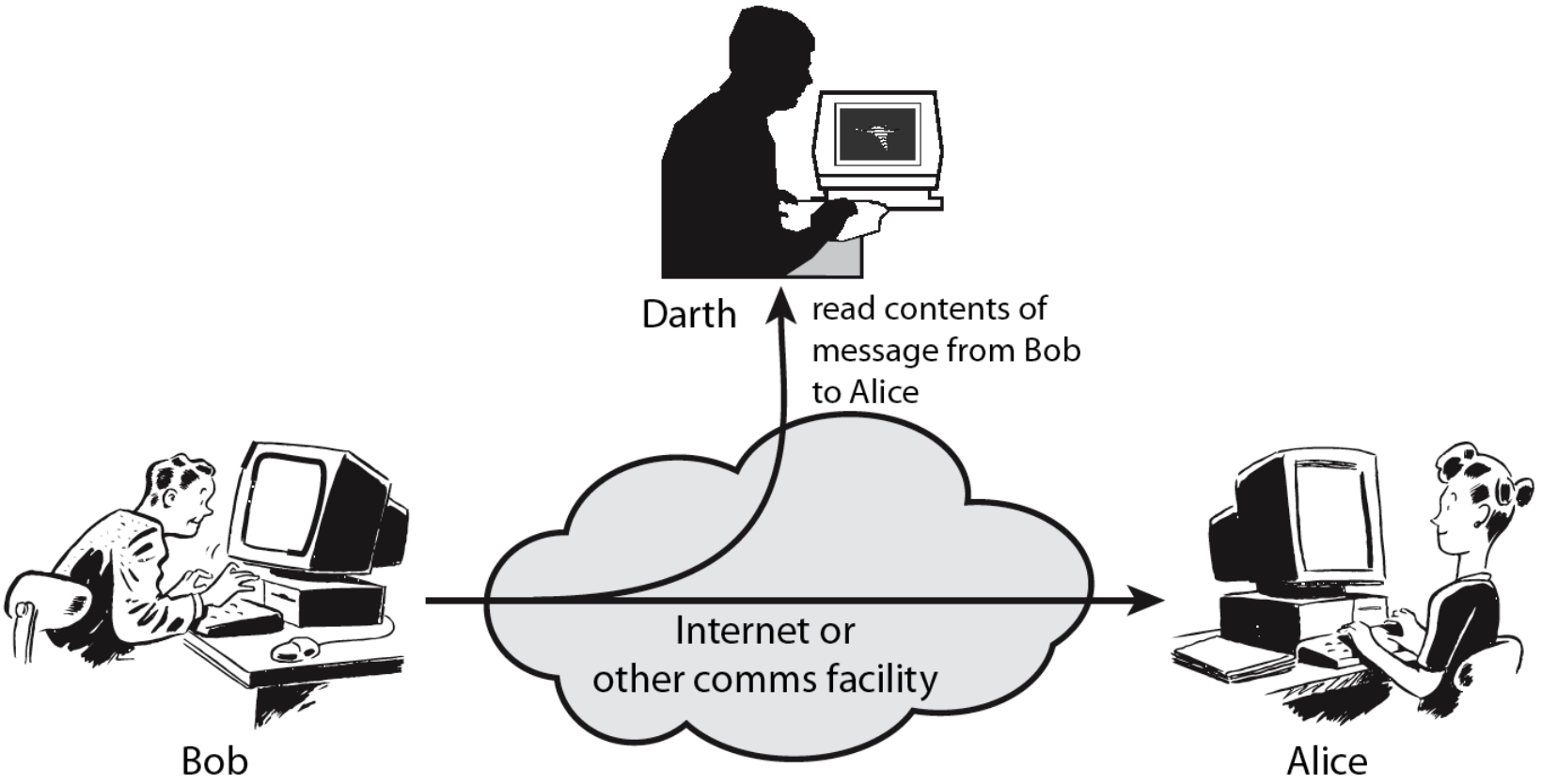
Roadmap

- Introduction to Security
- Introduction to Cryptography
- Symmetric key cryptography
- Hash and message authentication

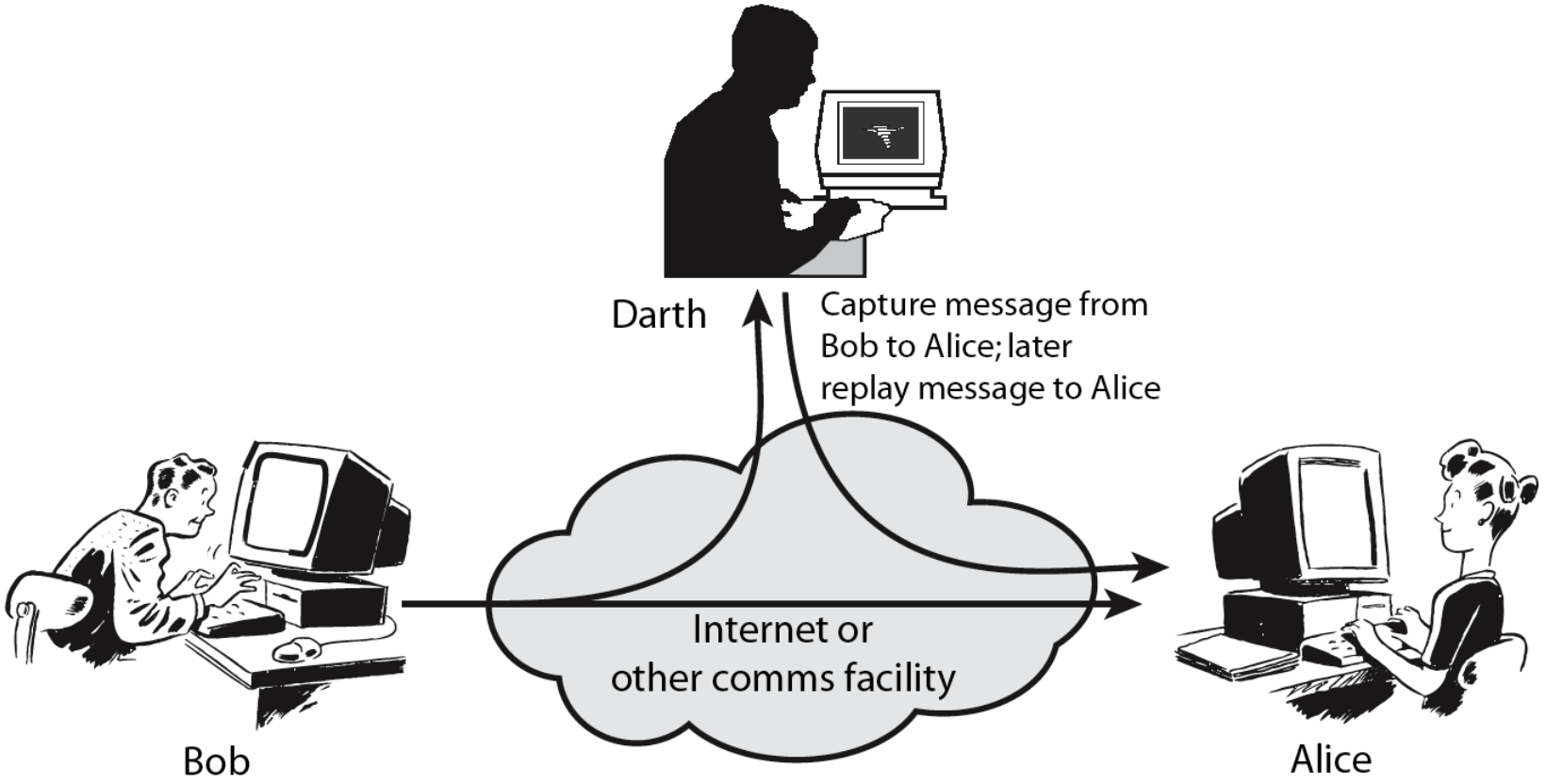
Introduction to Security

- A computer system or network application is said to be secure if it behaves normally even some bad guys try to attack it and make it misbehave.
- Definition of attack (RFC 2828)
  - An assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security property of a system.

Passive Attack



Active Attack



Threat Model

- A **threat model** defines the scope and assumptions of security that we consider
  - i.e., the problem formulation
- Two ways to define a threat model:
  - Attacker's perspective:
    - What does an attacker tries to do?
    - What are the assumptions?
  - Defender's perspective:
    - What are the security properties we want to achieve?

Security Properties

- *What are the security properties we want (from a defender's perspective)?*
- **Confidentiality**: the protection of data from unauthorized disclosure by passive attacks. Sometimes called privacy.
- **Authentication**: to assure that the message is from the source that it claims to be from.

Security Properties

- **Integrity**: content of message is unaltered.
- **Availability**: a system and a system resource remain accessible and usable upon demand.
- **Non-repudiation**: a sender cannot falsely deny that he sent a message
  - Note its difference from authentication

Roadmap

- Introduction to Security
- Introduction to Cryptography
- Symmetric key cryptography
- Hash and message authentication

What is Cryptography?

- Information of data may be attacked:
  - Eavesdropping: sniffing and recording of data transmitted over a channel
    - Passive attack
  - Modification, insertion, or deletion of messages or message content
    - Active attack
- **Cryptography** is the ability to send information between participants, in a mangled format, that prevents others from reading it.
  - Address the issue of information security.

What is Cryptography?

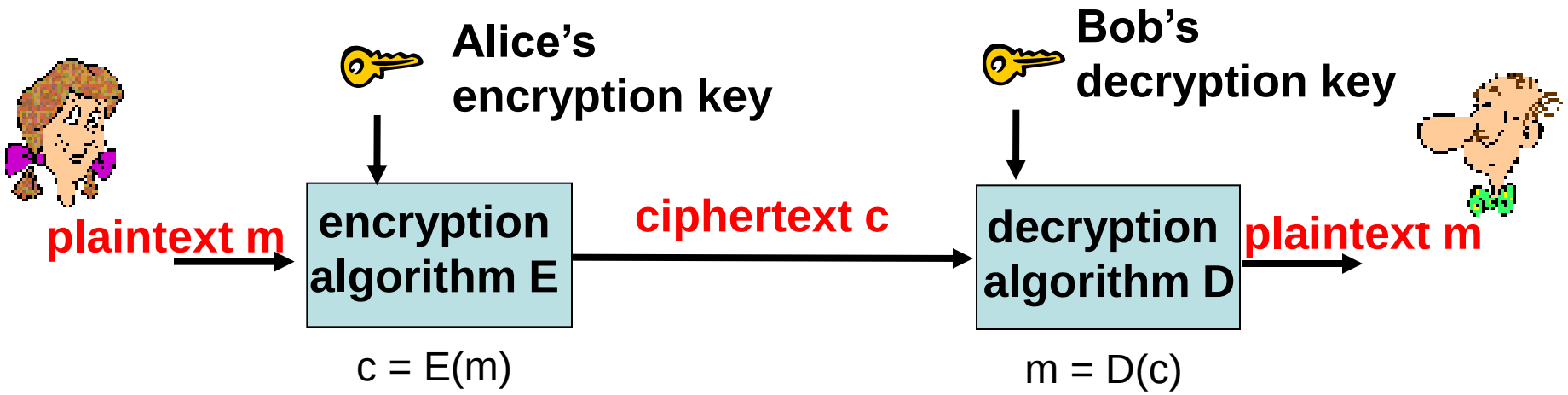
- A message in its original form is called plaintext or cleartext. The mangled information is called ciphertext. The process of producing ciphertext from plaintext is called encryption. The reverse of encryption is called decryption.
- Parties involved (to help our discussion):
  - Alice, Bob, Carol, Dave: good guys who communicate securely
  - Trudy and Eve: bad guys who disrupt

What is Cryptography?

- A cryptosystem is a system that is built on cryptographic algorithms.
- Cryptosystems generally involve encryption/decryption algorithms and secret values called keys.
- Without the knowledge of keys, it is computationally infeasible to recover protected information.
  - Computational difficulty

What is Cryptography?

- A cryptosystem should remain secure as long as the keys are kept secret, even through algorithm details are published
- A typical cryptosystem:



Attacks on Cryptography

- An attacker can try every possible key to recover the plaintext, i.e., via brute force.
- The goal of cryptanalysis is to recover keys based on cryptographic characteristics.
- Three types of attacks:
  - Ciphertext only: only ciphertext is known to attackers.
  - Known plaintext: some <plaintext, ciphertext> pairs are known to attackers
  - Chosen plaintext: Some selected plaintext and the corresponding ciphertext are known to attackers

Cryptographic Primitives

- Cryptographic primitives are specific cryptographic functions that form the basic building blocks of a cryptosystem.
- Examples:
  - Symmetric key cryptography
  - Public key cryptography
  - Hash functions
  - Digital signatures
  - Certificates

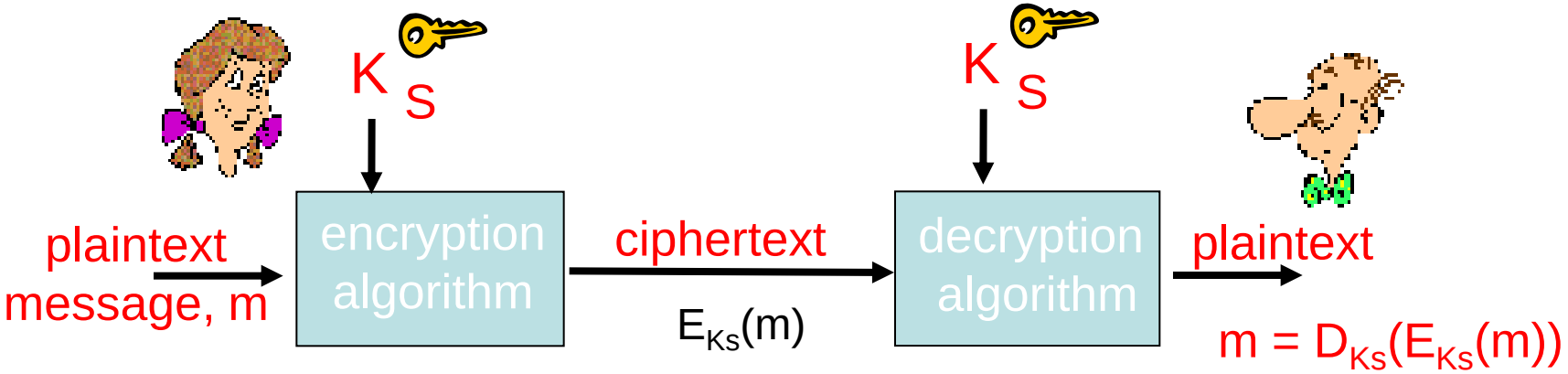
Cryptographic Primitives

- When we build a cryptosystem, leverage existing cryptographic primitives whenever possible.
- Our focus in this course is to apply existing cryptographic primitives into a cryptosystem.
  - In other words, how to have them interact with each other such that the cryptosystem remains secure.

Roadmap

- Introduction to Security
- Introduction to Cryptography
- Symmetric key cryptography
- Hash and message authentication

Symmetric Key Cryptography



- symmetric key crypto: Bob and Alice share same (symmetric) key: K
  - Also known as secret key crypto
- For now, assume Bob and Alice have secret and reliable ways to agree on key.

Generic Block Encryption

- Symmetric key cryptography is mainly built on block ciphers:
  - Algorithm takes a fixed-length block of plaintext and a fixed-length key, and generates a block of ciphertext with the same length as the input.
  - E.g., DES uses 64-bit blocks and a 56-bit key
- Another type is called stream ciphers, which encrypt one byte at a time.

Generic Block Encryption

- Definitely, key length can't be too short. But block length is also a matter.
- Block length consideration:
  - Too small: easy to reconstruct all mappings (e.g., chosen-plaintext attack, try all plaintext blocks)
  - Too big: performance degradation
  - 64-bit / 128-bit may be reasonable.

Generic Block Encryption

- Input and output blocks must form one-to-one mappings:
  - E.g., block length = 3 bits. One mapping structure:

input	output
000	110
001	111
010	101
011	100

input	output
100	011
101	010
110	000
111	001

- How many mapping structures are there if n-bit blocks are used?

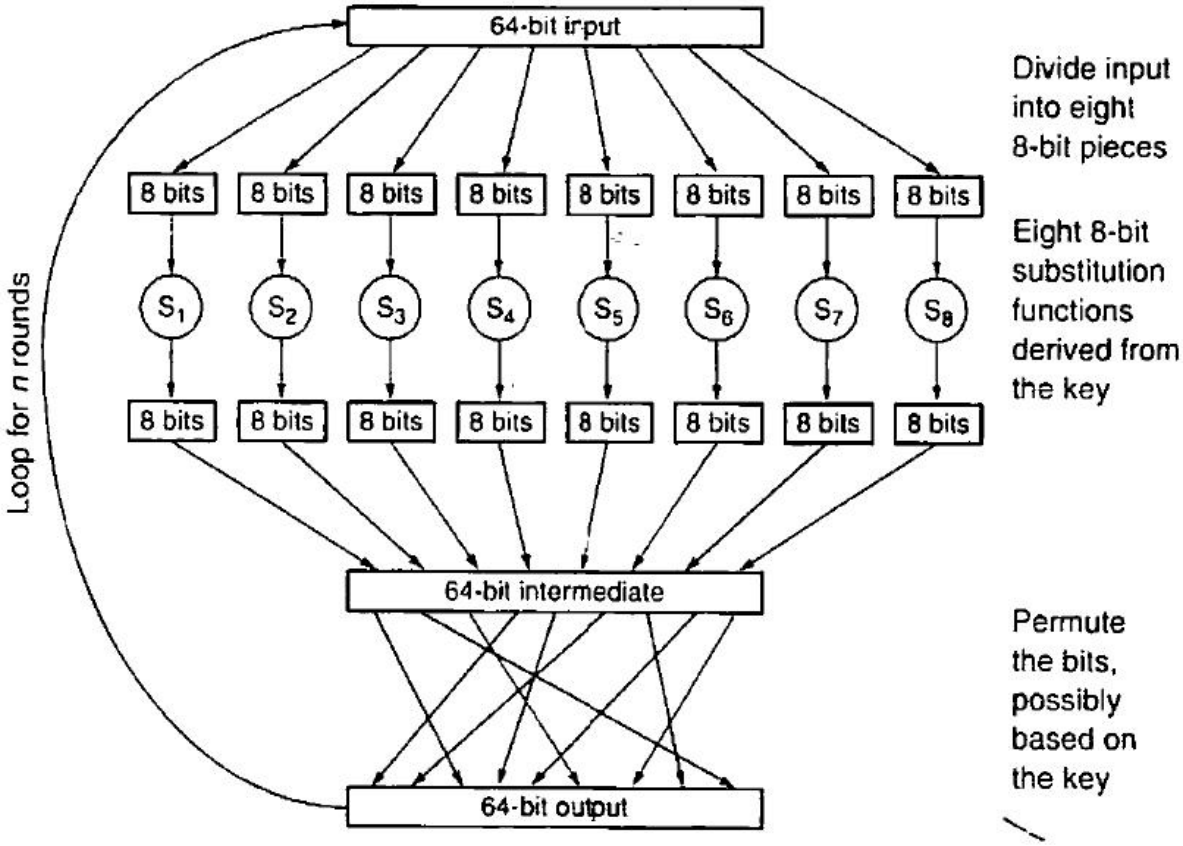
Generic Block Encryption

- Specifying the whole mapping structure is exhaustive.
- Symmetric key crypto. is to take a reasonable-length key and generate a mapping structure that looks random
- How many mapping structures can be specified if a k-bit key is used?

Generic Block Encryption

- Typical operations of symmetric key crypto
  - Substitution: specify mapping of an m-bit input to an m-bit output (assuming m is small)
  - Permutation: specify mapping of an input bit position to an output bit position (special case of substitution)
  - In each round, run substitution and permutation. Repeat many rounds.

Example of Block Encryption

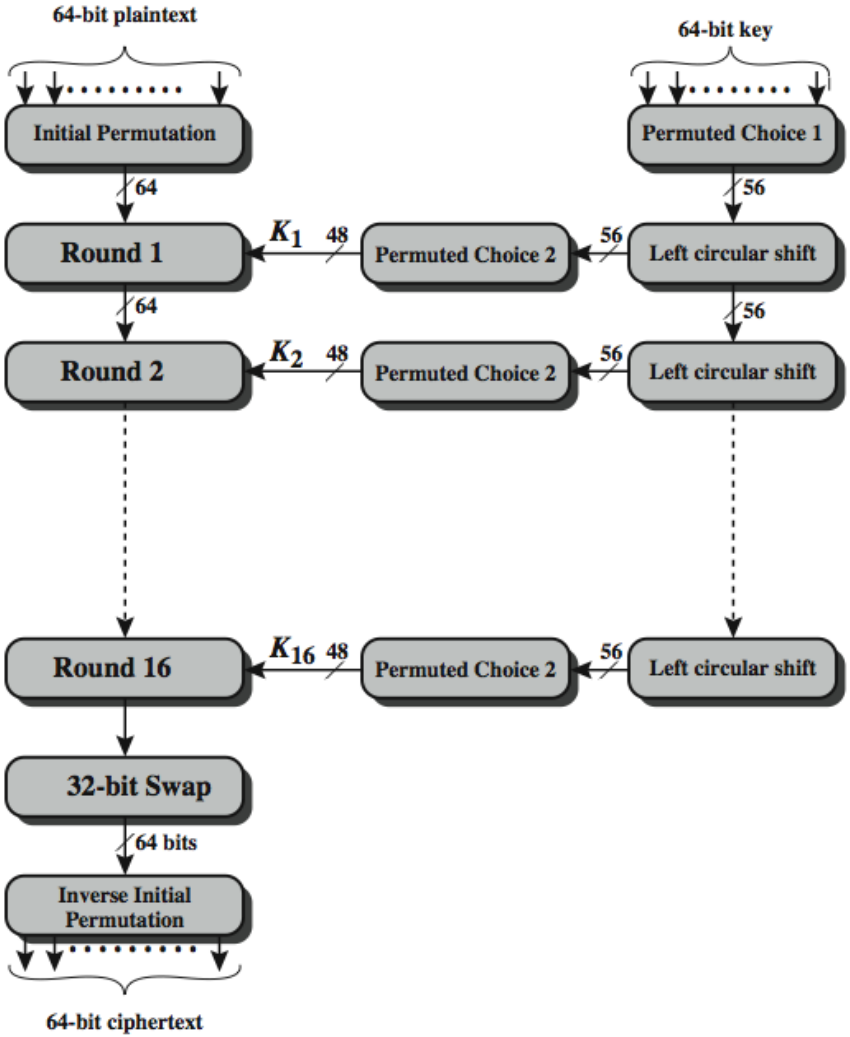


Kaufman et al. (2002), Page 62.

Data Encryption Standard (DES)

- adopted in 1977 by NBS (now NIST)
- encrypts 64-bit data using 56-bit key
- has widespread use
- with controversial security
- Why 56 bits, not 64 bits?
  - 8 bits are for parity checking of keys (is parity useful?)
  - 56 bits less secure

DES Overview



Avalanche Effect

- **Avalanche effect**: change of one input or key bit results in changing many output bits
  - A key desirable property of encryption algo
- In DES, for one round, change of one bit makes approx half of output bits change
- DES exhibits strong avalanche

Is DES Perfect?

- Problems of DES
  - Key size and block size are too short
    - A feasible DES cracker was available in 1998
  - Efficient in hardware. Slow in software
    - A variant called Triple-DES is too slow although more secure

Advanced Encryption Standard (AES)

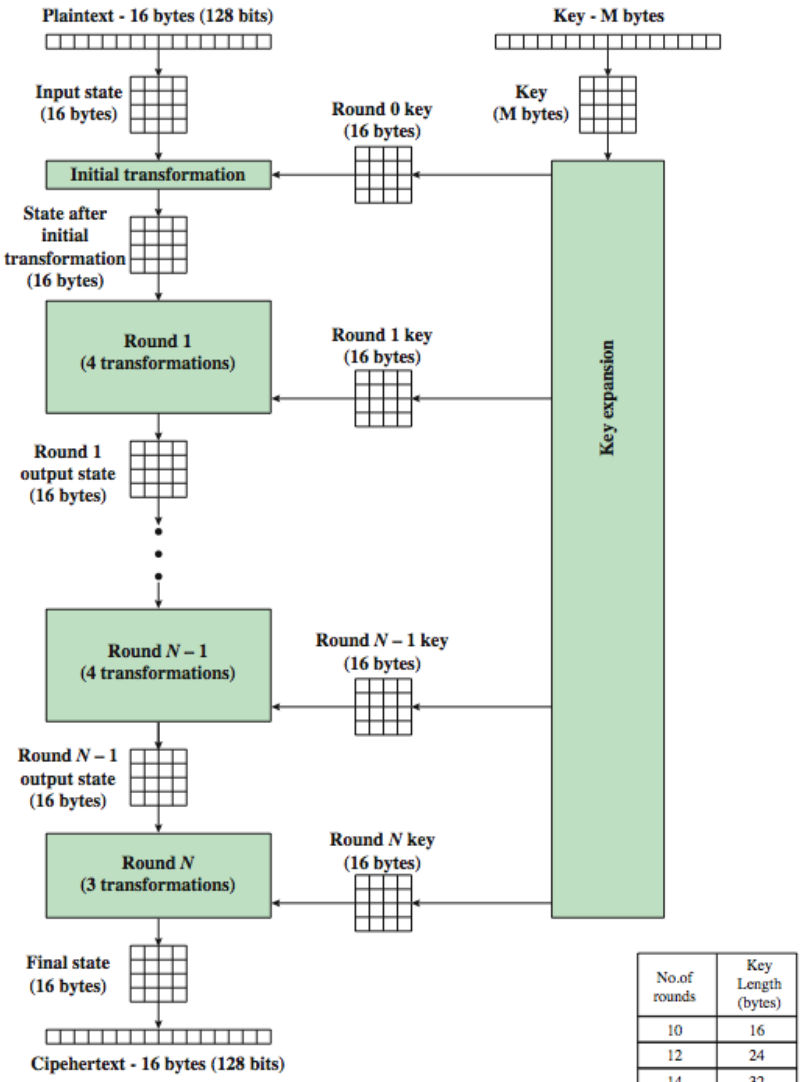
- new (Nov. 2001) symmetric-key NIST standard, replacing DES
- always processes data in 128 bit blocks regardless of the key size
- 128, 192, or 256 bit keys
- brute force decryption (try each key) takes 149 trillion years for AES
  - taking 1 sec on DES

AES

- Design goals of AES
  - resistant against known attacks
  - speed and code compactness on many CPUs
    - Can be efficiently implemented on 8-bit and 32-bit CPUs
  - design simplicity
- Now widely used

AES

- Overview: Encryption process in AES



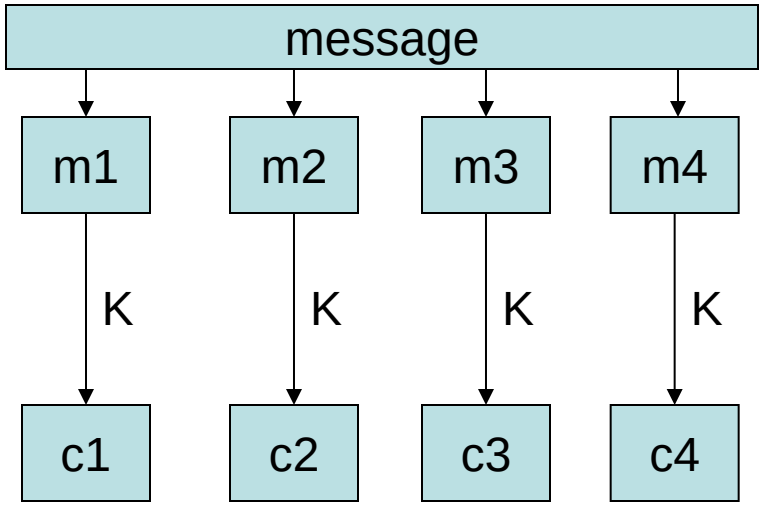
No.of rounds	Key Length (bytes)
10	16
12	24
14	32

Encrypting Large Messages

- Let block size = 64 bits.
- How do you encrypt a message larger than 64 bits?
- Modes of operations:
  - Electronic Code Block (ECB)
  - Cipher Block Chaining (CBC)
  - Output Feedback Mode (OFB) – not discussed
  - Counter Mode (CTR)

Electronic Code Block (ECB)

- Break a message into 64-bit blocks (padding the last one to a full block). Encrypt each block with a secret key.



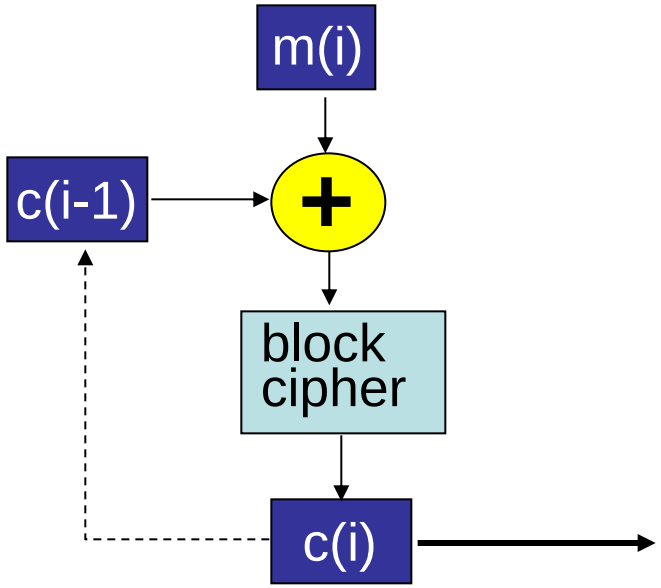
Electronic Code Block

- If two plaintext blocks are identical, the output ciphertext blocks are also identical
  - Reveal the pattern of information
- Can rearrange blocks to affect the protocol
  - E.g., know two blocks that store salaries $50K and $100K. Can reorder the two blocks and change the salaries of employees

Cipher Block Chaining (CBC)

- Make ciphertext blocks different even they are from the same plaintext blocks

- CBC has encryption of current block depend on result of previous block
- Provide initialization vector (IV) to encrypt first block
  - IV need not be secret
- Change IV over each session



Counter Mode (CTR)

- In CBC, what happen if a ciphertext block is dropped, or garbled?
- Counter mode:
  - c(i) = E(key, IV + block number, m(i))
  - Advantage: can decrypt an arbitrary block
  - Disadvantage: Same ciphertext block if key and IV are the same

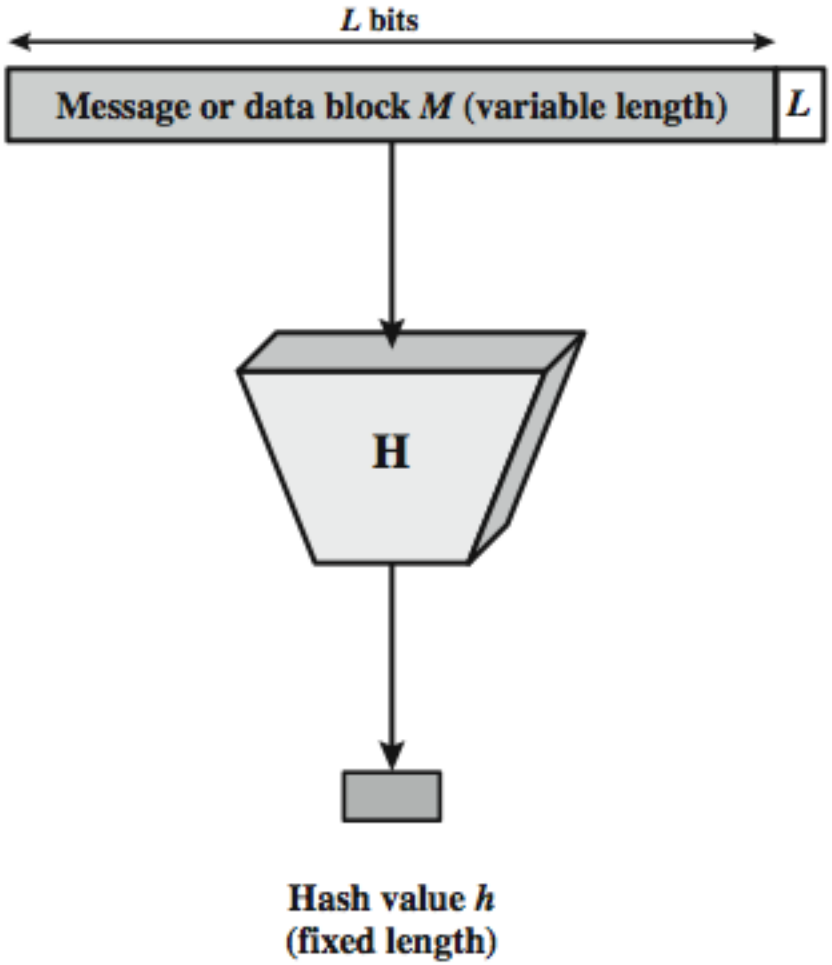
Roadmap

- Introduction to Security
- Introduction to Cryptography
- Symmetric key cryptography
- Hash and message authentication

Hashes

- Take arbitrary message to generate fixed-size output hash (or digest) h = H(m)
- Implementation of hash function is public
- Uses of hashes:
  - Sign hash instead of messages
  - Store digests of files for integrity checks (e.g., viruses may modify files)
  - Irreversible password hash database

Hashes


L bits
Message or data block M (variable length)
L
H
Hash value h
(fixed length)

Cryptographic Hash

- Requirements of a cryptographic hash:
  - One-way property: Given a hash value h, computationally infeasible to find x s.t. h = H(x)
  - Collision resistance property: Computationally infeasible to find any pair (x, y) s.t. H(x) = H(y)

Attacks on Hash Functions

- Attacks via brute force or cryptanalysis
- Attack on one-way:
  - find x such that H(x) equals a given value
- Attack on collision resistance:
  - find x and y such that H(x) = H(y)
- Which attack is easier to launch?

Birthday Attack

- Birthday paradox: in a group of people N, probability of two sharing same birthday > 0.5 if N >= 23
- Birthday attack:
  - given user prepared to sign a valid message x using k-bit hash
  - opponent generates $2^{k/2}$ variations x' of legitimate message x, all with essentially the same meaning, and saves them
  - opponent generates $2^{k/2}$ variations y' of a desired fraudulent message y
  - two sets of messages are compared to find pair with same hash (probability > 0.5 by birthday paradox)
  - have user sign the valid message, then substitute the forgery which will have a valid signature
- Conclusion: need large enough k

Hash Algorithms

- MD5 (RFC 1321)
  - computes 128-bit message digest
  - In 2005, collision attack is feasibly launched (using a laptop running with a few hours)

Secure Hash Algorithm

- SHA originally designed by NIST & NSA in 1993, and was revised in 1995 as SHA-1
- SHA-1
  - US standard for use with DSA signature scheme
  - produces 160-bit hash values
  - Unlike MD5, SHA-1 is not defined if input message length > 2^{64} bits (but enough in practice)
- recent 2005 results on security of SHA-1 have raised concerns on its use in future applications

Revised SHA

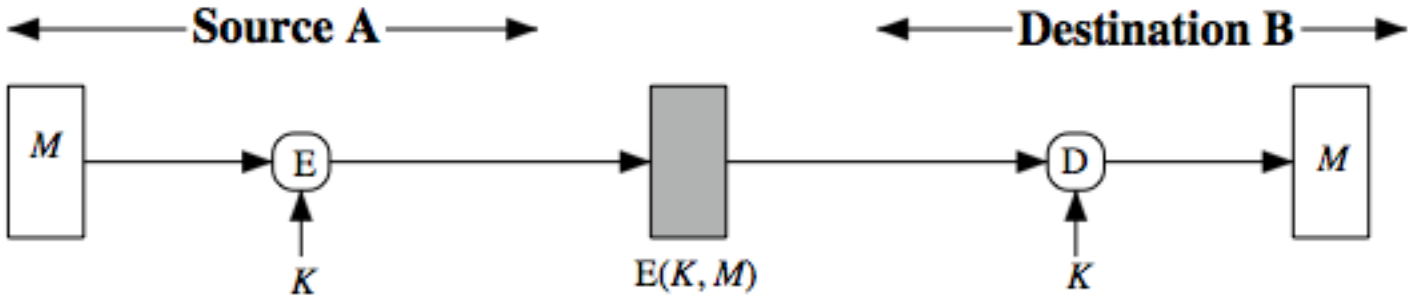
- adds 3 additional versions of SHA
  - SHA-256, SHA-384, SHA-512
- designed for compatibility with increased security provided by the AES cipher
- structure & detail is similar to SHA-1
- hence analysis should be similar
- but security levels are rather higher

Message Authentication

- message authentication is concerned with:
  - protecting the integrity of a message
  - validating identity of originator
  - non-repudiation of origin
    - No denial of message transmission by origin
- then three alternative functions used:
  - hash function – but can't guarantee identity
  - message encryption
  - message authentication code (MAC)

Message Encryption

- use encryption for authentication
- if symmetric encryption is used then:
  - receiver know sender must have created it and content can't be altered
- Drawback: not efficient



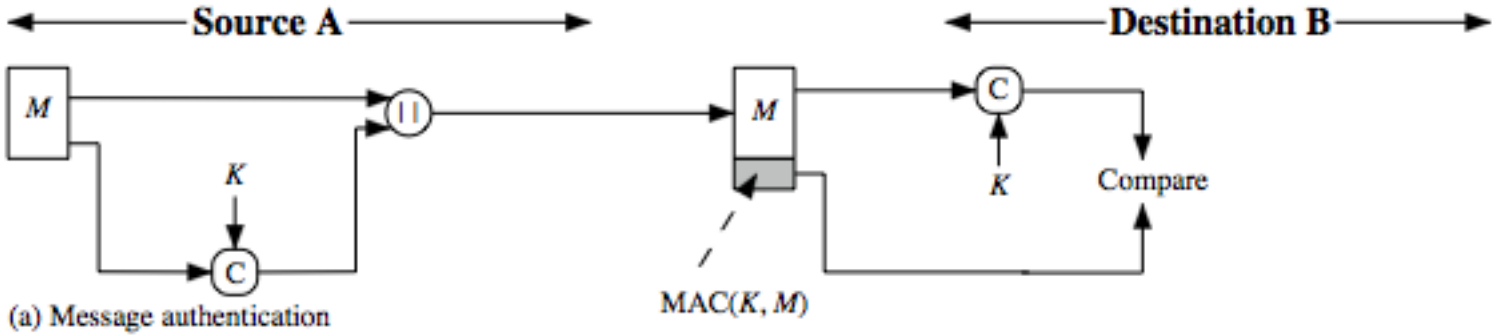
(a) Symmetric encryption: confidentiality and authentication

Message Authentication Code (MAC)

- generated by an algorithm that creates a small fixed-sized block
  - depending on both message and some key
  - like encryption though need not be reversible
- appended to message as a **signature**
- receiver performs same computation on message and checks it matches the MAC
- provides assurance that message is unaltered and comes from sender

MAC

- a small fixed-sized block of data
  - generated from message + secret key
  - MAC = C(K,M)
    - C(.) is some MAC function, may not be hash function
  - appended to message when sent



Keyed Hash Functions as MAC

- But no reason of not using hash function for MAC
- Advantages of using hash functions:
  - because hash functions are generally faster
  - crypto hash function code is widely available
- hash includes a key along with message
- original proposal:

```
KeyedHash = Hash(Key|Message)
```

  - some weaknesses were found with this
- eventually led to development of HMAC

HMAC (Hash-Based MAC)

- specified as Internet standard RFC2104
- HMAC algorithm:
  - Concatenates secret to front of message
  - Hashes concatenated message
  - Concatenates the secret to front of digest
  - Hashes the combination again

HMAC

- uses hash function on the message:

$$HMAC_K(M) = Hash[(K^+ \text{ XOR } opad) \,||\, Hash[(K^+ \text{ XOR } ipad) \,||\, M)]\,]$$

  - where K^+ is the key padded out to size
  - **opad**, **ipad** are specified padding constants
- overhead is just 3 more hash calculations than the message needs alone
  - i.e., 3 more blocks to hash (2 keys and inner hash)
- any hash function can be used
  - e.g., MD5, SHA-1

HMAC

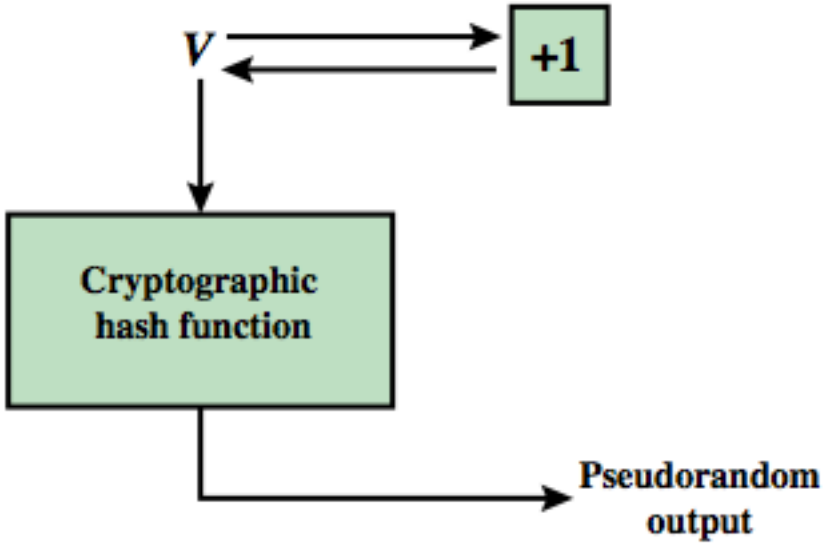
- proved security of HMAC relates to that of the underlying hash algorithm
- attacking HMAC requires either:
  - brute force attack on key used
  - birthday attack (but since keyed would need to observe a very large number of messages)
- choose hash function used based on speed verses security constraints

Pseudorandom Number Generator (PRNG)

- Essential elements of any pseudorandom number generator (PRNG) are a seed value and a deterministic algorithm for generating a stream of pseudorandom bits
- PRNG can be based on
  - Encryption algorithm
  - Hash function
  - MAC

Hash-Based PRNG

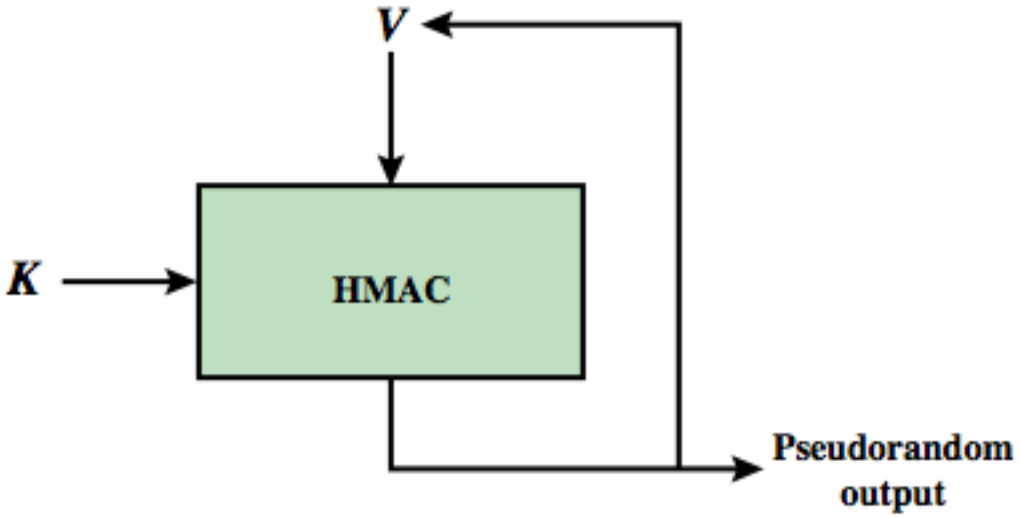
- Steps:
  - Take seed V
  - Repeatedly add 1
  - Hash V
  - Use n-bits of hash
- Secure if good hash is used



(a) PRNG using cryptographic hash function

MAC-based PRNG

- Steps:
  - Use a key K and seed V
  - Input based on last hash



(b) PRNG using HMAC

Summary and References

- Summary:
  - Introduction to security
    - Security defined by **security properties**
  - Introduction to cryptography
    - Cryptographic primitives
  - Symmetric key cryptography
  - Hash functions and message digests
- References:
  - Kaufman et al., Ch. 1-5
  - Stallings, Ch. 1, 2, 3, 5, 11, 12
  - Kurose & Ross, Ch. 8