Announcements

Homework 1

- Grade released
- Have ***1-week "rebuttal period"***
  - Submit re-grade request via GradeScope

Lecture 10

Protocols (Continued)

Chapters 9 and 11 in KPS

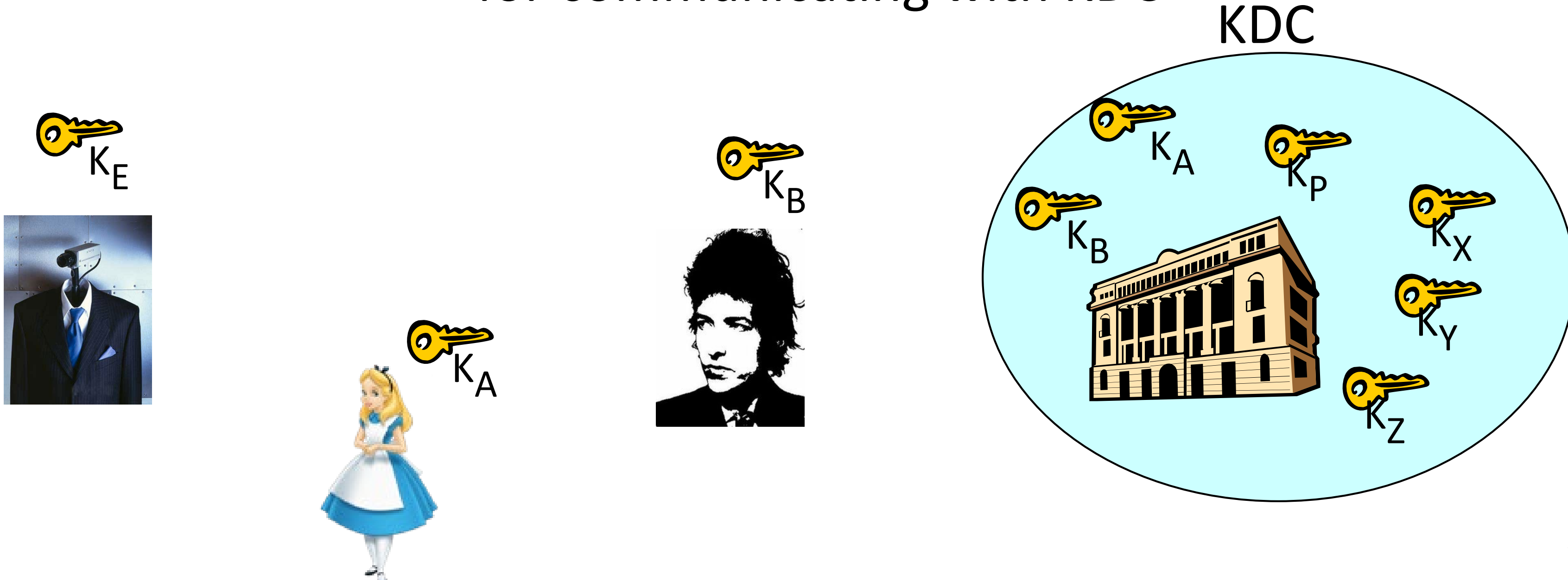
[lecture slides are adapted from previous slides by Prof. Gene Tsudik]

Recap: Key Distribution Center (KDC) aka Trusted Third Part (TTP)

- Alice and Bob need to share a key
- KDC shares different master key with *each* registered user (many users)
- Alice and Bob know their own master keys:

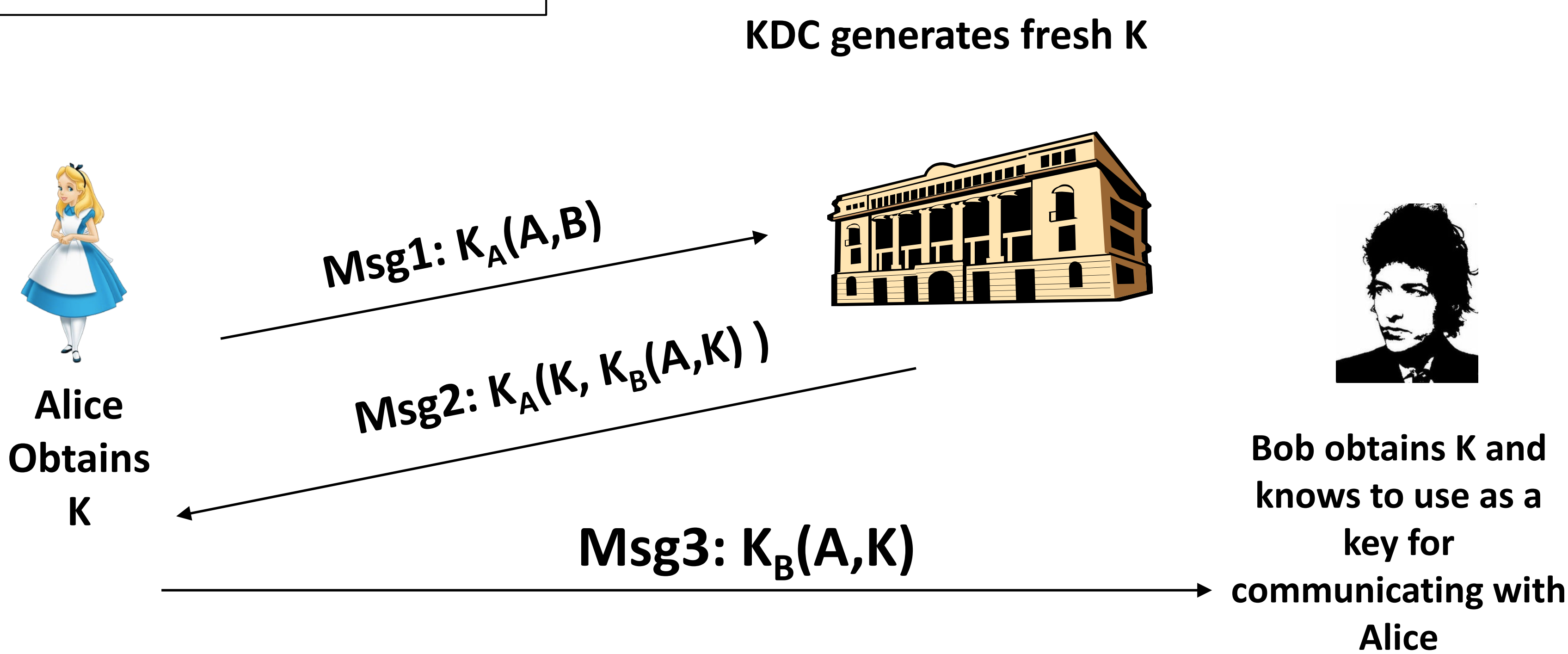
K_A and K_B

for communicating with KDC



Key Distribution Center (KDC) or Trusted Third Party (TTP)

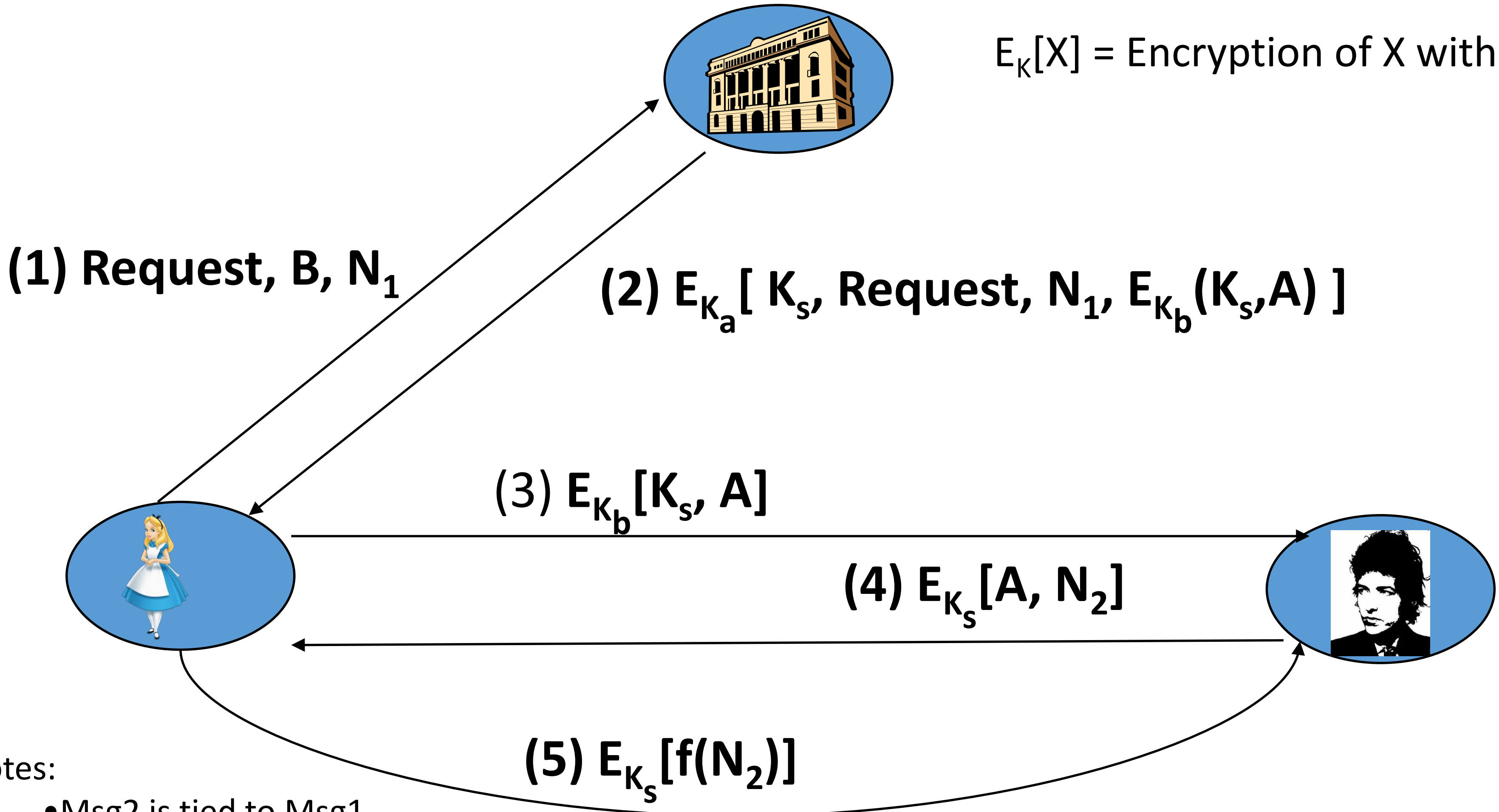
K(X) = Encryption of X with key K



- Alice and Bob communicate using **K** as a short-term (*session) key* for encryption and/or data integrity
- Note:
  - Msg2 is not tied to Msg1
  - Msg1 is possibly old
  - Msg2 is possibly old and so is Msg3
  - Bob and Alice don’t authenticate each other!

A Typical Key Distribution Scenario

$E_K[X]$ = Encryption of X with K



Notes:

- Msg2 is tied to Msg1
- Msg2 is fresh/new
- Msg3 is possibly old *
- Msg1 is possibly old (KDC doesn’t authenticate Alice)
- Bob authenticates Alice
- Bob authenticates KDC
- Alice DOES NOT authenticate Bob

Public Key Distribution

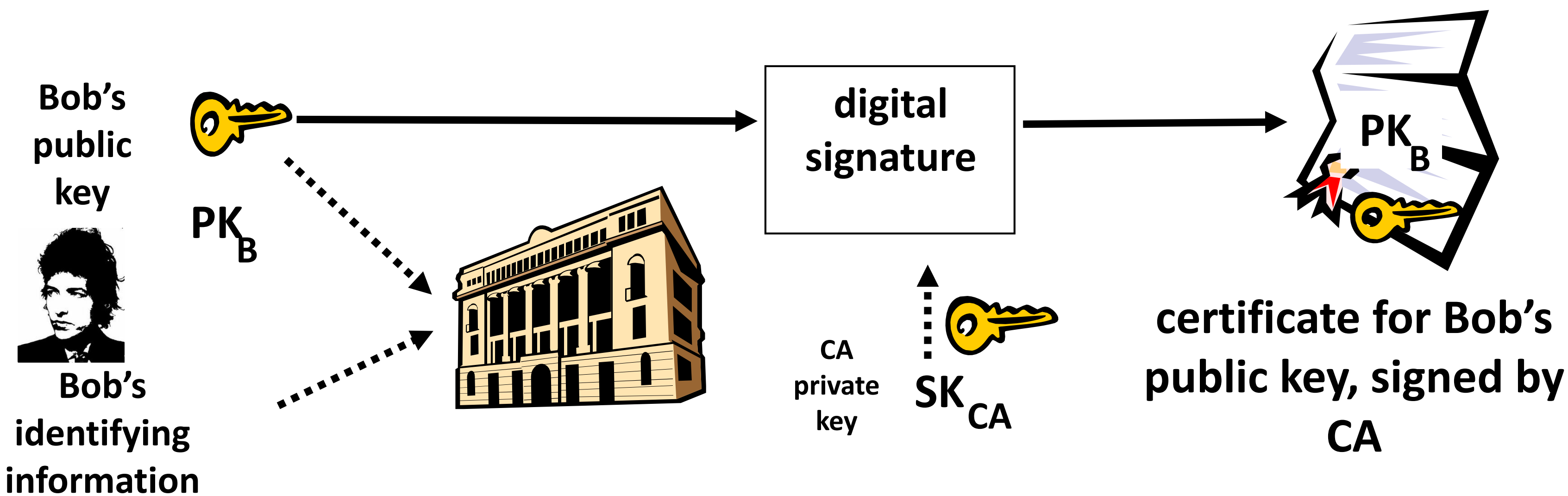
General schemes:

- Public announcement (e.g., in a newsgroup or email message)
  - Can be forged
- Publicly available directory
  - Can be tampered with
- Public-key certificates (PKCs) issued by trusted off-line Certification Authorities (CAs)

Certification Authorities

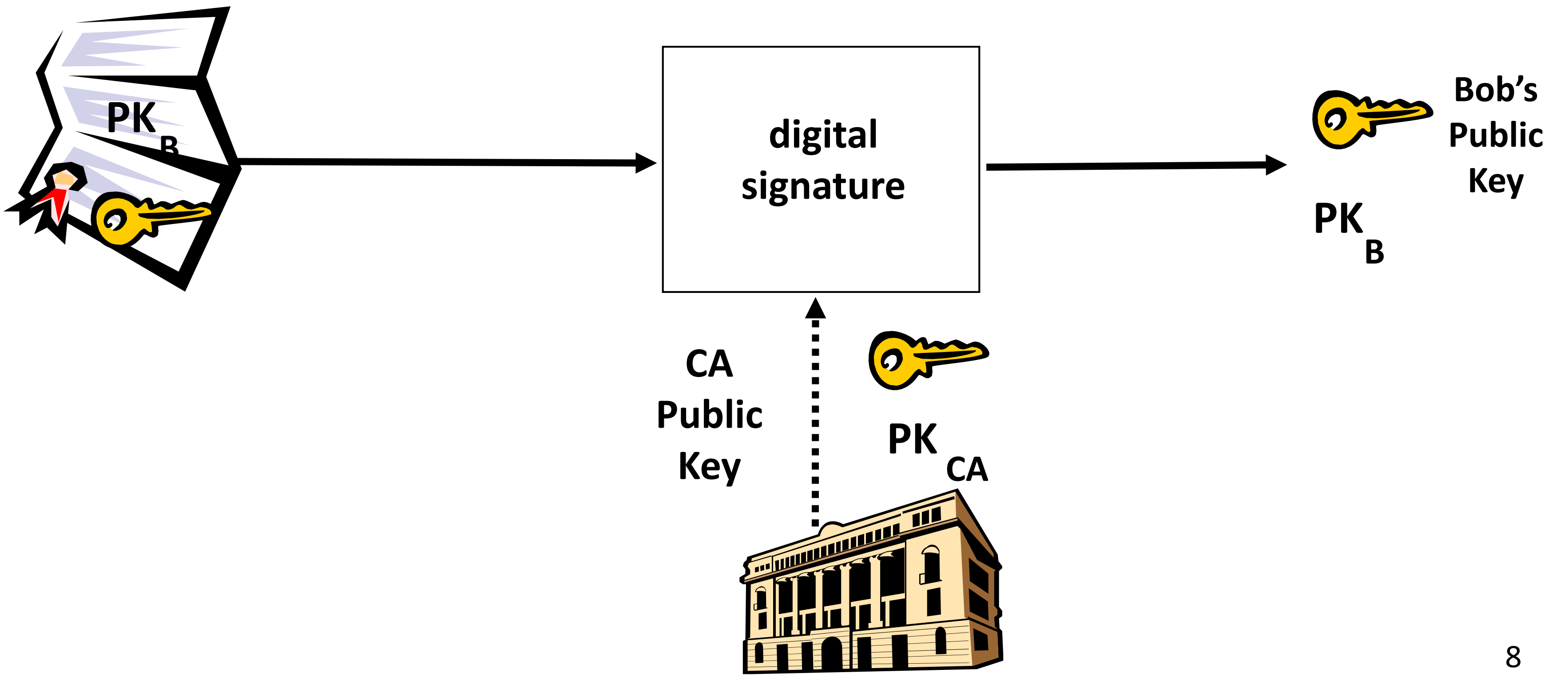
- Certification authority (CA): trusted, highly secure (physically and electronically) component
- Issues public key certificates; each binds a public key to a specific entity
- Each entity (user, host, etc.) registers its public key with CA.
  - Bob provides "proof of identity" to CA.
  - CA creates public key certificate binding Bob's ID/name to this public key.
  - Certificate containing Bob's public key is signed by CA:

CA says: "this is Bob's public key"



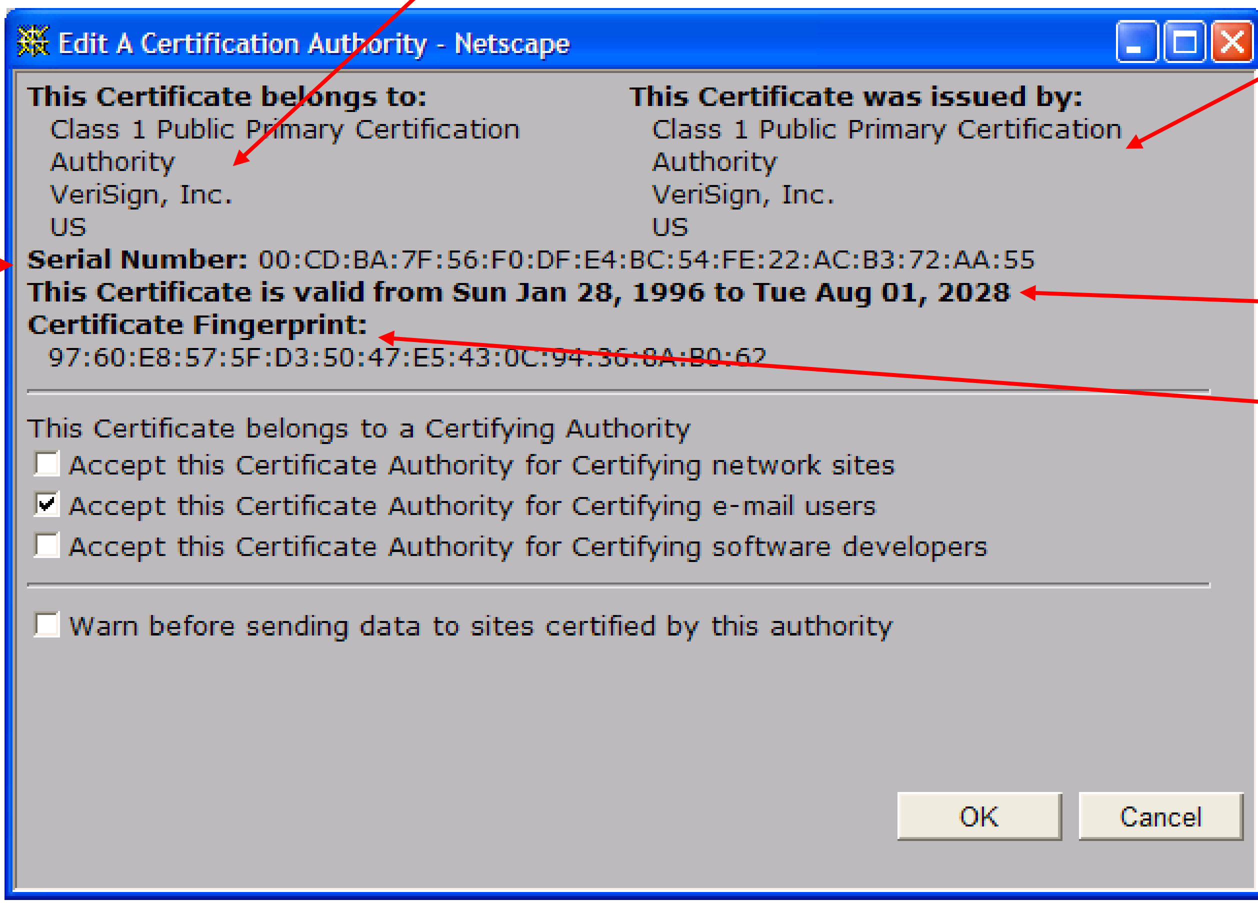
Certification Authority

- **When Alice wants to get Bob's public key:**
  - **Get Bob's certificate (from Bob or elsewhere)**
  - **Using CA's public key verify the signature on Bob's certificate**
  - **Check for expiration**
  - **Check for revocation (we'll talk about this later)**
  - **Extract Bob's public key**



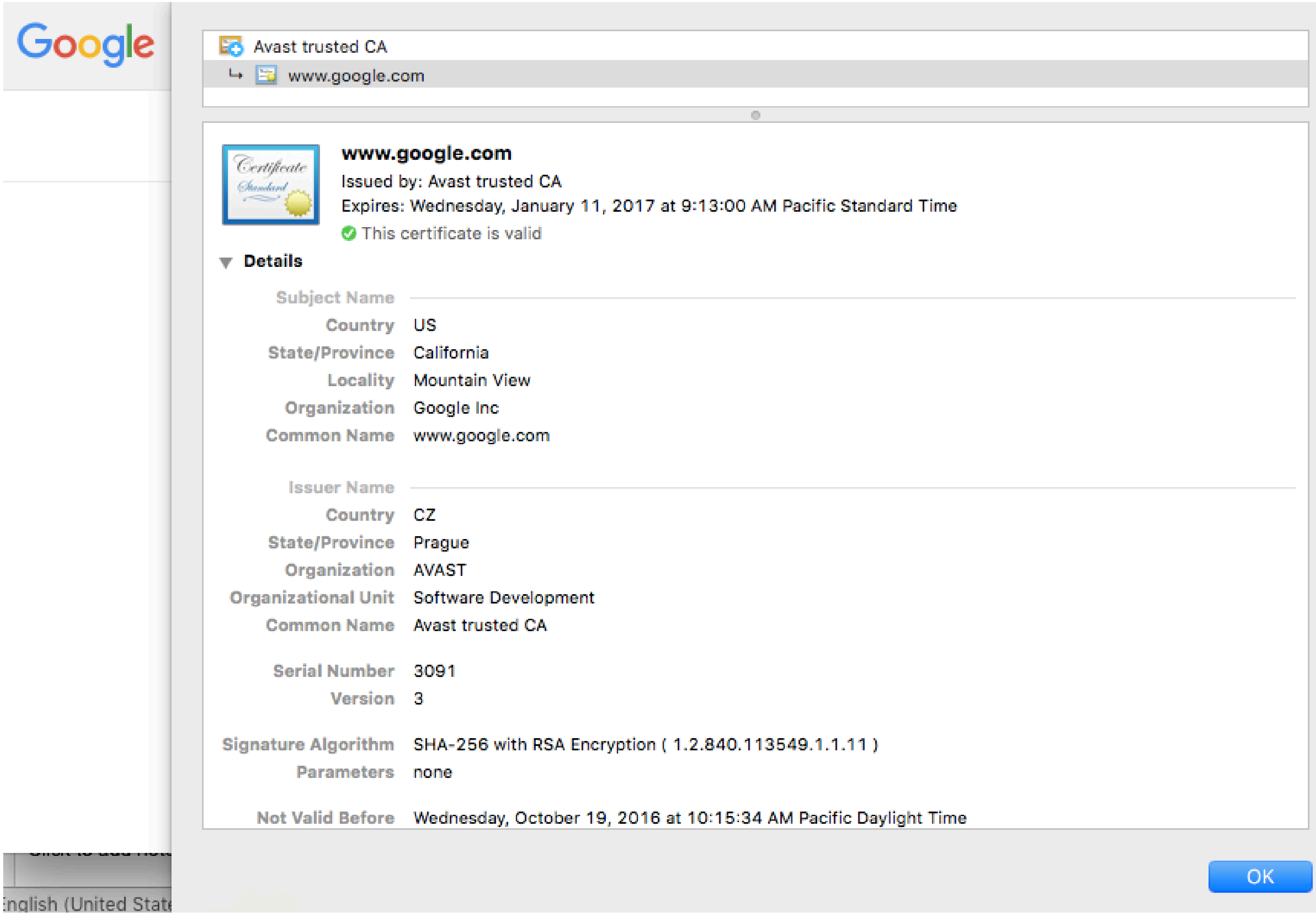
A Certificate Contains

- Serial number (unique to issuer)
- Info about certificate owner, including algorithm and key value itself (not shown)

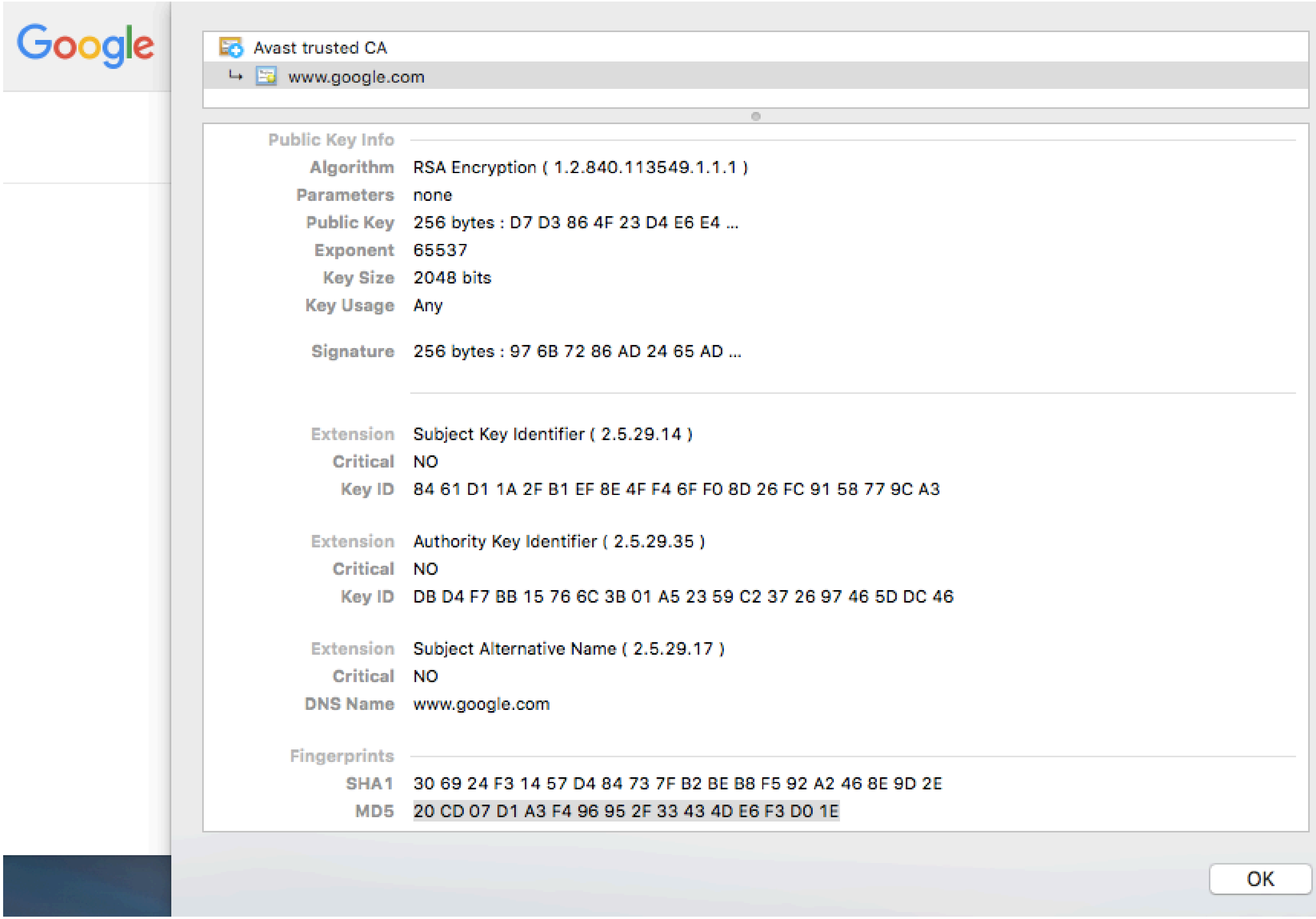


- info about certificate issuer
- valid dates
- digital signature by issuer

A Sample Certificate (1/2)



A Sample Certificate (2/2)

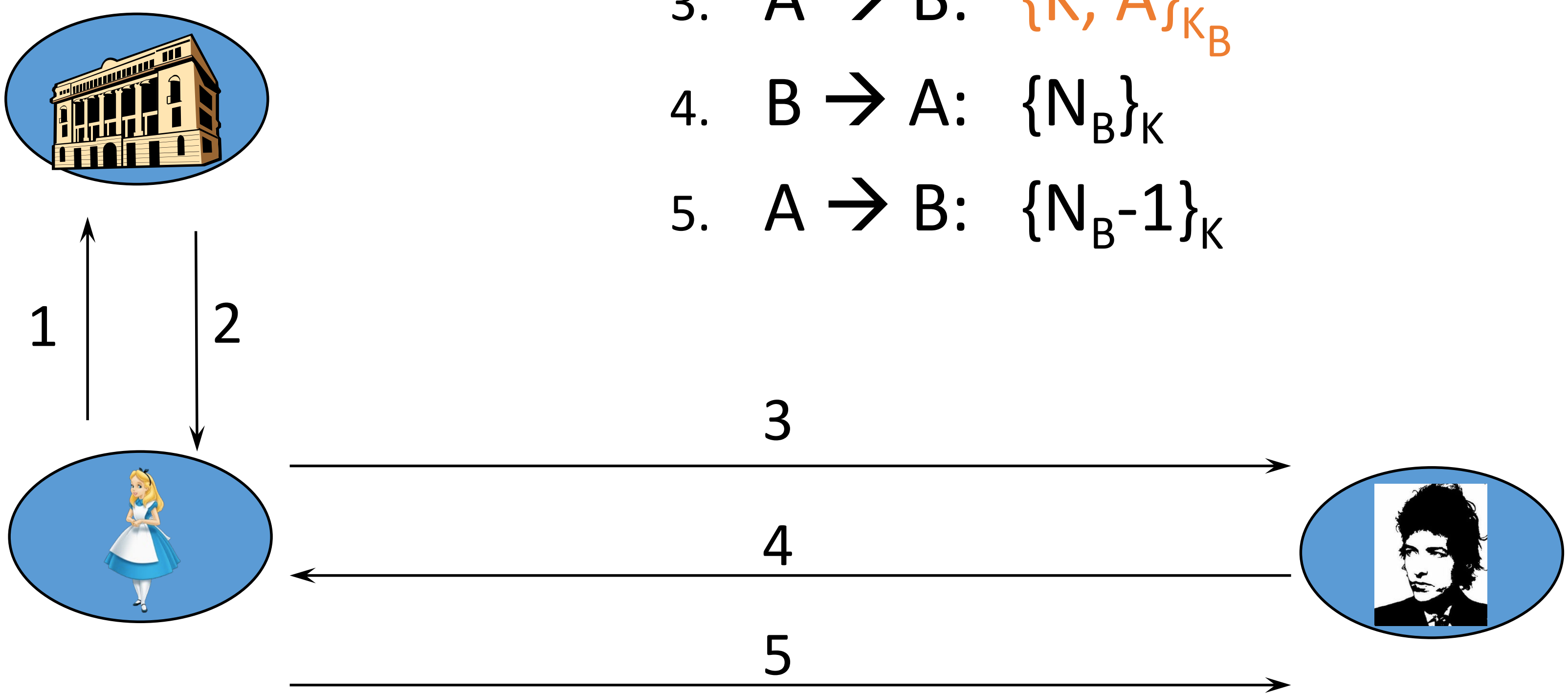


Back to Protocols

Needham-Schroeder Protocol (1978): First Distributed Security Protocol

$\{X\}_K$ = Encryption of X with key K

1. $A \rightarrow T$: A, B, N_A
2. $T \rightarrow A$: $\{N_A, B, K, \{K, A\}_{K_B}\}_{K_A}$
3. $A \rightarrow B$: $\{K, A\}_{K_B}$
4. $B \rightarrow A$: $\{N_B\}_K$
5. $A \rightarrow B$: $\{N_B-1\}_K$



Security?

Denning-Sacco Attack: suppose Eve recorded **an old** protocol session for which she somehow knows the session key K‘:

1. A → T: A, B, N_A
2. T → A: $\{N_A, B, K', \{K', A\}_{K_B}\}_{K_A}$
3. A → B: $\{K', A\}_{K_B}$

At a later time:

3. E → B: $\{K', A\}_{K_B}$
4. B → E: $\{N_B\}_{K'}$
5. E → B: $\{N_B - 1\}_{K'}$

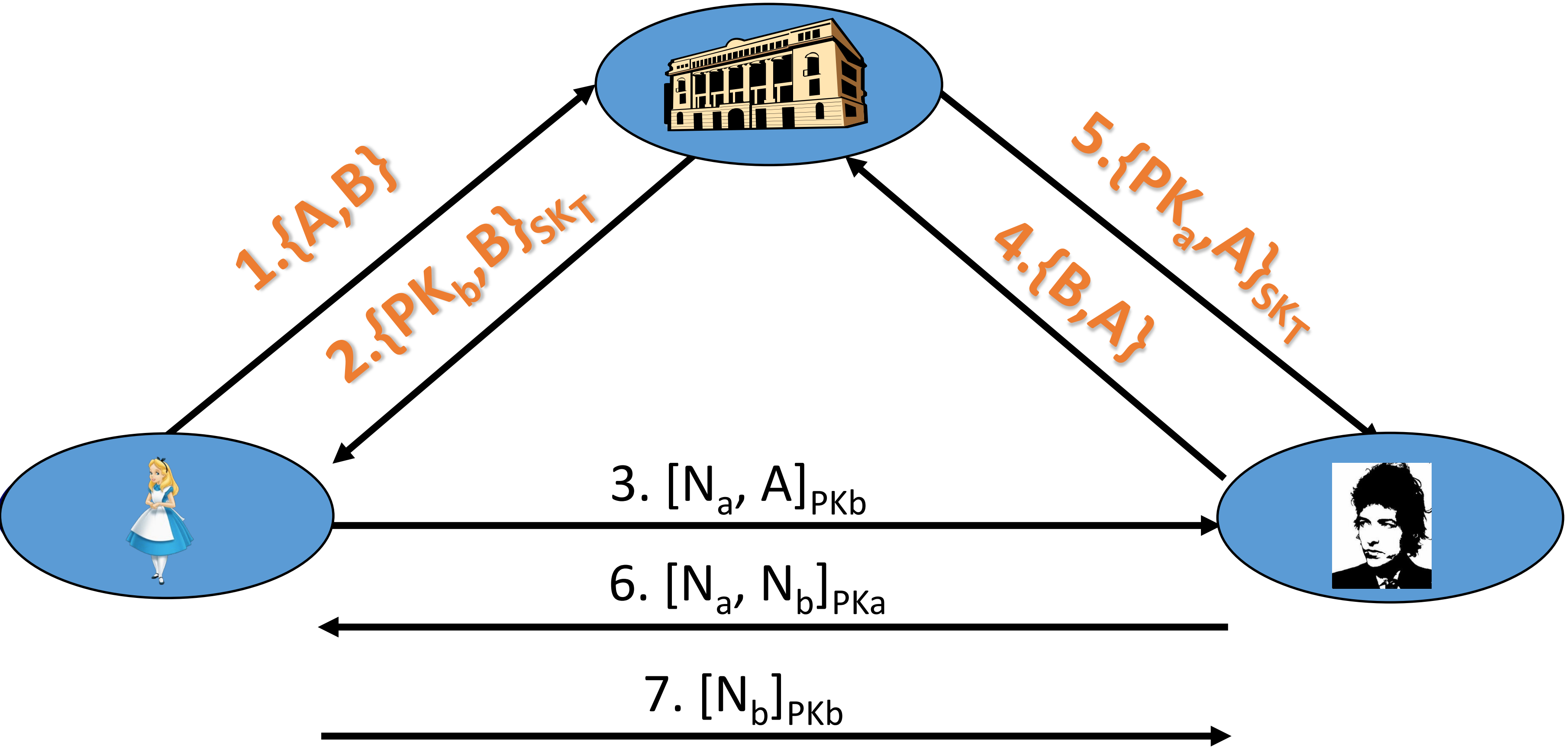
Fixing the Attack

- Bob has no guarantees about freshness of the message in step 3.
- Eve exploits this to impersonate Alice to Bob - old session keys are useful.
- Can be fixed by adding timestamps:
  - Limits usefulness of old session keys
  - Eve's attack becomes:

3: E → B: $\{K', T', A\}_{K_B}$

attack is now thwarted because T' is stale

PK-based Needham-Schroeder Protocol



- $CERT_B$ = Message 2, $CERT_A$ = Message 5
- PK_A: Alice's public key, PK_B: Bob's public key
- SK_T: TTP's secret (private) key used for signing
- Everyone knows TTP's public key PK_T

$[X]_K$ = Encryption of X with key K

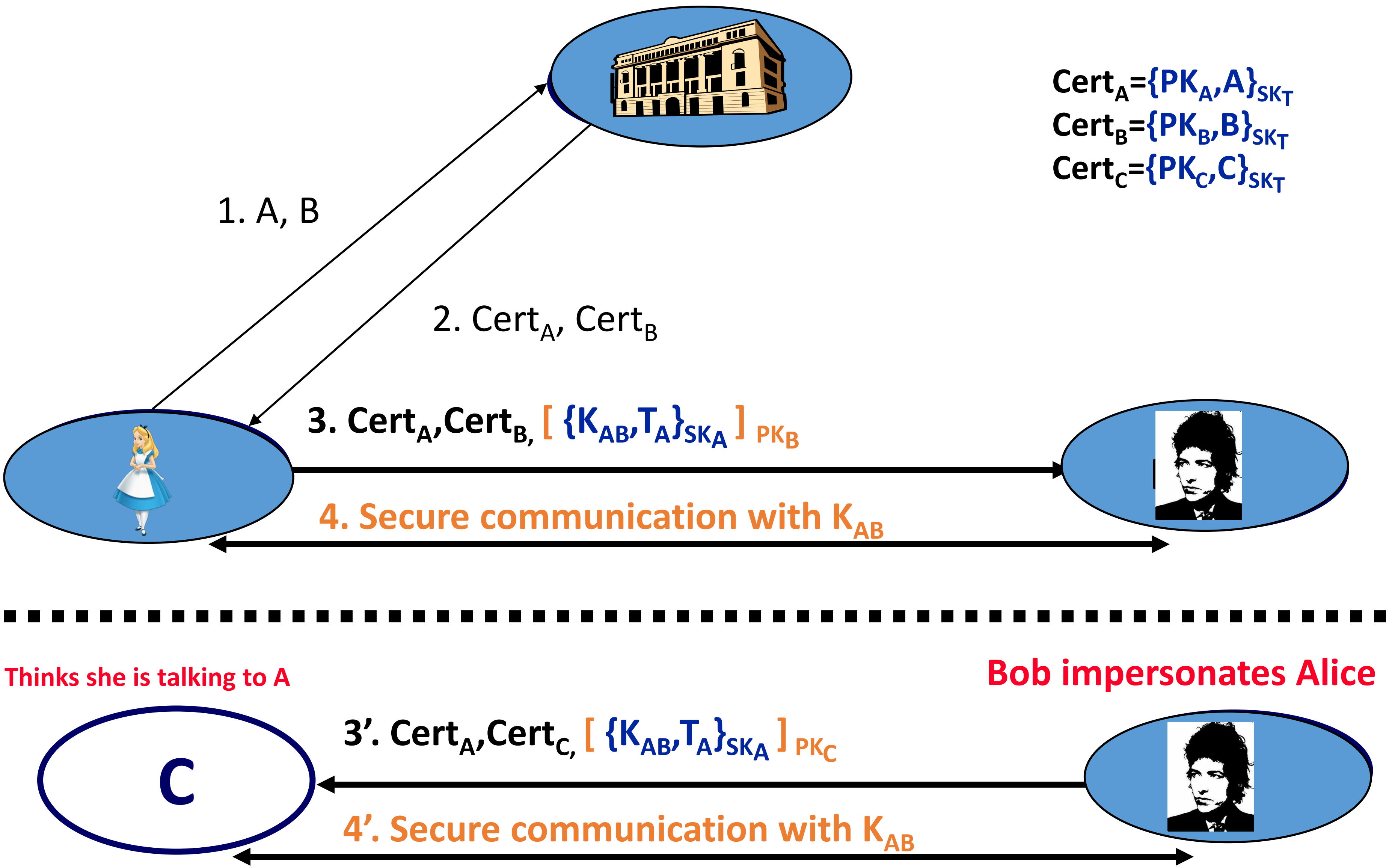
Another Attack

- 1, 2, 4, 5: Delivery of public key
- Does not guarantee freshness of the public key

How to solve it?

- Timestamp in messages 2 and 5 or challenges in messages 1&2 and 4&5
- Public Key Certificate: assign expiration time/data to each certificate (messages 2 and 5)

PK-based Denning-Sacco Attack



Lowe's Attack (Impersonation by Interleaving)

Original

3. $A \rightarrow B$: $[N_a, A]_{PKb}$

6. $B \rightarrow A$: $[N_a, N_b]_{PKa}$

7. $A \rightarrow B$: $[N_b]_{PK_b}$

Fix

3. $A \rightarrow B$: $[N_a, A]_{PKb}$

6. $B \rightarrow A$: $[B, N_a, N_b]_{PKa}$

7. $A \rightarrow B$: $[N_b]_{PK_b}$

Attack: E impersonates A

3. $A \rightarrow E$: $[N_a, A]_{Pke}$

3. $E \rightarrow B$: $[N_a, A]_{PKb}$

6. $B \rightarrow E$: $[N_a, N_b]_{Pka}$

6. $E \rightarrow A$: $[N_a, N_b]_{Pka}$

7. $A \rightarrow E$: $[N_b]_{Pke}$

7. $E \rightarrow B$: $[N_b]_{PK_b}$