*

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

19ITB302-Cryptography and Network Security

UNIT-4 INTERACTIVE TECHNIQUES AND TOOLS

Key Management and Distribution

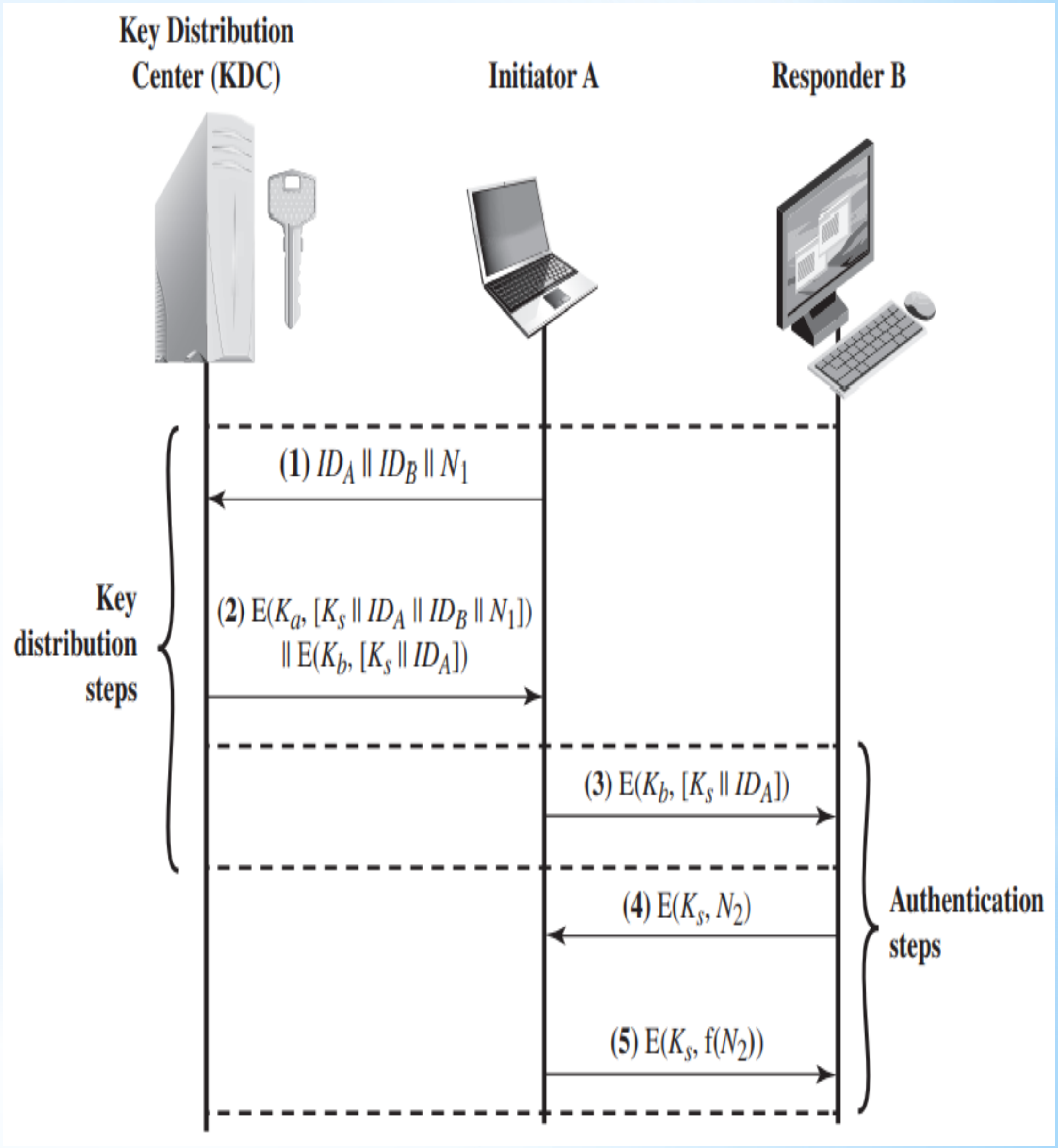
- Symmetric Key Distribution Using Symmetric Encryption
- Symmetric Key Distribution Using Asymmetric Encryption
- Distribution of Public Keys

Symmetric Key Distribution Using Symmetric Encryption

- For symmetric encryption to work, the two parties to an exchange must share the same key.
- That key must be protected from access by others.
- Furthermore, frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key.
- Therefore, the strength of any cryptographic system rests with the **key distribution technique**,
- A 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.

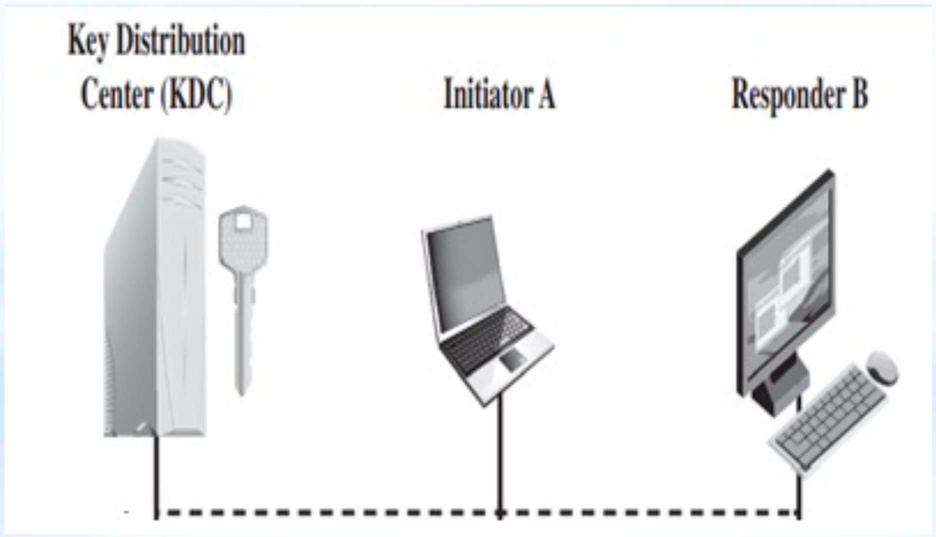
❑ Key Distribution Scenario

- ➢ The key distribution concept can be deployed in a number of ways. A typical scenario is illustrated in Figure.
- ➢ The scenario assumes that each user shares a unique master key with the key distribution center (KDC).



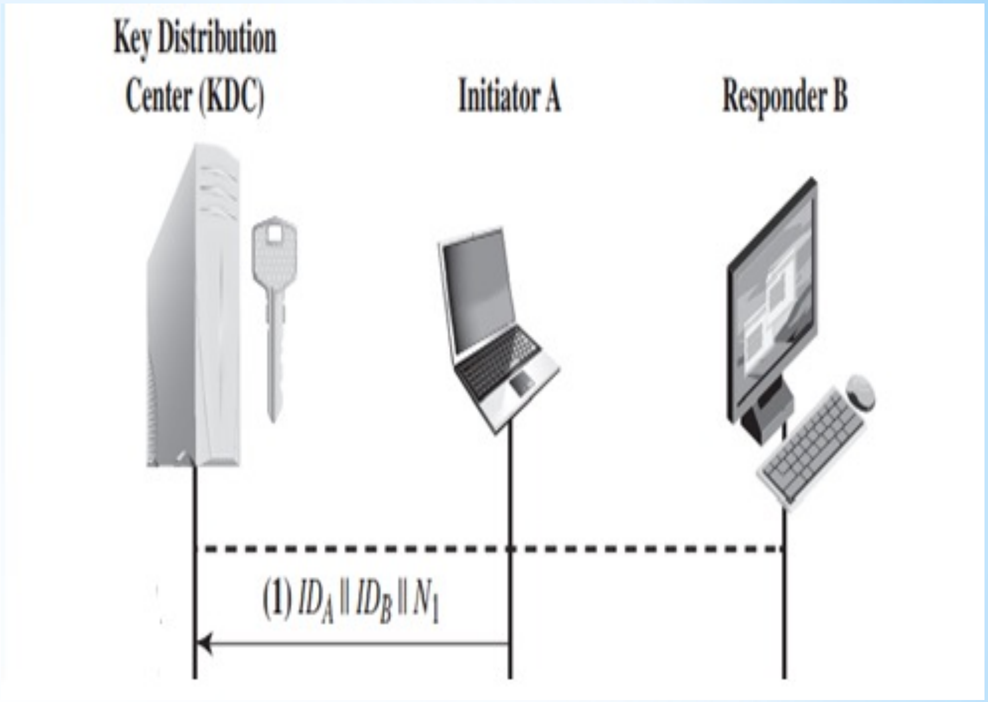
Key Distribution Center (KDC)
Initiator A
Responder B
(1) $ID_A \| ID_B \| N_1$
Key distribution steps
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$
(3) $E(K_b, [K_s \| ID_A])$
(4) $E(K_s, N_2)$
(5) $E(K_s, f(N_2))$
Authentication steps

- Let us assume that user A wishes to establish a logical connection with B and requires a one-time session key to protect the data transmitted over the connection.
- A has a master key, $\mathbf{K_a}$, known only to itself and the KDC.
- Similarly, B shares the master key K_b with the KDC.



- The following steps occur.

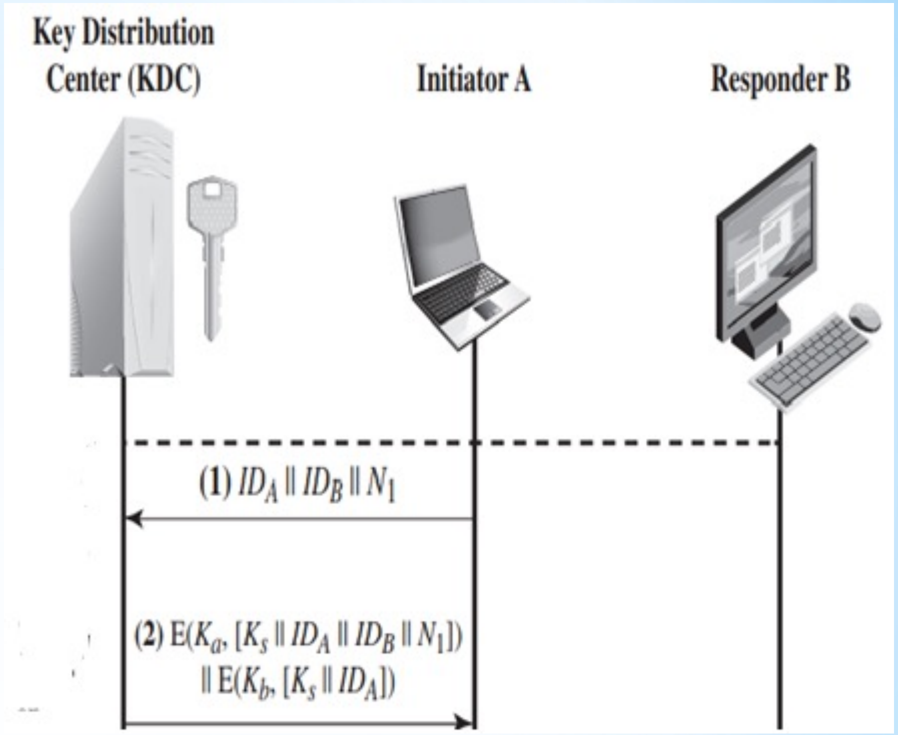
1. A issues a request to the KDC for a session key to protect a logical connection to B. The message includes the identity of A and B and a unique identifier, N_1, for this transaction, which we refer to as a ***nonce***.



- The ***nonce*** may be a timestamp, a counter, or a random number.
- The minimum requirement is that it differs with each request.
- It should be difficult for an opponent to guess the nonce. Thus, a random number is a good choice for a nonce.

2. The KDC responds with a message encrypted using K_a. Thus, A is the only one who can successfully read the message, and A knows that it originated at the KDC.

   The message includes two items intended for A:


Key Distribution Center (KDC)
Initiator A
Responder B
(1) $ID_A \| ID_B \| N_1$
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$

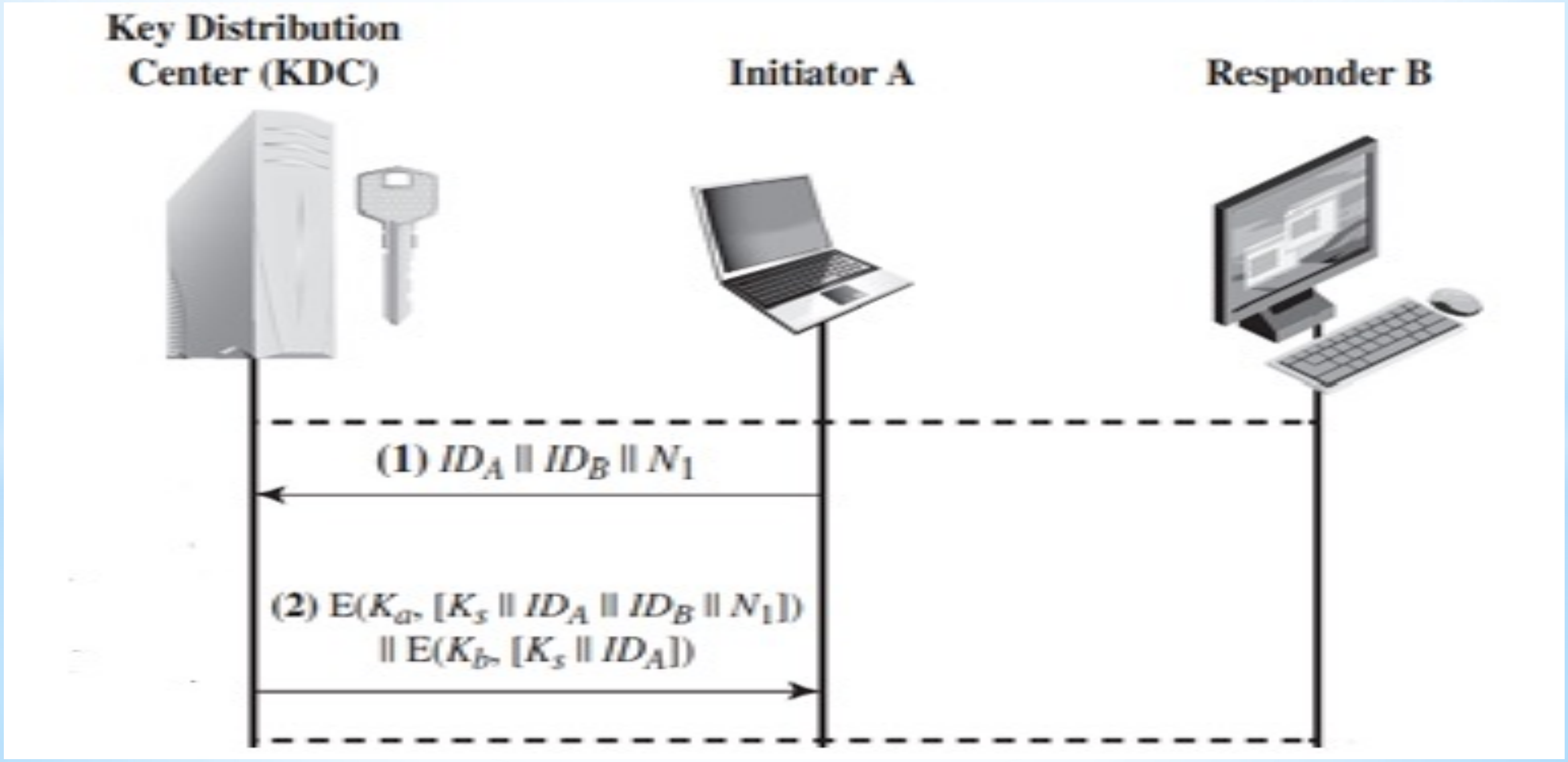
- The one-time session key, K_s, to be used for the session
- The original request message, including the nonce, to enable A to match this response with the appropriate request.

Thus, A can verify that its original request was not altered before reception by the KDC and, because of the nonce, that this is not a replay of some previous request.

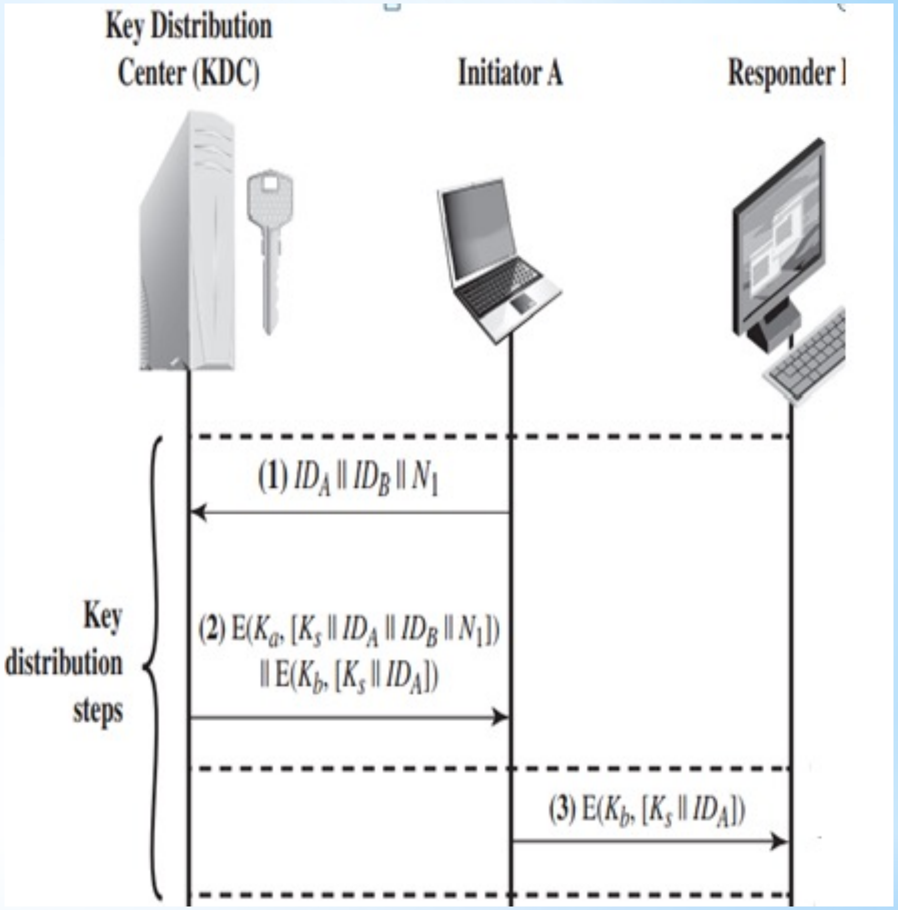
In addition, the message includes two items intended for B:

- The one-time session key, K_s, to be used for the session.
- An identifier of A (e.g., its network address), ID_A

These last two items are encrypted with K_b (the master key that the KDC shares with B). They are to be sent to B to establish the connection and prove A's identity.


Key Distribution Center (KDC)
Initiator A
Responder B
(1) $ID_A \| ID_B \| N_1$
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$

3. A stores the session key for use in the upcoming session and forwards to B the information that originated at the KDC for B, namely, $E(K_b,[K_s||ID_A])$. Because this information is encrypted with K_b, it is protected from eavesdropping. B now **knows** the session key (K_s), **knows** that the other party is A (from ID_A), and **knows** that the information originated at the KDC (because it is encrypted using K_b).

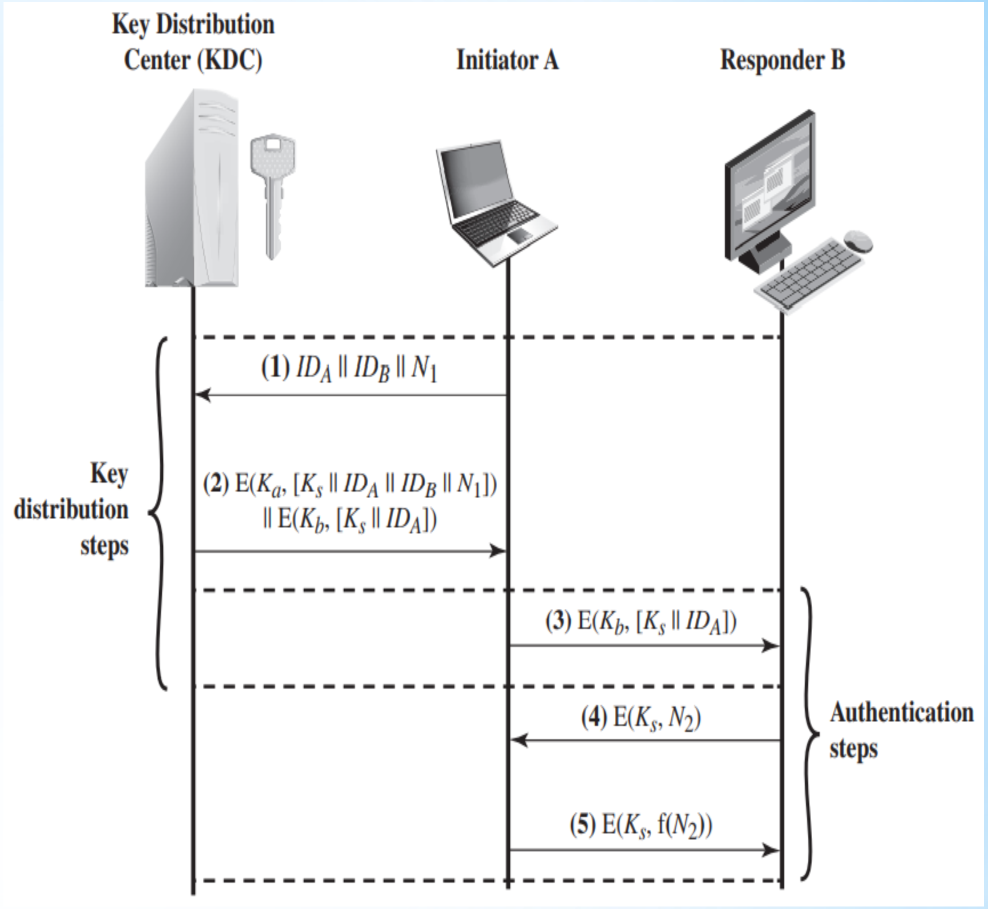

Key Distribution Center (KDC)
Initiator A
Responder
Key distribution steps
(1) $ID_A \| ID_B \| N_1$
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$
(3) $E(K_b, [K_s \| ID_A])$

At this point, a session key has been securely delivered to A and B, and they may begin their protected exchange.

However, two additional steps are desirable:

4. Using the newly minted session key for encryption, B sends a nonce, N_2, to A.

5. Also, using K_s, A responds with $f(N_2)$, where f is a function that performs some transformation on N_2 (e.g., adding one).


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

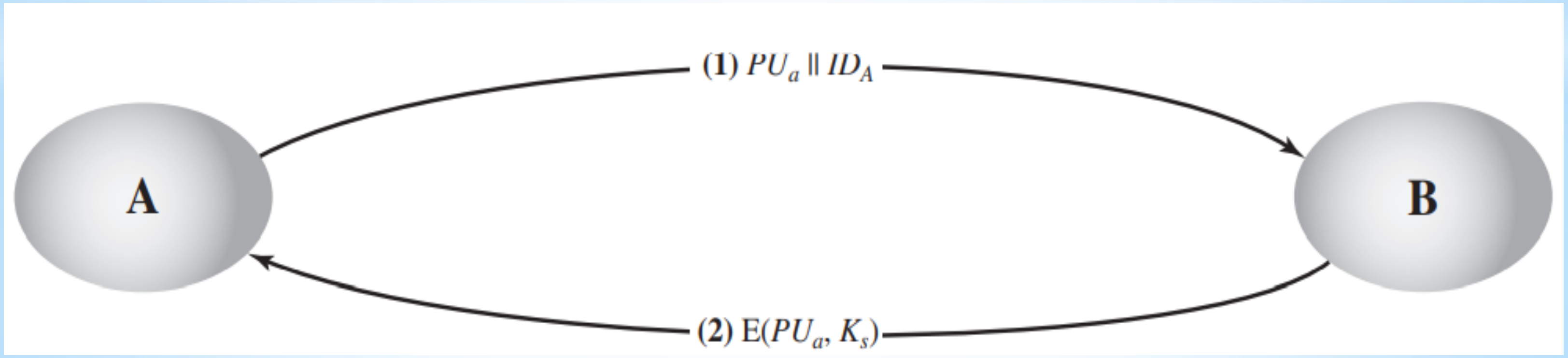
- ➢ These steps assure B that the original message it received (step 3) was not a replay.
- ➢ Note that the actual key distribution involves only steps 1 - 3, but that steps 4 and 5, as well as step 3, perform an authentication function.

Symmetric Key Distribution Using Asymmetric Encryption

❑ Simple Secret Key Distribution

➢ An extremely simple scheme was put forward by Merkle and as illustrated in Figure. If A wishes to communicate with B, the following procedure is employed:

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.



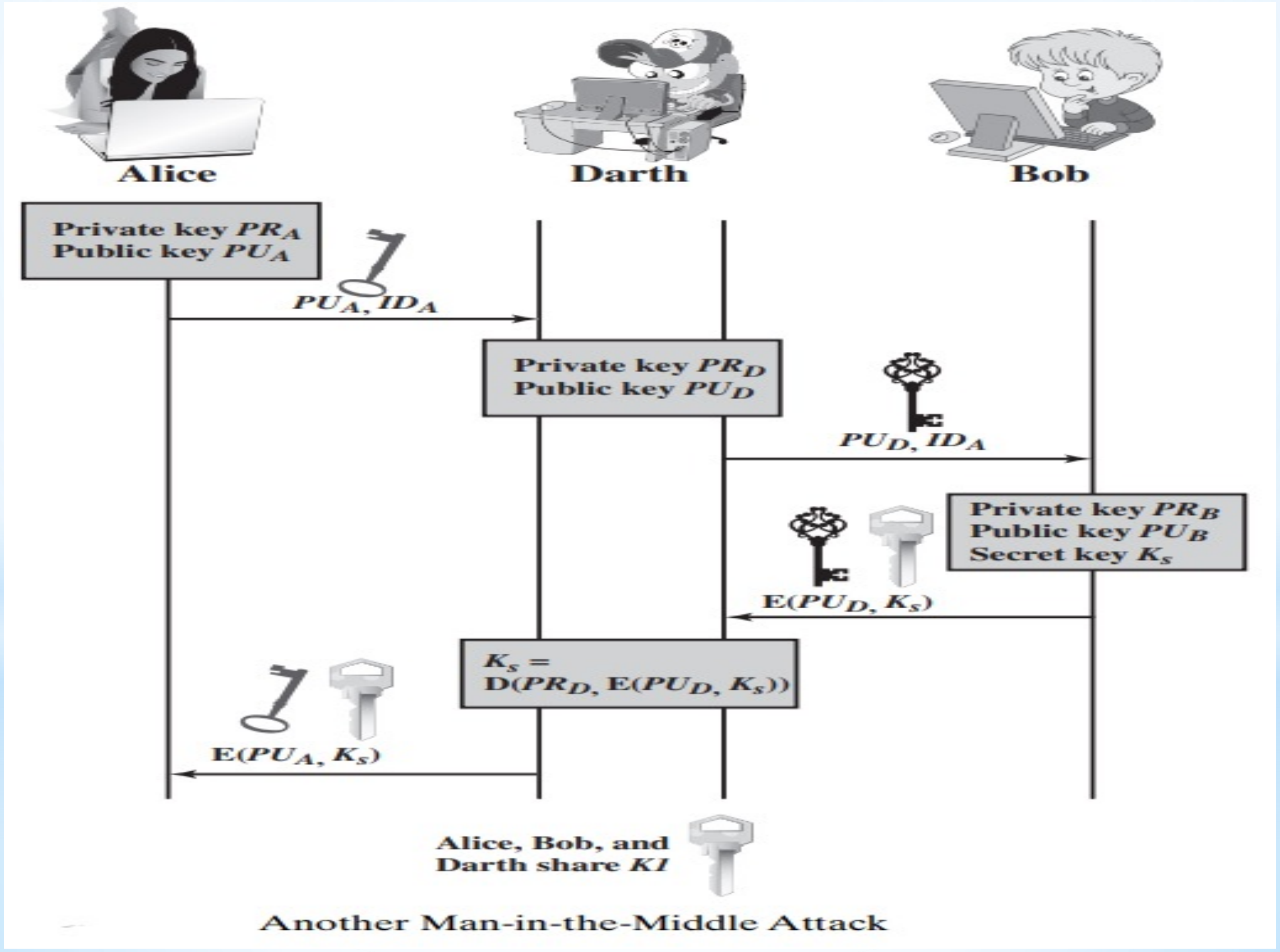
3. A computes $D(PR_a, E(PU_a, K_s))$ to recover the secret key. Because only A can decrypt the message, only A and B will know the identity of K_s.
4. A discards PU_a and PR_a and B discards PU_a.

- ➢ A and B can now securely communicate using conventional encryption and the session key K_s.
- ➢ At the completion of the exchange, both A and B discard K_s.
- ➢ Despite its simplicity, this is an attractive protocol.
  - No keys exist before the start of the communication and none exist after the completion of communication.
  - Thus, the risk of compromise of the keys is minimal.
  - At the same time, the communication is secure from eavesdropping.

- This protocol is insecure against an adversary who can intercept messages and then either relay the intercepted message or substitute another message.
- Such an attack is known as a **man-in-the-middle** attack.
- If an adversary, D, has control of the intervening communication channel, then D can compromise the communication in the following fashion without being detected:

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 $PU_s \| ID_A$ to B.

3. B generates a secret key, K_s, and transmits $E(PU_s, K_s)$.
4. D intercepts the message and learns K_s by: $D(PR_d, E(PU_d, K_s))$.
5. D transmits $E(PU_a, K_s)$ to A.

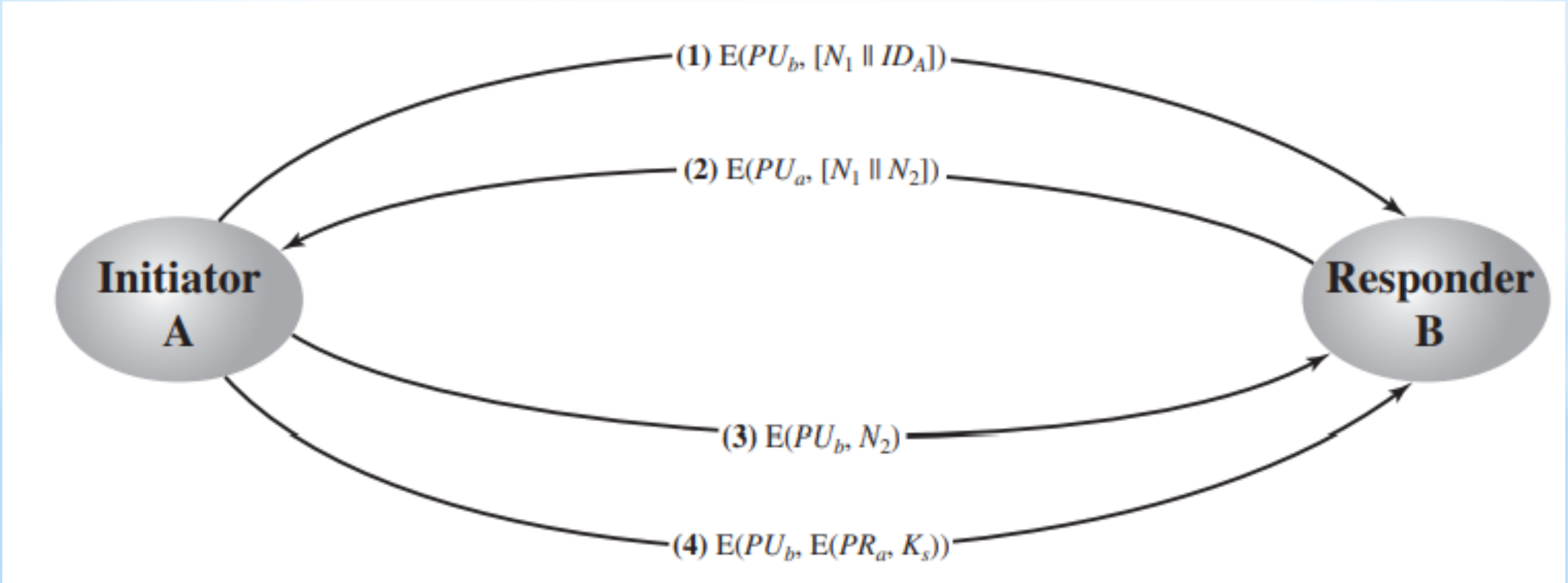


Another Man-in-the-Middle Attack

- The result is that both A and B know K_s and are unaware that K_s has also been revealed to D.
- A and B can now exchange messages using K_s.
- D no longer actively interferes with the communications channel but simply eavesdrops.
- Knowing K_s, D can decrypt all messages, and both A and B are unaware of the problem.
- Thus, this simple protocol is only useful in an environment where the only threat is eavesdropping

❑ Secret Key Distribution with Confidentiality and Authentication

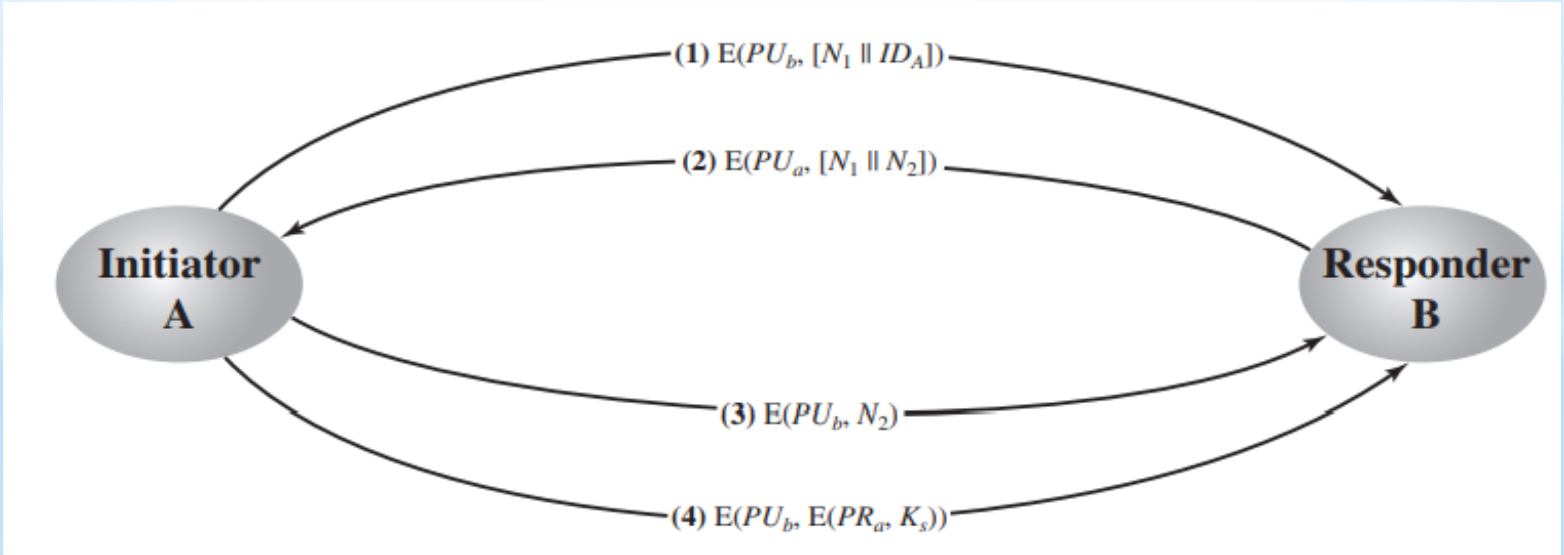
- ➢ The approach shown in the Figure provides protection against both active and passive attacks.
- ➢ We begin at a point when it is assumed that A and B have exchanged public keys by one of the described schemes.


(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

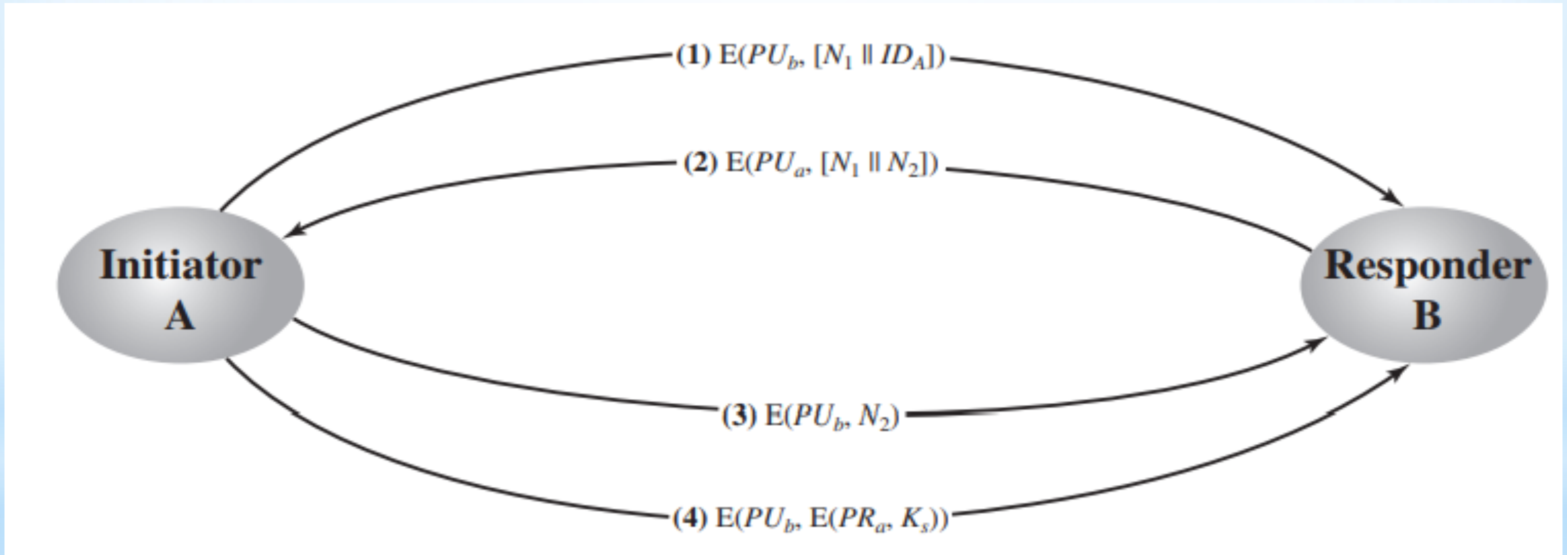
Then the following steps occur.

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.


(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

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 $M = E(PU_b, E(PR_a, K_s))$ 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 $D(PU_a, D(PR_b, M))$ to recover the secret key.


(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

The result is that this scheme ensures both confidentiality and authentication in the exchange of a secret key

Distribution Of Public Keys

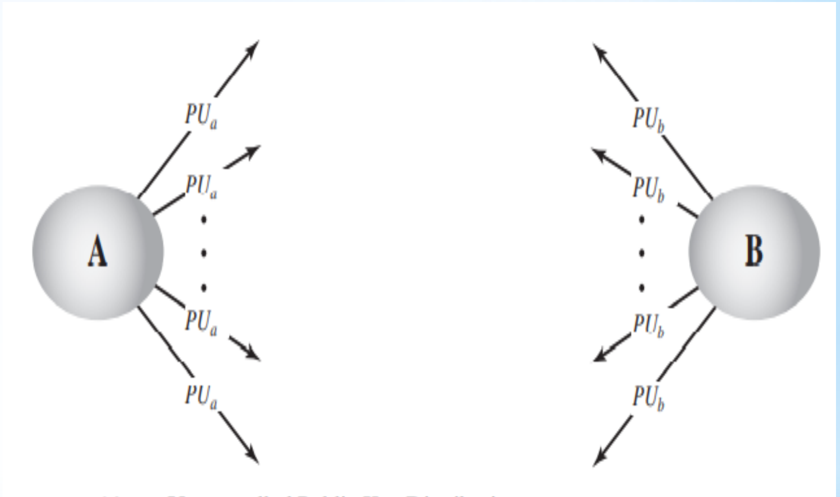
- Several techniques have been proposed for the distribution of public keys. Virtually all these proposals can be grouped into the following general schemes:

1. Public announcement
2. Publicly available directory
3. Public-key authority
4. Public-key certificates

Public Announcement of Public Keys

In a public-key encryption, any participant can send his or her public key to any other participant or broadcast the key to the community at large as shown in the figure. This approach has a major weakness:

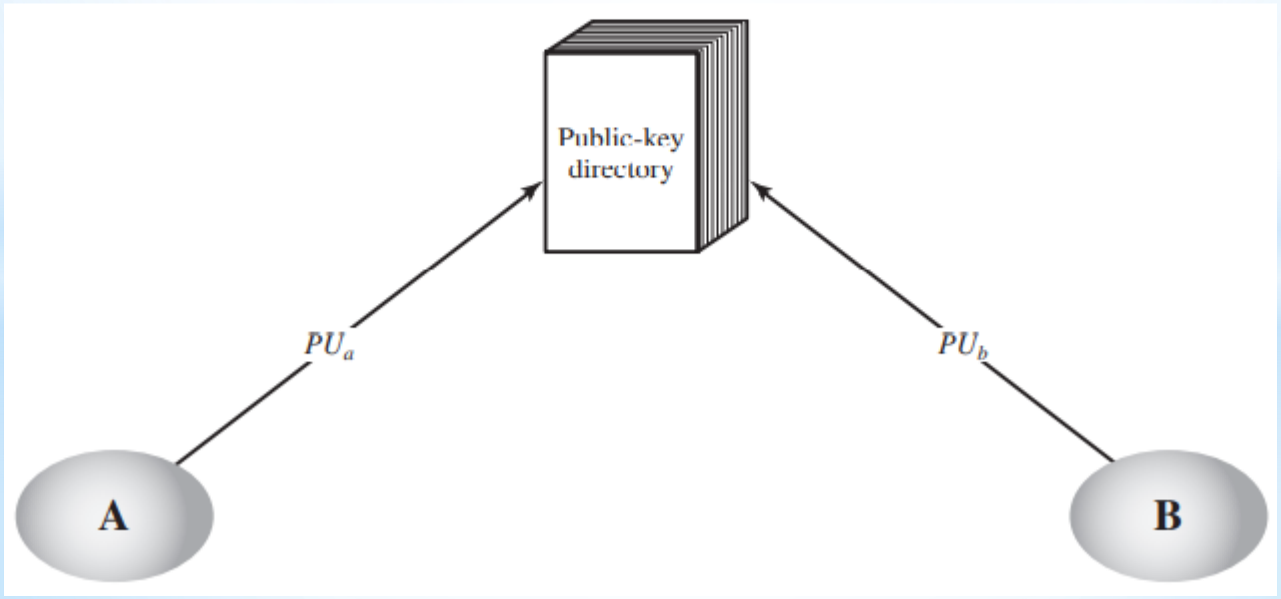
- Anyone can forge such a public announcement.
- That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key.
- Until such time as user A discovers the forgery and alerts other participants, the forger is able to read all encrypted messages intended for A and can use the forged keys for authentication



Uncontrolled Public-Key Distribution

Publicly Available Directory

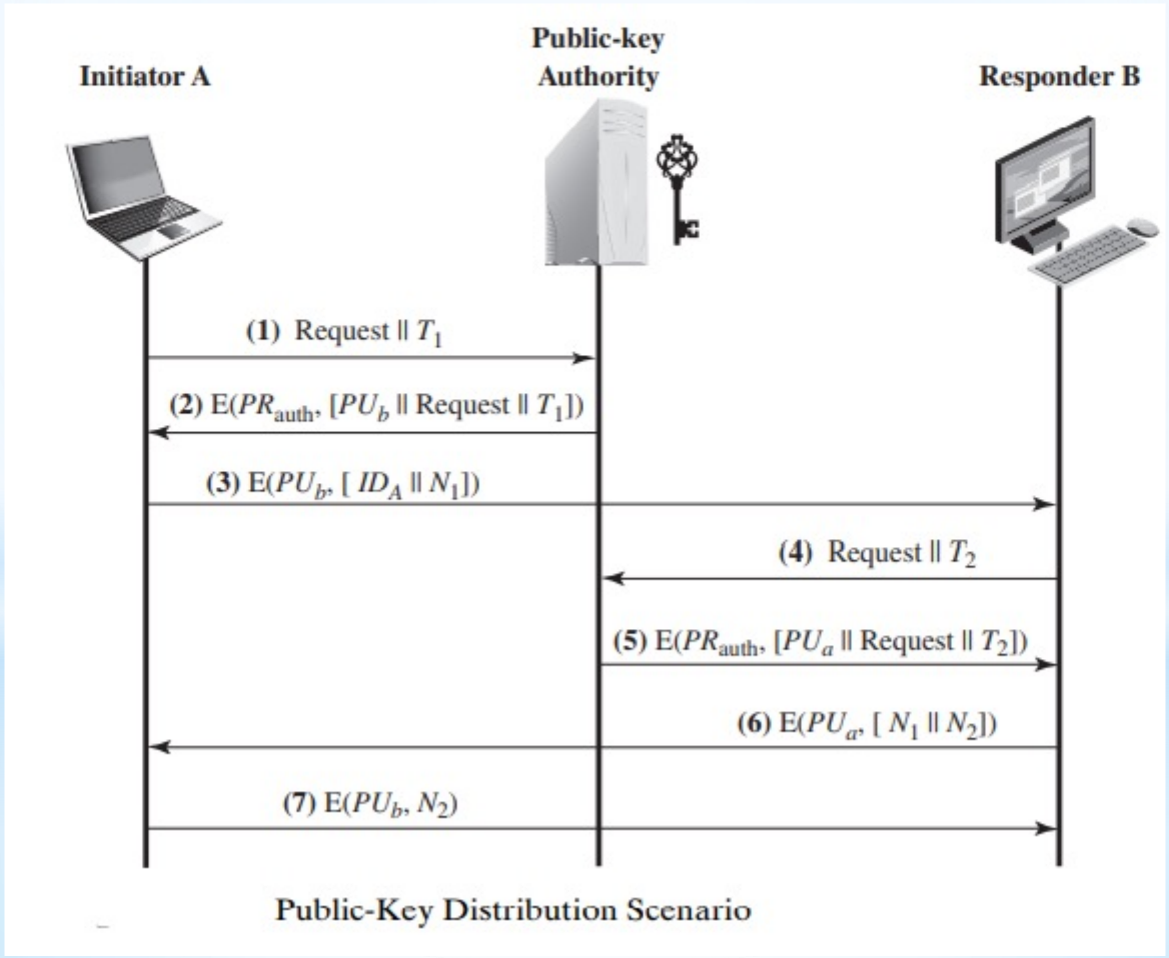
- A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys.
- Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization.
- This scheme is clearly more secure than individual public announcements but still has vulnerabilities.



Public-Key Authority

- Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory.
- A typical scenario is illustrated in the figure.
- As before, the scenario assumes that a central authority maintains a dynamic directory of public keys of all participants.
- In addition, each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.
- The following steps occur.

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.

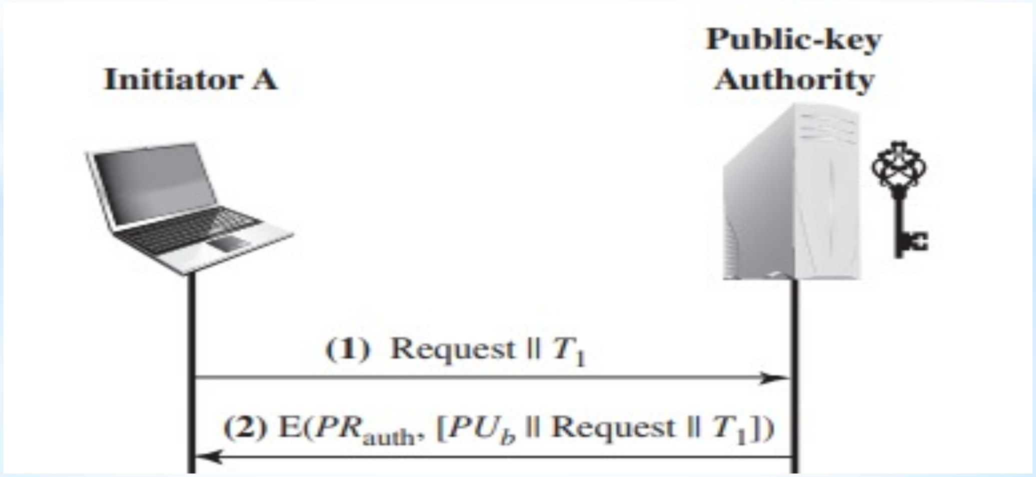

Initiator A
Public-key Authority
Responder B
(1) Request || T_1
(2) E(PR_{auth}, [PU_b || Request || T_1])
(3) E(PU_b, [ID_A || N_1])
(4) Request || T_2
(5) E(PR_{auth}, [PU_a || Request || T_2])
(6) E(PU_a, [N_1 || N_2])
(7) E(PU_b, N_2)

Public-Key Distribution Scenario

Therefore, A is assured that the message originated with the authority.

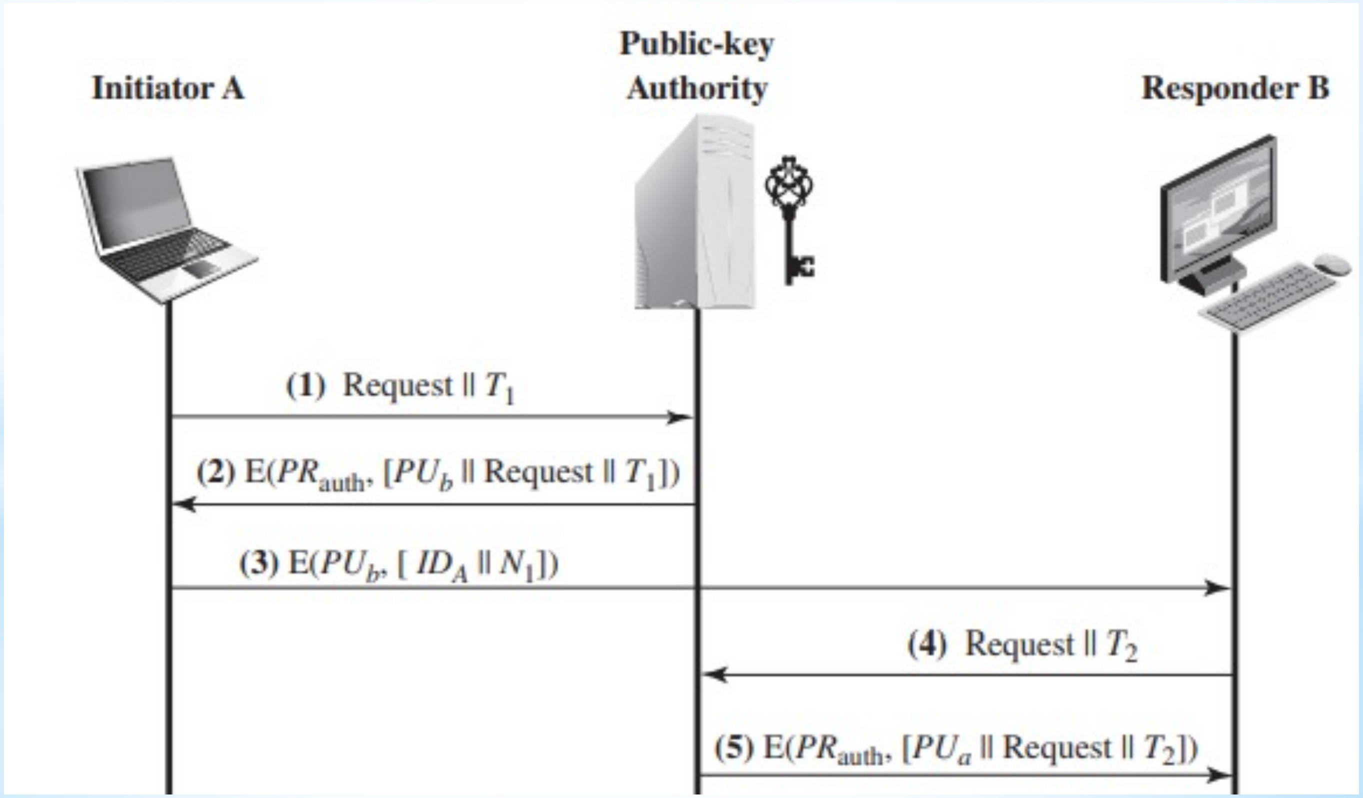
The message includes the following:

- 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



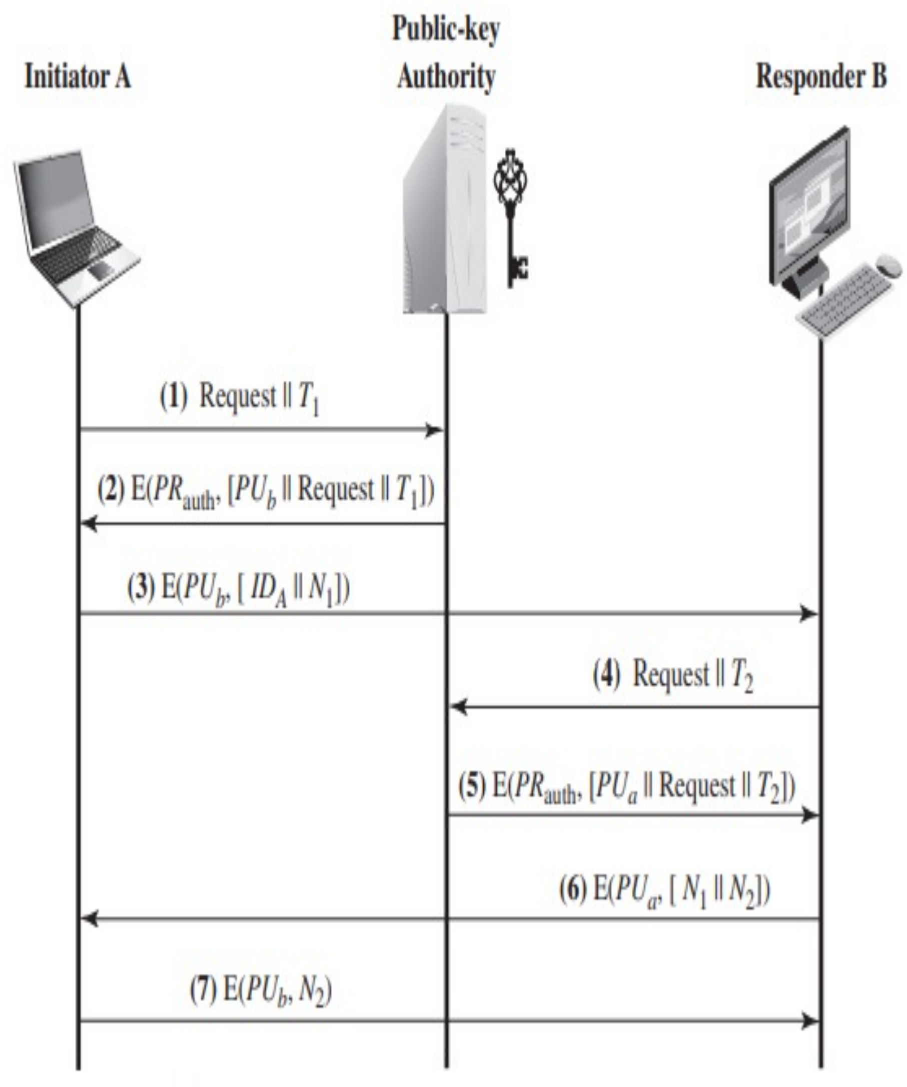
3. A stores B’s public key and also uses it to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.

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


Initiator A
Public-key Authority
Responder B
(1) Request ‖ T_1
(2) $E(PR_{auth}, [PU_b \| Request \| T_1])$
(3) $E(PU_b, [ID_A \| N_1])$
(4) Request ‖ T_2
(5) $E(PR_{auth}, [PU_a \| Request \| T_2])$

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.


Initiator A
Public-key Authority
Responder B
(1) Request || T_1
(2) E(PR_{auth}, [PU_b || Request || T_1])
(3) E(PU_b, [ID_A || N_1])
(4) Request || T_2
(5) E(PR_{auth}, [PU_a || Request || T_2])
(6) E(PU_a, [N_1 || N_2])
(7) E(PU_b, N_2)

Public-Key Distribution Scenario