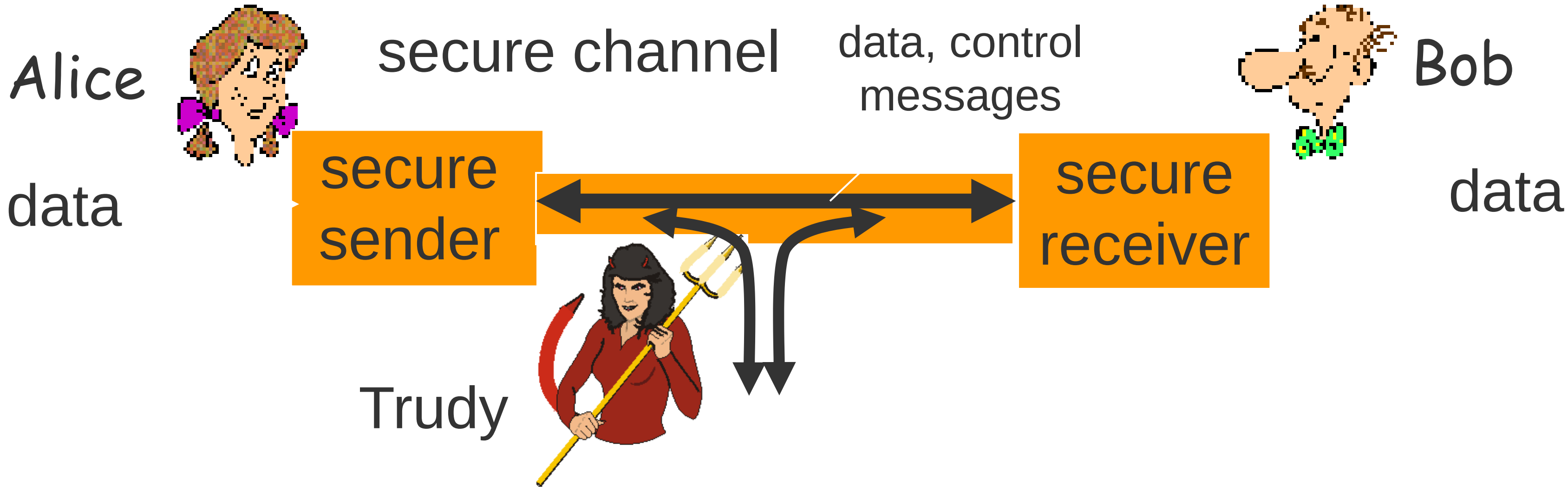
Network Security

Andy Carpenter

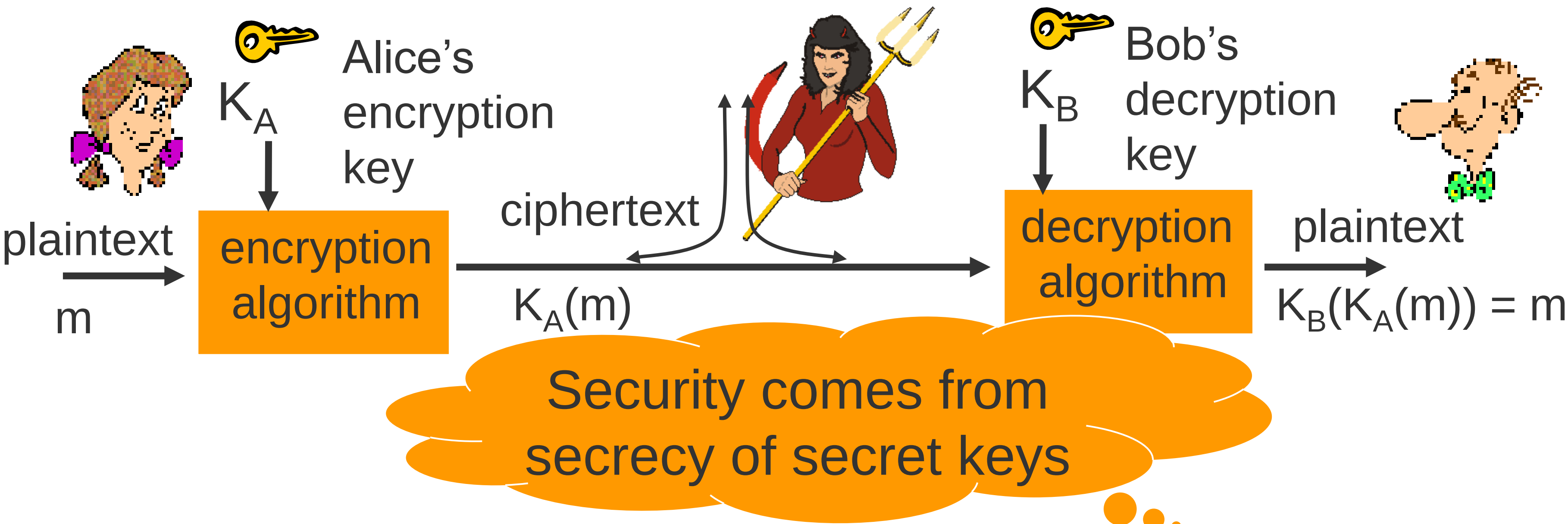
(Andy.Carpenter@manchester.ac.uk)

K&R: 8.1, P&D: 8

Network Security is What?



Security: Implementation



- Involves encrypting, and possibly, decrypting data
- Done by cryptographic algorithms that use keys
- Algorithms are well known, keys are unique
- Users referred to as participants or principles

K&R: 8.2.1

Encryption: Simple Scheme

- Cryptography is substituting one thing for another
- Monoalphabetic (one letter for another) cipher:

```
plaintext:   abcdefghijklmnopqrstuvwxyz

ciphertext:  mnbvcxzasdfghjklpoiuytrewq
```

E.g.:

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

- Q: How hard to break this simple cipher?:
  - brute force (how hard?)
  - other?

K&R: 8.2.1

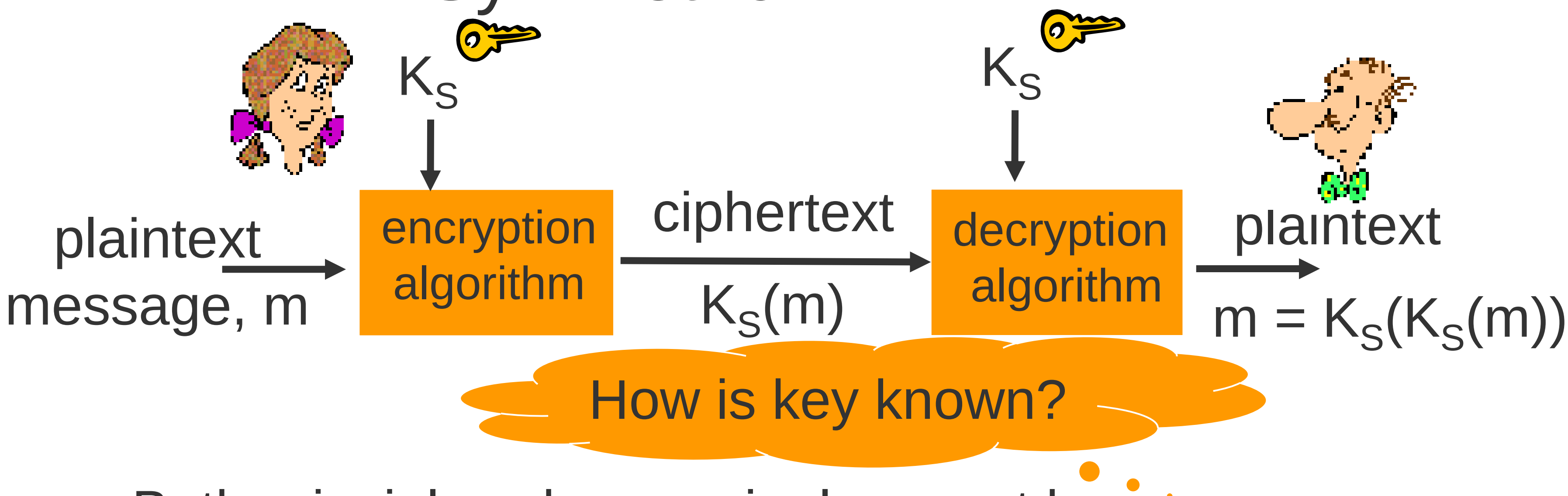
Encryption: Breaking it

- **Cipher-text only attack**: two approaches:
  - Search through all keys: for each try
    - must distinguish plaintext from gibberish
  - Statistical analysis
- **Known-plaintext attack**: Trudy has some plaintext corresponding to some ciphertext
  - e.g. in monoalphabetic cipher
  - Trudy determines pairings for a,l,i,c,e,b,o,
- **Chosen-plaintext attack**:
  - get the cyphertext for some chosen plaintext

Minimise use of keys

Cryptographic Algorithms: Symmetric

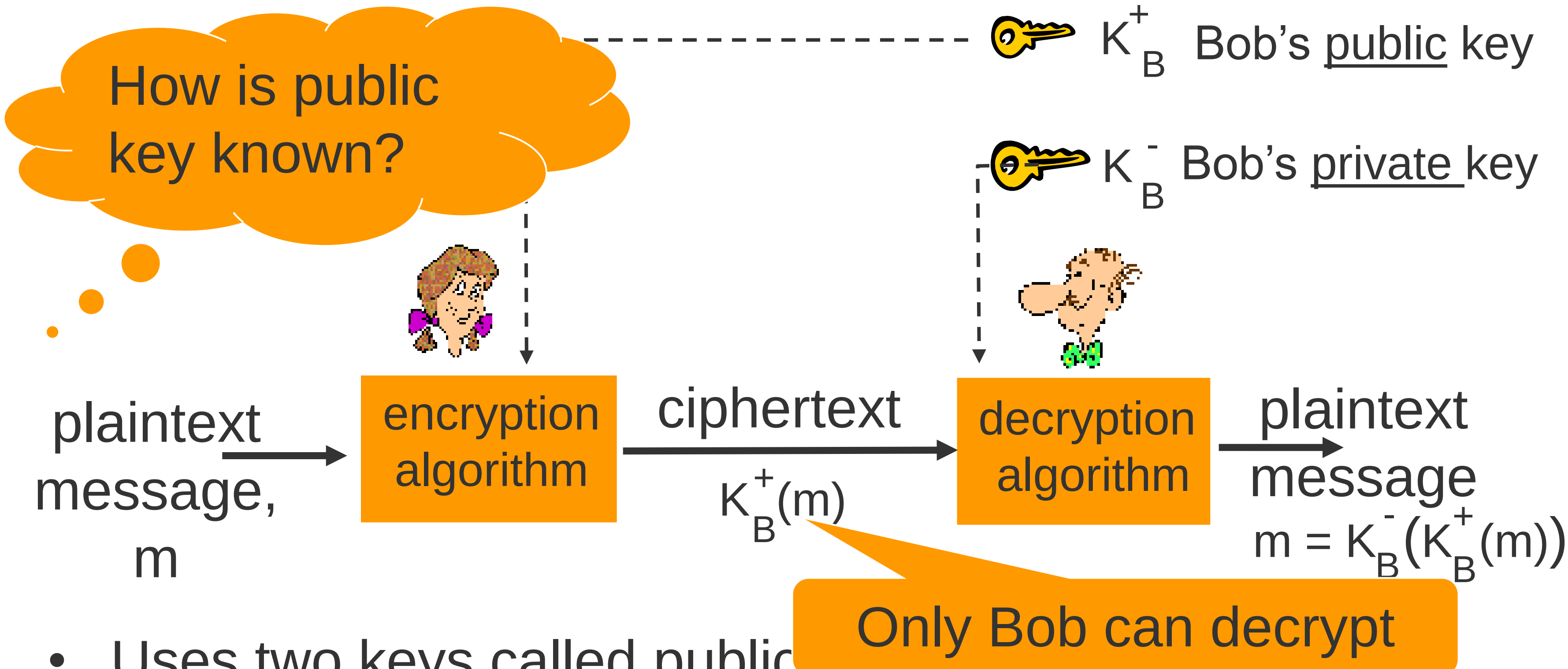
K&R: 8.2.1, P&D: 8.1



- Both principles share a single secret key
- Same key used to encrypt and decrypt data
- Examples:
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)

Cryptographic Algorithms: Public Key

K&R: 8.2.2, P&D: 8.1



- Uses two keys called public
- Encrypt using public key, decrypt using private key
- Example: Rivest, Shamir and Adleman (RSA)

Ciphers: RSA Property

- The following property will be *very* useful later:

$$K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$$

use public key first, followed by private key

use private key first, followed by public key

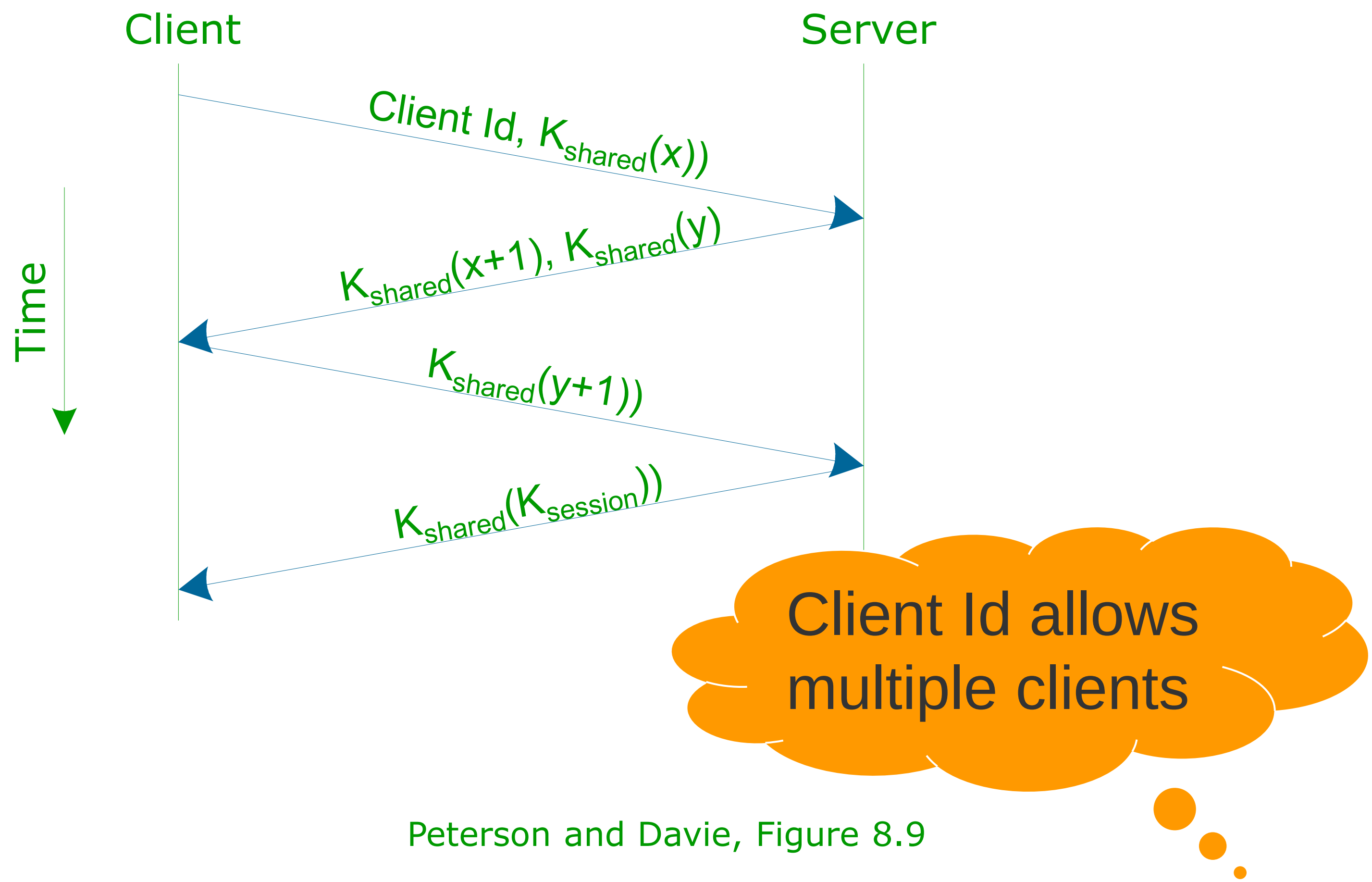
Result is the same!

Security Mechanisms

- Algorithms are only elements in network security
- Need mechanisms and protocols for specific tasks:
  - authentication of remote users
  - ensuring where data comes from
  - distributing keys
- Exponentiation is computationally intensive
  - DES is at least 100 times faster than RSA
- Public/private keys used to authenticate and securely exchange a shared symmetric key K_S
- Once have K_S, use symmetric key cryptography
- Good practice minimises the use of individual keys

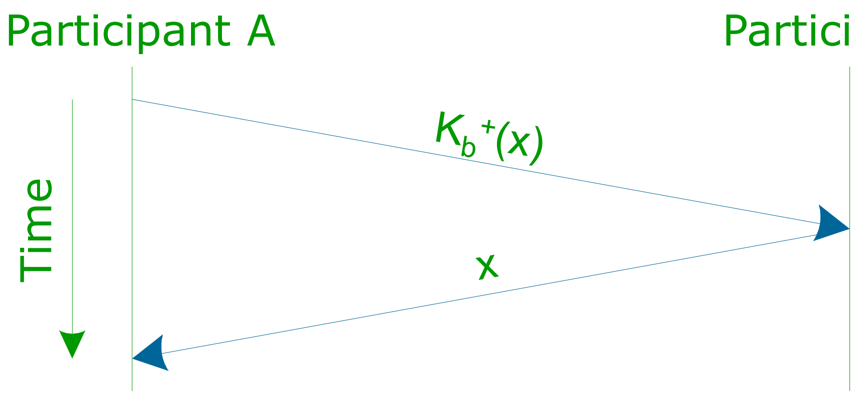
Authentication: Three-Way Handshake

- Assumes two participants share secret key, *k*



Peterson and Davie, Figure 8.9

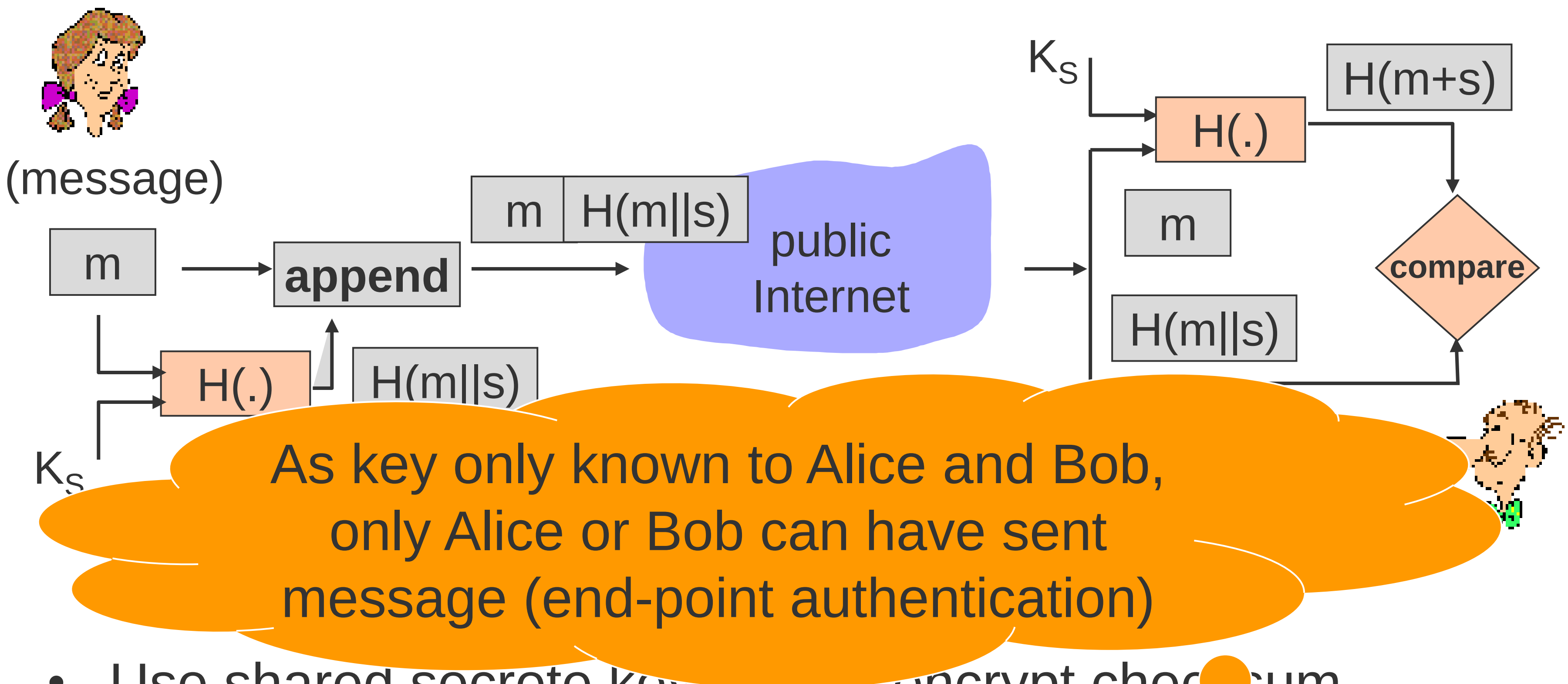
Authentication: Public Key



Peterson and Davie, Figure 8.11

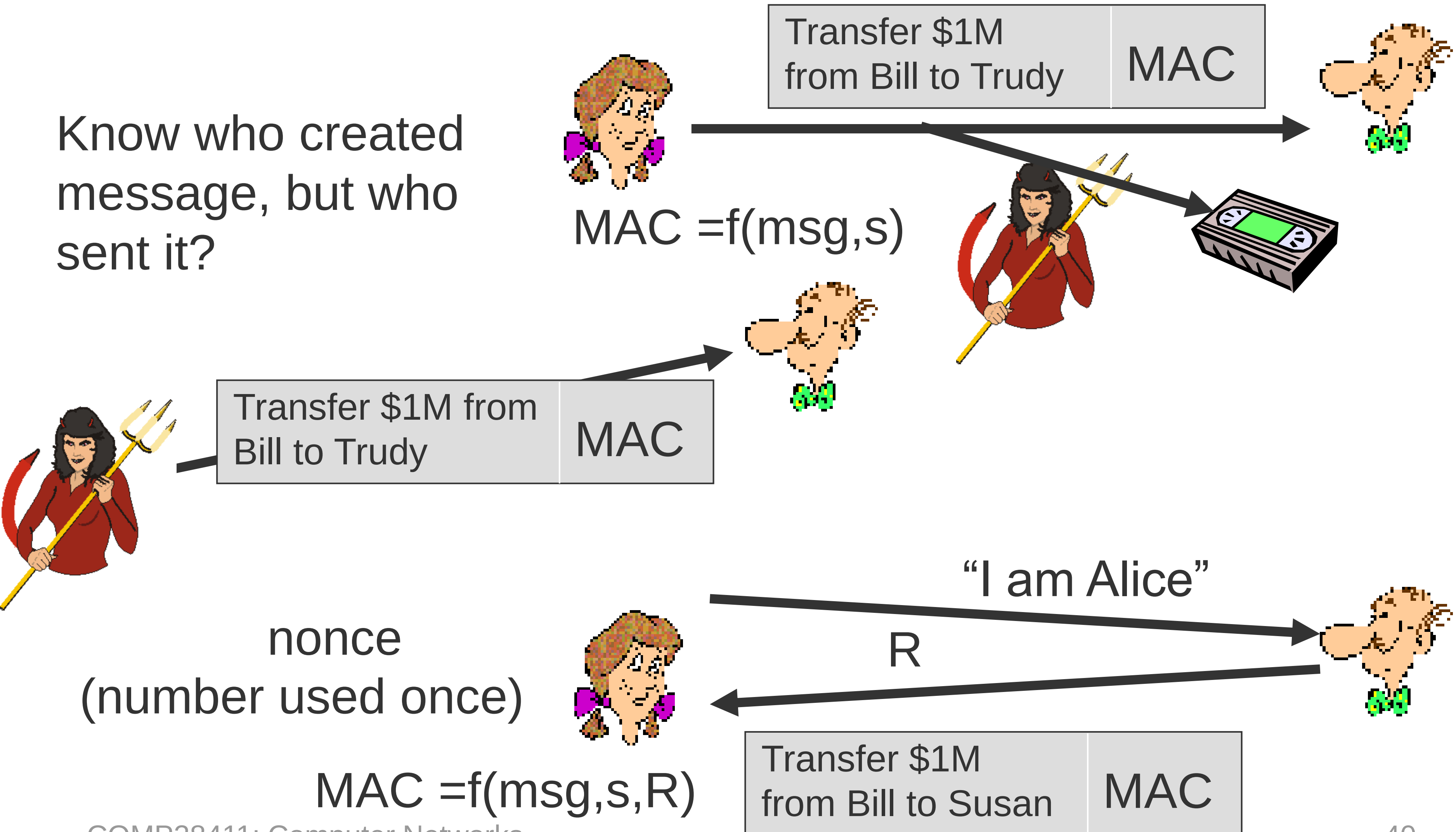
- A encrypts random number, *x*, using B's public key
- B proves knows corresponding private key by:
  - decrypting *x* and returning it to A
- Only authenticates B to A, reverse process for A to B

Message Integrity – Keyed Hash



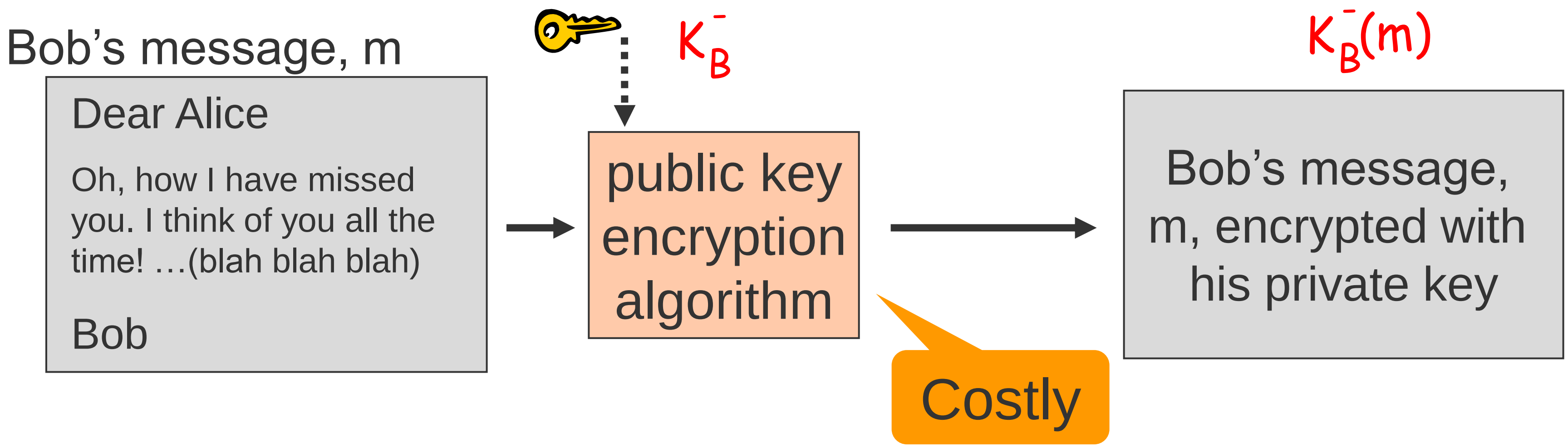
- Use shared secrete key, KS, to encrypt checksum
- Checksum = Message Authentication Code (MAC)
- Example: HMAC

Playback Attack and Defence



Message Integrity: Signature

- Message (encrypted) with Bob’s private key
  - only Bob can have sent (non-repudiation)



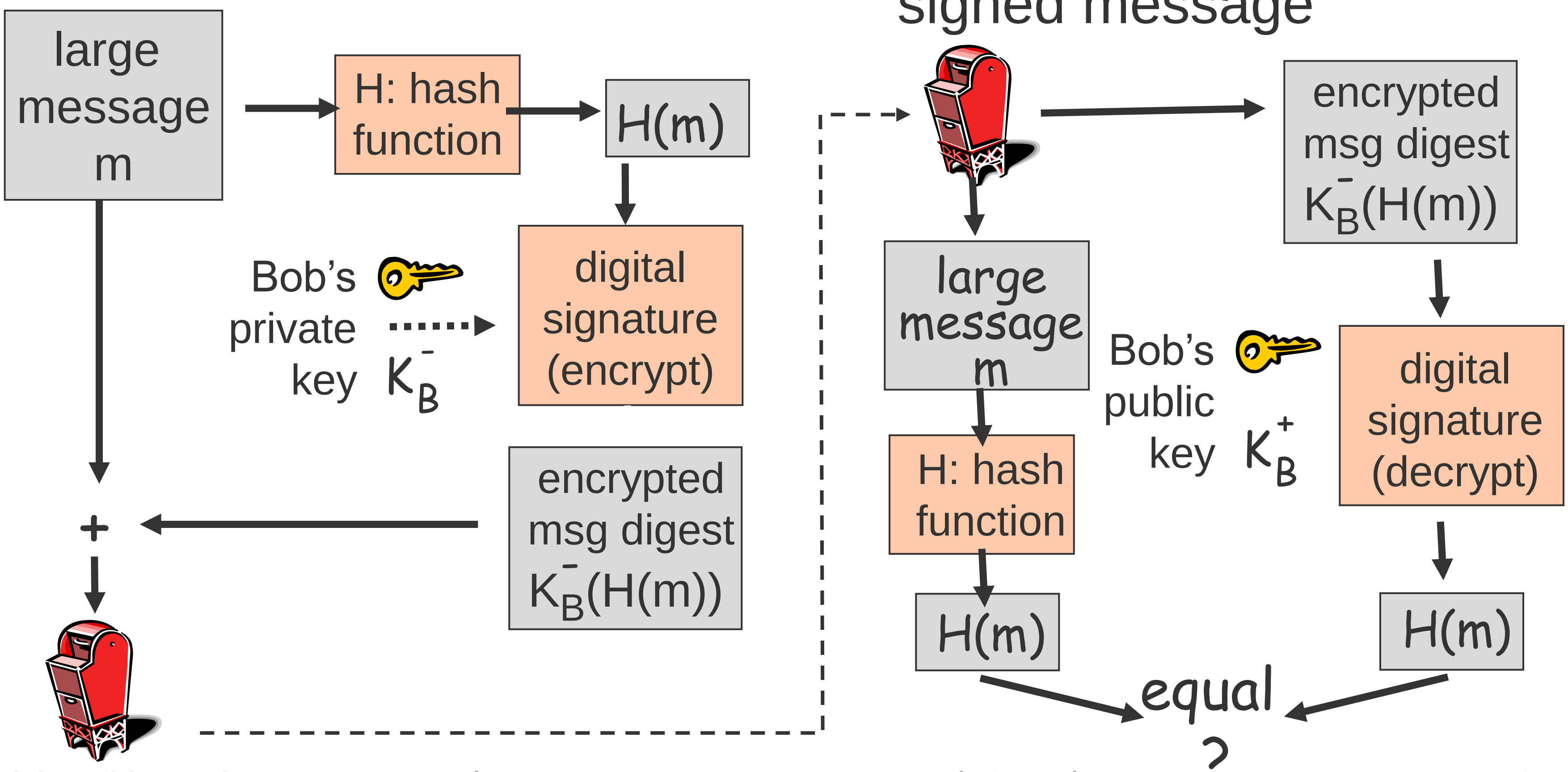
- Anyone can decrypt/verify sender

Note: $m = K^-_B\,(K^+_B(m)\,) = K^+_B\,(K^-_B(m)\,)$

Message Integrity – Digital Signatures

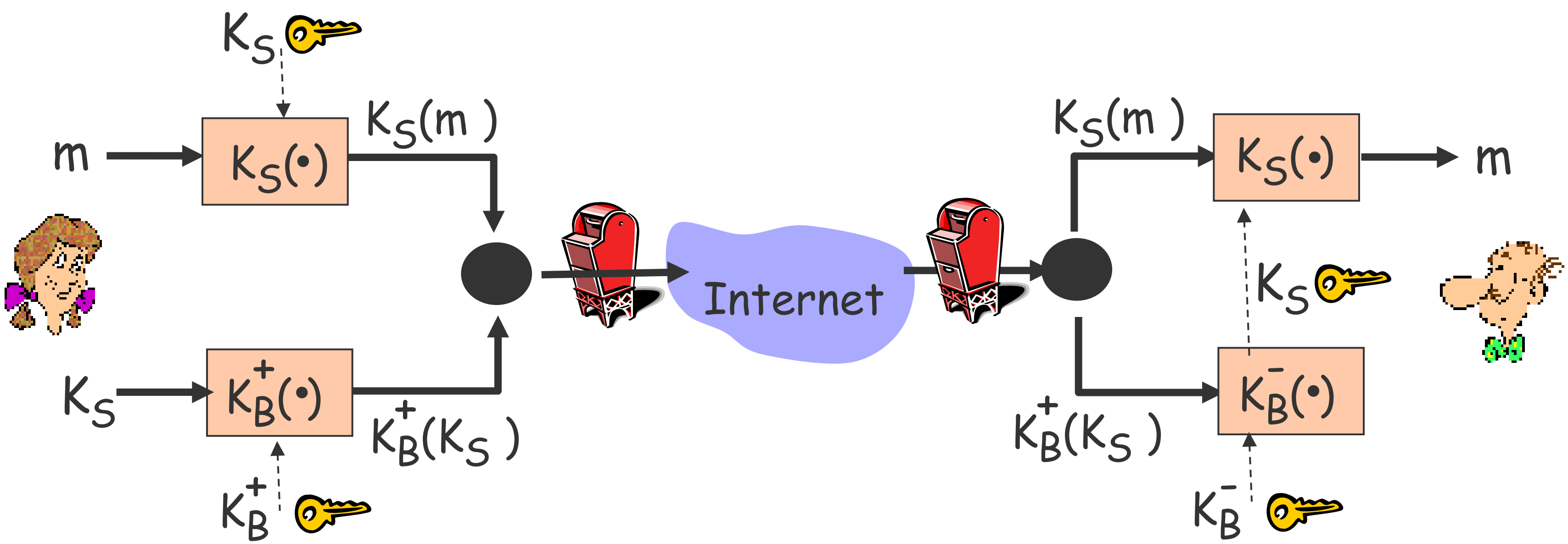
Bob sends digitally signed message

Alice verifies signature and integrity of digitally signed message



Secure e-mail (Confidentiality)

- Alice wants to send confidential e-mail, m, to Bob.

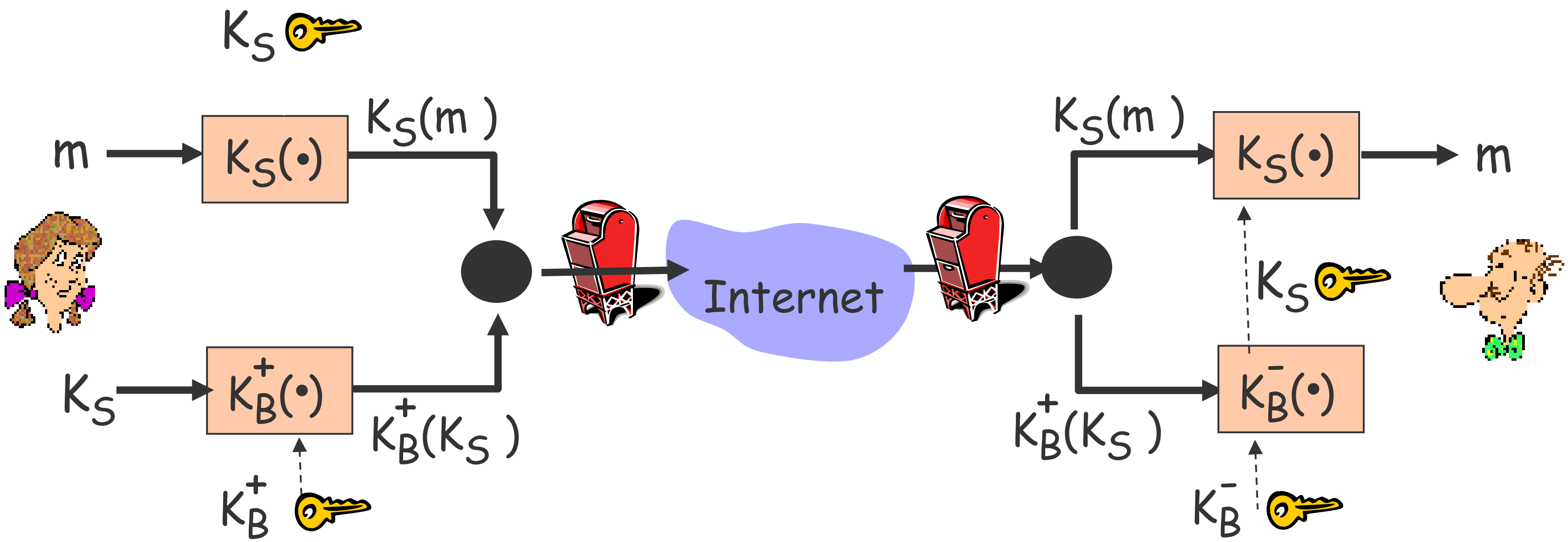


Alice:

- generates random *symmetric* private key, K_S.
- encrypts message with K_S (for efficiency)
- also encrypts K_S with Bob's public key.
- sends both $K_S(m)$ and $K_B^+(K_S)$ to Bob.

Secure e-mail (Confidentiality)

- Alice wants to send <u>confidential</u> e-mail, m, to Bob.

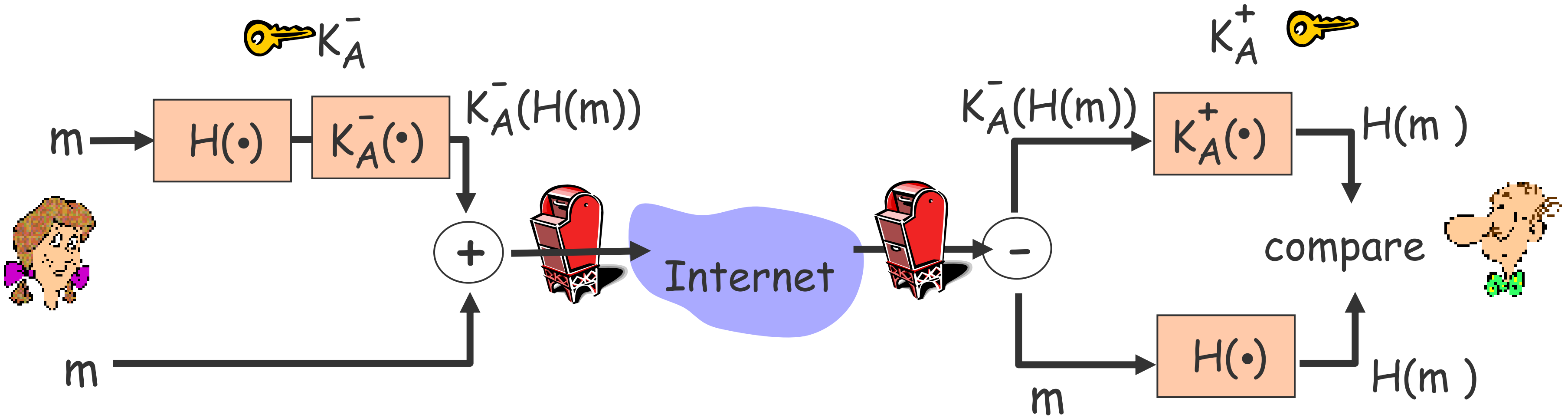


Bob:

- uses his private key to decrypt and recover K_S
- uses K_S to decrypt $K_S(m)$ to recover m

Secure e-mail (Sender Authentication)

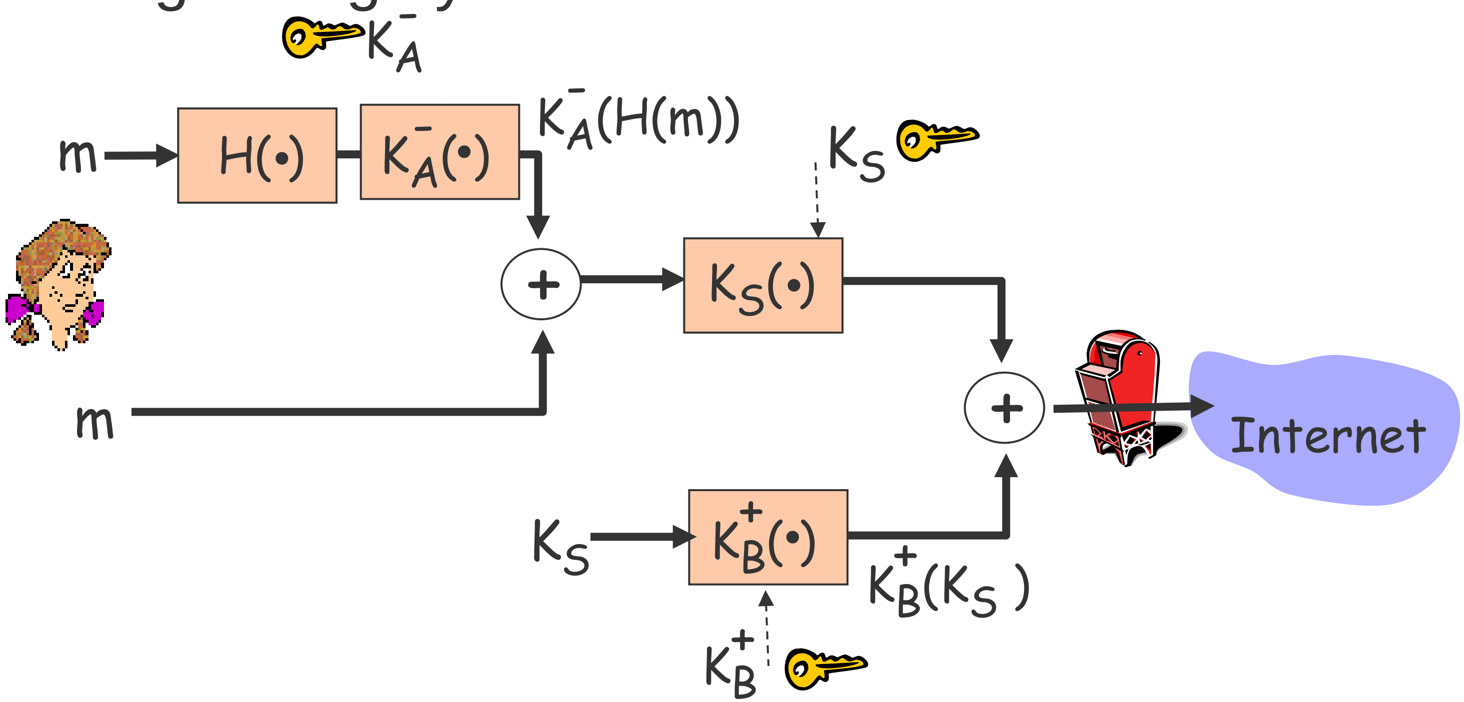
- Alice wants to provide sender authentication message integrity.



- Alice digitally signs message.
- sends both message (in the clear) and digital signature.

Secure e-mail (Sender Authentication)

- Alice wants to provide secrecy, sender authentication, message integrity.



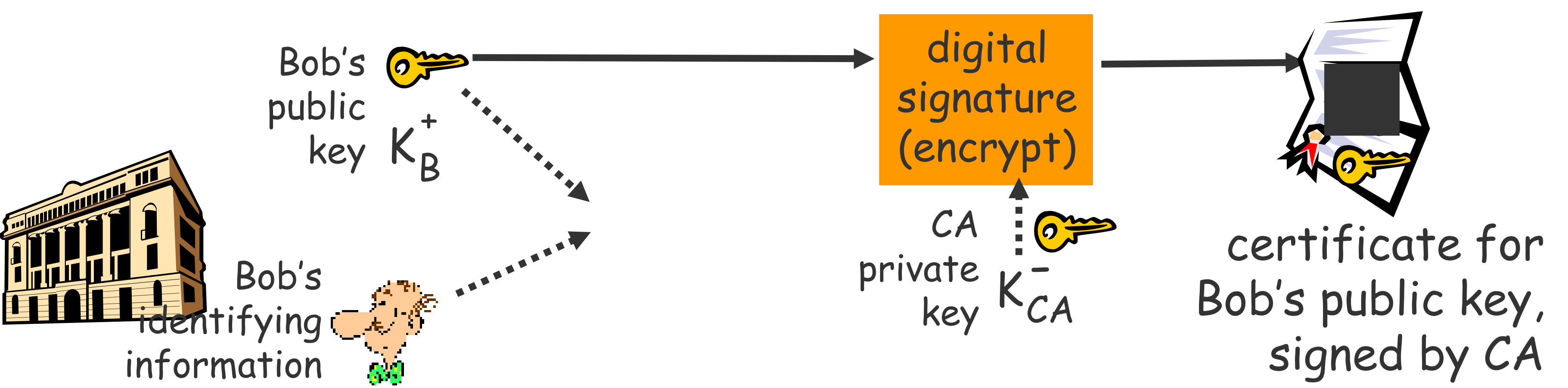
Alice uses three keys:

Public Key Distribution

- Cryptography depends on knowing public keys
- Sending public key without modification protection means
  - no confirmation that belongs to claimed owner
- But, modification protection requires a key …
- Reduce magnitude of problem using digital certificates
- Aspects:
  - using digital certificates to verify public keys
  - building “chains of trust” using certificates
  - structure/content of certificates (X.509 standard)
  - how certificates are cancelled (revoked)

Certification Authorities

- Certification authority (CA): binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
  - E provides "proof of identity" to CA.
  - CA creates certificate binding E to its public key.
  - certificate containing E's public key digitally signed by CA – CA says "this is E's public key"

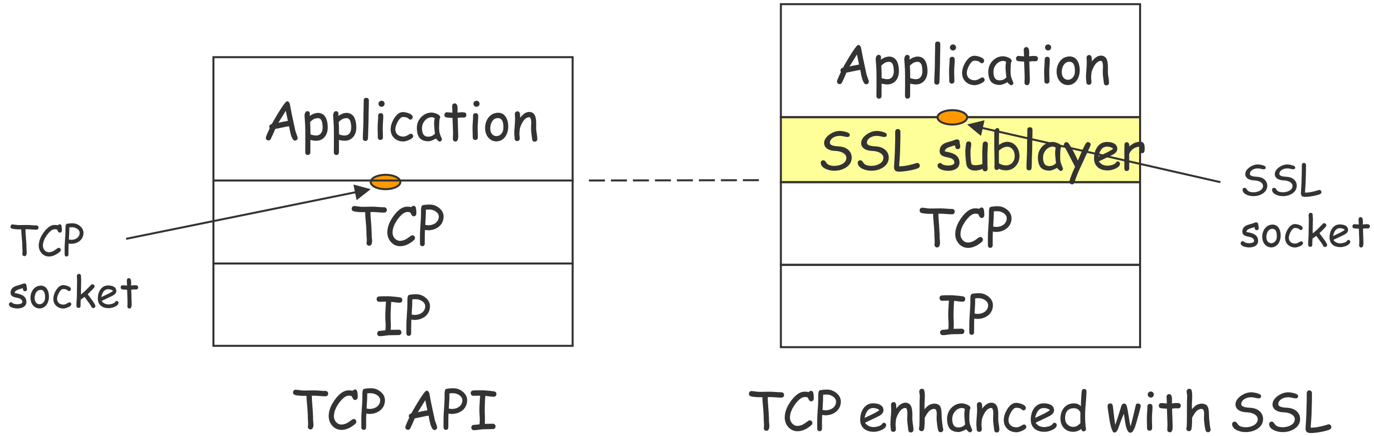


Implementing Network Security

- Implemented various levels of network
- Application, e.g. PGP, SSH
  - provides application-to-application security
  - each application must implement its own security
- Transport, e.g. TLS/SSL
  - provides application-to-application security
  - single implementation for all applications
- Network, e.g. IPSEC
  - used to build complete secure networks

Transport Layer Security (TLS)/SSL

- Transport protocol with built-in security mechanisms
- Provides security to any TCP-based application
  - e.g., e-commerce via web (shttp)
- Security services:
  - server authentication, data encryption
  - client authentication (optional)



Summary

- Keystone of security is encryption
- For authentication public-key algorithms are used
- Once authorised, participants use shared (session) key
- Session keys are used to implement privacy
- Core is mechanism used to distribute public keys
- Elements now used to build secure Internet applications
- Can implement at application, transport or network level
- Until networks fully secure:
  - firewalls provide protection from external threats