Week1: Intro to Cryptography

Some slides have been taken from:

- *Computer Networking: A Top Down Approach Featuring the Internet*, 7th edition. Jim Kurose, Keith Ross. Addison-Wesley, 2016.

What is network security?

confidentiality: only sender, intended receiver should "understand" message contents

- sender encrypts message
- receiver decrypts message

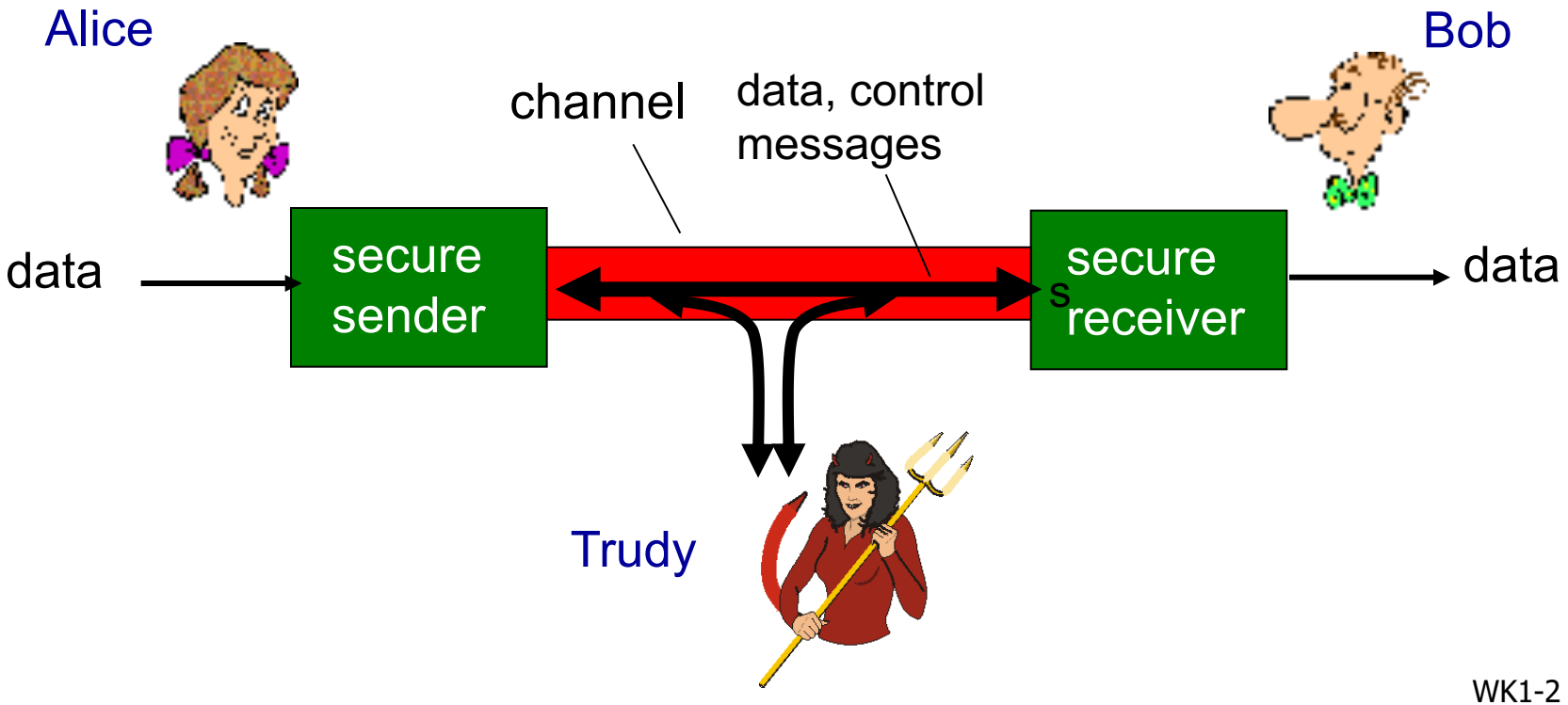
authentication: sender, receiver want to confirm identity of each other

message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

access and availability: services must be accessible and available to users

Friends and enemies: Alice, Bob, Trudy

- well-known in network security world
- Bob, Alice (lovers!) want to communicate "securely"
- Trudy (intruder) may intercept, delete, add messages



Who might Bob, Alice be?

- … well, *real-life* Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates
- other examples?

There are bad guys (and girls) out there!

Q: What can a "bad guy" do?

A: A lot!

- *eavesdrop:* intercept messages
- actively *insert* messages into connection
- *impersonation:* can fake (spoof) source address in packet (or any field in packet)
- *hijacking:* "take over" ongoing connection by removing sender or receiver, inserting himself in place
- *denial of service:* prevent service from being used by others (e.g., by overloading resources)

Chapter 8 roadmap

What is network security?

Principles of cryptography

8.3 Message integrity and digital signatures

8.4 End-point authentication

8.5 Securing e-mail

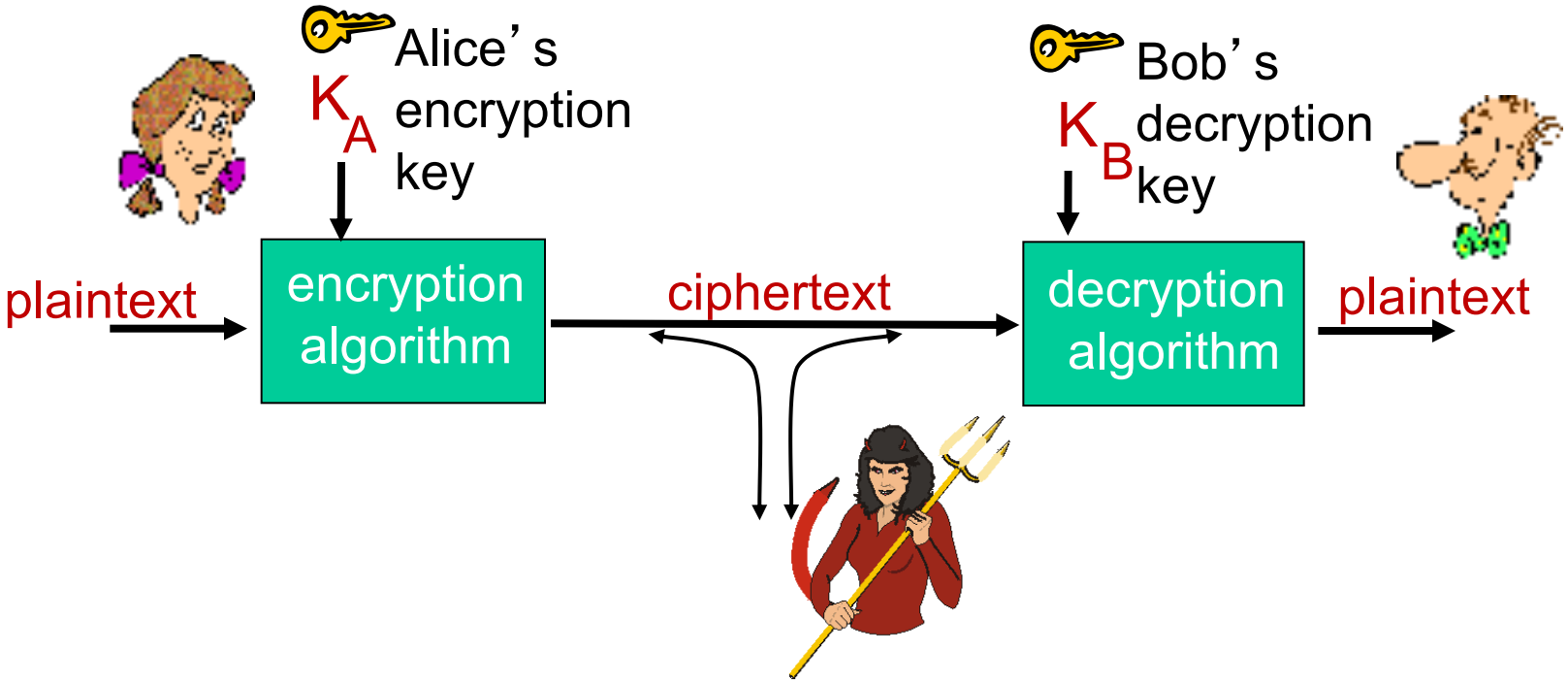
8.6 Securing TCP connections: SSL

8.7 Network layer security: IPsec and VPNs

8.8 Securing wireless LANs

8.9 Operational security: firewalls and IDS

What is a Cryptosystem?



m plaintext message

$K_A(m)$ ciphertext, encrypted with key K_A

$m = K_B(K_A(m))$

Secret Key vs. Secret Algorithm

- Secret Algorithm
  - Military
  - Hard to keep an algorithm secret, if used widely
    - Reverse engineering
- Secret keys
  - Commercial
  - Publish the algorithm while keeping the key secret
  - The design and analysis of today's cryptographic algorithms is highly mathematical. Do not try to design your own algorithms.

Simple encryption scheme

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

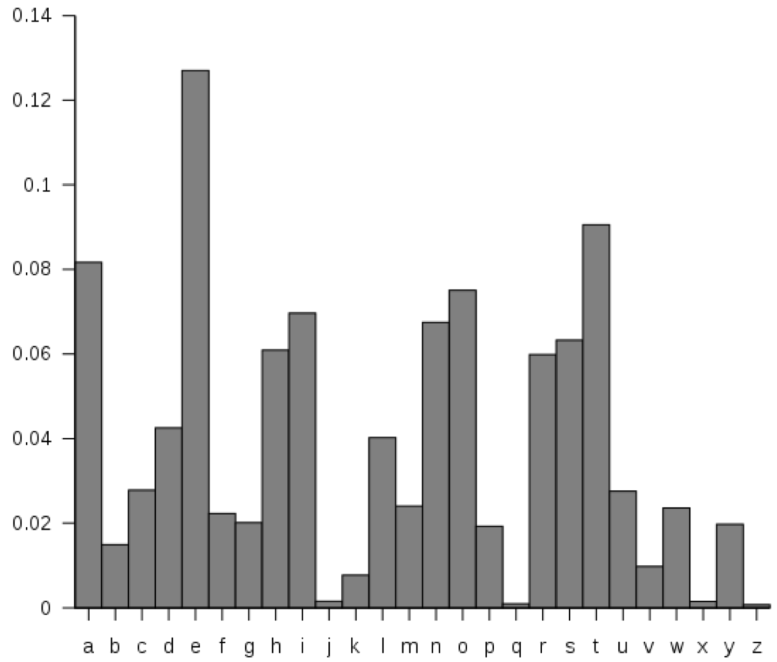
```
plaintext:   abcdefghijklmnopqrstuvwxyz
                ↓                     ↓
ciphertext:  mnbvcxzasdfghjklpoiuytrewq
```

e.g.:

```
Plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc
```

Encryption key: mapping from set of 26 letters to set of 26 letters

Letter Frequency in English



Letter	Relative frequency
A	8.2%
B	1.5%
C	2.8%
D	4.3%
E	12.7
…	…
I	7%
…	…
T	9.1%
…	…
Z	0.1%

Breaking an Encryption Scheme

- Ciphertext only
  - Trudy (bad guy) has access to enough ciphertext only
  - brute-force search until finding a "recognizable plaintext" (Exhaustive)
  - Need enough ciphertext
- Known plaintext
  - Trudy can obtain some <plaintext, ciphertext> pairs
    - secret may be revealed (by spy or time)
  - eg, in monoalphabetic cipher, trudy determines pairings for a,l,i,c,e,b,o,
- Chosen plaintext
  - Trudy can choose a plaintext and have its ciphertext computed
  - Can break mono-alphabetic cipher
    - `the quick brown fox jumps over the lazy dog`

A more sophisticated encryption approach

- n substitution ciphers, $M_1, M_2, \ldots, M_n$
- cycling pattern:
  - e.g., n=5: M_1, M_3, M_4, M_3, M_2; M_1, M_3, M_4, M_3, M_2; ..
- for each new plaintext symbol, use subsequent substitution pattern in cyclic pattern
  - dog: d from M_1, o from M_3, g from M_4

Encryption key: n substitution ciphers, and cyclic pattern

- key need not be just n-bit pattern

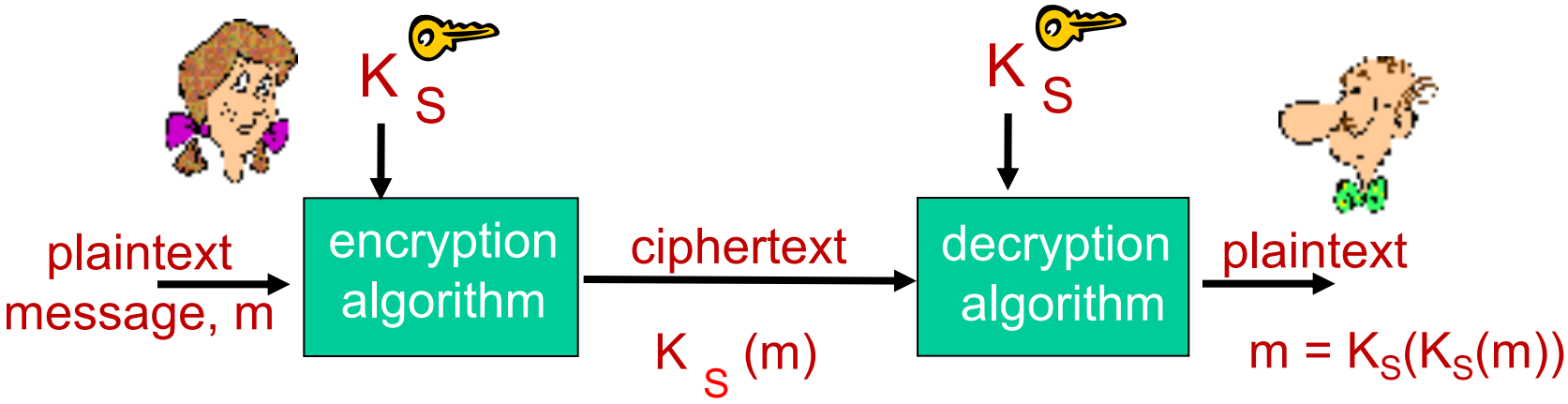
Computational Difficulty

- Algorithm should be efficient to compute but significantly difficult for a brute-force cryptanalysis
  - Brute-force cryptanalysis: try all keys until "looks like" plaintext
  - a longer key means more work for brute-force cryptanalysis
  - encryption: O(N+1), brute-force: $O(2^{N}+1)$
    - twice as hard with each additional bit
- Advances in computing benefit cryptographer more, but make old uses of cryptography easier to break
  - DES (56 bit key) was standardized in 1977. It took 56 hours to break it in 1998, less than 1 day in 2008

Types of Cryptographic Functions

- Crypto often uses keys:
  - Algorithm is known to everyone
  - Only "keys" are secret
- Private key cryptography (symmetric cryptography)
  - one shared key
- Public key cryptography (asymmetric cryptography)
  - two keys (public and private)
- Hash function
  - Zero keys!!
  - How can this be useful?

Symmetric key cryptography



symmetric key crypto: Bob and Alice share same (symmetric) key: K_S

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher

Q: how do Bob and Alice agree on key value?

Summary

- What is cryptography
- Three ways to break cryptography
- Trivial ciphers
- Cryptographic functions and their applications
- Next week: private key cryptography