An Introduction to Cyber Security – CS 573

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

Week 8

<u>Required Week Eight Readings</u>

1. "Blind Signatures for Untraceable Payments," David Chaum
https://sceweb.sce.uhcl.edu/yang/teaching/csci5234WebSecurityFall2011/Chaum-blind-signatures.PDF

***2.* Finish *From CIA to APT: An Introduction to Cyber Security*, E. Amoroso & M. Amoroso**

Twitter: @hashtag_cyber
LinkedIn: Edward Amoroso

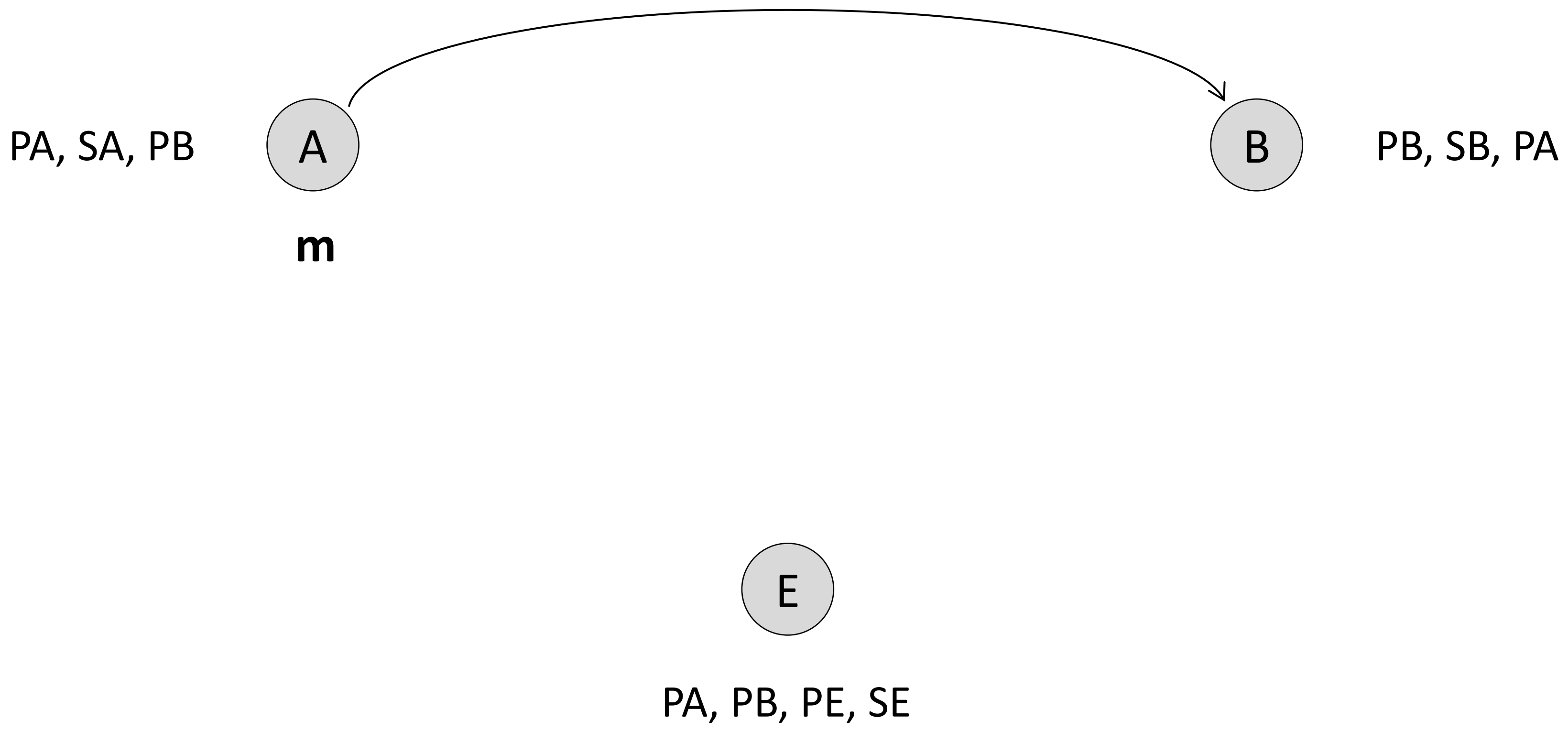
Week 8: Key Distribution, Digital Signing, SSL, and Secure eCommerce

What are the Basic Properties of
Public Key Cryptography?
(recap from last week’s Zoom)

Week 8

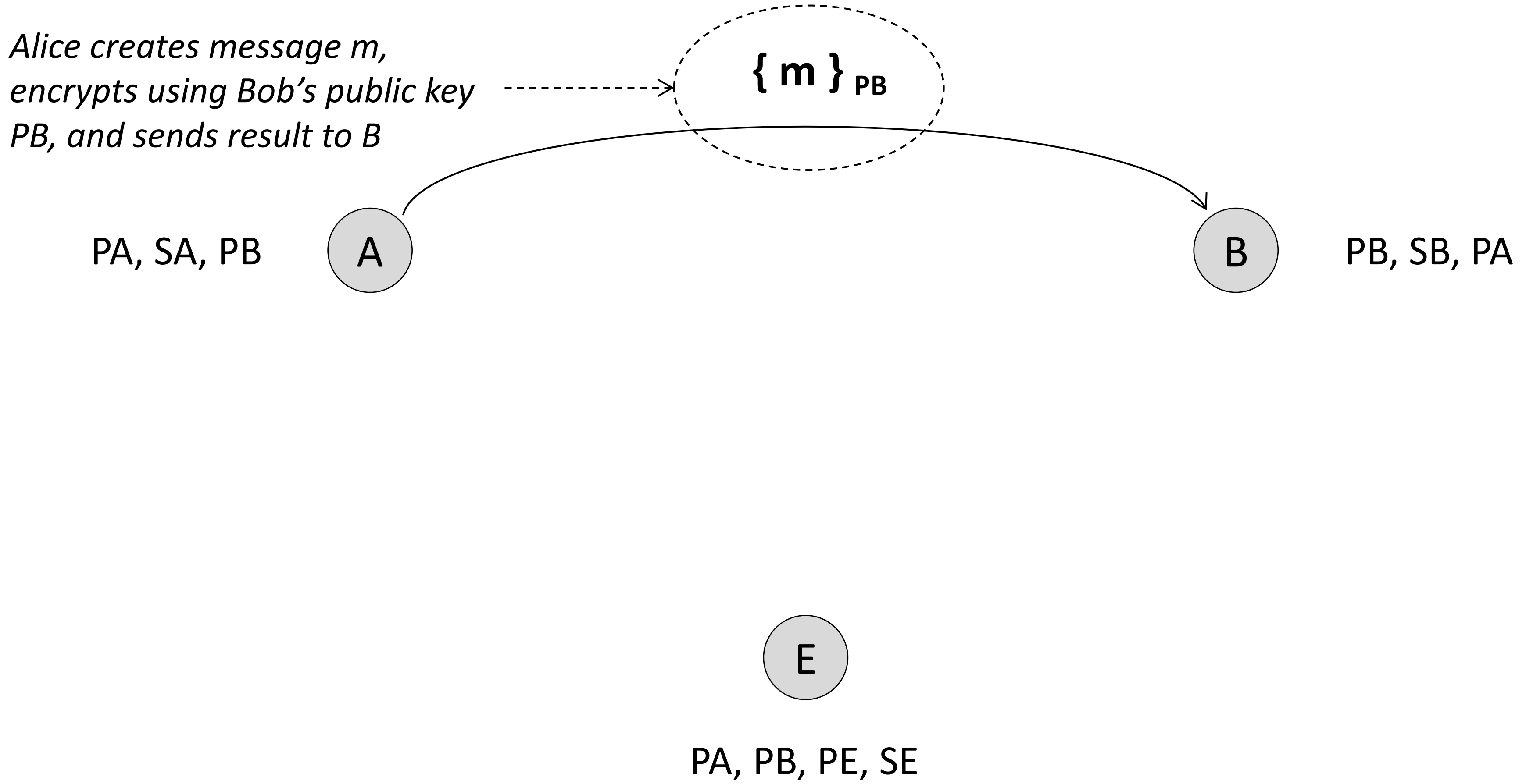
Sending a Secret Message

Alice creates message m . . .



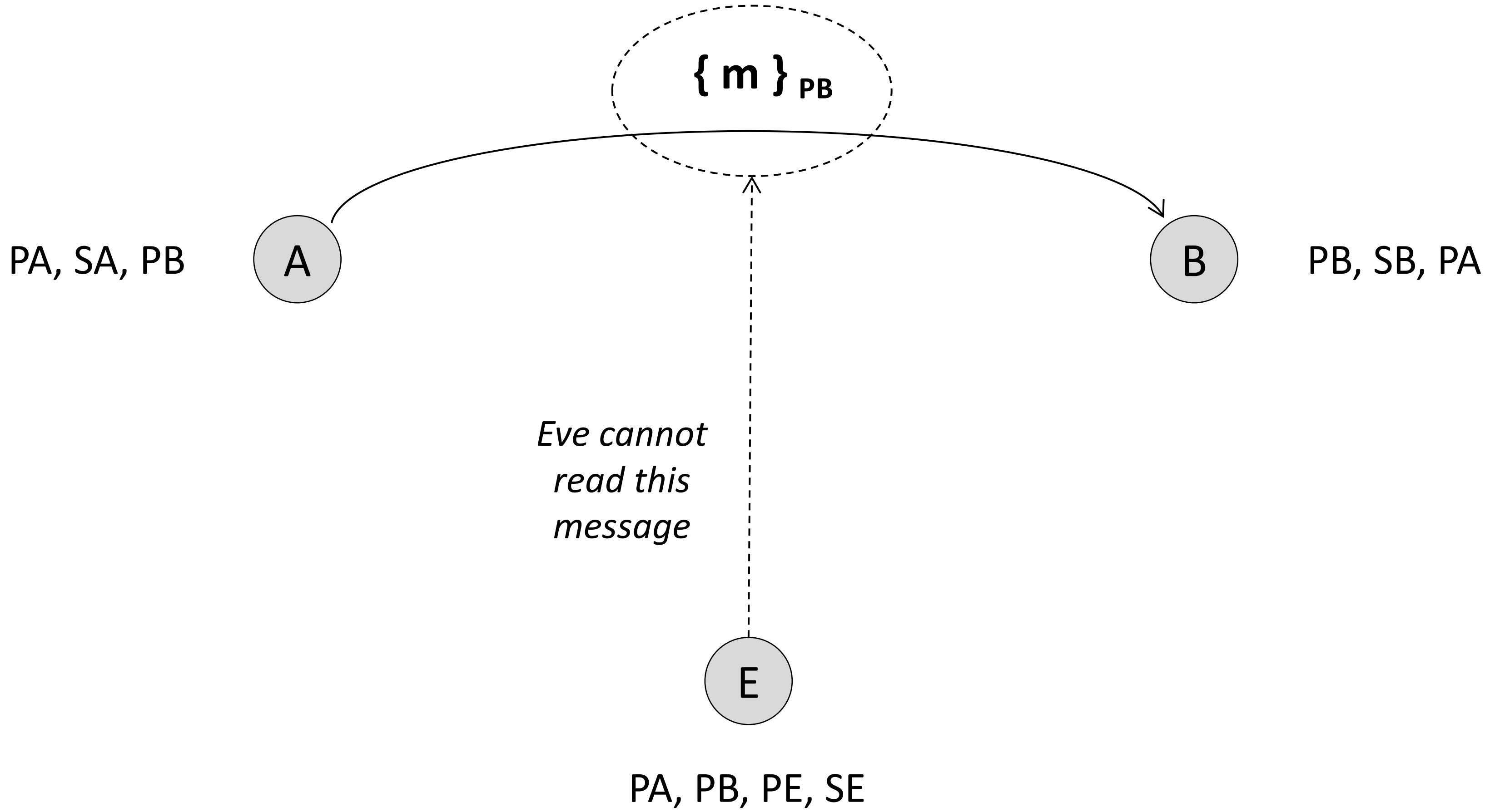
Week 8

Sending a Secret Message



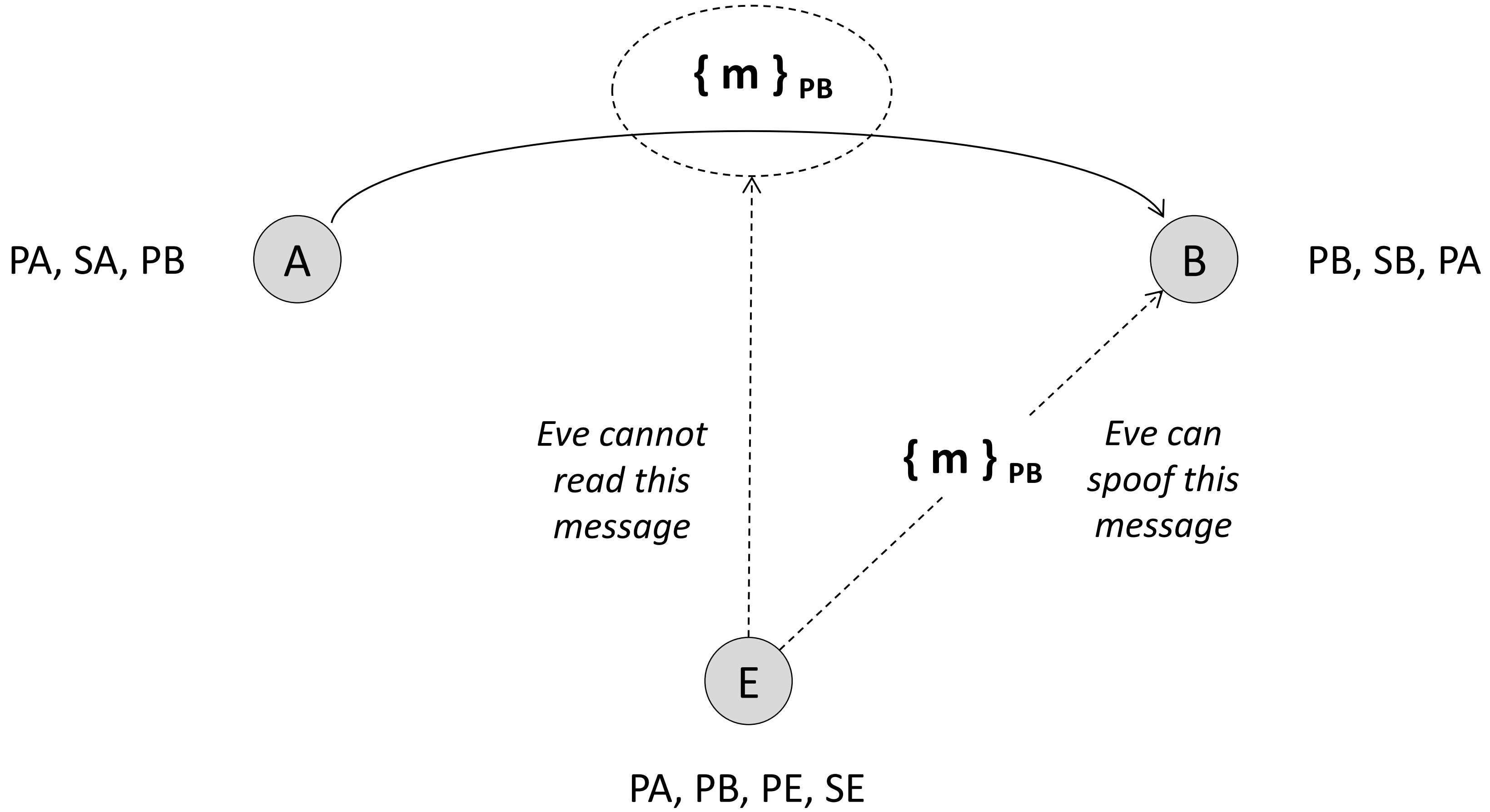
Week 8

Sending a Secret Message



Week 8

Sending a Secret Message



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

Week 8

Sending a Secret Message

$\{ m \}_{PB}$

PA, SA, PB — A

B — PB, SB, PA

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

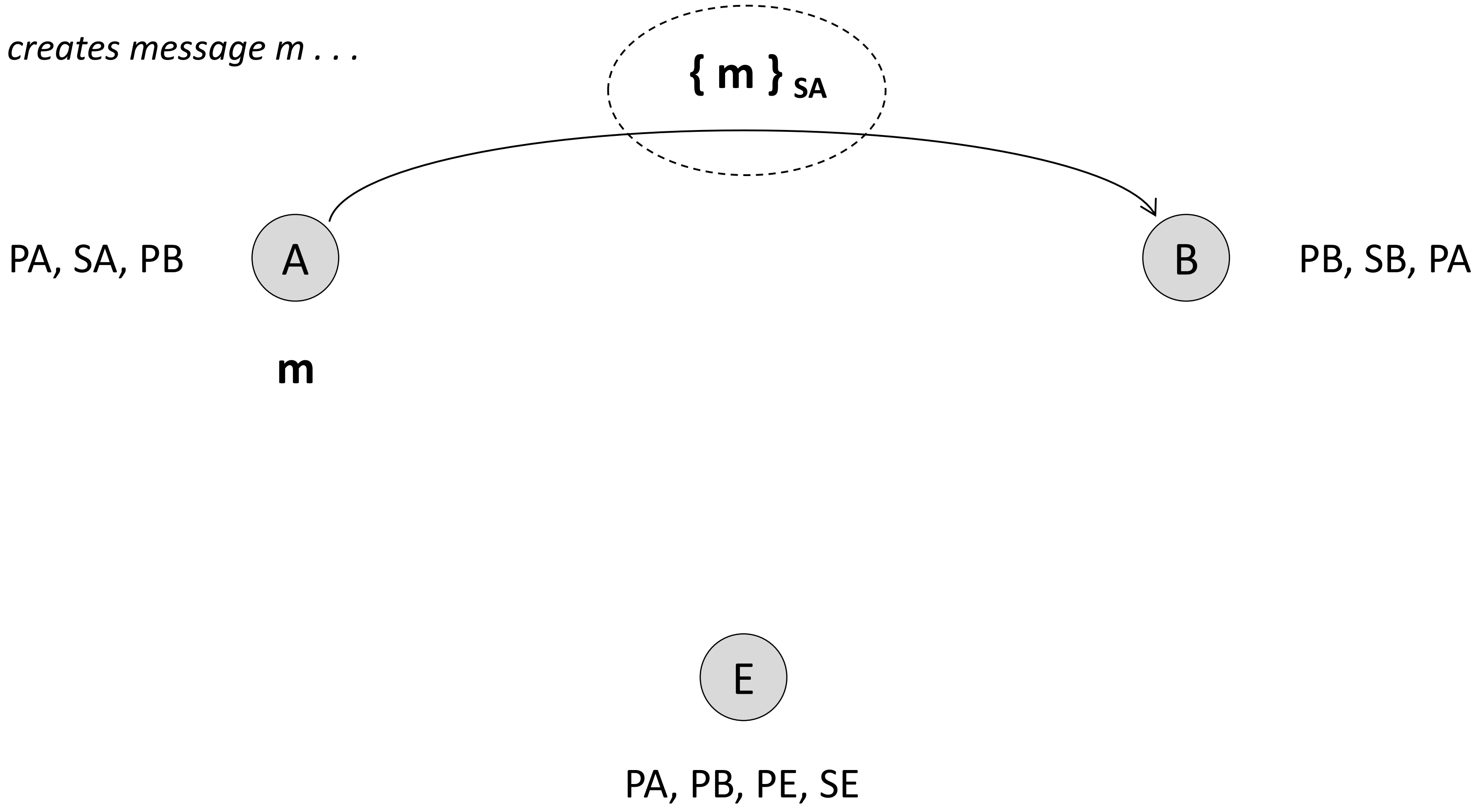
Secrecy Between A and B? **YES**
Authentication of A by B? **NO**

E

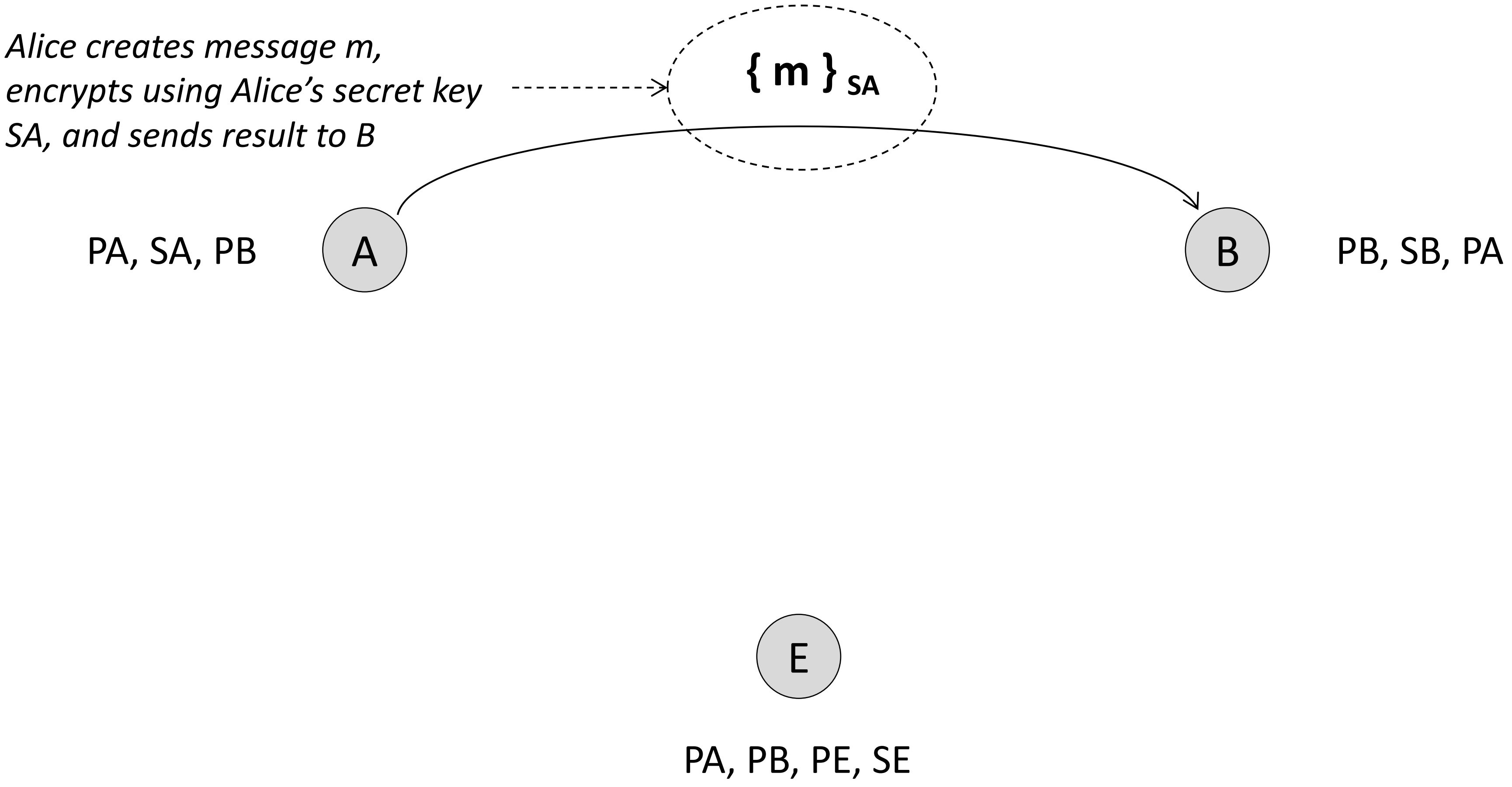
PA, PB, PE, SE

Sending a Signed Message

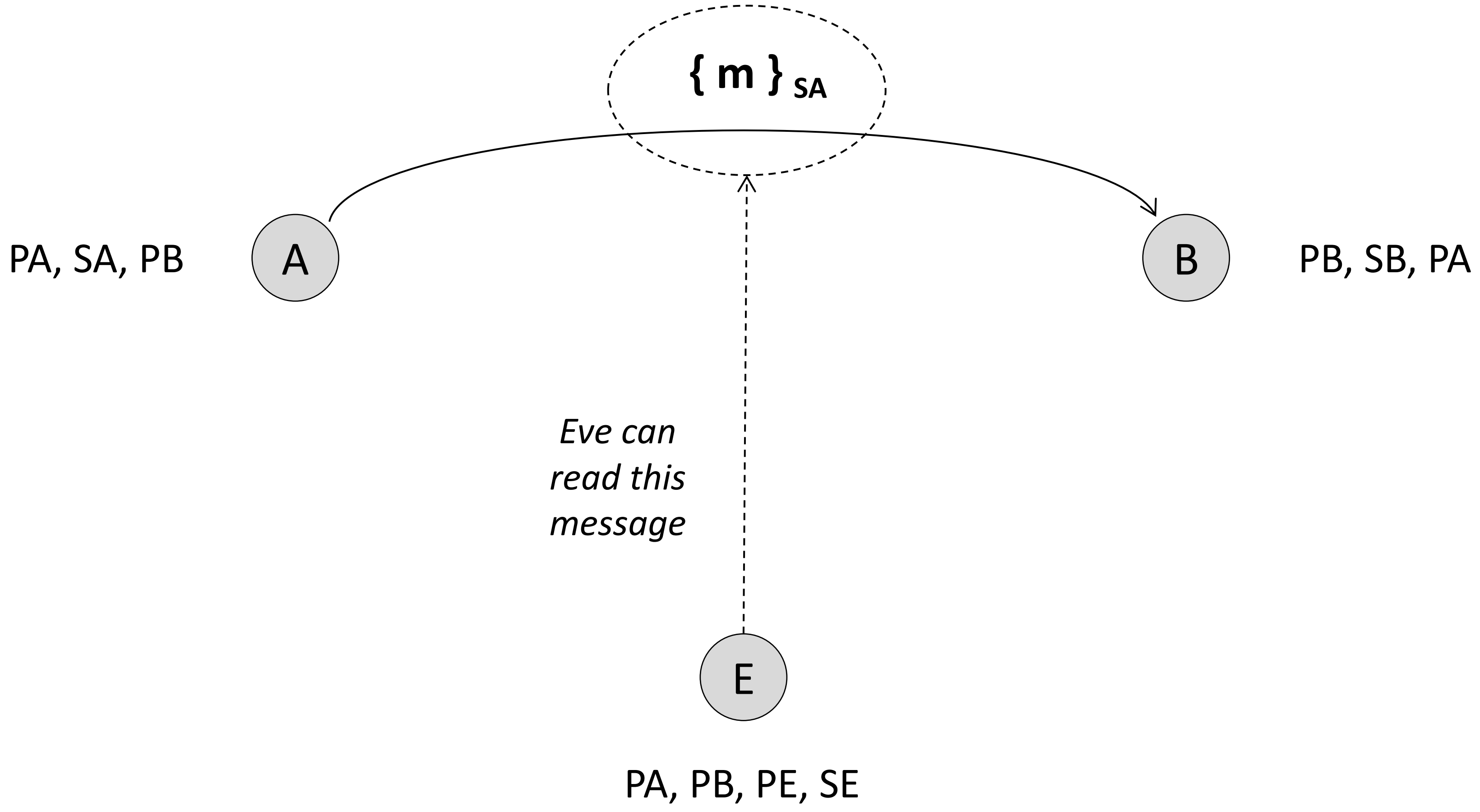
Alice creates message m . . .



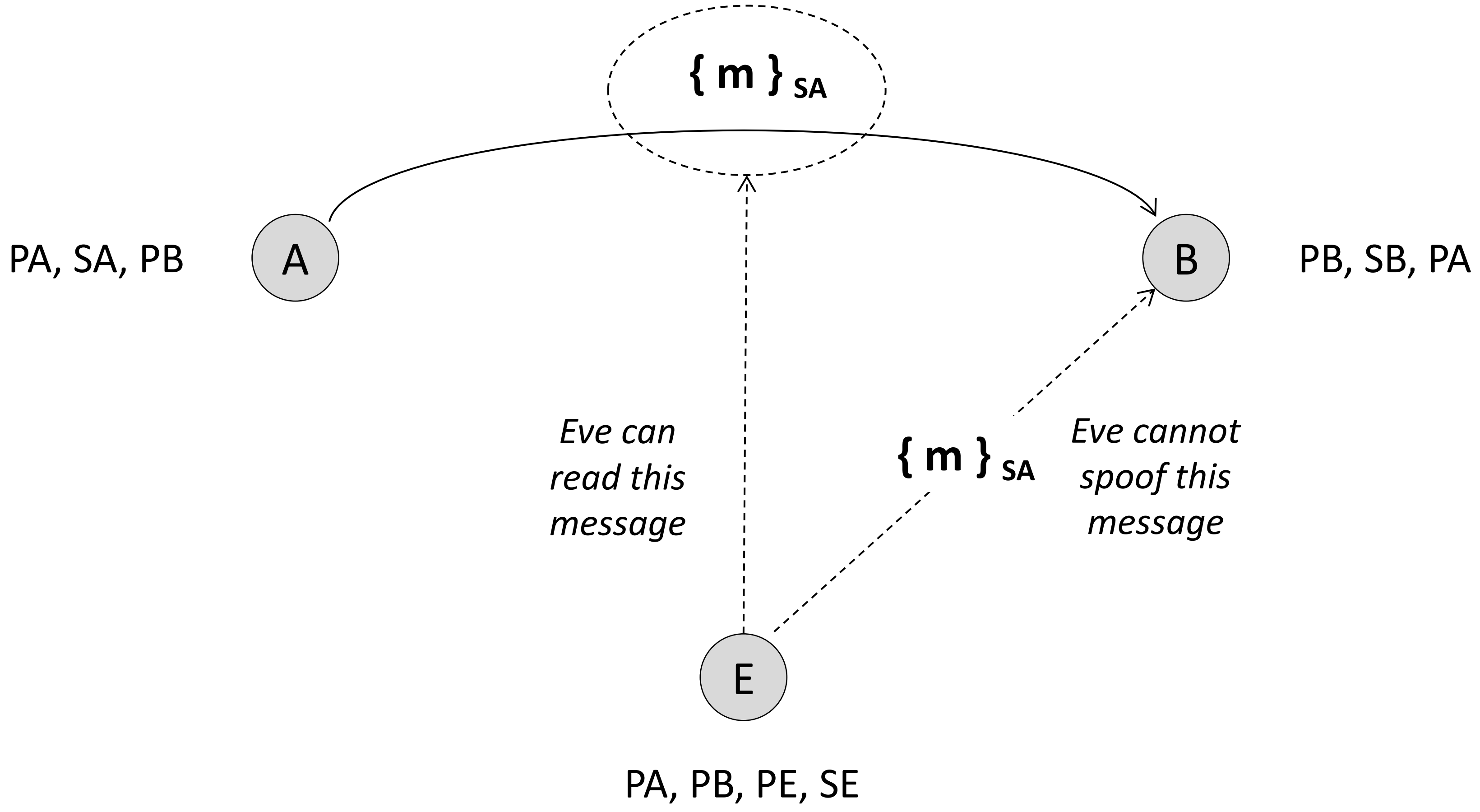
Sending a Signed Message



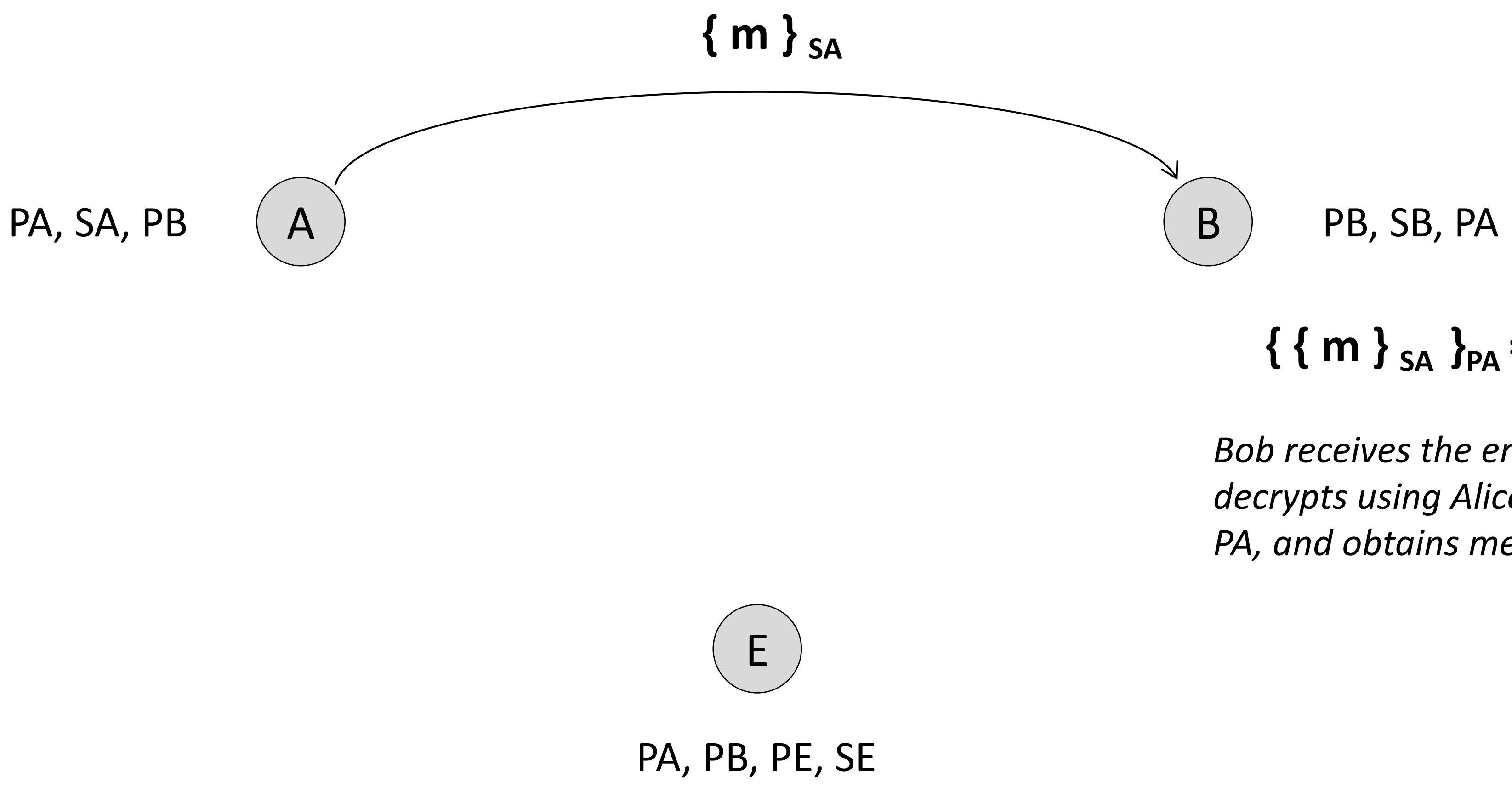
Sending a Signed Message



Sending a Signed Message



Sending a Signed Message



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

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

Week 8

Sending a Signed Message

$\{ m \}_{SA}$

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

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

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

E

PA, PB, PE, SE

Week 8

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 8

Secure Message Exchange

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

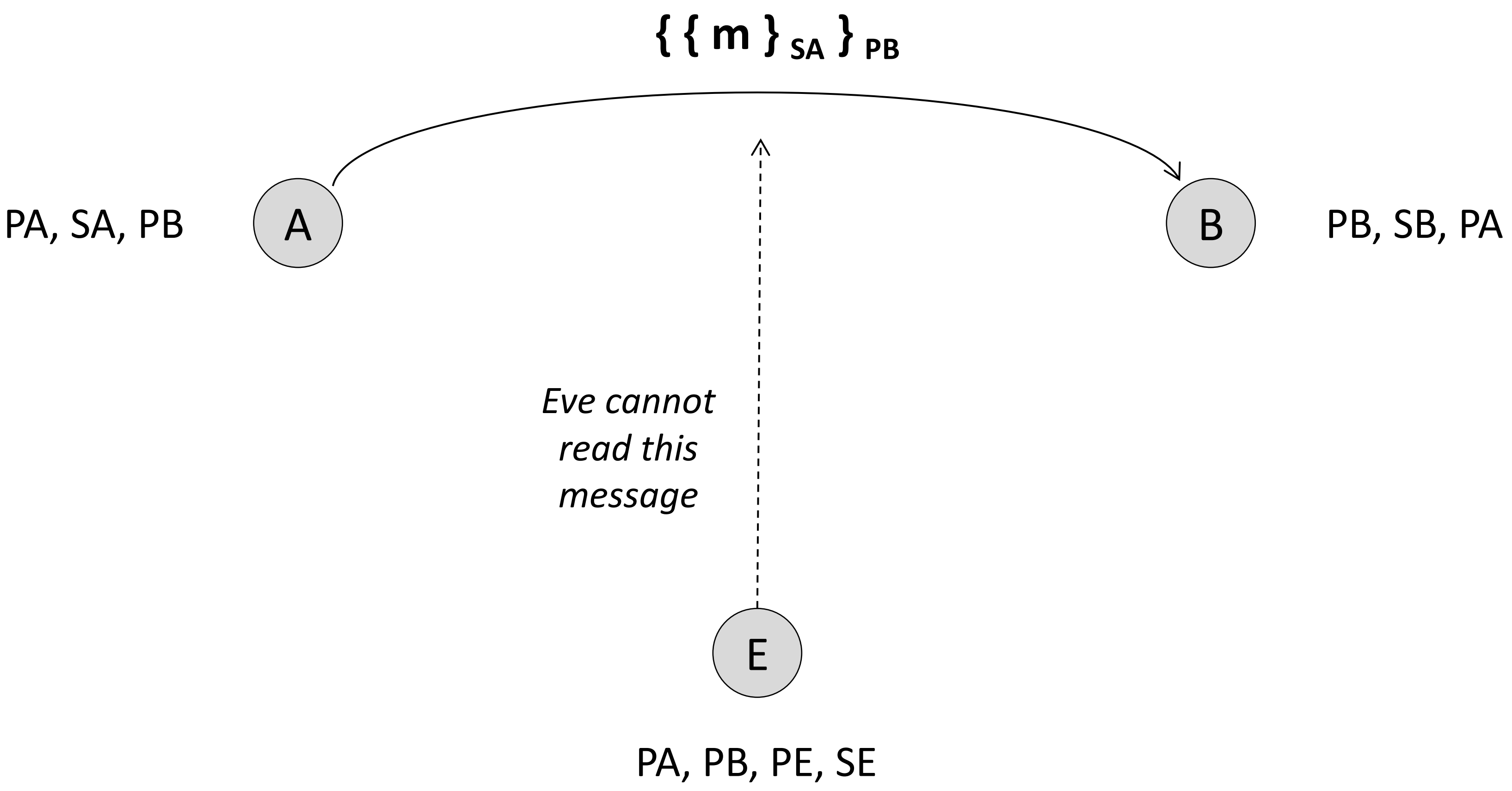
PA, SA, PB — A — B — PB, SB, PA

*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*

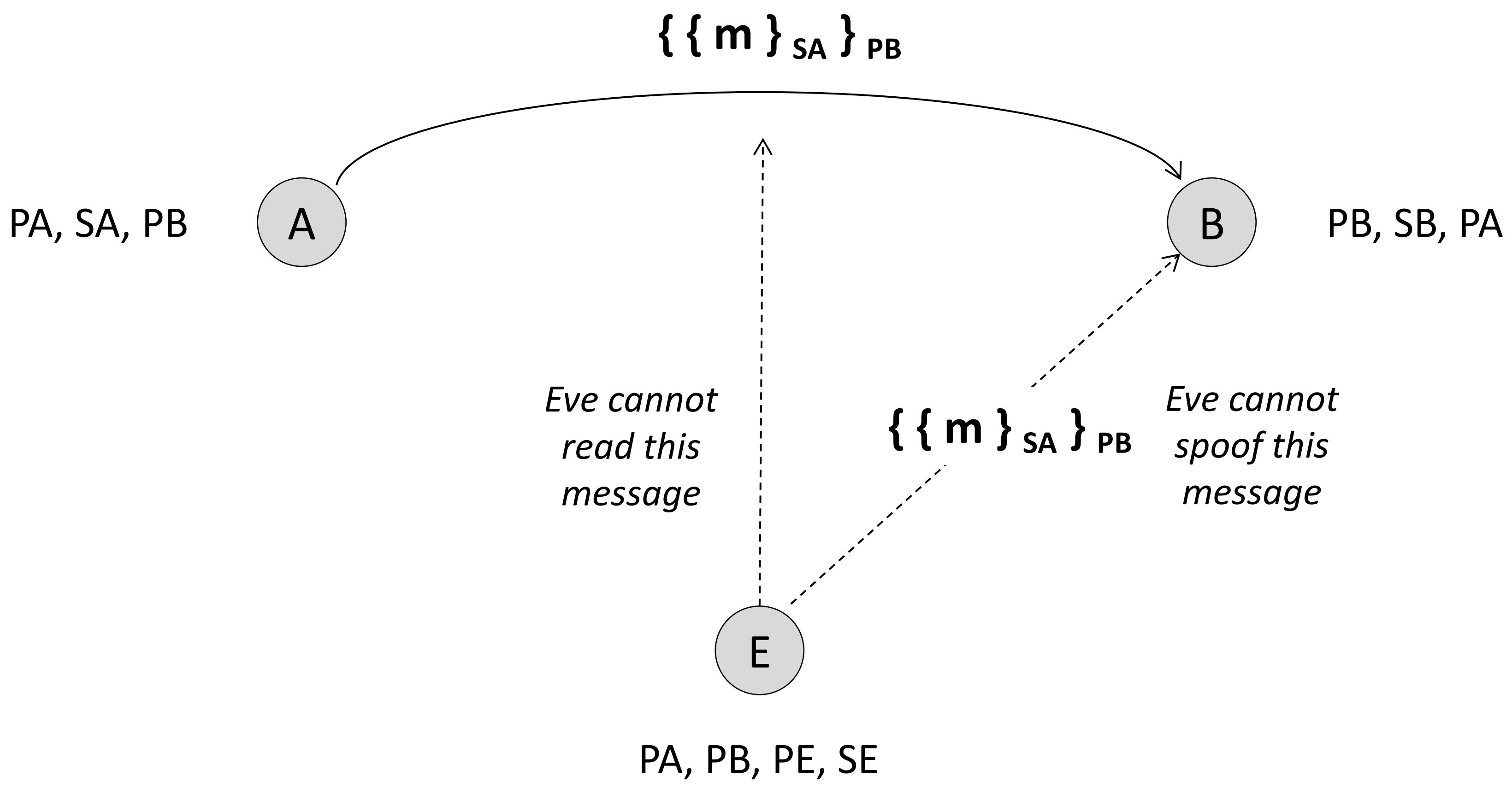
E

PA, PB, PE, SE

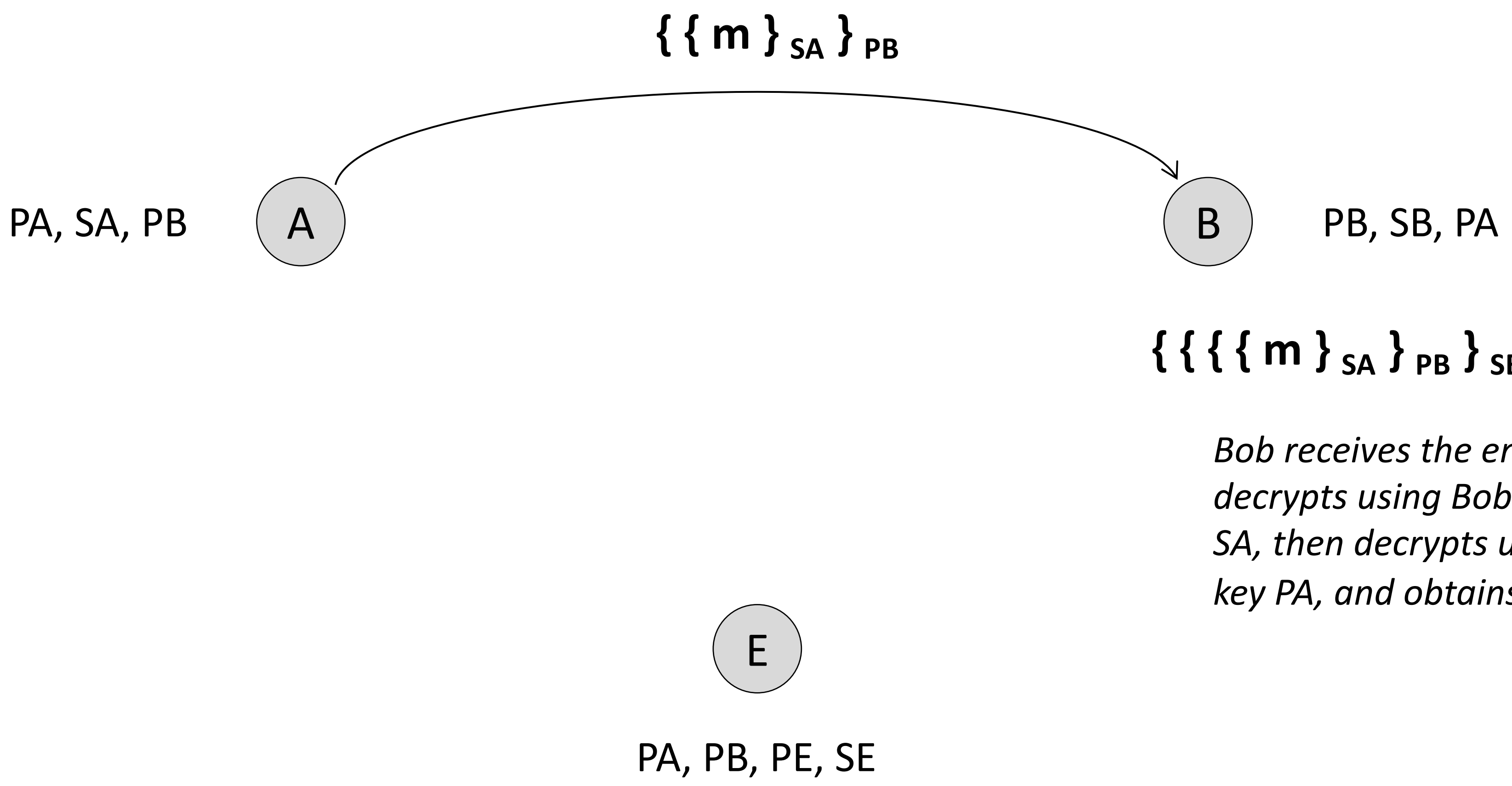
Secure Message Exchange



Secure Message Exchange



Secure Message Exchange



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

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

Week 8

Secure Message Exchange

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

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**

Week 8

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**

Week 8

Secure Message Exchange

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

PA, SA, PB A B PB, SB, PA

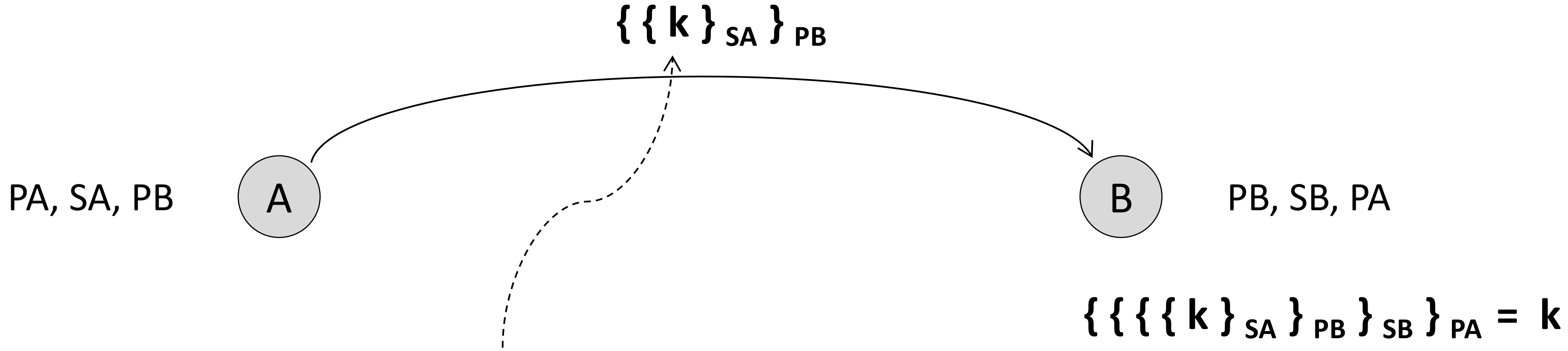
$\{ \{ \{ \{ m \}_{SA} \}_{PB} \}_{SB} \}_{PA} = m$

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

Does this approach scale? **YES**

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

Secure Key Exchange



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 8

Secure Key Exchange

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

PA, SA, PB $\quad A \longrightarrow B \quad$ 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 8

Diffie-Hellman Key Exchange

A B

<u>Goal</u>:

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

Week 8

Diffie-Hellman Key Exchange

p, g A B p, g

__Assume Two Publicly Known Parameters__:

p: Large Prime – Typically 1024 Bits
g: Primitive Element

Week 8

Diffie-Hellman Key Exchange

p, g, a A

B *p, g, b*

Step 1:

A and B each locally generate
private random values a and b

Diffie-Hellman Key Exchange

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

A

B

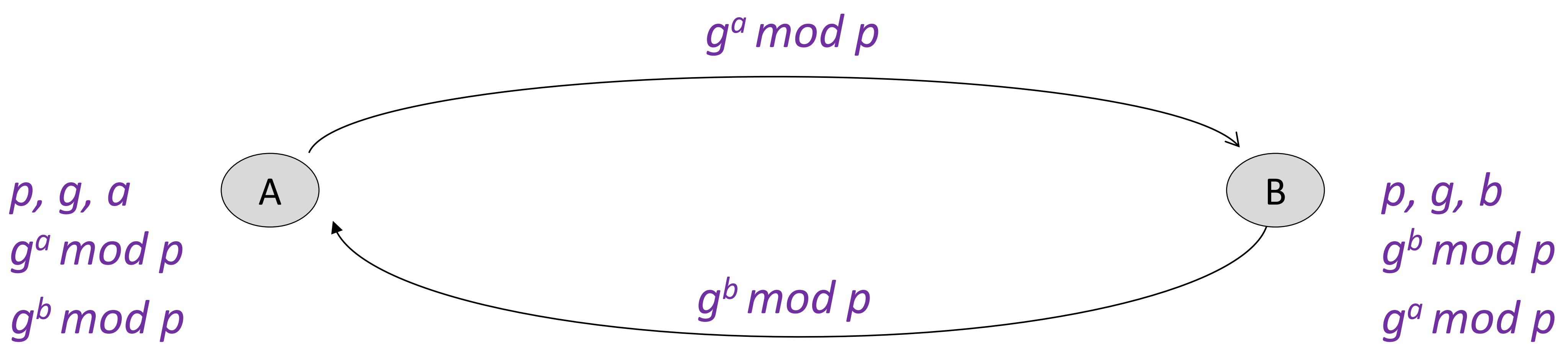
p, g, b
$g^b \bmod p$

<u>Step 2</u>:

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

Week 8

Diffie-Hellman Key Exchange



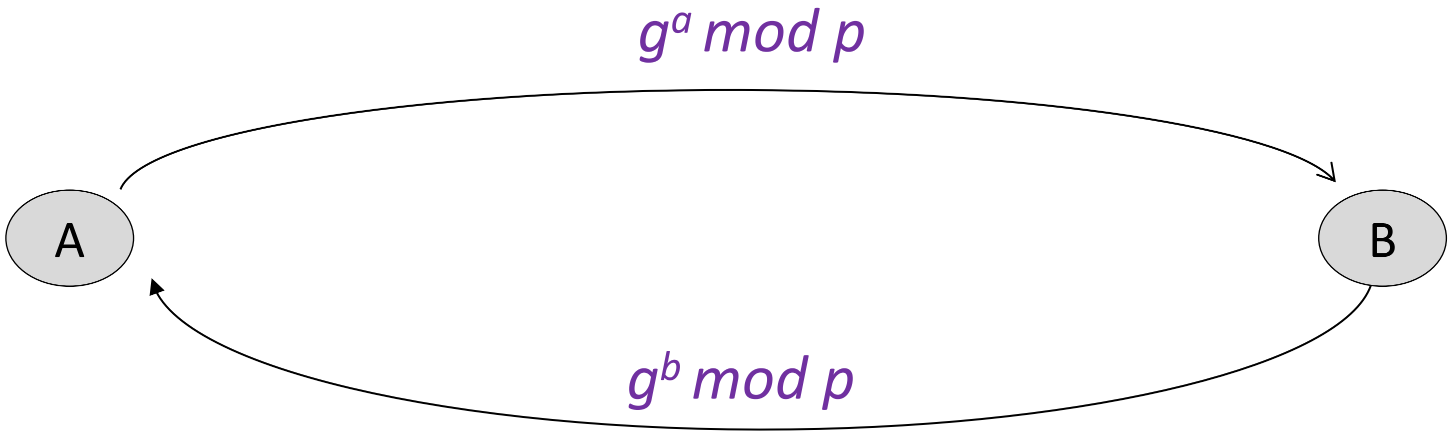
__Step 3__:

A sends $g^a \bmod p$ to B
B send $g^b \bmod p$ to A

Week 8

Diffie-Hellman Key Exchange

p, g, a
$g^a \bmod p$
$g^b \bmod p$
$(g^b \bmod p)^a$



p, g, b
$g^b \bmod p$
$g^a \bmod p$
$(g^a \bmod p)^b$

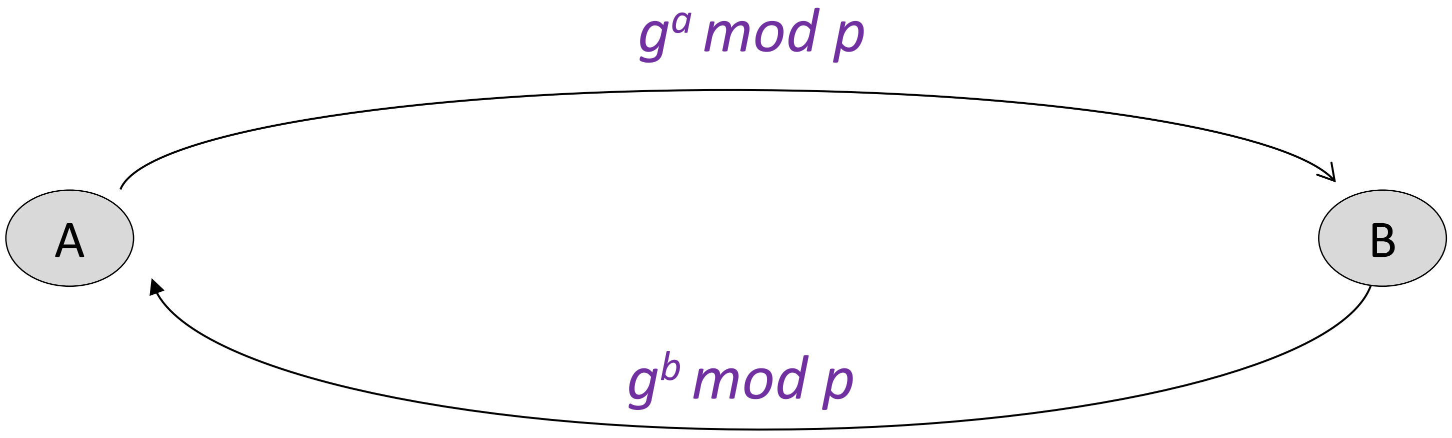
<u>Step 4</u>:

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

Week 8

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

<u>Step 5</u>:

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

Diffie-Hellman Key Exchange

$A \xrightarrow{g^a \bmod p} B$

$B \xrightarrow{g^b \bmod p} A$

A:

p, g, a

$g^a \bmod p$

$g^b \bmod p$

$(g^b \bmod p)^a =$

$g^{ba} \bmod p$

B:

p, g, b

$g^b \bmod p$

$g^a \bmod p$

$(g^a \bmod p)^b =$

$g^{ab} \bmod p =$

$g^{ba} \bmod p$

__Step 5__:

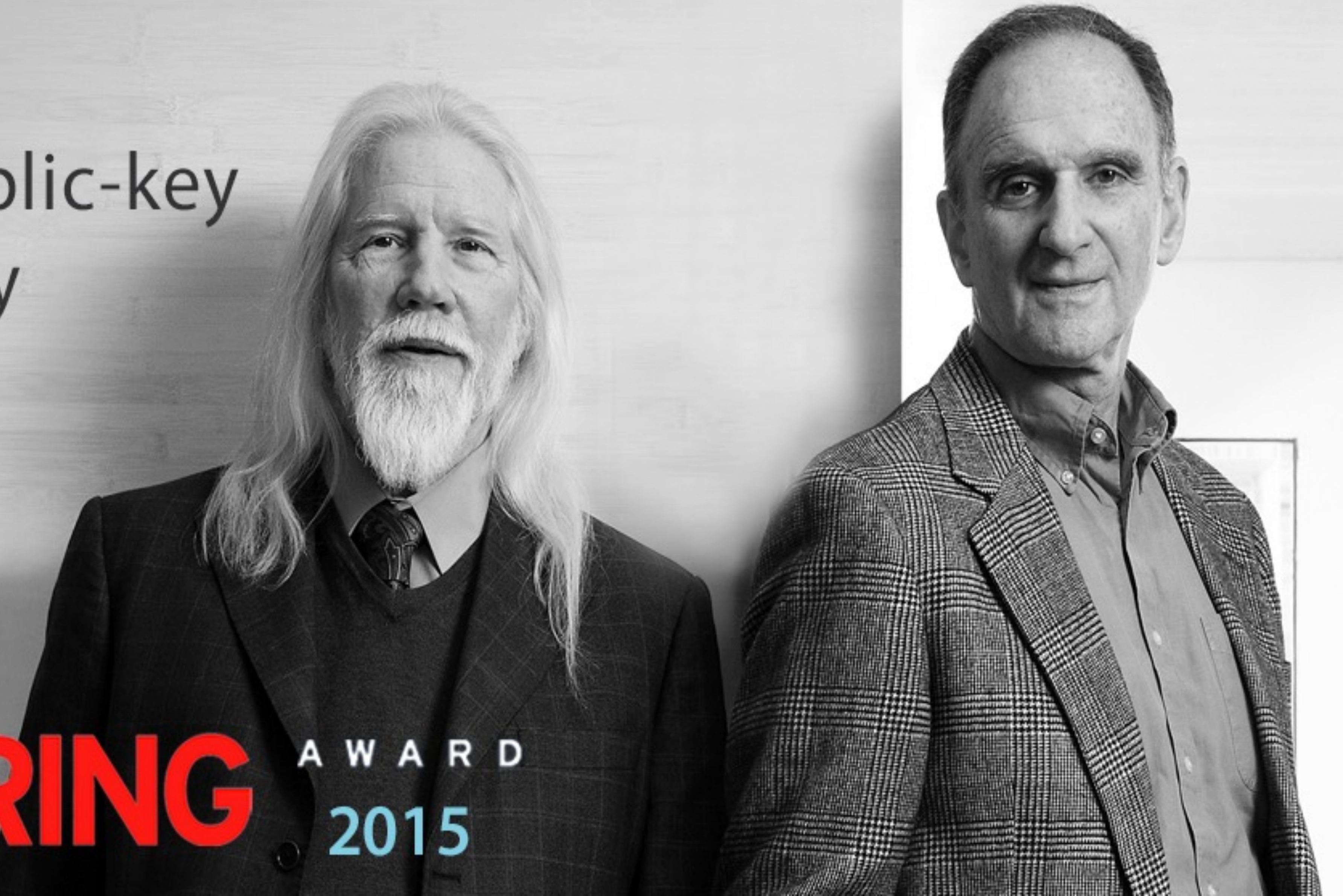
Shared Secret:

$g^{ba} \bmod p$

Week 8

WHITFIELD DIFFIE & MARTIN HELLMAN

Invented public-key cryptography

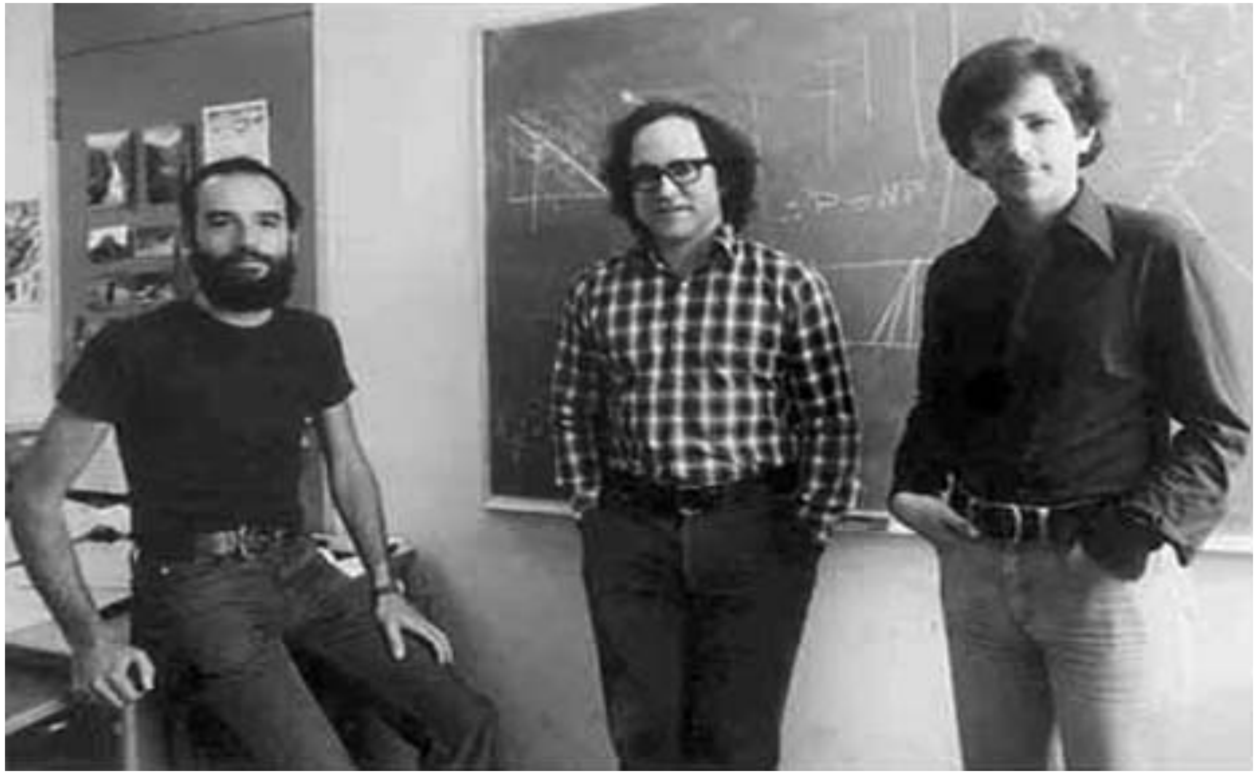


A.M. TURING AWARD 2015

How Does the Original RSA Algorithm Work?

RSA Algorithm

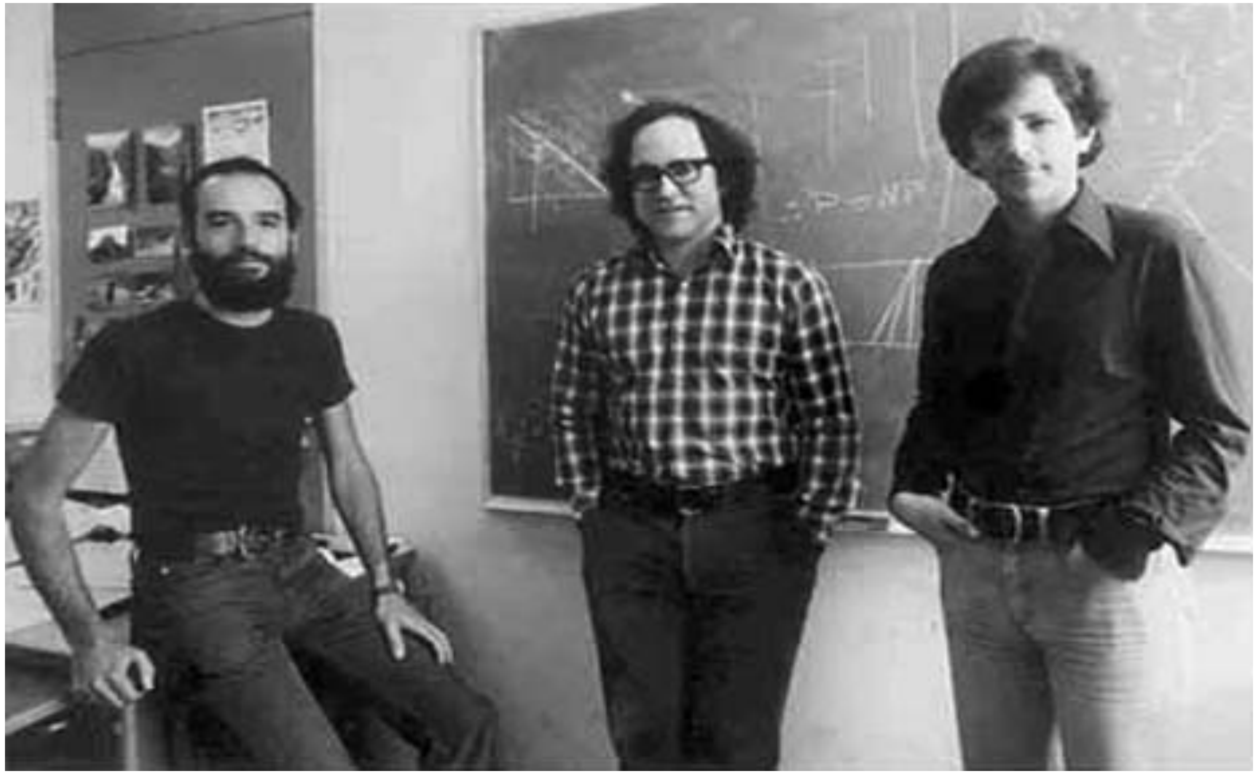
Step 1: Select two prime numbers p and q, each about 100 decimal digits in length



RSA Algorithm

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

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

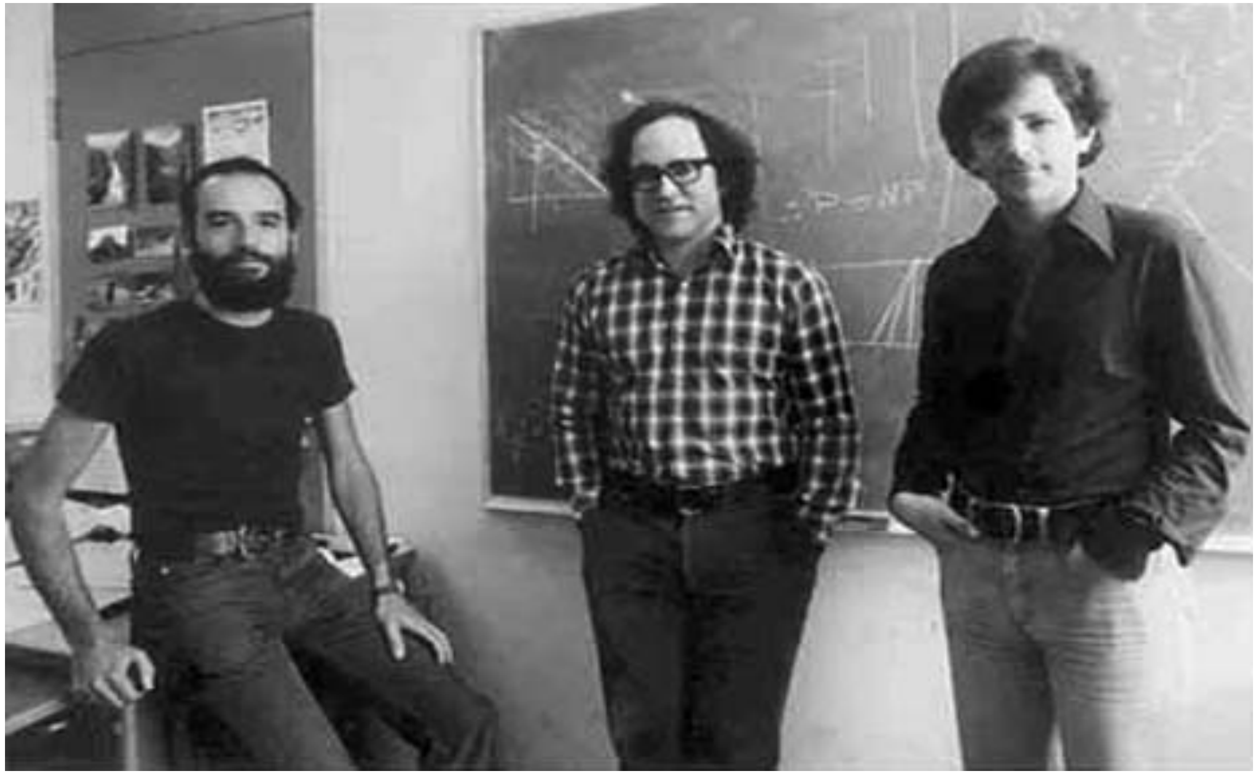


RSA Algorithm

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

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

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



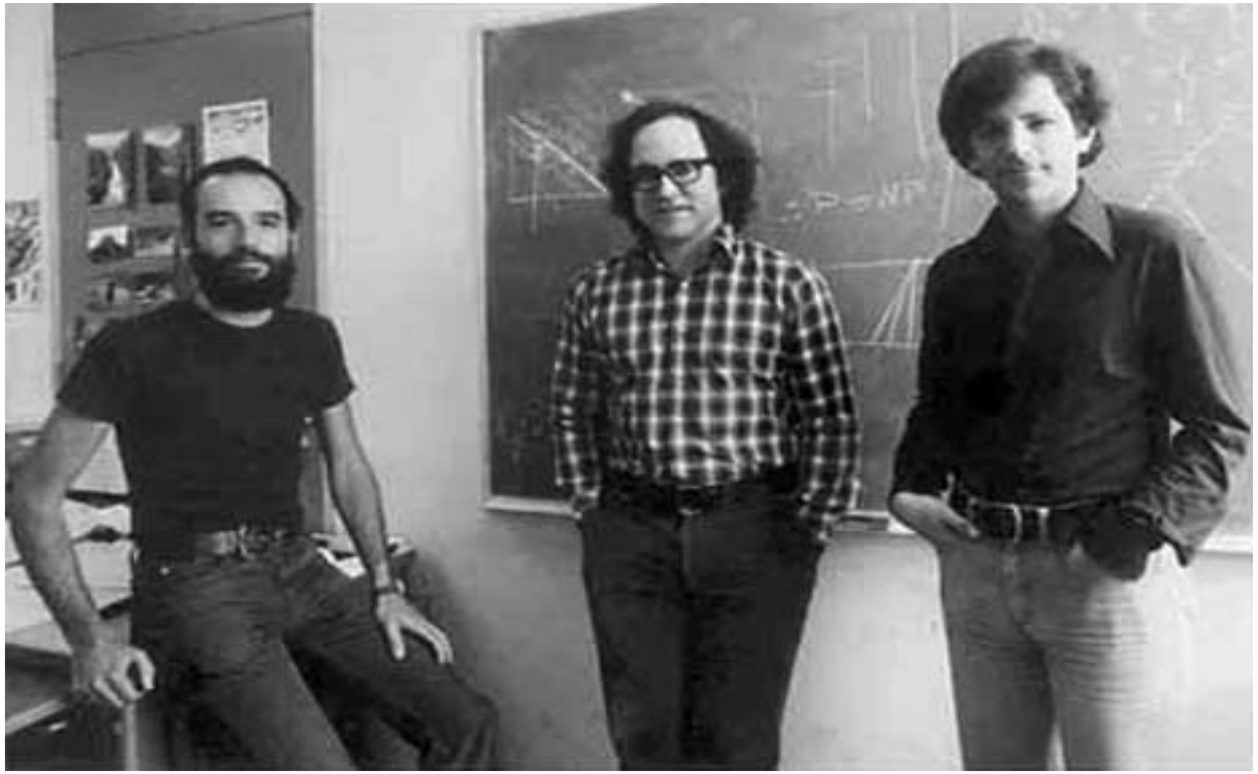
RSA Algorithm

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

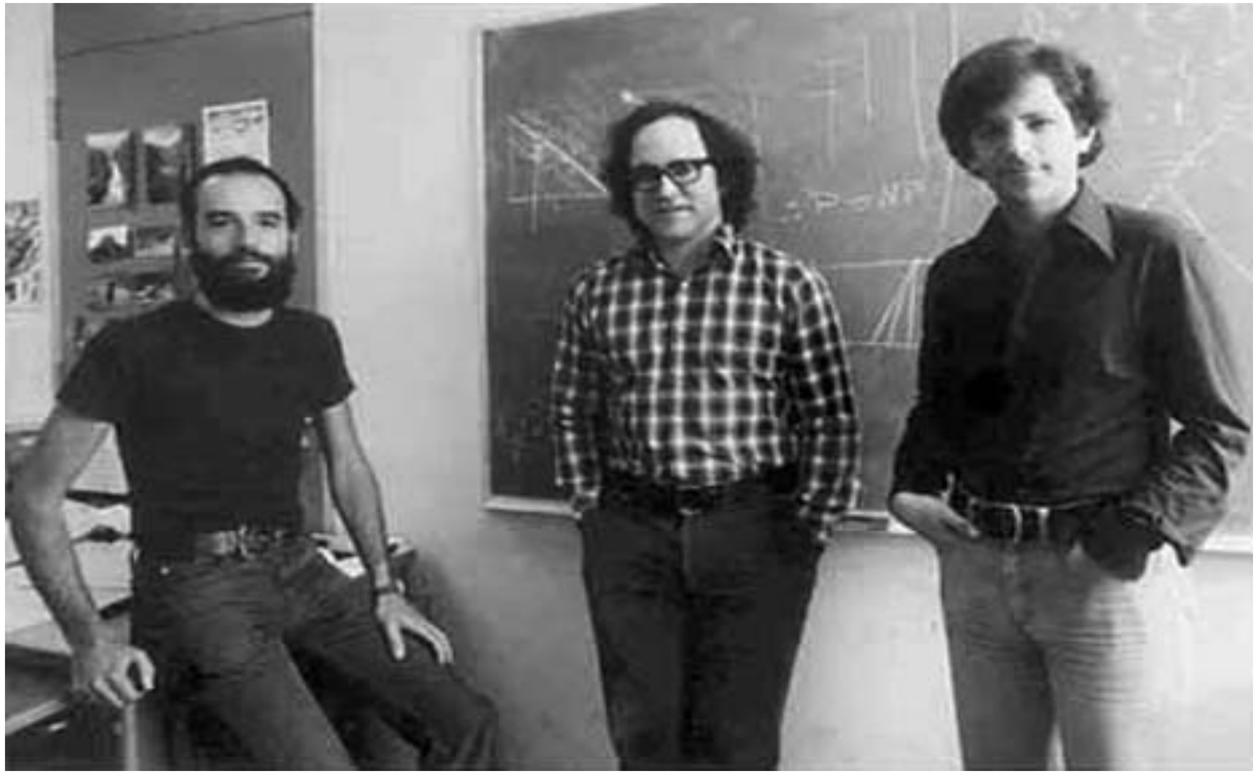
Step 1: Select two prime numbers p and q, each about 100 decimal digits in length

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
Ψ = (p – 1)(q – 1)

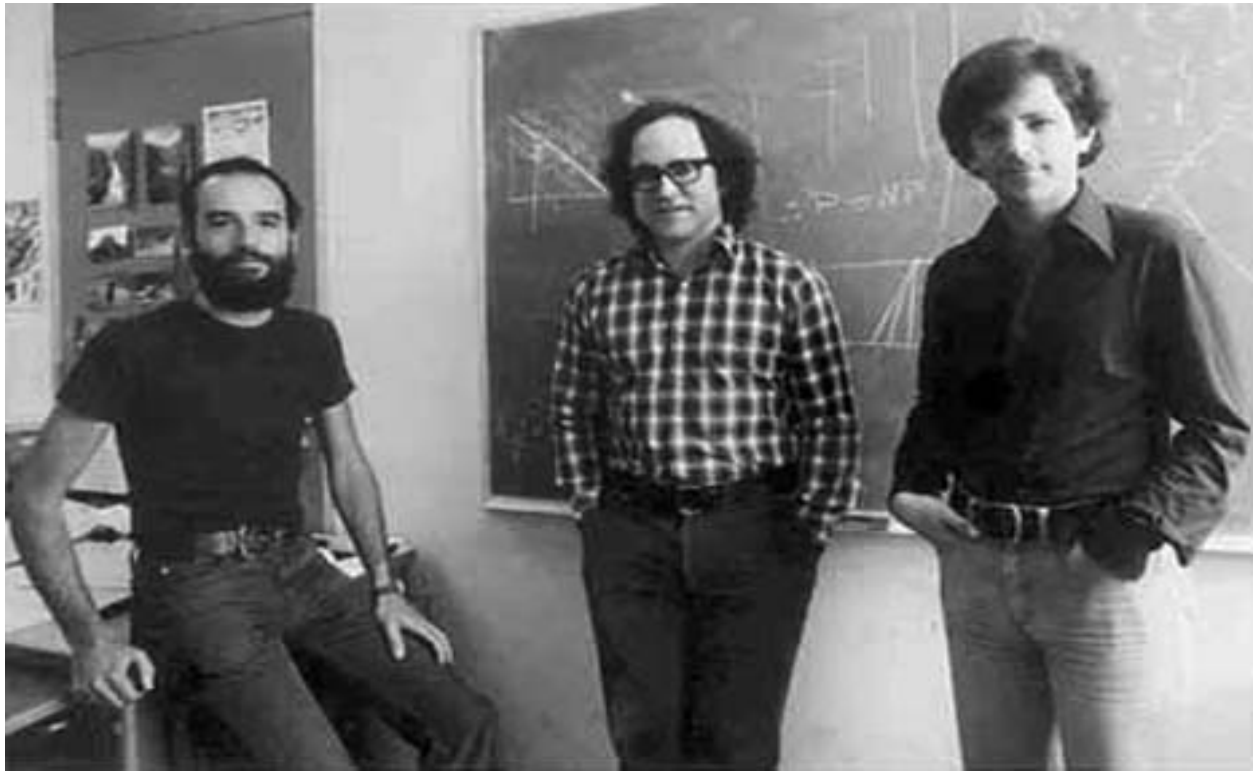
Step 3: Select integer E between 3 and Ψ, which has no common factors with Ψ

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

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

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



Week 8

RSA Algorithm

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

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

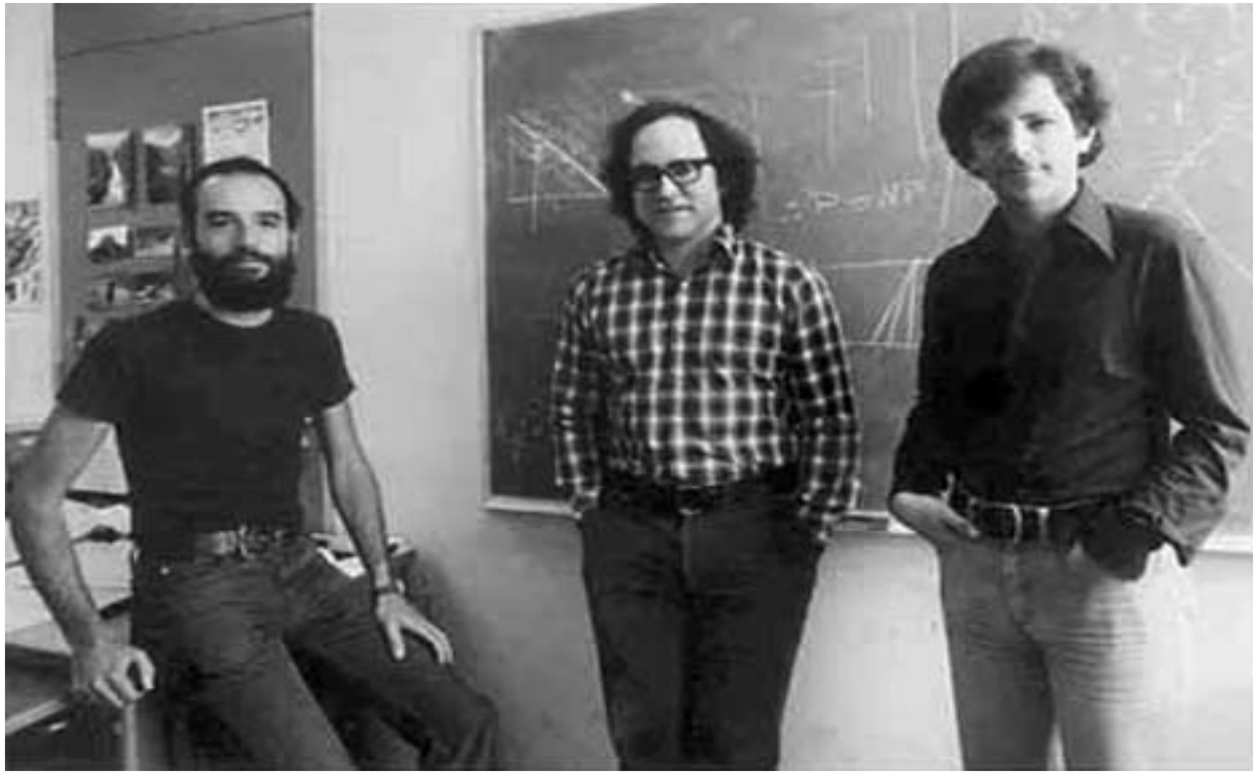
Step 3: Select integer E between 3 and Ψ, which has no common factors with Ψ

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, Ψ = 8 Select E = 5, D = 5

Encrypt “2”: $2^5 \bmod 15 = 2$

Decrypt “2”: $2^5 \bmod 15 = 2$


Week 8
RIVEST, ADI SHAMIR & LEN ADLEMAN
acm
AWARD
2002
A.M. TURING
RSA public-key cryptography

Who Really Invented Public Key Technology?
(Hint: UK)

Week 8

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

Week 8

James Ellis, Engineer at GCHQ – Circa 1969

Week 8

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.

NTIS SOP memo 7 Jan 60

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

DIVISION: Electronics ~~(6)~~ 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

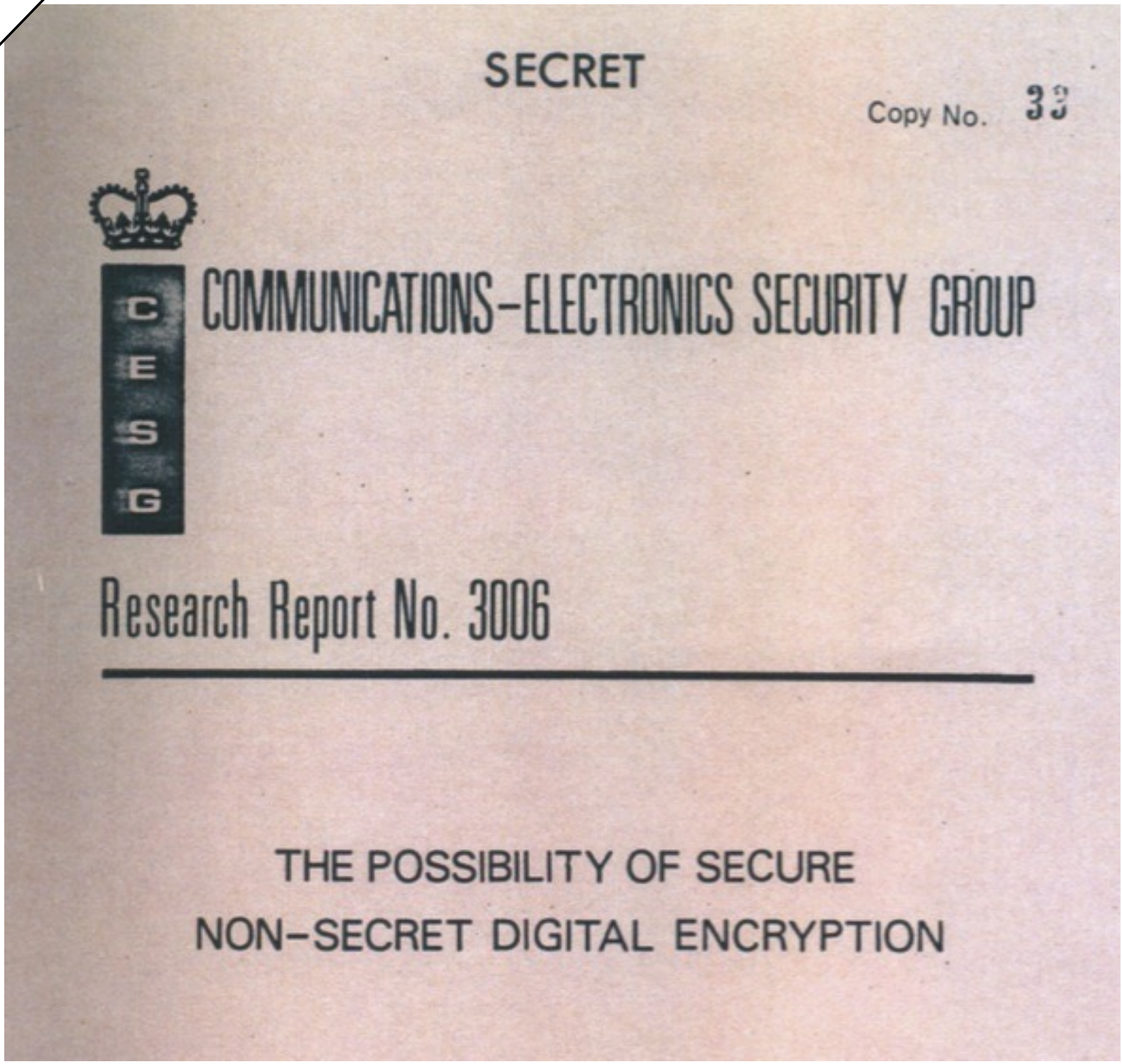
SECRET

Week 8

GCHQ – Original and New Headquarters in Cheltenham, UK

Week 8

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

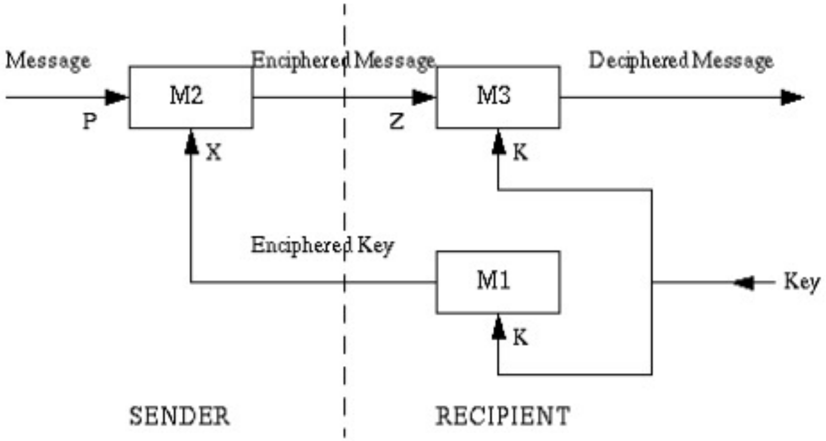


Fig. 1

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: -

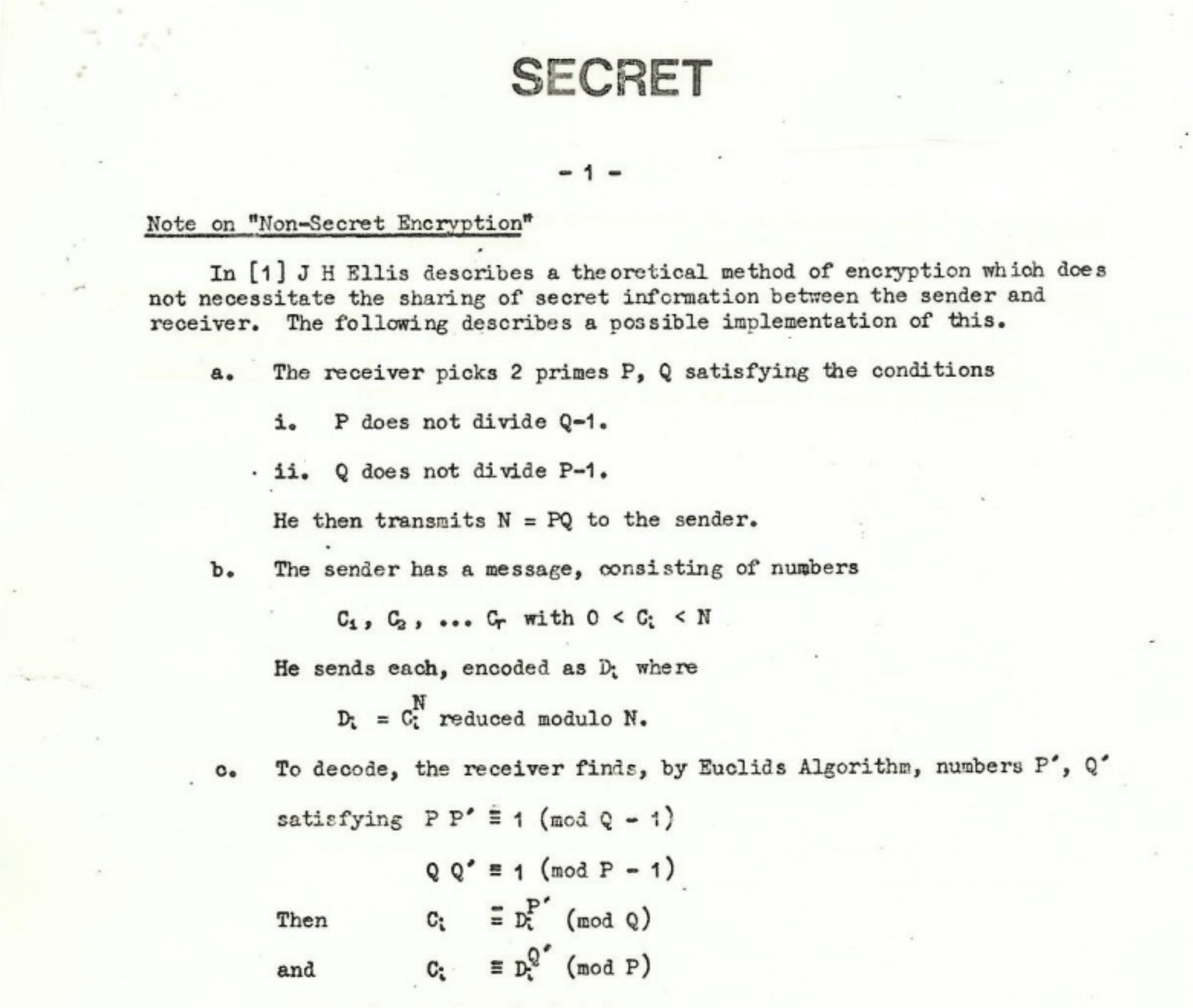
   a. Let k have n different possible values and p have m different possible values, for simplicity take them to be the integers 1 to n and 1 to m respectively. Let x have the same range of values as k, and z have the same range as p.

   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.

Week 8

Clifford Cocks and Malcolm Williamson



SECRET

- 1 -

Note on "Non-Secret Encryption"

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

$C_1, C_2, \ldots C_r$ 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'

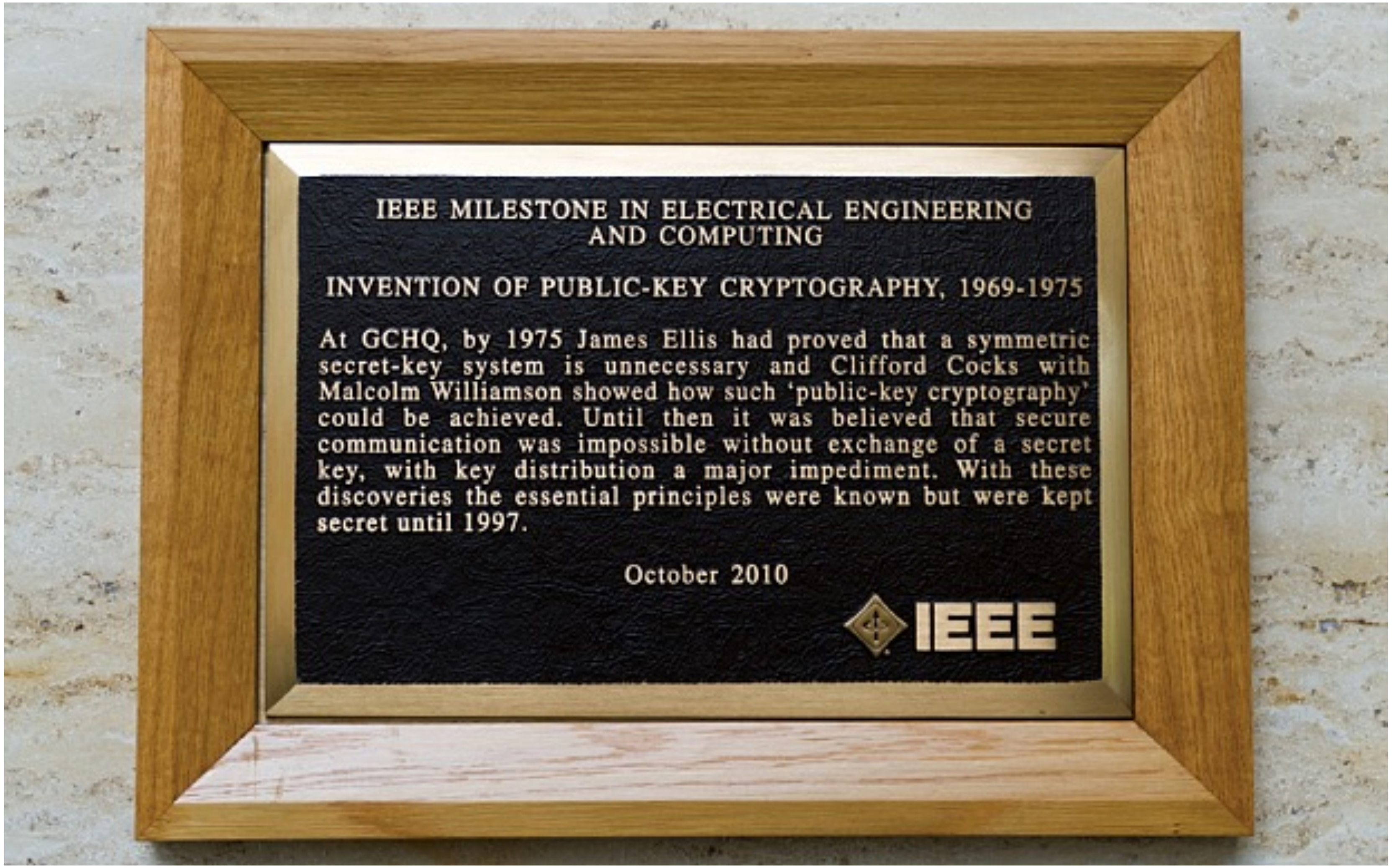
satisfying $P P' \equiv 1 \pmod{Q-1}$

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

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

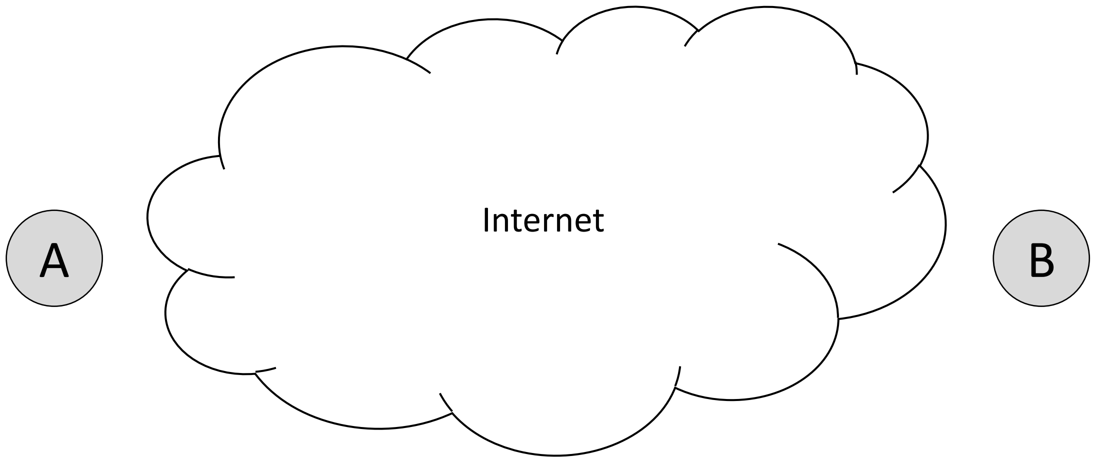
and $C_i \equiv D_i^{Q'} \pmod{P}$

Credit Where Credit is Due

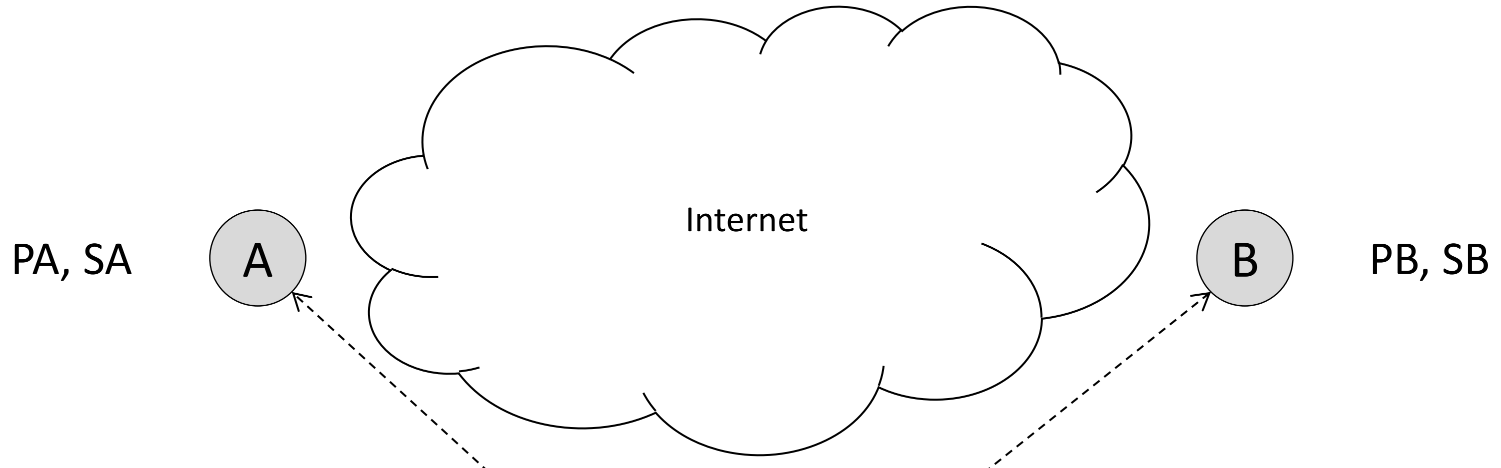


How are Keys Distributed?

Public Key Distribution

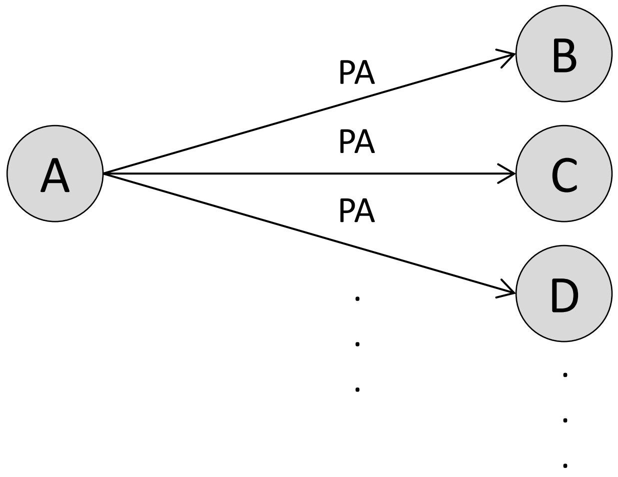


Public Key Distribution



Initial State: A, B, and CA generate their own key pairs but do not possess other public keys

Public Key Distribution – Manual Distribution

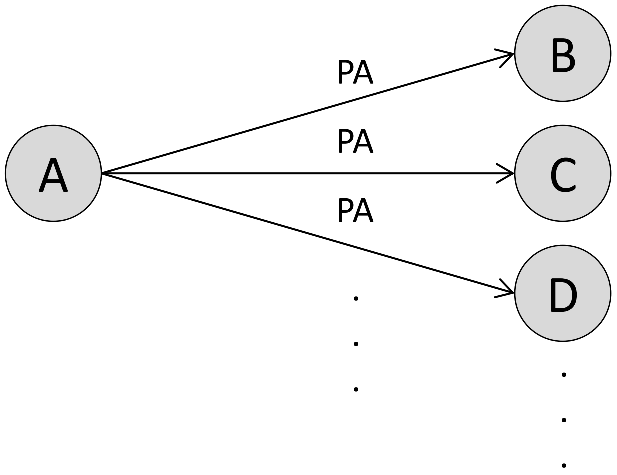


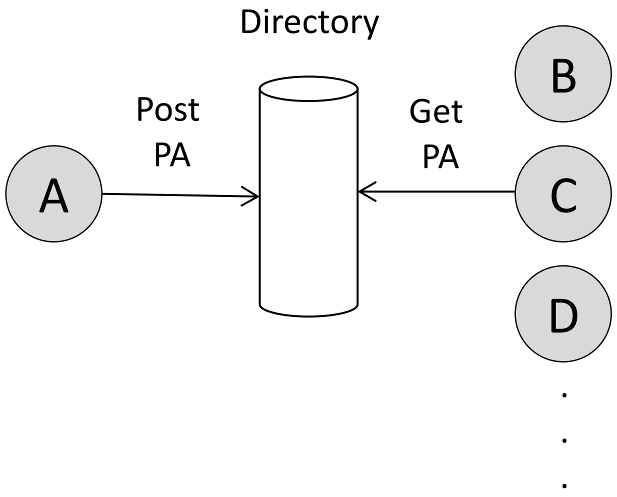
Manual Distribution:

- Easy, attach to email, etc.
- Does not scale across large groups
- One new participant to group of size X, requires X key actions

Week 8

Public Key Distribution – Directory Post





Manual Distribution:

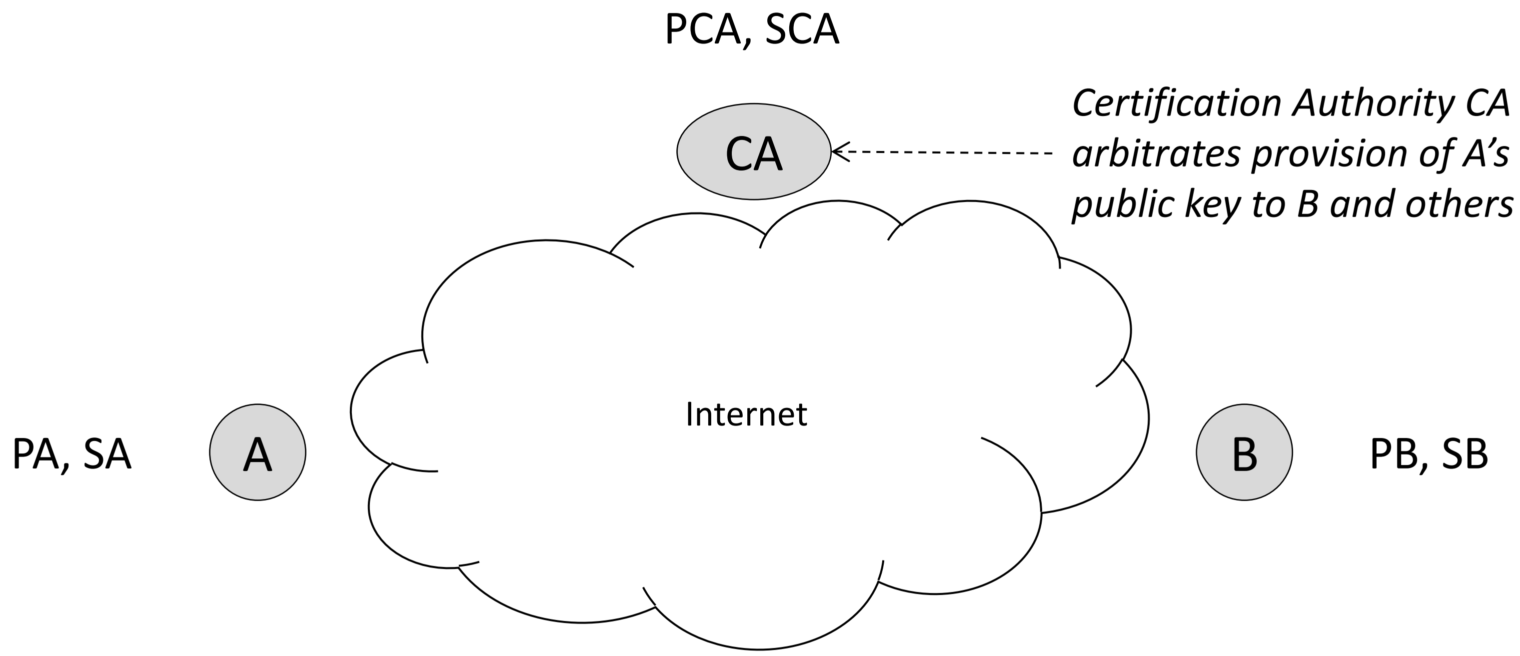
- Easy, attach to email, etc.
- Does not scale across large groups
- One new participant to group of size X, requires X key actions

Directory Post Distribution:

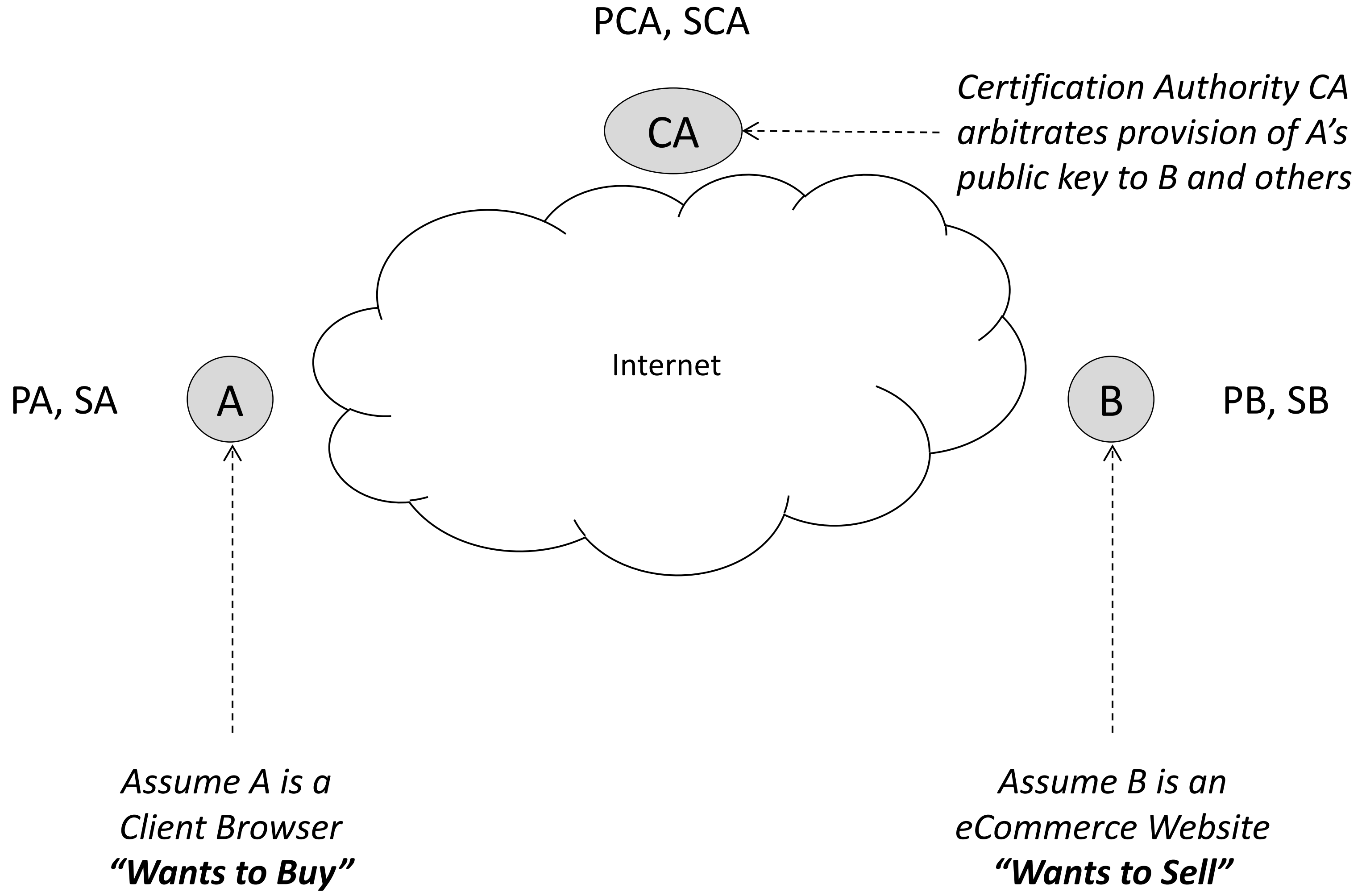
- Easy for enterprise directories
- Does not scale across large groups
- Vulnerable to outage – SPOF
- One new participant to group of size X, requires 1 post to directory

Week 8

Public Key Distribution – Certification Authority

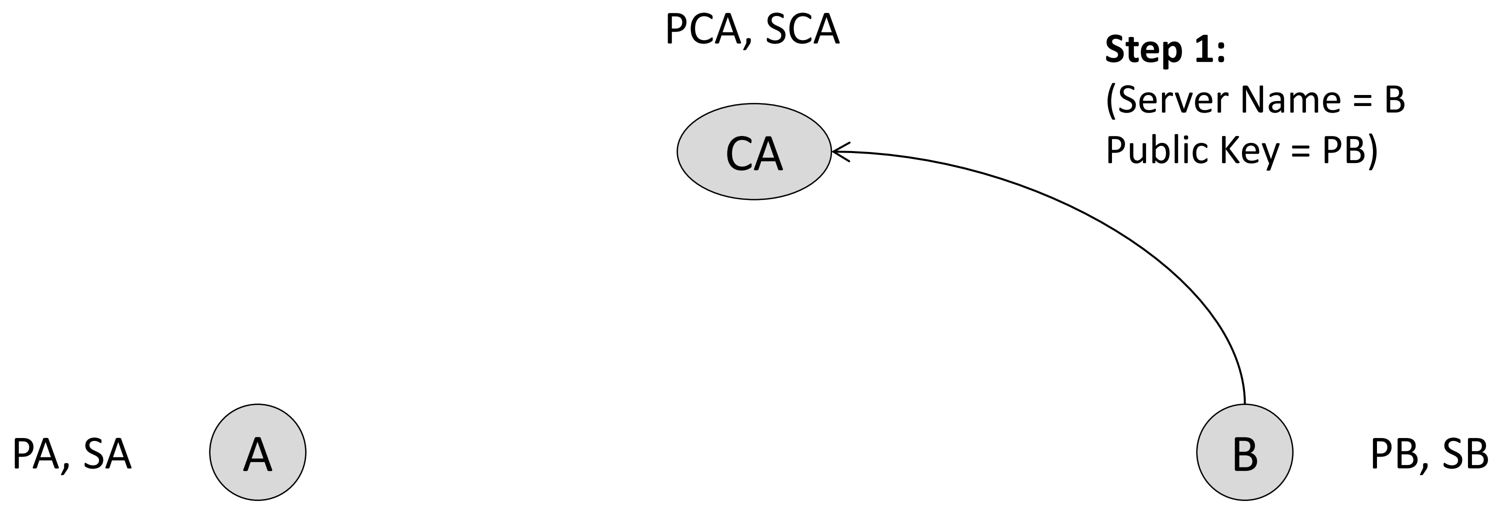


Public Key Distribution – Certification Authority



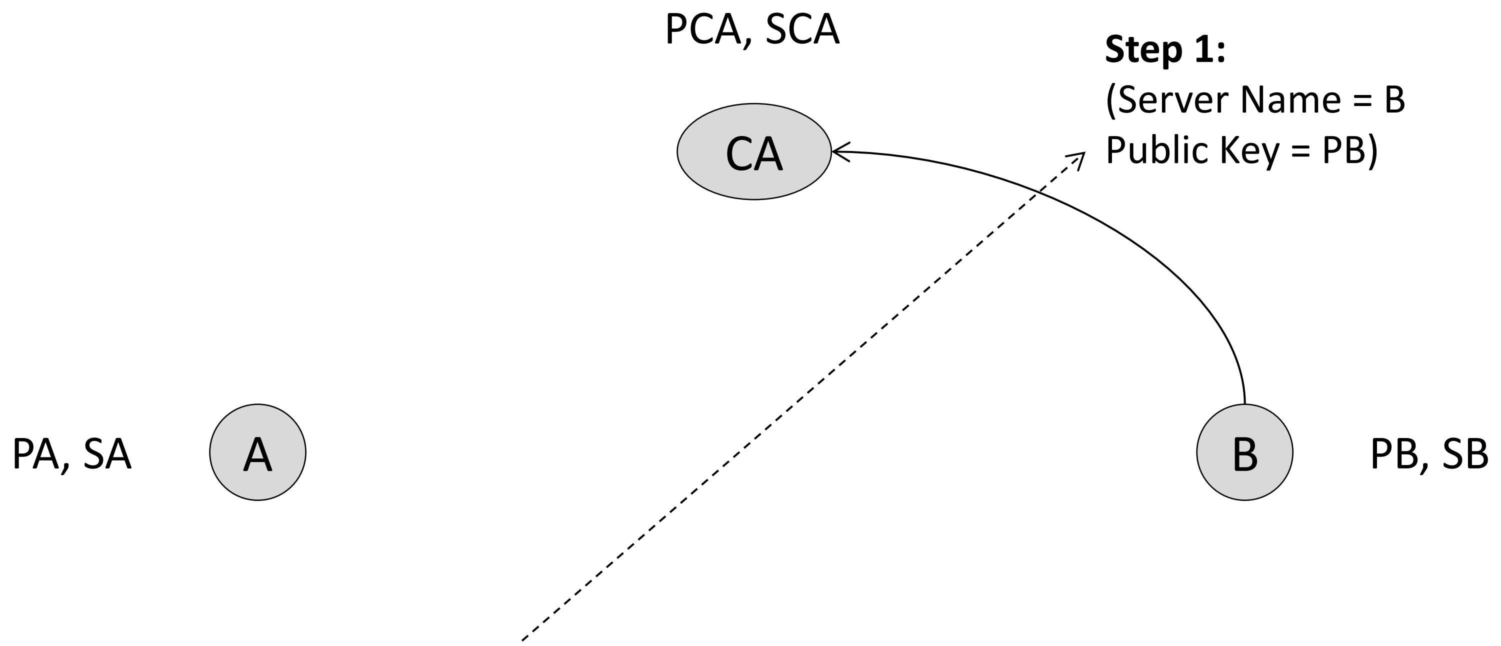
Week 8

Public Key Distribution – Certification Authority



Week 8

Public Key Distribution – Certification Authority

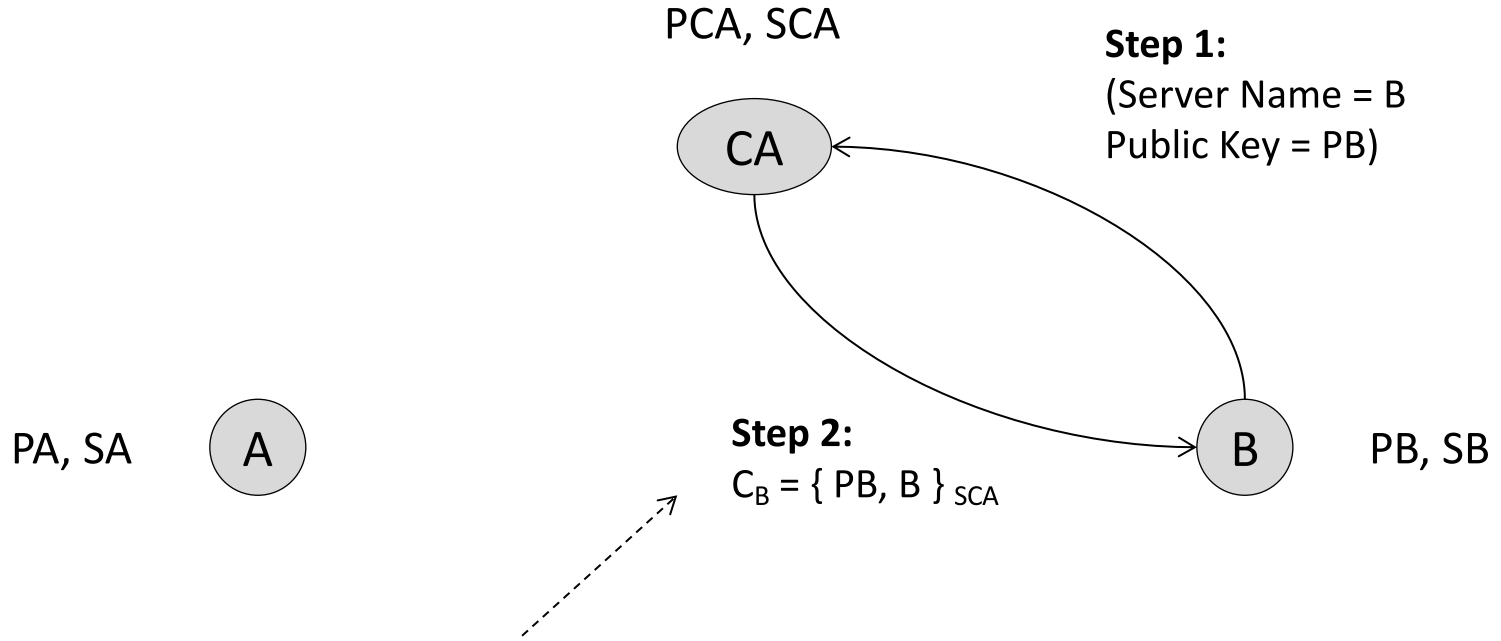


Three Potential Assurance Levels Between B and CA:

- *Low*: Attributable Email from B’s Server Admin to CA
- *Medium*: Out of Band Authentication of B’s Server Admin by CA
- *High*: In-Person Authentication of B’s Server Admin by CA

Week 8

Public Key Distribution – Certification Authority

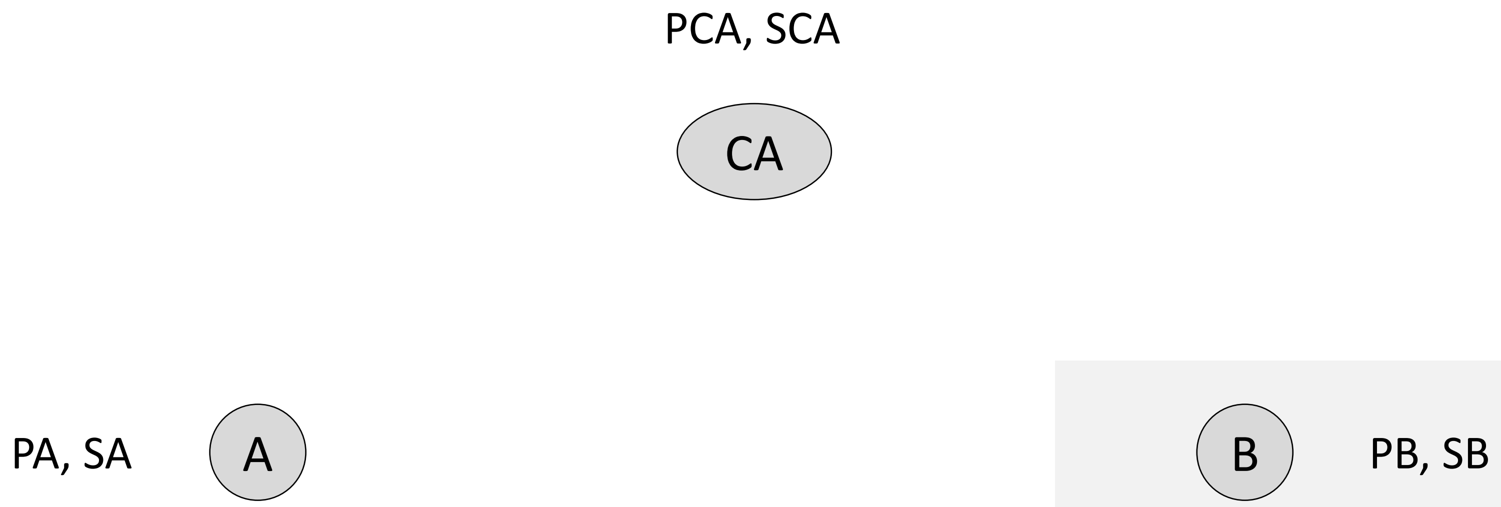


CA Sign's the Server B with Certificate C_B:

- *Certificate follows X.509 v3 Standard*
- *Certificate encrypted with CA's Private Key SCA*

Week 8

Public Key Distribution – Certification Authority

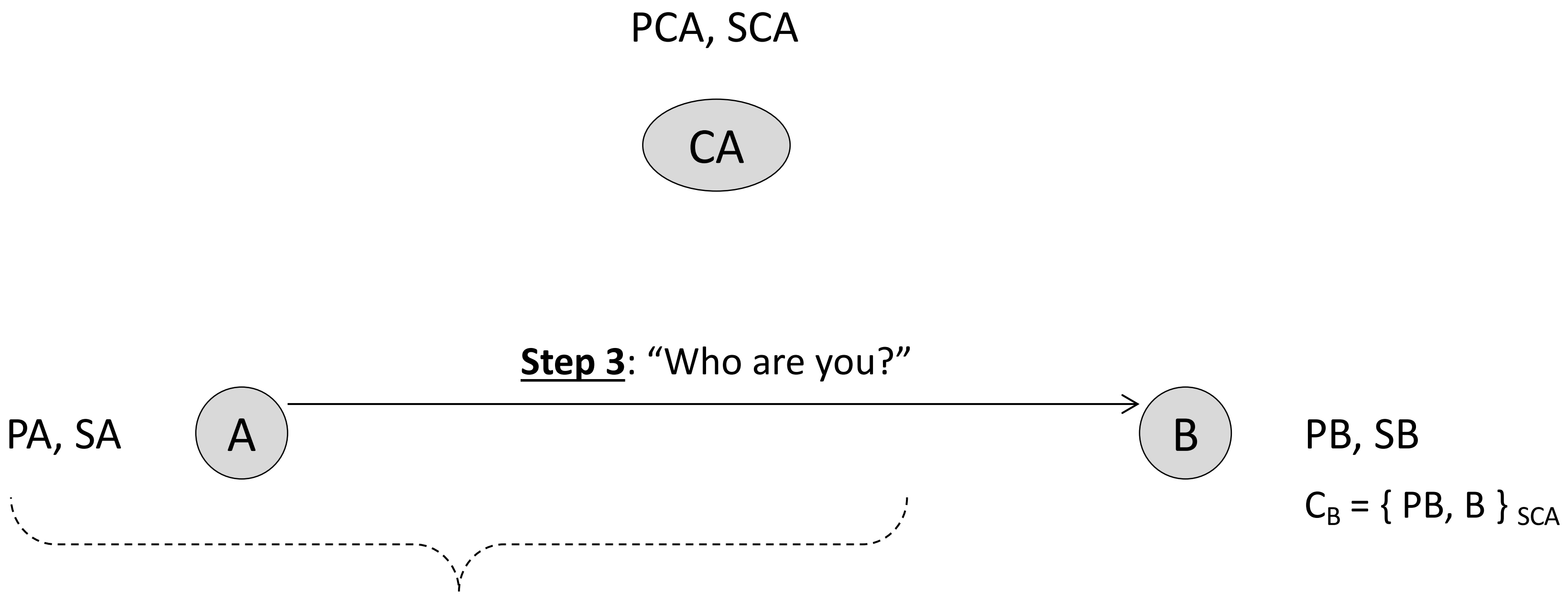


Server Now Signed With:

C_B = { PB, B } $_{SCA}$

Week 8

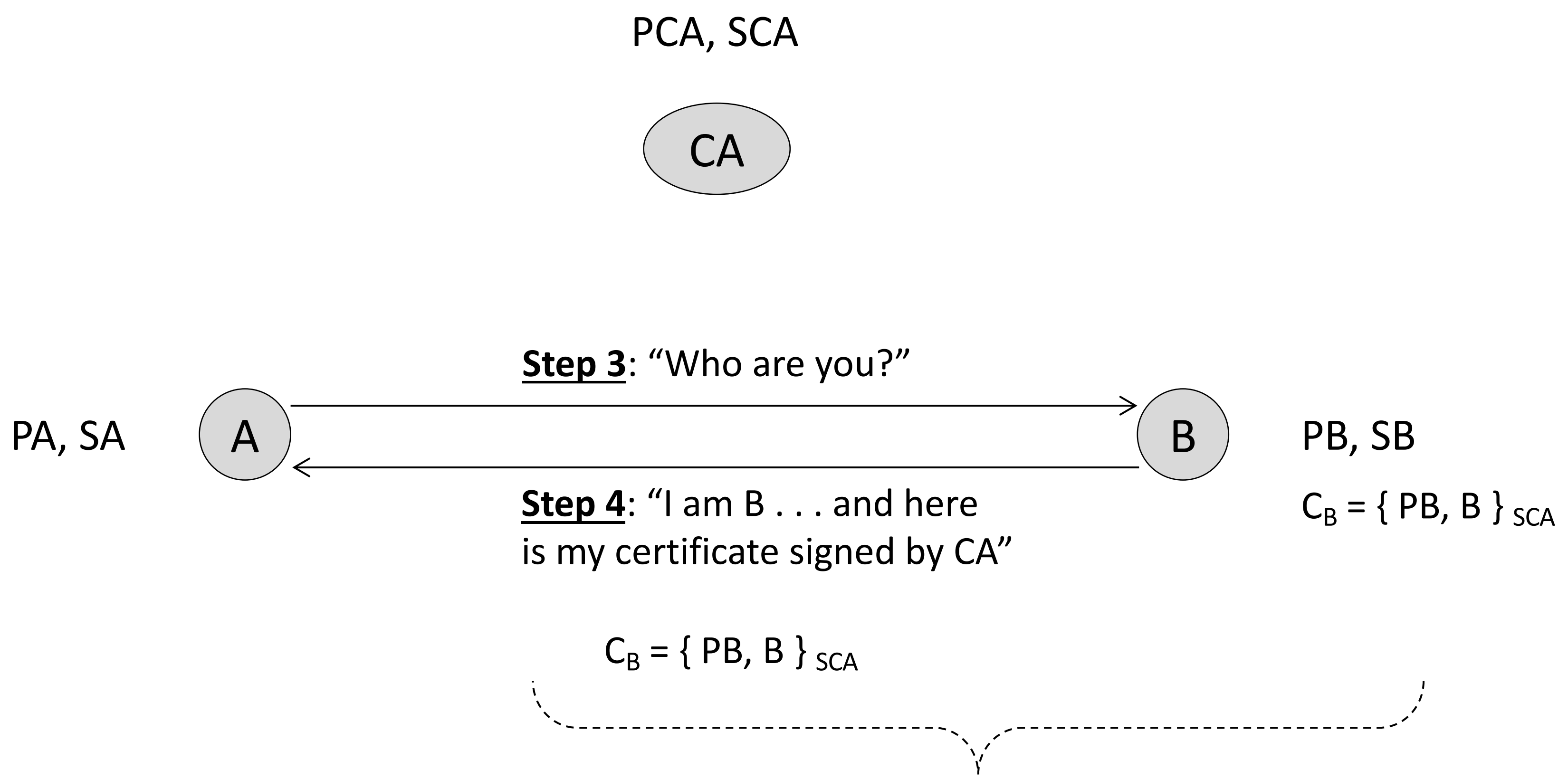
Public Key Distribution – Certification Authority



Server Authentication:
A has a browser and presumably wants to buy something on B's Website

Week 8

Public Key Distribution – Certification Authority



Server Response:
B send its certificate to A in order to authenticate and send PB for encryption

Public Key Distribution – Certification Authority

PCA, SCA

CA

Step 3: "Who are you?"

PA, SA

A

B

PB, SB

C_B = { PB, B } $_{SCA}$

Step 4: "I am B . . . and here is my certificate signed by CA"

C_B = { PB, B } $_{SCA}$

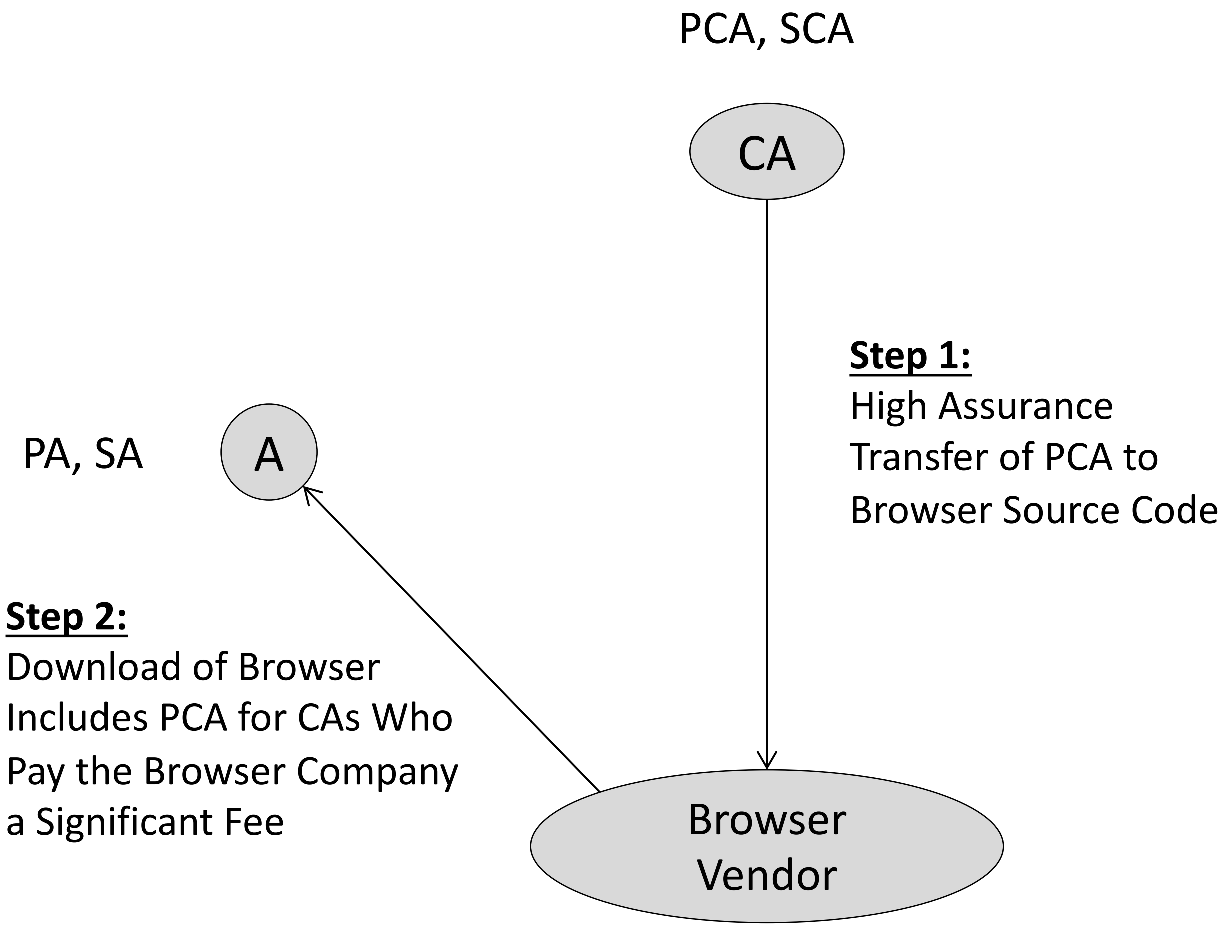
Step 5: "A needs the public key PCA of CA to decrypt:

{ { PB, B } $_{SCA}$ } $_{PCA}$ ➔ PB

A's Dilemma:
How does it get PCA into its browser to decrypt the certificate signed by CA?

How Did Netscape Solve the CA Public Key Problem?

Embedding Certificates in Browsers



Resulting Protocol: Secure Sockets Layer (SSL)

Marc Andreessen
Netscape Browser Founder and Internet Billionaire Shown in Mid-1990's

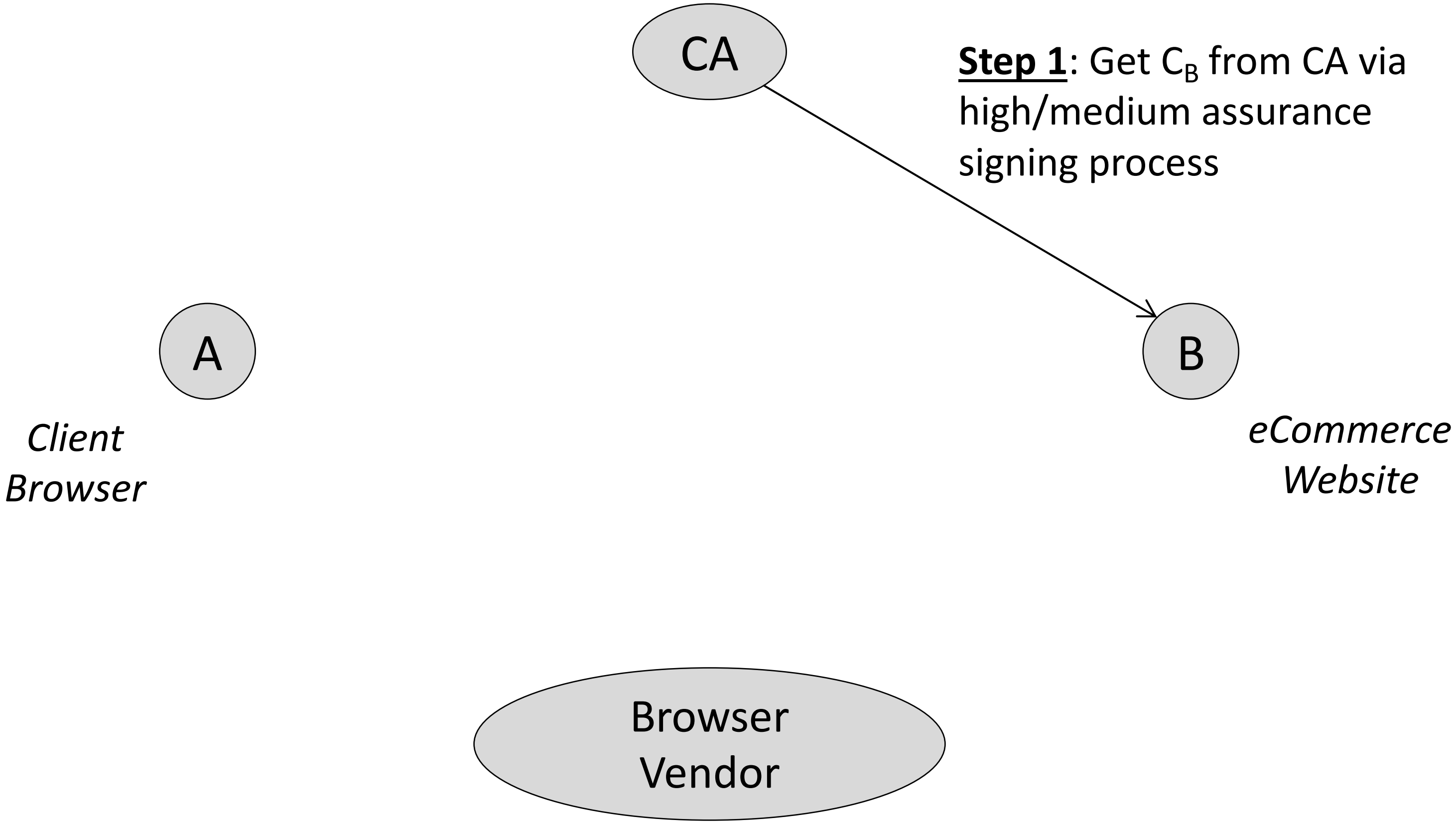
Week 8

Netscape's Historic IPO

Actual Scenario – Post IPO

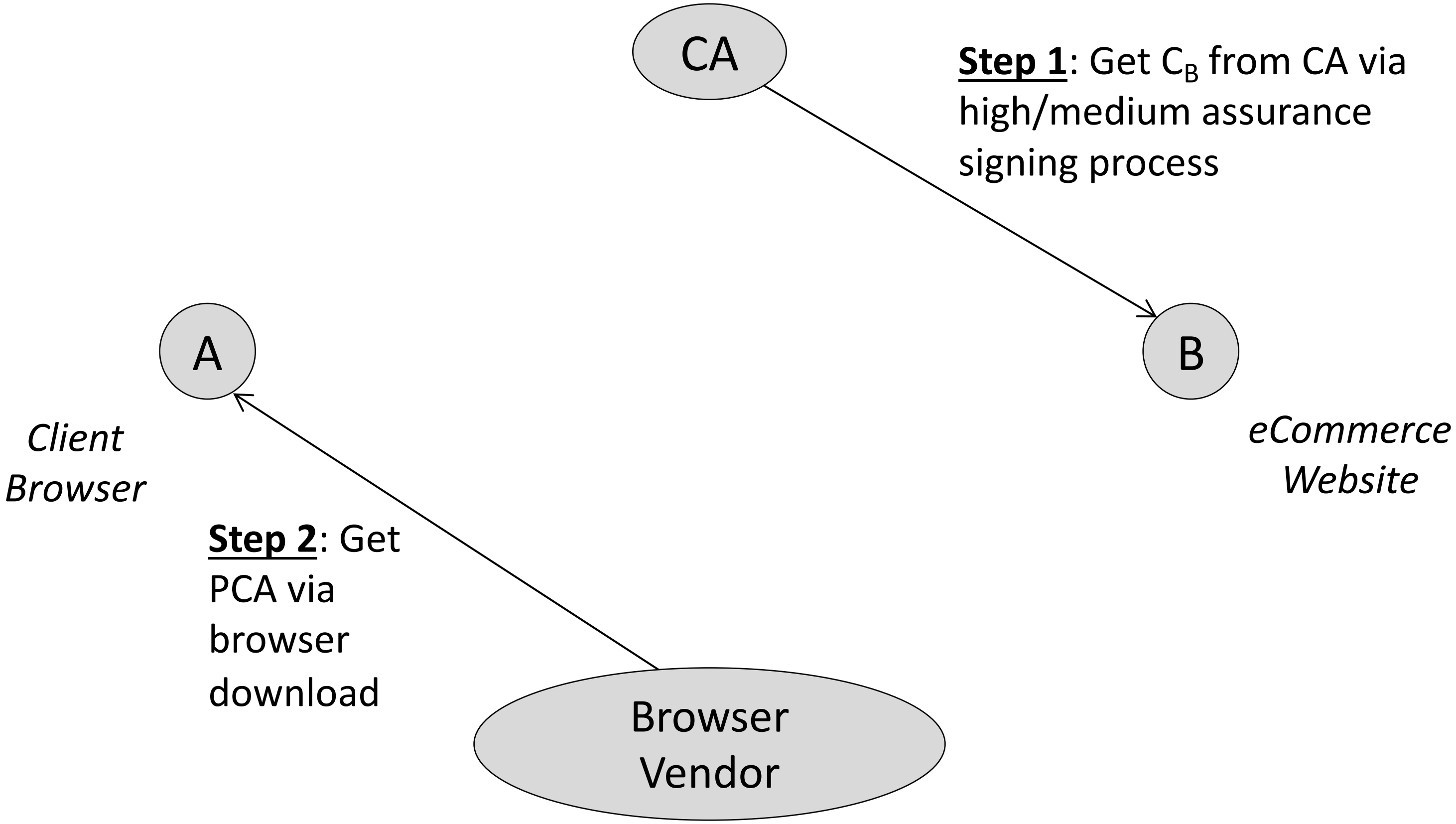
- Netscape shares opened at $28.
- By the end of the trading day, they were going for $75.
- The five-million-share IPO was oversubscribed by 100 million shares.
- Book Value of $16 million was transformed into market value of a billion dollar.

SSL PKI/CA – Secure eCommerce

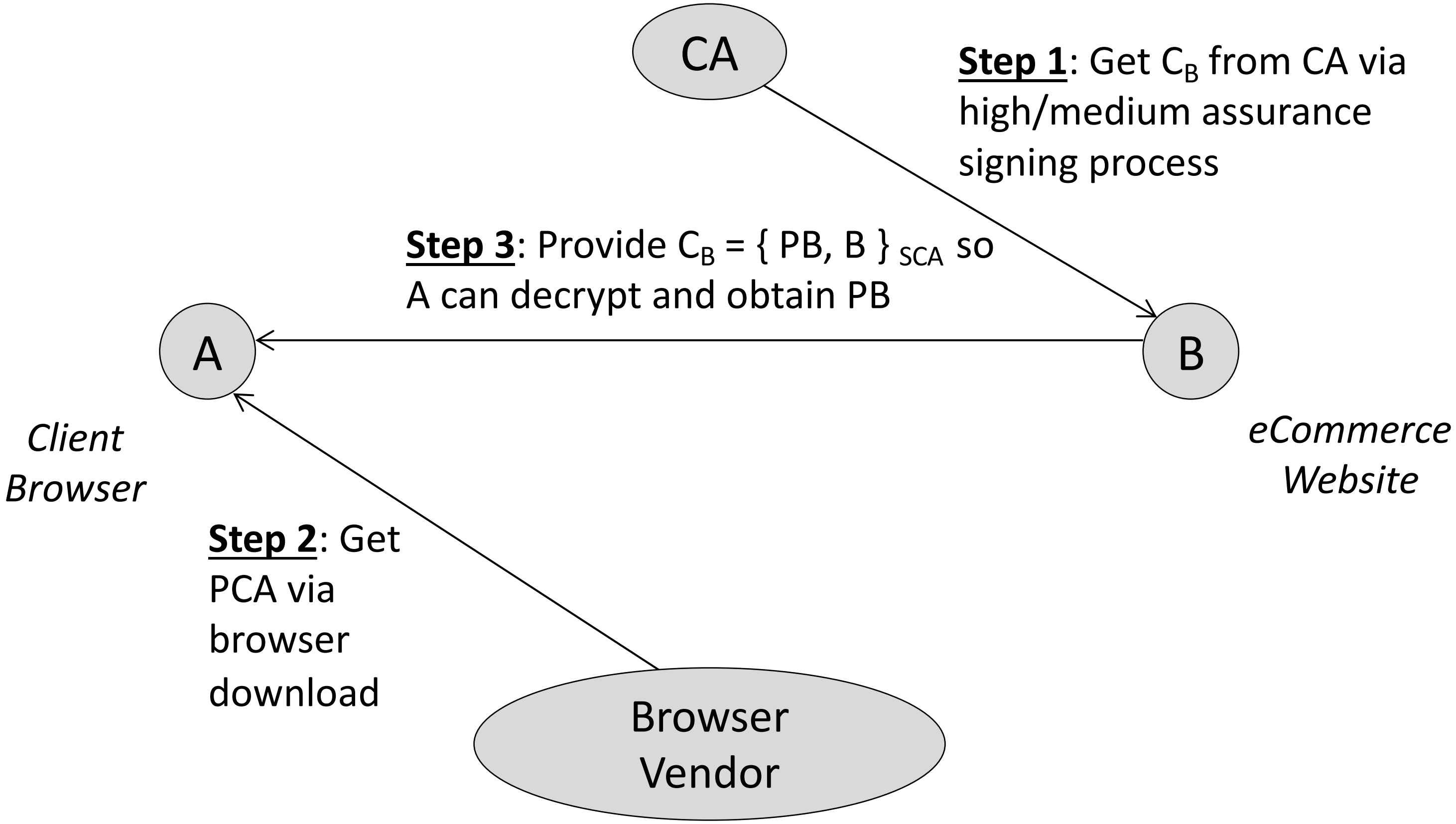


Week 8

SSL PKI/CA – Secure eCommerce

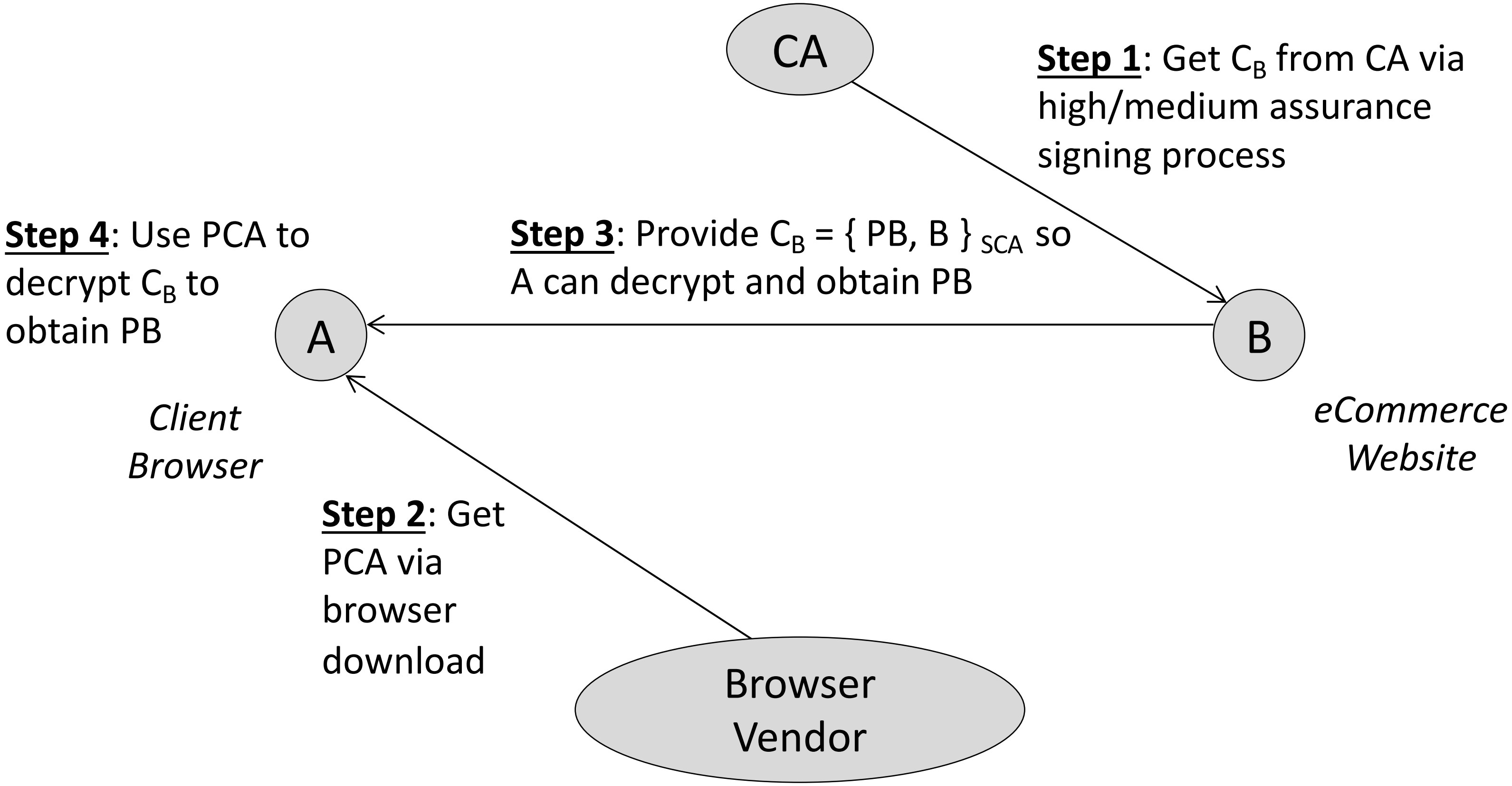


SSL PKI/CA – Secure eCommerce



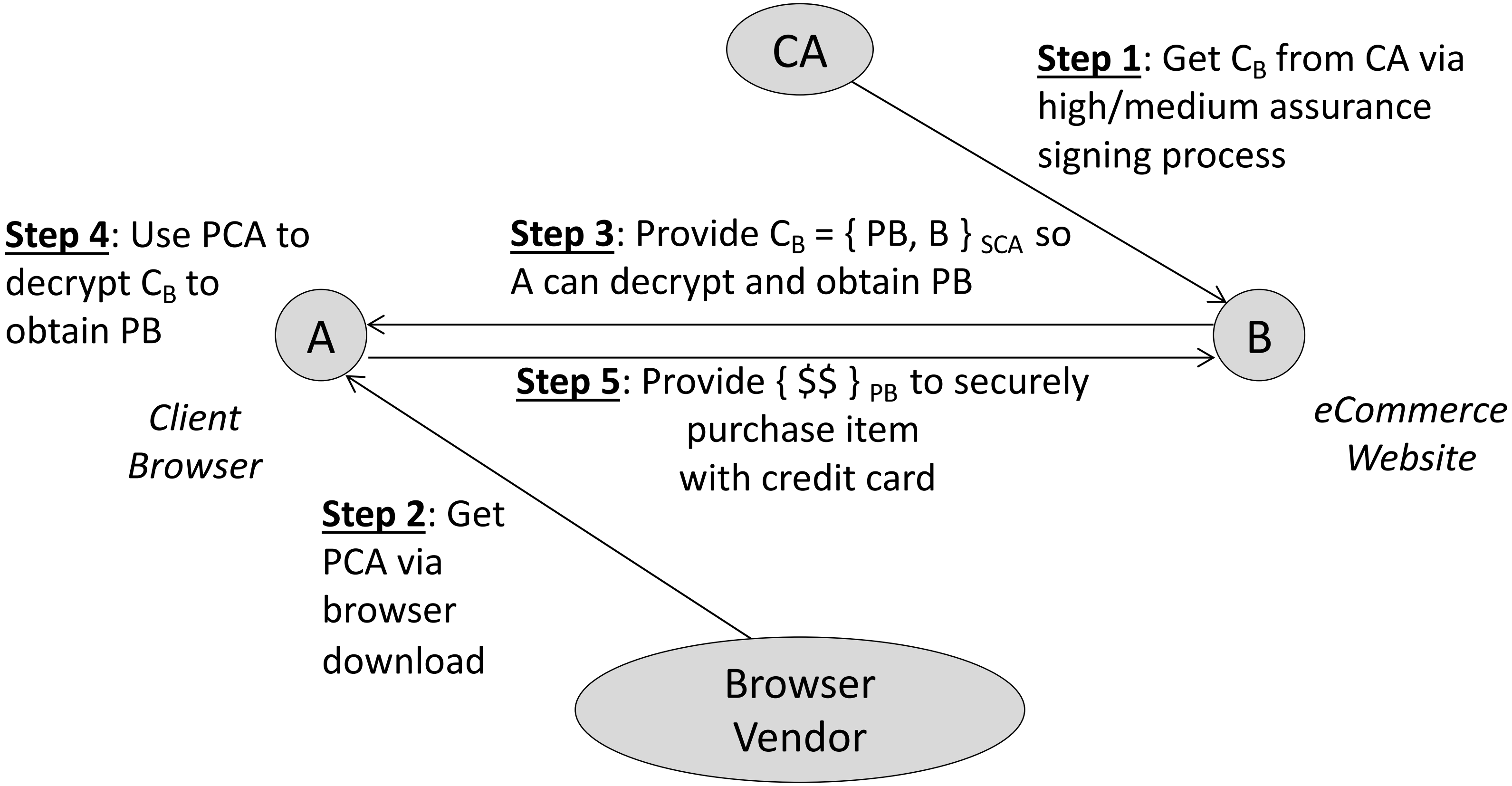
Week 8

SSL PKI/CA – Secure eCommerce



Week 8

SSL PKI/CA – Secure eCommerce

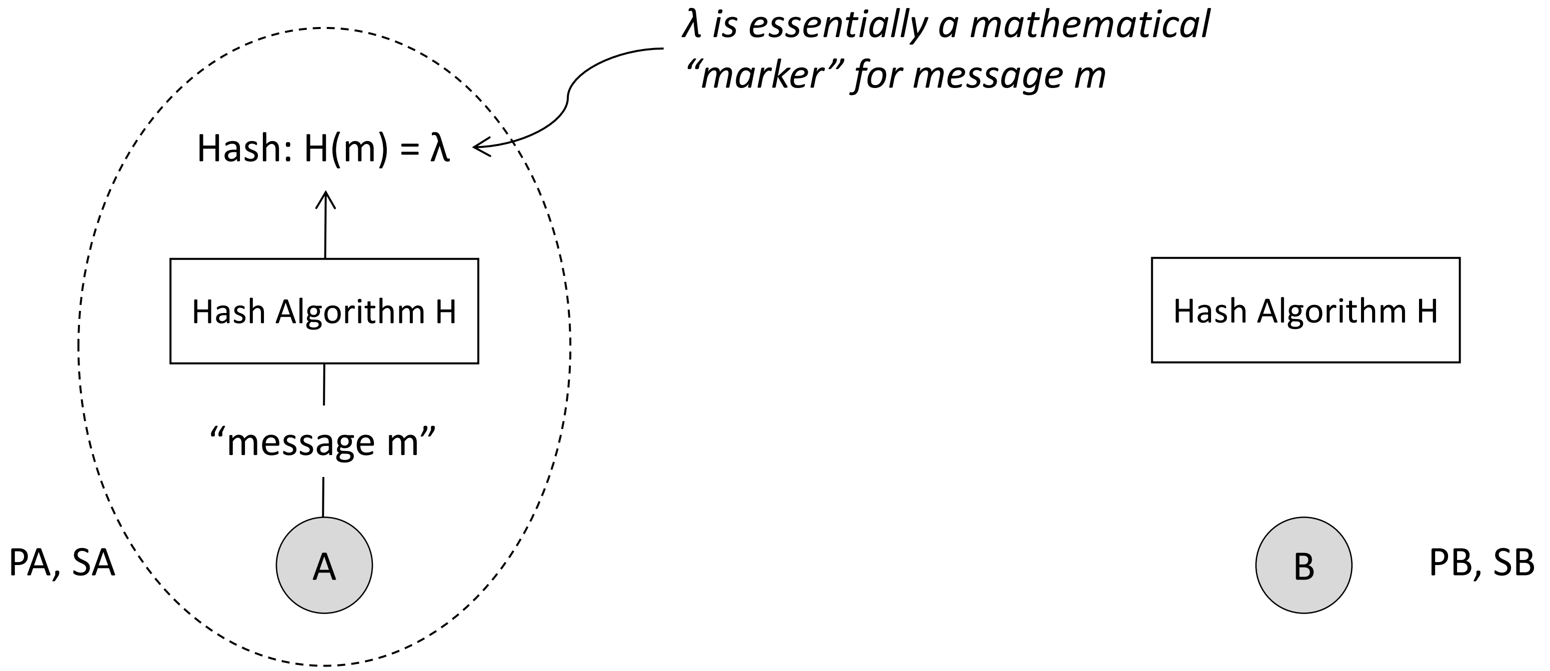


Week 8

How Does Hashing Work?

Week 8

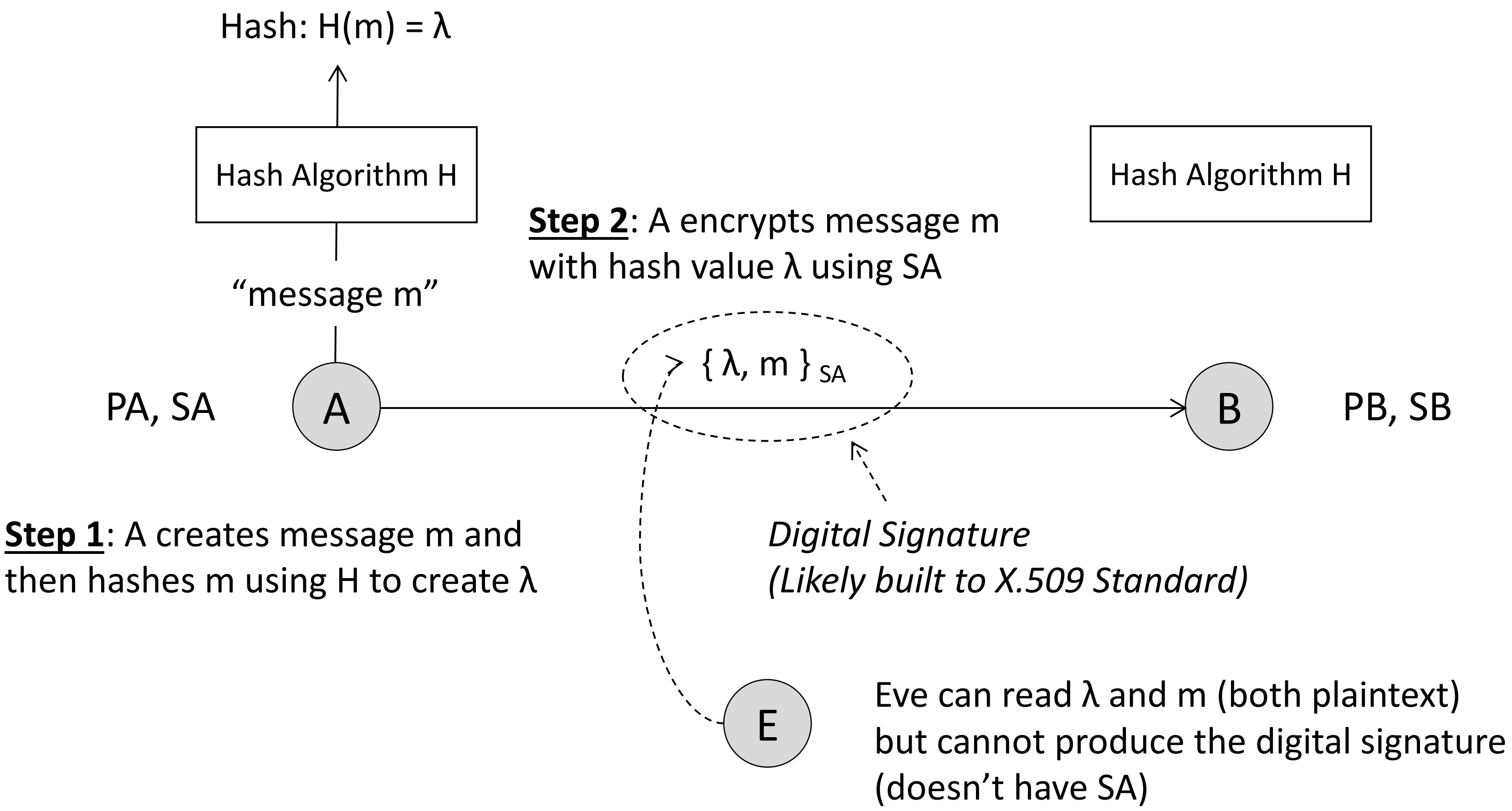
Hashing for Digital Signature



Step 1: A creates message m and then hashes m using H to create λ

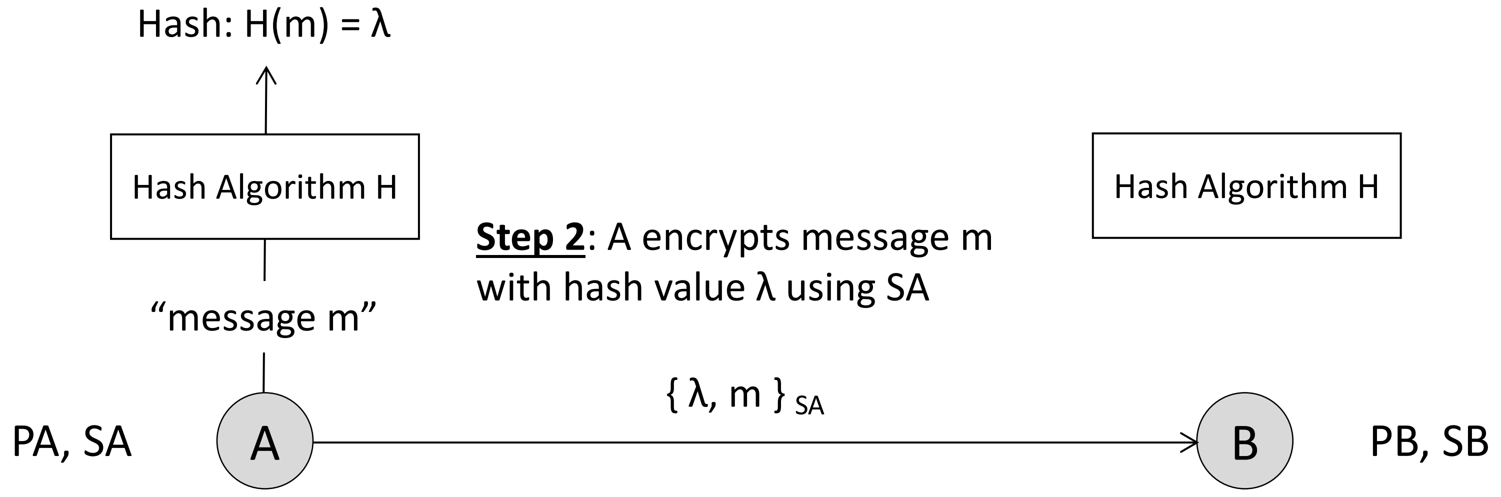
- *Hash Algorithm*: "Variable length input" (domain) to "fixed length output" (co-domain)
- *Hash Algorithm + Keys = Message Digest Algorithm*

Hashing for Digital Signature



Week 8

Hashing for Digital Signature



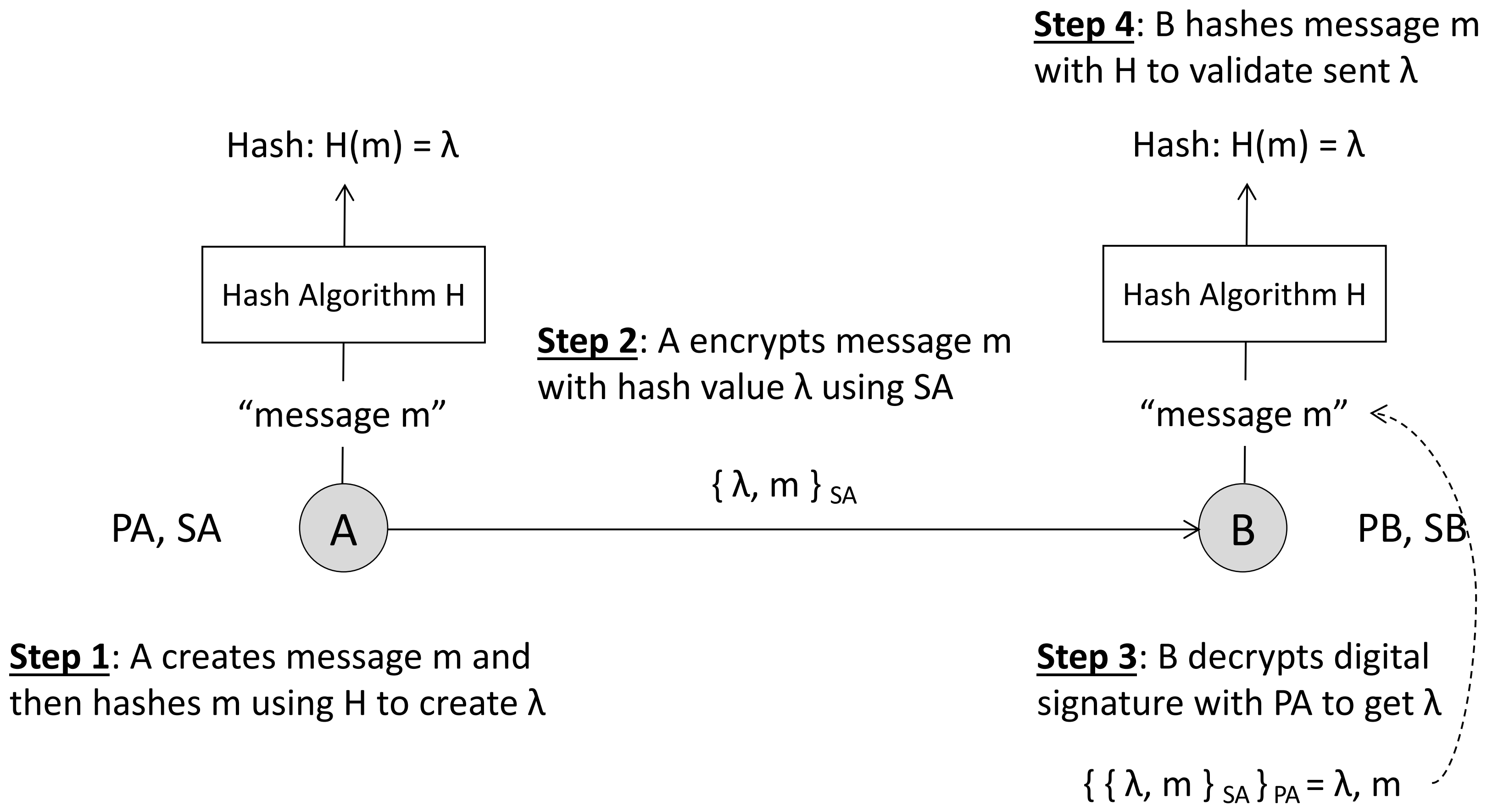
Step 1: A creates message m and then hashes m using H to create λ

Step 3: B decrypts digital signature with PA to get λ

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

Week 8

Hashing for Digital Signature

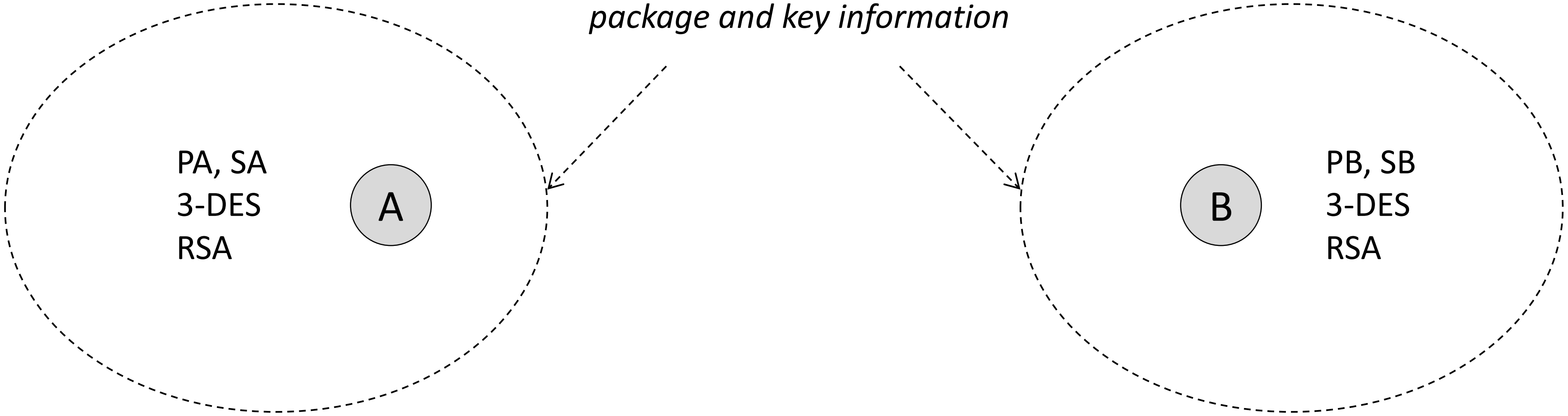


How is Email Secured?

Week 8

Secret Email

Sender and receiver must have the same email security package and key information



Week 8

Secret Email

Sender initiates the secure email send via key management and encryption tasks

PA, SA
3-DES
RSA

A

Step 1: Generate 3-DES key K for bulk encryption

Step 2: 3-DES encrypt message m using key K

$\{ m \}_K$

Step 3: RSA encrypt key K using PB

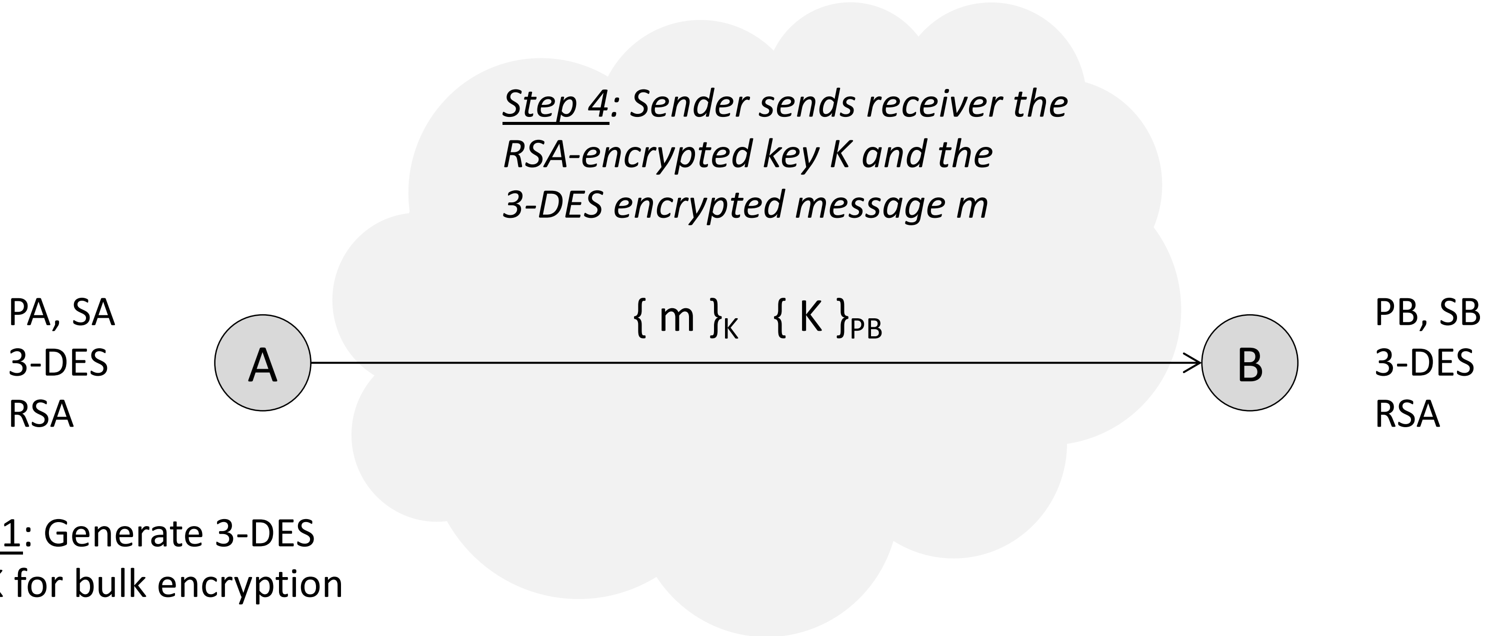
$\{ K \}_{PB}$

B

PB, SB
3-DES
RSA

Week 8

Secret Email



Step 1: Generate 3-DES key K for bulk encryption

Step 2: 3-DES encrypt message m using key K

$\{ m \}_K$

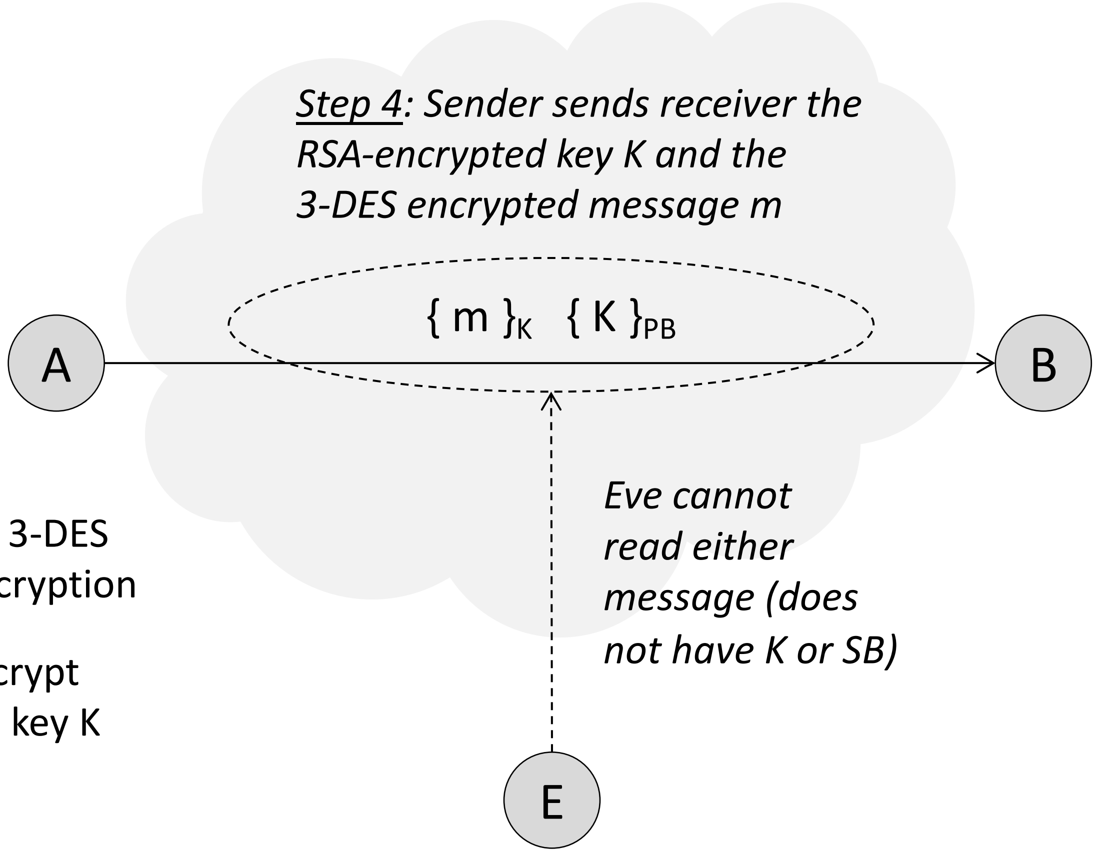
Step 3: RSA encrypt key K using PB

$\{ K \}_{PB}$

Week 8

Secret Email

PA, SA
3-DES
RSA



PB, SB
3-DES
RSA

PA, PB

Step 1: Generate 3-DES key K for bulk encryption

Step 2: 3-DES encrypt message m using key K

$\{ m \}_K$

Step 3: RSA encrypt key K using PB

$\{ K \}_{PB}$

Secret Email

<u>Step 4</u>: Sender sends receiver the RSA-encrypted key K and the 3-DES encrypted message m

$\{ m \}_K \quad \{ K \}_{PB}$

PA, SA
3-DES
RSA

A $\longrightarrow$ B

PB, SB
3-DES
RSA

<u>Step 1</u>: Generate 3-DES key K for bulk encryption

<u>Step 2</u>: 3-DES encrypt message m using key K

$\{ m \}_K$

<u>Step 3</u>: RSA encrypt key K using PB

$\{ K \}_{PB}$

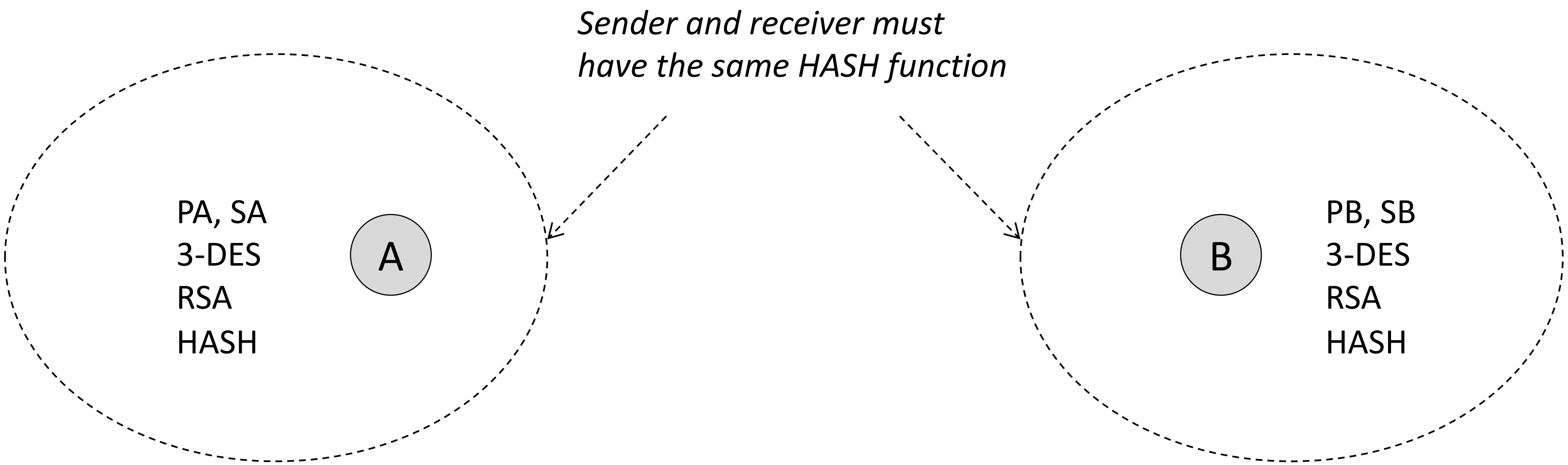
<u>Step 5</u>: Receiver decrypts the RSA-encrypted key with SB to get K and then decrypts the 3-DES encrypted message to get m

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

$\{ \{ m \}_K \}_K = m$

Week 8

Digitally Signed Email



Week 8

Digitally Signed Email

Sender initiates the signed email send via key management and encryption tasks

PA, SA
3-DES
RSA
HASH

A

B

PB, SB
3-DES
RSA
HASH

<u>Step 1</u>: Generate hash of message m using HASH

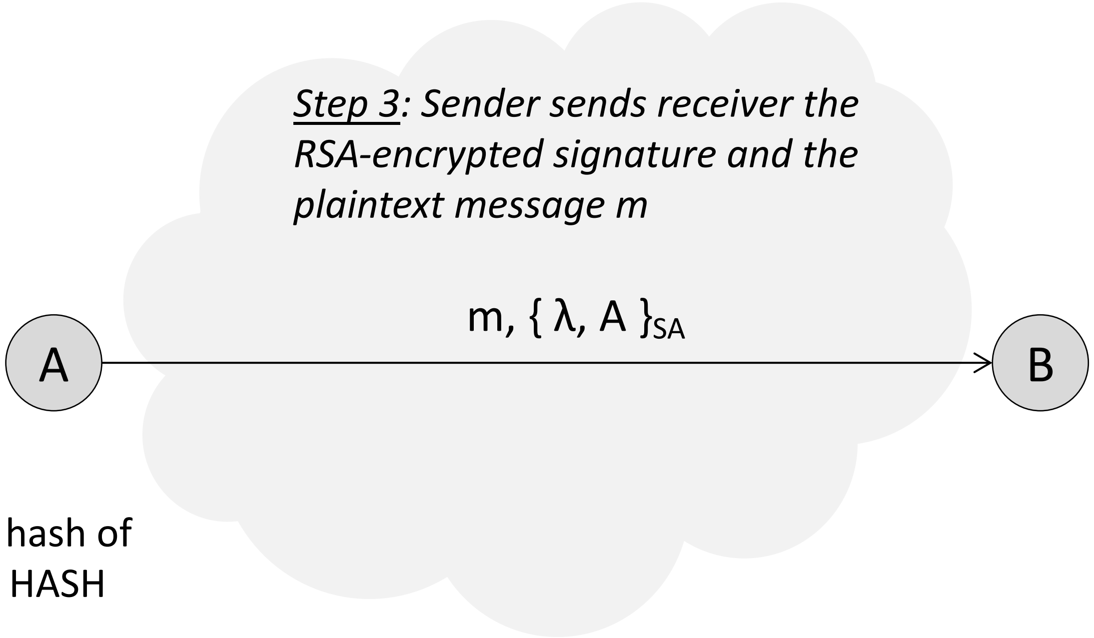
$$\text{HASH (m)} = \lambda$$

<u>Step 2</u>: RSA encrypt λ and A using SA to form digital signature

$$\{ \lambda, A \}_{SA}$$

Digitally Signed Email

PA, SA
3-DES
RSA
HASH



PB, SB
3-DES
RSA
HASH

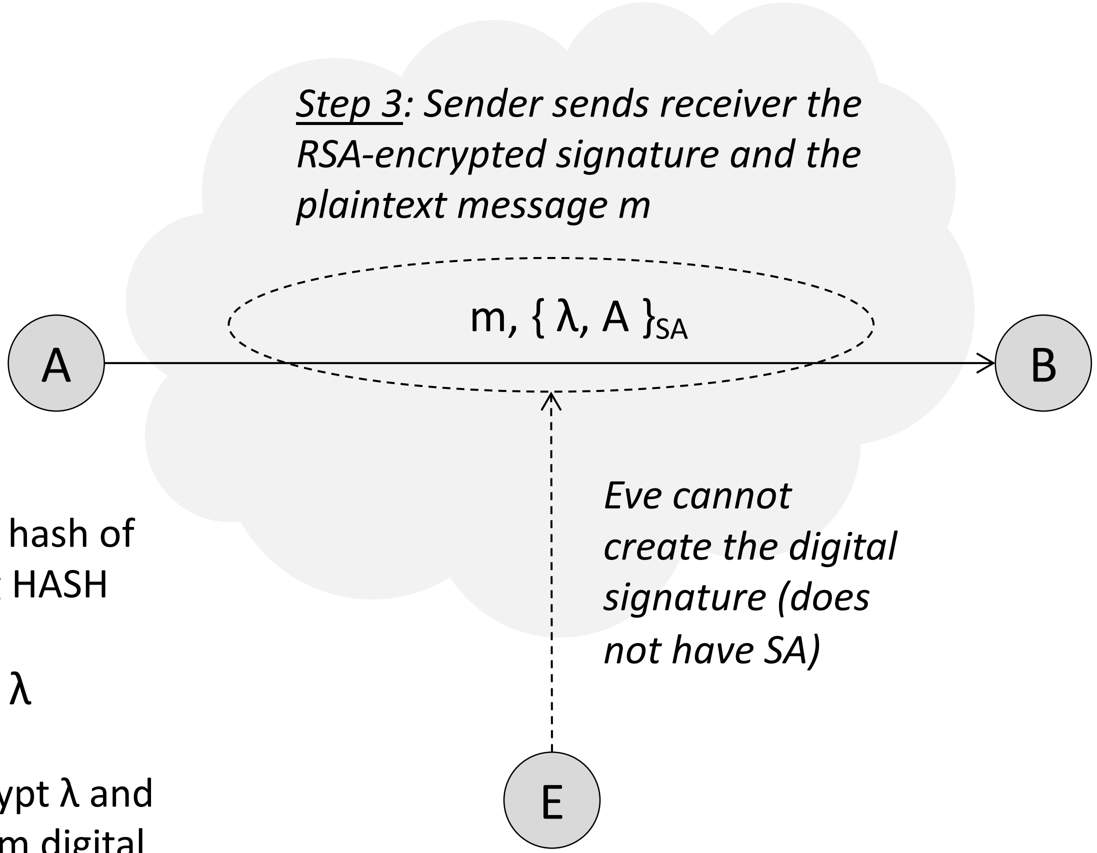
Step 1: Generate hash of message m using HASH

$$HASH(m) = \lambda$$

Step 3: RSA encrypt λ and A using SA to form digital signature

$$\{ \lambda, A \}_{SA}$$

Digitally Signed Email



PA, SA
3-DES
RSA
HASH

PB, SB
3-DES
RSA
HASH

PA, PB

Step 1: Generate hash of message m using HASH

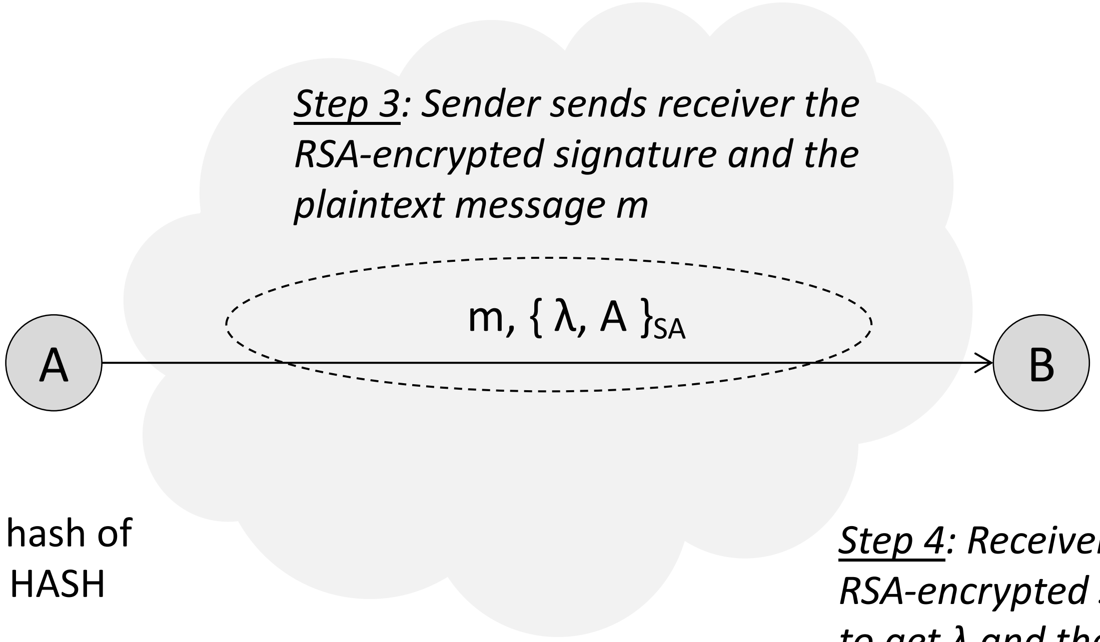
$$\text{HASH}(m) = \lambda$$

Step 3: RSA encrypt λ and A using SA to form digital signature

$$\{ \lambda, A \}_{SA}$$

Digitally Signed Email

PA, SA
3-DES
RSA
HASH



PB, SB
3-DES
RSA
HASH

Step 1: Generate hash of message m using HASH

$$HASH\ (m) = \lambda$$

Step 3: RSA encrypt λ and A using SA to form digital signature

$$\{ \lambda, A \}_{SA}$$

Step 4: Receiver decrypts the RSA-encrypted signature with SA to get λ and then locally computes HASH (m) to check validity

$$\{ \{ \lambda, A \}_{SA} \}_{PA} = \lambda, A$$

$$HASH\ (m) = \lambda$$

How Might Virtual Banking be Secured?

Week 8

Banking Security

"Wants to buy Teddy Bear On-line from M for $10.00"

Purchaser
P

PP, SP,
PM, PB

"Selling Teddy Bears On-line for $10.00"

Merchant
M

PM, SM
PP, PB

"Maintains Bank Accounts for P and M with Real Money Balances"

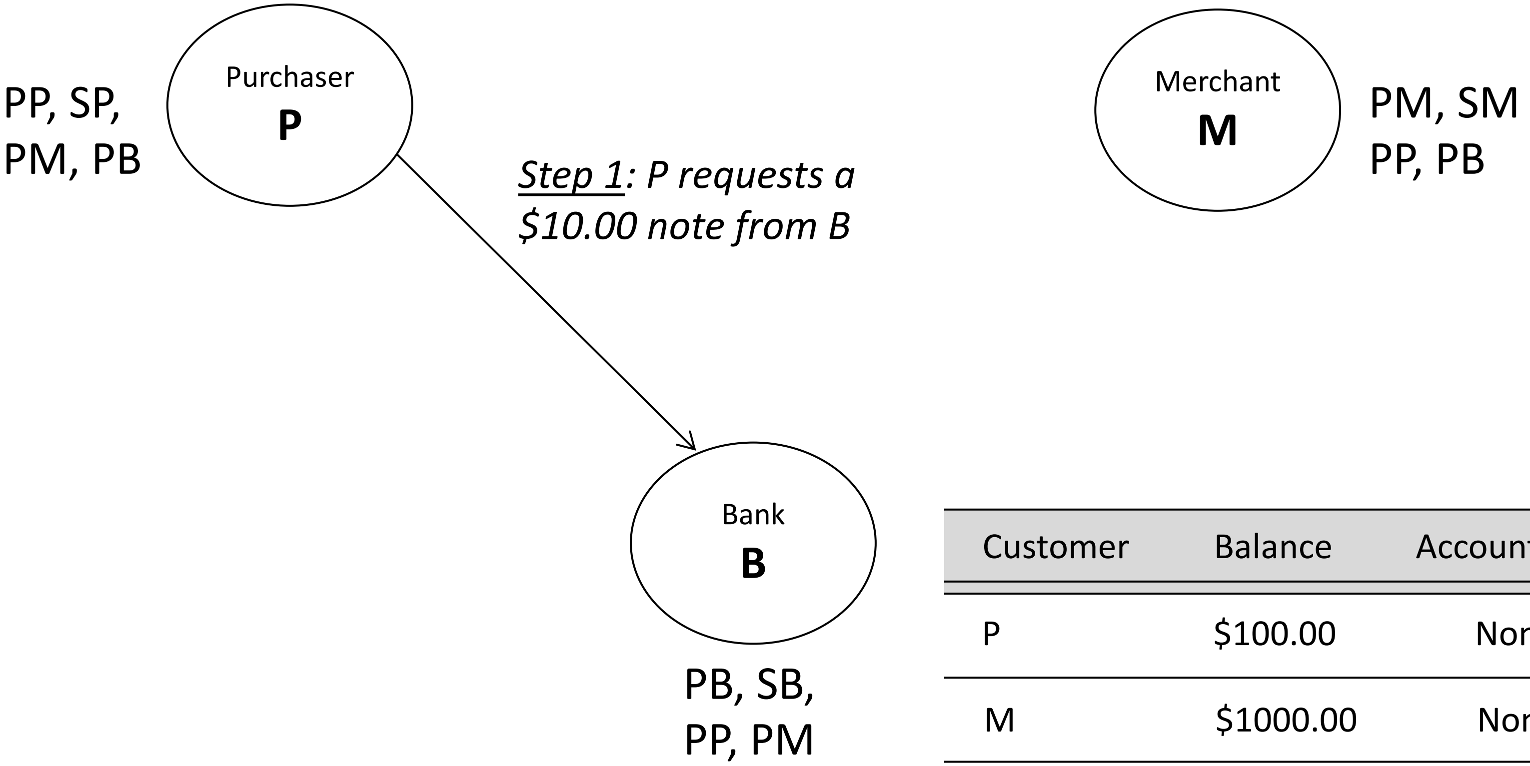
Bank
B

PB, SB,
PP, PM

Customer	Balance	Account Notes
P	$100.00	None
M	$1000.00	None

Week 8

Banking Security



Customer	Balance	Account Notes
P	$100.00	None
M	$1000.00	None

Week 8

Banking Security

PP, SP,
PM, PB

Purchaser
P

Merchant
M

PM, SM
PP, PB

Step 1: P requests a $10.00 note from B

Step 2: B reduces P's balance by $10.00

Step 3: B creates and Sends a $10.00 note to P

{ { $10.00, Serial Number 2468} SB } PP

Follows some standard bank note format with a random, unique serial number

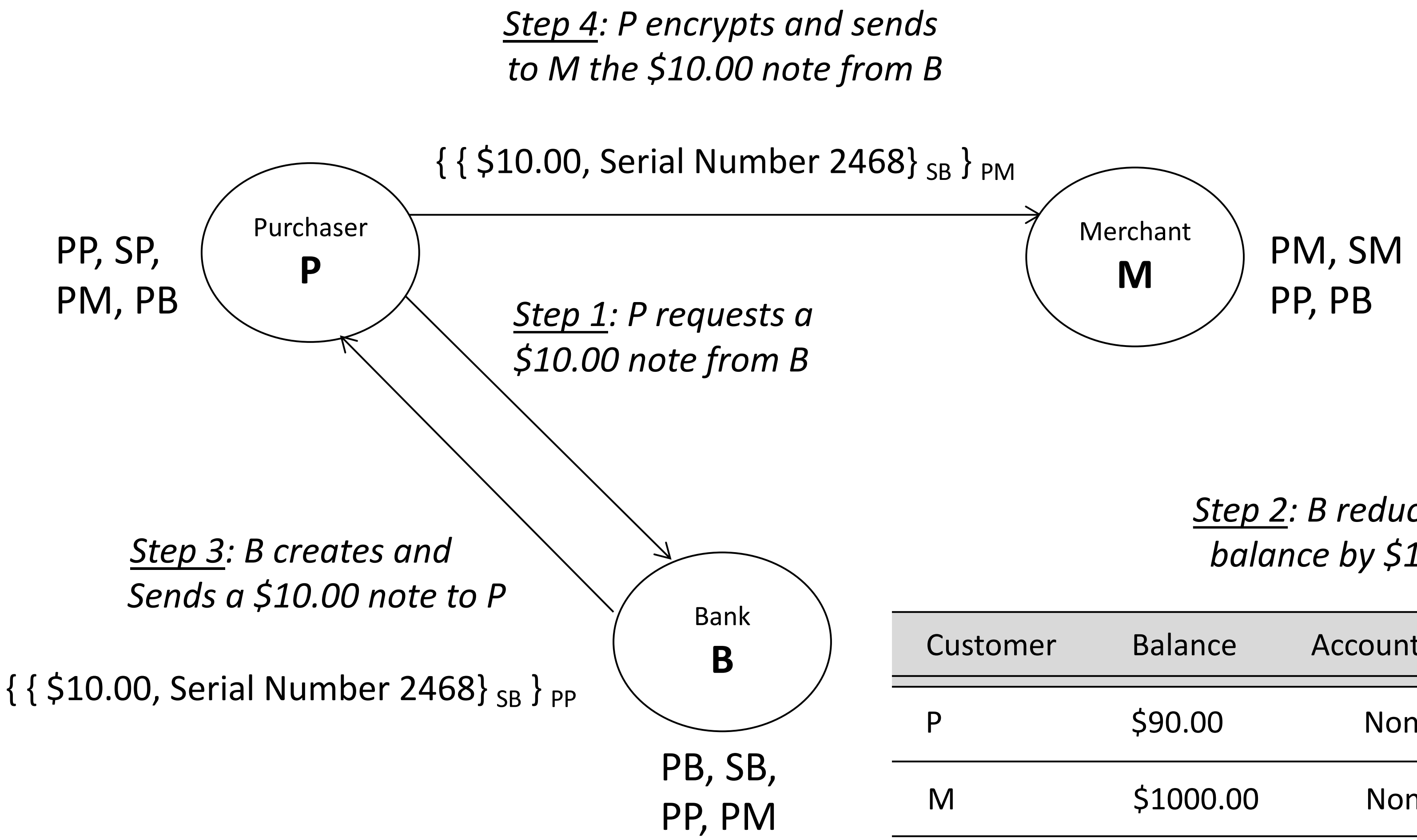
Bank
B

PB, SB,
PP, PM

Customer	Balance	Account Notes
P	$90.00	None
M	$1000.00	None

Week 8

Banking Security

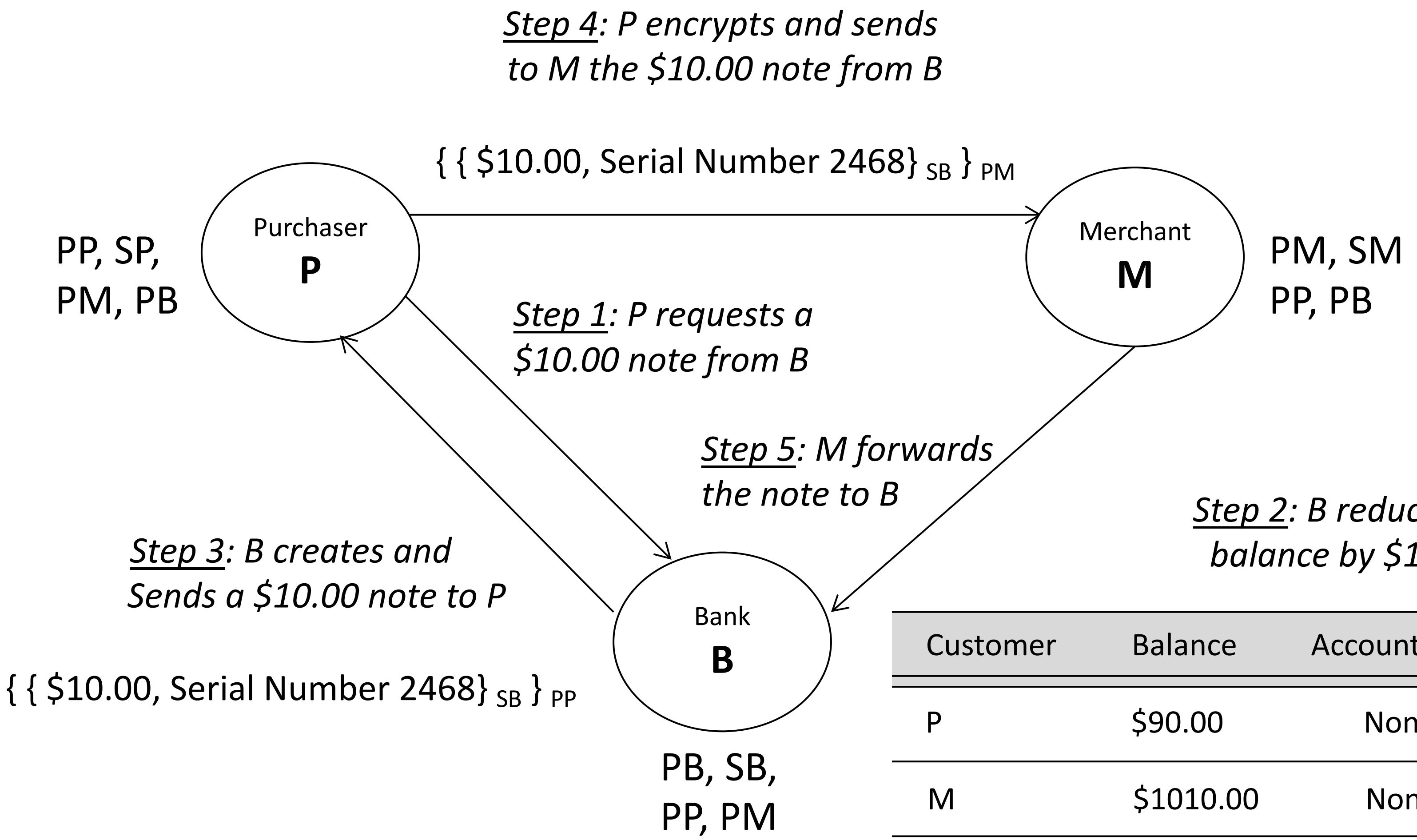


Step 2: B reduces P's balance by $10.00

Customer	Balance	Account Notes
P	$90.00	None
M	$1000.00	None

Week 8

Banking Security



Step 2: B reduces P's balance by $10.00

Customer	Balance	Account Notes
P	$90.00	None
M	$1010.00	None

Step 6: B decrypts, checks serial number, and credits M's account

What is a Blinding Protocol?

Week 8

Chaum’s Blinding Protocol: Goal

Alice

Bob

<u>Step 1</u>: ”Send Bob an encrypted secret number without necessary key information for Bob to decrypt.”

<u>Step 2</u>: ”Attest to the validity of the encrypted secret number without decrypting or reading it (i.e., fully blind attestation)”

<u>Step 3</u>: ”Send back to Alice a digitally signed attestation of the validity of the secret number.”

David Chaum
University of California at Berkeley
Founder DigiCash (defunct)

Week 8

Chaum's Blinding Protocol: Goal

Alice (Client)	Network	Bob (Server)
1. CREATE $\{\texttt{Serial \# 145167, \$2.00}\}_{K1}$		
	2. SEND $\{\texttt{Serial \# 145167, \$2.00}\}_{K1}$ →	
		3. SIGN $\{\{\texttt{Serial \# 145167, \$2.00}\}_{K1}\}_{SB}$
	4. RESPOND ← $\{\{\texttt{Serial \# 145167, \$2.00}\}_{K1}\}_{SB}$	
4. SIGNED CERTIFICATE $\{\{\texttt{Serial \# 145167, \$2.00}\}_{K1}\}_{SB}$		

Verifiable with K1 and PB

Chaum's Blinding Protocol: Implementation

Alice (Client) | **Network** | **Bob (Server)**

1. CREATE 1000 NOTES

$\{\text{Serial \# 145167, \$2.00}\}_{K1}$

$\{\text{Serial \# 246600, \$2.00}\}_{K2}$

...

$\{\text{Serial \# 938012, \$2.00}\}_{K1000}$

2. SEND 1000 NOTES
(All encrypted with 1000 different keys)

3. REQUEST RANDOM 999 KEYS
All 999 Keys except K_n

4. SEND RANDOM 999 KEYS
All 999 Keys except K_n

5. DECRYPT AND CHECK RANDOM 999 MESSAGES

$\{\{\text{Serial \# 145167, \$2.00}\}_{K1}\}_{K1}$

$\{\{\text{Serial \# 246600, \$2.00}\}_{K2}\}_{K2}$

...

$\{\{\text{Serial \# 938012, \$2.00}\}_{K1000}\}_{K1000}$

6. SIGN and SEND nth MESSAGE WITH KEY Kn

$\{\{\text{Serial \# 119975, \$2.00}\}_{Kn}\}_{SB}$

Verifiable with Kn and PB

7. SIGNED CERTIFICATE FROM BOB

$\{\{\text{Serial \# 119975, \$2.00}\}_{Kn}\}_{SB}$