CSE 484 (Winter 2010)

Software Security: Attacks, Defenses, and Design Principles

Tadayoshi Kohno

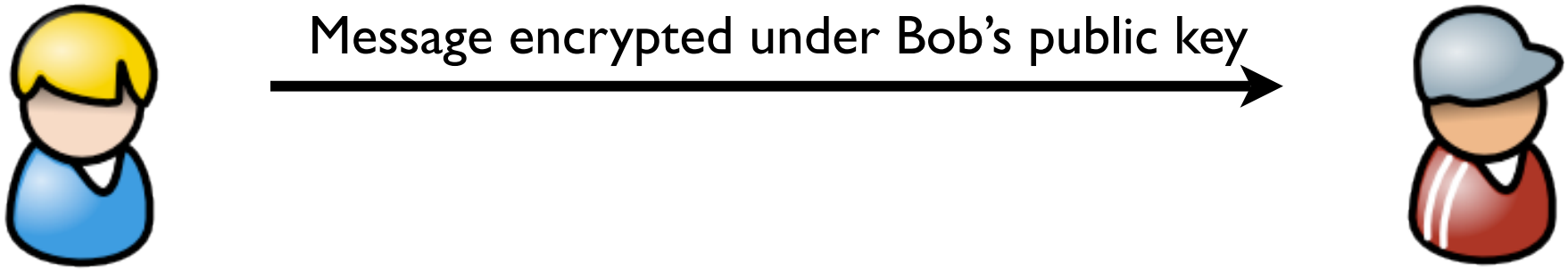
Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Goals for Today

- Cryptography Overview (Continued)
- Brief History
- Under the hood: Symmetric cryptography

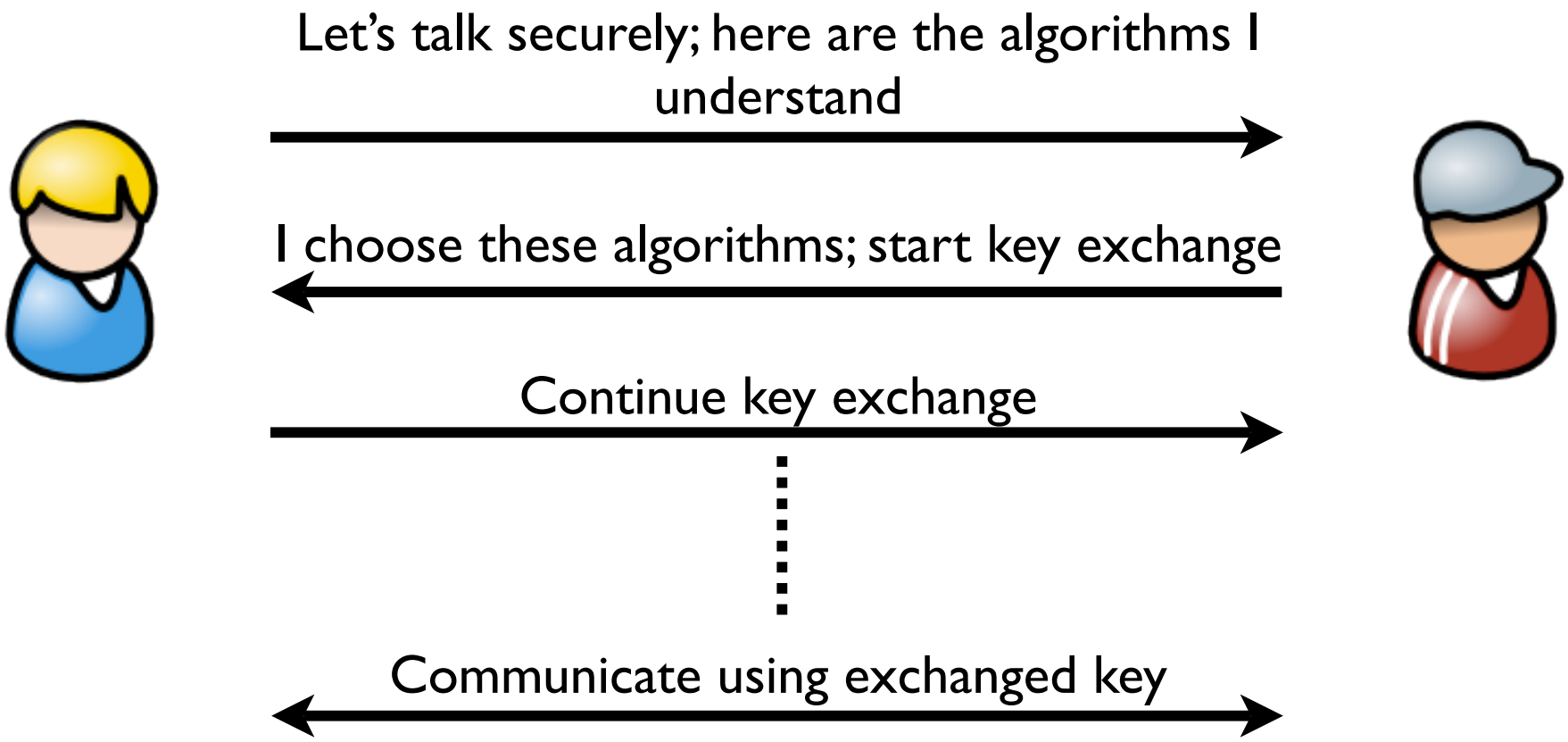
One-way Communications

PGP is a good example



Interactive Communications

In many cases, it's probably a good idea to just use a standard protocol/system like SSH, SSL/TLS, etc...



Let's Dive a Bit Deeper

One-way Communications

(*Informal* example; ignoring, e.g., signatures)

1. Alice gets Bob's public key; Alice *verifies* Bob's public key (e.g., via CA)
2. Alice generates random symmetric keys K1 and K2
3. Alice encrypts the message M the key K1; call result C
4. Alice authenticates (MACs) C with key K2; call the result T
5. Alice encrypts K1 and K2 with Bob's public key; call the result D

6. Send D, C, T

(Assume Bob's private key is encrypted on Bob's disk.)

7. Bob takes his password to derive key K3
8. Bob decrypts his private key with key K3
9. Bob uses private key to decrypt K1 and K2
10. Bob uses K2 to verify MAC tag T
11. Bob uses K1 to decrypt C

Interactive Communications

(*Informal* example; details omitted)

1. Alice and Bob exchange public keys and certificates
2. Alice and Bob use CA's public keys to verify certificates and each other's public keys
3. Alice and Bob take their passwords and derive symmetric keys
4. Alice and Bob use those symmetric keys to decrypt and recover their asymmetric private keys.
5. Alice and Bob use their asymmetric private keys and a *key exchange* algorithm to derive a shared symmetric key

(They key exchange process will require Alice and Bob to generate new pseudorandom numbers)

6. Alice and Bob use shared symmetric key to encrypt and authenticate messages

(Last step will probably also use random numbers; will need to rekey regularly; may need to avoid replay attacks,...)

What cryptosystems have you heard of? (Past or present)

History

- Substitution Ciphers
  - Caesar Cipher
- Transposition Ciphers
- Codebooks
- Machines

- Recommended Reading: **The Codebreakers** by David Kahn and **The Code Book** by Simon Singh.
  - Military uses
  - Rumrunners
  - ....

Classic Encryption

- Goal: To communicate a secret message
- Start with an *algorithm*
- Caesar cipher (substitution cipher):

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

GHIJKLMNOPQRSTUVWXYZABCDEF
```

Then add a secret key

- Both parties know that the secret word is "victory":

  ABCDEFGHIJKLMNOPQRSTUVWXYZ

  VICTORYABDEFGHJKLMNPQSUWXZ

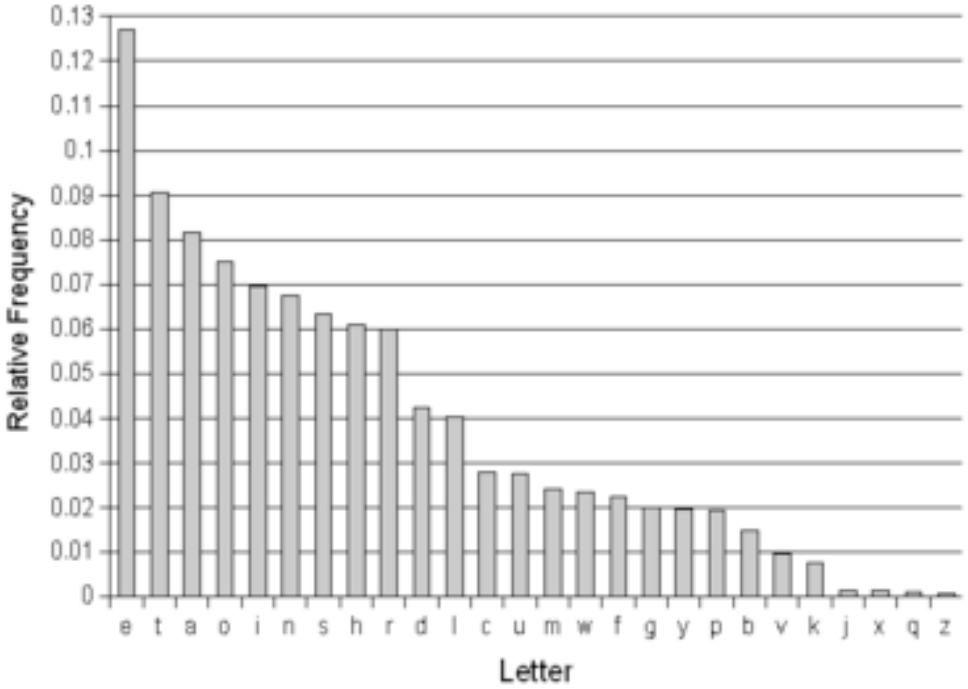
- "state of the art" for thousands of years

Cryptographers vs Cryptanalysts

- A battle that continues today
- Cryptographers try to devise more clever algorithms and keys
- Cryptanalysts search for vulnerabilities
- Early cryptanalysts were linguists:
  - frequency analysis
  - properties of letters

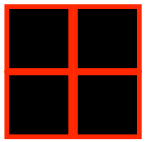
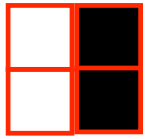
Cryptanalysis and probabilities

Letter	Frequency
a	8.167%
b	1.492%
c	2.782%
d	4.253%
e	12.702%
f	2.228%
g	2.015%
h	6.094%
i	6.966%
j	0.153%
k	0.772%
l	4.025%



From http://en.wikipedia.org/wiki/Letter_frequencies

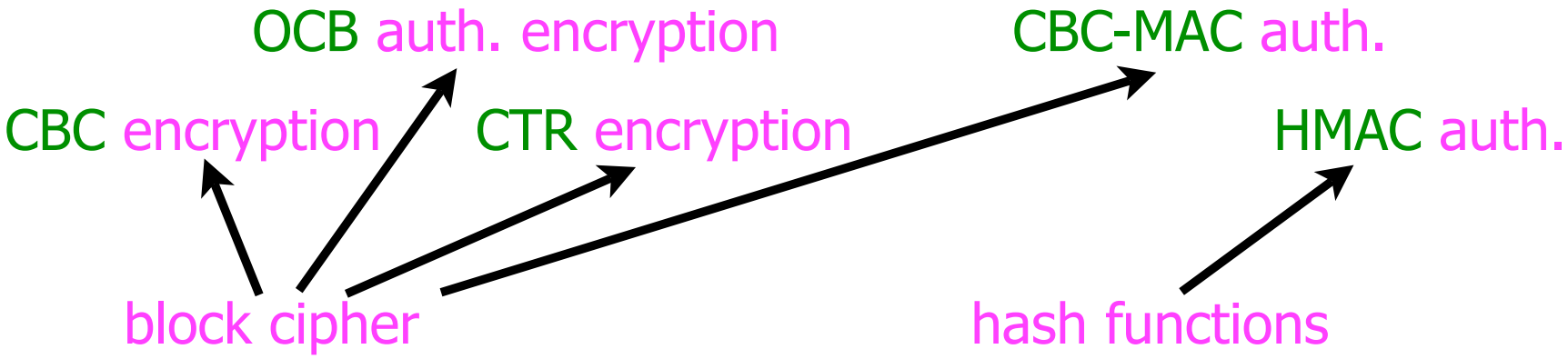
Diversity in Modern Crypto

- **Visual Cryptography**
- Take a black and white bitmap image
- Encode 0 as: 
- Encode 1 as: 
- 1 xor 0 = 0 xor 1 = 1: 
- 1 xor 1 = 0 xor 0 = 0:  or 
- Nice toolkit online here: http://www.cl.cam.ac.uk/~fms27/vck/

See also http://www.cs.washington.edu/homes/yoshi/cs4hs/cse-vc.html

How this is achieved today

- Layered approach:
  - Cryptographic primitives, like block ciphers, stream ciphers, hash functions, and one-way trapdoor permutations
  - Cryptographic protocols, like CBC mode encryption, CTR mode encryption, HMAC message authentication
- Public algorithms (Kerckhoff's Principle)
- Security proofs based on assumptions (not this course)



Attack Scenarios for Encryption

- Ciphertext-Only
- Known Plaintext
- Chosen Plaintext
- Chosen Ciphertext (and Chosen Plaintext)