S7 B-TECH COMPUTER SCIENCE AND ENGINEERING
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

PREPARED BY
DR SUREKHA MARIAM VARGHESE

MODULE 5

- (Key Distribution and System Security)
- Key management – Distribution of secret keys using symmetric and asymmetric encryption,
- Distribution of public keys. System security – Intruders, Intrusion detection techniques,
- Password management. Malicious software – Viruses, Related threats, Countermeasures.
- Distributed Denial of Service (DDoS) attacks – Types, Countermeasures.

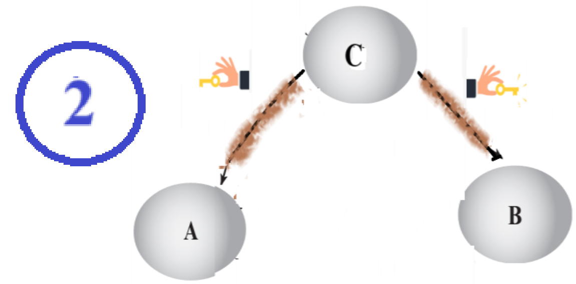
KEY DISTRIBUTION

- Strength of Cryptographic Systems relies heavily on the robustness of the key distribution technique
- Essential for securely delivering keys to parties without exposure
- Symmetric Encryption requires both parties to share and protect the same key
- Frequent Key Changes to Limit data compromise if a key is exposed
- Types of Encryption*:
  - End-to-End Encryption: Keeps data secure throughout transmission accessible only by sender and recipient
  - Point-to-Point Encryption: Encrypts data at the sender's device, decrypts it at intermediary point, and then re-encrypts it for the fina recipient
  - Link Encryption: Secures data at each step; exposes it at each node

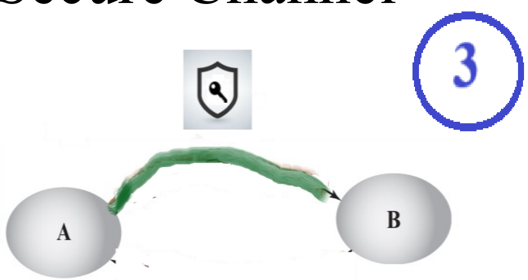
KEY DISTRIBUTION -A AND B

- Physical Transfer

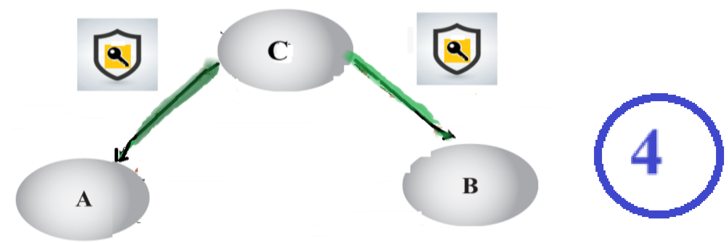
Trusted Party



Secure Channel



Trusted Party



tion 3 is applicable to link encryption, .EE, and P2PE, ensuring that keys are protected during transmission.

Option 4 involves a Key Distribution Center (KDC) and is widely used for E2EE and P2PE, where each user shares a unique key with the KDC for secure key distribution

KDC- DISTRIBUTING KEYS

- In this scheme, a key distribution center is responsible for distributing keys to pairs of users (hosts, processes, applications) as needed.
- Each user must share a unique key with the key distribution center for purposes of key distributon

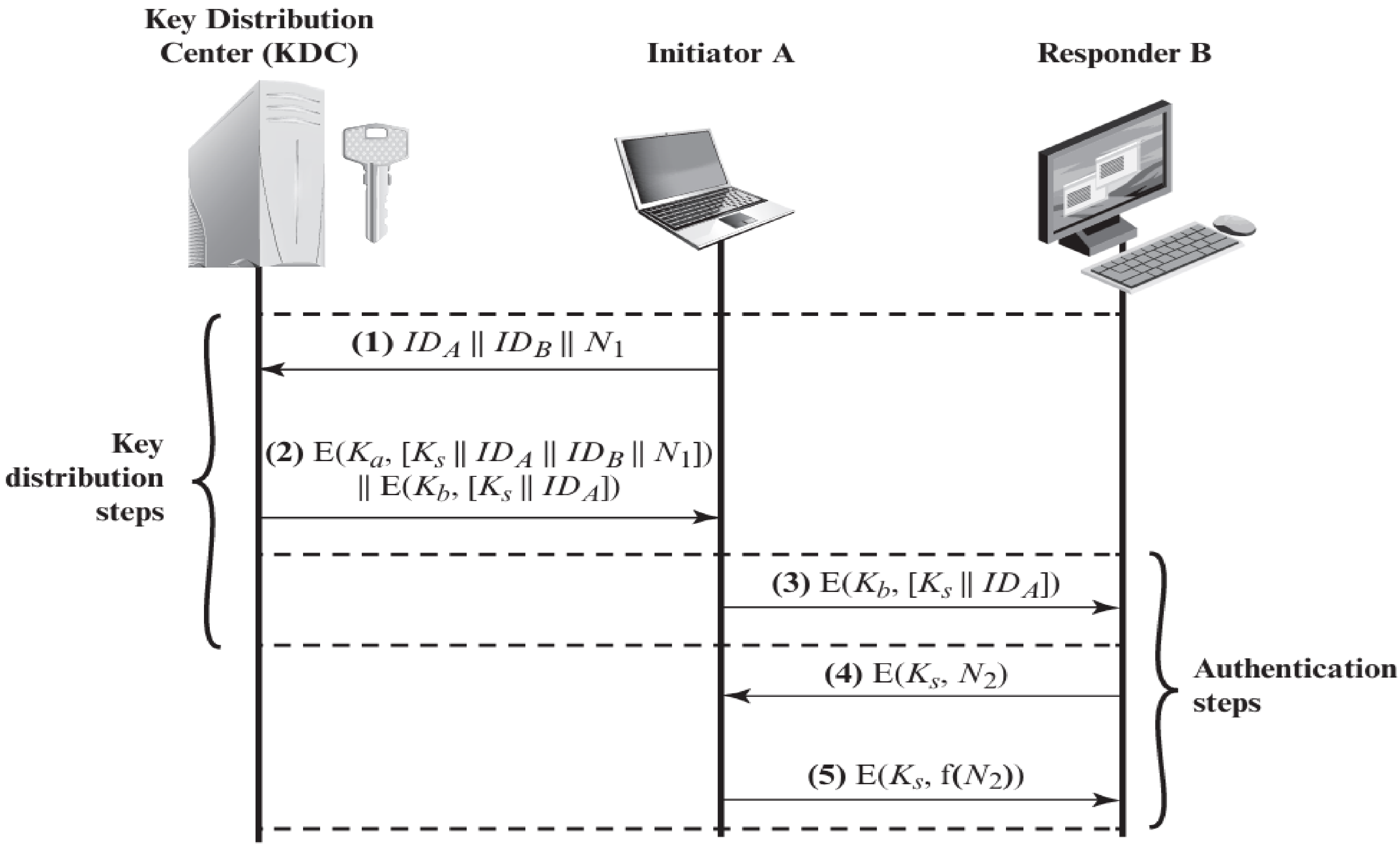
Session Key and Master Key

- Key distribution center based on use of a hierarchy of keys
- At a minimum, two levels of keys are used
- Session Key
  - Communication between end systems encrypted using temp key
  - Used for the duration of a logical connection, and then discarded
  - Each session key obtained from KDC over the same network
  - Session keys are transmitted in encrypted form, using a master key
- Master Key - shared by KDC and an end system or user
  - Each end user shares unique master key with KDC
  - Only N masterkeys instead of N(N-1)/2 session keys

USE OF NONCE

- A timestamp, a counter, or a random number; differs with eac request. to prevent masquerade, difficult for opponent to gues
- Preventing Replay Attacks
  - A nonce ensures that old communications cannot be reused in repla attacks. If an attacker tries to resend a previous message, recipient can detect reuse and reject the message
- Ensuring Freshness
  - By including a unique nonce in each communication, both parties c be confident that message is recent and not a replay of old mess
- Synchronization
  - Used to synchronize the state between parties during key exchange

A Key Distribution Scenario



STEPS

- A requests a session key from the KDC for a connection to B. The message includes A's and B's identities and a nonce, N1.
- KDC responds with a message encrypted using Ka- master key of A. The message includes:
  - Session key, Ks
  - Original request with N1
  - Ks and A's identifier (IDA) encrypted with Kb
- A stores Ks and forwards E(Kb, [Ks, IDA]) to B.
- B sends a nonce, N2, to A using Ks.
- A responds with f(N2) using Ks

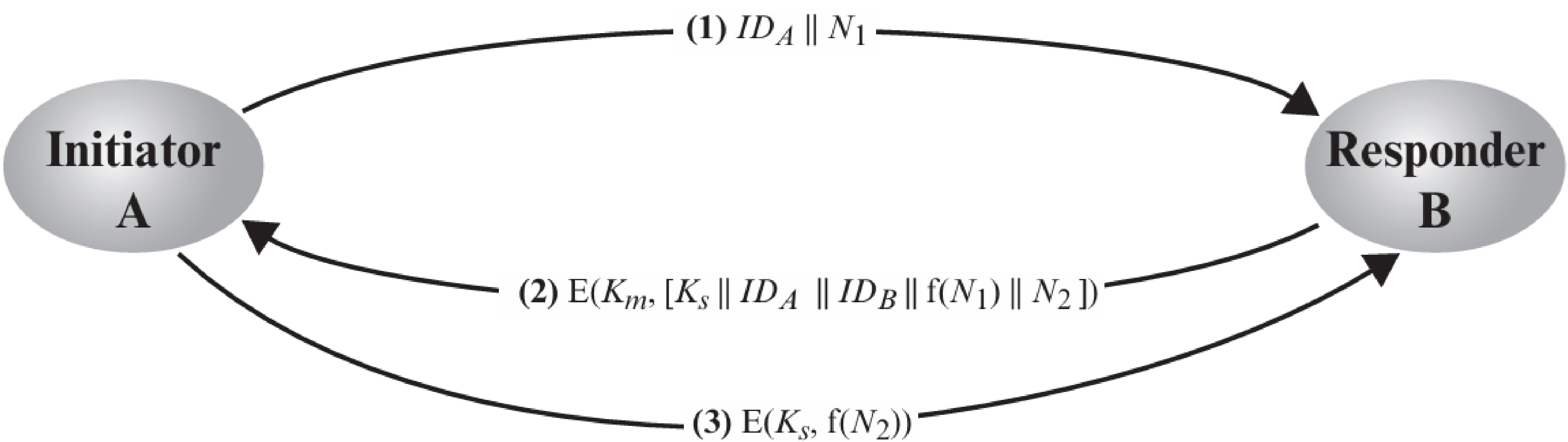
HIERARCHICAL KEY CONTROL

- Key distribution function entrusted to a hierarchy of KDCs.
- Local KDCs: Each local KDC handles key distribution within its domain (e.g., a LAN or building).
- Inter-domain Communication: For entities in different domains, local KDCs communicate through a global KDC.
  - Key Selection: Any of the involved KDCs (local or global) can select the key
- The hierarchical concept can be extended to three or even more layers,

DECENTRALIZED KEY DISTRIBUTION

A session key may be established with the following sequence of steps

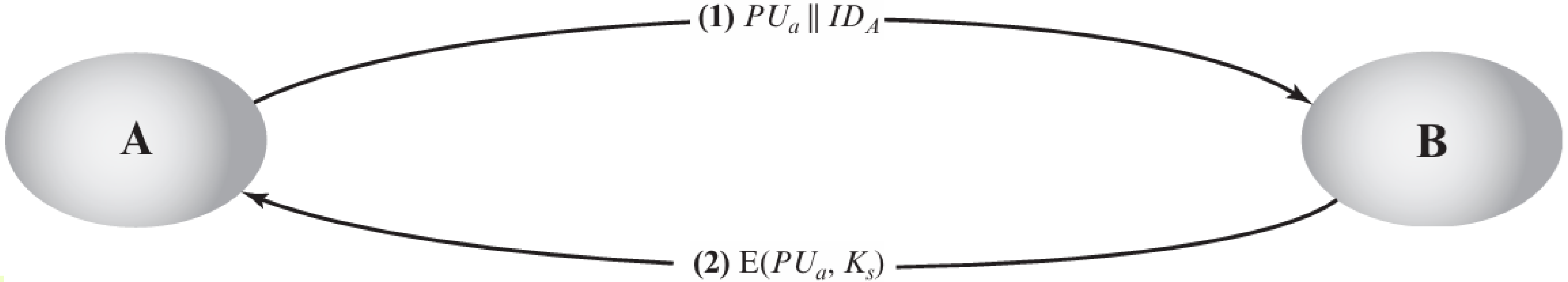
1. A issues a request to B for a session key and includes a nonce, N_1.
2. B responds with a message that is encrypted using the shared master key. The response includes the session key selected by B, an identifier of B, the value $f(N_1)$, and another nonce, N_2.
3. Using the new session key, A returns $f(N_2)$ to B.



SYMMETRIC KEY DISTRIBUTION USING ASYMMETRIC ENCRYPTION

- Simple Secret Key Distribution

1. A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a message to B consisting of PU_a and an identifier of A, ID_A.
2. B generates a secret key, K_s, and transmits it to A, which is encrypted with A's public key.
3. A computes $D(PR_a, E(PU_a, K_s))$ to recover the secret key. Because only A can decrypt the message, only A and B will know the identity of K_s.
4. A discards PU_a and PR_a and B discards PU_a.



MAN IN THE MIDDLE ATTACK

1. A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a message intended for B consisting of PU_a and an identifier of A, ID_A.
2. D intercepts the message, creates its own public/private key pair $\{PU_d, PR_d\}$ and transmits $PU_d \| ID_A$ to B.
3. B generates a secret key, K_s, and transmits $E(PU_d, K_s)$.
4. D intercepts the message and learns K_s by computing $D(PR_d, E(PU_d, K_s))$.
5. D transmits $E(PU_a, K_s)$ to A.

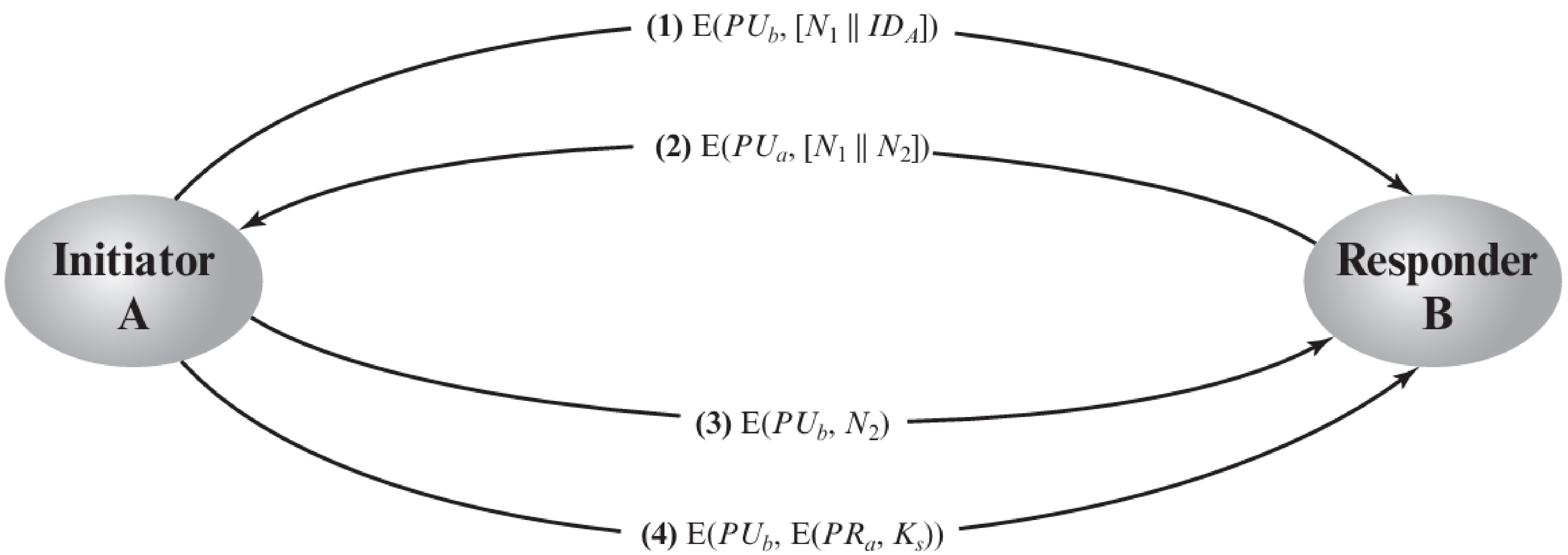
SECRET KEY DISTRIBUTION WITH CONFIDENTIALITY & AUTHENTICATION

- A and B have exchanged public keys

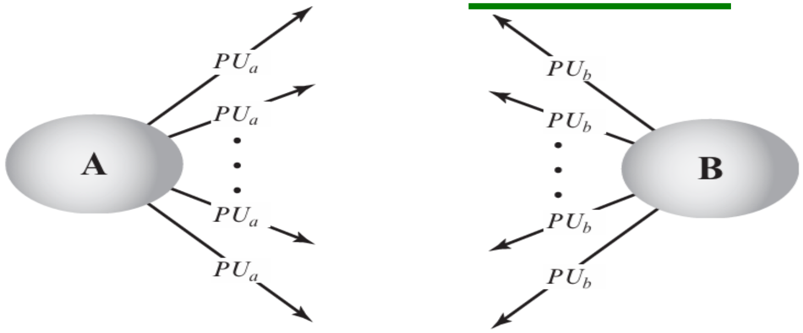
1. A uses B's public key to encrypt a message to B containing an identifier of A(ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.
2. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B (N_2). Because only B could have decrypted message (1), the presence of N_1 in message (2) assures A that the correspondent is B.
3. A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.
4. A selects a secret key K_s and sends $M = \text{E}(PU_b, \text{E}(PR_a, K_s))$ to B. Encryption of this message with B's public key ensures that only B can read it; encryption with A's private key ensures that only A could have sent it.
5. B computes $\text{D}(PU_a, \text{D}(PR_b, M))$ to recover the secret key.

SECRET KEY DISTRIBUTION WITH CONFIDENTIALITY & AUTHENTICATION CONTD..

1. A uses B's public key to encrypt a message $A(ID_A)$ and a nonce (N_1) to B
2. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) and a nonce by B (N_2)
3. A returns N_2, encrypted using B's public key, to assure B that its A.
4. A selects a secret key K_s and sends $M = E(PU_b, E(PR_a, K_s))$ to B.
5. B computes $D(PU_a, D(PR_b, M))$ to recover the secret key.



DISTRIBUTION OF PUBLIC KEYS



- Public announcement
  - Forgery possible
- Publicly available directory
  - dynamic directory of public keys
  - Maintenance and distribution by some trusted entity
- Public-key authority
- Public-key certificates

PUBLIC KEY AUTHORITY

1. A sends a timestamped message to the public-key authority containing a request for the current public key of B.
2. The authority responds with a message that is encrypted using the authority's private key, PR_{auth}. Thus, A is able to decrypt the message using the authority's public key. Therefore, A is assured that the message originated with the authority. The message includes the following:
   - B's public key, PU_b, which A can use to encrypt messages destined for B
   - The original request used to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority
   - The original timestamp given so A can determine that this is not an old message from the authority containing a key other than B's current public key
3. A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.

Public Key Authority Contd..

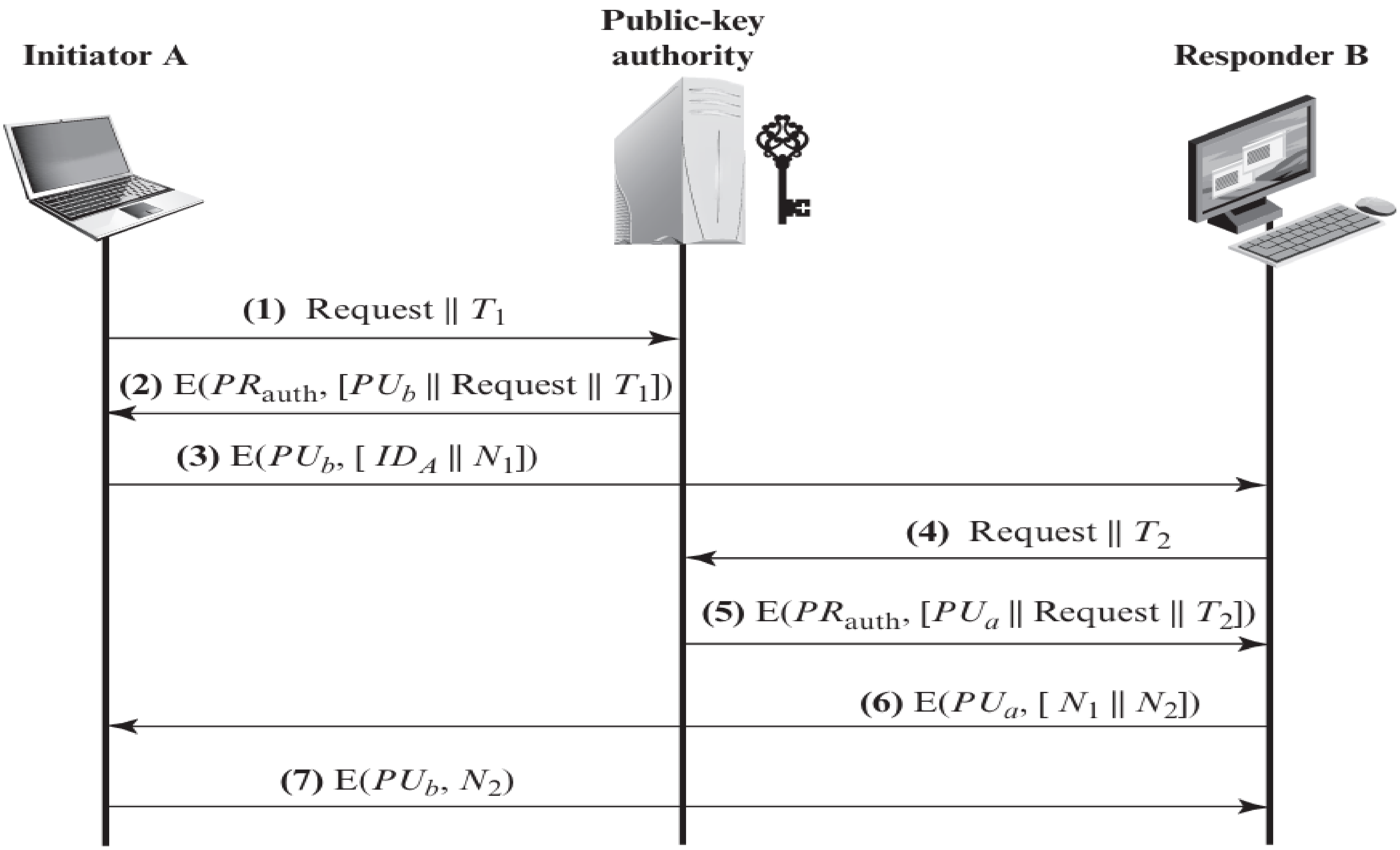
4, 5. B retrieves A's public key from the authority in the same manner as A retrieved B's public key.

At this point, public keys have been securely delivered to A and B, and they may begin their protected exchange. However, two additional steps are desirable:

6. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B (N_2). Because only B could have decrypted message (3), the presence of N_1 in message (6) assures A that the correspondent is B.

7. A returns N_2, which is encrypted using B's public key, to assure B that its correspondent is A.

PUBLIC KEY AUTHORITY CONTD..



Public key Certificates Based Key Exchange

- Allow key exchange without contacting a public-key authority, ensuring reliability
- Certificate Components: Public key, key owner's identifier, signed by a trusted third party (certificate authority)
- Participants apply to the certificate authority with a public key, requesting a certificate through secure communication
- Process:
  - User securely presents public key to the authority and obtains a certificate
  - User publishes the certificate
  - Others can verify the certificate's validity via the trusted signature

PUBLIC KEY CERTIFICATES

- Requirements:
- Any participant can read the certificate to identify the owner and public key
- Any participant can verify the certificate's authenticity/ genuineness by checking the trusted signature
- Any participant can verify the certificate's time validity
- Only the certificate authority can create/update certificates
- X.509 standard - universally accepted scheme for formatting public-key certificates