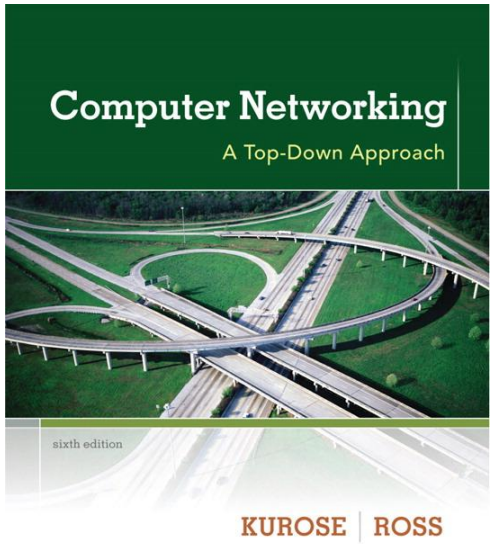
Security Building Blocks: CIA

- Confidentiality: cryptography
- Integrity: digital signatures
- Availability



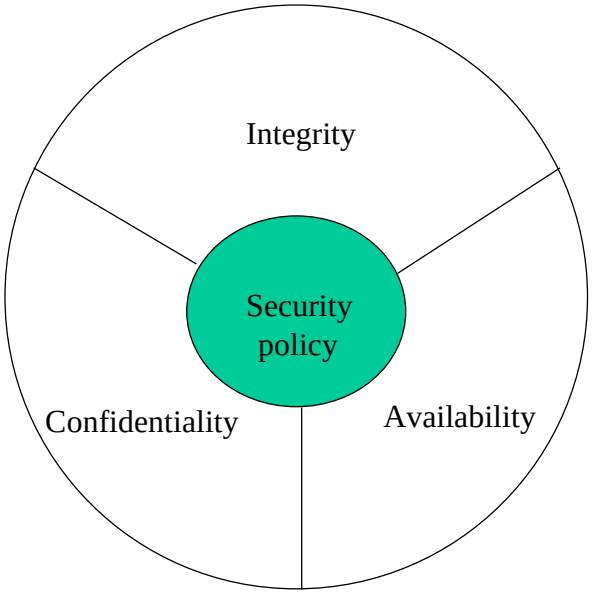
Networking: A Top Down Approach
7th edition
Jim Kurose, keith Ross
Addison-Wesley
March 2017

Note: some of the slides are directly or indirectly from the Kurose/Ross text
Contact Dr Jim Martin at jmarty at Clemson.edu for copyright concerns.

Terminology

- **Information security**: refers to the processes and methodologies which are designed and implemented to protect print, electronic, or any other form of confidential, private and sensitive information or data from unauthorized access, use, misuse, disclosure, destruction, modification, or disruption
- **CyberSecurity**: some view it as a type of information security
  - Gartner defines it as: Cybersecurity encompasses a broad range of practices, tools and concepts related closely to those of information and operational technology security. Cybersecurity is distinctive in its inclusion of the offensive use of information technology to attack adversaries.“
  - Others view it as broader than information security as it includes physical + software infrastructure (aka cyberinfrastructure or critical infrastructure)
- **Network security:** can be viewed as a subset of information security. Includes procedures such that we have confidence that information and services that are available on a network can not be accessed by unwanted users and can not be misused.
- **Internet Security:** represents measures/procedures to protect data during their transmission over a collection of interconnected networks

Network Security



- Confidentiality: involves encrypting data to prevent eavesdropping
- Integrity: ensures that data has not been tampered with in transit.
  - Data integrity, message authentication
- Availability: ensures resources (data, services, etc.) are always available in a timely manner.
  - End user authentication
  - Authorization
  - Intrusion Detection

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

Security Services

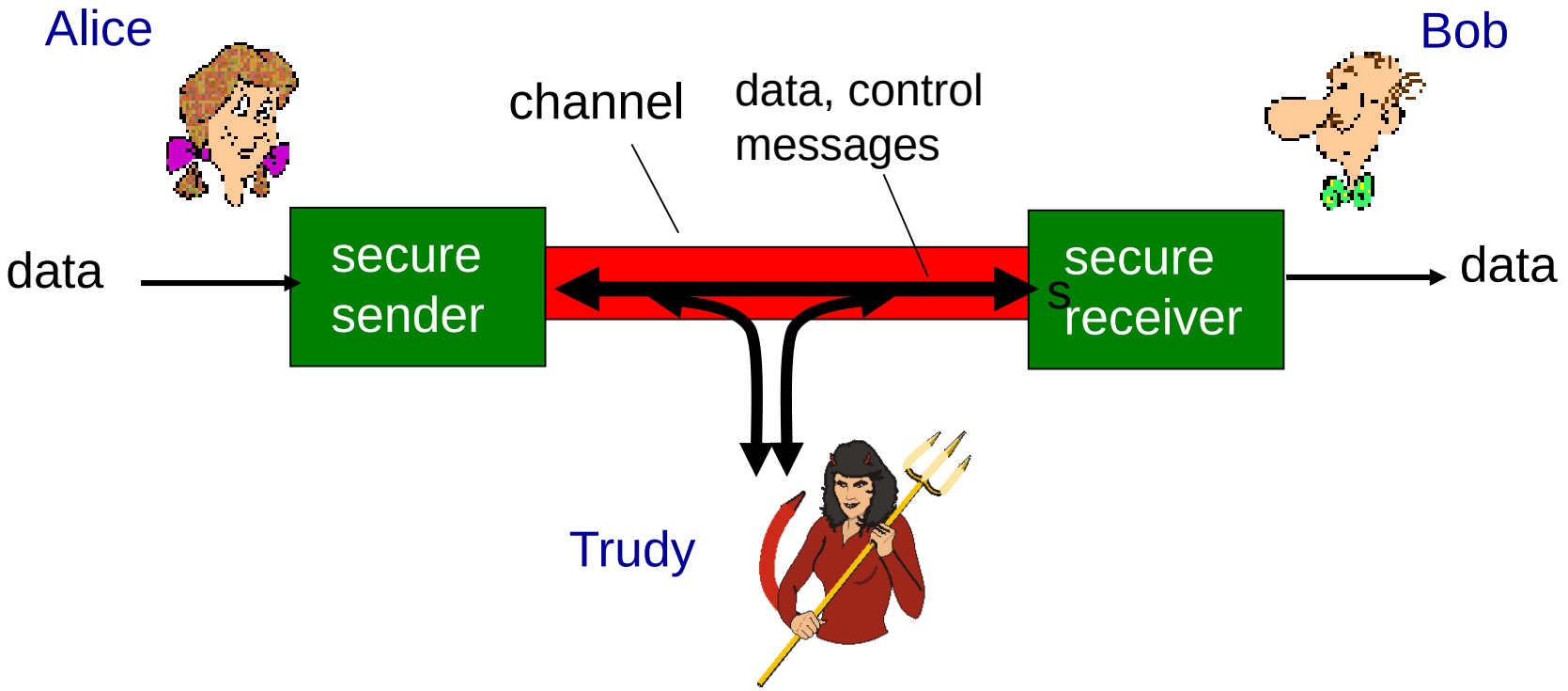
- The following are components of OSI standard security architecture (referred to as X.800)
  - Authentication - assurance that the communicating entity is the one claimed
  - Access Control - prevention of the unauthorized use of a resource
  - Data Confidentiality –protection of data from unauthorized disclosure
  - Data Integrity - assurance that data received is as sent by an authorized entity
  - Non-Repudiation - protection against denial by one of the parties in a communication

Basic Terminology

- **Plaintext (P or m) -** the original message
- **Ciphertext (C) -** the coded message
- **cipher -** algorithm for transforming plaintext to ciphertext
- **Key (K) -** info used in cipher known only to sender/receiver
- **encipher (encrypt) -** converting plaintext to ciphertext
- **decipher (decrypt) -** recovering ciphertext from plaintext
- **cryptography -** study of encryption principles/methods
- **cryptanalysis (codebreaking) -** the study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology -** the field of both cryptography and cryptanalysis

Friends and enemies: Alice, Bob, Trudy

- well-known in network security world
- Bob and Alice want to communicate “securely”
- Trudy (intruder) may intercept, delete, add messages



Who might Bob, Alice be?

- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates

There are bad guys (and girls) out there!

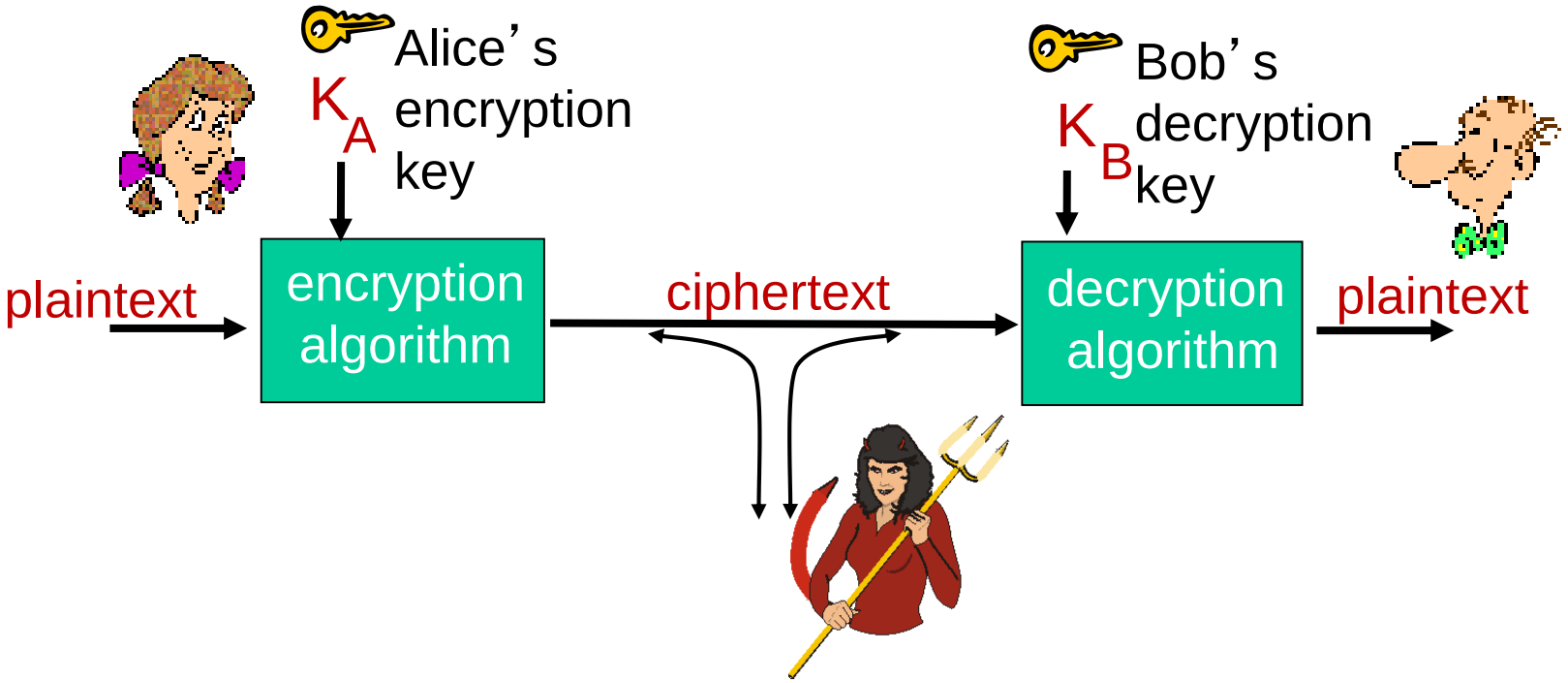
Q: What can a "bad guy" do?

A:

-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)

The language of cryptography



m plaintext message

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

$m = K_B(K_A(m))$

Cryptographic Algorithms

- Cryptography is the backbone of network security as it is used to provide (in part): confidentiality, authentication, integrity and nonrepudiation.
- A cryptographic algorithm, aka a cipher, is the mathematical function used for encryption and decryption.
- Restricted algorithm: if the security of the algorithm is based on keeping the algorithm secret.
- Better approach : use a key algorithm.

Cryptography

- can characterize by:
  - type of encryption operations used
    - substitution / transposition / product
  - number of keys used
    - single-key or private / two-key or public
  - way in which plaintext is processed
    - block / stream

Encryption/Decryption with Keys

- A range of values for a key (K) is the keyspace.
- Two classes: symmetric key and public key.
- Symmetric algorithms have been around for many years.
  - Two building blocks: Substitution and transposition.

Classical Substitution Ciphers

- where letters of plaintext are replaced by other letters or by numbers or symbols
- or if plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

- earliest known substitution cipher
- by Julius Caesar
- first attested use in military affairs
- replaces each letter by 3rd letter on
- example:

```
meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB
```

Caesar Cipher

- can define transformation as:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

- mathematically give each letter a number

```
a b c d e f g h i j k  l  m
0 1 2 3 4 5 6 7 8 9 10 11 12
n  o  p  q  r  s  t  u  v  w  x  y  Z
13 14 15 16 17 18 19 20 21 22 23 24 25
```

- Since this was applied to writing, the language of the plaintext was limited to 26. (assumes spaces are not encrypted)
- Caesar cipher defined:

$C = \mathrm{E}(p) = (p + k) \bmod (26)$

$p = \mathrm{D}(\mathrm{C}) = (\mathrm{C} - k) \bmod (26)$

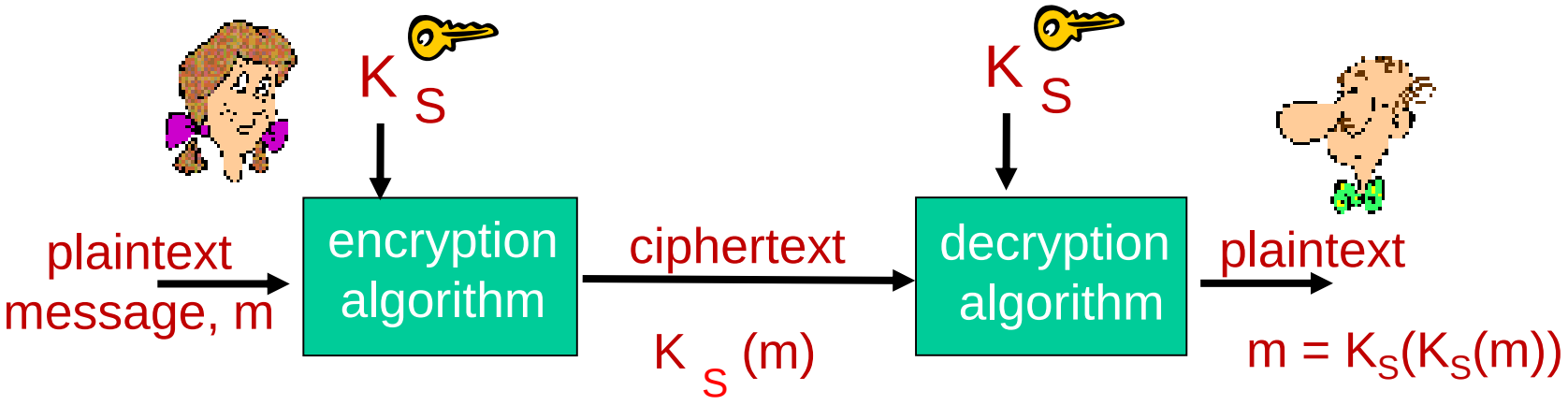
Cryptanalysis of Caesar Cipher

- If it is known that a given ciphertext is a Caeser cipher, brute-force cyrptanalysis is easy.
  - The encryption algorithms are known
  - There are 26 keys, although only 25 key useful (a key of 0 would be pretty silly!!!)
  - The language of the plaintext is known and easily recognized
- Cycle through all keys, display the decrypted

Modern Symmetric Key Algorithms

- Aka conventional algorithms.
  - Stream algorithms: operate on data a bit or an octet at a time.
  - Block algorithms: operate on data a block at a time (e.g., 64 bit block size).
  - Tradeoff: speed versus robust security
  - Both sides require the same key.
  - The secret is defined by the key and not the algorithm.

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?

DES: Data Encryption Standard

- Developed in 1970's by IBM.
- Block algorithm, 64-bit block size
- 56-bit key
- 64 bits of cleartext go through mutliple 'rounds' leading to 64 bits of ciphertext
- Based on the Feistel Cipher incorporates:
  - Diffusion: the statistical structure of the plaintext is dissipated into long range statistics.
  - Confusion: the relationship between the statistics of the ciphertext and the value of the key is made as complex as possible.
- Strength of DES:even though large keyspace (2EXP56), can be broken in hours on optimized hardware.
  - Some knowledge of the expected plaintext is needed but essentially a brute force attack.
- Triple-DES was developed to address this problem.
  - 168-bit key, but computationally expensive.
  - Possible alternative is Advanced Encryption Standard (AES)

Types of Cryptanalytic Attacks

- **ciphertext only**
  - only know algorithm / ciphertext, statistical, can identify plaintext
- **known plaintext**
  - know/suspect plaintext & ciphertext to attack cipher
- **chosen plaintext**
  - select plaintext and obtain ciphertext to attack cipher
- **chosen ciphertext**
  - select ciphertext and obtain plaintext to attack cipher

Brute Force Search

- always possible to simply try every key
- most basic attack, proportional to key size
- assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\ \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\ \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ years	6.4×10^{6} years

Key Distribution

- Symmetric encryption requires two parties to exchange the key.
- The strength of any crypto system rests on the key distribution technique.
- Options:
  - A selects a key and physically delivers to B
  - A third party selects the key and physically delivers to A and B.
  - If A and B have communicated previously with a key, one party can transmit the new key to the other using the old key for encryption.
  - If A and B have a secure connection to C, C can deliver the key on the encrypted links to A and B.

Public Key Algorithms

- Aka asymmetric algorithms.
  - The key used for encryption is different from the key used for decryption.
  - The encryption key is the generally the public key.
  - The decryption key is the private key.

Public Key Algorithms

- A and B agree on a public key cryptosystem
- A sends to B its public key
- B encrypts the data using A’s public key
- A decrypts B’s message using its private key

Public Key Algorithms

- Diffie and Hellman came up with the breakthrough method in 1976 that addressed the following requirements:
  - Computationally easy for party B to generate a key pair
  - Computationally easy for a sender A, knowing B's public key and the plaintext message, to generate the ciphertext
  - Computationally easy for receiver B to decrypt the resulting ciphertext using the private key.
  - Computationally infeasible for an opponent, knowing the public key to determine the private key.
  - Computationally infeasible for an opponent, knowing the public key and the ciphertext to recover the original message.

Public Key Algorithms

- Public key algorithms are based on mathematical functions rather than substitution and permutations.
- Profound impact in the area of confidentiality, key distribution and authentication.
- Two misconceptions:
  - Public key encryption is more secure than symmetric.
  - Public key encryption that has made symmetric encryption out of date.

Public Key Cryptography

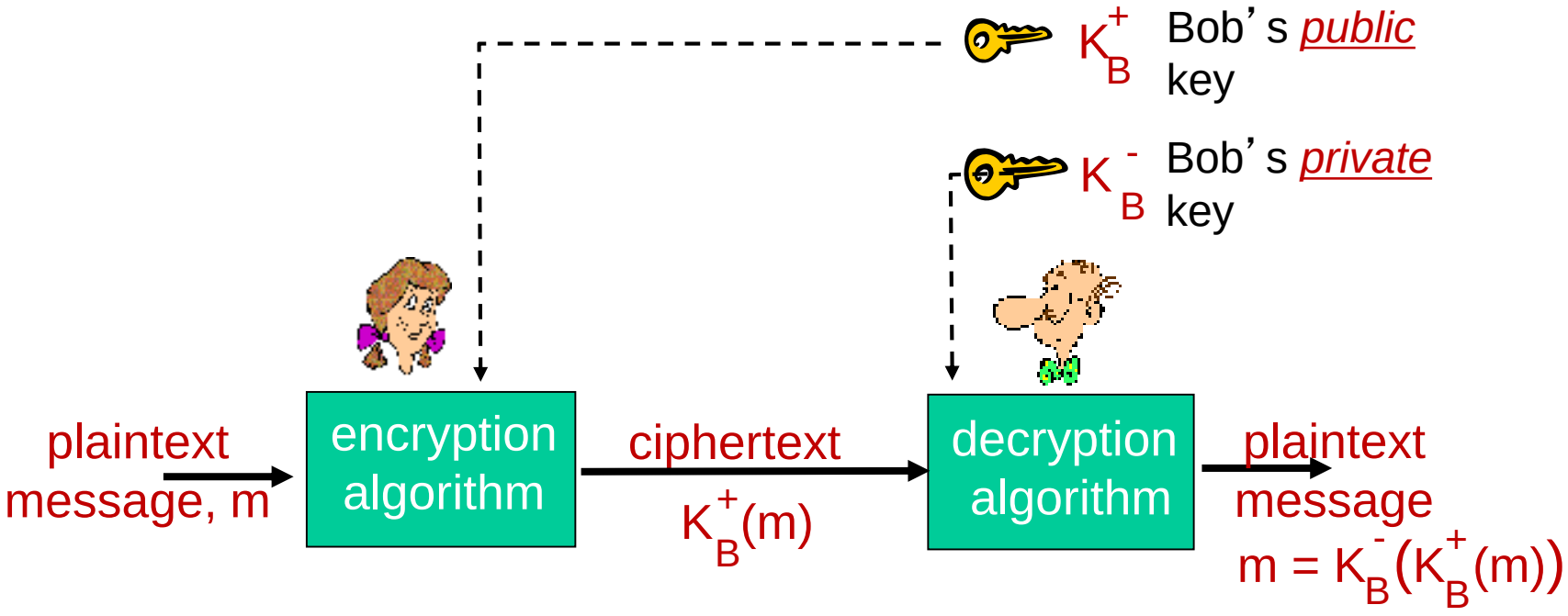
symmetric key crypto

- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?

public key crypto

- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

Public key cryptography



Public Key Cryptography

symmetric key crypto

- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?

public key crypto

- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

Public key encryption algorithms

requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that

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

2. given public key K_B^+, it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

RSA: Creating public/private key pair

1. choose two large prime numbers p, q. (e.g., 1024 bits each)
2. compute $n = pq$, $z = (p-1)(q-1)$
3. choose e (with $e<n$) that has no common factors with z (e, z are "relatively prime").
4. choose d such that $ed-1$ is exactly divisible by z. (in other words: ed mod $z = 1$).
5. *public* key is (n,e). *private* key is (n,d).

K_B^+ K_B^-

Why is RSA secure?

- suppose you know Bob's public key (n,e). How hard is it to determine d?
- essentially need to find factors of n without knowing the two factors p and q
  - fact: factoring a big number is hard

Public Key Algorithms

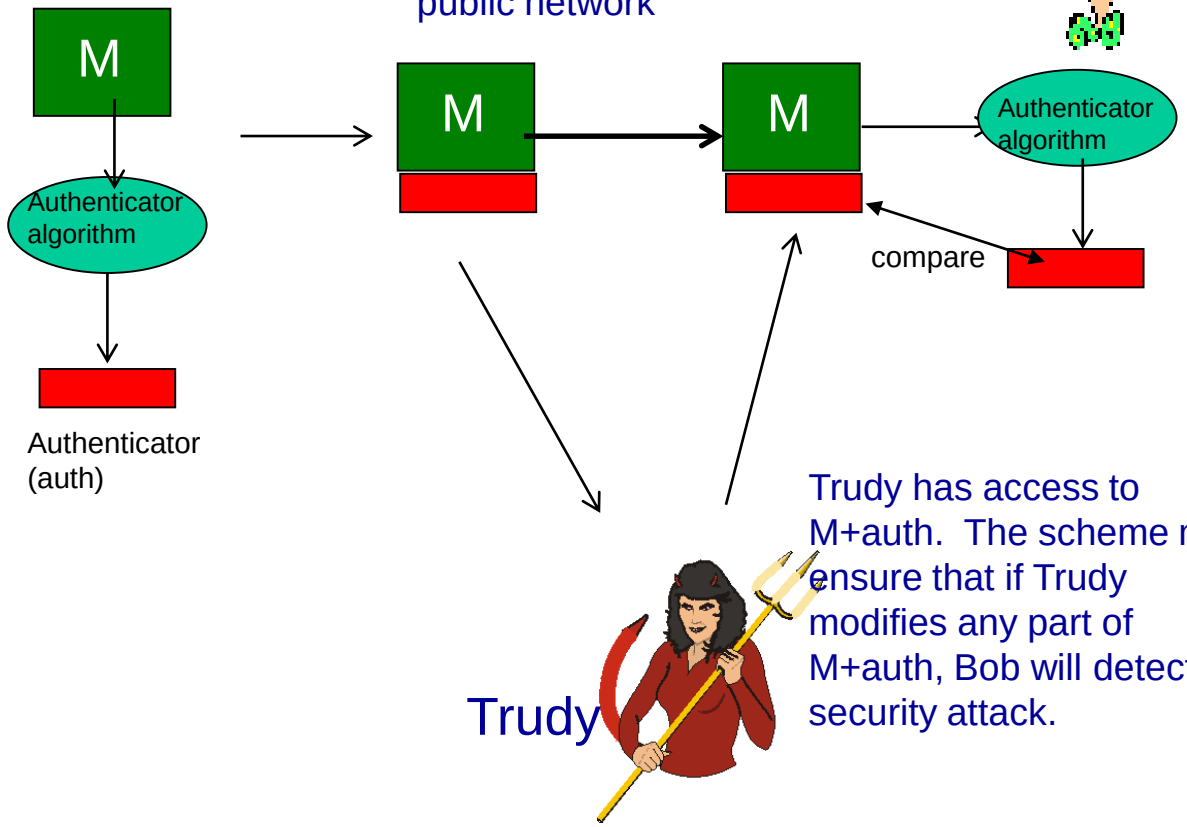
- The method allows for :
  - Encryption/decryption
  - Digital signatures
  - Key exchange

Public Key Algorithms

- Summary:
  - Public Key encryption is as secure as symmetric encryption.
    - It solves the key management problem.
    - Enables digital signatures.
  - Drawback: Very slow (1000 times slower than symmetric algorithms).
  - Generally used to distribute session keys in a hybrid cryptosystem.

Message Authentication

Alice sends message M to Bob



Bob receives M+auth and applies the authenticator algorithm to verify if M was modified

Trudy has access to M+auth. The scheme must ensure that if Trudy modifies any part of M+auth, Bob will detect the security attack.

Message Authentication

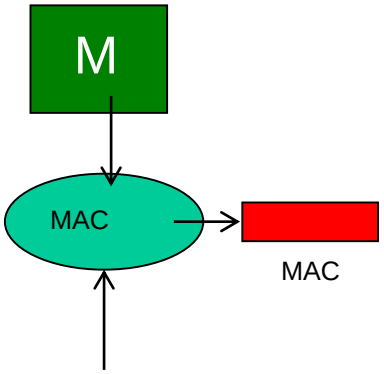
- There are at least two widely used ‘authenticator’ methods:
  - Message Authentication Code (MAC)
  - Hash
- The term ‘message digest’ is used to identify the value produced by the authenticator algorithm

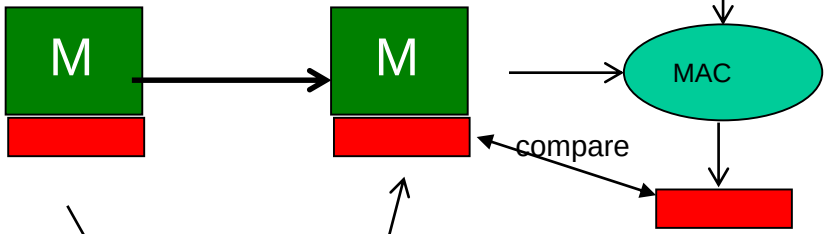
Message Authentication Code (MAC)

Alice sends message M to Bob

Msg + MAC sent over the public network

Bob receives M+MAC and applies the MAC using the shared secret key, K to produce MAC. This MAC should be identical to the MAC that was received.





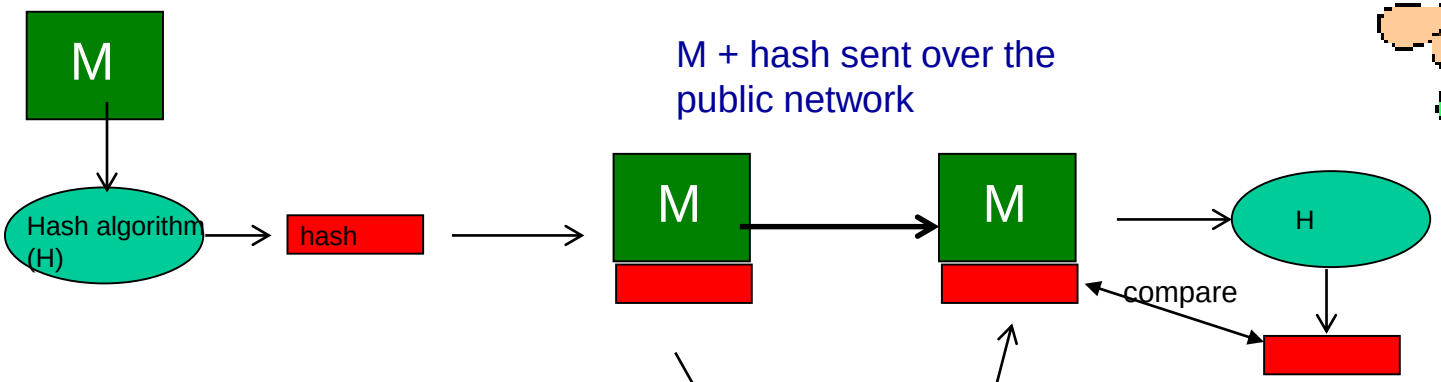
Trudy needs the secret key in order to carry out a 'woman-in-the-middle' attack

Trudy

Hash

Alice sends message M to Bob

Bob receives M+hash and applies the hash algorithm to produce a hash which must match the hash that was received.



Requirements of a cryptographic hash

- H can be applied to a block of data of any size
- H produces a fixed-length output (e.g., MD5 produces a 128 bit hash)
- For a given hash generated by H(M), it is computationally infeasible to recreate M such that H(x) = hash
  - A hash is one-way function
- It is computationally infeasible to find to different messages, M1 and M2, such that H(M1)=H(M2)
  - A hash should exhibit strong collision resistance

If Trudy can determine the hash algorithm, she can modify M, update the hash, and forward to Bob.

Trudy

Message Authentication

- A MAC authenticator is similar to encryption and is considered computationally complex
- A hash is preferred as it is less complex
  - Problem is how to prevent Trudy from being able to compute a valid hash after modifying the message data?

Secure Message Authentication Using a Hash

- At least three approaches for implementing secure message authentication using a hash
  1. Alice generates the hash AND then encrypts the hash using a shared secret key that is available to Bob.
  2. Shared secret : Similar to #1 except does not require encryption
  3. Public key: this is referred to as a digital signature.

Authentication: Hash with Symmetric Encryption

Alice sends message M to Bob

M

Hash algorithm (H)

hash

Encryption (E)

Secret key (K)

Encrypted hash (Ehash)

M + Ehash sent over the public network

M

M

Trudy

Trudy will not be able to create a valid hash as she does not have the secret key

Bob receives M+Ehash and applies the hash algorithm on M to produce a hash which must match the decrypted hash that was received.

H

compare

Decrypt (D)

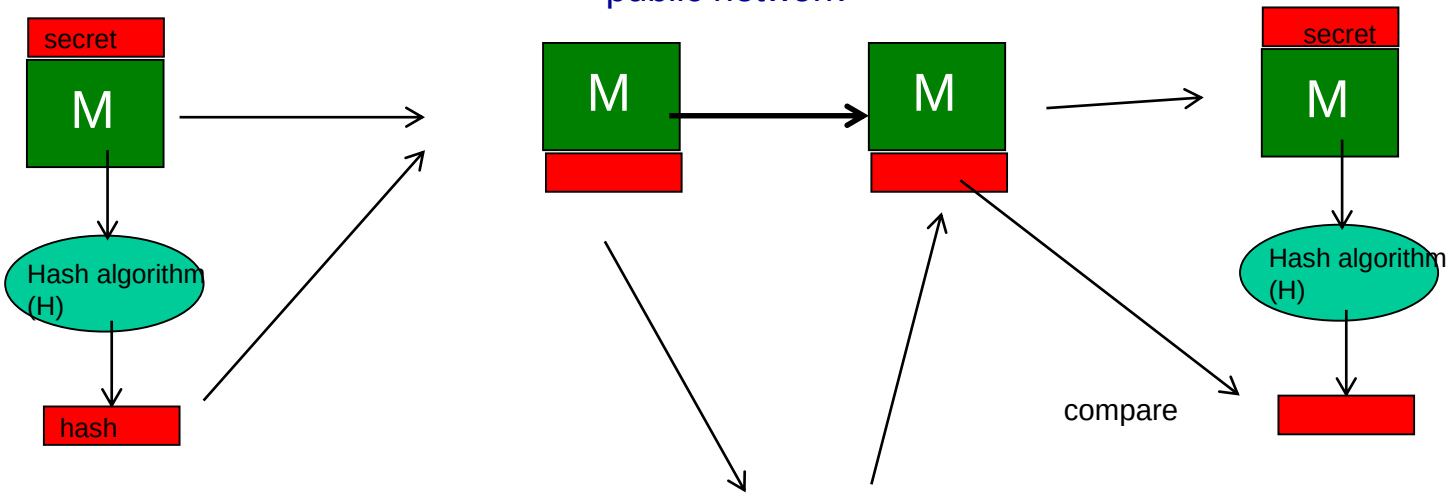
Secret key (K)

Authentication: Hash with Shared Secret

Alice sends message M to Bob

Bob receives M+hash and applies the hash algorithm on M AND the shared secret to produce a hash which must match the hash that was received.

M + hash sent over the public network

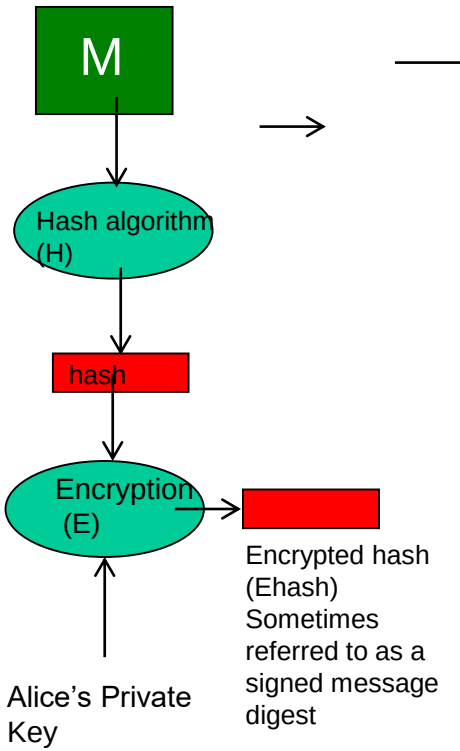


Trudy

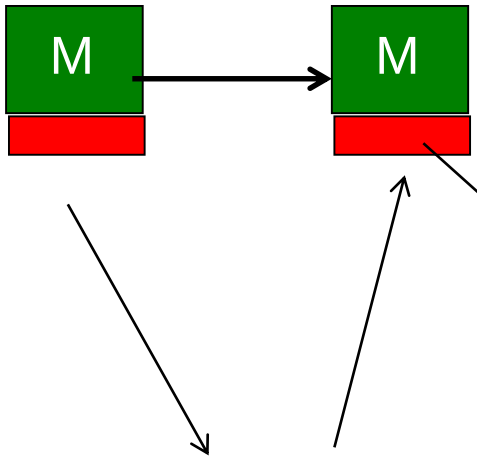
Trudy will not be able to create a valid hash as she does not have the shared secret

Authentication: Digital Signature

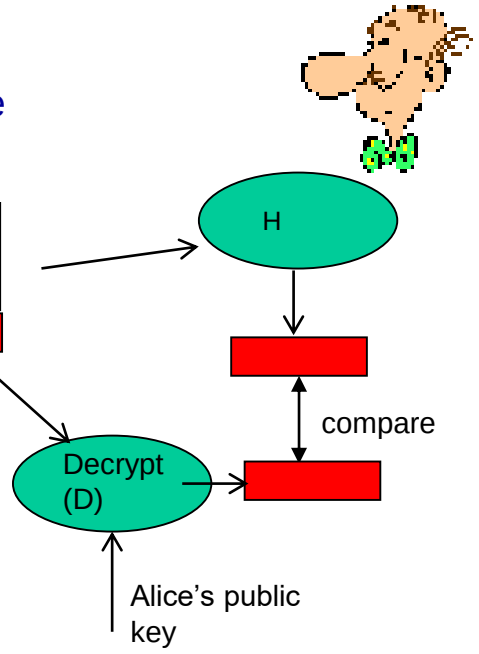
Alice sends message M to Bob, signs The message using her private key



M + Ehash sent over the public network



Bob receives M and applies the hash algorithm to produce a hash which must match the decrypted hash that was also received.



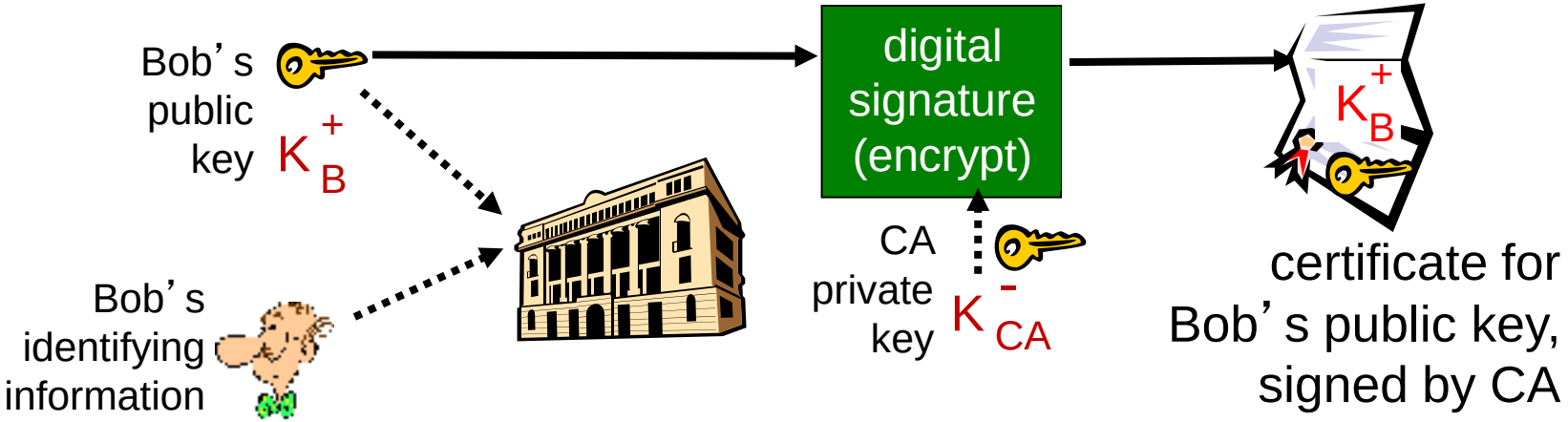
I Trudy modifies M, she will not be able to create an updated valid hash that would appear to Bob as M was signed by Alice.

Public-Key Certification

- Consider this problem: Anyone can pretend to be Alice….what if Trudy send's to Bob her public key claiming it is Alice's public key?
- A public key infrastructure is required
  - Certificate: hold a public key (e.g., Alice's), an expiration date of the certificate, crypto options or details, all of which is digitally signed by a trusted third party
    - Standard format is X.509 certificates
  - Certificate Authority (CA): Usually a government agency or a financial institution, provides a service to digitally sign other's (e.g., Alice's) certificate.

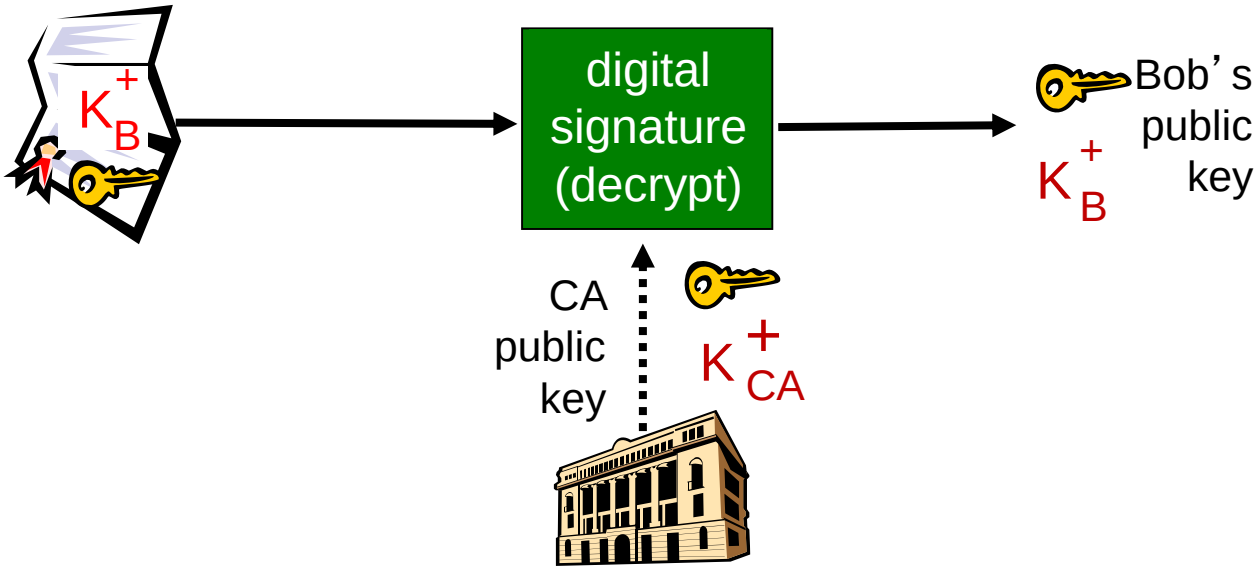
Certification authorities

- *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"



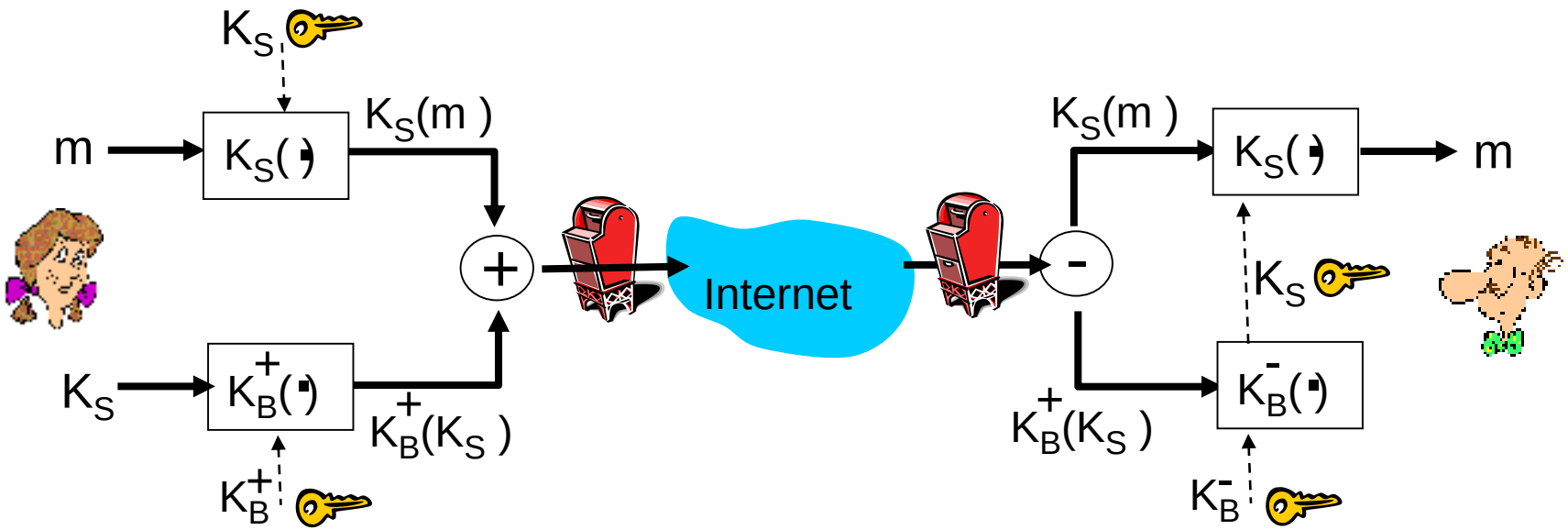
Certification authorities

- when Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere).
  - apply CA's public key to Bob's certificate, get Bob's public key



Secure e-mail

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

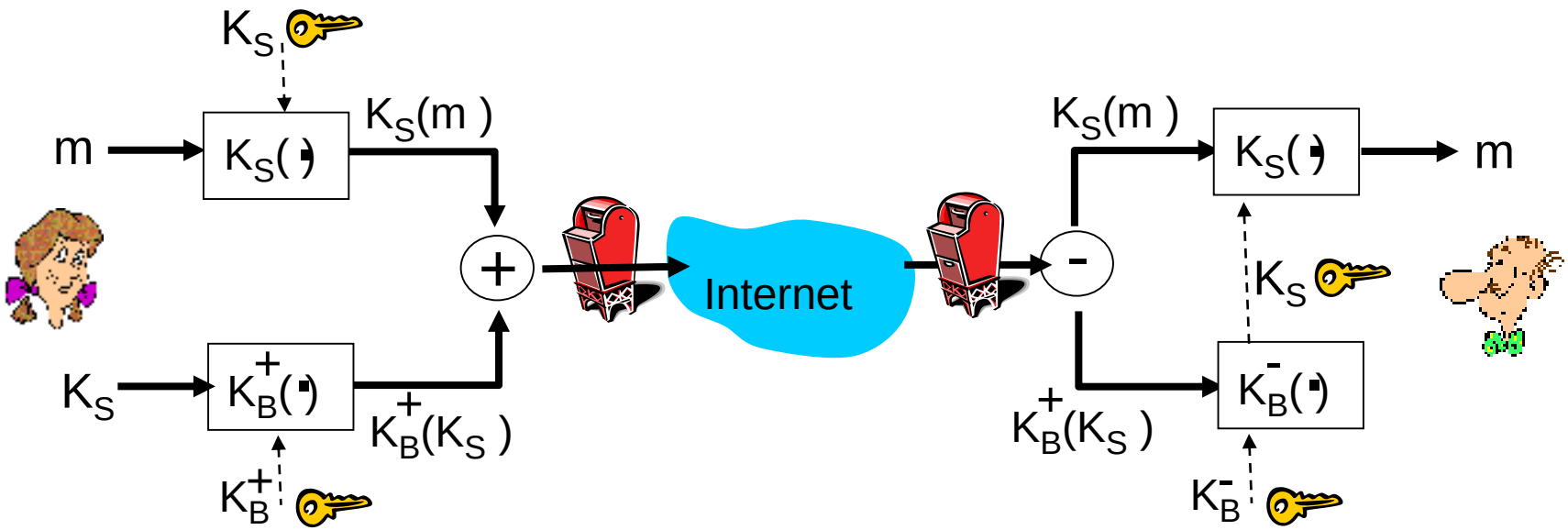


Alice:

- 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

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

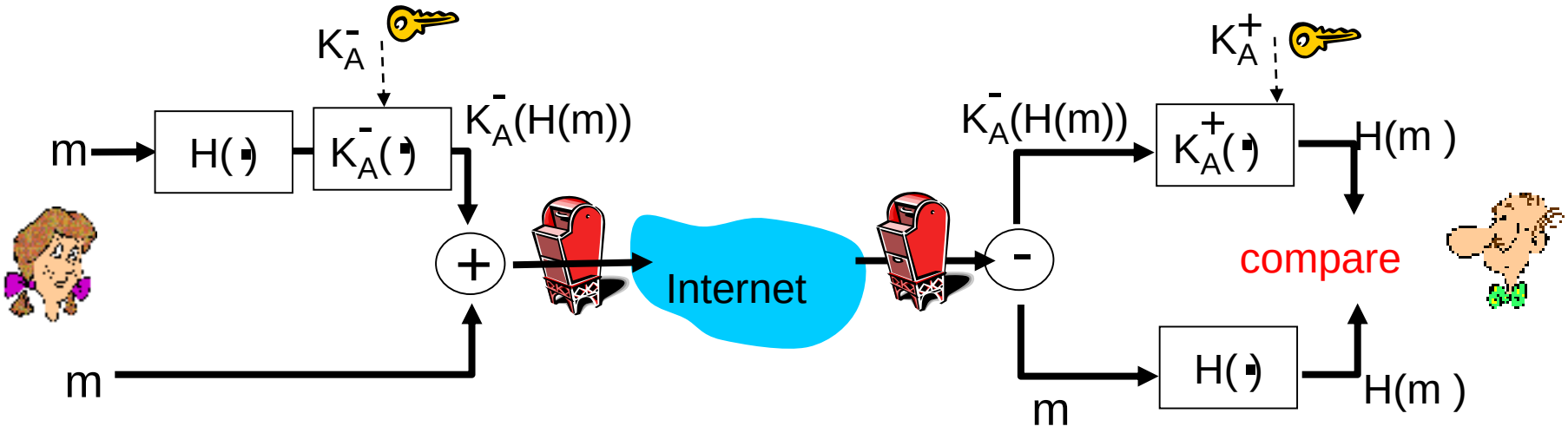


Bob:

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

Secure e-mail (continued)

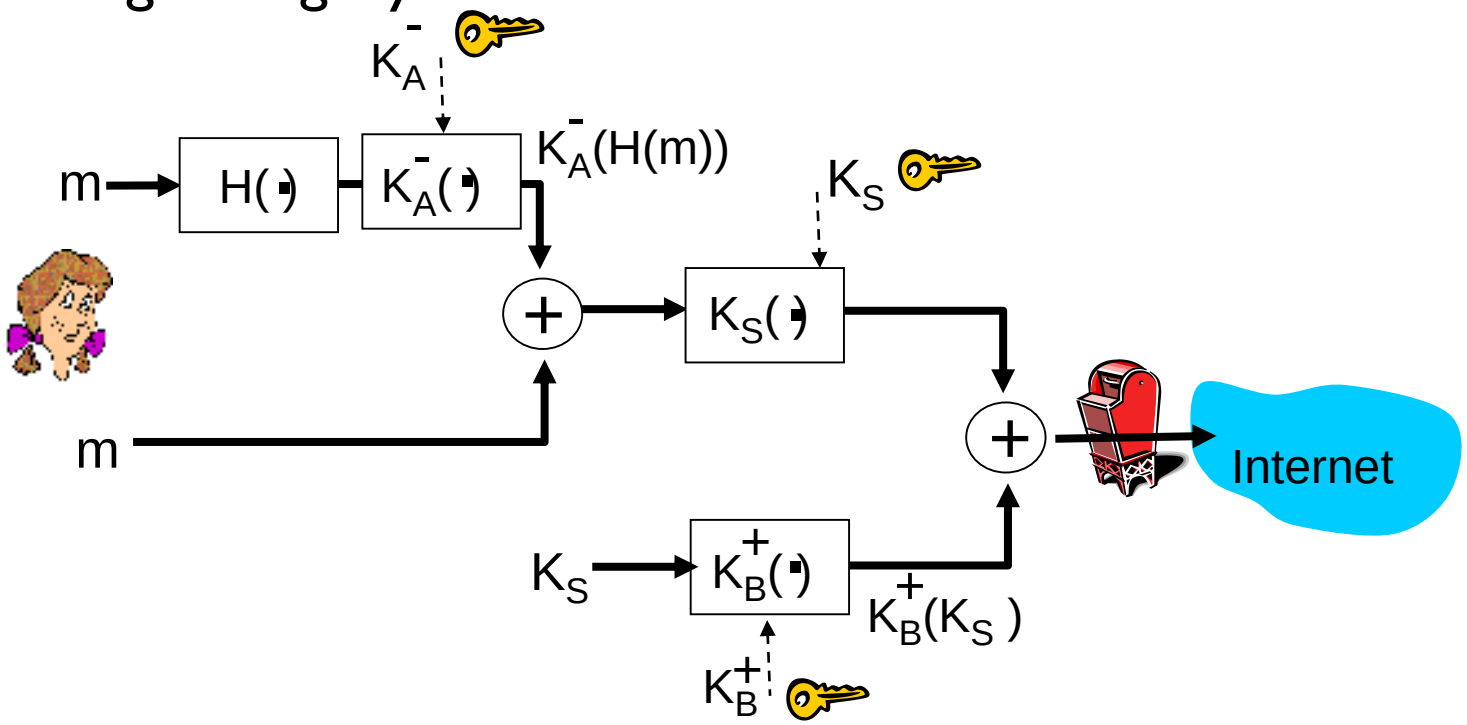
- Alice wants to provide sender authentication message integrity



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

Secure e-mail (continued)

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

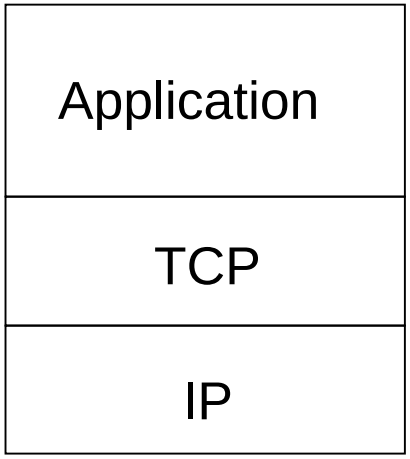


Alice uses three keys: her private key, Bob's public key, newly created symmetric key

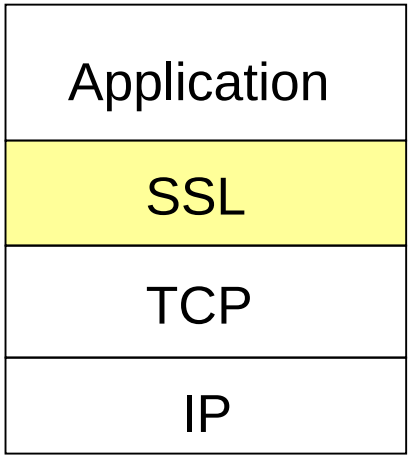
SSL: Secure Sockets Layer

- widely deployed security protocol
  - supported by almost all browsers, web servers
  - https
  - billions $/year over SSL
- mechanisms: [Woo 1994], implementation: Netscape
- variation -TLS: transport layer security, RFC 2246
- provides
  - *confidentiality*
  - *integrity*
  - *authentication*
- original goals:
  - Web e-commerce transactions
  - encryption (especially credit-card numbers)
  - Web-server authentication
  - optional client authentication
  - minimum hassle in doing business with new merchant
- available to all TCP applications
  - secure socket interface

SSL and TCP/IP



normal application



application with SSL

- SSL provides application programming interface (API) to applications
- C and Java SSL libraries/classes readily available