Southern Connecticut State University

Computer Networking and Security

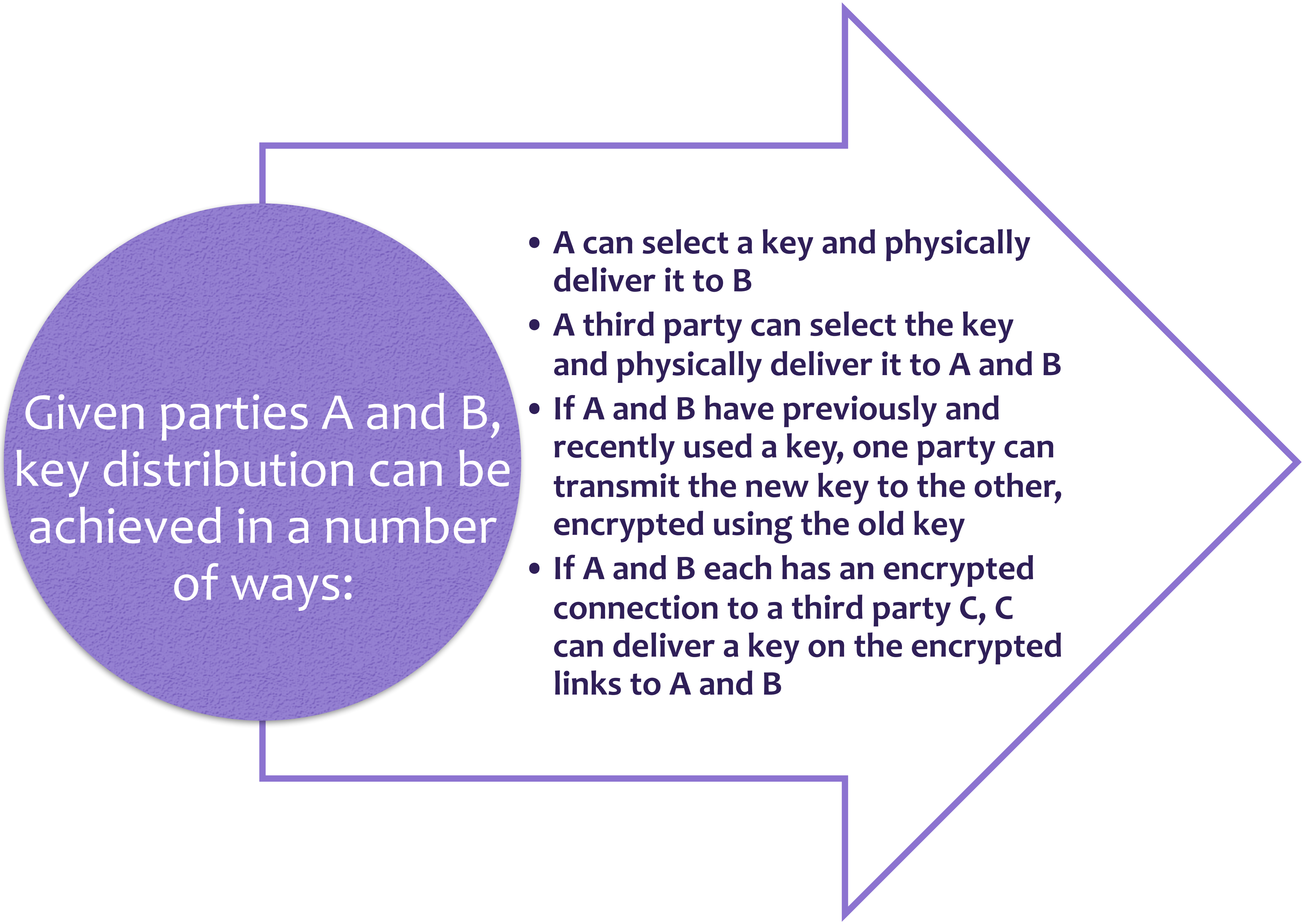
Instructor: Dr. Hao Wu

Week 13 Key Management

Key Distribution Technique

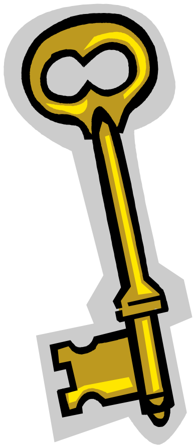
- Term that refers to the means of delivering a key to two parties who wish to exchange data without allowing others to see the key
- For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others
- Frequent key changes are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

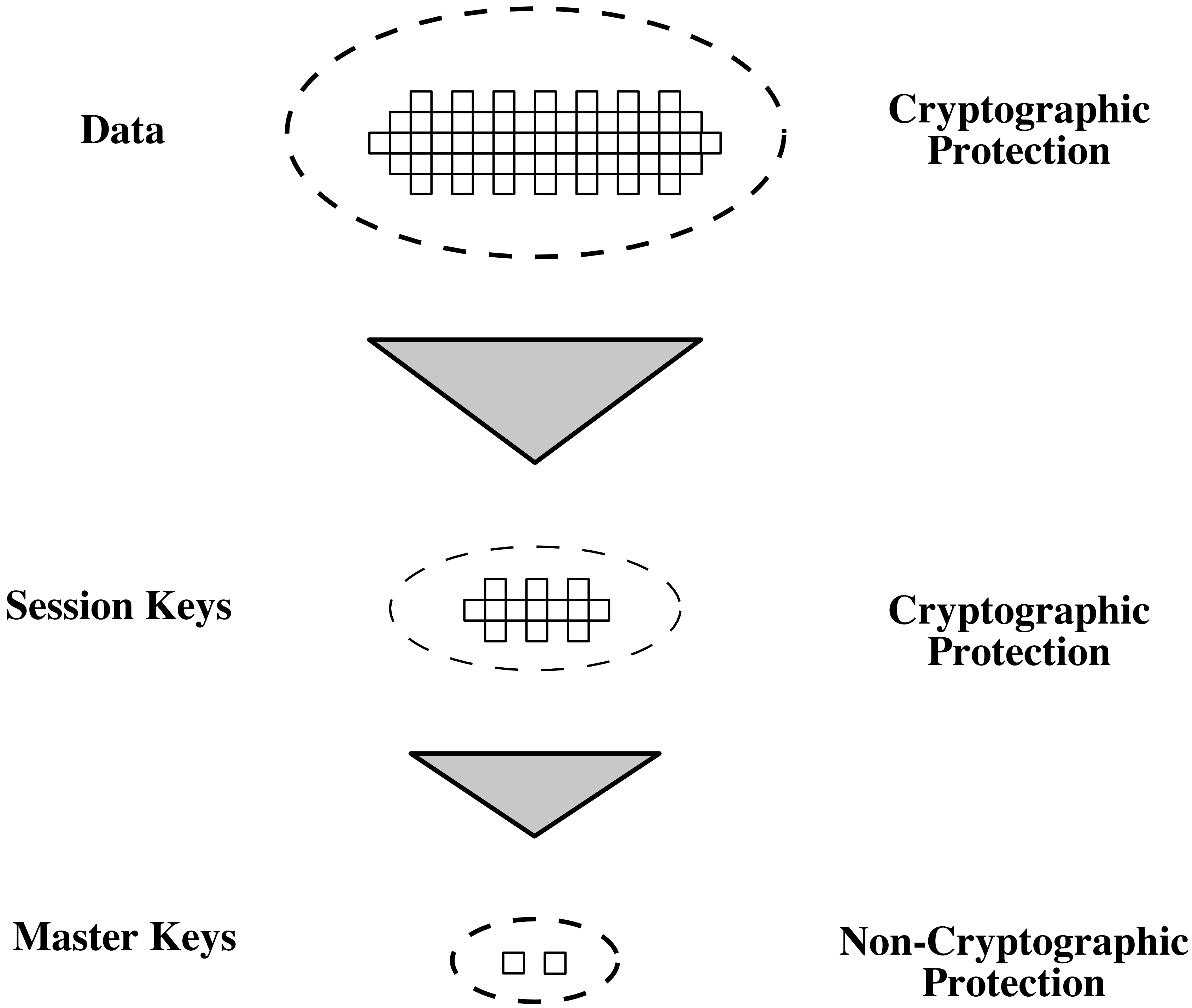
Symmetric Key Distribution

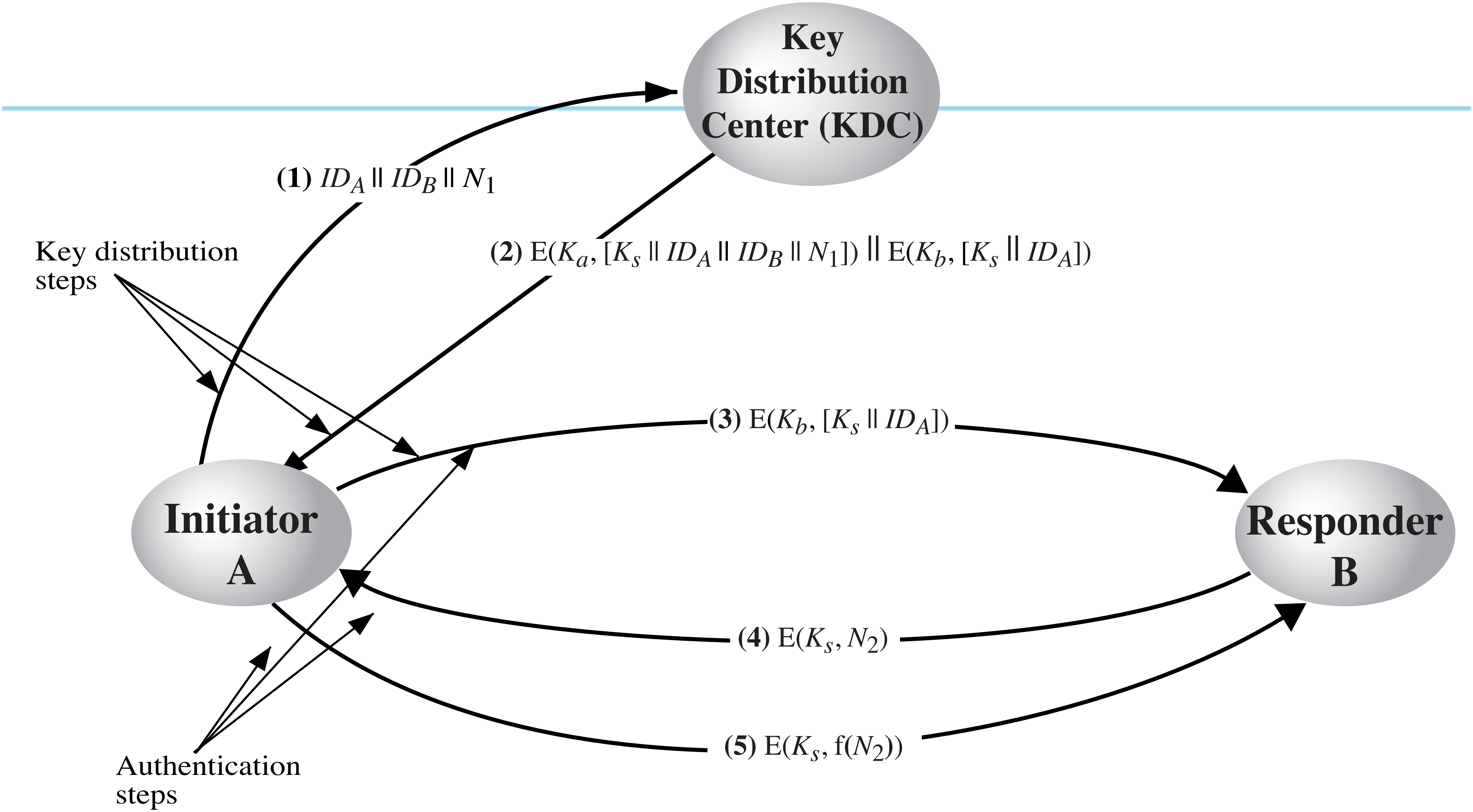


Given parties A and B, key distribution can be achieved in a number of ways:

- 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




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

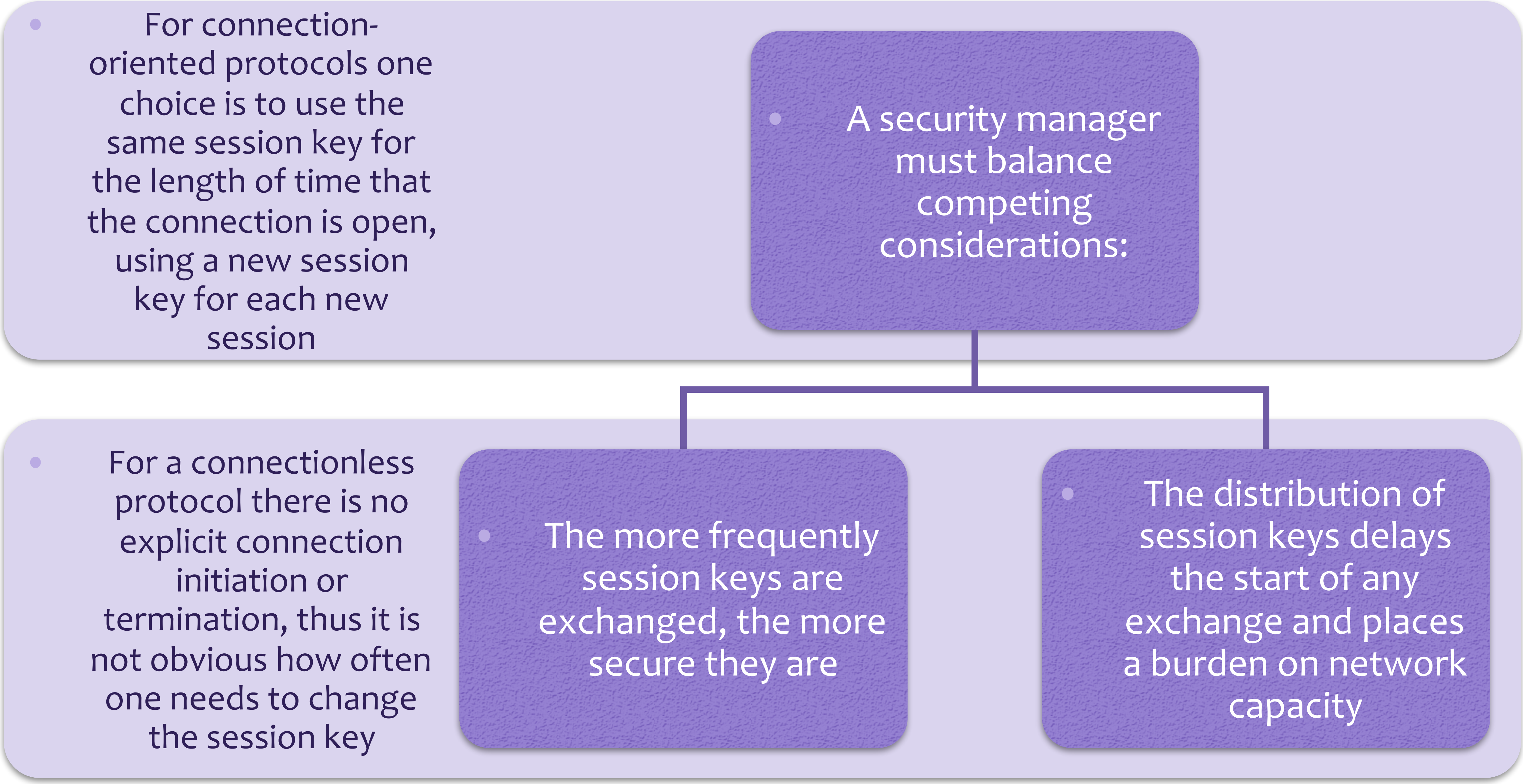

Key Distribution Center (KDC)
(1) $ID_A \parallel ID_B \parallel N_1$
Key distribution steps
(2) $E(K_a, [K_s \parallel ID_A \parallel ID_B \parallel N_1]) \parallel E(K_b, [K_s \parallel ID_A])$
(3) $E(K_b, [K_s \parallel ID_A])$
Initiator A
Responder B
(4) $E(K_s, N_2)$
(5) $E(K_s, f(N_2))$
Authentication steps

Key Distribution Scenario

Hierarchical Key Control

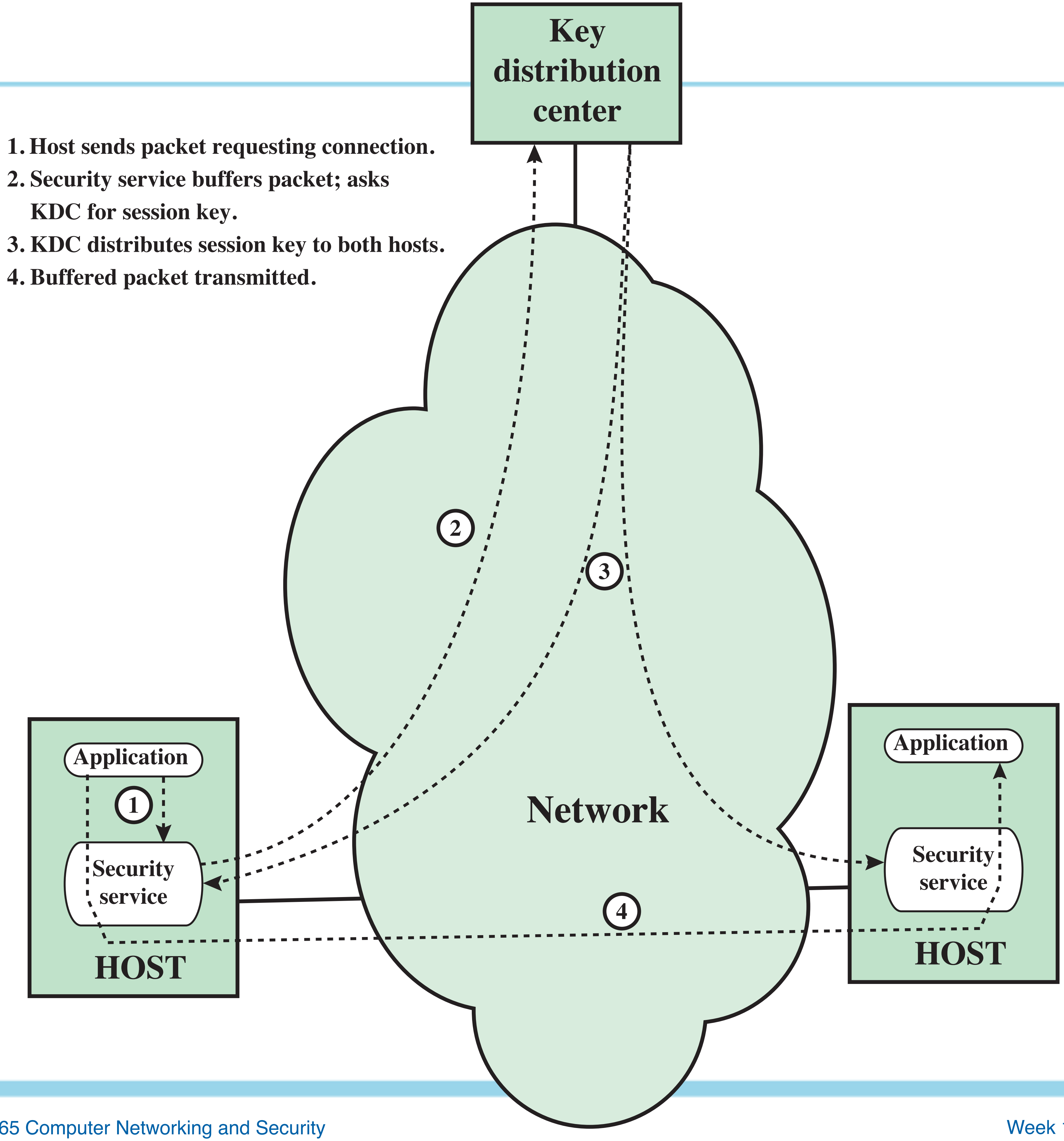
- For communication among entities within the same local domain, the local KDC is responsible for key distribution
  - If two entities in different domains desire a shared key, then the corresponding local KDC’s can communicate through a global KDC
- The hierarchical concept can be extended to three 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

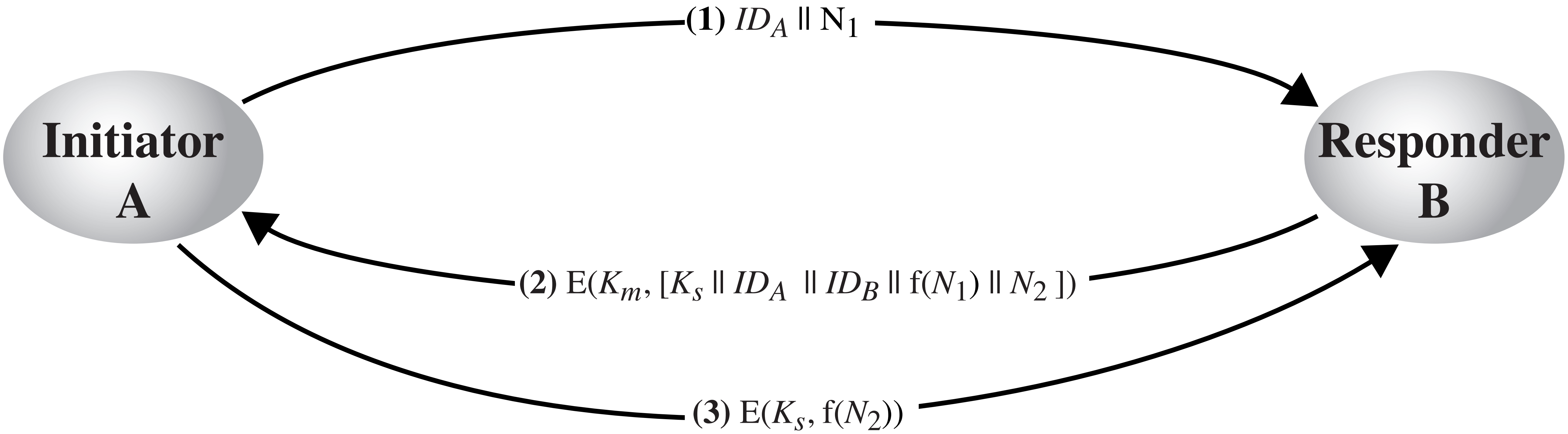


- For connection-oriented protocols one choice is to use the same session key for the length of time that the connection is open, using a new session key for each new session
- For a connectionless protocol there is no explicit connection initiation or termination, thus it is not obvious how often one 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



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.


Initiator
A
Responder
B
(1) $ID_A \parallel N_1$
(2) $E(K_m, [K_s \parallel ID_A \parallel ID_B \parallel f(N_1) \parallel N_2])$
(3) $E(K_s, f(N_2))$

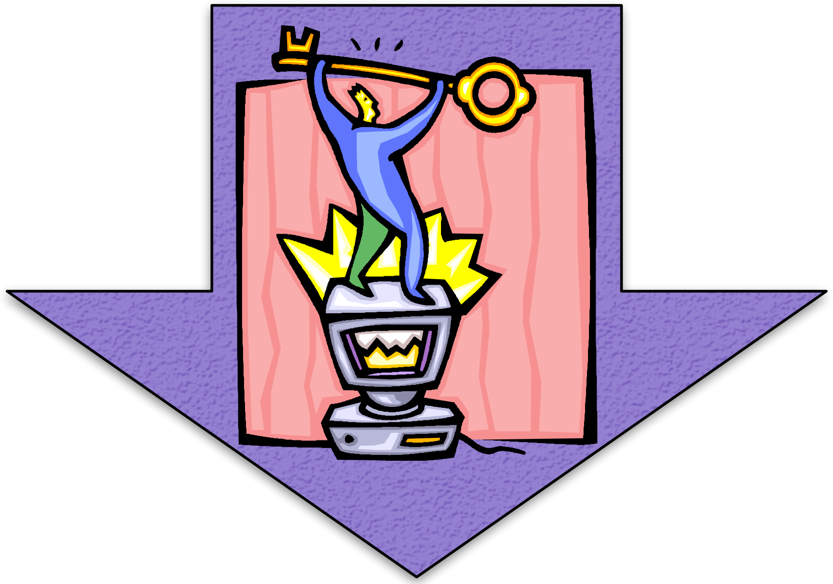
Decentralized Key Distribution

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

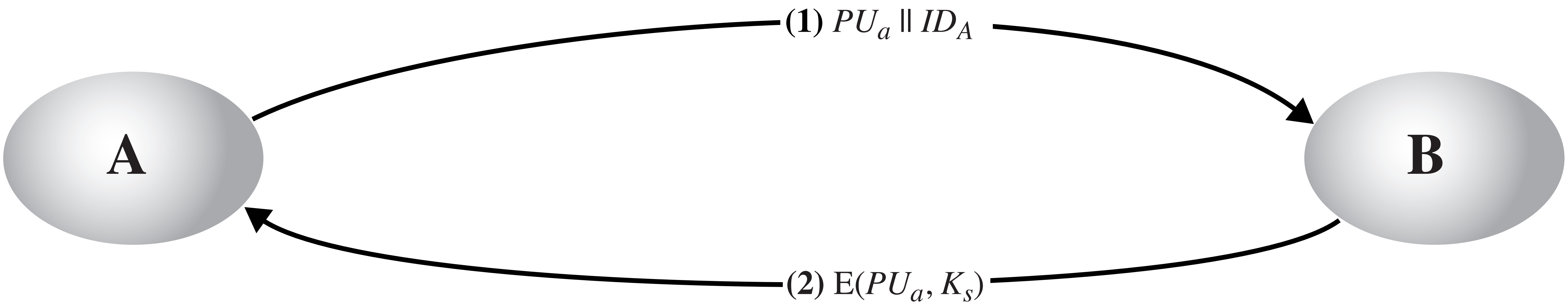
Key Controls

- Associate a tag with each key
  - For use with DES and makes use of the extra 8 bits in each 64-bit DES key
  - The eight non-key bits ordinarily reserved for parity checking form the key tag
  - Because the tag is embedded in the key, it is encrypted along with the key when that key is distributed, thus providing protection

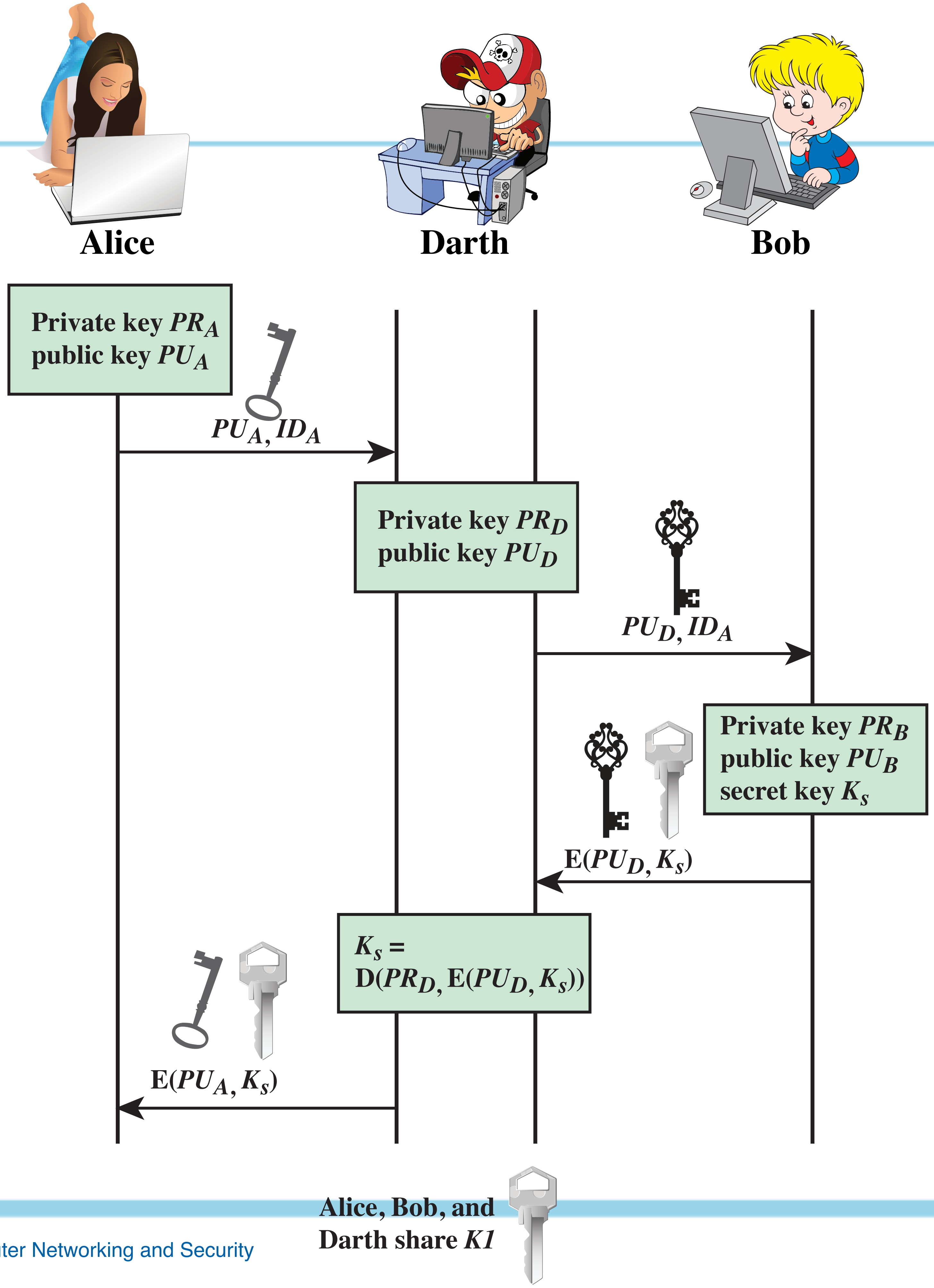


Drawbacks:

- The tag length is limited to 8 bits, limiting its flexibility and functionality
- Because the tag is not transmitted in clear form, it can be used only at the point of decryption, limiting the ways in which key use can be controlled



Simple Use of Public-Key Encryption to Establish a Session Key



Private key PR_A
public key PU_A

PU_A, ID_A

Private key PR_D
public key PU_D

PU_D, ID_A

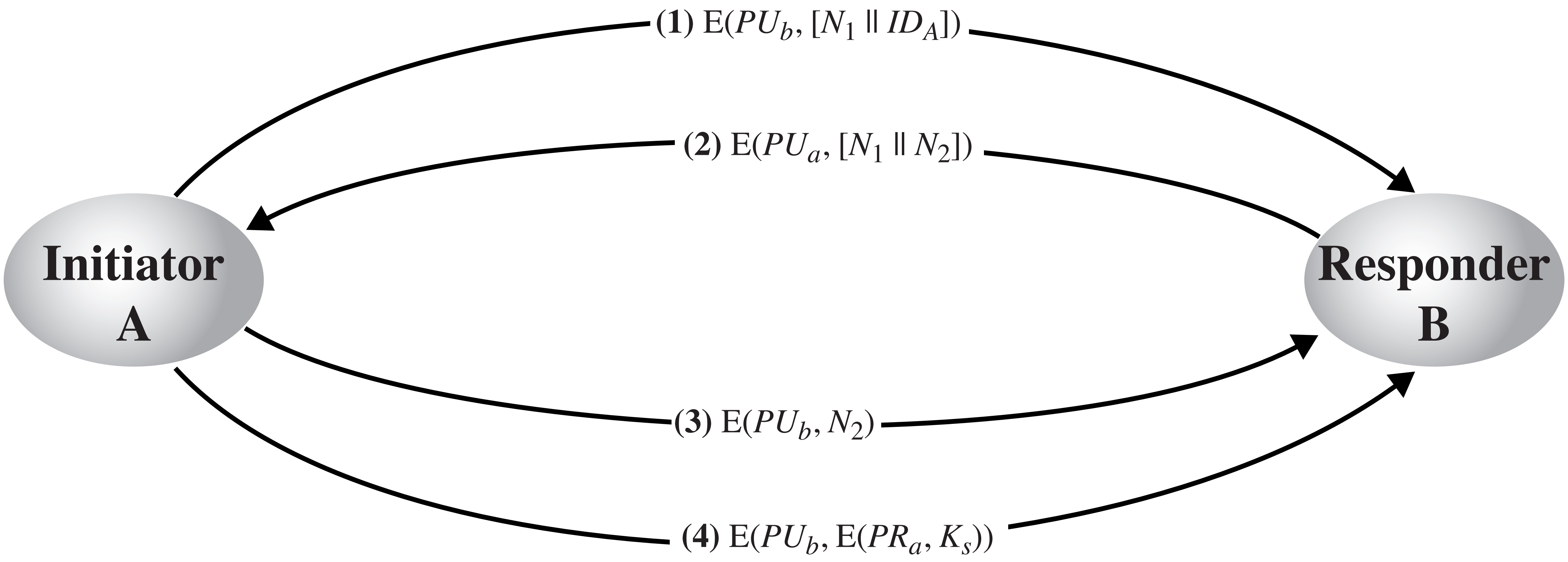
Private key PR_B
public key PU_B
secret key K_s

$E(PU_D, K_s)$

$K_s =$
$D(PR_D, E(PU_D, K_s))$

$E(PU_A, K_s)$

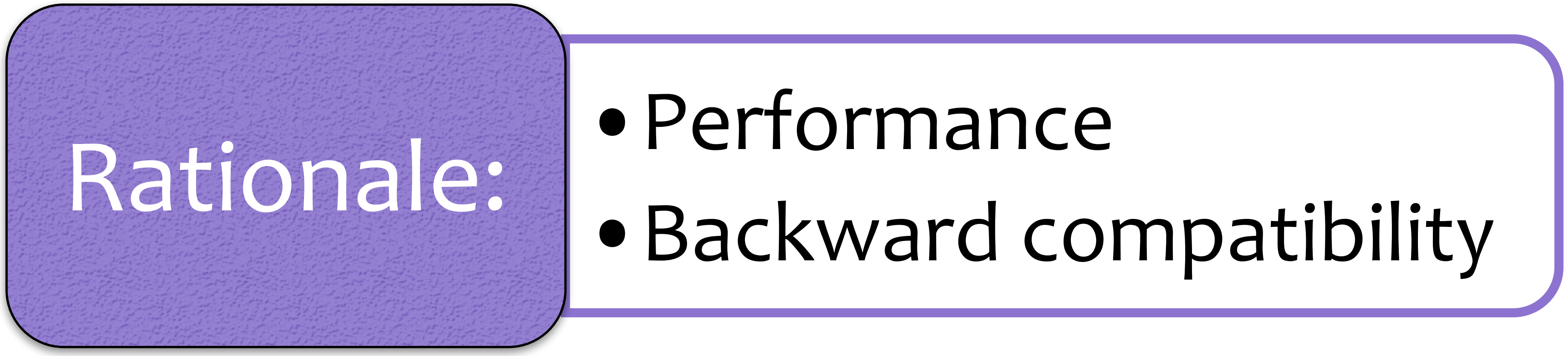
Alice, Bob, and
Darth share *K1*


(1) E(PU_b, [N_1 ‖ ID_A])
(2) E(PU_a, [N_1 ‖ N_2])
Initiator A
Responder B
(3) E(PU_b, N_2)
(4) E(PU_b, E(PR_a, K_s))

Public-Key Distribution of Secret Keys

A Hybrid Scheme

- In use on IBM mainframes
- Retains the use of a key distribution center (KDC) that shares a secret master key with each user and distributes secret session keys encrypted with the master key
- A public-key scheme is used to distribute the master keys

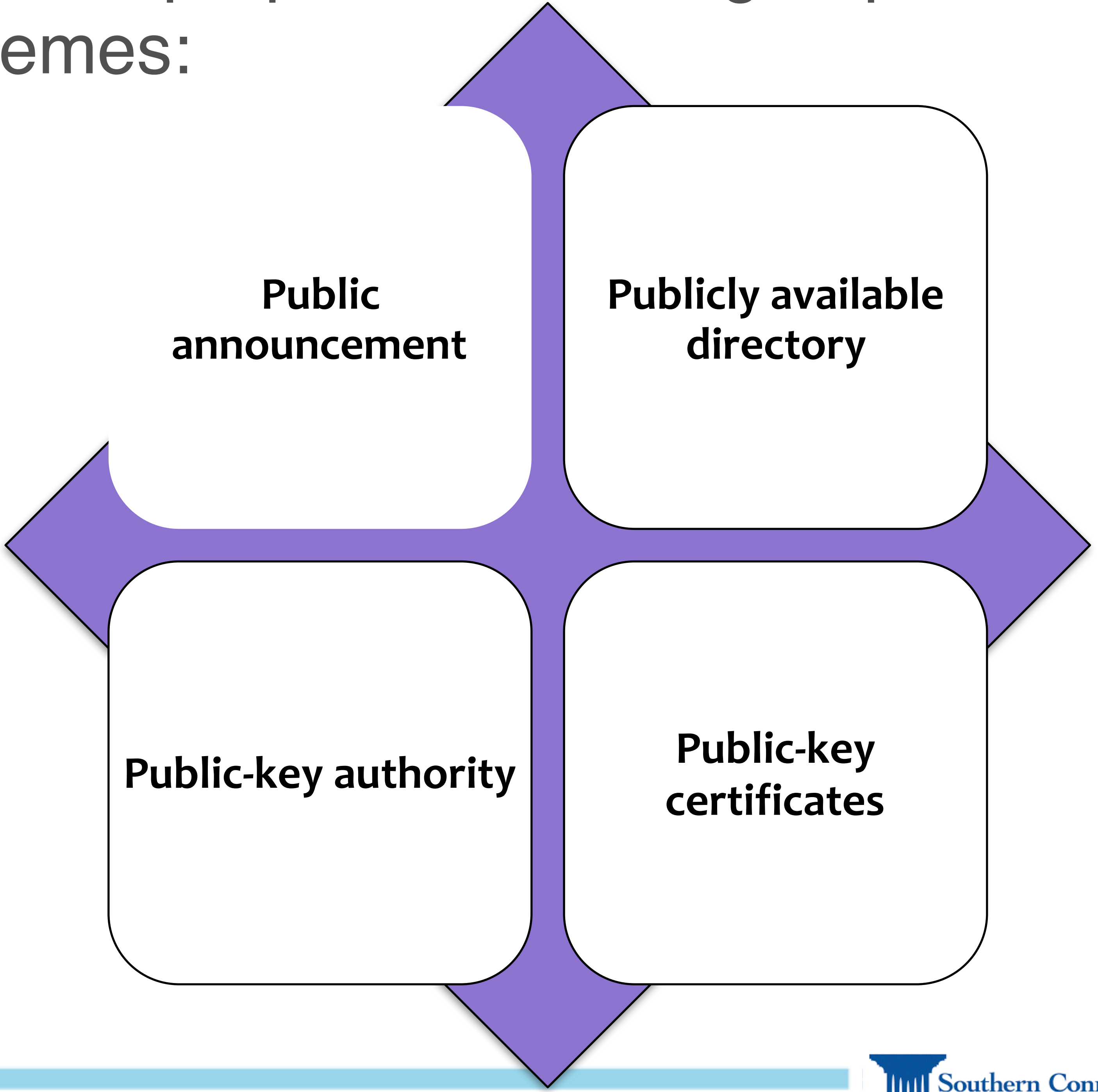


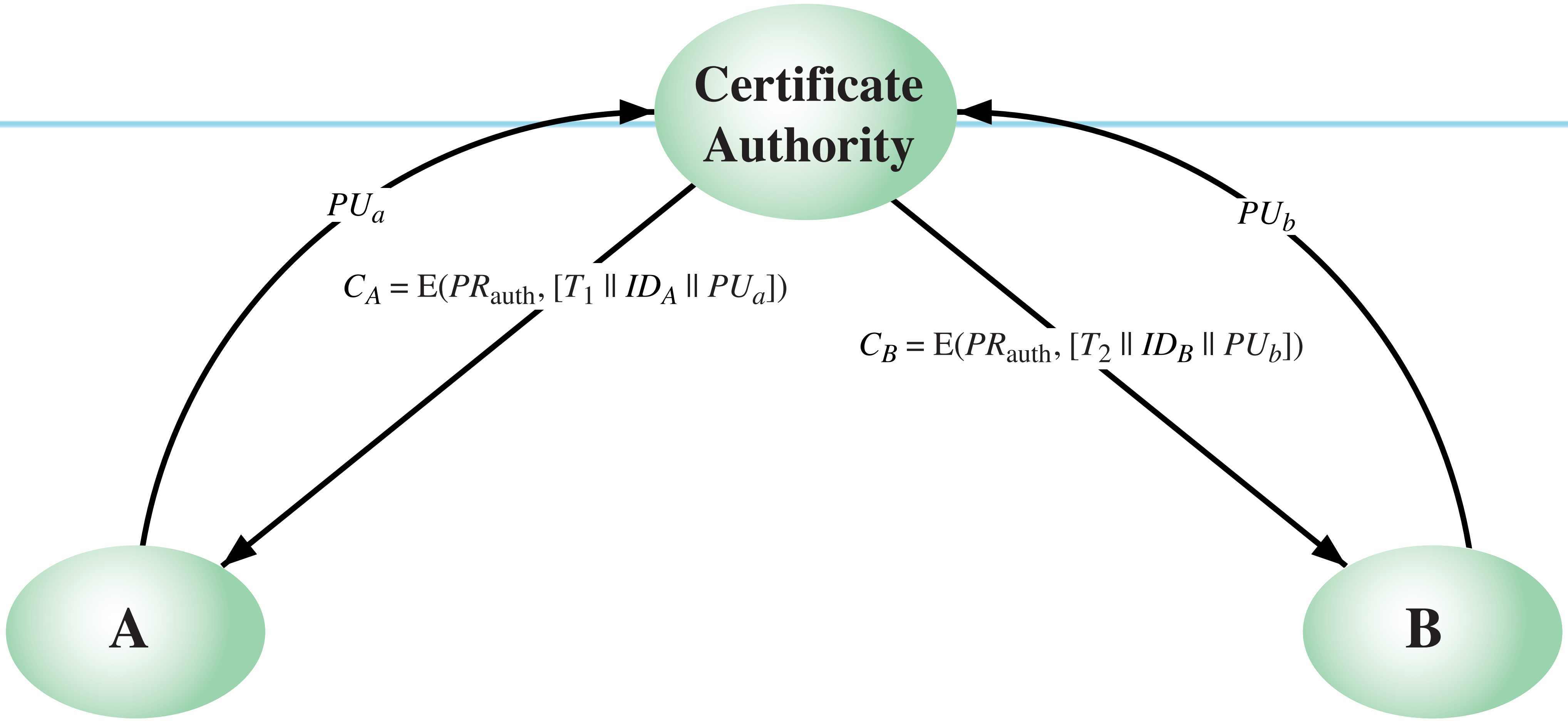
Rationale:

- Performance
- Backward compatibility

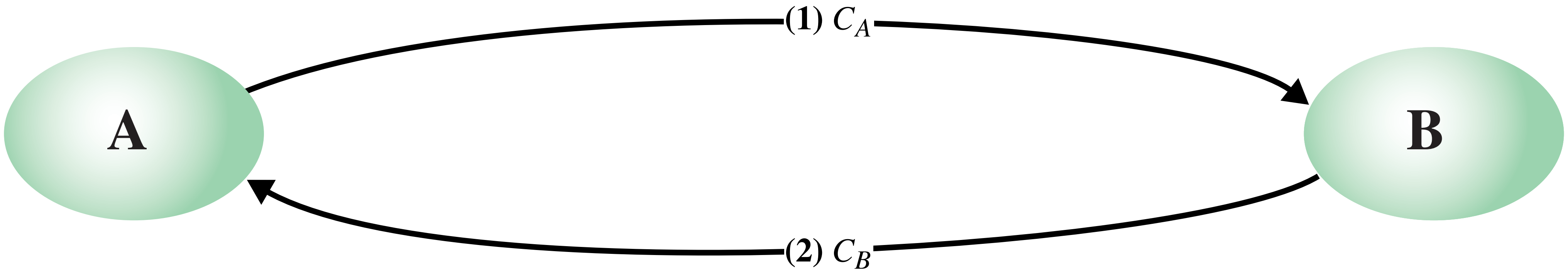
Distribution of Public Keys

- Several techniques have been proposed for the distribution of public keys. Virtually all these proposals can be grouped into the following general schemes:




Certificate Authority
PU_a
$C_A = E(PR_{auth}, [T_1 \| ID_A \| PU_a])$
PU_b
$C_B = E(PR_{auth}, [T_2 \| ID_B \| PU_b])$
A
B

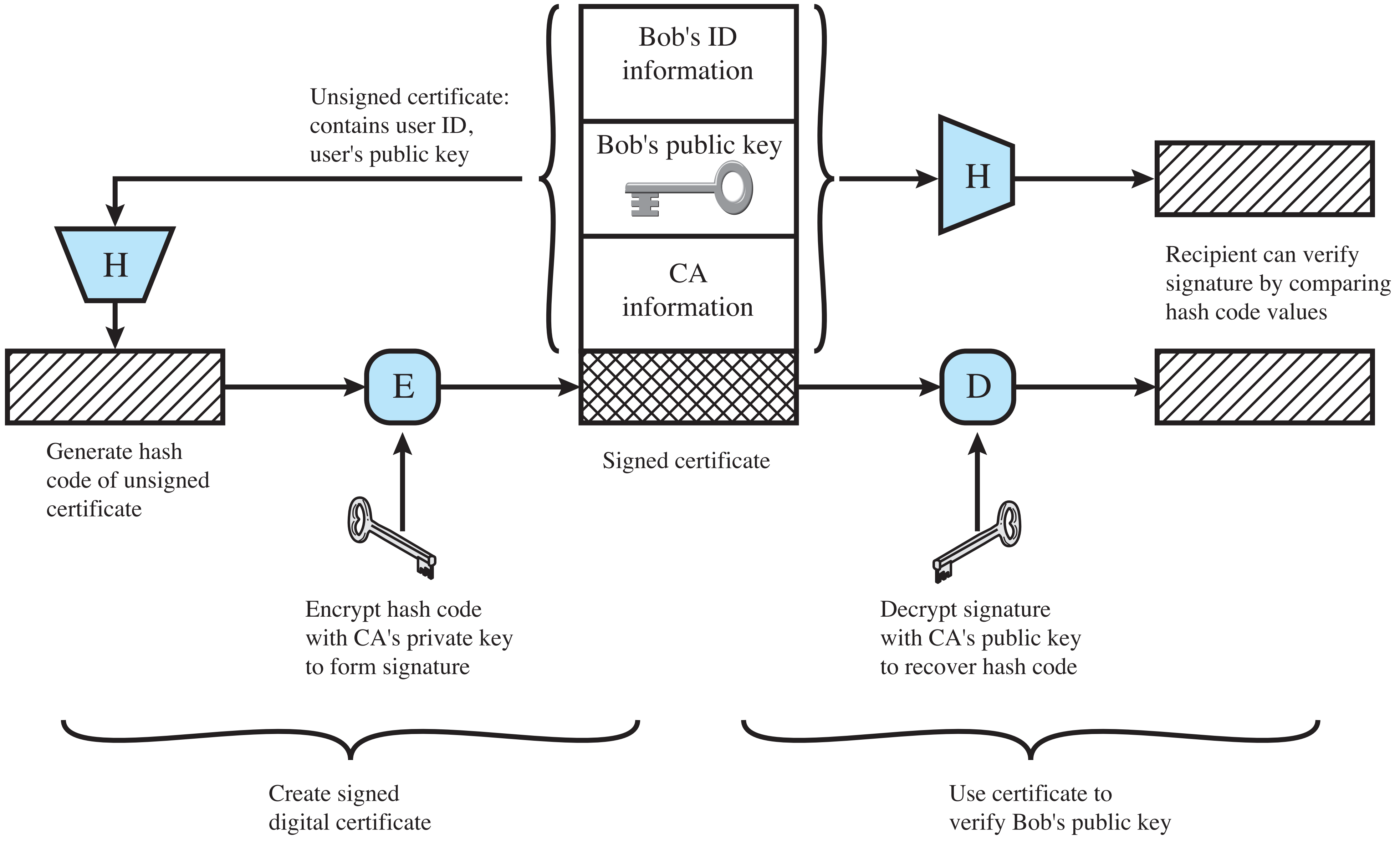
(a) Obtaining certificates from CA



(b) Exchanging certificates

X.509 Certificates

- Part of the X.500 series of recommendations that define a directory service
  - The directory is, in effect, a server or distributed set of servers that maintains a database of information about users
- X.509 defines a framework for the provision of authentication services by the X. 500 directory to its users
  - Was initially issued in 1988 with the latest revision in 2012
  - Based on the use of public-key cryptography and digital signatures
  - Does not dictate the use of a specific algorithm but recommends RSA
  - Does not dictate a specific hash algorithm
- Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority
- X.509 defines alternative authentication protocols based on the use of public-key certificates

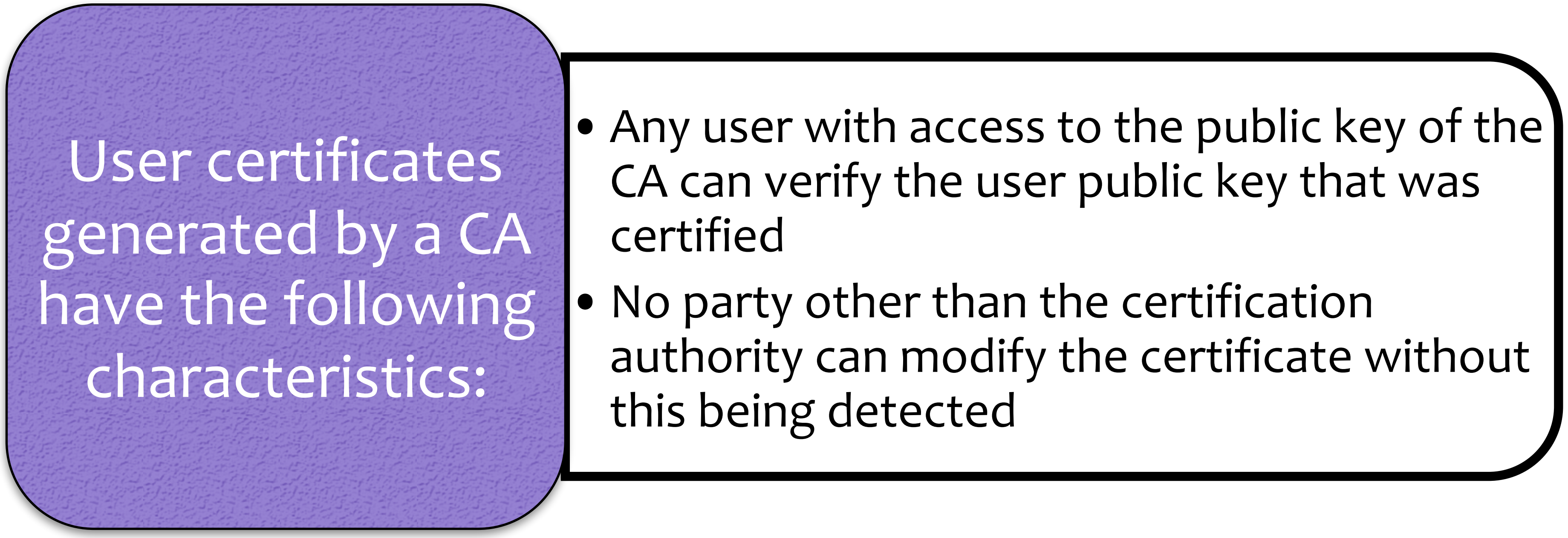

Bob's ID information
Bob's public key
CA information
Unsigned certificate: contains user ID, user's public key
H
Generate hash code of unsigned certificate
E
Encrypt hash code with CA's private key to form signature
Signed certificate
H
Recipient can verify signature by comparing hash code values
D
Decrypt signature with CA's public key to recover hash code
Create signed digital certificate
Use certificate to verify Bob's public key

Certificates

- Version
- Serial number
- Signature algorithm identifier
- Issuer name
- Period of validity
- Subject name
- Subject's public-key information
- Issuer unique identifier
- Subject unique identifier
- Extensions
- Signature

Obtaining a Certificate

- Because certificates are unforgeable, they can be placed in a directory without the need for the directory to make special efforts to protect them
  - In addition, a user can transmit his or her certificate directly to other users
- Once B is in possession of A’s certificate, B has confidence that messages it encrypts with A’s public key will be secure from eavesdropping and that messages signed with A’s private key are unforgeable



User certificates generated by a CA have the following characteristics:

- Any user with access to the public key of the CA can verify the user public key that was certified
- No party other than the certification authority can modify the certificate without this being detected

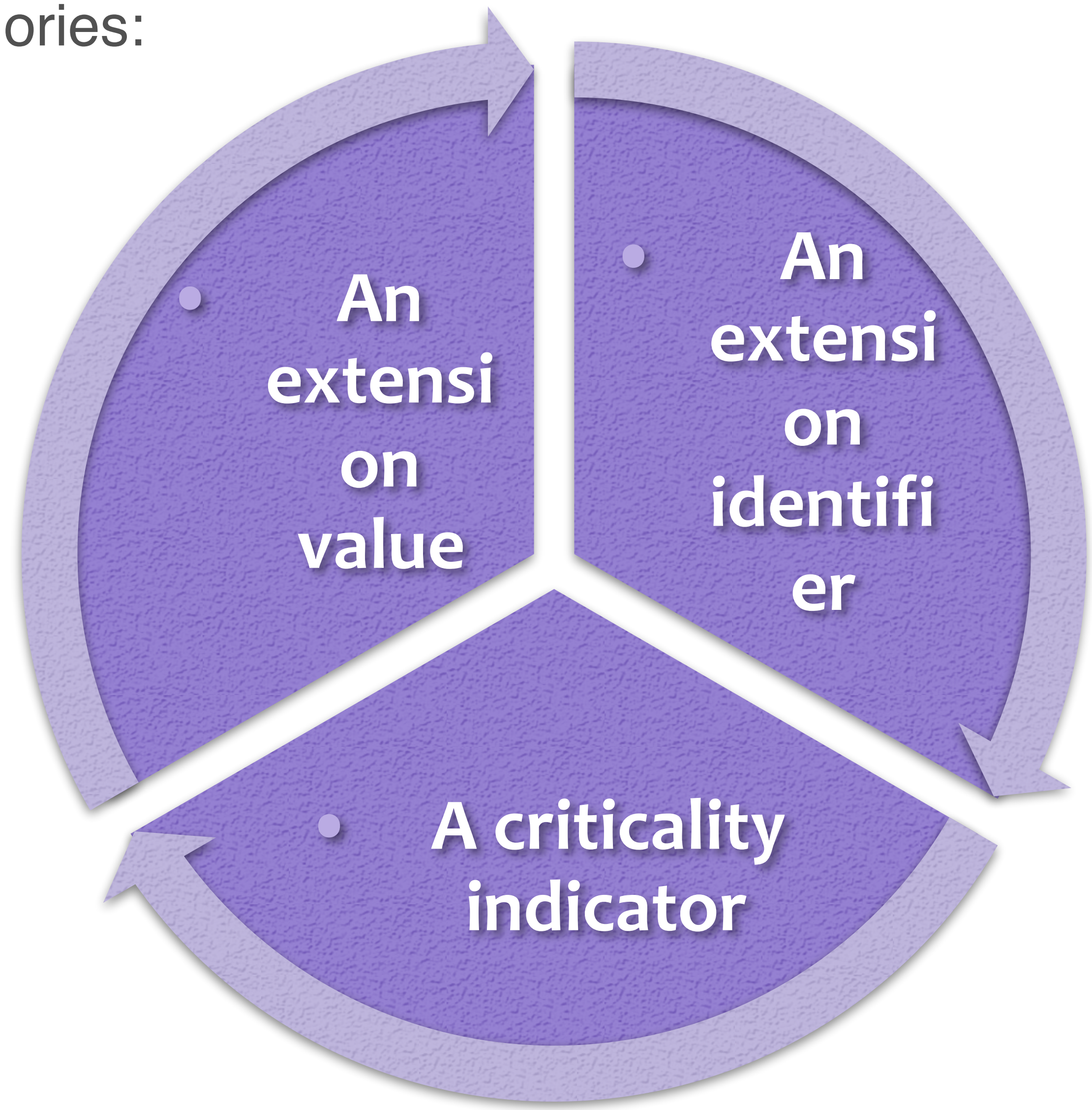
Certificate Revocation

- Each certificate includes a period of validity
  - Typically a new certificate is issued just before the expiration of the old one
- It may be desirable on occasion to revoke a certificate before it expires, for one of the following reasons:
  - The user's private key is assumed to be compromised
  - The user is no longer certified by this CA
  - The CA's certificate is assumed to be compromised
- Each CA must maintain a list consisting of all revoked but not expired certificates issued by that CA
  - These lists should be posted on the directory

X.509 Version 3

- Version 2 format does not convey all of the information that recent design and implementation experience has shown to be needed
- Rather than continue to add fields to a fixed format, standards developers felt that a more flexible approach was needed
  - Version 3 includes a number of optional extensions
- The certificate extensions fall into three main categories:
  - Key and policy information
  - Subject and issuer attributes
  - Certification path constraints

Each extension consists of:

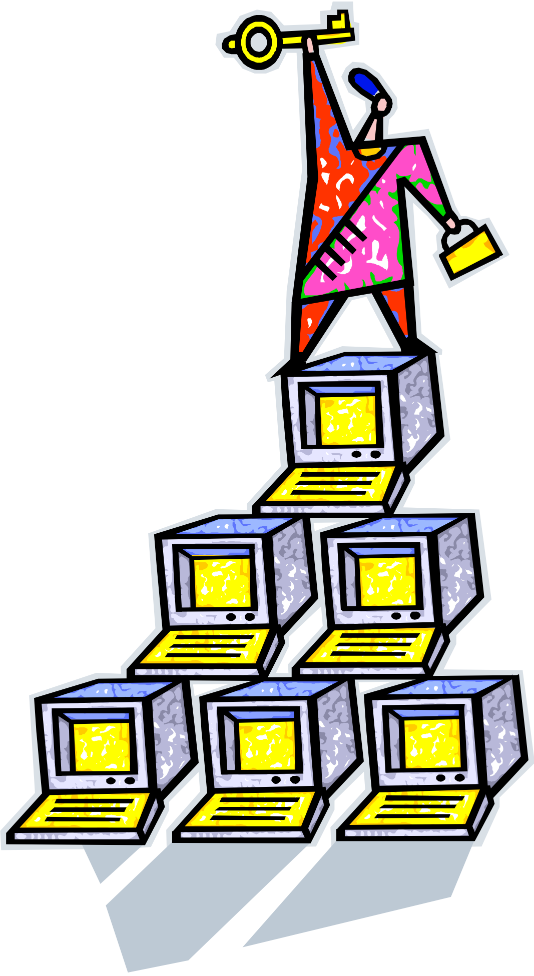


Key and Policy Information

- These extensions convey additional information about the subject and issuer keys plus indicators of certificate policy
- A certificate policy is a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements

Included are:

- Authority key identifier
- Subject key identifier
- Key usage
- Private-key usage period
- Certificate policies
- Policy mappings



Certificate Subject and Issuer Attributes

- These extensions support alternative names, in alternative formats, for a certificate subject or certificate issuer
- Can convey additional information about the certificate subject to increase a certificate user's confidence that the certificate subject is a particular person or entity
- The extension fields in this area include:
  - Subject alternative name
  - Issuer alternative name
  - Subject directory attributes

Certification Path Constraints

- These extensions allow constraint specifications to be included in certificates issued for CAs by other CAs
- The constraints may restrict the types of certificates that can be issued by the subject CA or that may occur subsequently in a certification chain
- The extension fields in this area include:
  - Basic constraints
  - Name constraints
  - Policy constraints