University of BRISTOL

Computer System- B Security

Introduction to Cryptography

Sanjay Rawat

Agenda

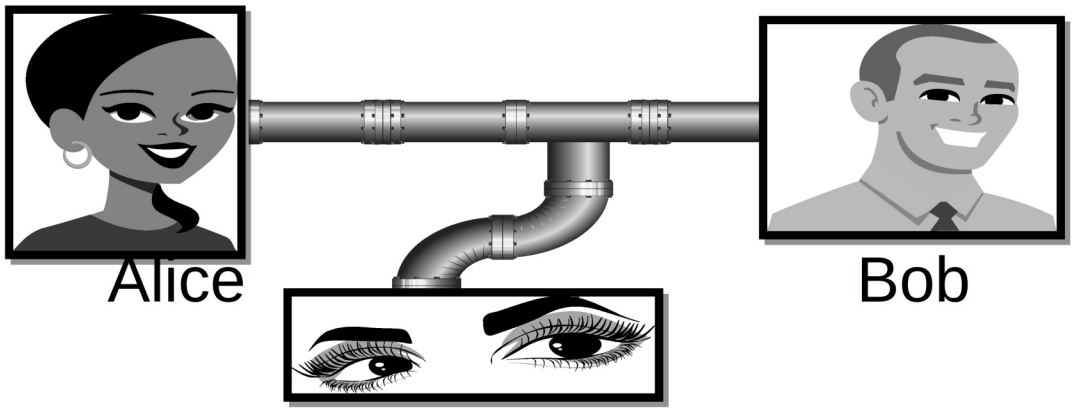
- A non-technical brief introduction to cryptography
- Where/how/why they are used in practice (real examples to follow)
- You will have more rigorous treatment in other units
- By custom, we use *Alice*, Bob and Eve as actors (parties).

Cryptographic Concepts

- **Encryption**: a means to allow two parties to establish confidential communication over an insecure channel that is subject to eavesdropping.

Cryptographic Concepts

- **Encryption**: a means to allow two parties to establish confidential communication over an insecure channel that is subject to eavesdropping.

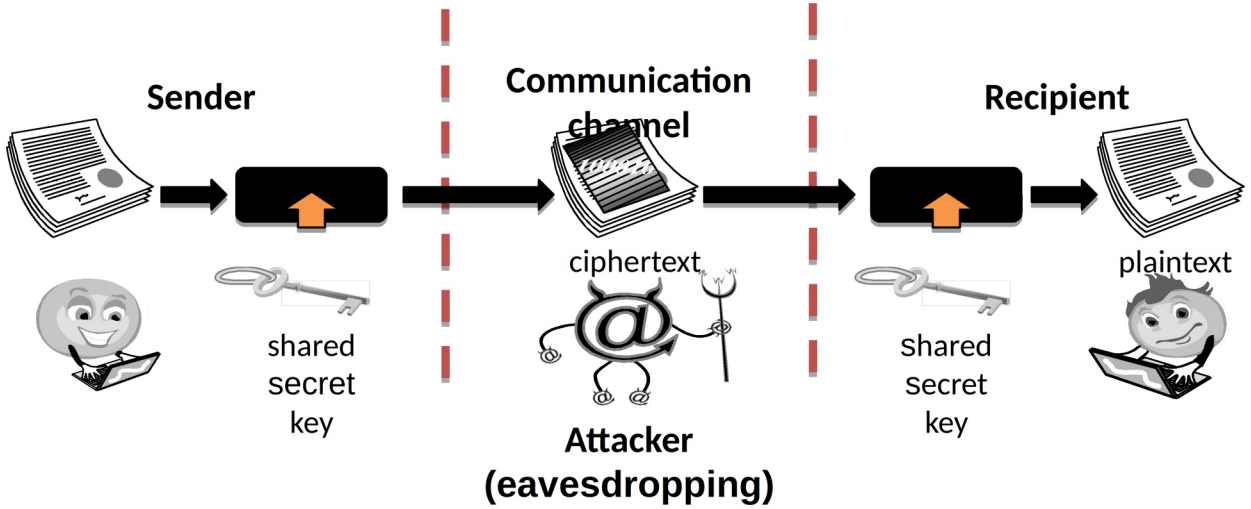


Encryption and Decryption

- The message M is called the **plaintext.**
- Alice will convert plaintext M to an encrypted form using an encryption algorithm E that outputs a **ciphertext C for M.**

Encryption and Decryption

- The message M is called the **plaintext.**
- Alice will convert plaintext M to an encrypted form using an encryption algorithm E that outputs a **ciphertext C for M.**



Encryption and Decryption

- As equations:

$$C = E(M)$$

$$M = D(C)$$

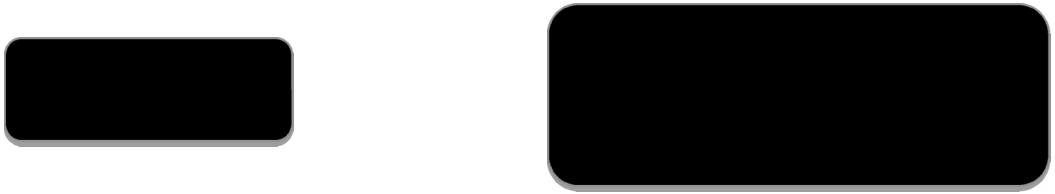
- The encryption and decryption algorithms are chosen so that it is infeasible for someone other than Alice and Bob to determine plaintext M from ciphertext C. Thus, ciphertext C can be transmitted over an insecure channel that can be eavesdropped by an adversary.

Cryptosystem

1. The set of possible plaintexts
2. The set of possible ciphertexts
3. The set of encryption keys
4. The set of decryption keys
5. The correspondence between encryption keys and decryption keys
6. The encryption algorithm to use
7. The decryption algorithm to use

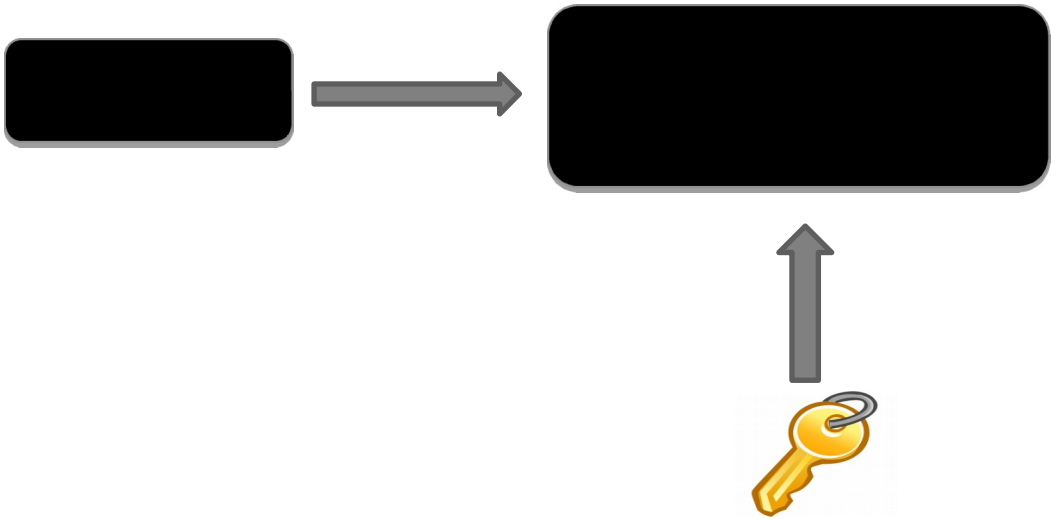
Basic Concepts

Basic Concepts

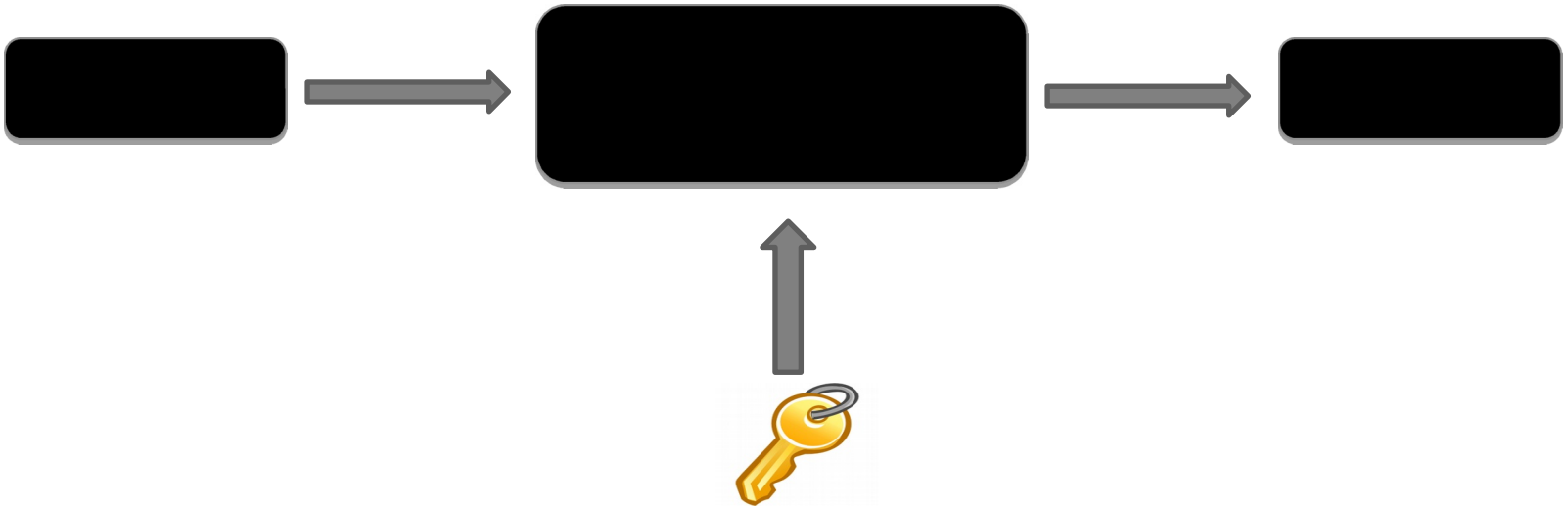


Basic Concepts

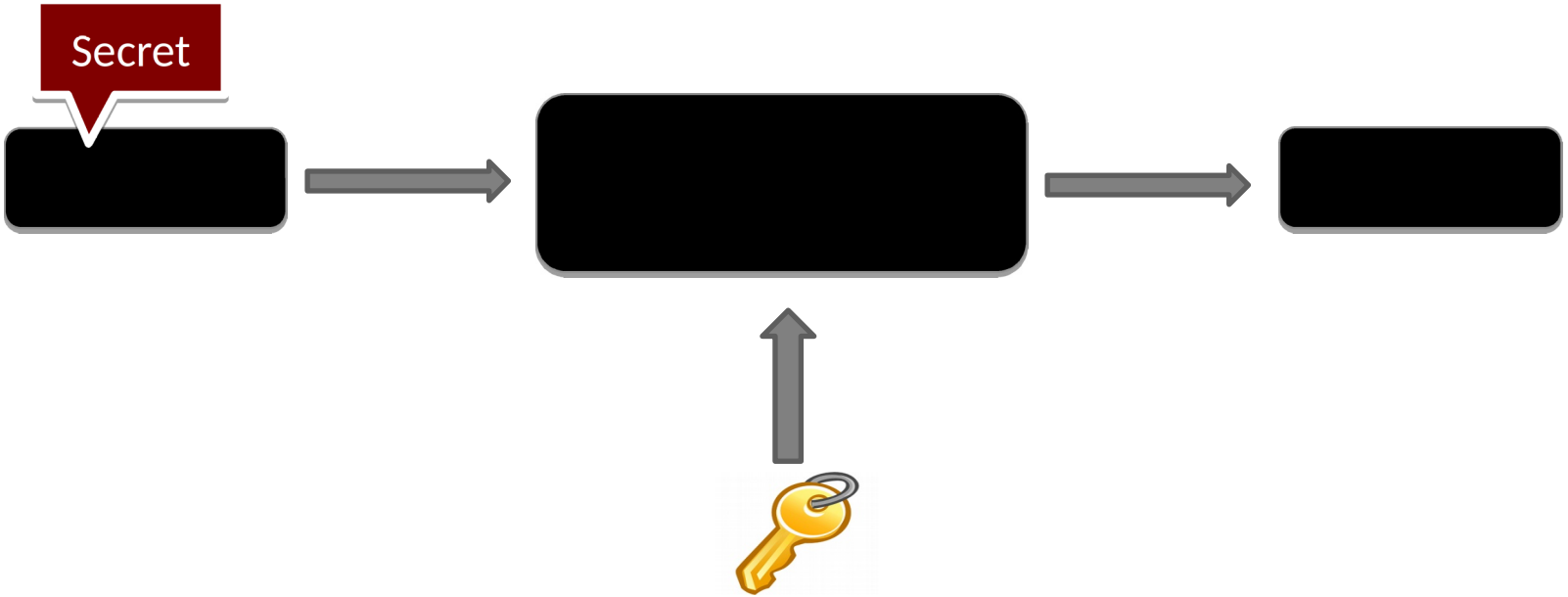
Basic Concepts



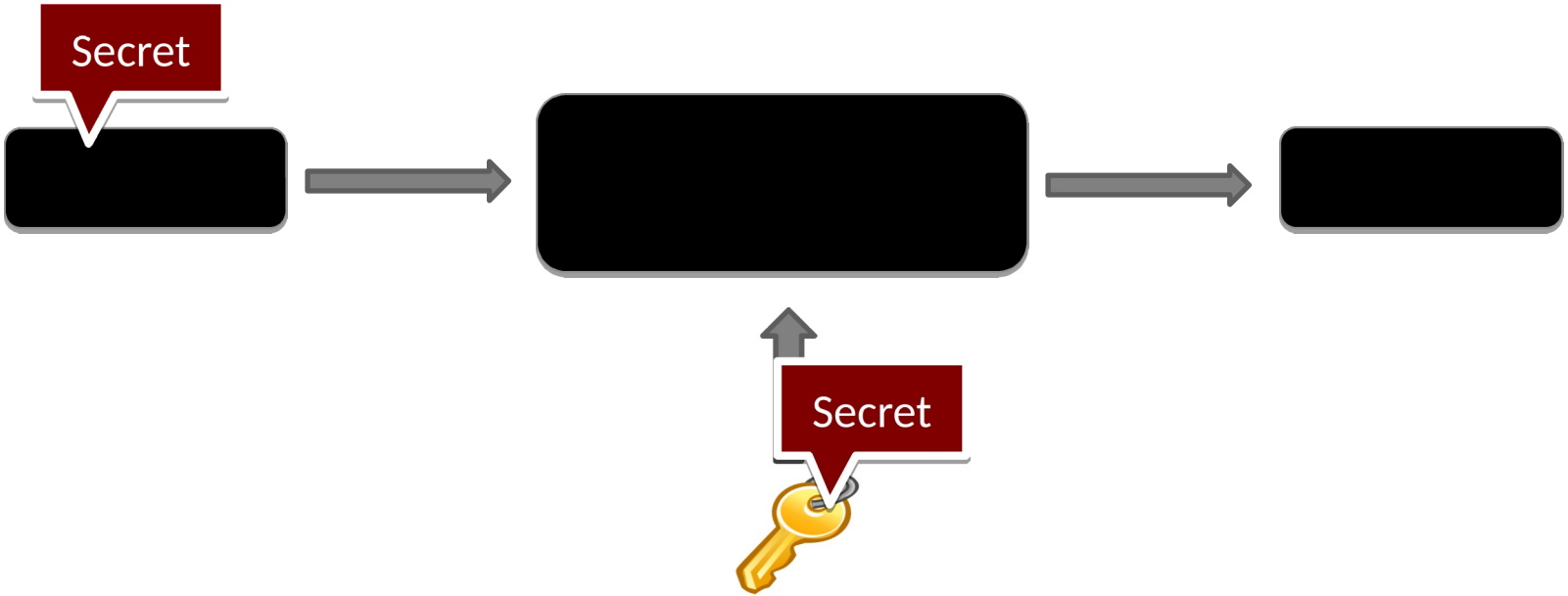
Basic Concepts



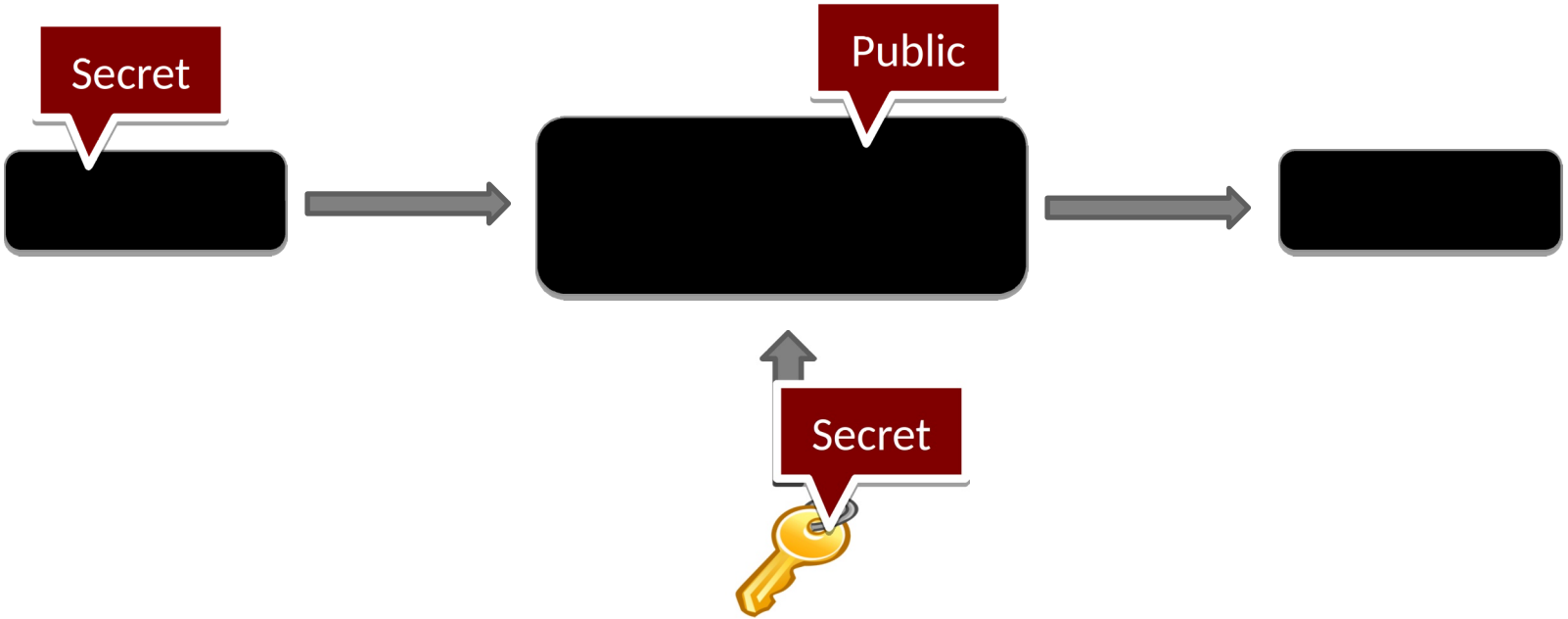
Basic Concepts



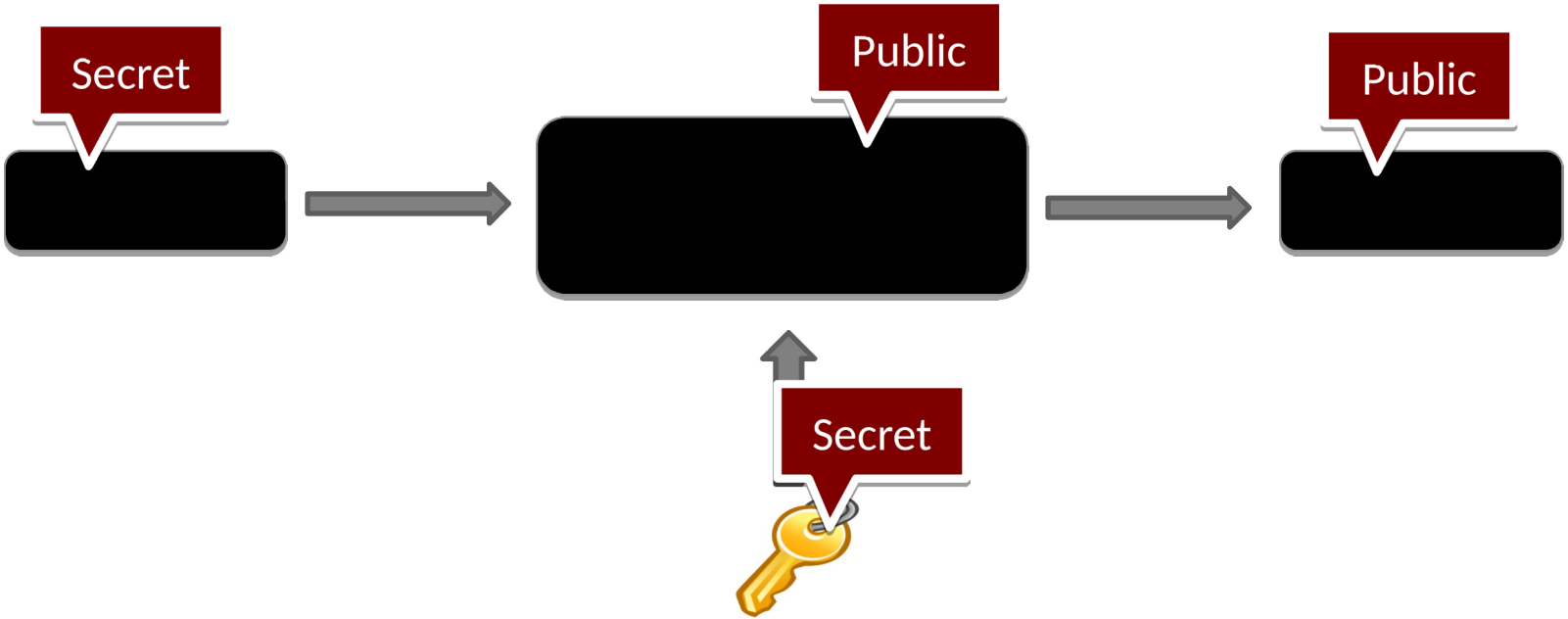
Basic Concepts



Basic Concepts



Basic Concepts

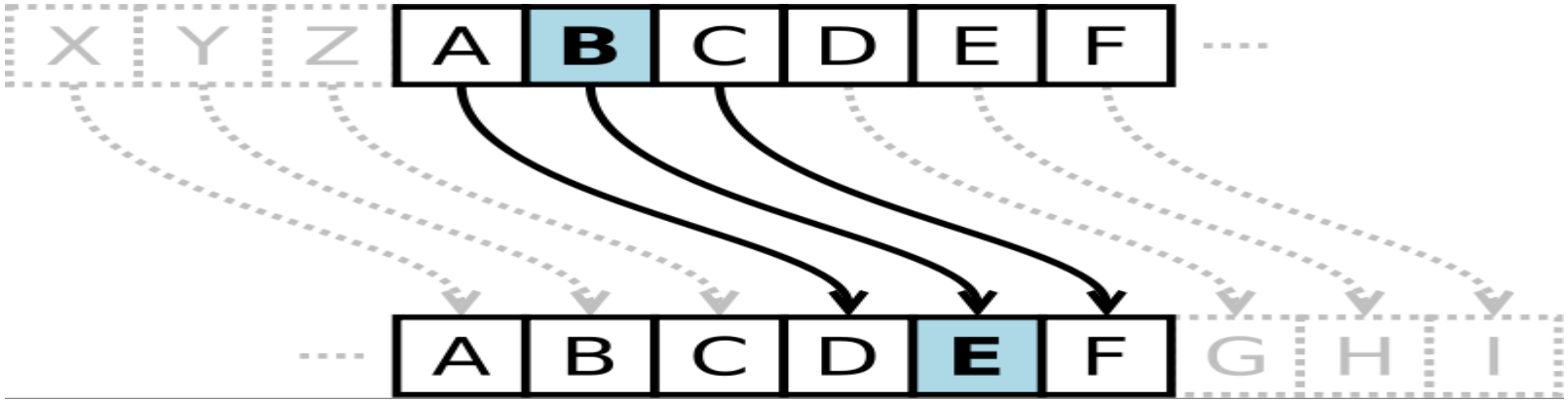


Caesar Cipher

- Replace each letter with the one “three over” in the alphabet.

Caesar Cipher

- Replace each letter with the one “three over” in the alphabet.

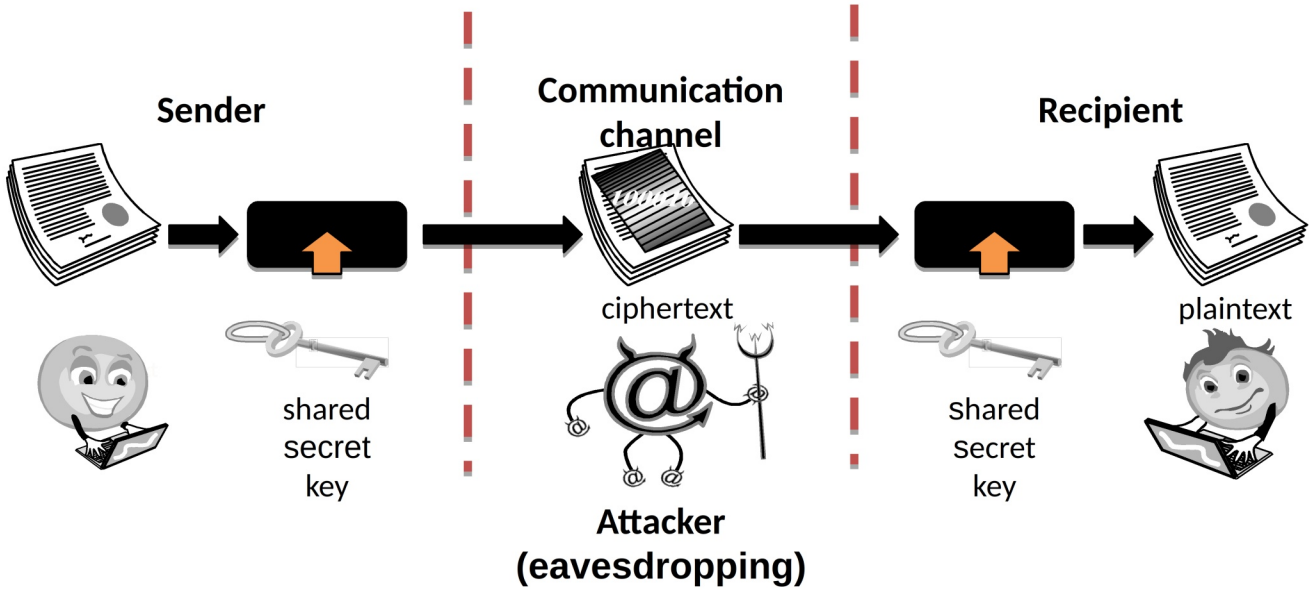


Symmetric Cryptosystems

- Alice and Bob share a secret key, which is used for both encryption and decryption.

Symmetric Cryptosystems

- Alice and Bob share a secret key, which is used for both encryption and decryption.



Symmetric Key Distribution

- Requires each pair of communicating parties to share a (separate) secret key.

Symmetric Key Distribution

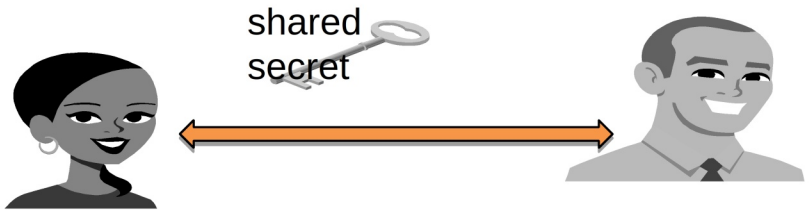
- Requires each pair of communicating parties to share a (separate) secret key.

Symmetric Key Distribution

- Requires each pair of communicating parties to share a (separate) secret key.

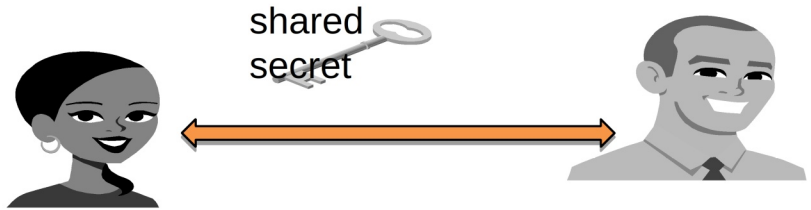
Symmetric Key Distribution

- Requires each pair of communicating parties to share a (separate) secret key.



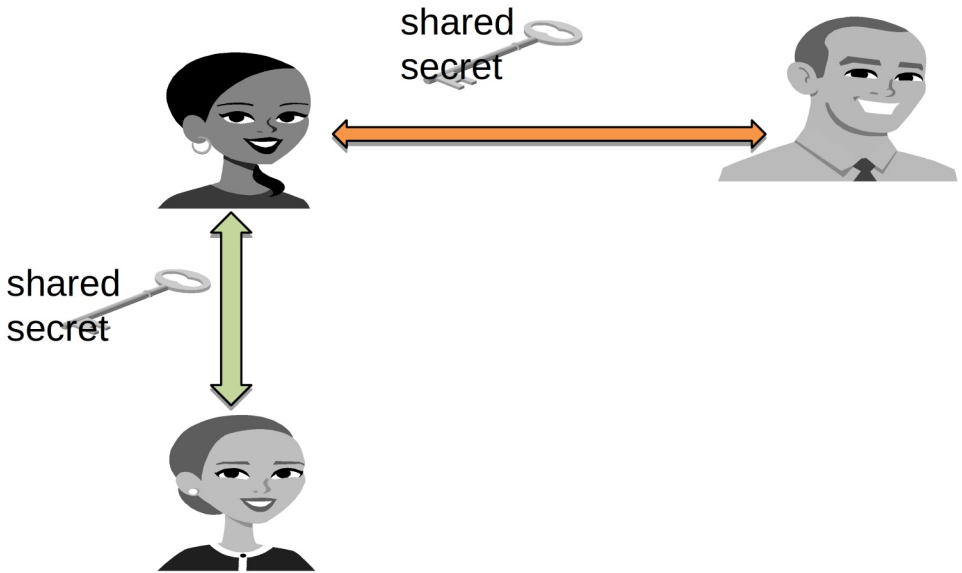
Symmetric Key Distribution

- Requires each pair of communicating parties to share a (separate) secret key.



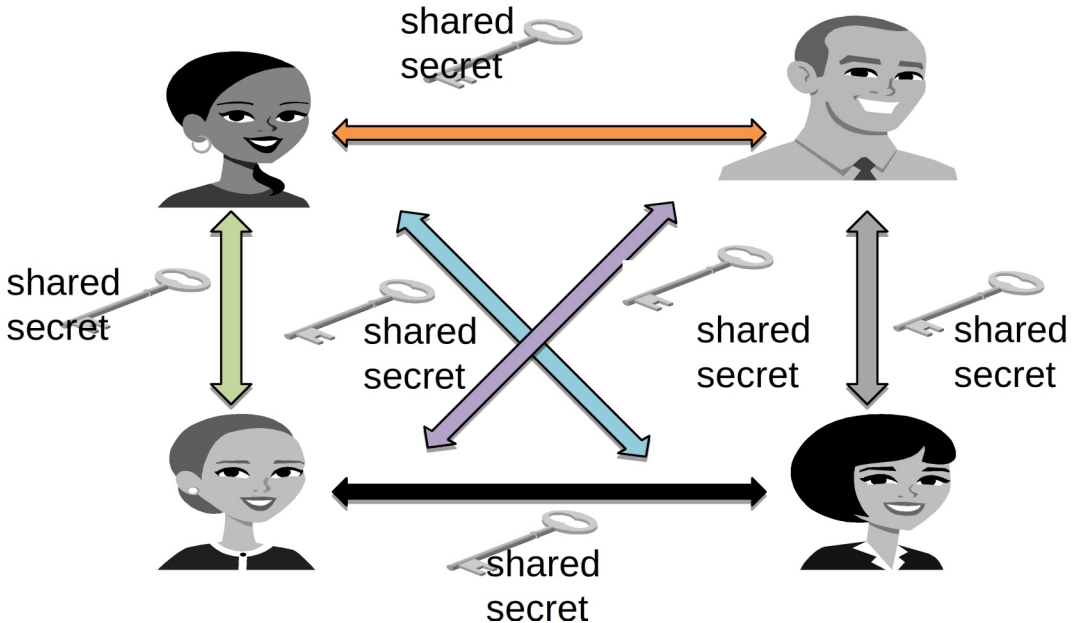
Symmetric Key Distribution

- Requires each pair of communicating parties to share a (separate) secret key.



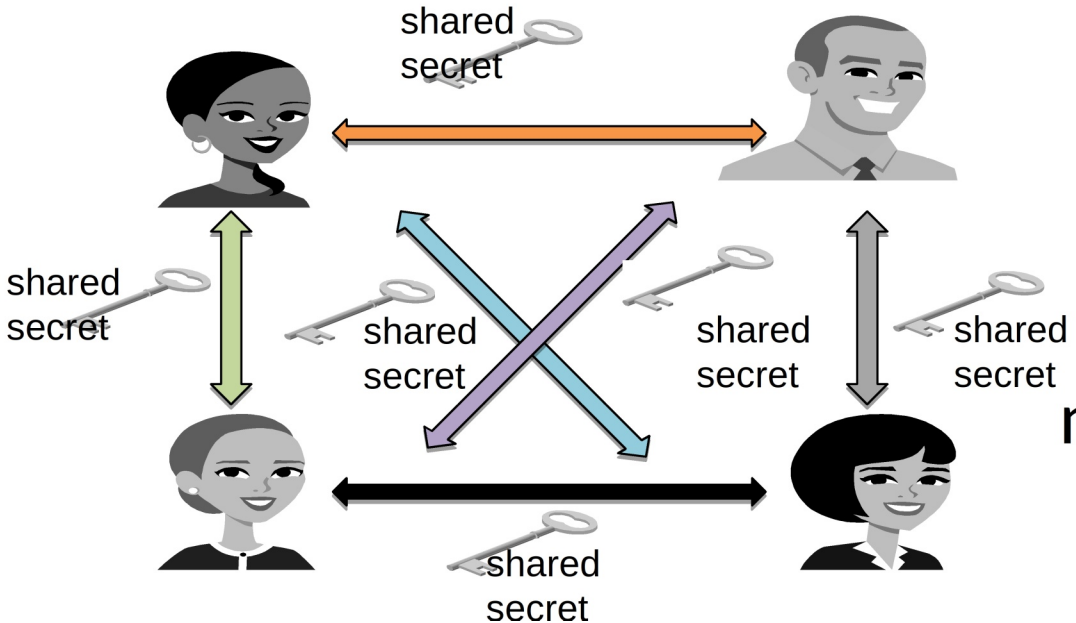
Symmetric Key Distribution

- Requires each pair of communicating parties to share a (separate) secret key.



Symmetric Key Distribution

- Requires each pair of communicating parties to share a (separate) secret key.



Symmetric Key conti...

- Example:
  - Advanced Encryption Standard -AES (current standard)
    - key lengths: 128, 192 and 256 bits
  - Data Encryption Standard- DES (triple DES)
- Computationally scalable for large messages
- Hardware implementation is available.
  - AES-NI (intel)
- Key distribution is a challenge!

Public-Key Cryptography

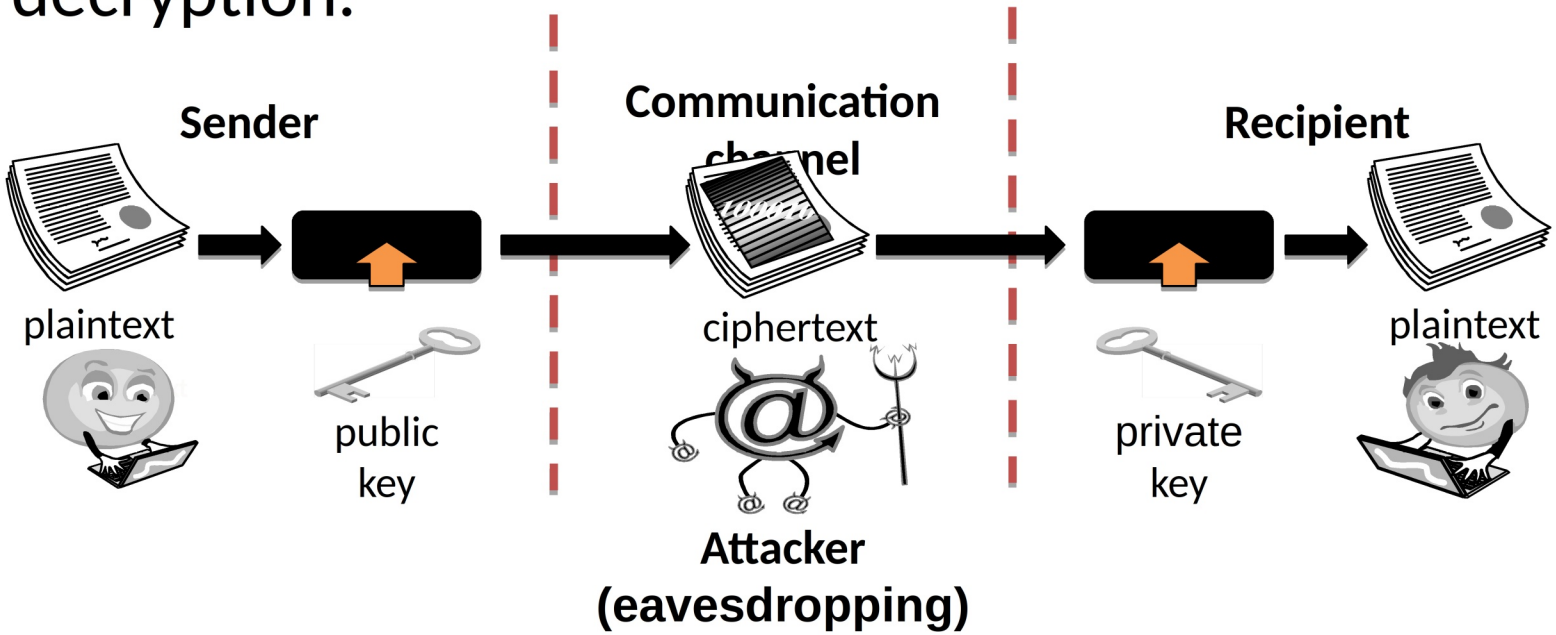
- Bob has two keys: a **private key,** SB, which Bob keeps secret, and a **public key,** PB, which Bob broadcasts widely.
  - In order for Alice to send an encrypted message to Bob, she need only obtain his public key, PB, use that to encrypt her message, M, and send the result, $C = E_{PB}(M)$, to Bob. Bob then uses his secret key to decrypt the message as $M = D_{SB}(C)$.

Public-Key Cryptography

- Separate keys are used for encryption and decryption.

Public-Key Cryptography

- Separate keys are used for encryption and decryption.

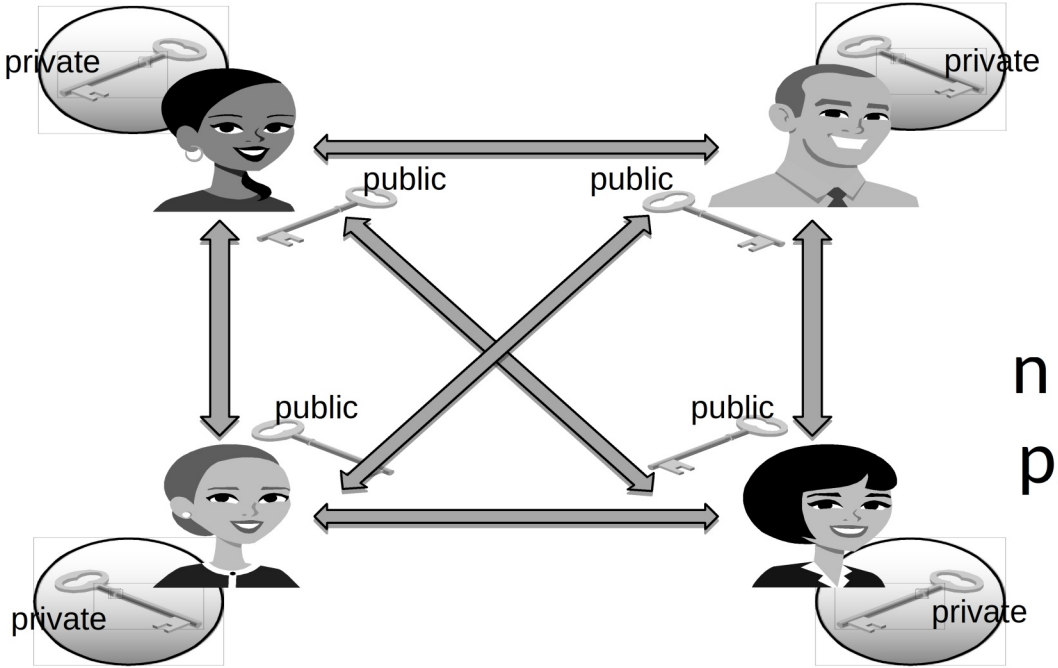


Public Key Distribution

- Only one key is needed for each recipient

Public Key Distribution

- Only one key is needed for each recipient



n key pairs

Public Key conti..

- Examples:
  - Rivest Shamir Adleman (RSA)
    - Recommended key size: 1,024 to 4,096 bit typical
  - ElGamal encryption
- Computationally very expensive
  - Handling large message is ineffecient
  -

Message Authentication

- So far, we covered secrecy of the message i.e. confidentiality.
- message authentication is concerned with:
  - protecting the integrity of a message
  - validating identity of originator
  - non-repudiation of origin (dispute resolution)
- Three alternative functions used:
  - message encryption
  - message authentication code (MAC)
  - hash function

Message Authentication

- Message authentication is a mechanism or service used to verify the integrity of a message.
- Message authentication assures that data received are exactly as sent by (i.e., contain no modification, insertion, deletion, or replay) and
- that the purported identity of the sender is valid.

Authentication Functions

- Message encryption: The ciphertext of the entire message serves as its authenticator
- Message authentication code (MAC): A function of the ***message and a secret key that produces a fixed-length value*** that serves as the authenticator
- Hash function: A function that maps a ***message of any length into a fixed-length hash value***, which serves as the authenticator

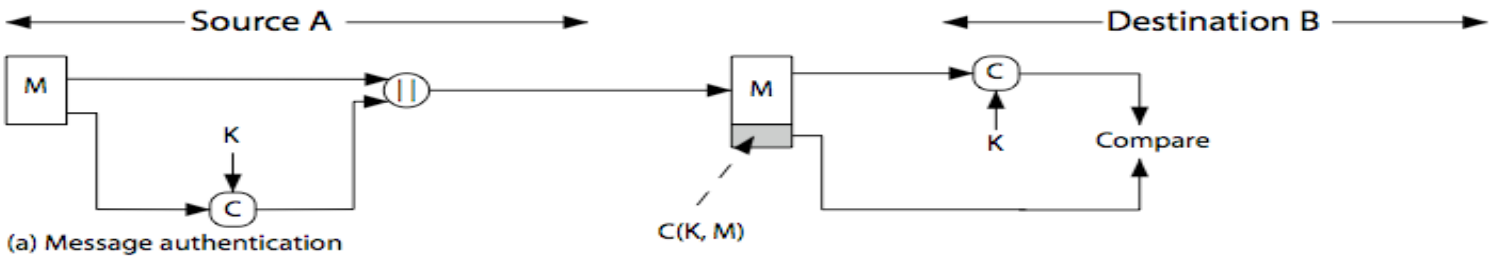
Digital Signatures

- Public-key encryption provides a method for doing digital signatures
- To sign a message, M, Alice just encrypts it with her private key, SA, creating $C = E_{SA}(M)$.
- Anyone can decrypt this message using Alice's public key, as $M' = D_{PA}(C)$, and compare that to the message M.

Message Authentication Code (MAC)

- Allows for Alice and Bob to have data integrity, if they share a secret key.
- Generated by an algorithm that creates a small fixed-sized block depending on both message M and some secret key K s.t. MAC = C(K,M), where
  - M = input message
  - C = MAC function
  - K = shared secret key
  - MAC = message authentication code
- Appended to message as a **signature**
- Receiver performs same computation on message and checks it matches the MAC
- Provides assurance that message is unaltered and comes from sender

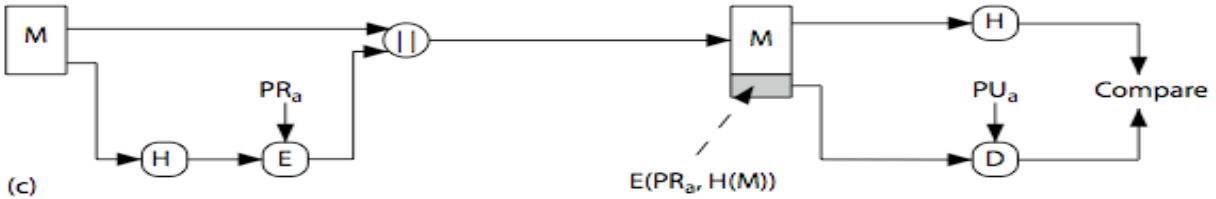
MAC conti...


Source A
Destination B
M
K
C
||
M
C
K
Compare
C(K, M)
(a) Message authentication

Cryptographic Hash Functions

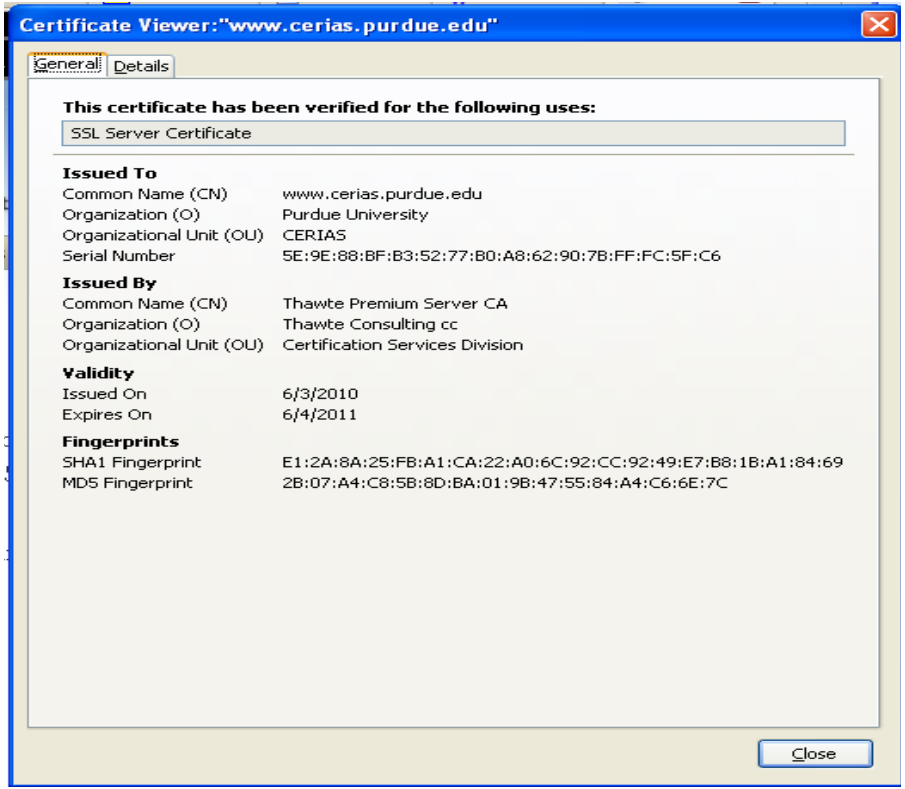
- A checksum on a message, M, that is:
- **One-way**: it should be easy to compute Y=H(M), but hard to find M given only Y
- **Collision-resistant:** it should be hard to find two messages, M and N, such that H(M)=H(N).
- **Examples:** SHA-1, SHA-256.

Application of Hash



Digital Certificates

- **certificate authority (CA)** digitally signs a binding between an identity and the public key for that identity.



Public Distribution of Secret Keys

- Public-key algorithms are slow
- *So we usually want to use symmetric key encryption to protect message contents*
- Hence need to share secret (session) key
- There are alternatives for negotiating a suitable session

Diffie-Hellman Key Exchange

- First public-key type scheme proposed by Diffie & Hellman in 1976 along with the exposition of public key concepts[1]
- Practical method for public exchange of a secret key
- Used in a number of real-world commercial products/protocols

1. Now know that Williamson (UK CESG) secretly proposed the concept in 1970

Diffie-Hellman Key Exchange

- a public-key distribution scheme
  - cannot be used to exchange an arbitrary message
  - rather it can establish a common key
  - known only to the two participants
- value of key depends on the participants (and their private and public key information)
- security relies on the difficulty of computing discrete logarithms (similar to factoring) – hard