CSE 484 / CSE M 584 (Winter 2013)

(Start) Cryptography

Tadayoshi Kohno

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

Goals for Today

- Cryptography (Start)
- More on software security defenses after you've had more experience with Lab 1

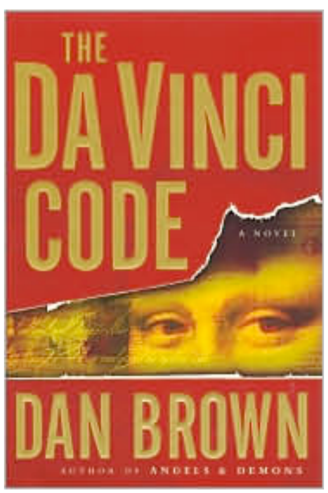
- Lab 1 part 1 due this Friday
- Security Reviews / Current Events due this Friday

- Impressed with the activity on the forums!!

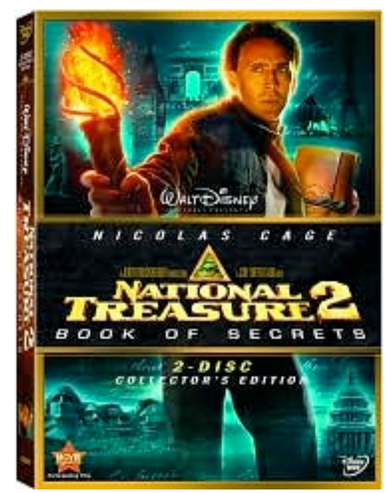
Abj sbe fbzr
Pelcgbtencul!

Pelcgbtencul naq Frphevgl

- Neg naq fpvrapr bs cebgrpgvat bhe vasbezngvba.
- Xrrcvat vg cevingr, vs jr jnag cevinpl
- Cebgrpgvat vgf vagrtevgl, vs jr jnag gb nibvq sbetrevrf.







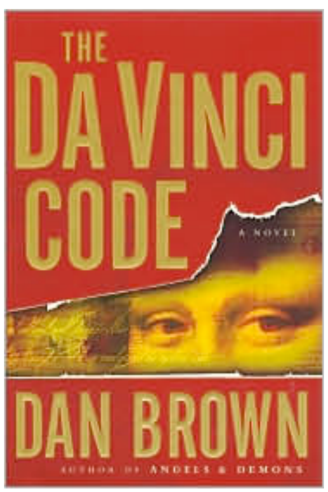
Vzntrf sebz Jvxvcrqvn naq Onearf naq Aboyr

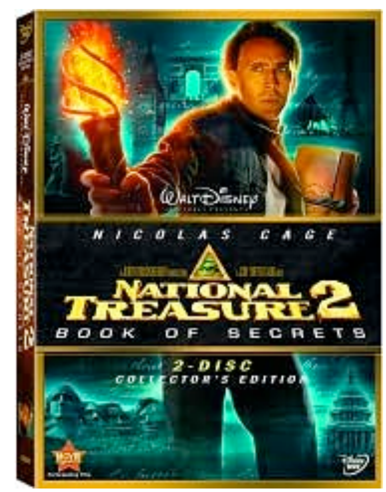
Fbzr gubhtugf nobhg pelcgbtencul

- Pelcgbtencul bayl bar fznyy cvrpr bs n ynetre flfgrz
- Zhfg cebgrpg ragver flfgrz
  - Culfvpny frphevgl
  - Bcrengvat flfgrz frphevgl
  - Argjbex frphevgl
  - Hfref
  - Pelcgbtencul (sbyybjvat fyvqrf)
- "Frphevgl bayl nf fgebat nf gur jrnxrfg yvax"
  - Arrq gb frpher jrnx yvaxf
  - Ohg abg nyjnlf pyrne jung gur jrnxrfg yvax vf (qvssrerag nqirefnevrf naq erfbheprf, qvssrerag nqirefnevny tbnyf)
  - Pelcgb snvyherf znl abg or (vzzrqvngryl) qrgrpgrq
- Pelcgbtencul urycf nsgre lbh'ir vqragvsvrq lbhe guerng zbqry naq tbnyf
  - Snzbhf dhbgr: "Gubfr jub guvax gung pelcgbtencul pna fbyir gurve ceboyrzf qbrfa'g haqrefgnaq pelcgbtencul naq qbrfa'g haqrefgnaq gurve ceboyrzf."

Cryptography and Security

- Art and science of *protecting* our *information*.
  - Keeping it private, if we want privacy
  - Protecting its integrity, if we want to avoid forgeries.



Images from Wikipedia and Barnes and Noble

Some thoughts about cryptography

- Cryptography only one small piece of a larger system
- Must protect entire system
  - Physical security
  - Operating system security
  - Network security
  - Users
  - **Cryptography** (following slides)
- "Security only as strong as the weakest link"
  - Need to secure weak links
  - But not always clear what the weakest link is (different adversaries and resources, different adversarial goals)
  - Crypto failures may not be (immediately) detected
- Cryptography helps after you've identified your threat model and goals
  - Famous quote: "Those who think that cryptography can solve their problems doesn't understand cryptography and doesn't understand their problems."

Improved se

◆ RFIDs in car ke
- RFIDs in car ke
- Result: Car ja

boingboing
A Directory Of Wonderful Things

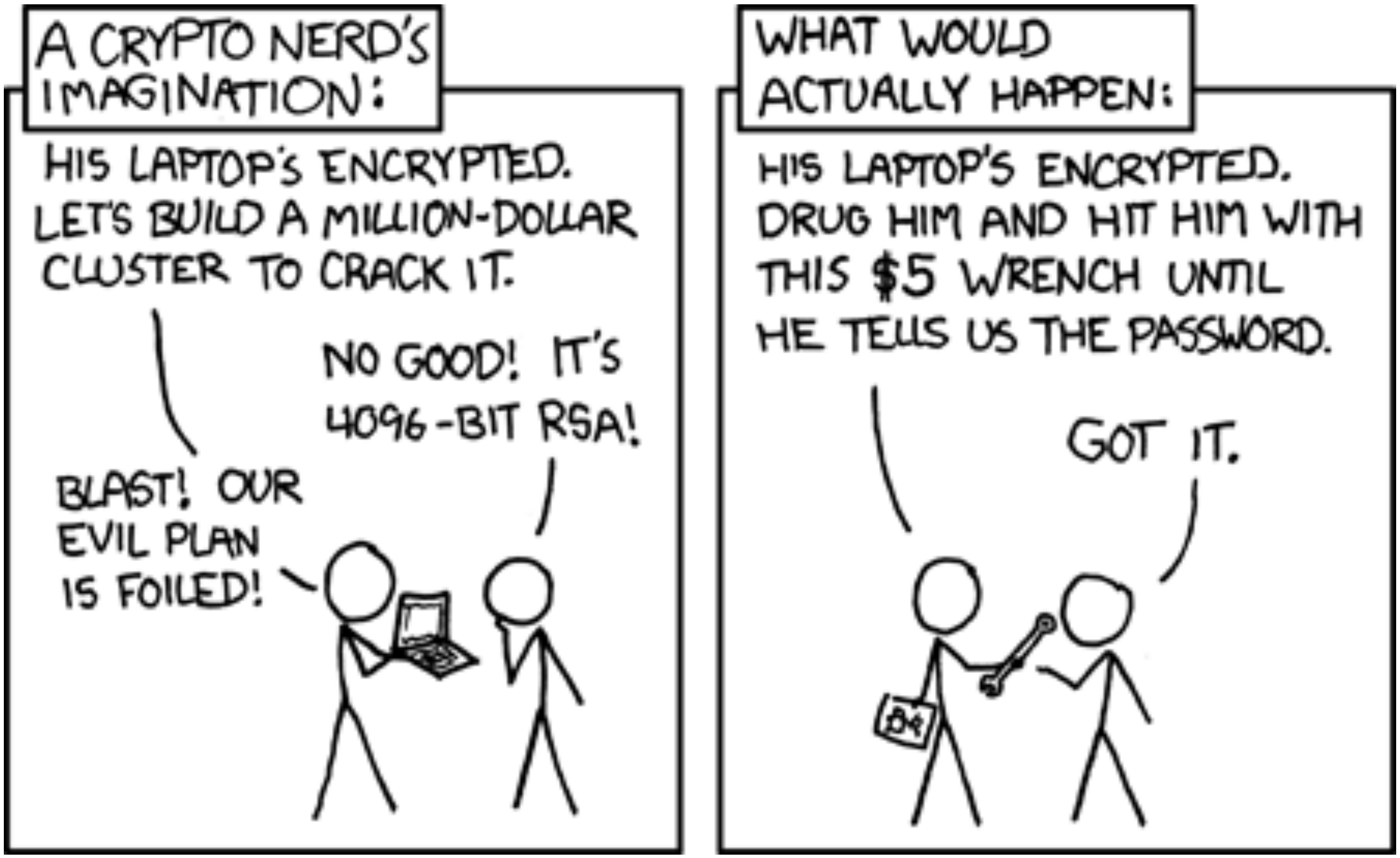
Biometric car lock defeated by cutting off owner's finger

POSTED BY CORY DOCTOROW, MARCH 31, 2005 7:53 AM | PERMALINK

Andrei sez, "'Malaysia car thieves steal finger.' This is what security visionaries Bruce Schneier and Ross Anderson have been warning about for a long time. Protect your $75,000 Mercedes with biometrics and you risk losing whatever body part is required by the biometric mechanism."

> ...[H]aving stripped the car, the thieves became frustrated when they wanted to restart it. They found they again could not bypass the immobiliser, which needs the owner's fingerprint to disarm it.
>
> They stripped Mr Kumaran naked and left him by the side of the road - but not before cutting off the end of his index finger with a machete.

XKCD: http://xkcd.com/538/



Key Entry Pad (4-digit PIN)

- This is the key pad on my office safe.
- Inside my safe is a copy of final exam.
- How long would it take a you to break in?

- Answer (combinatorics):
  - 10^4 tries *maximum*.
  - 10^4 / 2 tries on *average*.
- Answer (unit conversion):
  - 3 seconds per try --> 4 hours and 10 minutes on average

Image from profmason.com

Key Entry Pad (4-digit PIN)

Image from profmason.com

- Now assume the safe automatically calls police after 3 failed attempts.
- What is the probability that you will guess the PIN within 3 tries?
- (Assume no repeat tries.)

- Answer (combinatorics):
  - 10000 choose 3 possible choices for the 3 guesses
  - $1 \times (9999 \text{ choose } 2)$ possible choices contain the correct PIN
  - So success probability is 3 / 10000

Key Entry Pad (4-digit PIN)

- Could you do better at guessing the PIN?

  - Answer (*chemical* combinatorics):
    - Put different chemical on each key (NaCl, KCl, LiCl, ...)

Image from profmason.com

Idea from http://eprint.iacr.org/2003/217.ps

Key Entry Pad (4-digit PIN)

- Could you do better at guessing the PIN?

  - Answer (*chemical* combinatorics):
    - Put different chemical on each key (NaCl, KCl, LiCl, ...)
    - Observe residual patterns after I access safe

Image from profmason.com

Idea from http://eprint.iacr.org/2003/217.ps

Key Entry Pad (4-digit PIN)

- Could you do better at guessing the PIN?

- Answer (*chemical* combinatorics):
  - Put different chemical on each key (NaCl, KCl, LiCl, ...)
  - Observe residual patterns after I access safe

Image from profmason.com

Idea from http://eprint.iacr.org/2003/217.ps

Key Entry Pad (4-digit PIN)

- Could you do better at guessing the PIN?
  - Answer (*chemical* combinatorics):
    - Put different chemical on each key (NaCl, KCl, LiCl, ...)
    - Observe residual patterns after I access safe

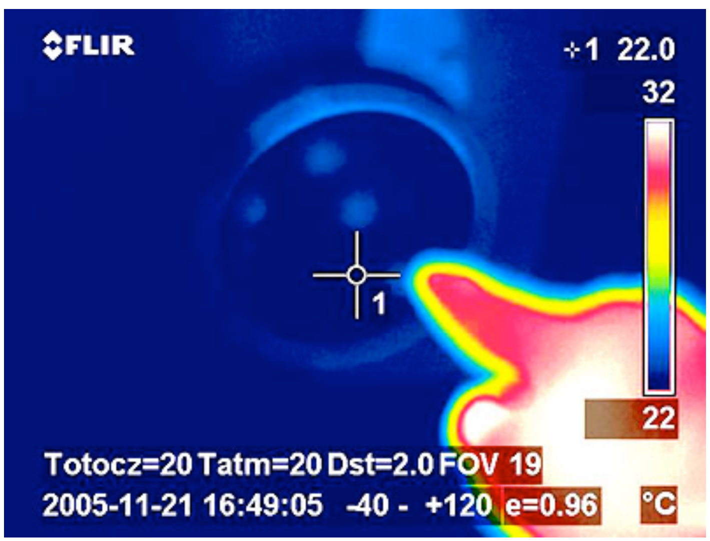
Lesson: Consider the complete system, physical security, etc

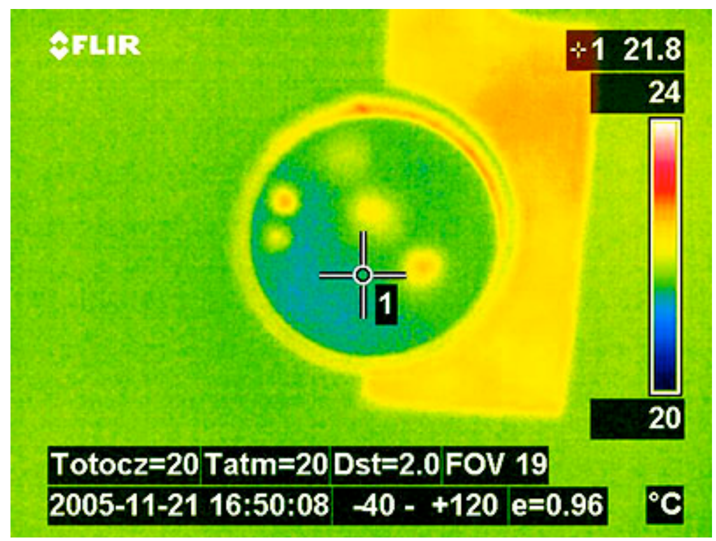
Lesson: Think outside the box

Image from profmason.com

Idea from http://eprint.iacr.org/2003/217.ps

Thermal Patterns



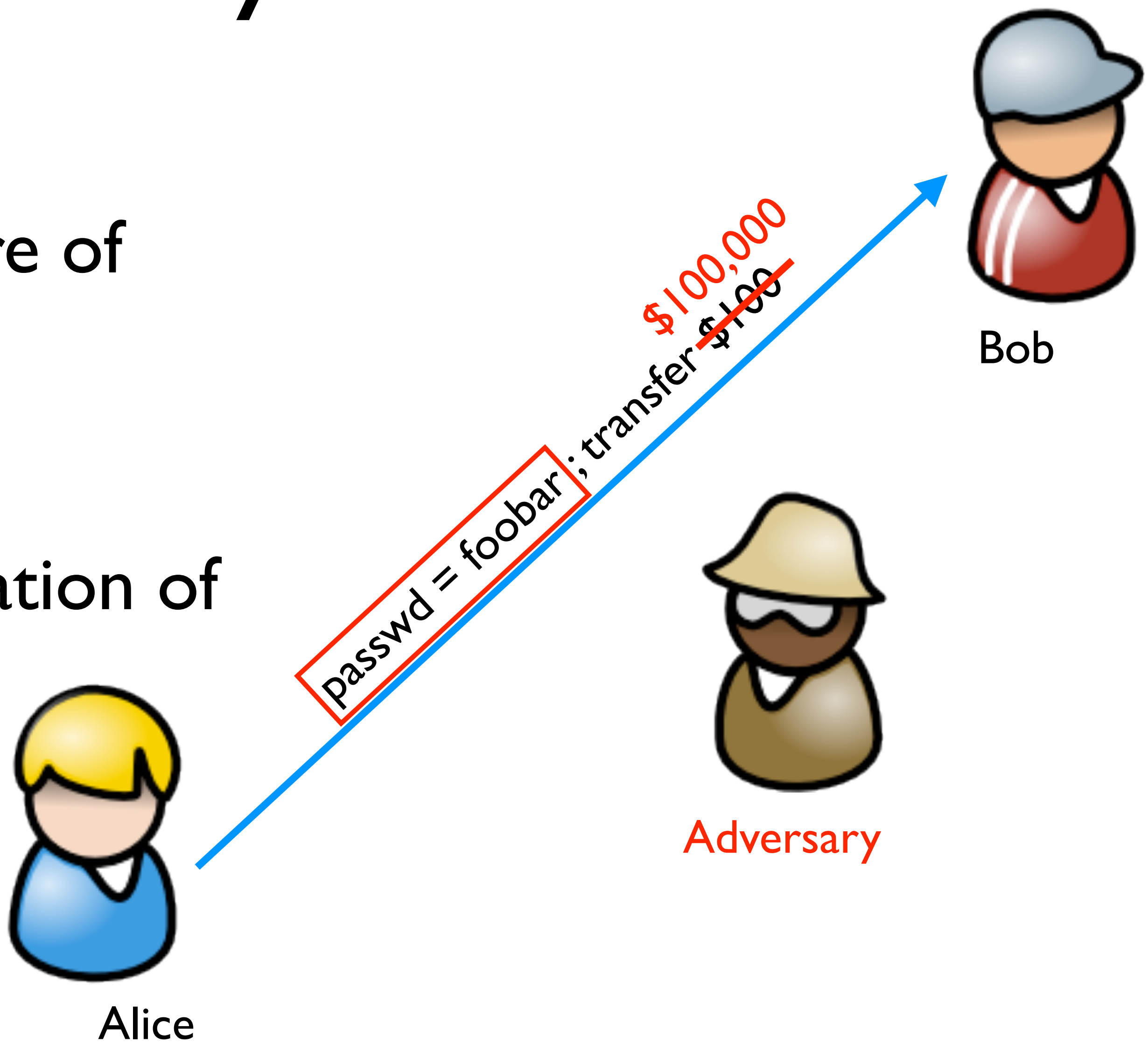


Images from http://lcamtuf.coredump.cx/tsafe/

Common Communication Security Goals

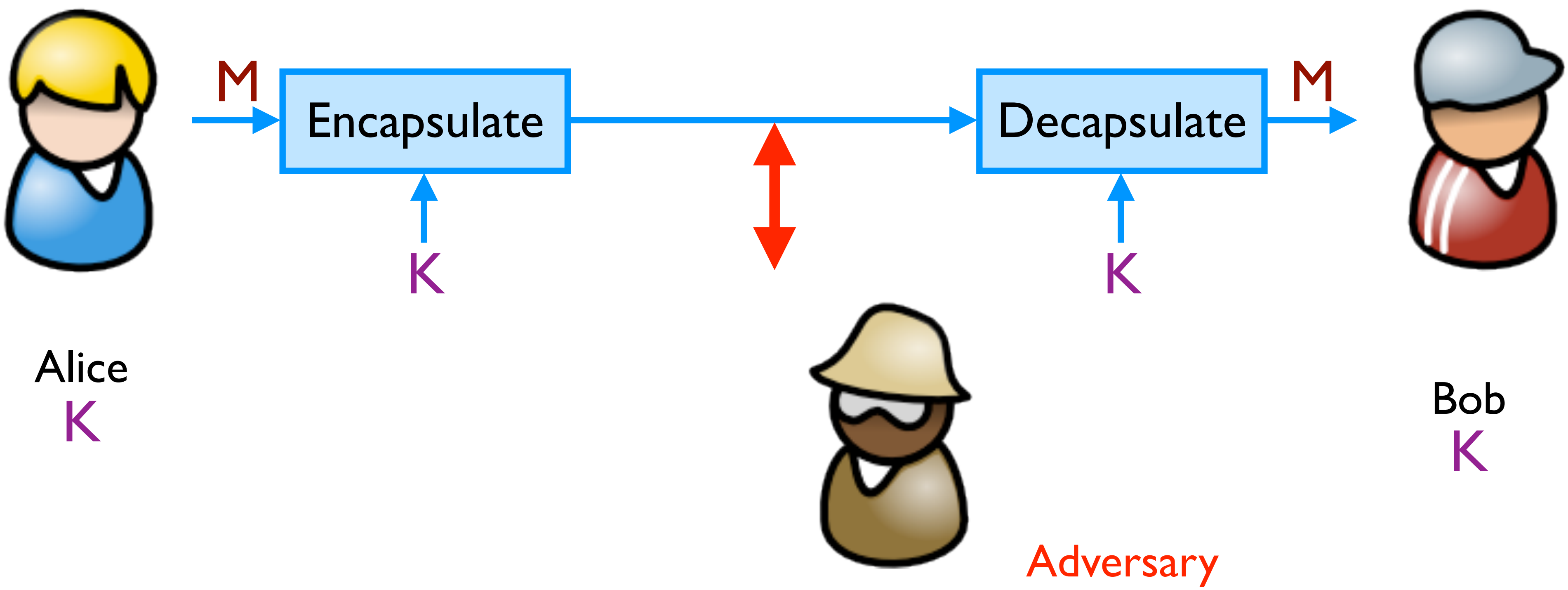
Privacy of data
Prevent exposure of information

Integrity of data
Prevent modification of information



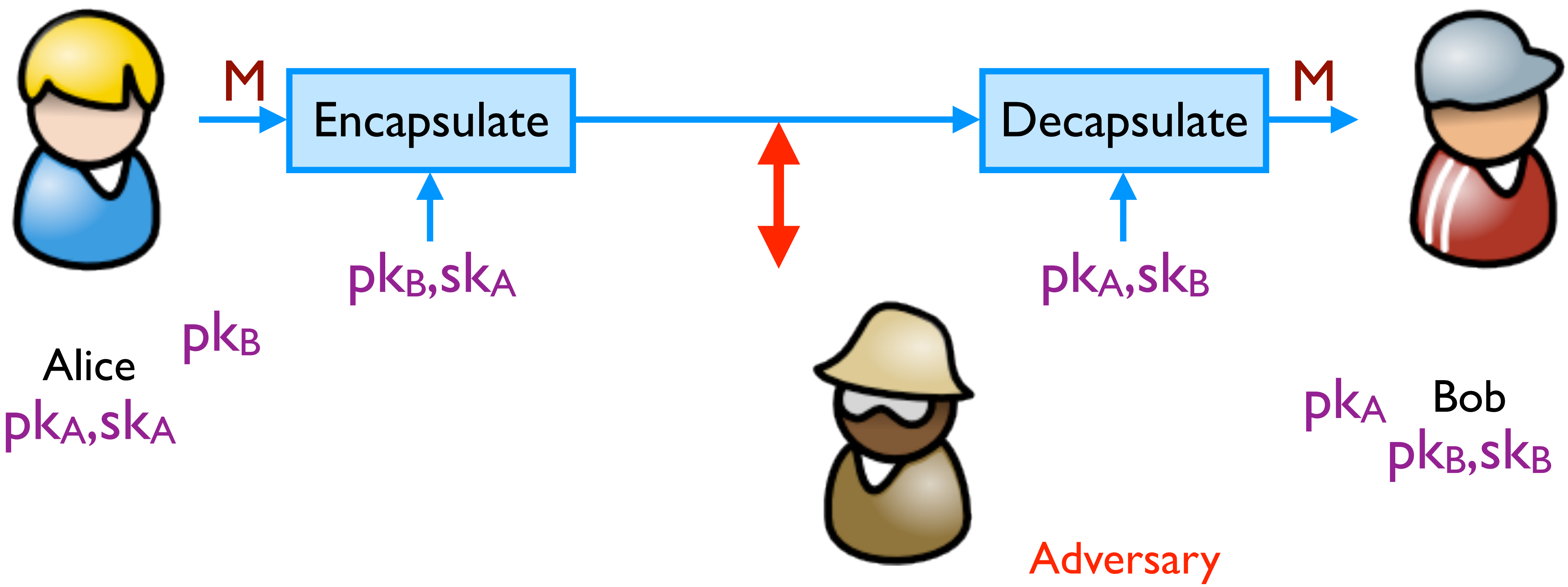
Symmetric Setting

Both communicating parties have access to a shared random string K, called the key.



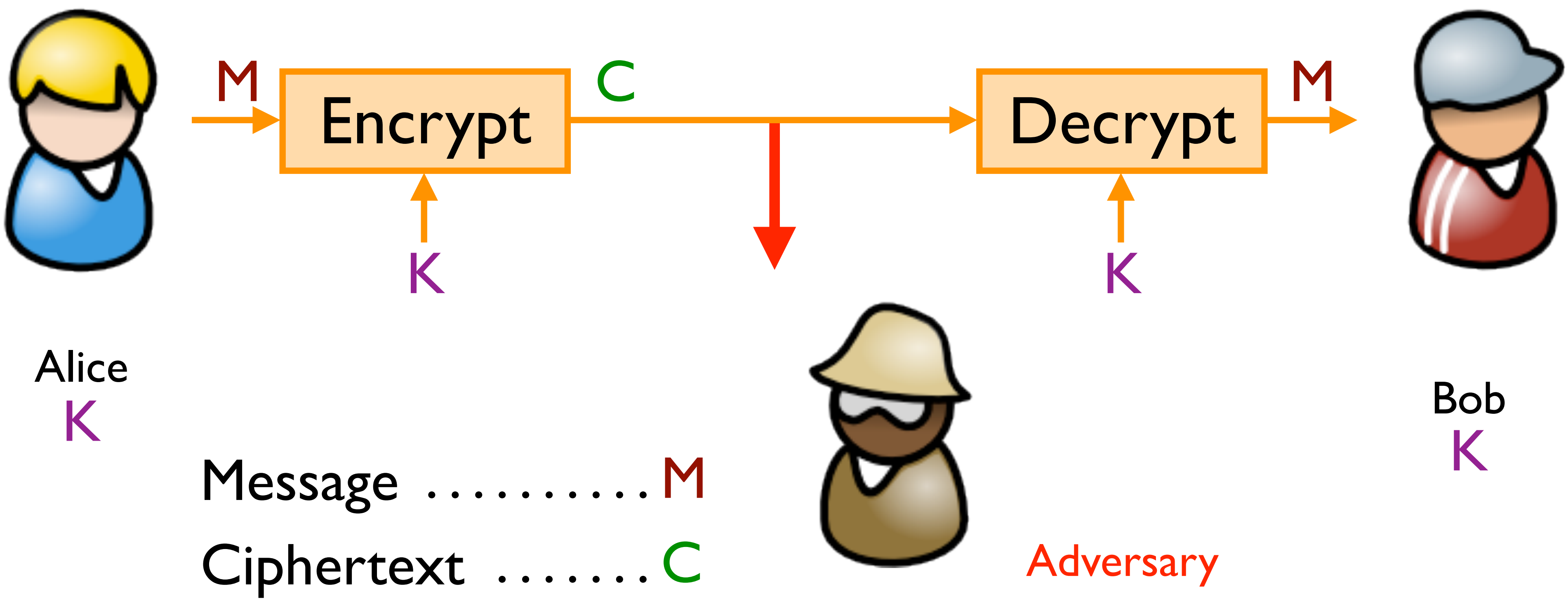
Asymmetric Setting

Each party creates a public key pk and a secret key sk.



Achieving Privacy (Symmetric)

Encryption schemes: A tool for protecting privacy.

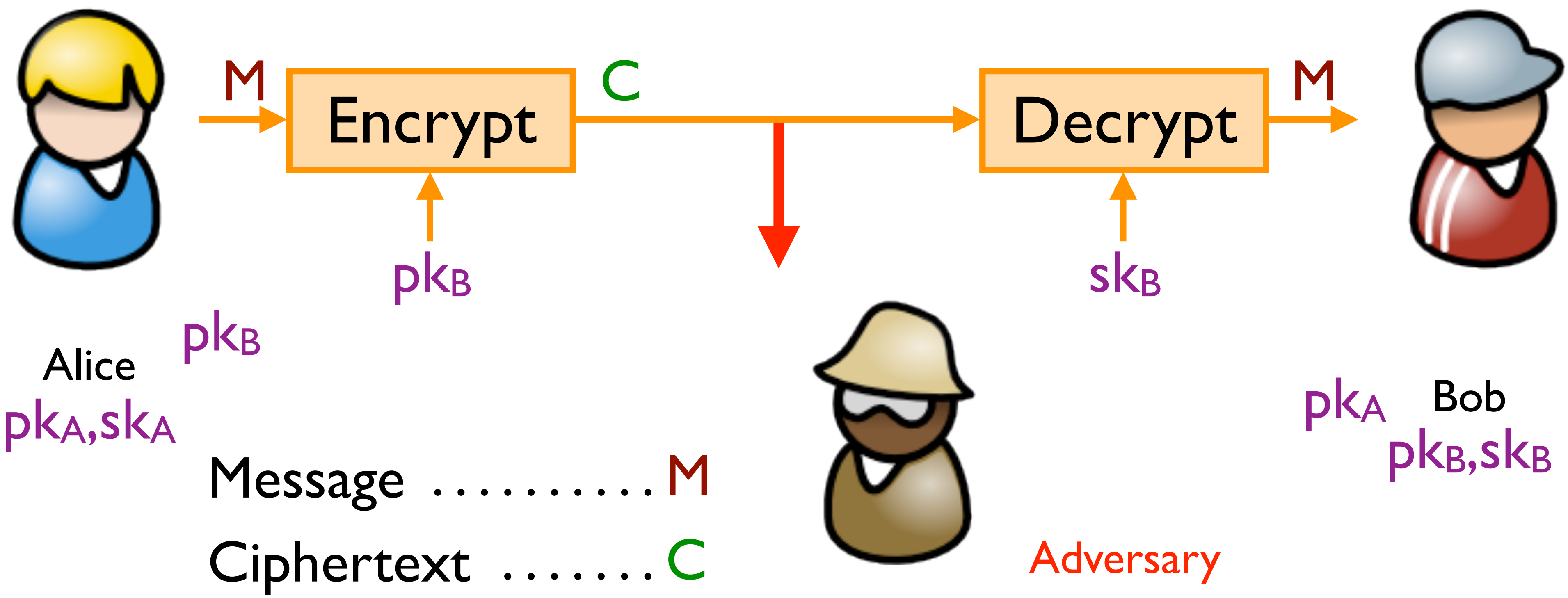


Message . . . . M

Ciphertext C

Achieving Privacy (Asymmetric)

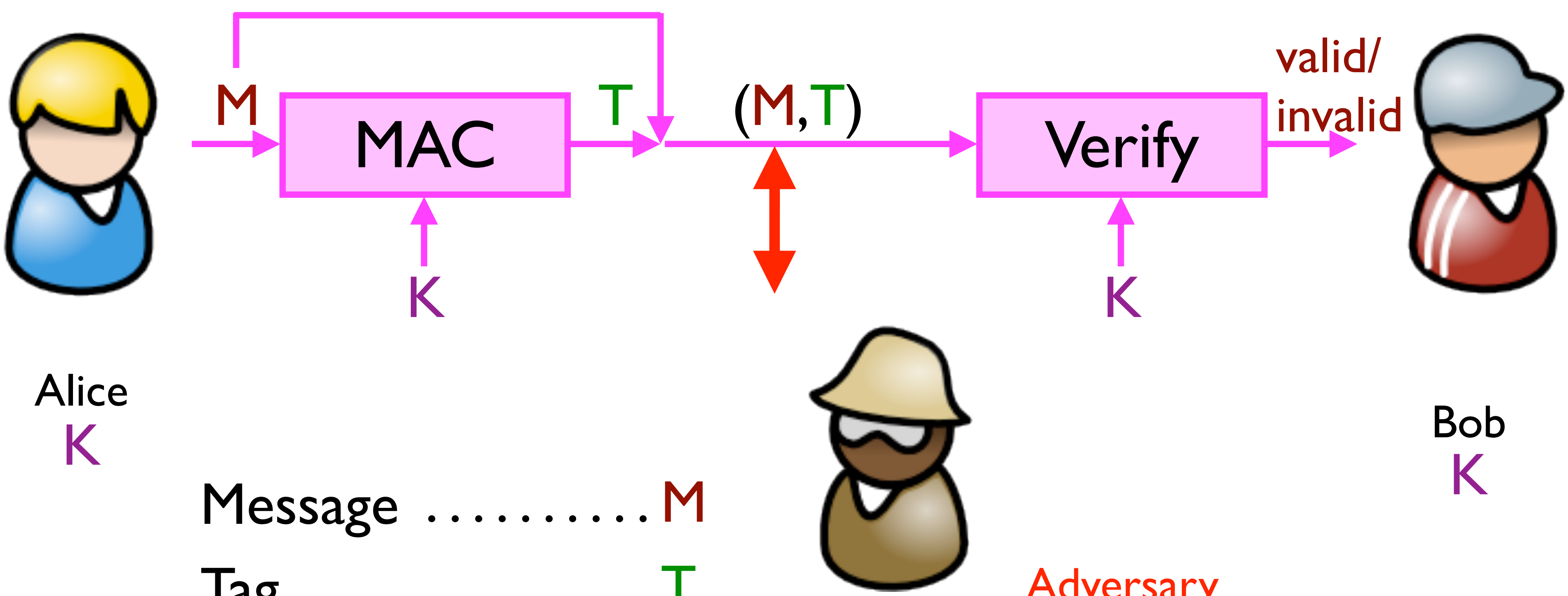
Encryption schemes: A tool for protecting privacy.



Achieving Integrity (Symmetric)

Message authentication schemes: A tool for protecting integrity.

(Also called message authentication codes or MACs.)

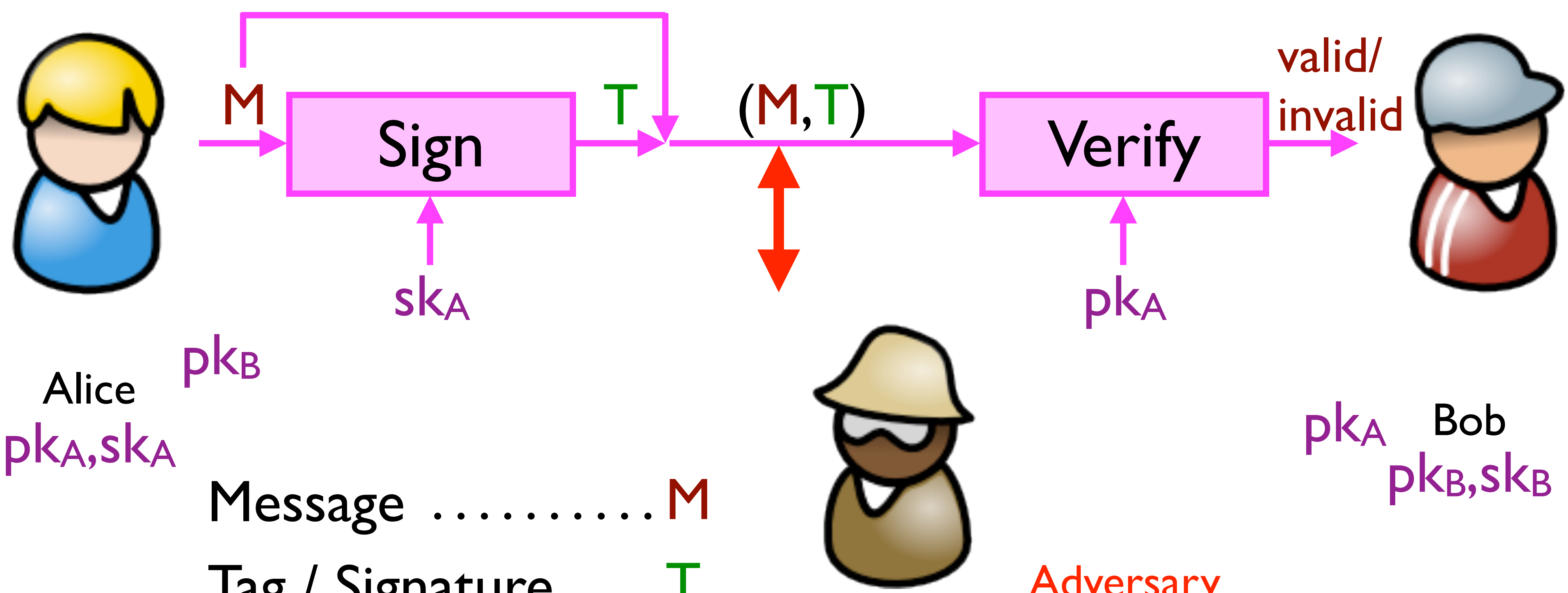


Message . . . . M

Tag . . . . T

Achieving Integrity (Asymmetric)

Digital signature schemes: A tool for protecting integrity and authenticity.

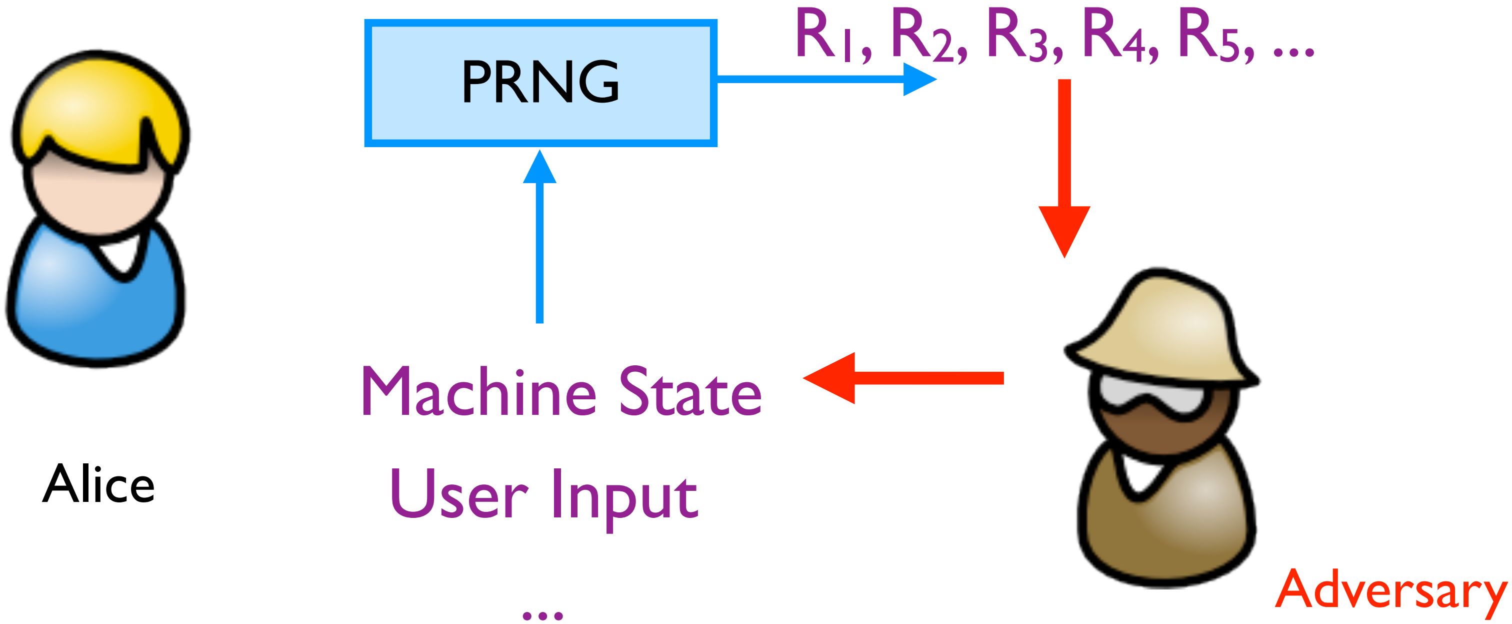


Message . . . . M

Tag / Signature . . . . T

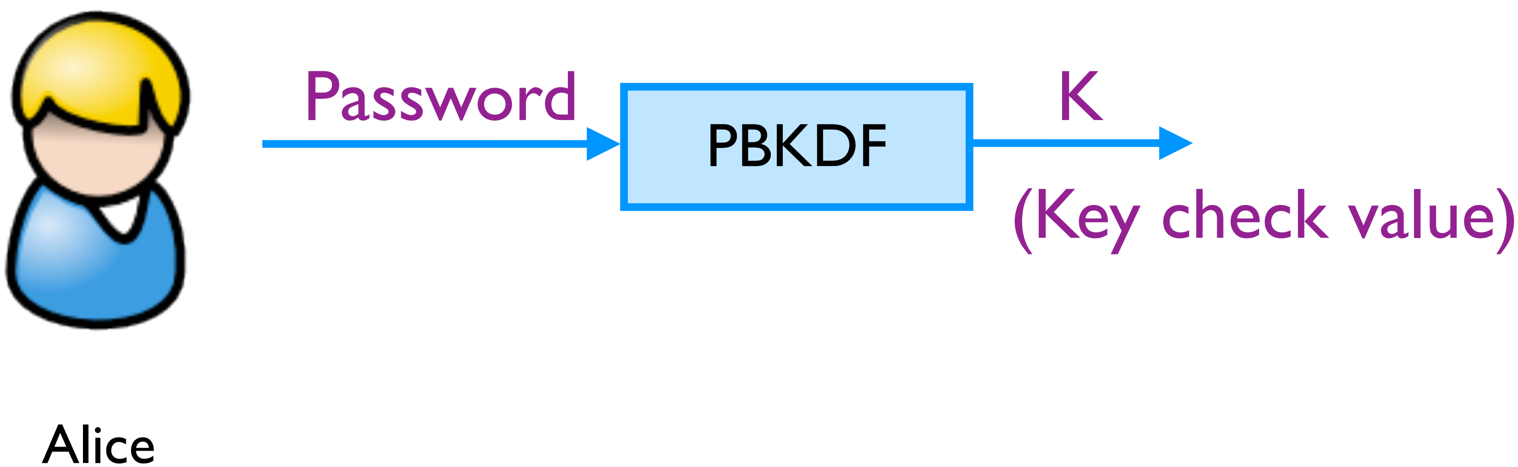
"Random" Numbers

Pseudorandom Number Generators (PRNGs)



Getting keys: PBKDF

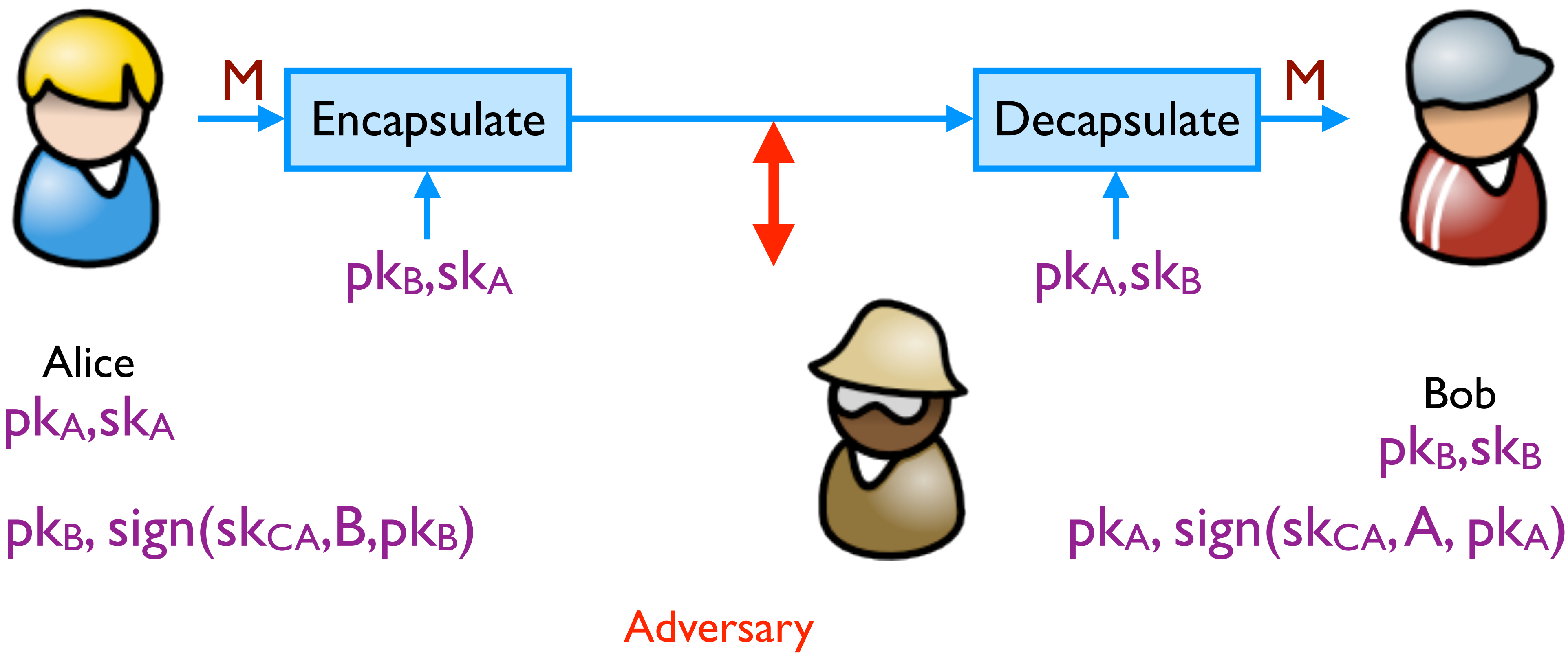
Password-based Key Derivation Functions



Getting keys: CAs

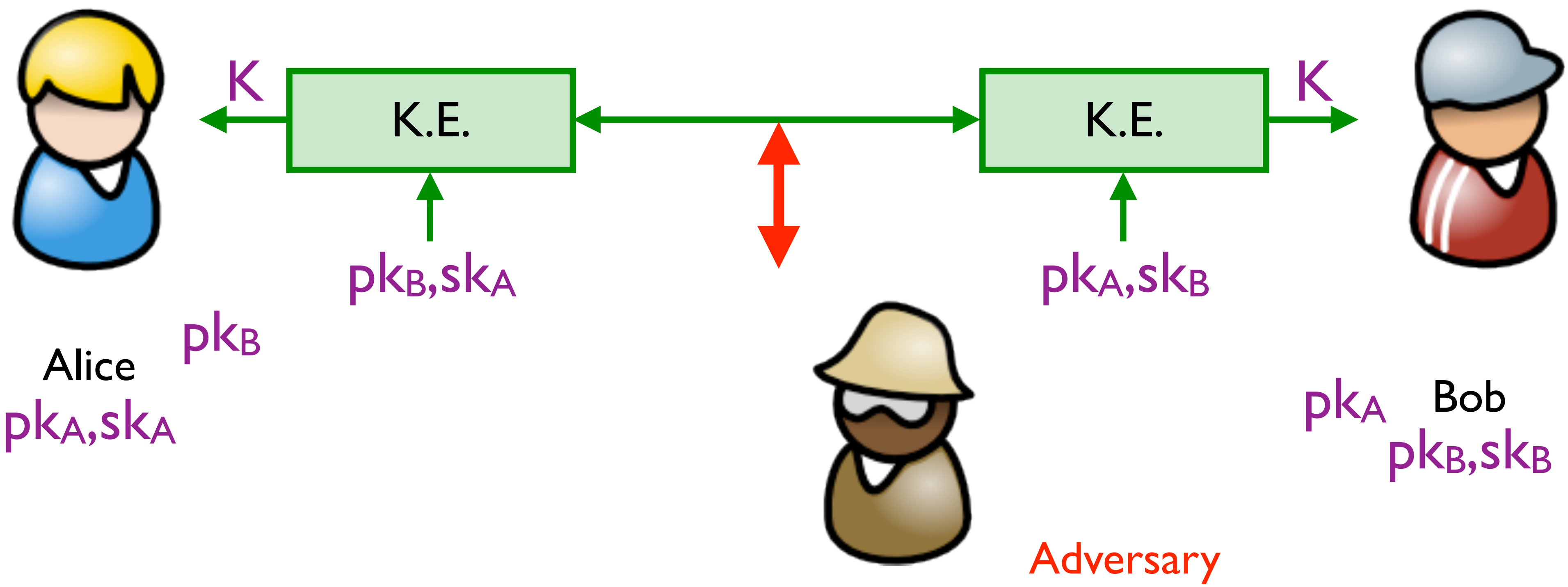
Each party creates a public key pk and a secret key sk.

(Public keys signed by a trusted third party: a certificate authority.)



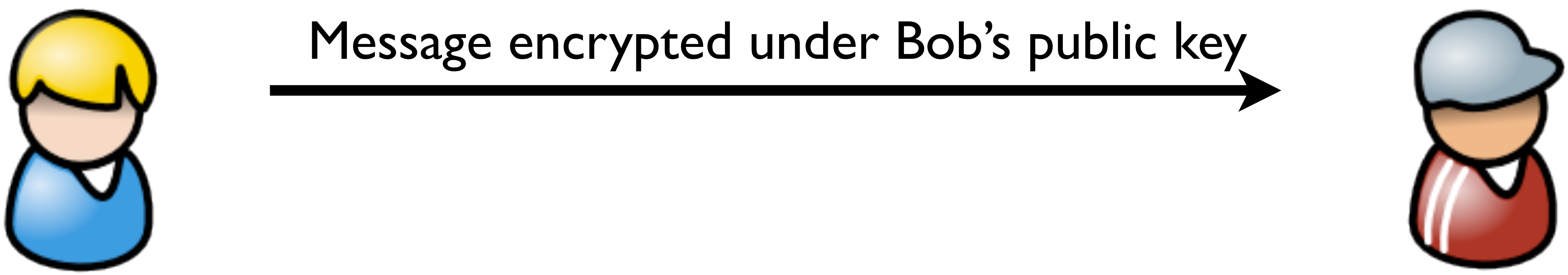
Getting keys: Key exchange

Key exchange protocols: A tool for establishing a shared symmetric key from public keys



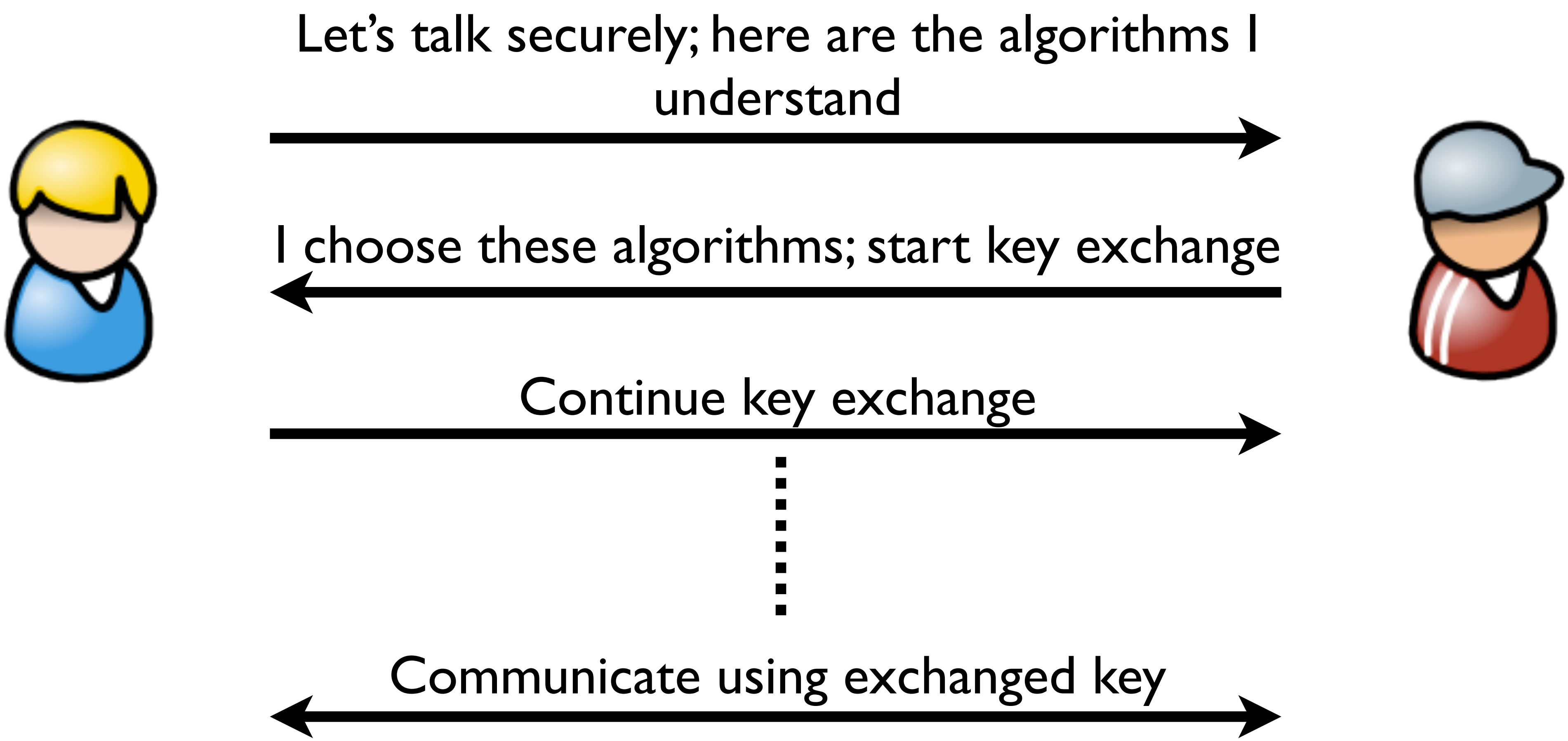
One-way Communications

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
```

VICTORY`ABDEFGHJKLMNPQSUWXZ`

- "state of the art" for thousands of years

Kerckhoff’s Principle

- Security of a cryptographic object should depend **only** on the secrecy of the secret (private) key
- Security should not depend on the secrecy of the algorithm itself.

Mid-way Summary

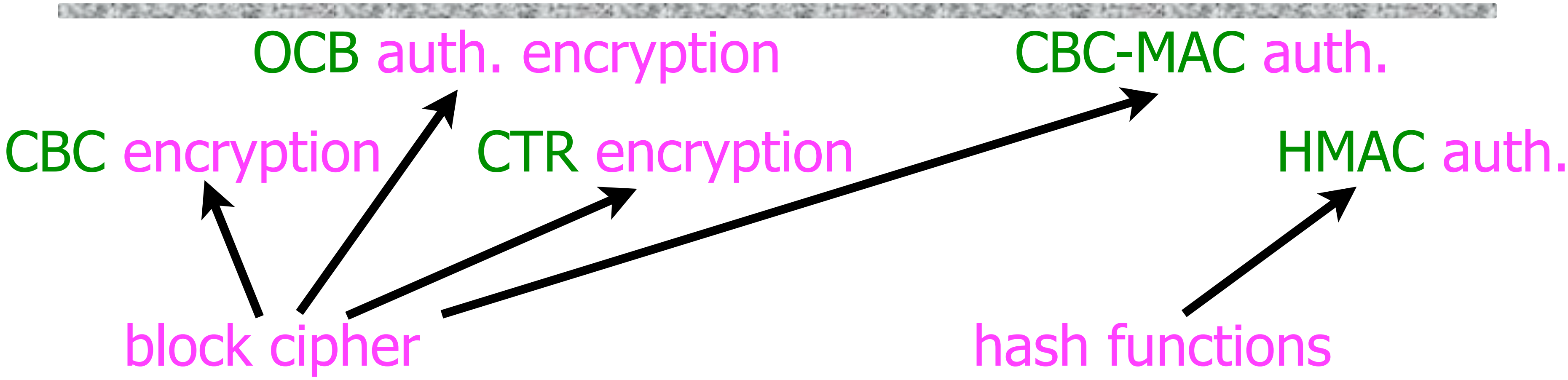
- **Symmetric cryptography**
  - Both sides know ***shared key***, no one else knows anything. Can ***encrypt, decrypt, sign/MAC, verify***
  - Computationally lightweight
  - ***Challenge:*** How do you privately share a key?
- **Asymmetric cryptography**
  - Everyone has a ***public*** *key* that everyone else knows; and a paired ***secret*** *key* that is private
  - Public key can ***encrypt***; only secret key can ***decrypt***
  - Secret key can ***sign/MAC***, public key can ***verify***
  - Computationally expensive
  - ***Challenge:*** How do you validate a public key?

Mid-way Summary

- **Where are public keys from?**
  - One solution: keys for ***Certificate Authorities*** *a priori* known by browser, OS, etc.
- **Where are shared keys from?**
  - In person exchange, snail mail, etc.
  - If we have verifiable public/private keys: ***key exchange*** protocol generates a shared key for symmetric cryptography

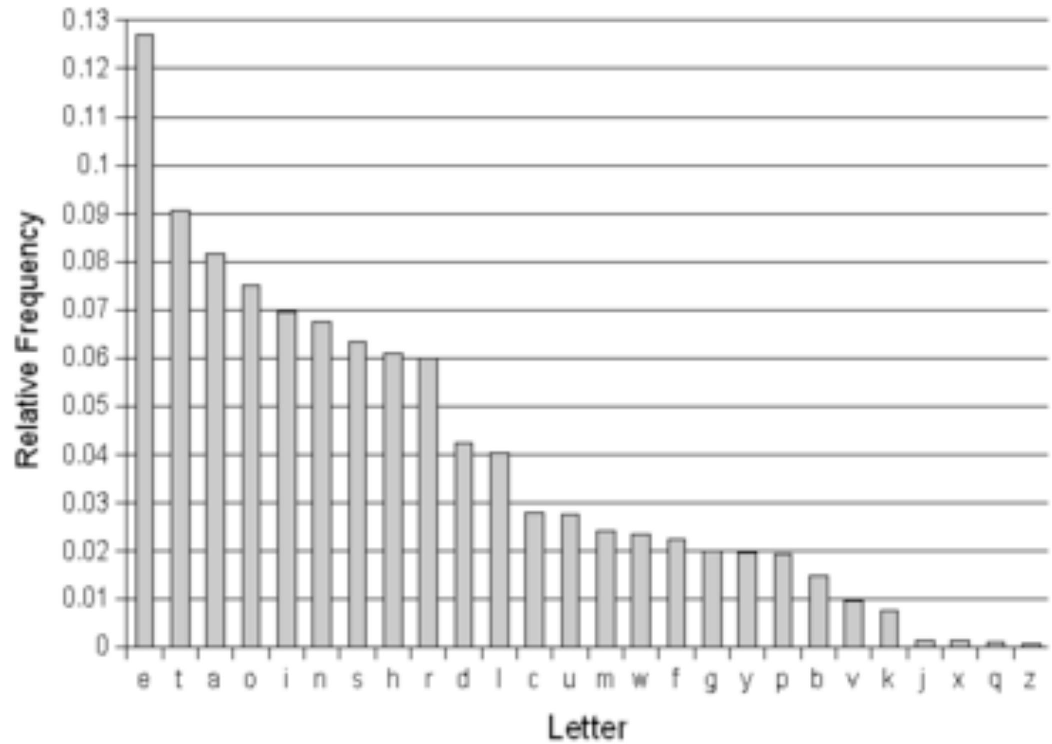
How cryptosystems work today

- 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)



“Old Days” 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%



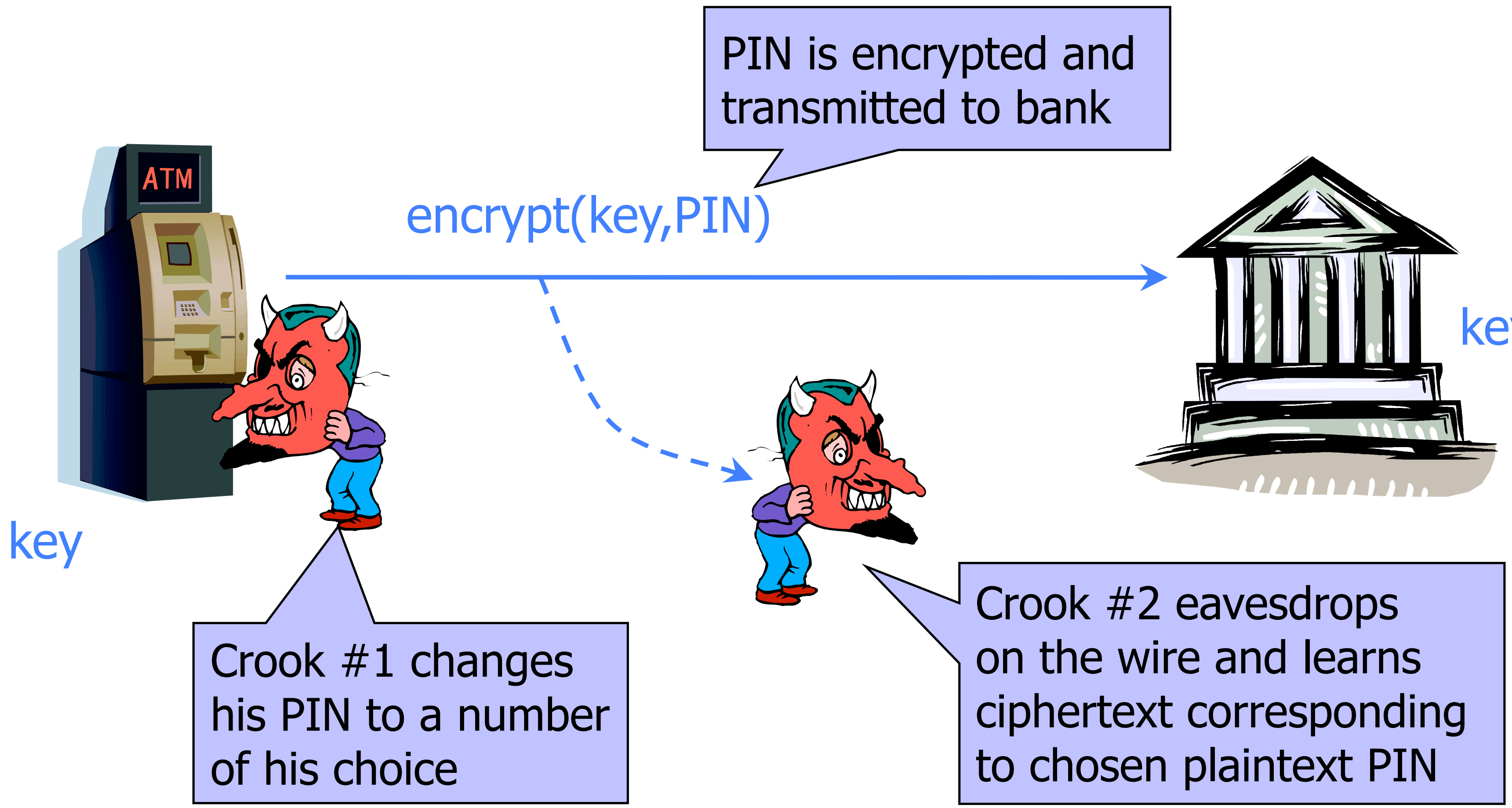
From http://en.wikipedia.org/wiki/Letter_frequencies

Attack Scenarios for Encryption

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

- (General advice: Target strongest level of privacy possible -- even if not clear why -- for extra "safety")

Chosen-Plaintext Attack

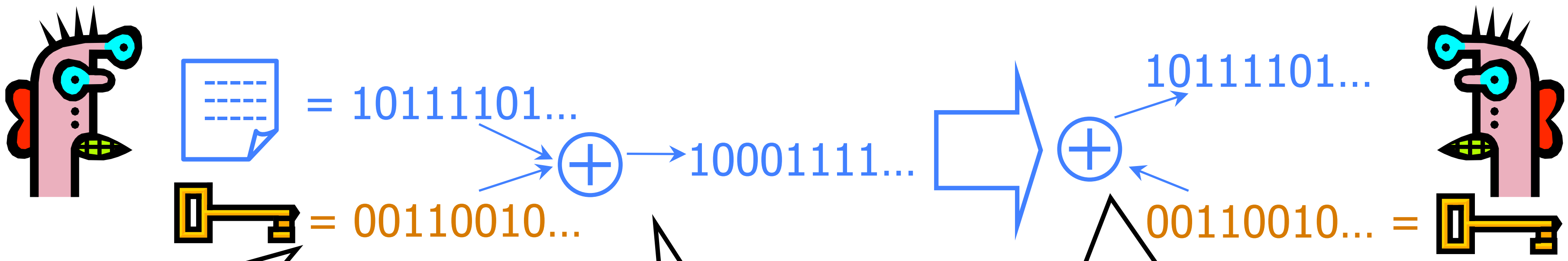


… repeat for any PIN value

Attack Scenarios for Integrity

- What do you think these scenarios should be?

One-Time Pad



Key is a random bit sequence as long as the plaintext

Encrypt by bitwise XOR of plaintext and key:
ciphertext = plaintext ⊕ key

Decrypt by bitwise XOR of ciphertext and key:
ciphertext ⊕ key =
(plaintext ⊕ key) ⊕ key =
plaintext ⊕ (key ⊕ key) =
plaintext

Advantages of One-Time Pad

- Easy to compute
  - Encryption and decryption are the same operation
  - Bitwise XOR is very cheap to compute
- As secure as theoretically possible
  - Given a ciphertext, all plaintexts are equally likely, regardless of attacker's computational resources
  - …as long as the key sequence is truly random
    - True randomness is expensive to obtain in large quantities
  - …as long as each key is same length as plaintext
    - But how does the sender communicate the key to receiver?