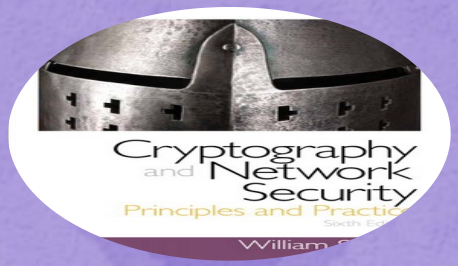
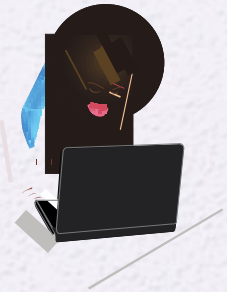


Digital Signatures

Transmit

Alice



Bob's private key

Mess.

Digital signature generation algorithm

S

Bob's signature for *M*

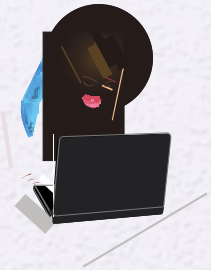
Message *M*

S

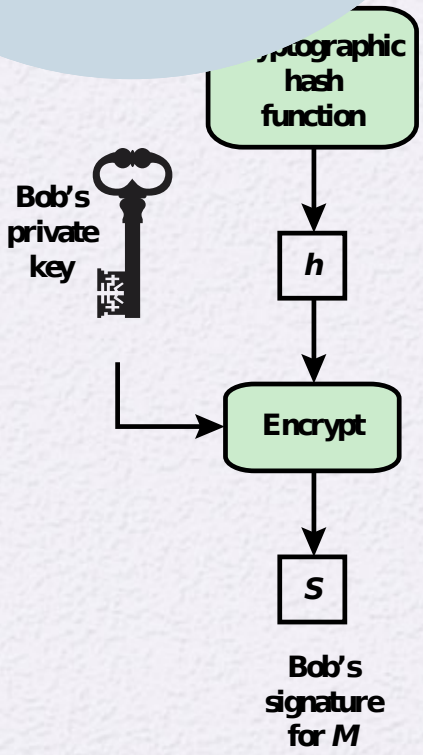
Bob's public key

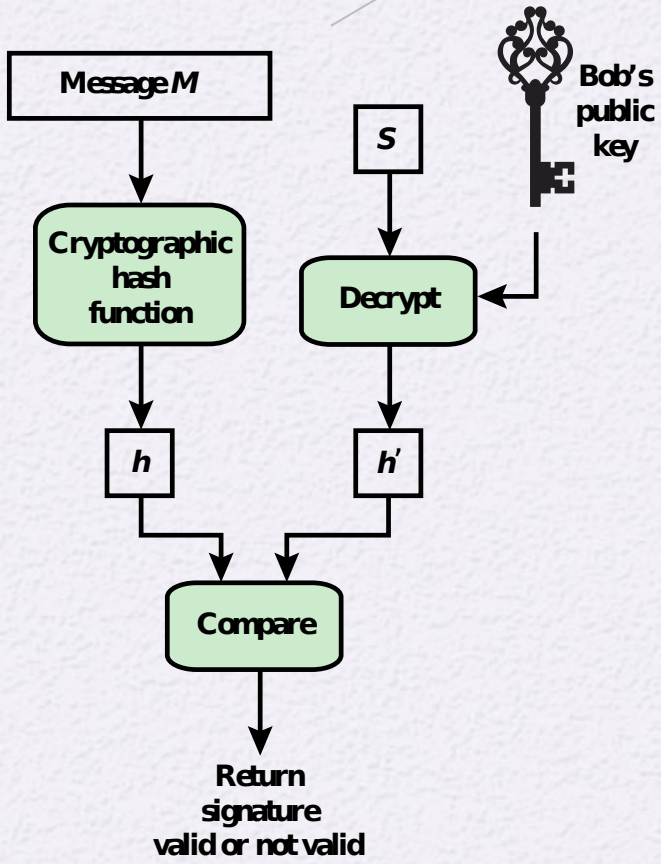
Digital signature verification algorithm

Return signature valid or not valid

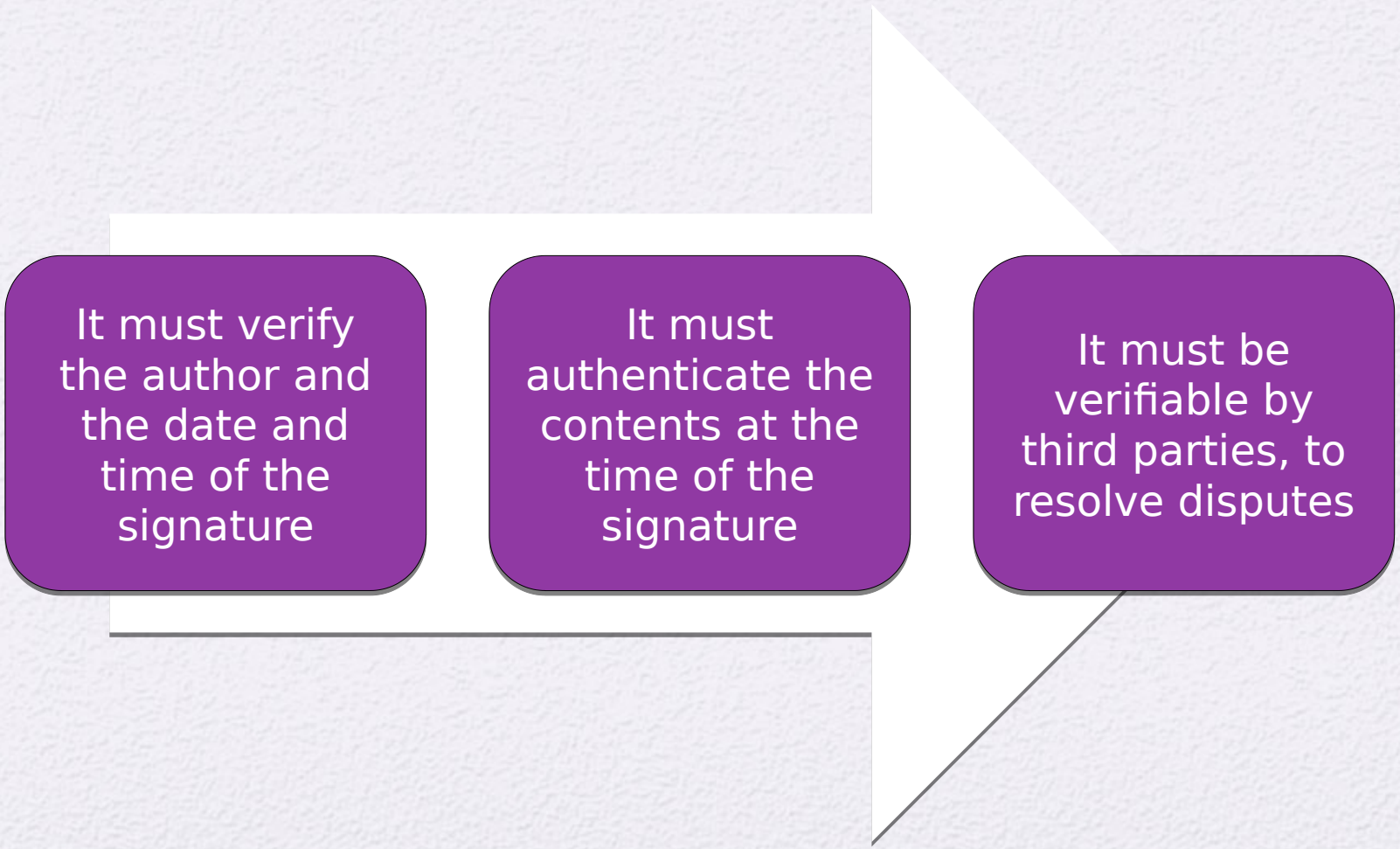


Alice

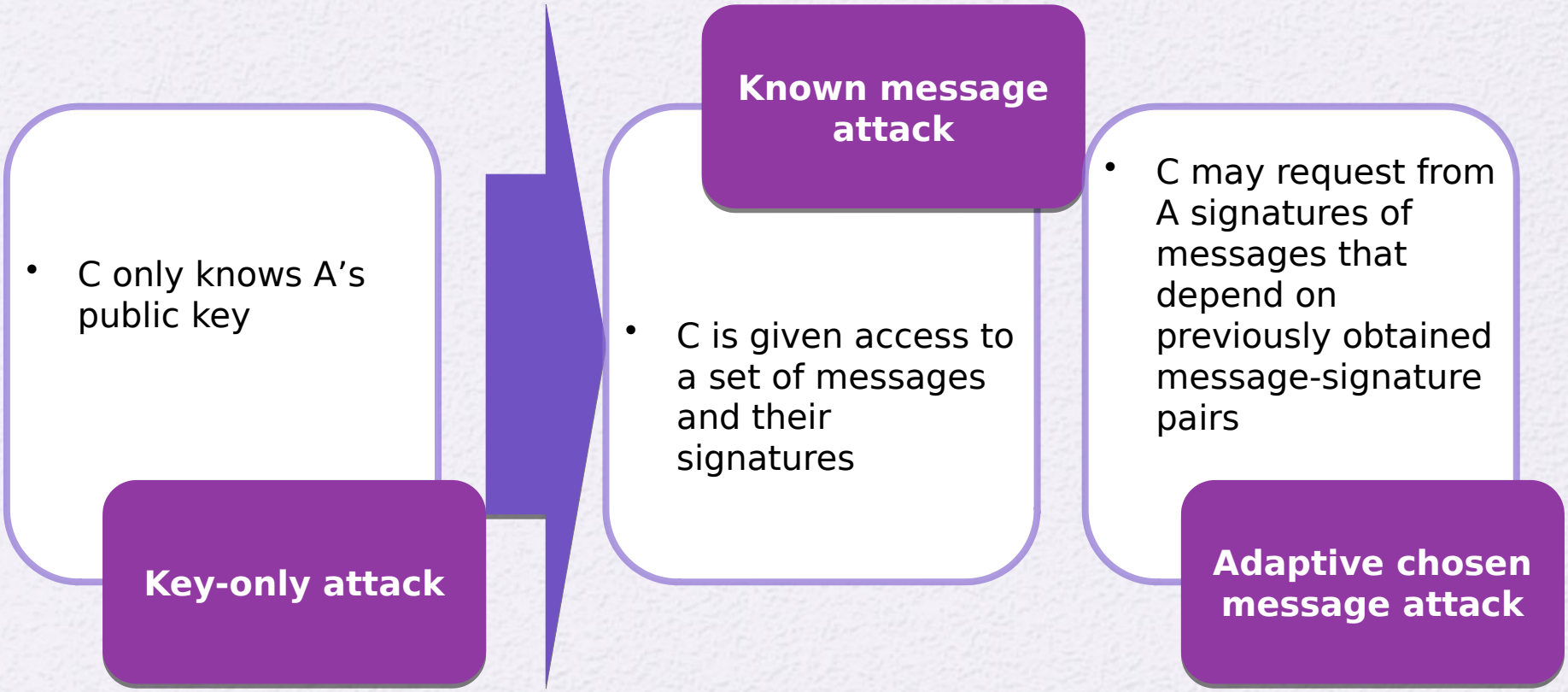

ypographic
hash
function
Bob's
private
key
h
Encrypt
S
Bob's
signature
for M


Message M
Cryptographic
hash
function
h
S
Decrypt
h'
Bob's
public
key
Compare
Return
signature
valid or not valid

Digital Signature Properties



Attacks


C only knows A’s public key
Key-only attack
Known message attack
C is given access to a set of messages and their signatures
C may request from A signatures of messages that depend on previously obtained message-signature pairs
Adaptive chosen message attack

Forgeries

Total break

- C determines A's private key

Universal forgery

- C finds an efficient signing algorithm that provides an equivalent way of constructing signatures on arbitrary messages

Selective forgery

- C forges a signature for a particular message chosen by C

Existential forgery

- C forges a signature for at least one message; C has no control over the messag e

Digital Signature Requirements

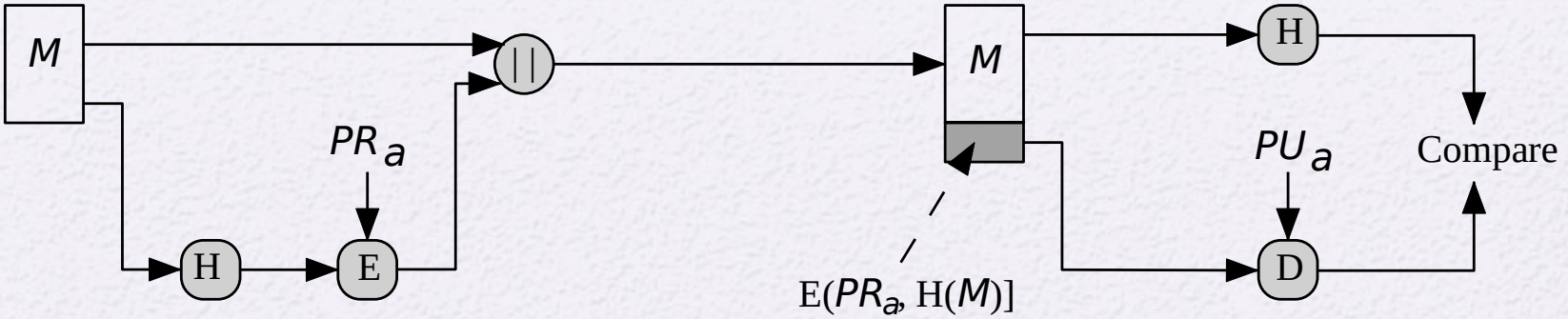
- The signature must be a bit pattern that depends on the message being signed
- The signature must use some information unique to the sender to prevent both forgery and denial
- It must be relatively easy to produce the digital signature
- It must be relatively easy to recognize and verify the digital signature
- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message
- It must be practical to retain a copy of the digital signature in storage

Direct Digital Signature

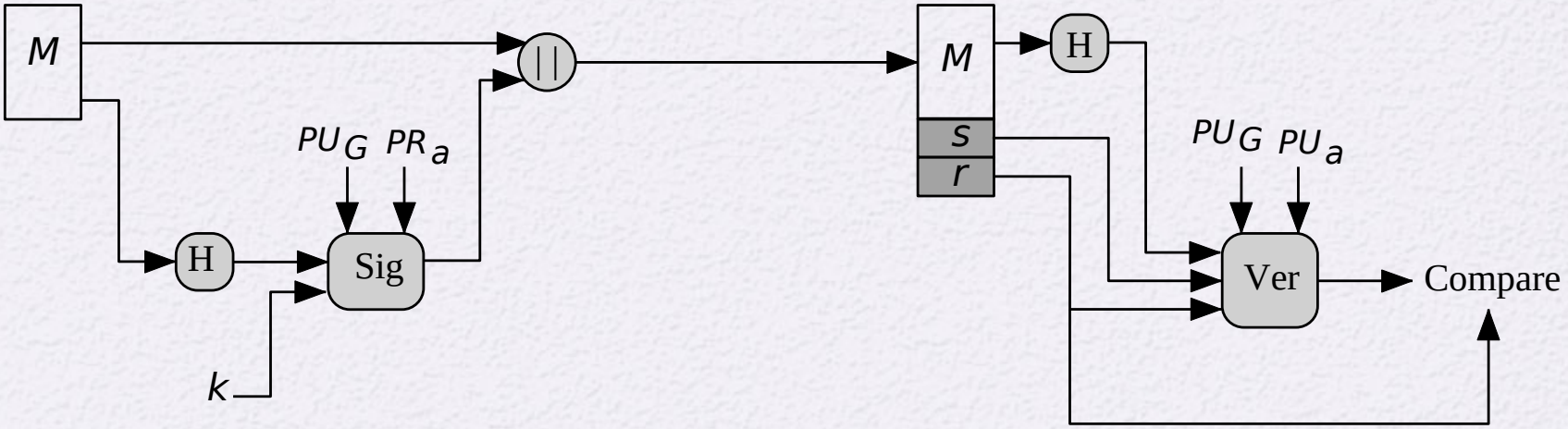
- Refers to a digital signature scheme that involves only the communicating parties
  - It is assumed that the destination knows the public key of the source
- Confidentiality can be provided by encrypting the entire message plus signature with a shared secret key
  - It is important to perform the signature function first and then an outer confidentiality function
  - In case of dispute some third party must view the message and its signature
- The validity of the scheme depends on the security of the sender's private key
  - If a sender later wishes to deny sending a particular message, the sender can claim that the private key was lost or stolen and that someone else forged his or her signature
  - One way to thwart or at least weaken this ploy is to require every signed message to include a timestamp and to require prompt reporting of compromised keys to a central authority

NIST Digital Signature Algorithm

- Published by NIST as Federal Information Processing Standard FIPS 186
- Makes use of the Secure Hash Algorithm (SHA)
- The latest version, FIPS 186-3, also incorporates digital signature algorithms based on RSA an elliptic curve cryptography



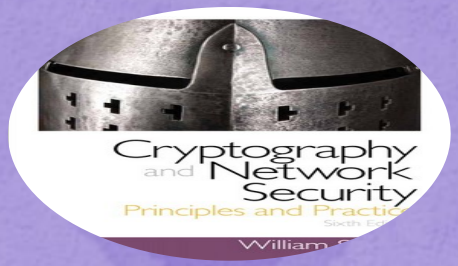
(a) RSA Approach



(b) DSA Approach

RSA-PSS

- RSA Probabilistic Signature Scheme
- Included in the 2009 version of FIPS 186
- Latest of the RSA schemes and the one that RSA Laboratories recommends as the most secure of the RSA schemes
- For all schemes developed prior to PSS is has not been possible to develop a mathematical proof that the signature scheme is as secure as the underlying RSA encryption/decryption primitive
- The PSS approach was first proposed by Bellare and Rogaway
- This approach, unlike the other RSA-based schemes, introduces a randomization process that enables the security of the method to be shown to be closely related to the security of the RSA algorithm itself



Key Management and Distribution

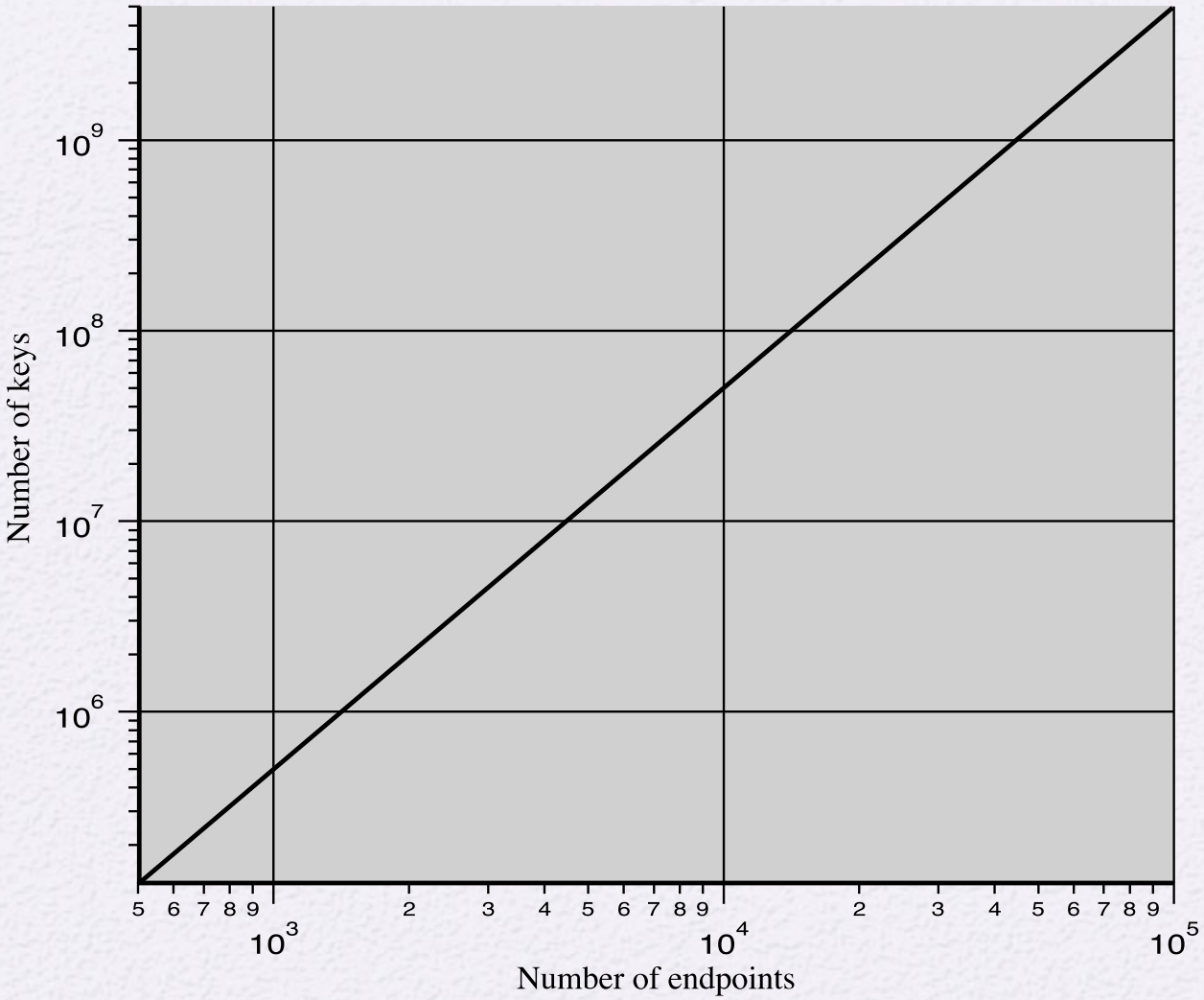
Key Distribution Technique

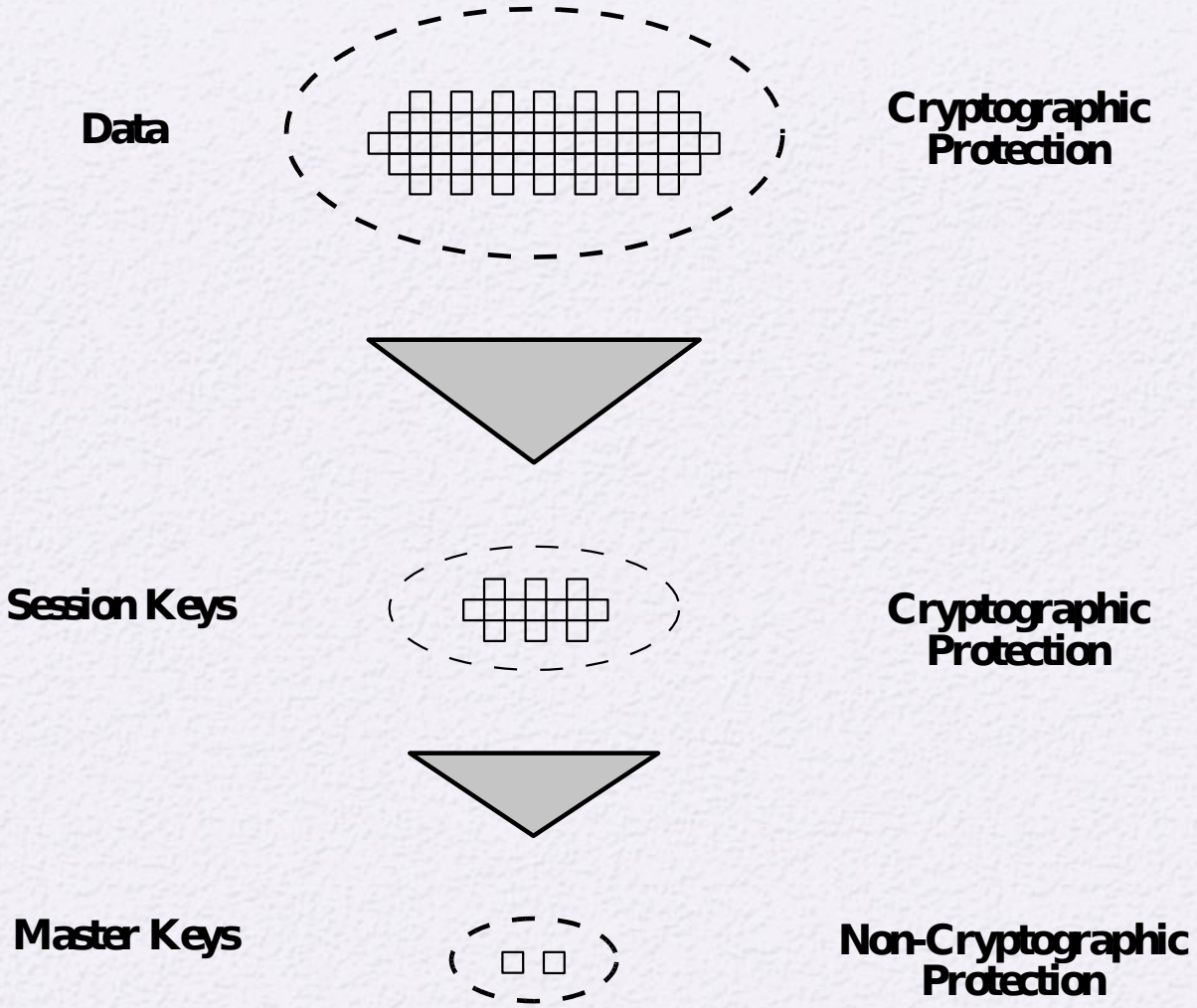
- Term that refers to the means of delivering a key to two parties who wish to exchange data without allowing others to see the key
- For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others
- Frequent key changes are desirable to limit the amount of data compromised if an attacker learns the key

Symmetric Key Distribution

Given parties A and B, key distribution can be achieved in a number of ways:

- **A can select a key and physically deliver it to B**
- **A third party can select the key and physically deliver it to A and B**
- **If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key**
- **If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B**


Number of keys
10^9
10^8
10^7
10^6
5 6 7 8 9
10^3
2 3 4 5 6 7 8 9
10^4
2 3 4 5 6 7 8 9
10^5
Number of endpoints


Data
Cryptographic Protection
Session Keys
Cryptographic Protection
Master Keys
Non-Cryptographic Protection

Hierarchical Key Control

- For communication among entities within the same local domain, the local KDC is responsible for key distribution
  - If two entities in different domains desire a shared key, then the corresponding local KDC's can communicate through a global KDC
- The hierarchical concept can be extended to three or more layers
- Scheme minimizes the effort involved in master key distribution because most master keys are those shared by a local KDC with its local entities
  - Limits the range of a faulty or subverted KDC to its local area only

Session Key Lifetime

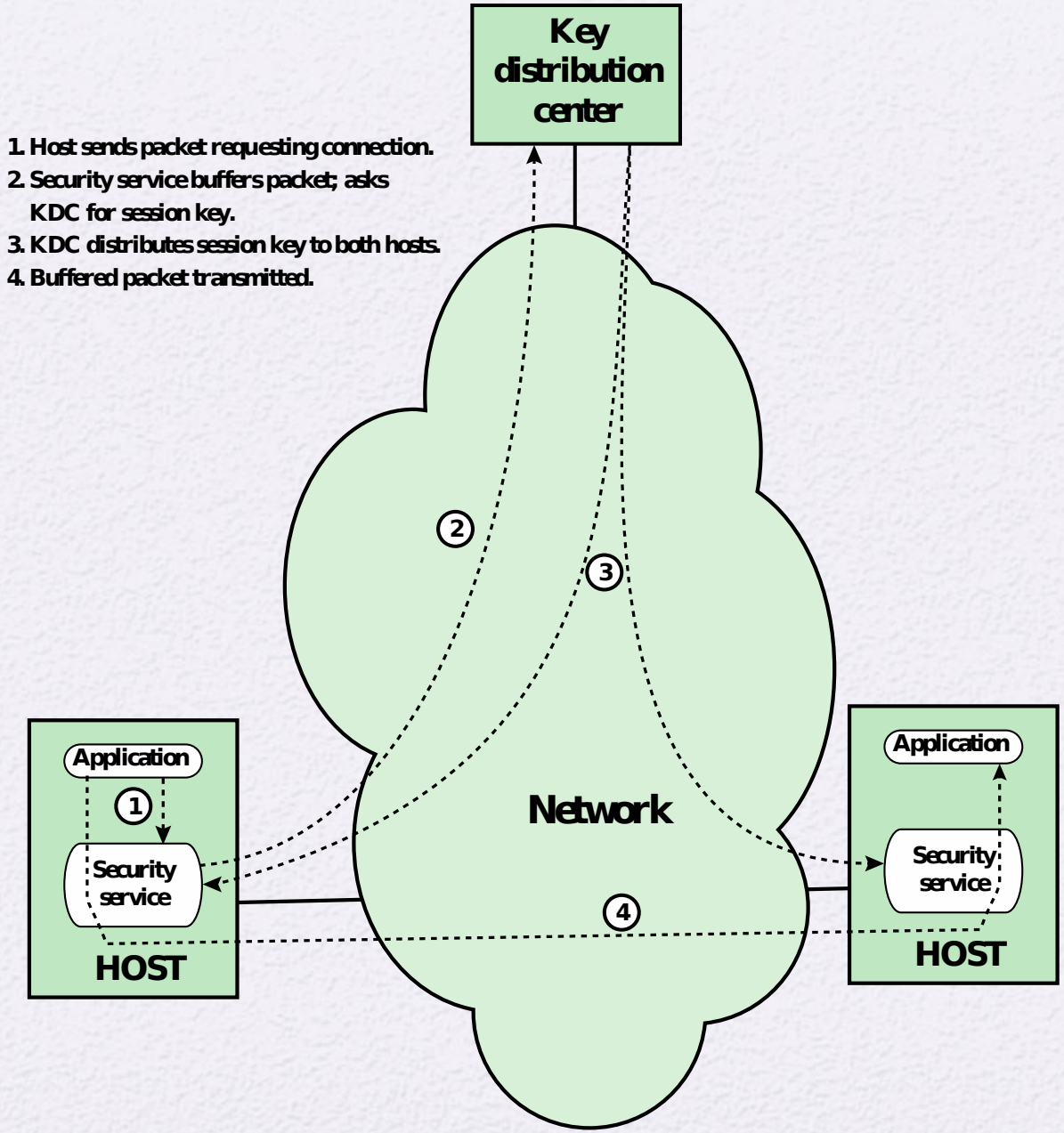
For connection-oriented protocols one choice is to use the same session key for the length of time that the connection is open; using a new session key for each new session

For a connectionless protocol there is no explicit connection initiation or termination, thus it is not obvious how often one needs to change the session key

A security manager must balance competing considerations:

The more frequently session keys are exchanged, the more secure they are

The distribution of session keys delays the start of any exchange and places a burden on network capacity


Key distribution center
1. Host sends packet requesting connection.
2. Security service buffers packet; asks KDC for session key.
3. KDC distributes session key to both hosts.
4. Buffered packet transmitted.
2
3
Network
Application
1
Security service
HOST
Application
Security service
HOST
4

Controlling Key Usage

- The concept of a key hierarchy and the use of automated key distribution techniques greatly reduce the number of keys that must be manually managed and distributed
- It also may be desirable to impose some control on the way in which automatically distributed keys are used
  - For example, in addition to separating master keys from session keys, we may wish to define different types of session keys on the basis of use