Week 7

An Introduction to Cyber Security – CS 573

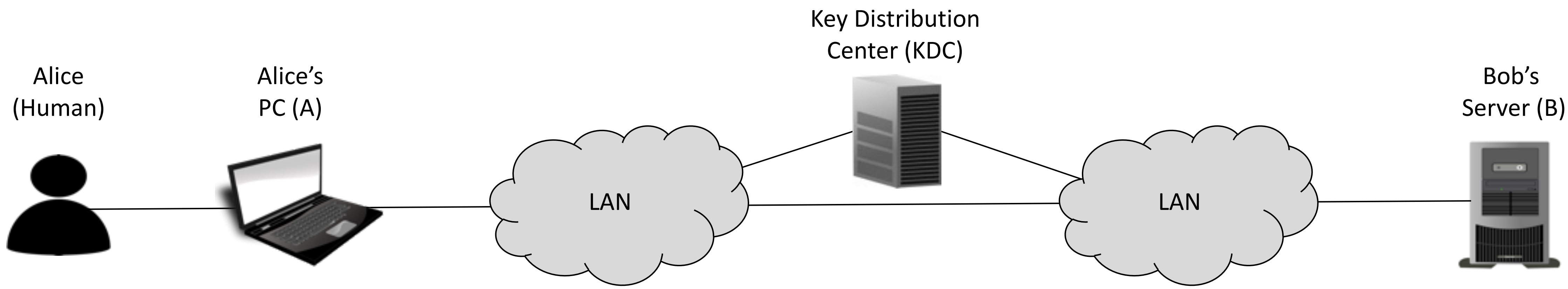
Instructor: Dr. Edward G. Amoroso
eamoroso@tag-cyber.com

Week 7

Week 7: Public Key Cryptography

How Does Kerberos Work?

Kerberos: A Complex Solution to a Simple Password Problem



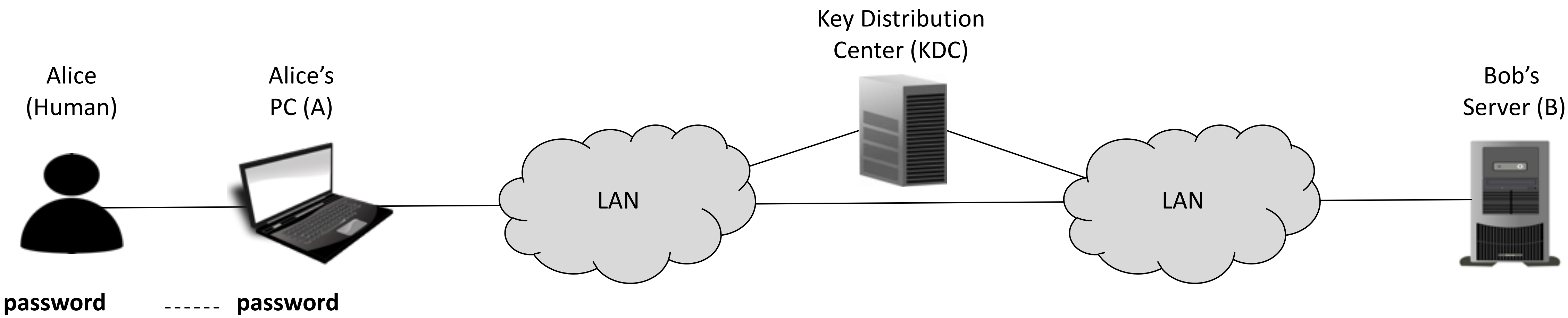
Basic Kerberos Concept:

- Invented at MIT in 1980's as part of Project Athena
- Goal is that Alice (client) can authenticate to Bob (server) without using a password on the local area network (LAN);
- Key Distribution Center (KDC) enables this process using conventional cryptography (i.e., no public key technology)
- **Used primarily for Single Sign-On (SSO)**

Week 7

Kerberos – Preconditions

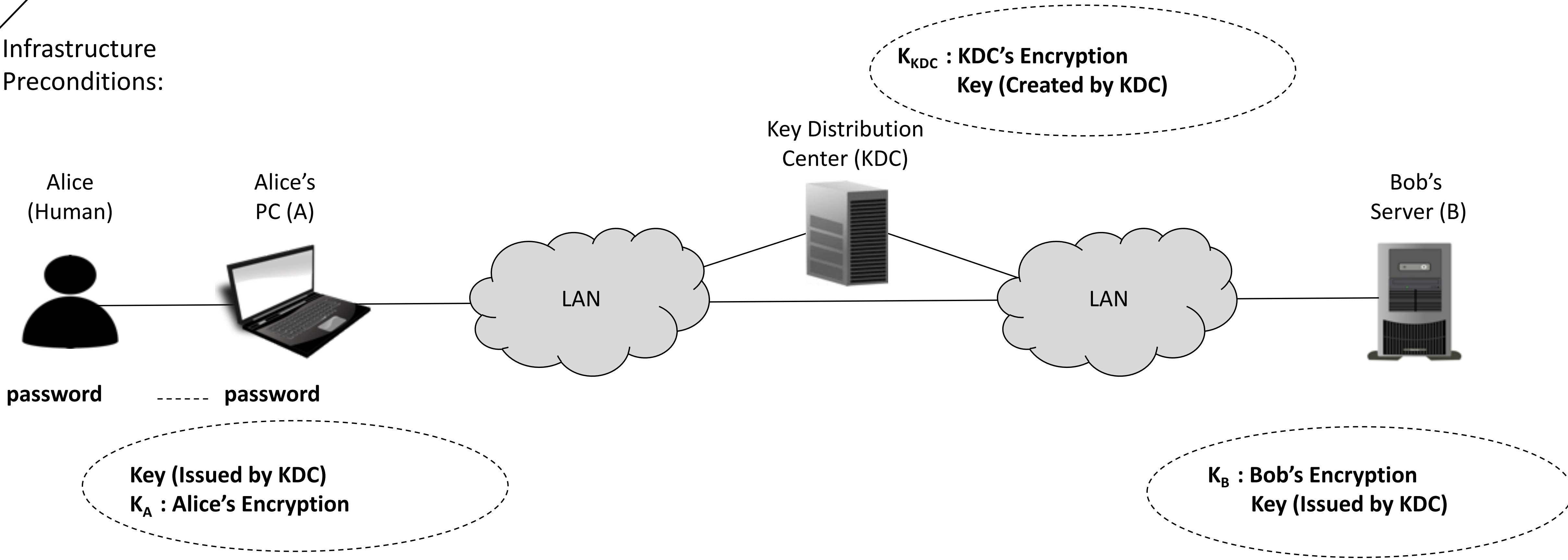
Infrastructure Preconditions:



Kerberos password set up for Alice to log into her PC (Never used over any LAN)

Kerberos – Keys

Infrastructure
Preconditions:

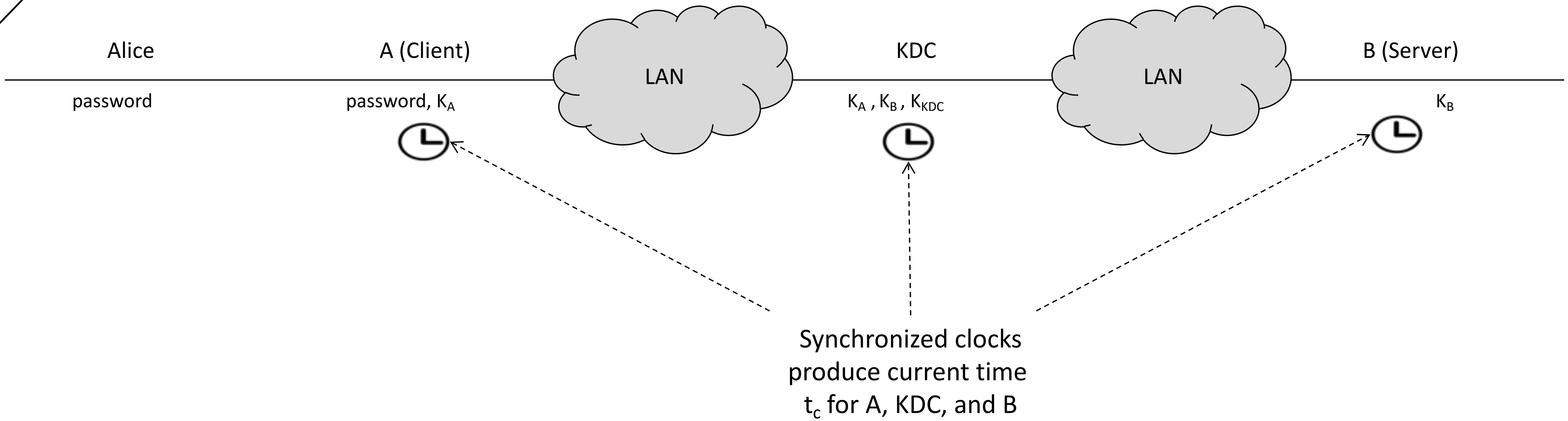


Kerberos KDC creates and issues
three cryptographic keys: $\mathbf{K_A}$, $\mathbf{K_B}$, $\mathbf{K_{KDC}}$

$\{ \{ m \}_{\underset{\text{Encrypt}}{K_A}} \}_{\underset{\text{Decrypt}}{K_A}} = m$ $\quad\{ \{ m \}_{K_B} \}_{K_B} = m$ $\quad\{ \{ m \}_{K_{KDC}} \}_{K_{KDC}} = m$

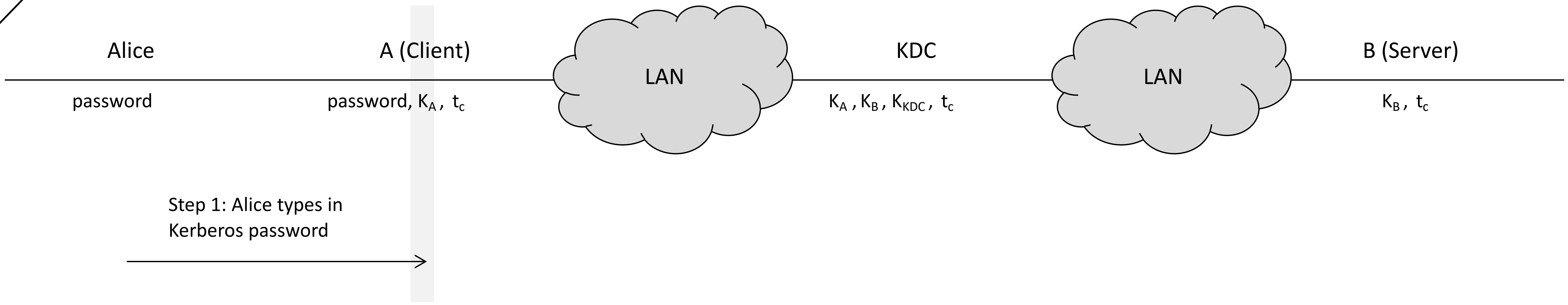
Week 7

Kerberos – Clocks

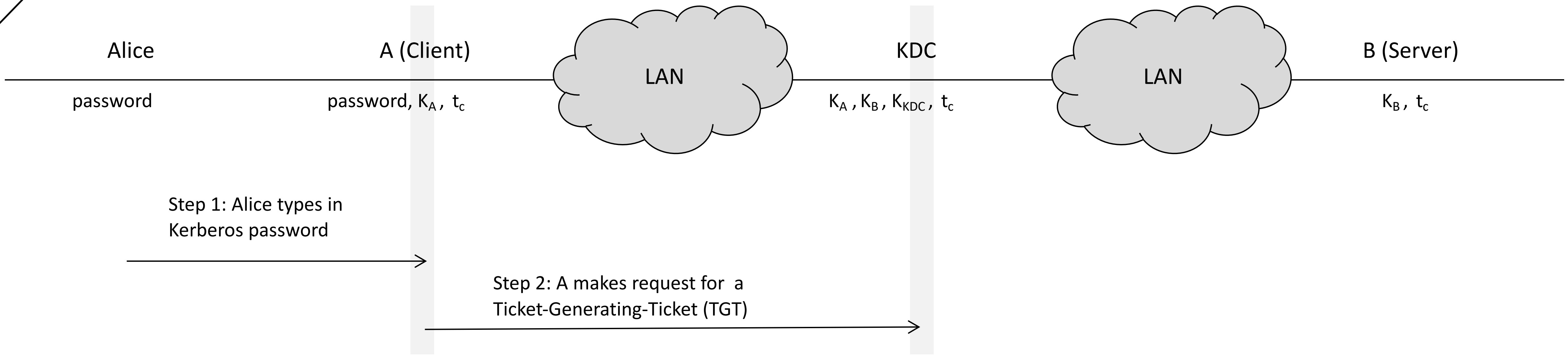


Week 7

Kerberos Step 1: Type in Local Password

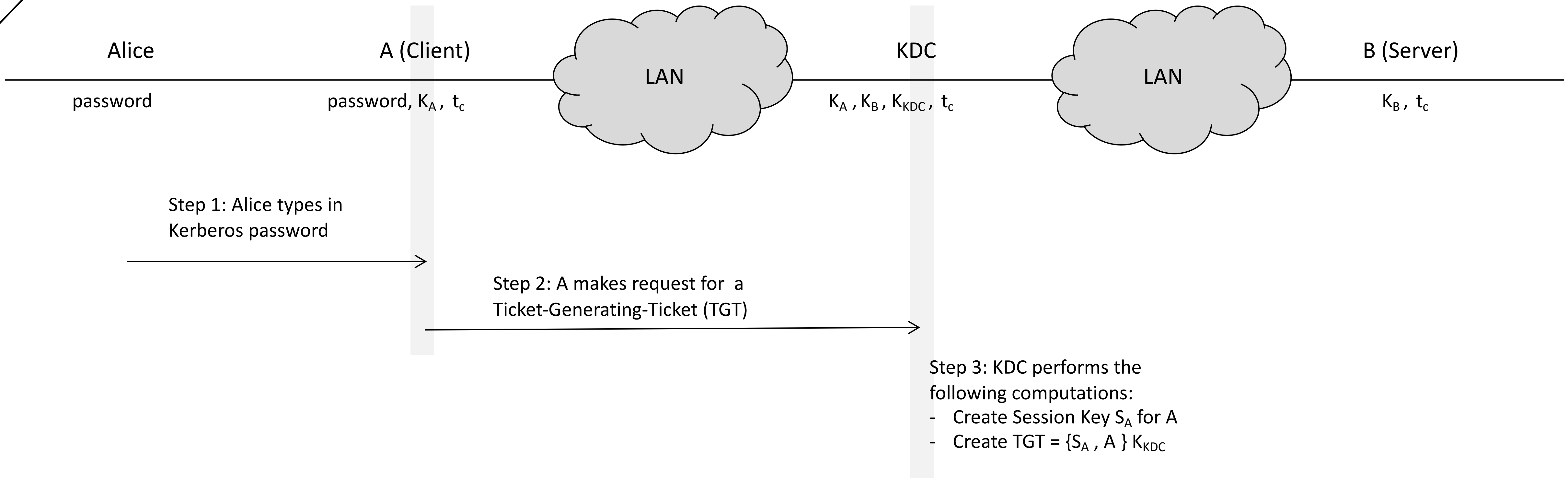


Kerberos Step 2: Request TGT and Session Key



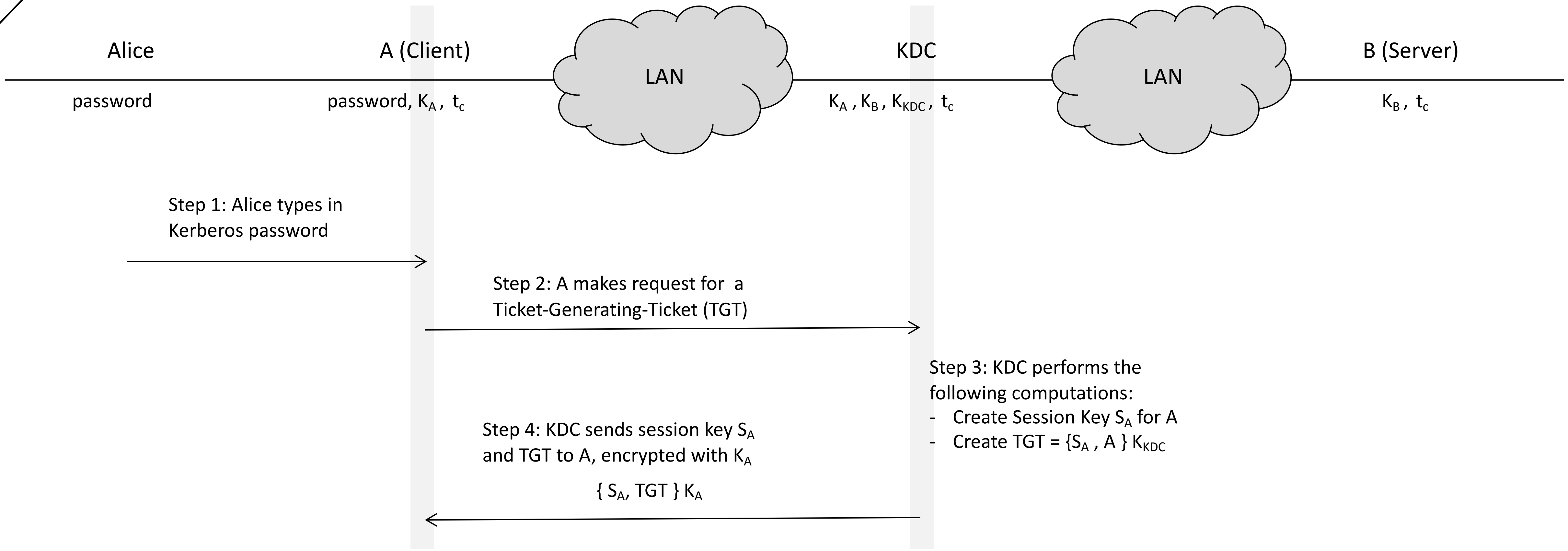
Week 7

Kerberos Step 3: Create Session Key and TGT



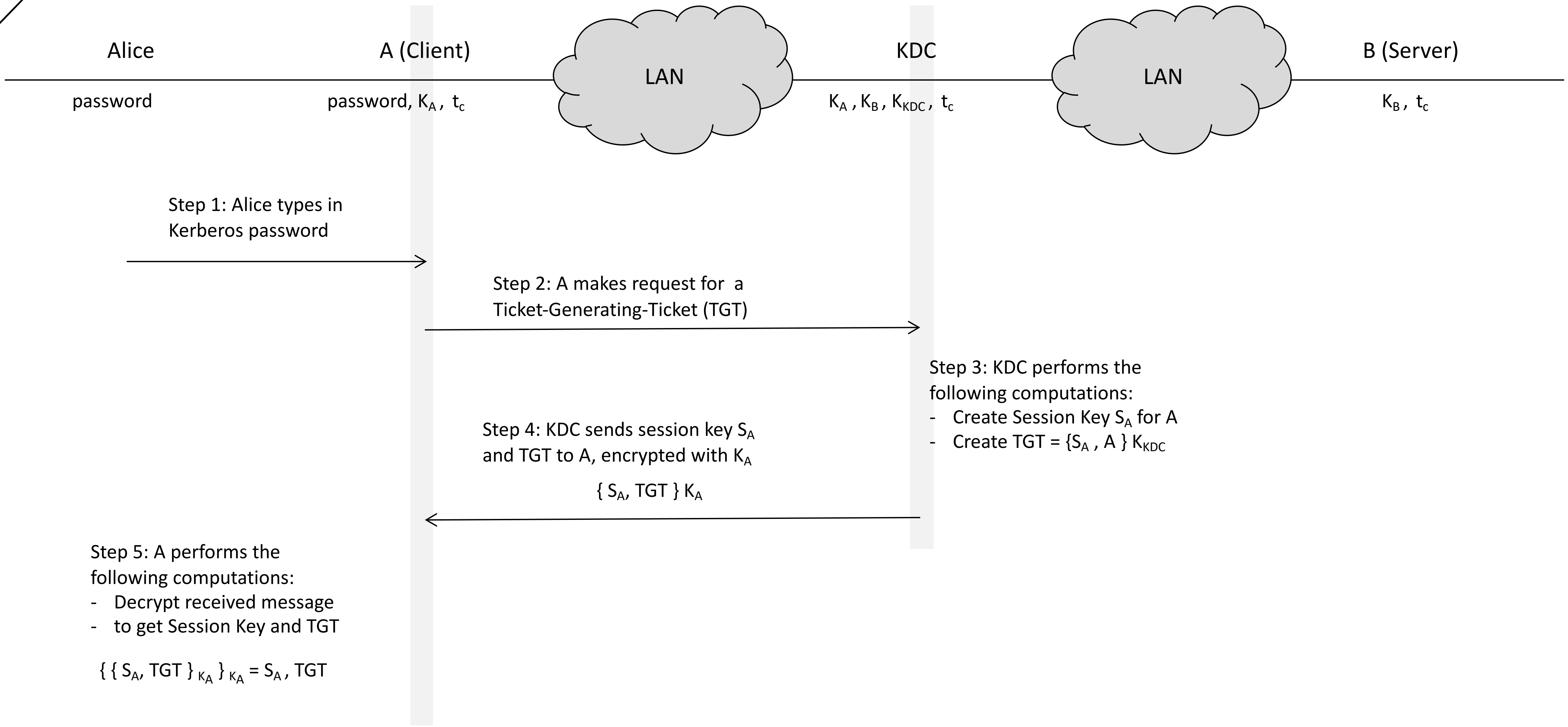
Week 7

Kerberos Step 4: Provide Session Key and TGT



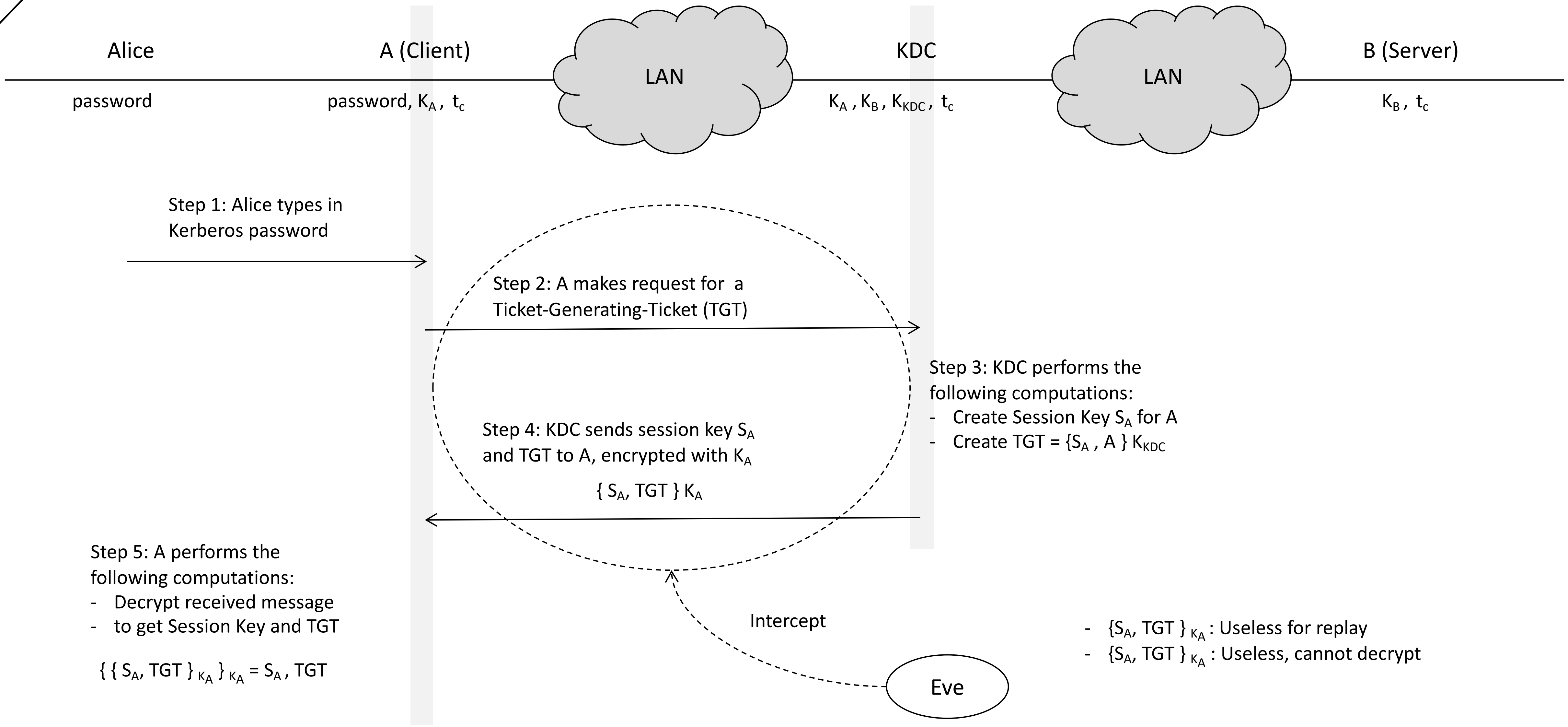
Week 7

Kerberos Step 5: Decrypt Session Key and Store TGT



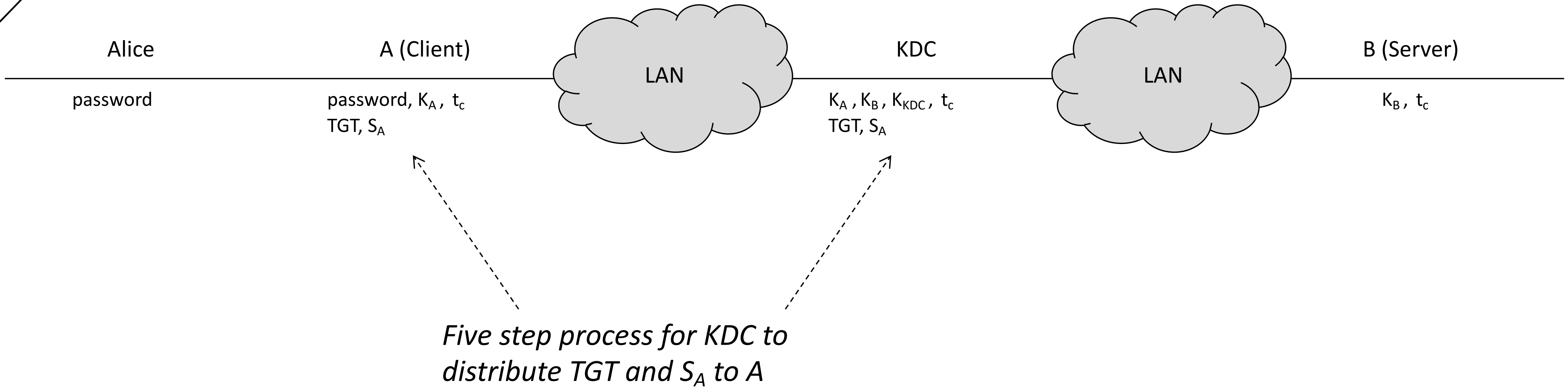
Week 7

Kerberos – Through Five Steps: Eve Cannot Hack

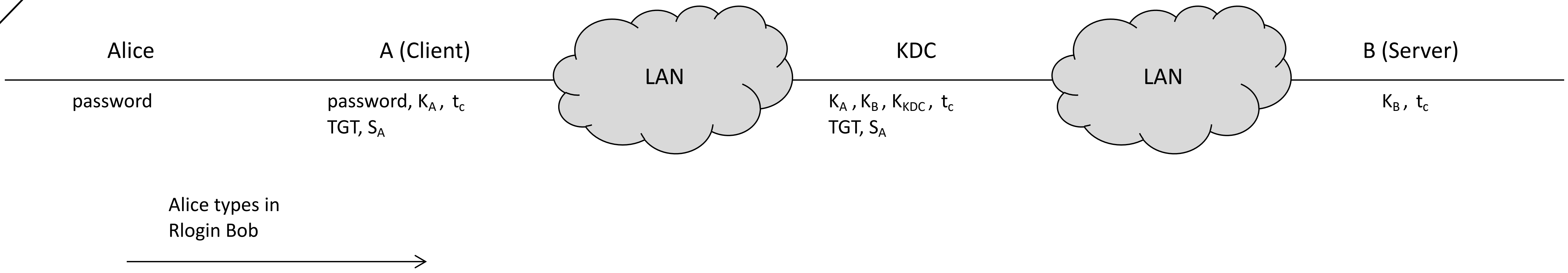


Week 7

Kerberos – Result of Five Steps

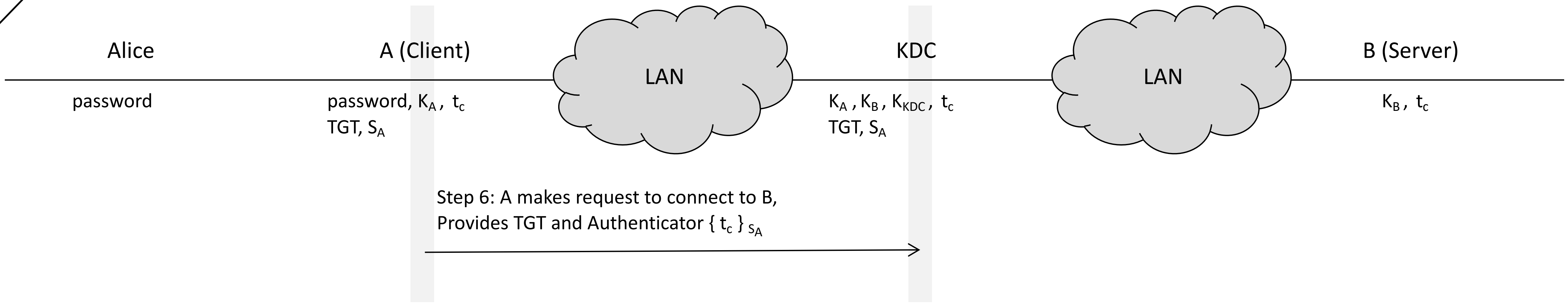


Kerberos – Result of Five Steps



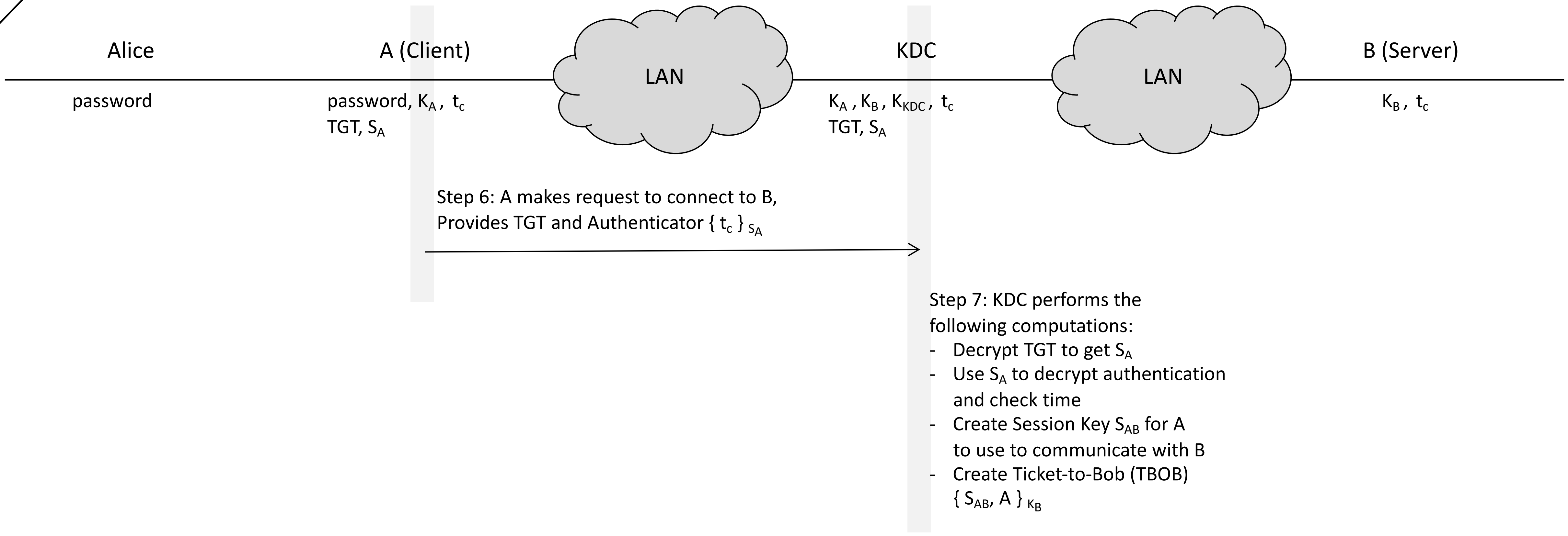
Week 7

Kerberos Step 6: Request Login to B



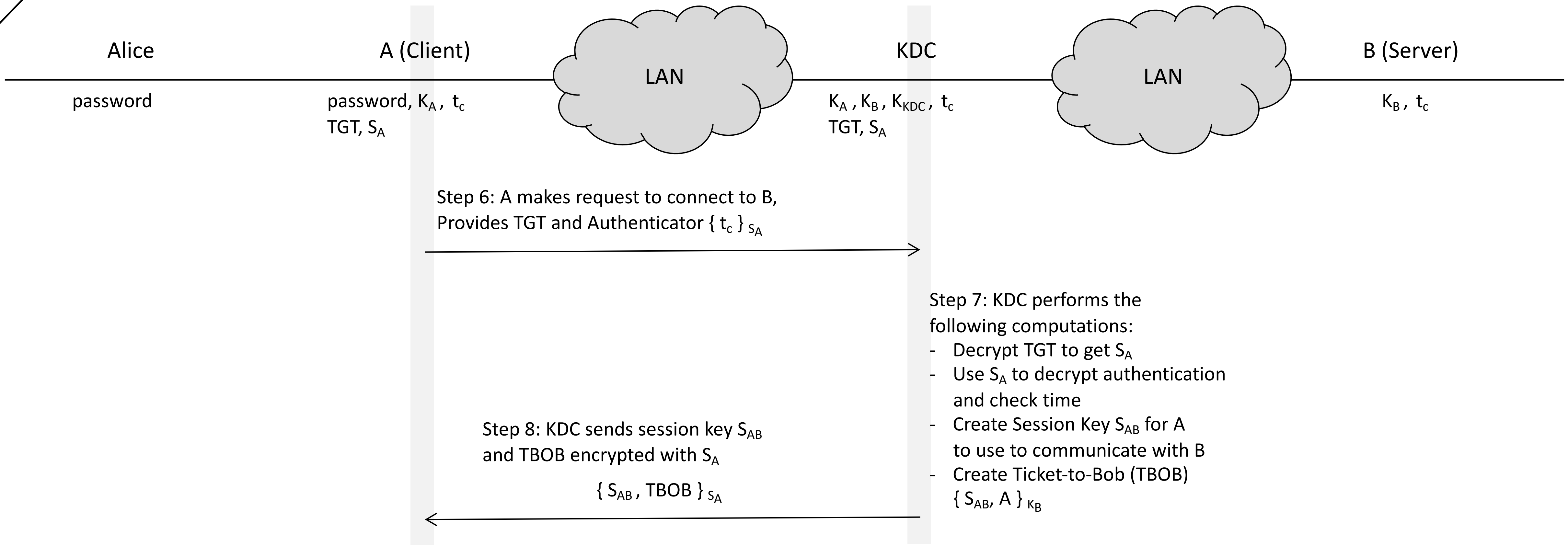
Week 7

Kerberos Step 7: Create Session Key and Ticket to Bob



Week 7

Kerberos Step 8: Send Session Key and Ticket to Bob



Week 7

Kerberos Step 9: Decrypt Session Key and Store Ticket to Bob

Alice | A (Client) | LAN | KDC | LAN | B (Server)

Alice: password

A (Client): password, K_A, t_c
TGT, S_A

KDC: K_A, K_B, K_{KDC}, t_c
TGT, S_A

B (Server): K_B, t_c

Step 6: A makes request to connect to B,
Provides TGT and Authenticator $\{ t_c \}_{S_A}$

Step 7: KDC performs the
following computations:
- Decrypt TGT to get S_A
- Use S_A to decrypt authentication
  and check time
- Create Session Key S_{AB} for A
  to use to communicate with B
- Create Ticket-to-Bob (TBOB)
  $\{ S_{AB}, A \}_{K_B}$

Step 8: KDC sends session key S_{AB}
and TBOB encrypted with S_A

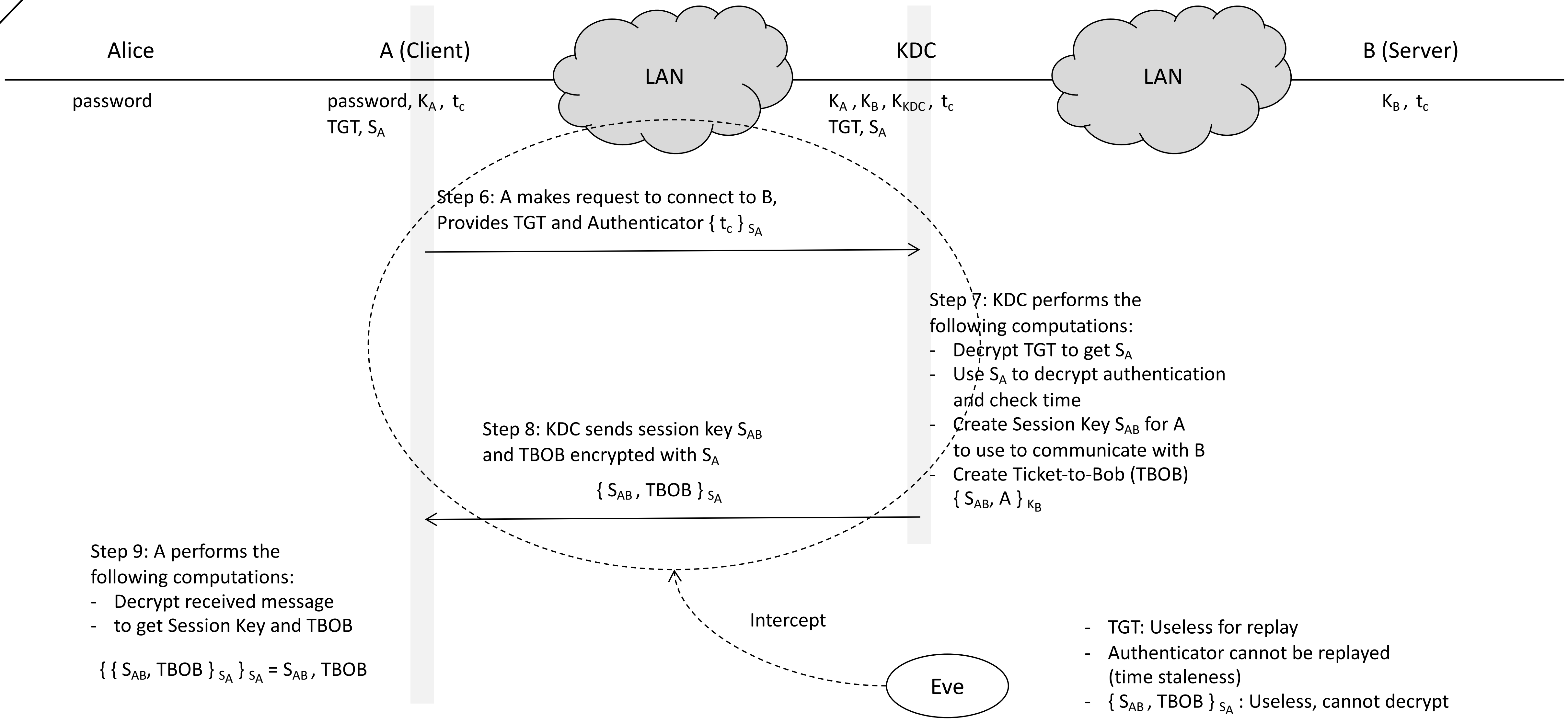
$\{ S_{AB}, TBOB \}_{S_A}$

Step 9: A performs the
following computations:
- Decrypt received message
- to get Session Key and TBOB

$\{ \{ S_{AB}, TBOB \}_{S_A} \}_{S_A} = S_{AB}, TBOB$

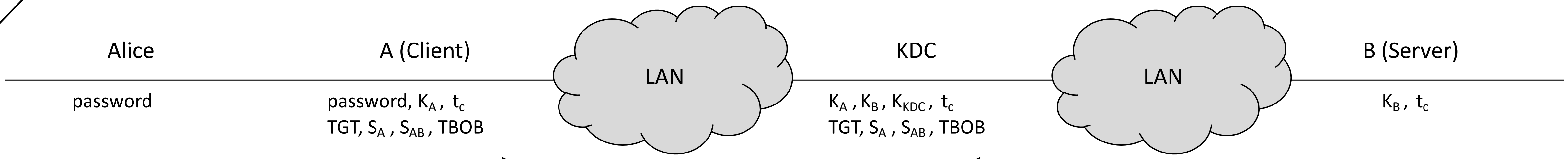
Week 7

Kerberos – Through Nine Steps: Eve Cannot Hack



Week 7

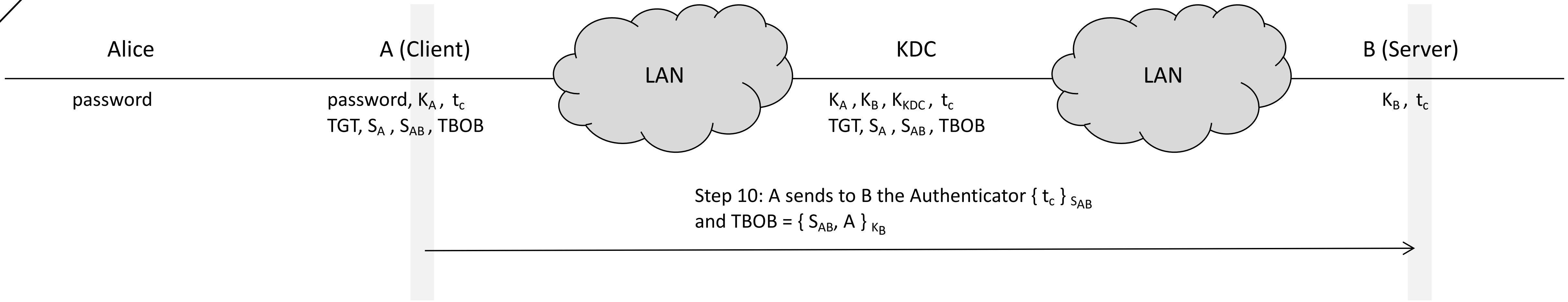
Kerberos – Result of Nine Steps



Four step process for KDC to distribute TBOB and S_{AB} to A

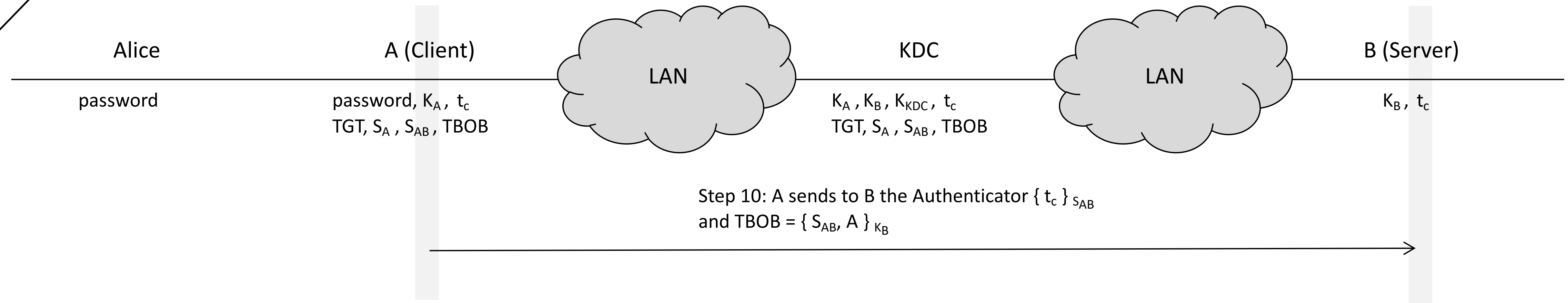
Week 7

Kerberos Step 10: Send Bob the Ticket to Bob



Week 7

Kerberos Step 11: Decrypt Ticket to Bob and Check Time

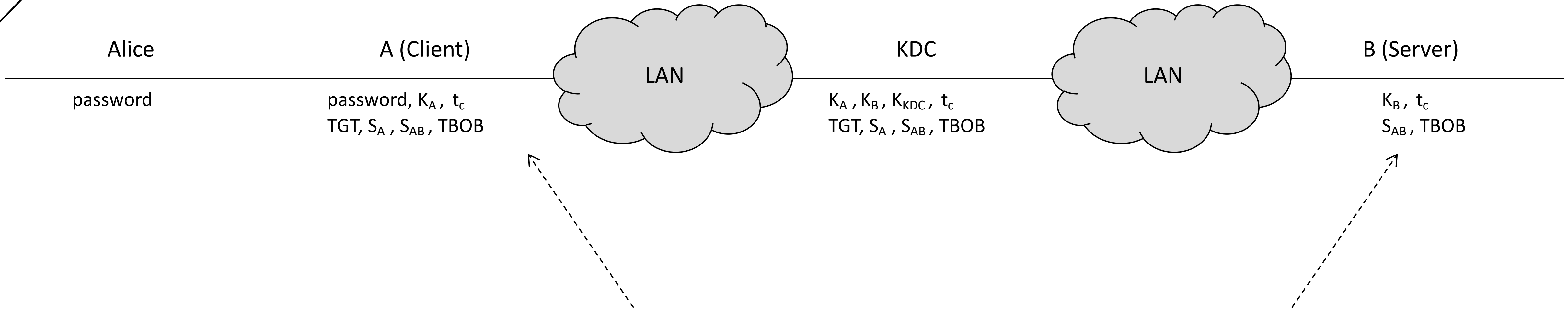


Step 11: B performs the following computations:

- Decrypt TBOB to get S_{AB}
- Use S_{AB} to decrypt authentication and check time

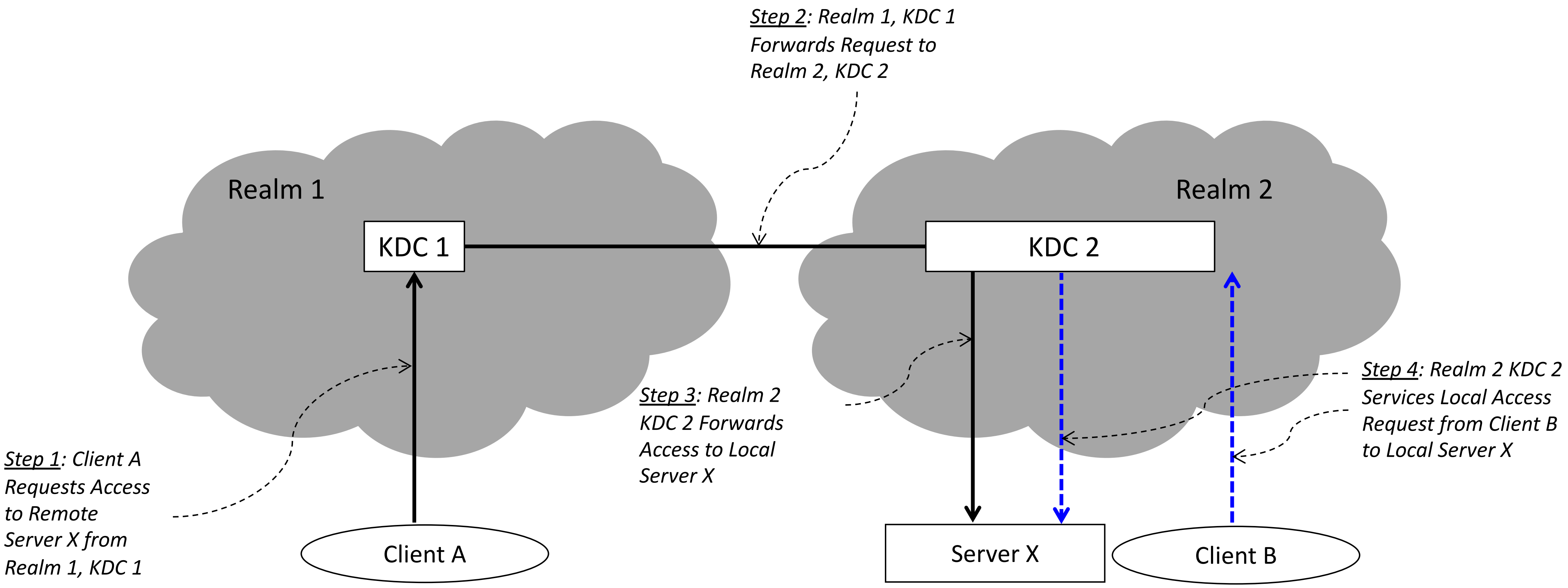
Week 7

Kerberos – Result of Eleven Steps



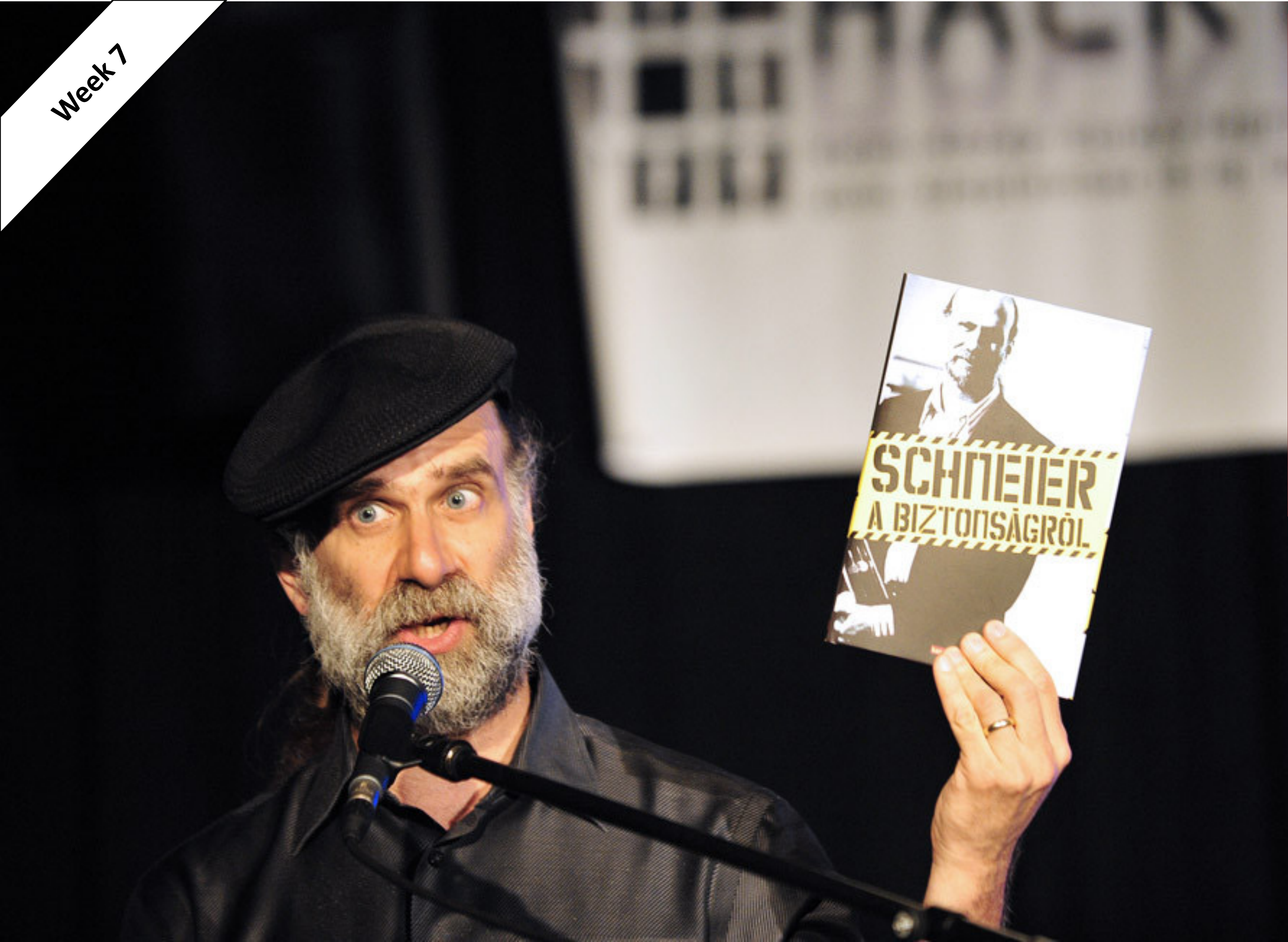
Two step process (plus nonce messages) for A to use TBOB to get S_{AB} to B

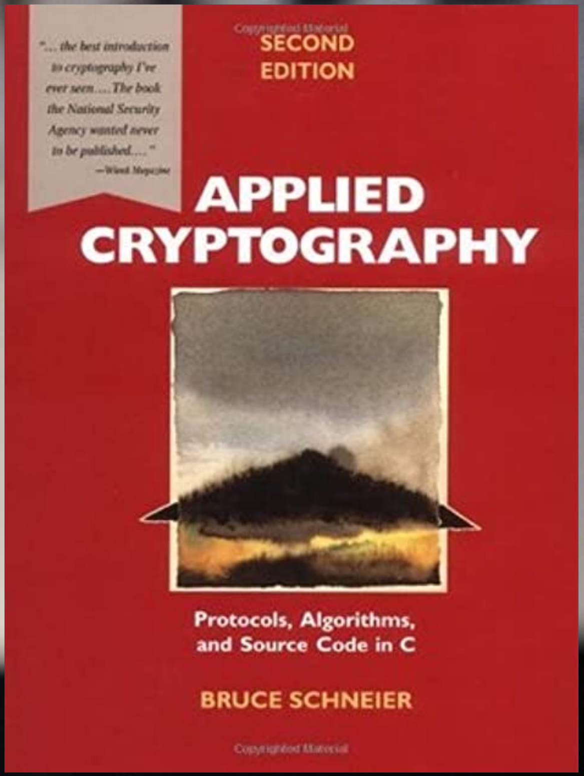
Kerberos – Realms



What Properties of Conventional Cryptography Must Be Maintained?

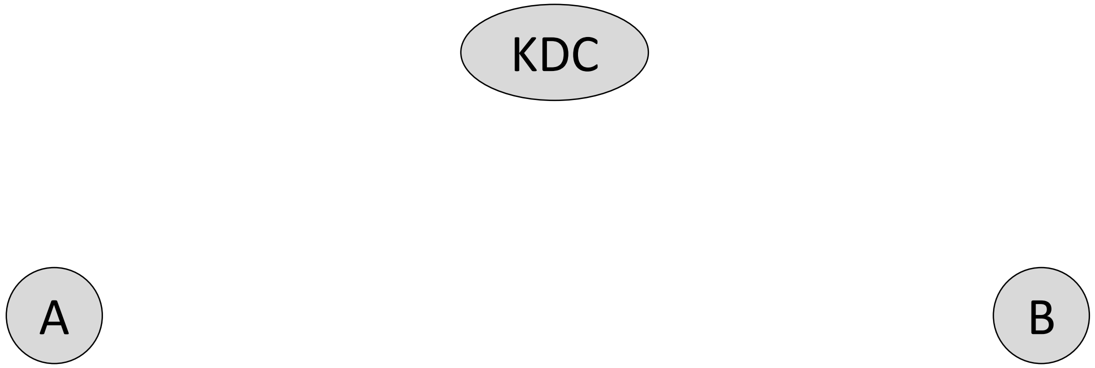
Week 7


SCHNEIER
A BIZTONSÁGRÓL

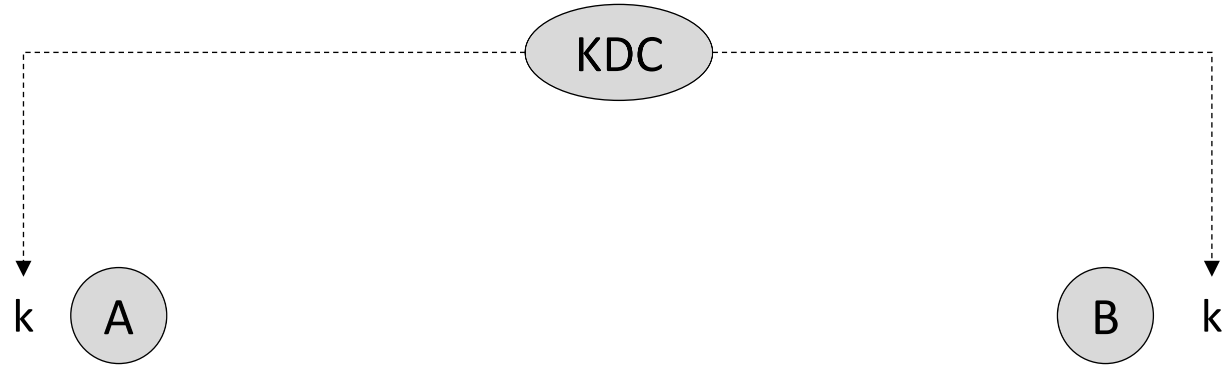

"... the best introduction to cryptography I've ever seen.... The book the National Security Agency wanted never to be published...."
—Wired Magazine
SECOND EDITION
APPLIED CRYPTOGRAPHY
Protocols, Algorithms, and Source Code in C
BRUCE SCHNEIER

Week 7

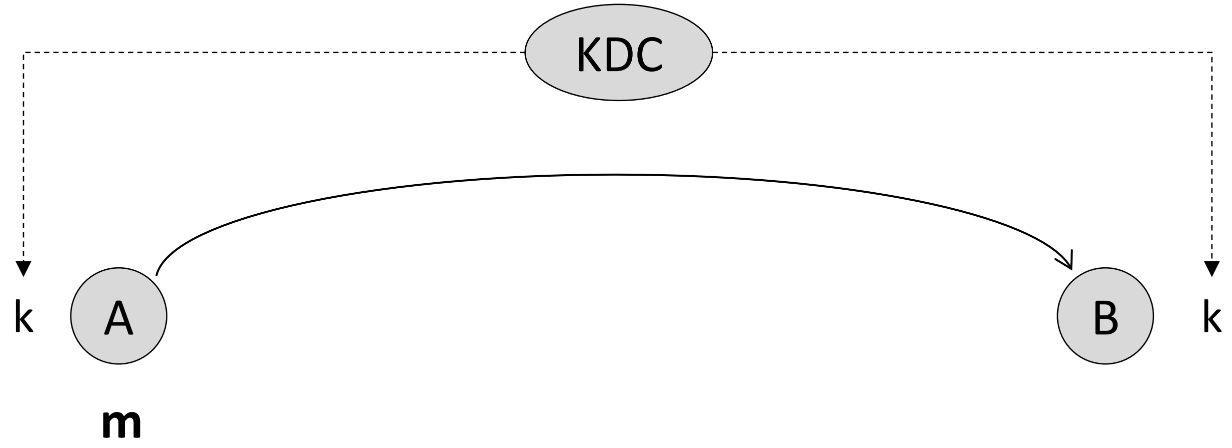
Conventional Cryptography



Conventional Cryptography



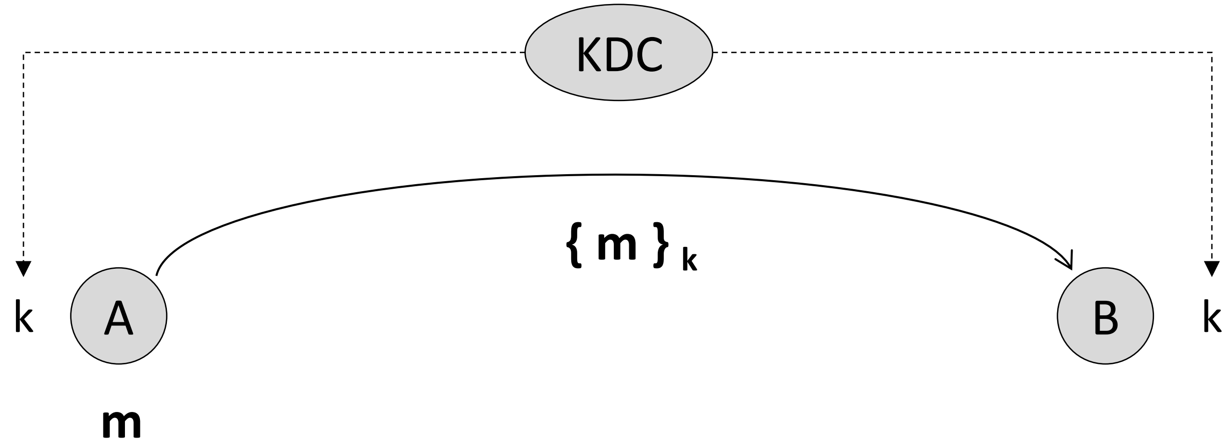
Conventional Cryptography



Alice creates message m . . .

Week 7

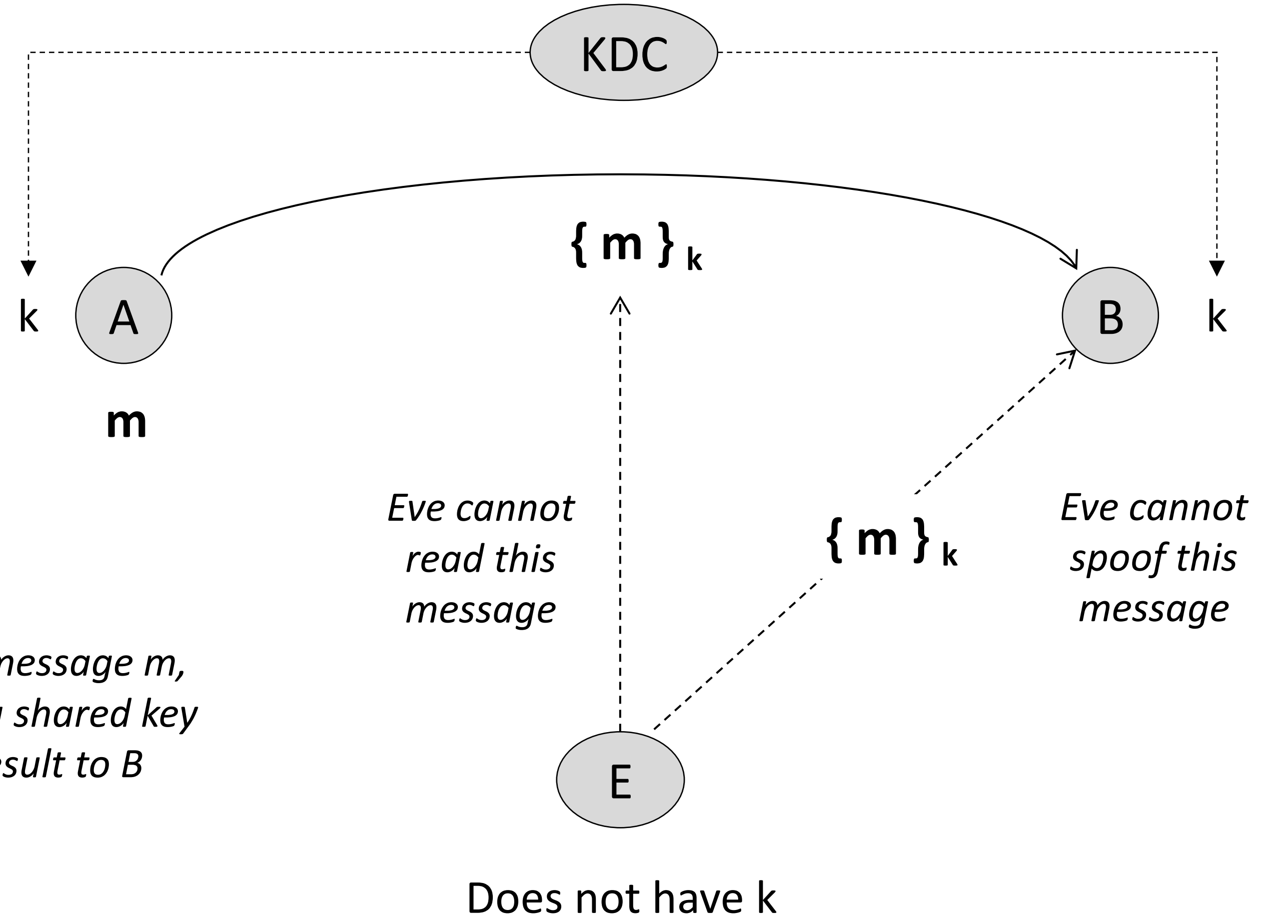
Conventional Cryptography



Alice creates message m, encrypts using shared key k and sends result to B

Week 7

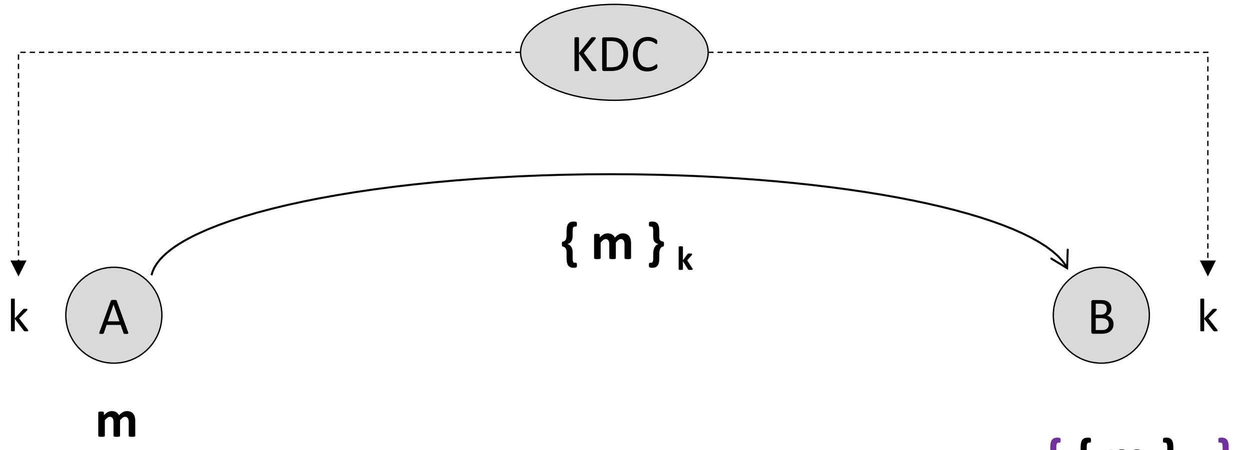
Conventional Cryptography



Alice creates message m, encrypts using shared key k and sends result to B

Week 7

Conventional Cryptography

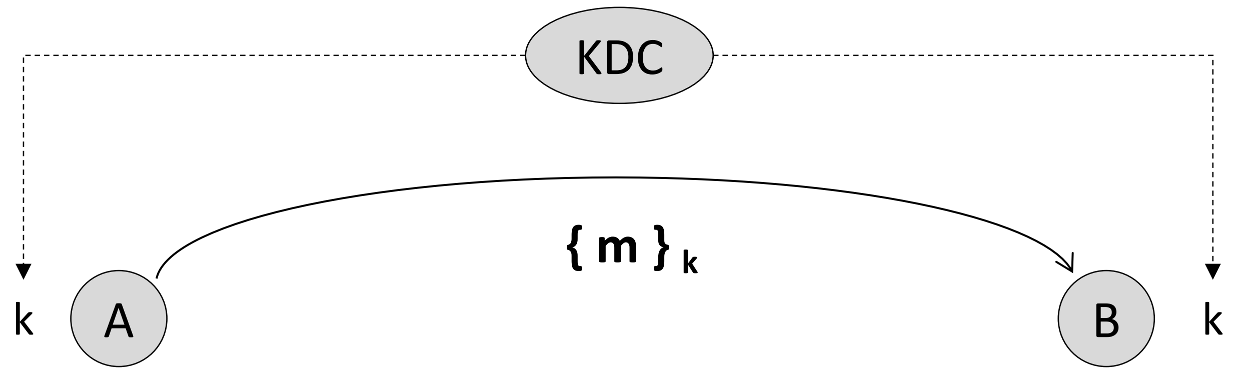


$\{\{m\}_k\}_k = m$

Bob receives encrypted message, and decrypts using shared key k and obtains message m

Week 7

Conventional Cryptography

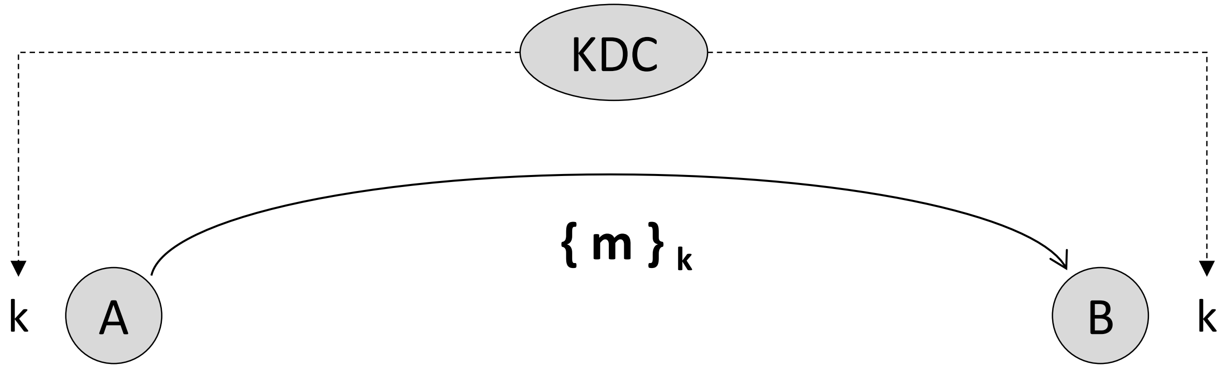


m

Secrecy Between A and B? **YES**
Authentication of A by B? **YES**

$\{\{m\}_k\}_k = m$

Conventional Cryptography



m

$\{\{ m \}_k\}_k = m$

Secrecy Between A and B? **YES**
Authentication of A by B? **YES**

Does this approach scale? **NO**

What are the Basic Properties of Public Key Cryptography?

Public Key Cryptography Basics

Two Communicants: A and B

1. A generates pair of keys PA and SA
2. B generates pair of keys PB and SB

Week 7

Public Key Cryptography Basics

Two Communicants: A and B

1. A generates pair of keys PA and SA
2. B generates pair of keys PB and SB
3. Properties:

$$\{ \{ m \}_{PA} \}_{SA} = m$$
$$\{ \{ m \}_{SA} \}_{PA} = m$$
$$\{ \{ m \}_{PA} \}_{X} = m \quad \Rightarrow \quad (X = SA)$$
$$\{ \{ m \}_{SA} \}_{X} = m \quad \Rightarrow \quad (X = PA)$$

Concept proposed by Whit Diffie and Marty Hellman, Stanford and Ralph Merkle, UC Berkeley – circa 1976

Public Key Cryptography Basics

Two Communicants: A and B

1. A generates pair of keys PA and SA
2. B generates pair of keys PB and SB
3. Properties:

$$\{ \{ m \}_{PA} \}_{SA} = m$$
$$\{ \{ m \}_{SA} \}_{PA} = m$$
$$\{ \{ m \}_{PA} \}_{X} = m \implies (X = SA)$$
$$\{ \{ m \}_{SA} \}_{X} = m \implies (X = PA)$$

Concept proposed by Whit Diffie and Marty Hellman, Stanford and Ralph Merkle, UC Berkeley – circa 1976

Requirements:

(i) Keep SA, SB secret to A, B

(ii) Make PA, PB public to all

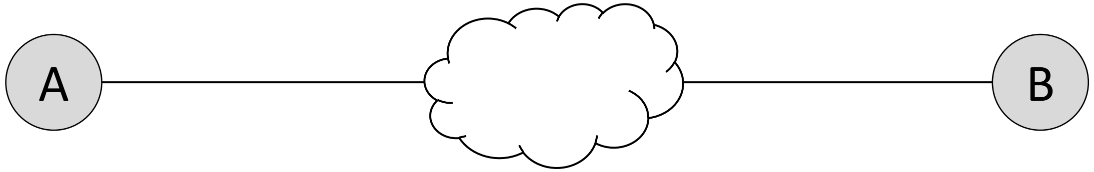
(iii) No KDC required to generate keys

*"**Address Scaling Issue**"*

Understanding Public Key Technology

"Assume A is a client"

"Assume B is a server"



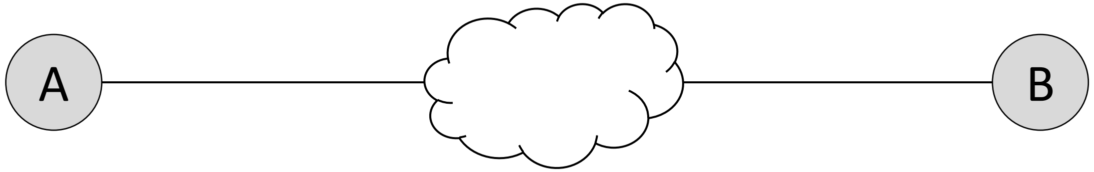
Week 7

Understanding Public Key Technology

No Key Distribution Center (KDC) Required

"Assume A is a client"

"Assume B is a server"



User A Locally Generates Key Pair:

PA: Public Key of A
SA: Secret Key of A

User B Locally Generates Key Pair:

PB: Public Key of B
SB: Secret Key of B

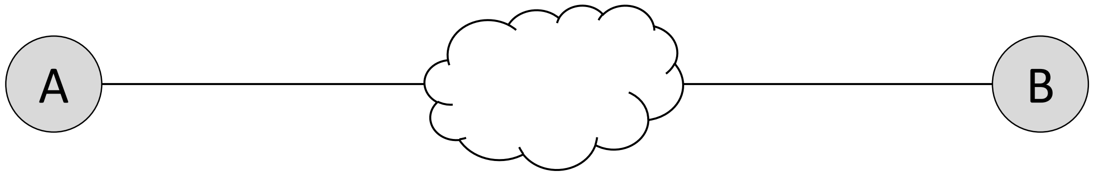
Week 7

Understanding Public Key Technology

No Key Distribution Center (KDC) Required

"Assume A is a client"

"Assume B is a server"



User A Locally Generates Key Pair:

PA: Public Key of A
SA: Secret Key of A

Common Key Generation Algorithm Required (e.g., RSA)

User B Locally Generates Key Pair:

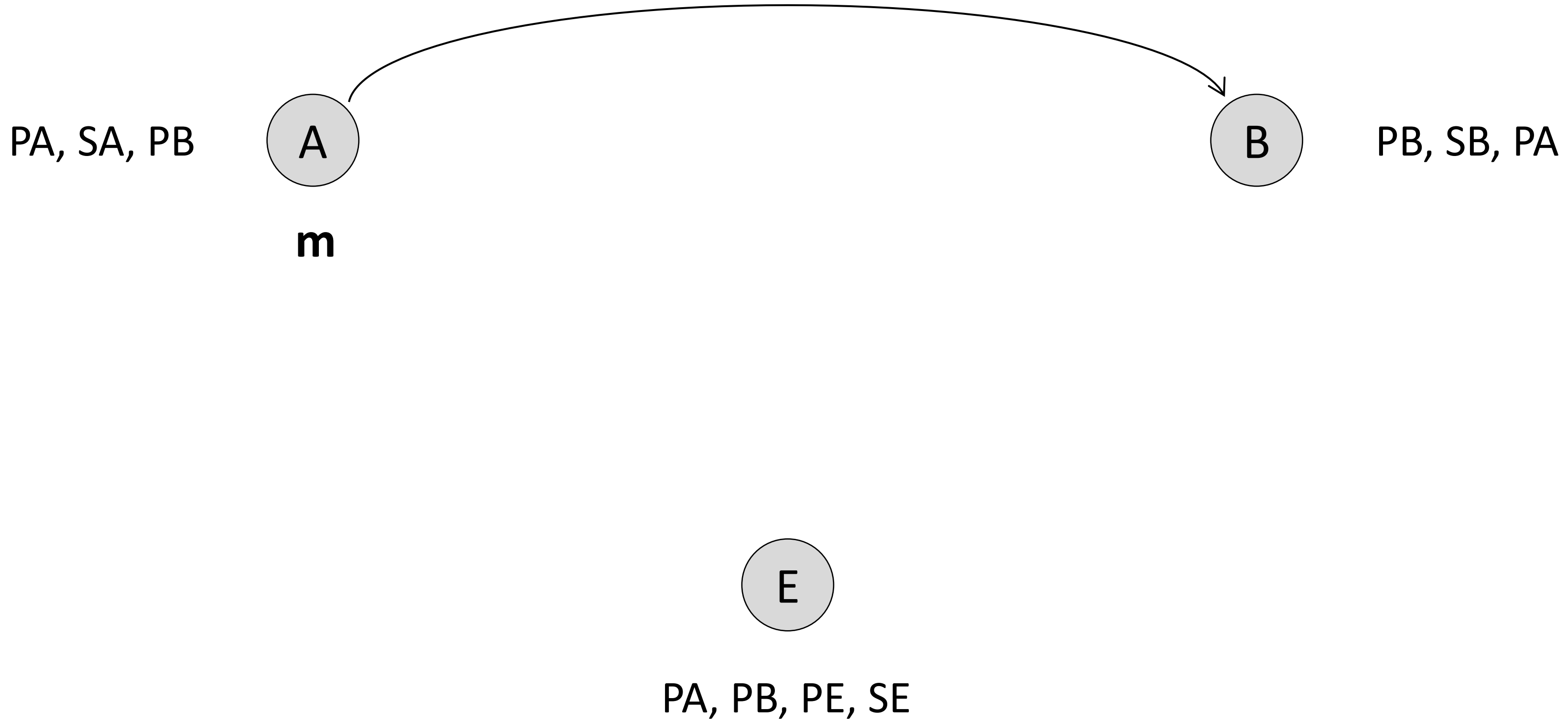
PB: Public Key of B
SB: Secret Key of B

Public Key Infrastructure (PKI)

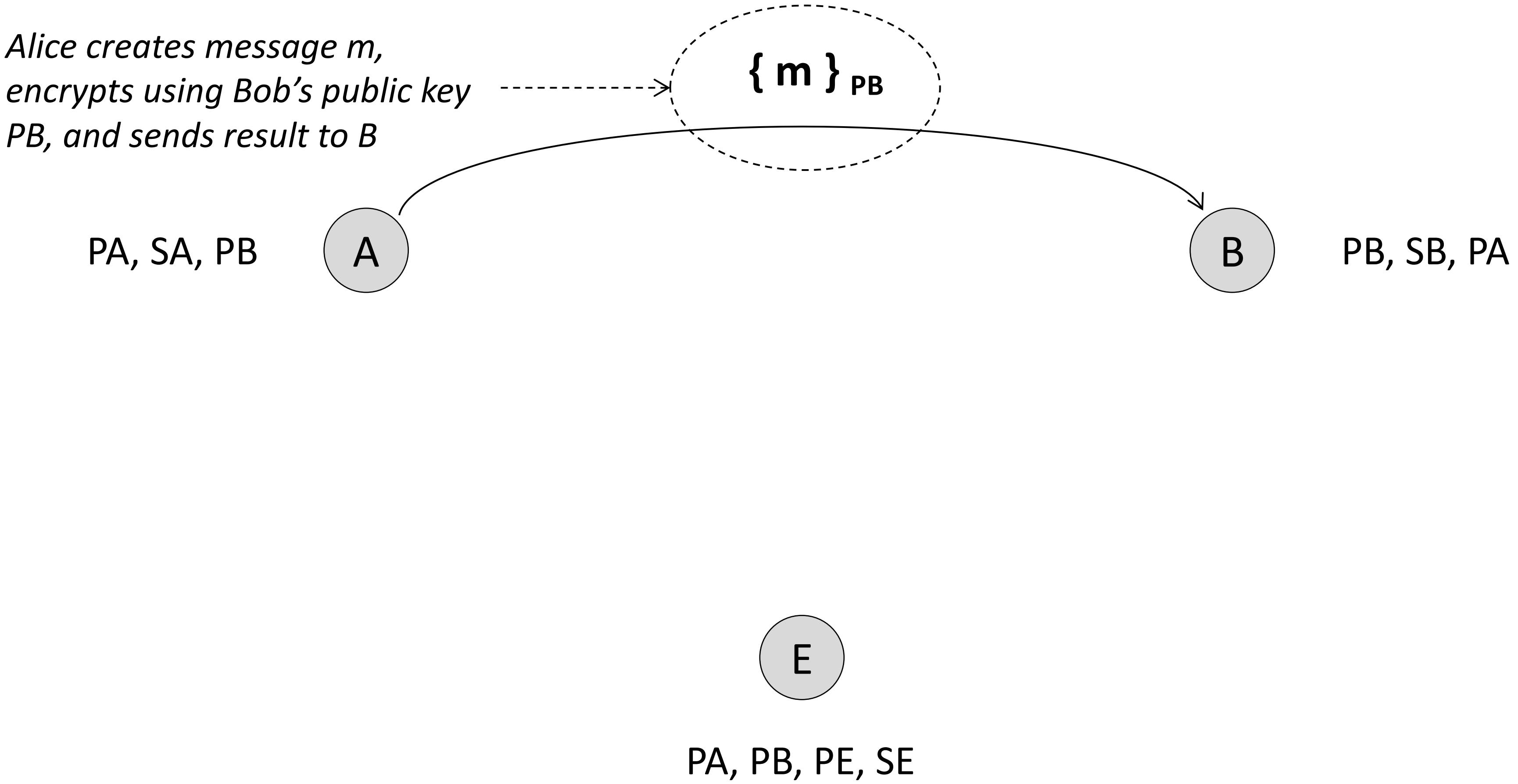
Week 7

Sending a Secret Message

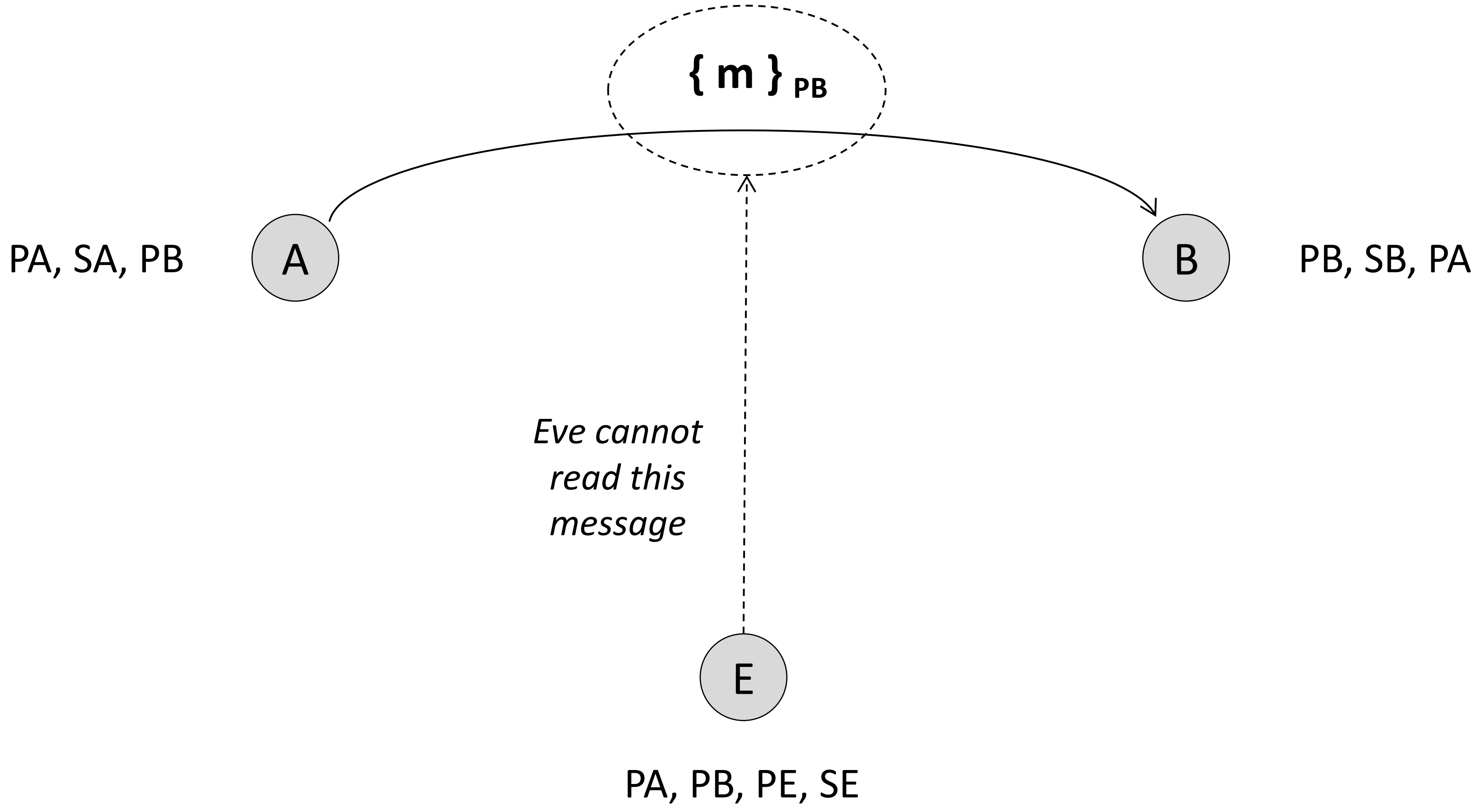
Alice creates message m . . .



Sending a Secret Message

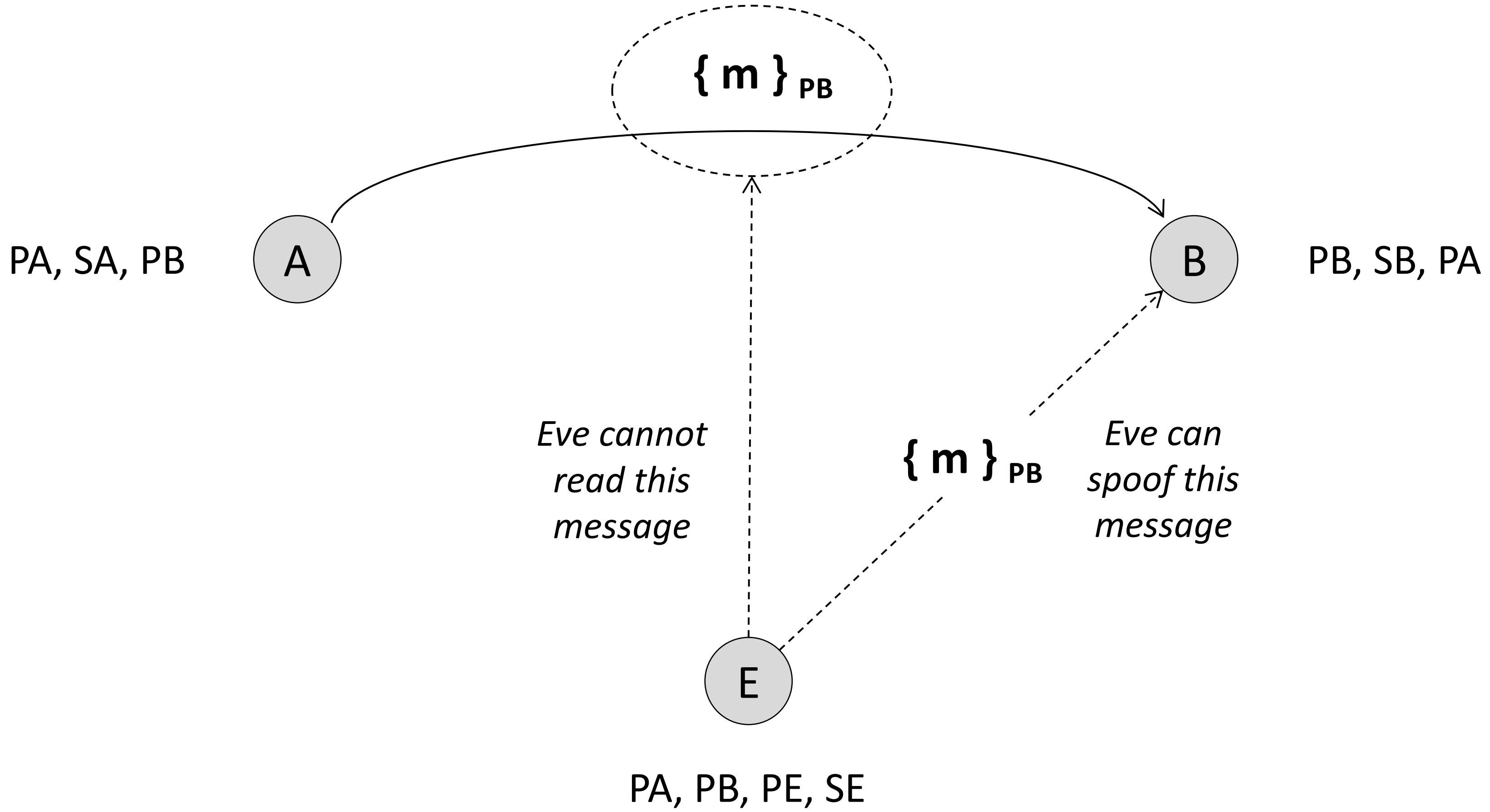


Sending a Secret Message



Week 7

Sending a Secret Message



Week 7

Sending a Secret Message

$\{ m \}_{PB}$

PA, SA, PB — A

B — PB, SB, PA

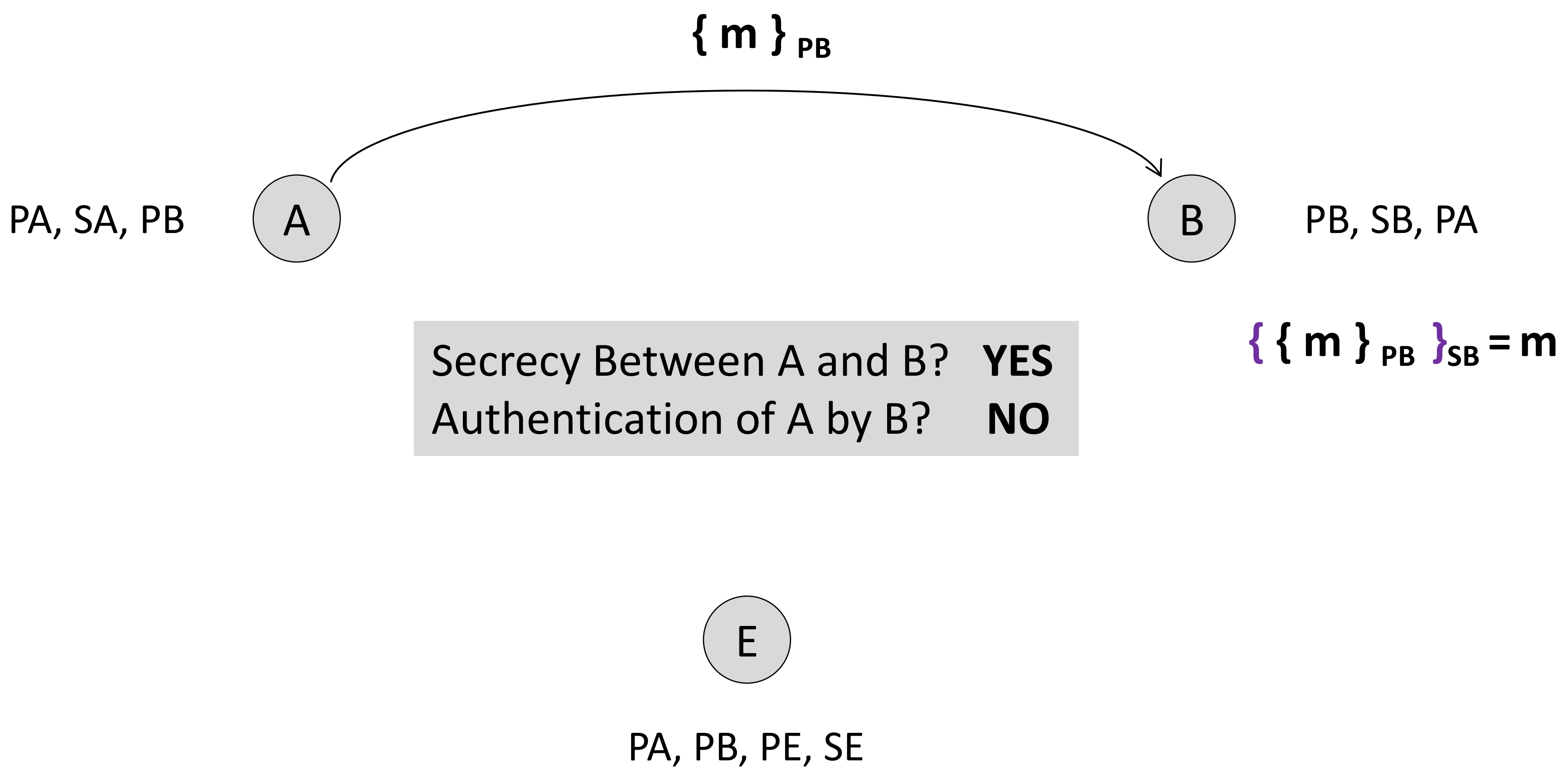
$\{ \{ m \}_{PB} \}_{SB} = m$

Bob receives the encrypted message, decrypts using Bob's secret key SB, and obtains message m

E

PA, PB, PE, SE

Sending a Secret Message

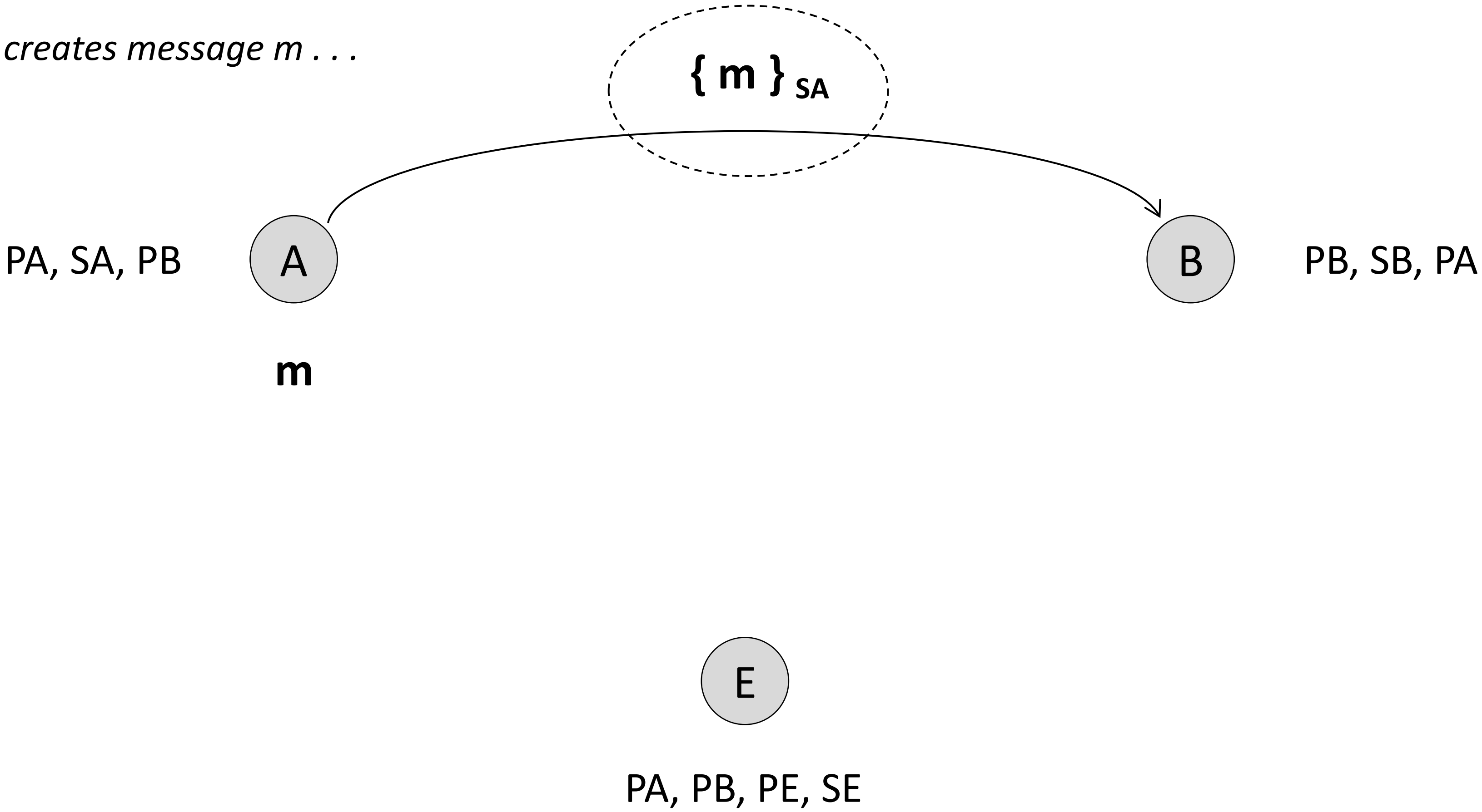


Secrecy Between A and B? **YES**
Authentication of A by B? **NO**

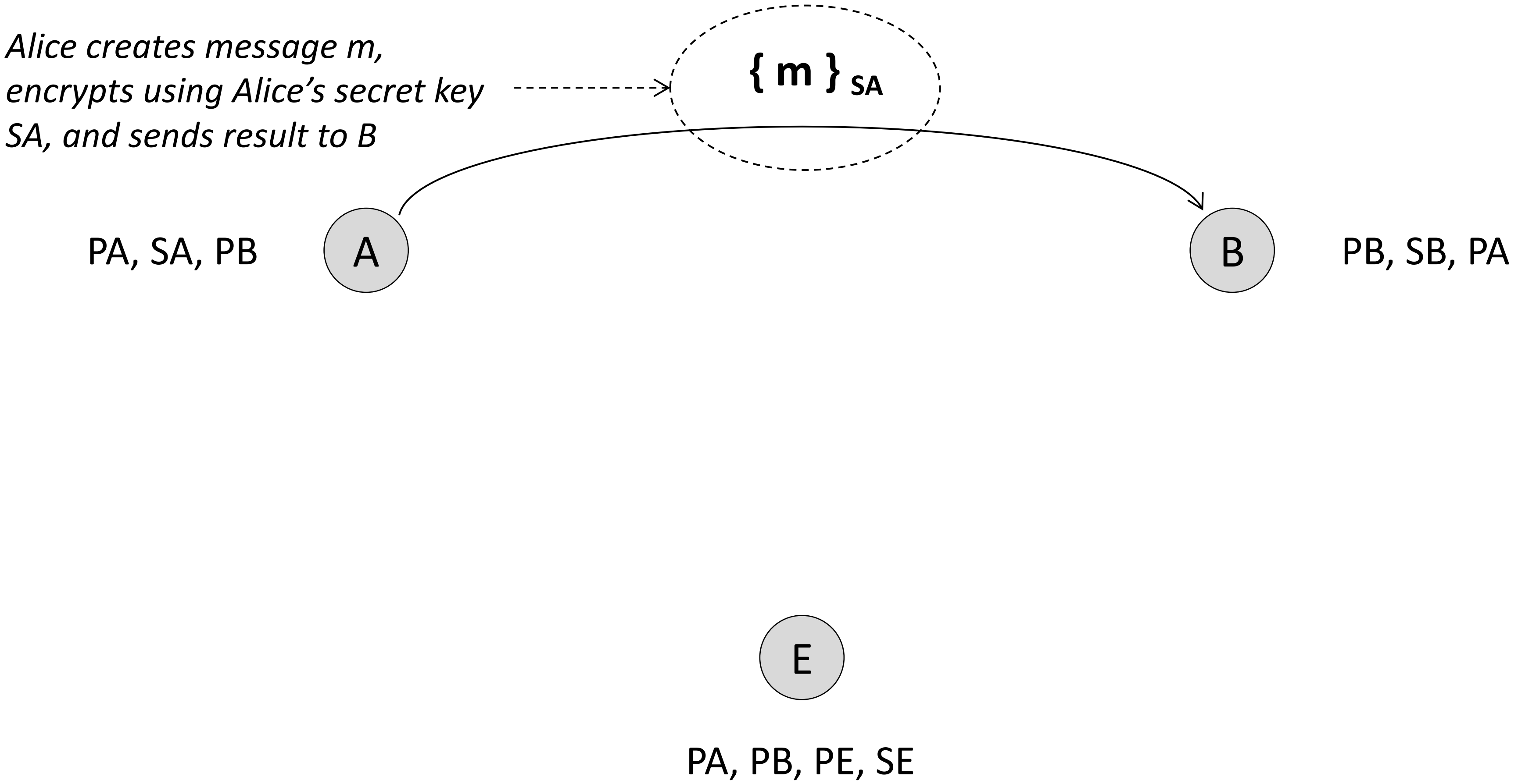
Week 7

Sending a Signed Message

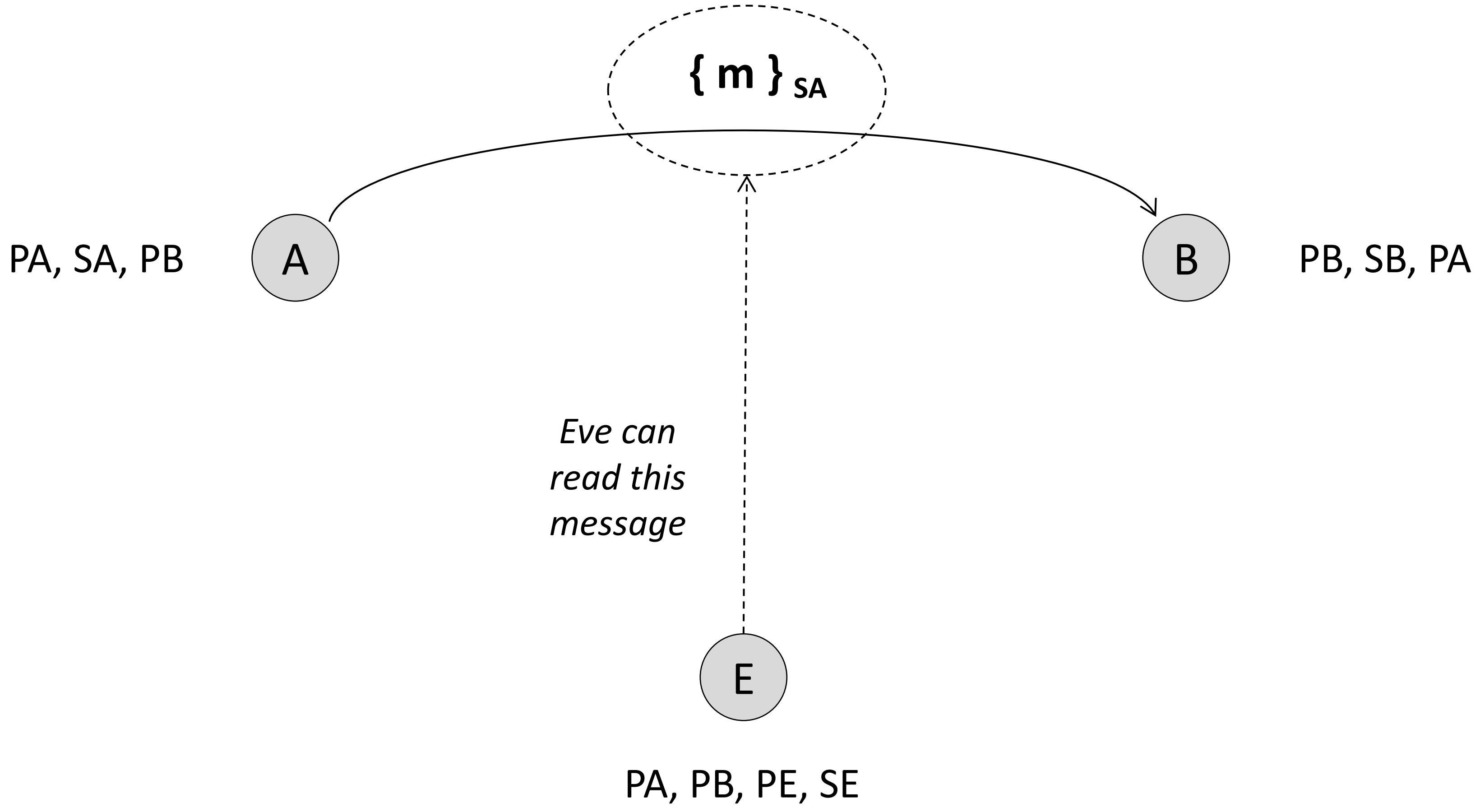
Alice creates message m . . .



Sending a Signed Message

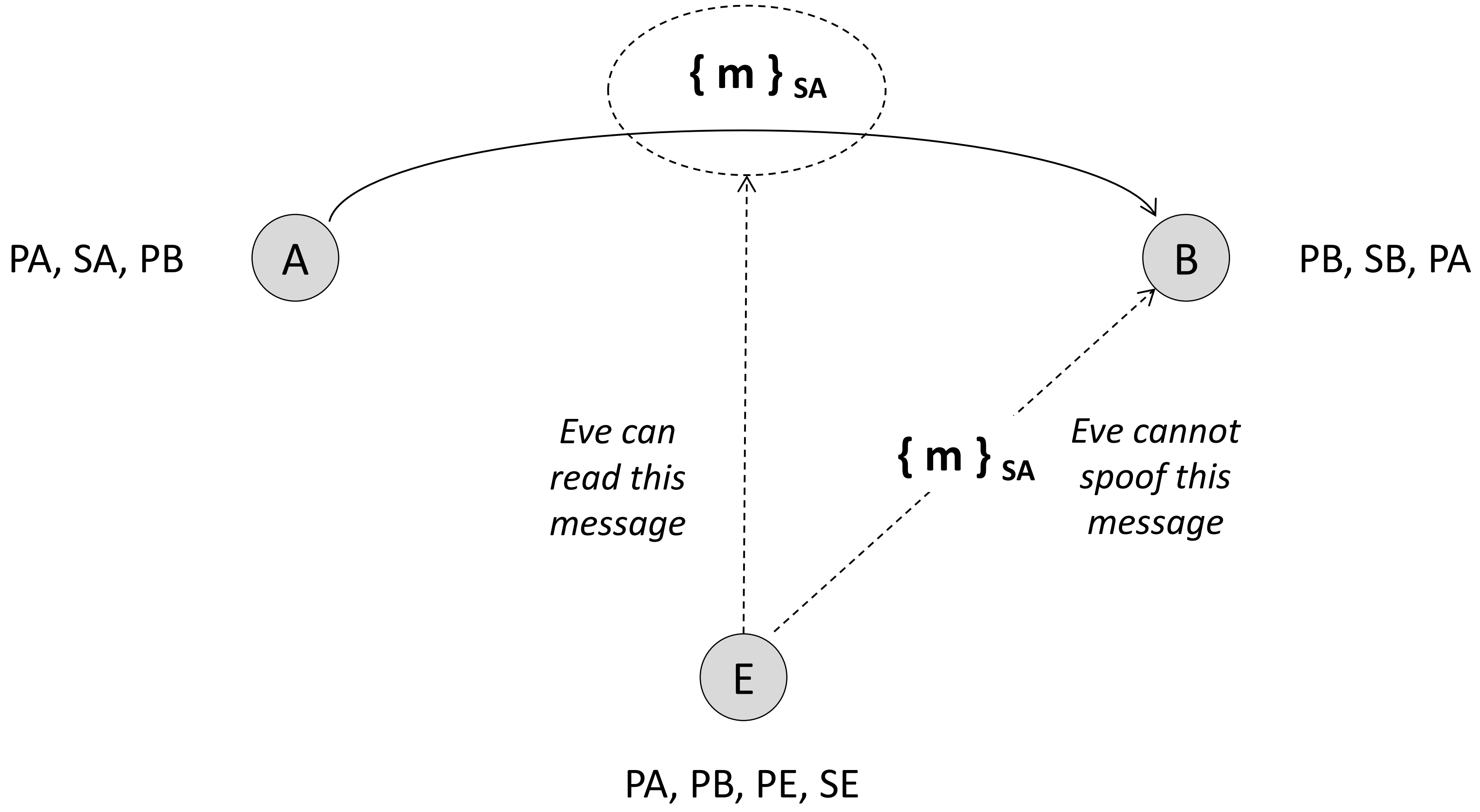


Sending a Signed Message



Week 7

Sending a Signed Message



Week 7

Sending a Signed Message

$\{ m \}_{SA}$

PA, SA, PB A

B PB, SB, PA

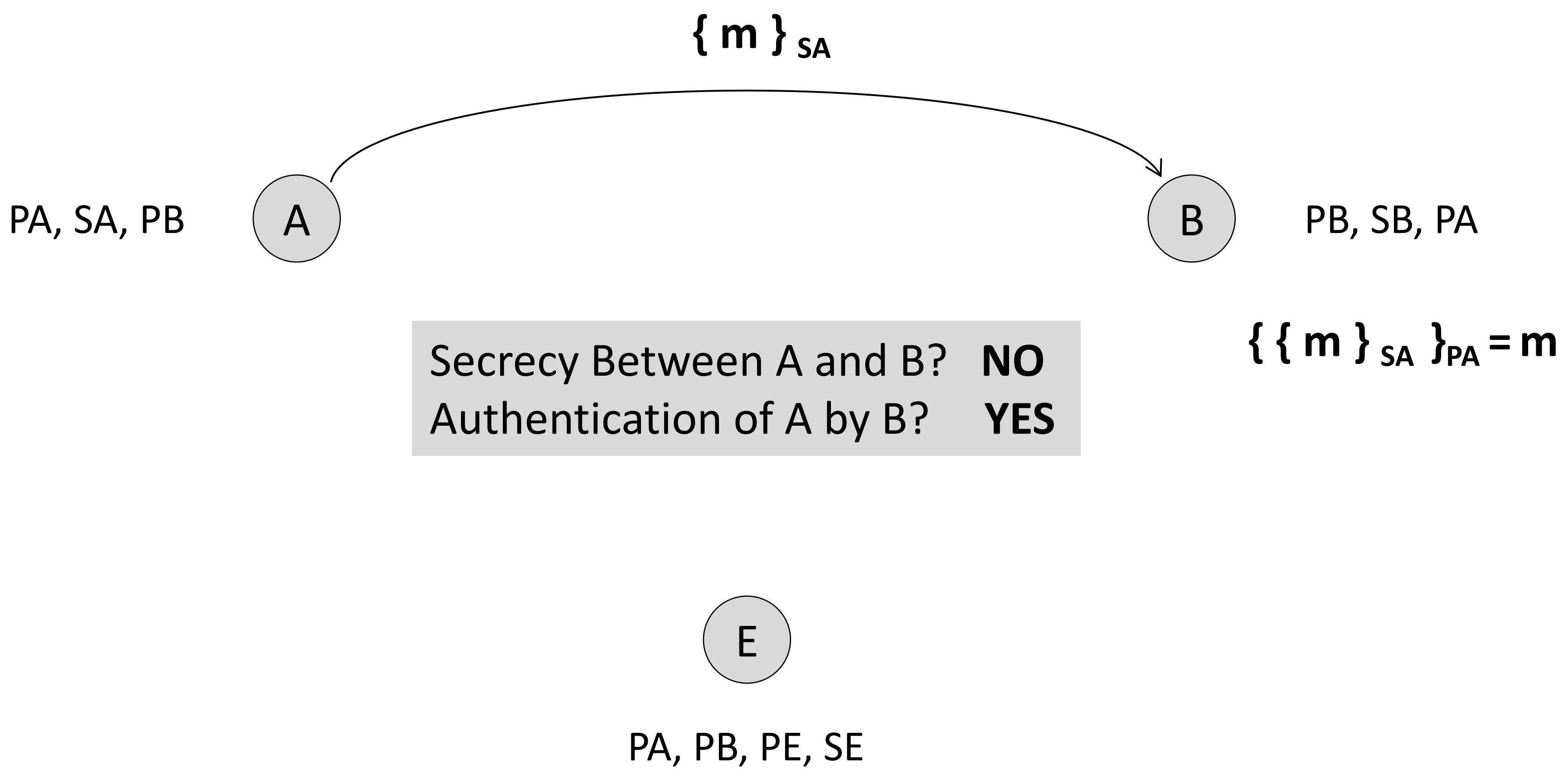
$\{ \{ m \}_{SA} \}_{PA} = m$

Bob receives the encrypted message, decrypts using Alice's public key PA, and obtains message m

E

PA, PB, PE, SE

Sending a Signed Message



Week 7

Secure Message Exchange

$\{ m \}_{SA}$

PA, SA, PB — A → B — PB, SB, PA

Alice creates a message m,
encrypts it with a public key algorithm
using her secret key SA . . .

E

PA, PB, PE, SE

Week 7

Secure Message Exchange

$\{ \{ m \}_{SA} \}_{PB}$

PA, SA, PB A

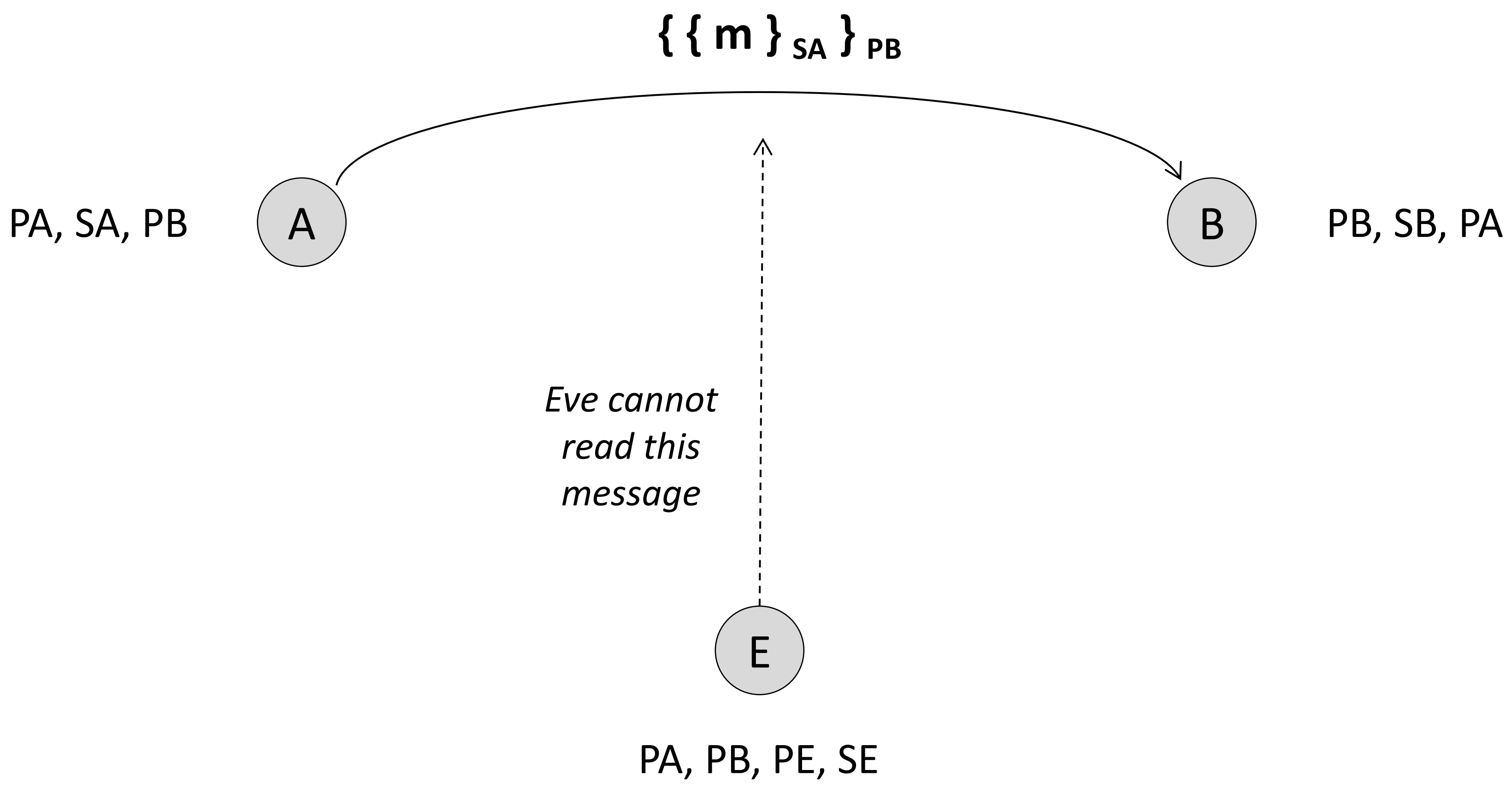
B PB, SB, PA

Alice creates a message m,
encrypts it with a public key algorithm
using her secret key SA, encrypts it again
using a public key algorithm with Bob's
public key PB, and sends the result to Bob

E

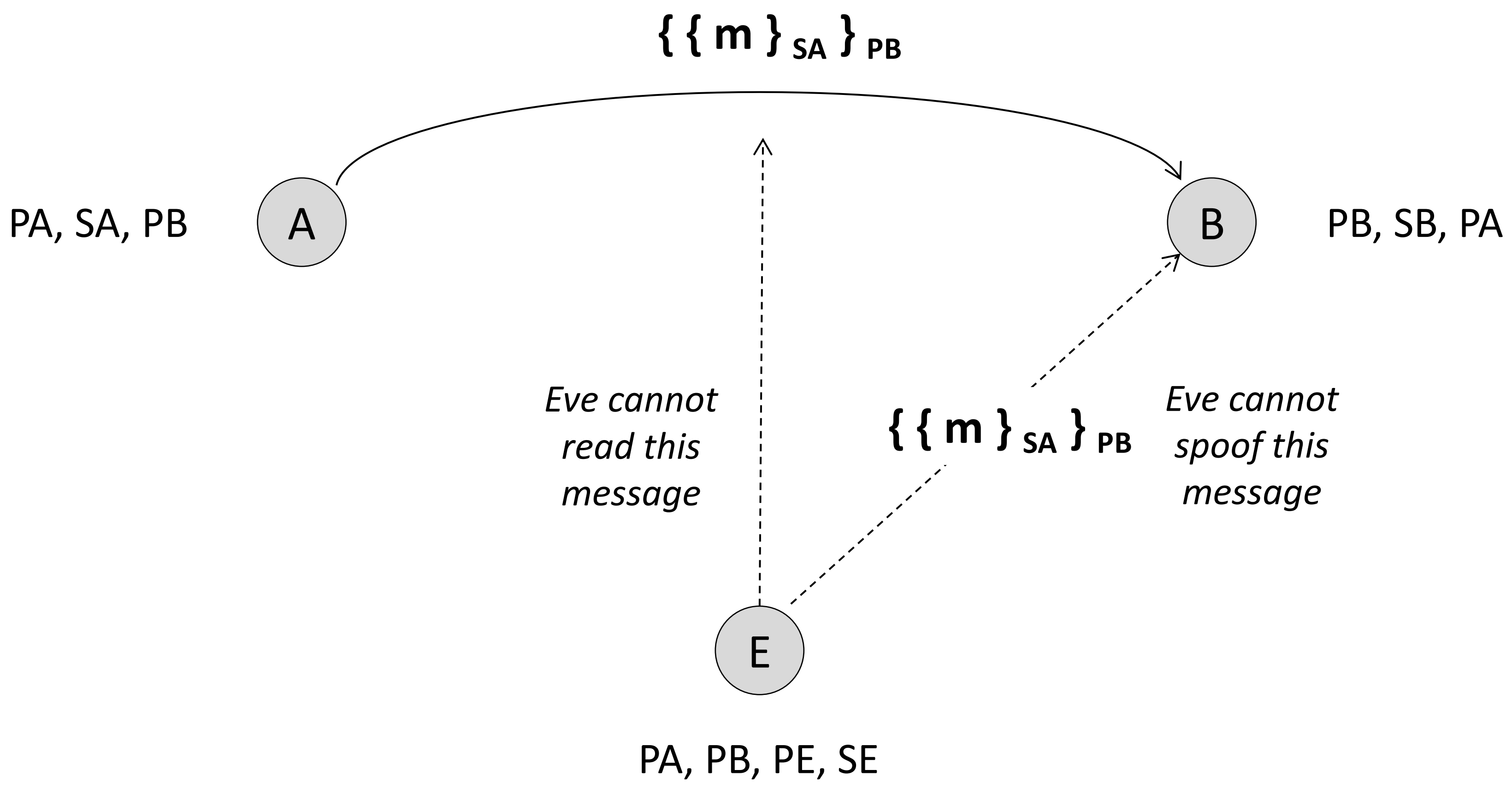
PA, PB, PE, SE

Secure Message Exchange



Week 7

Secure Message Exchange



Week 7

Secure Message Exchange

$\{ \{ m \}_{SA} \}_{PB}$

PA, SA, PB A

B PB, SB, PA

$$\{ \{ \{ \{ m \}_{SA} \}_{PB} \}_{SB} \}_{PA} = m$$

Bob receives the encrypted message, decrypts using Bob's secret key SA, then decrypts using Alice's public key PA, and obtains message m

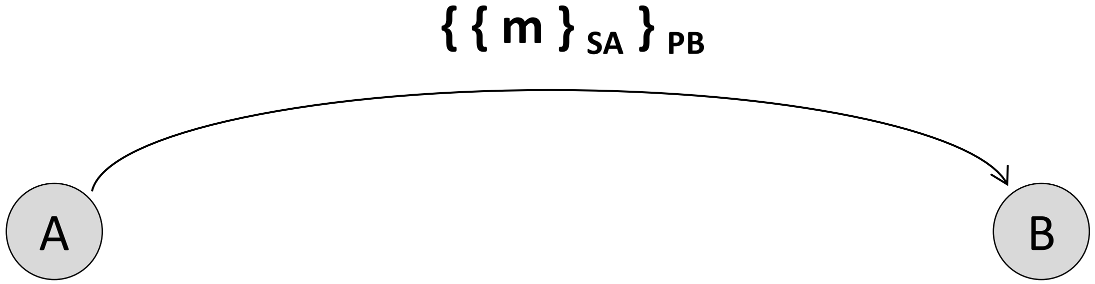
E

PA, PB, PE, SE

Week 7

Secure Message Exchange

PA, SA, PB



PB, SB, PA

$$\{\{\{\{m\}_{SA}\}_{PB}\}_{SB}\}_{PA} = m$$

Secrecy Between A and B? **YES**
Authentication of A by B? **YES**

Week 7

Secure Message Exchange

$\{\{m\}_{SA}\}_{PB}$

PA, SA, PB — A — B — PB, SB, PA

$\{\{\{\{m\}_{SA}\}_{PB}\}_{SB}\}_{PA} = m$

Secrecy Between A and B? **YES**
Authentication of A by B? **YES**

Does this approach scale? **YES**

Week 7

Secure Message Exchange

PA, SA, PB $\quad A \xrightarrow{\{\{m\}_{SA}\}_{PB}} B \quad$ PB, SB, PA

$$\{\{\{\{m\}_{SA}\}_{PB}\}_{SB}\}_{PA} = m$$

Secrecy Between A and B? **YES**
Authentication of A by B? **YES**

Does this approach scale? **YES**

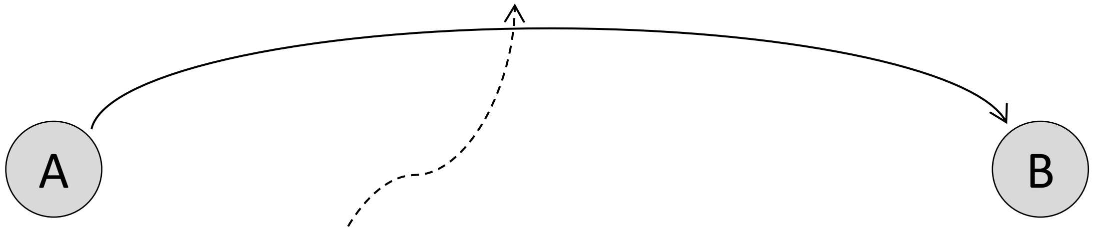
Is this approach efficient (cryptographically)? **NO**

Week 7

Secure Key Exchange

$\{ \{ k \}_{SA} \}_{PB}$

PA, SA, PB



PB, SB, PA

$\{ \{ \{ \{ k \}_{SA} \}_{PB} \}_{SB} \}_{PA} = k$

Alice generates a key k for some bulk encryption algorithm (like 3-DES) and provides this key to B using secure key exchange

- *Scalable*
- *Secret*
- *Authenticated*

Week 7

Secure Key Exchange

PA, SA, PB $\quad A \xrightarrow{\{\{k\}_{SA}\}_{PB}} B \quad$ PB, SB, PA

$$\{\{\{\{k\}_{SA}\}_{PB}\}_{SB}\}_{PA} = k$$

Secrecy Between A and B? **YES**
Authentication of A by B? **YES**

Does this approach scale? **YES**

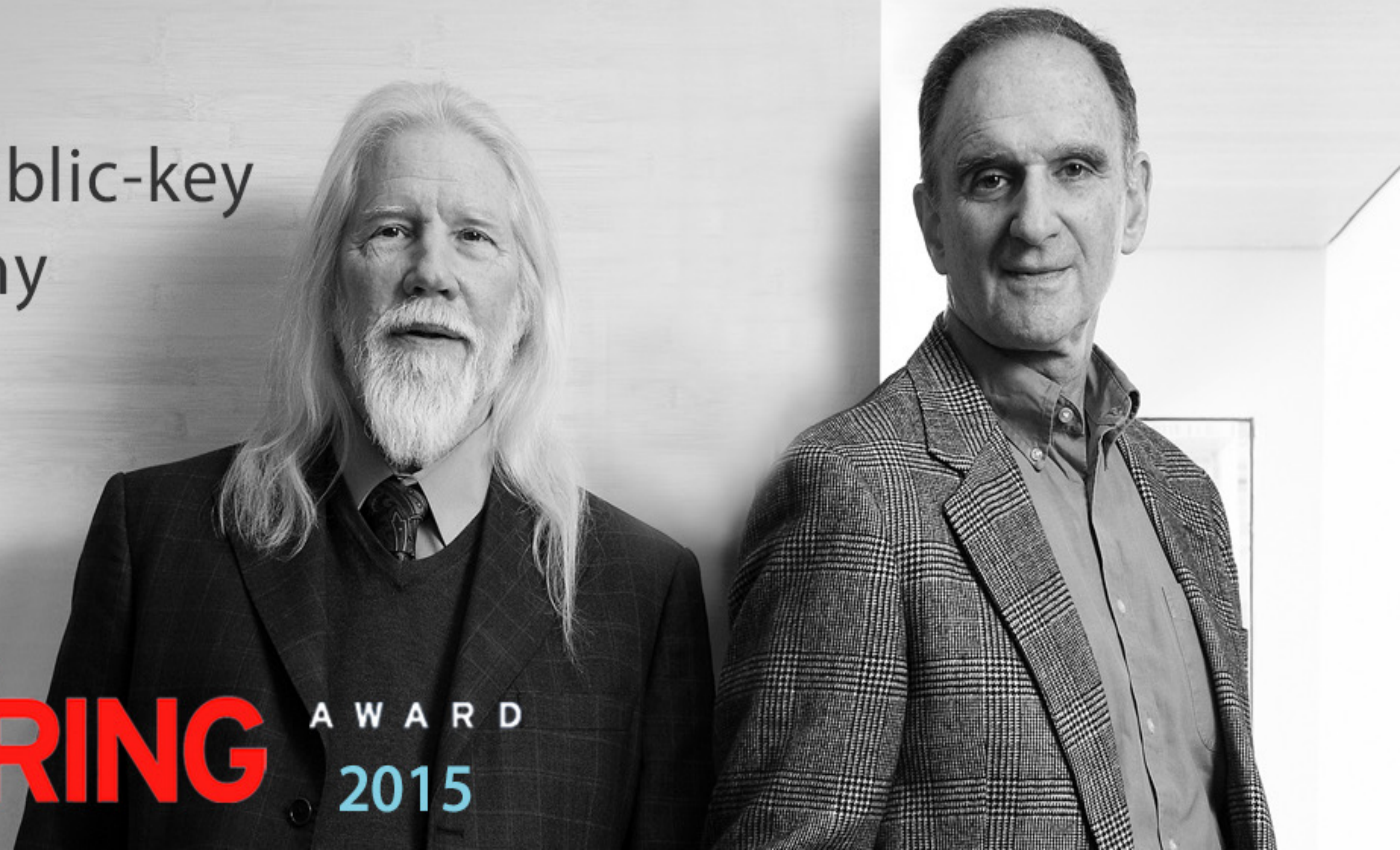
Is this approach efficient (cryptographically)? **YES**

How Does Diffie-Hellman Key Exchange Work?

Week 7

WHITFIELD DIFFIE & MARTIN HELLMAN

Invented public-key cryptography



A.M. TURING AWARD 2015

Week 7

Diffie-Hellman Key Exchange

A

B

<u>Goal</u>:

A and B share an encryption key k
with no KDC assistance

Week 7

Diffie-Hellman Key Exchange

p, g A B p, g

<u>Assume Two Publicly Known Parameters</u>:

p: Large Prime – Typically 1024 Bits

g: Primitive Element

Diffie-Hellman Key Exchange

p, g, a A

B *p, g, b*

<u>Step 1</u>:

A and B each locally generate <u>private</u> random values a and b

Diffie-Hellman Key Exchange

p, g, a
$g^a \bmod p$

A

B

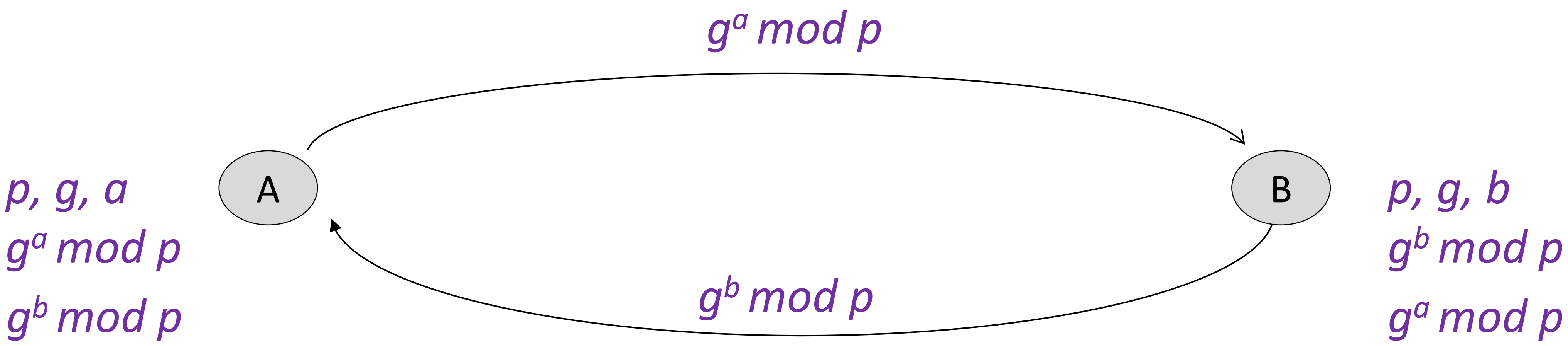
p, g, b
$g^b \bmod p$

<u>Step 2</u>:

A calculates $g^a \bmod p$
B calculates $g^b \bmod p$

Week 7

Diffie-Hellman Key Exchange

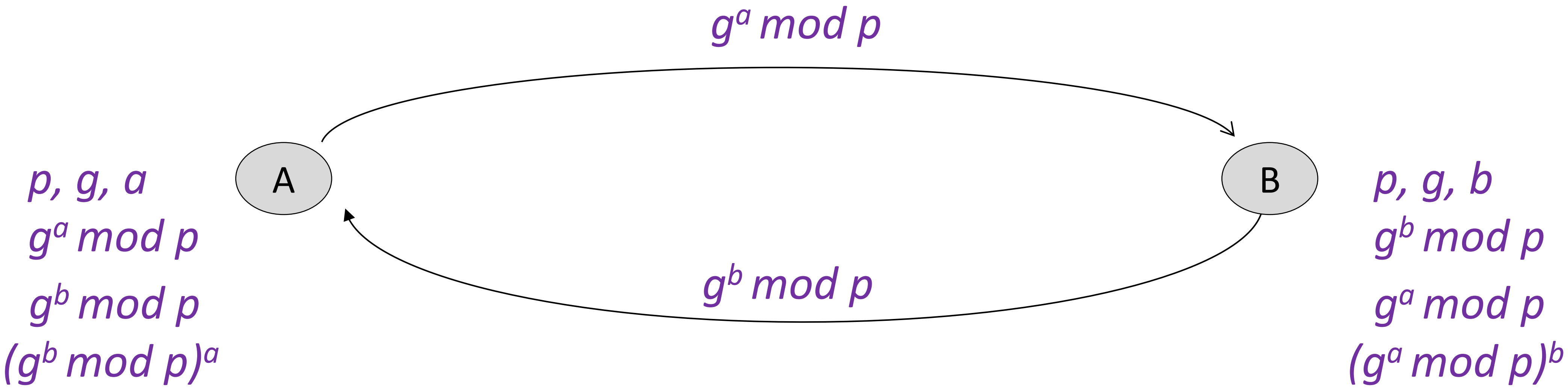


<u>Step 3</u>:

A sends g^a mod p to B
B send g^b mod p to A

Week 7

Diffie-Hellman Key Exchange

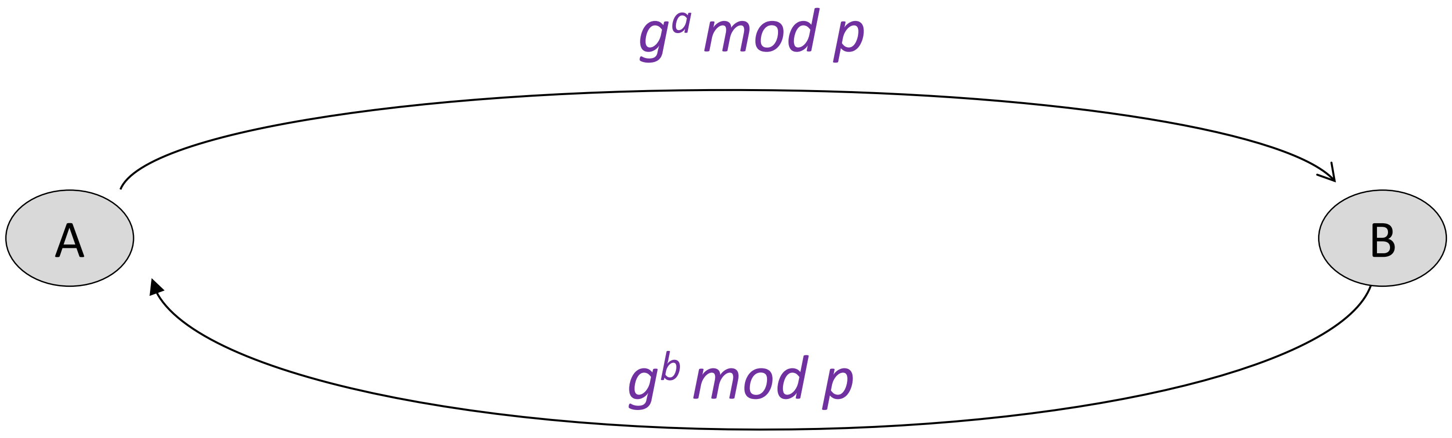


<u>Step 4</u>:

A computes $(g^a \bmod p)^b$ to B
B computes $(g^b \bmod p)^a$ to A

Week 7

Diffie-Hellman Key Exchange



p, g, a
$g^a \bmod p$
$g^b \bmod p$
$(g^b \bmod p)^a =$
$g^{ba} \bmod p$

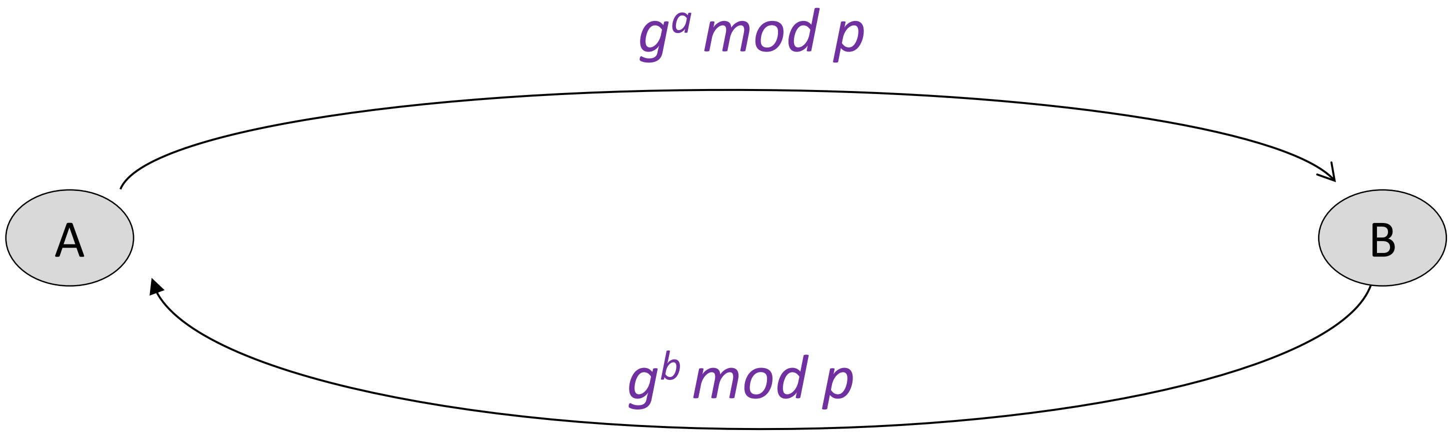
p, g, b
$g^b \bmod p$
$g^a \bmod p$
$(g^a \bmod p)^b =$
$g^{ab} \bmod p$

__Step 5__:

Shared Secret:
$g^{ab} \bmod p$

Week 7

Diffie-Hellman Key Exchange



p, g, a
$g^a \bmod p$
$g^b \bmod p$
$(g^b \bmod p)^a =$
$g^{ba} \bmod p$

p, g, b
$g^b \bmod p$
$g^a \bmod p$
$(g^a \bmod p)^b =$
$g^{ab} \bmod p =$
$g^{ba} \bmod p$

<u>Step 5</u>:

Shared Secret:
$g^{ba} \bmod p$

Week 7

Public Key Cryptography – Original Paper By Diffie and Hellman

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring

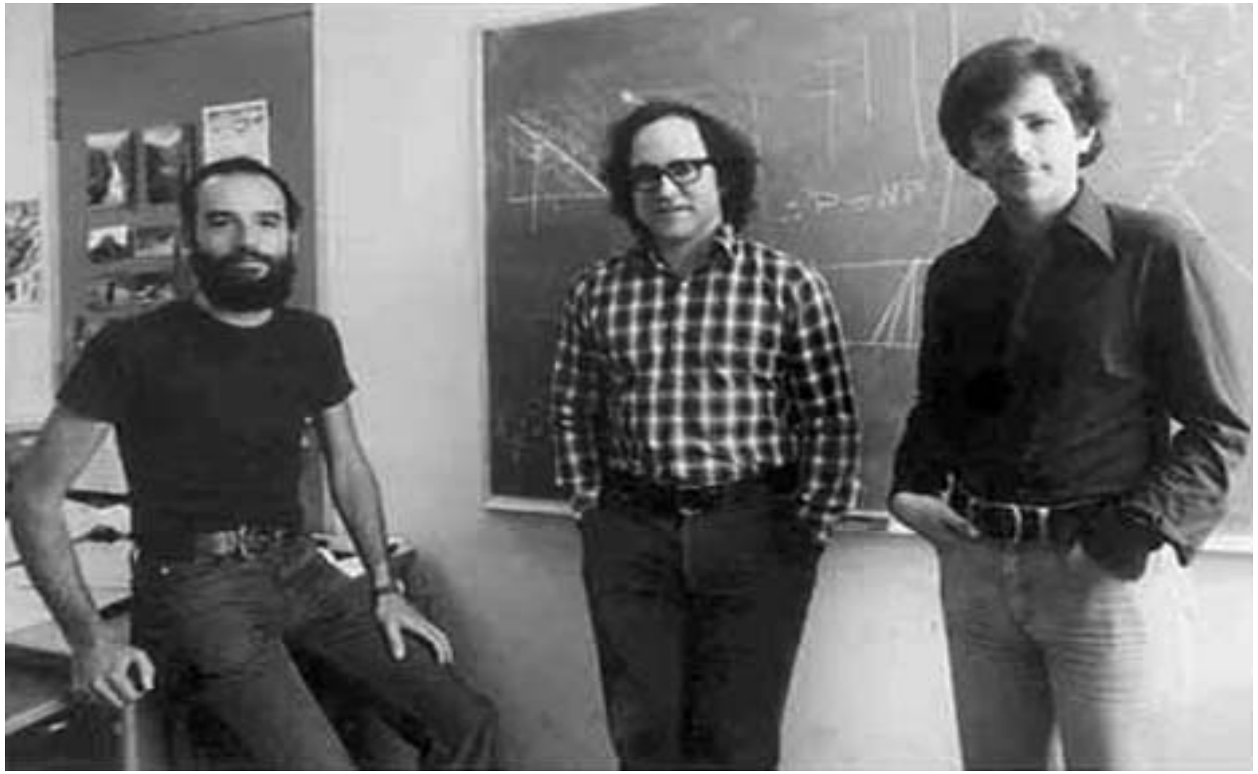
How Does the Original RSA Algorithm Work?

Week 7


RIVEST, ADI SHAMIR & LEN ADLEMAN
acm
AWARD
2002
A.M. TURING
RSA public-key cryptography

RSA Algorithm

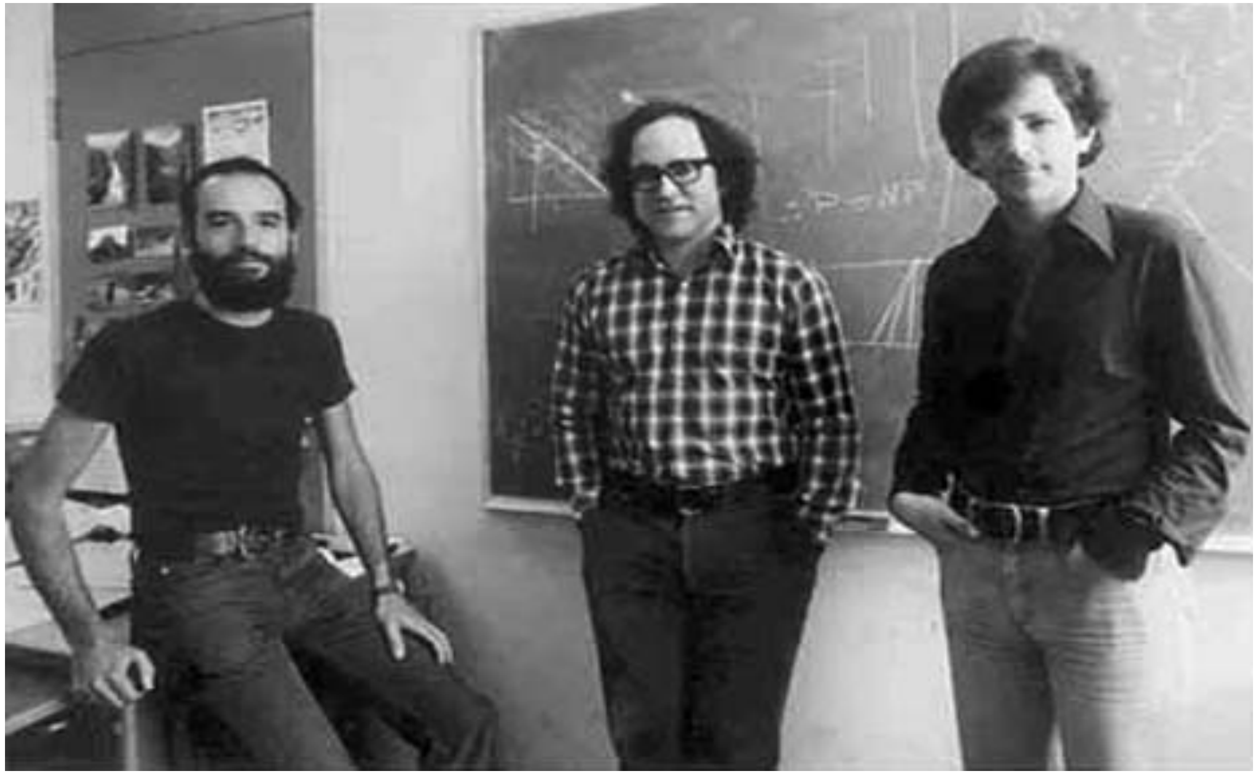
Step 1: Select two prime numbers p and q, each about 100 decimal digits in length



RSA Algorithm

Step 1: Select two prime numbers p and q, each about 100 decimal digits in length

Step 2: Calculate n = pq and
$\Psi = (p - 1)(q - 1)$

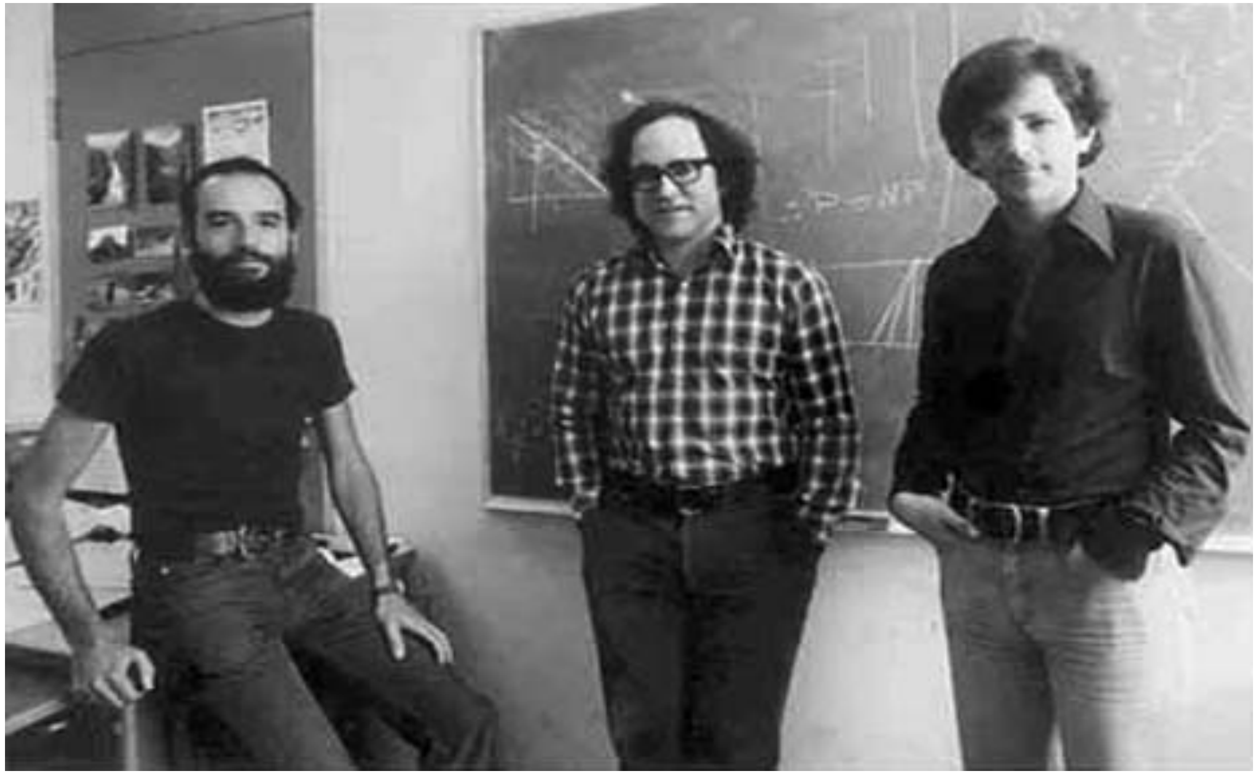


RSA Algorithm

Step 1: Select two prime numbers p and q, each about 100 decimal digits in length

Step 2: Calculate n = pq and
Ψ = (p – 1)(q – 1)

Step 3: Select integer E between 3 and Ψ, which has no common factors with Ψ



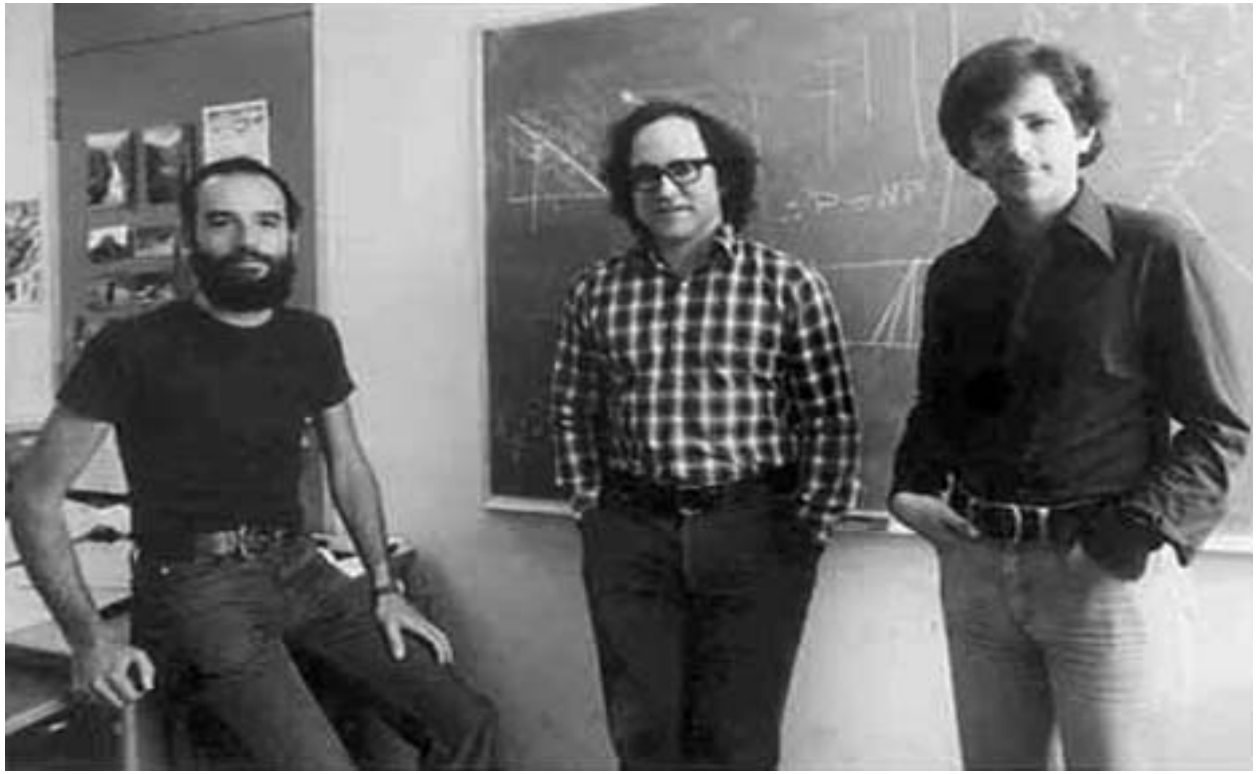
RSA Algorithm

Step 1: Select two prime numbers p and q, each about 100 decimal digits in length

Step 2: Calculate n = pq and
Ψ = (p – 1)(q – 1)

Step 3: Select integer E between 3 and Ψ, which has no common factors with Ψ

Step 4: Select integer D such that DE differs by 1 from a multiple of Ψ



RSA Algorithm

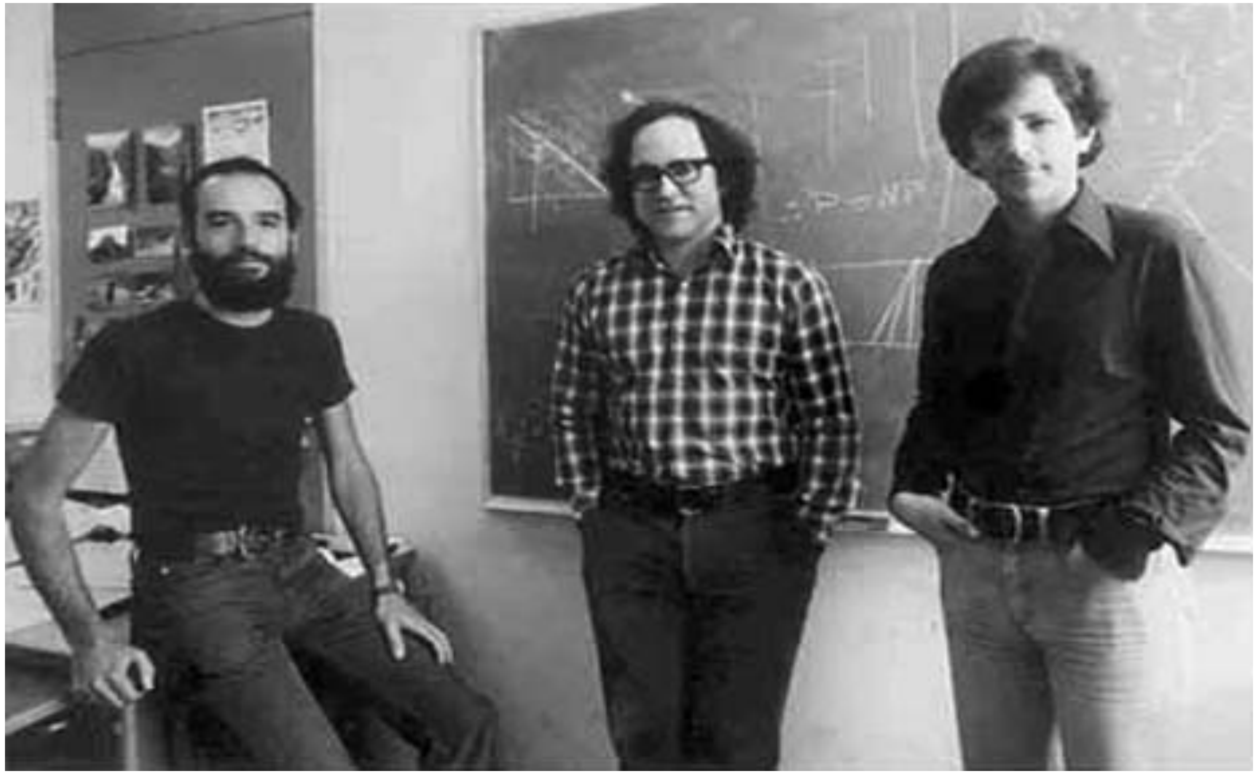
Step 1: Select two prime numbers p and q, each about 100 decimal digits in length

Step 2: Calculate n = pq and
Ψ = (p – 1)(q – 1)

Step 3: Select integer E between 3 and Ψ, which has no common factors with Ψ

Step 4: Select integer D such that DE differs by 1 from a multiple of Ψ

Step 5: Make E, n public, but
keep p, q, D and Ψ secret



RSA Algorithm

Step 1: Select two prime numbers p and q, each about 100 decimal digits in length

Step 2: Calculate $n = pq$ and $\Psi = (p - 1)(q - 1)$

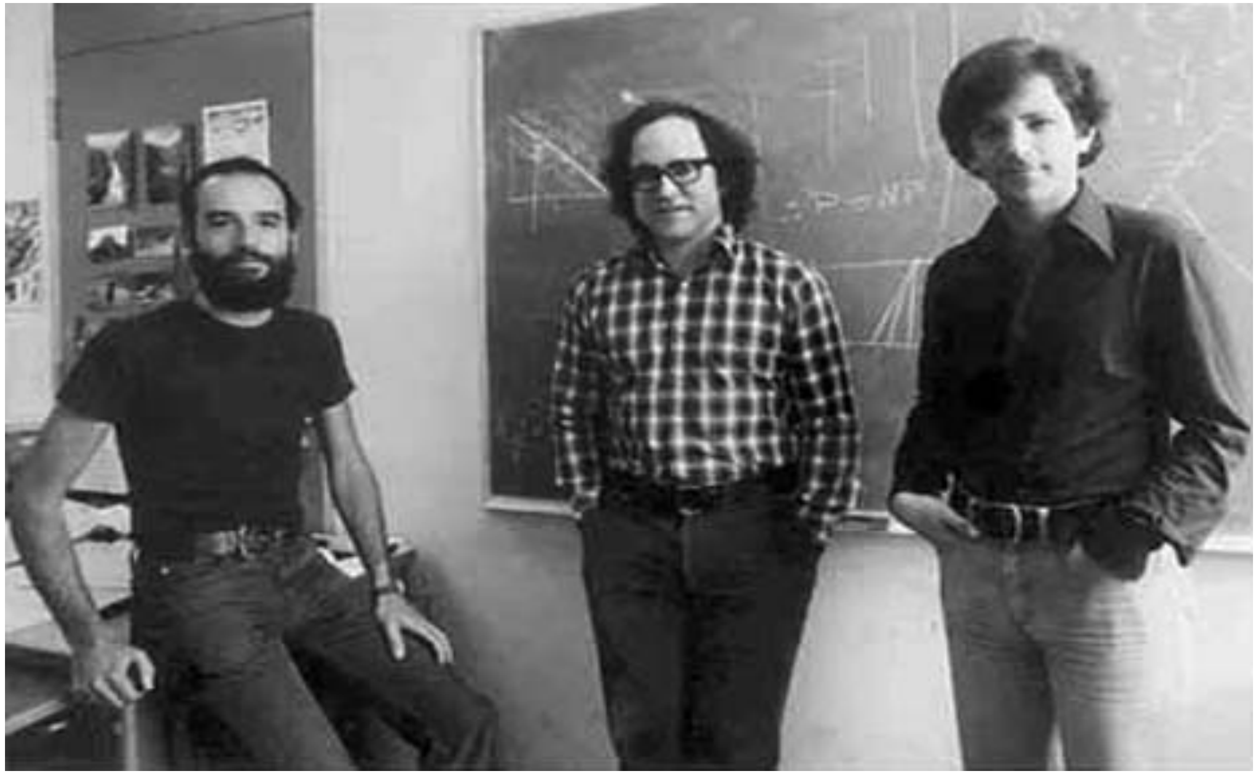
Step 3: Select integer E between 3 and Ψ, which has no common factors with Ψ

Step 4: Select integer D such that DE differs by 1 from a multiple of Ψ

Step 5: Make E, n public, but keep p, q, D and Ψ secret

Encryption: $C = P^E \bmod n$

Decryption: $P = C^D \bmod n$



Week 7

RSA Algorithm

Step 1: Select two prime numbers p and q, each about 100 decimal digits in length

Step 2: Calculate $n = pq$ and $\Psi = (p - 1)(q - 1)$

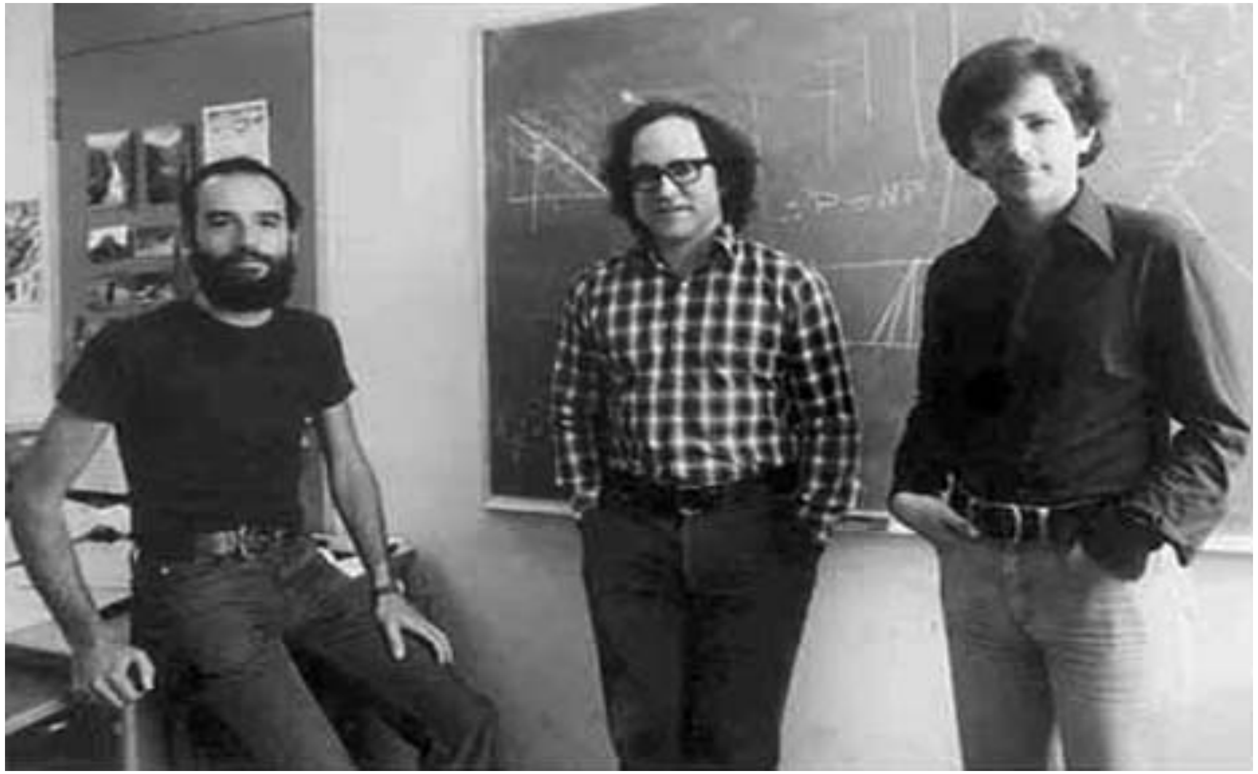
Step 3: Select integer E between 3 and Ψ, which has no common factors with Ψ

Step 4: Select integer D such that DE differs by 1 from a multiple of Ψ

Step 5: Make E, n public, but keep p, q, D and Ψ secret

Encryption: $C = P^E \bmod n$

Decryption: $P = C^D \bmod n$



Example: $p = 3$, $q = 5$, $n = 15$, $\Psi = 8$ Select $E = 5$, $D = 5$

Encrypt “2”: $2^5 \bmod 15 = 2$

Decrypt “2”: $2^5 \bmod 15 = 2$

Who <u>Really</u> Invented Public Key Technology?
(Hint: UK)

Week 7

James Ellis, Engineer at GCHQ – Circa 1969

Week 7

Bell Labs – Project C43 (1944)

SECRET 054 450

ATI- 29345

DIVISION (None)

ORIG. AGENCY NO. (None)

PUBLISHING AGENCY NO. 4573A

U

TITLE: Final Report - Part I - Speech Privacy Systems - Interception, Diagnosis, Decoding, Evaluation

AUTHOR(S): Koenig, W.

ORIGINATING AGENCY: Bell Telephone Labs., Inc., New York, N. Y.

PUBLISHED BY: Office of Scientific Research and Development, NDRC, Div. 13

DATE	DOC. CLASS.	COUNTRY	LANGUAGE	PAGES	ILLUSTRATIONS
Oct '44	Secr.	U.S.	Eng.	111	photos, tables, diagrs

ABSTRACT:

The results of three years' experience in diagnosing, decoding, and evaluating speech privacy systems are summarized. Speech privacy systems may be used in connection with radio telephone systems or wire systems, but radio interception problems only are discussed. The decoding techniques described apply to wire as well as to radio communications. The sound spectrograph is described including its history, method of operation, and capabilities. It analyzes speech in terms of its three basic dimensions, frequency, amplitude, and time; and portrays the analysis in the form of spectrograms. Basic speech scrambling methods are also explained in which the original speech is transmitted with its parts modified, displaced, or interchanged. Cryptanalysis and cryptography, which apply to telegraph types of communication, are also described.

UTIS SOP memo 7 Aug 60

DISTRIBUTION: ~~Copies of this report obtainable from Air Documents Division; Attn: MCIDXD~~

DIVISION: Electronics ~~(8)~~ 5

SECTION: Communications (1) 6

SUBJECT HEADINGS: Communication systems, Secret (23992.87); Decoders (28877)

AD-A800 206

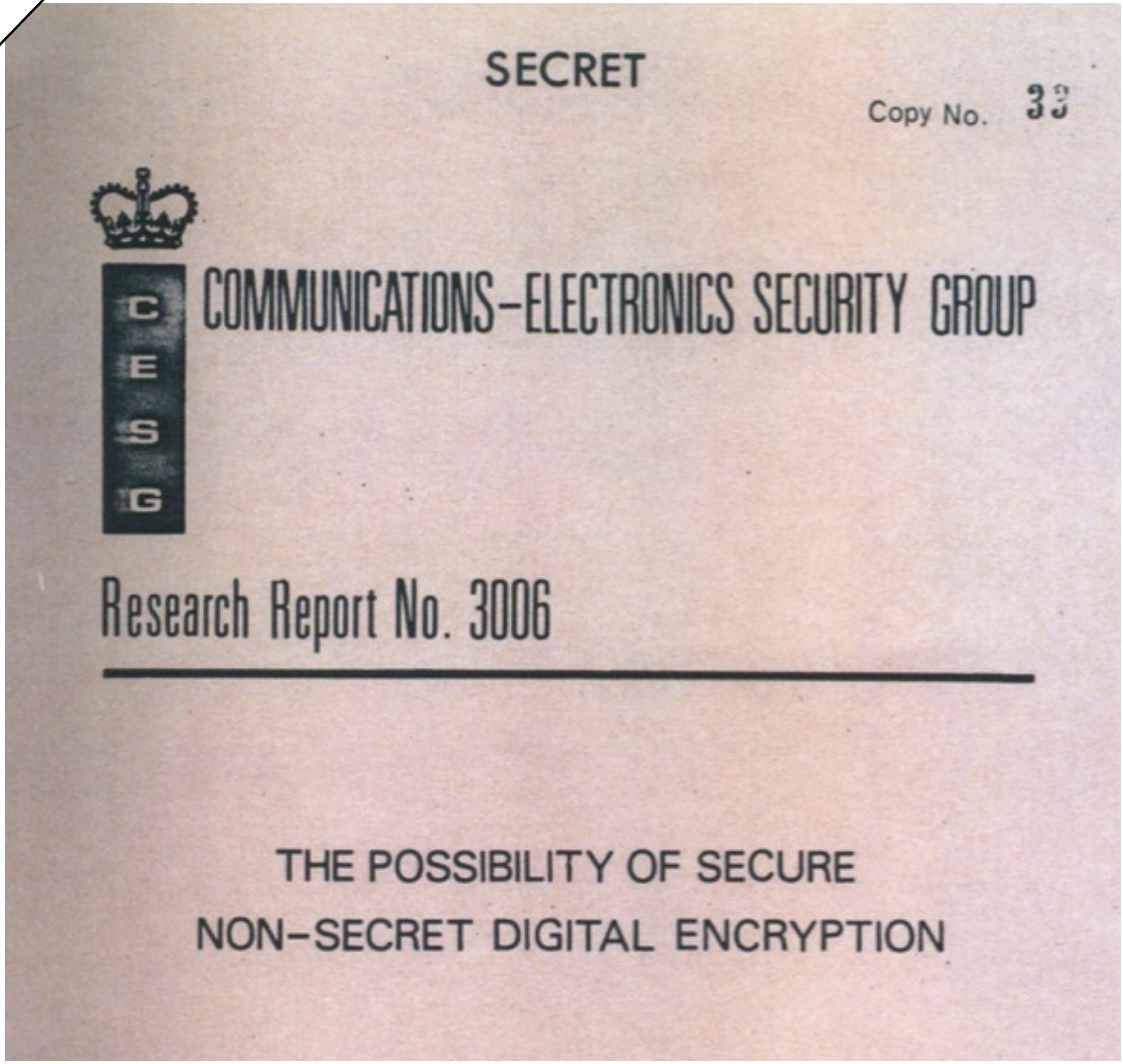
...CAL INDEX

Wright-Patterson Air Force Base
Dayton, Ohio

SECRET

GCHQ – Original and New Headquarters in Cheltenham, UK

James Ellis' Paper 1970 – Classified for Three Decades


SECRET

Copy No. 33

COMMUNICATIONS–ELECTRONICS SECURITY GROUP

C E S G

Research Report No. 3006

THE POSSIBILITY OF SECURE NON–SECRET DIGITAL ENCRYPTION

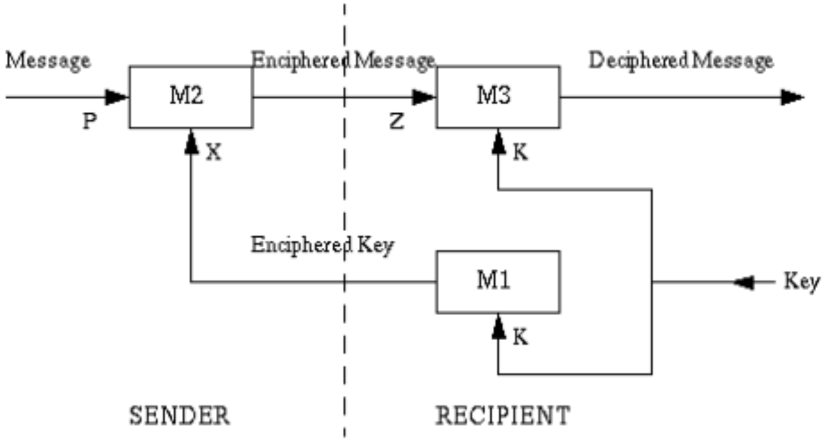


Fig. 1

13. The following properties are clearly essential. It must be impossible for the interceptor to obtain p from z without knowing k even though he knows x. Also, since a knowledge of k would enable him to decipher z, he must be unable to obtain k from x. Finally M3 must have the property of being able to decipher z. To obtain these properties we specify the look-up tables corresponding to MI, M2 and M3 in the following way: -

   a. Let k have n different possible values and p have m different possible values, for simplicity take them to be the integers 1 to n and 1 to m respectively. Let x have the same range of values as k, and z have the same range as p.

   b. MI can be defined as a linear look-up table of n entries whose contents are the numbers 1 to n in a random order, where "random" implies that the output is sufficiently uncorrelated with the input so that the position of a particular entry in the table cannot be found in a simpler way than by searching through the table.

   c. M2 corresponds to an n by m rectangular table in which the entries for a fixed value of x consists of the numbers 1 to m in random order, and where the columns for the various values of x are suitable uncorrelated with one another.

Clifford Cocks and Malcolm Williamson

SECRET

- 1 -

<u>Note on "Non-Secret Encryption"</u>

In [1] J H Ellis describes a theoretical method of encryption which does not necessitate the sharing of secret information between the sender and receiver. The following describes a possible implementation of this.

a. The receiver picks 2 primes P, Q satisfying the conditions

i. P does not divide Q-1.

ii. Q does not divide P-1.

He then transmits N = PQ to the sender.

b. The sender has a message, consisting of numbers

$C_1, C_2, \dots C_r$ with $0 < C_i < N$

He sends each, encoded as D_i where

$D_i = C_i^N$ reduced modulo N.

c. To decode, the receiver finds, by Euclids Algorithm, numbers P′, Q′

satisfying $P P' \equiv 1 \pmod{Q-1}$

$Q Q' \equiv 1 \pmod{P-1}$

Then $C_i \equiv D_i^{P'} \pmod{Q}$

and $C_i \equiv D_i^{Q'} \pmod{P}$

Week 7

Credit Where Credit is Due