An Introduction to Cyber Security – CS 573

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

Week 5

Required Week Five Readings

1. "Password Authentication with Insecure Communication," Leslie Lamport https://lamport.azurewebsites.net/pubs/password.pdf

2. Chapters 12 through 16: *From CIA to APT: An Introduction to Cyber Security*, E. Amoroso & M. Amoroso

Twitter: @hashtag_cyber
LinkedIn: Edward Amoroso

Week 5: Authentication Protocols

How Do Nation States Cryptanalyze Encrypted Data? (Warm-Up Topic)

Week 5

Three Methods for Cryptanalysis

- **Ciphertext Only**
  - Cryptanalyst only has encrypted text
  - No hints or codebooks

Week 5

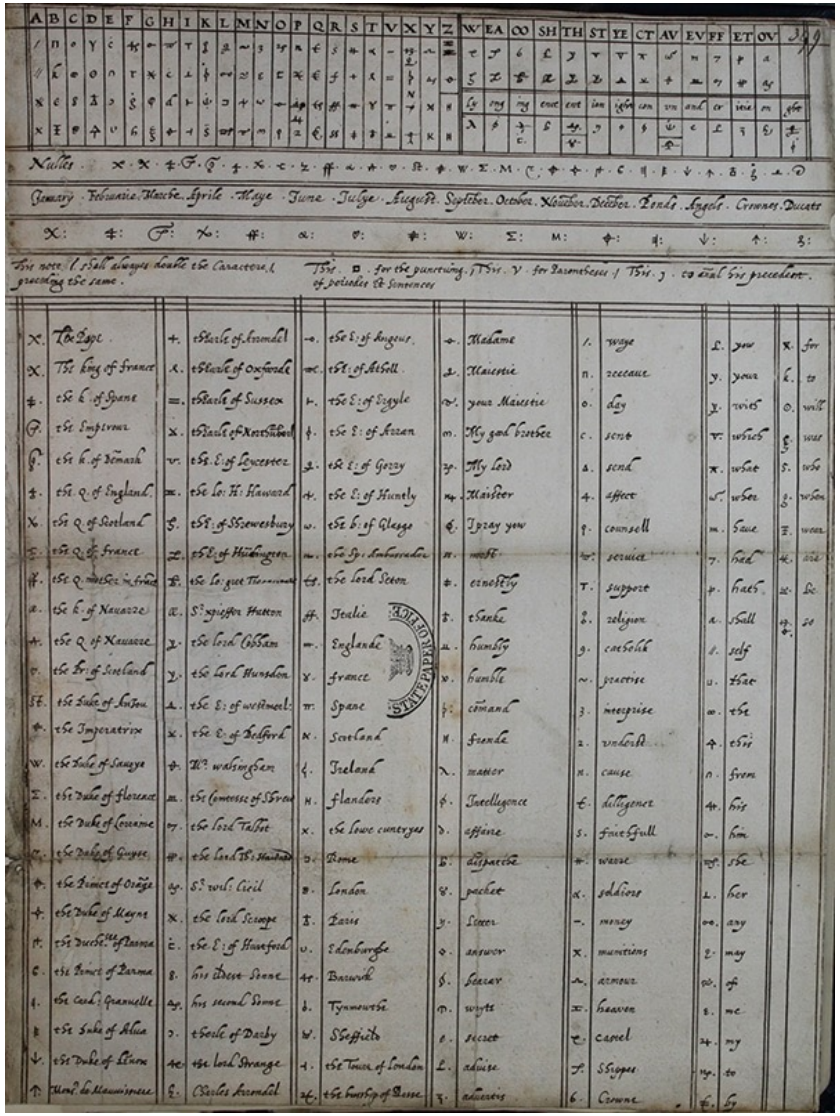
Three Methods for Cryptanalysis

- **Ciphertext Only**
  - Cryptanalyst only has encrypted text
  - No hints or codebooks
- **Known Plaintext**
  - Cryptanalyst observes hints (no control)
  - Tiny codebook examples can be developed

Week 5

Three Methods for Cryptanalysis

- **Ciphertext Only**
  - Cryptanalyst only has encrypted text
  - No hints or codebooks
- **Known Plaintext**
  - Cryptanalyst observes hints (no control)
  - Tiny codebook examples can be developed
- **Chosen Plaintext**
  - Cryptanalyst has the encryption function
  - Codebook can be developed



Week 5

Three Methods for Cryptanalysis

- **Ciphertext Only**
  - Cryptanalyst only has encrypted text
  - No hints or codebooks
- **Known Plaintext**
  - Cryptanalyst observes hints (no control)
  - Tiny codebook examples can be developed
- **Chosen Plaintext**
  - Cryptanalyst has the encryption function
  - Codebook can be developed

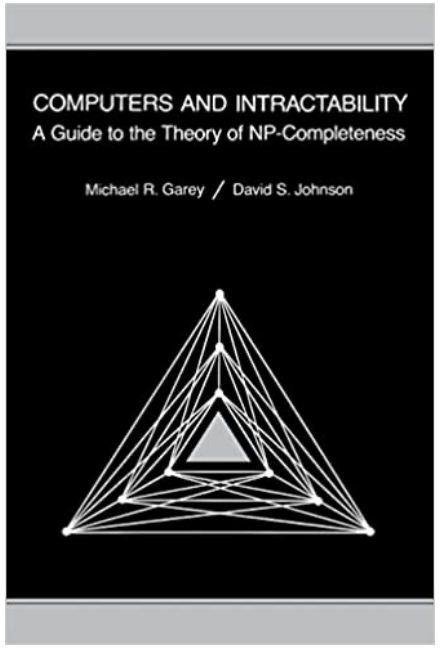
Two requirements protect encrypted text:

1. The encryption function must be cryptographically hard
2. The cleartext and ciphertext domains must be huge

Two Implications:

You must try every possible case to find the encryption function

The number of possible cases cannot feasibly be covered



What is Authentication?

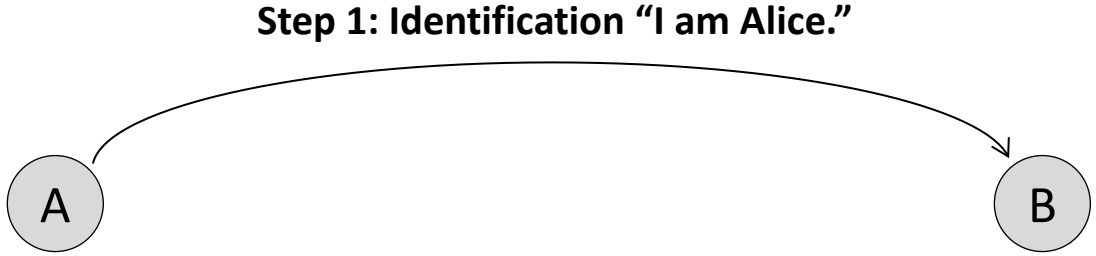

Week 5

Week 5

Authentication:
Process of validating a reported identity.

Week 5

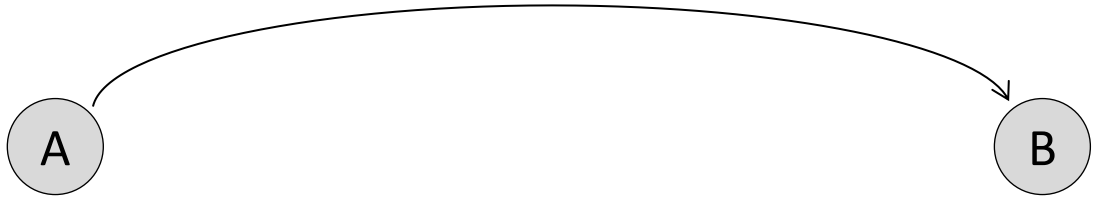
One-Factor Authentication Schema



Week 5

One-Factor Authentication Schema

Step 1: Identification “I am Alice.”

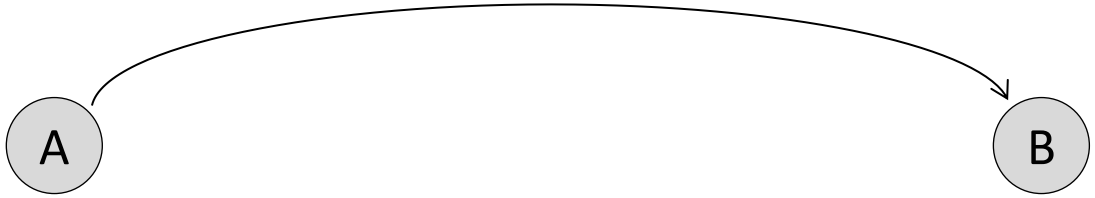


Server Validates Client: “Client Authentication”

Week 5

One-Factor Authentication Schema

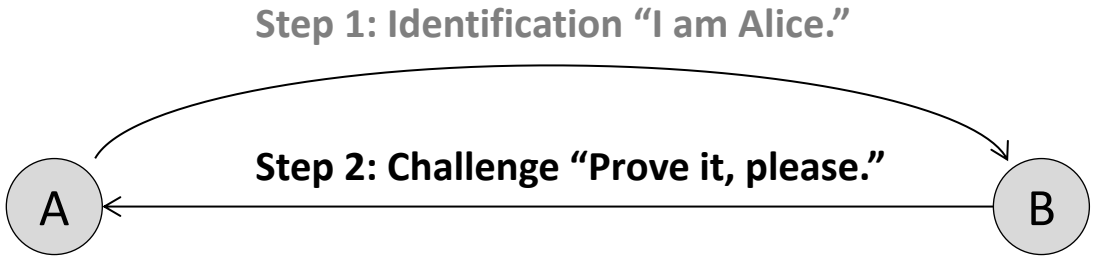
Step 1: Identification “I am Alice.”



Client Validates Server: “Server Authentication”

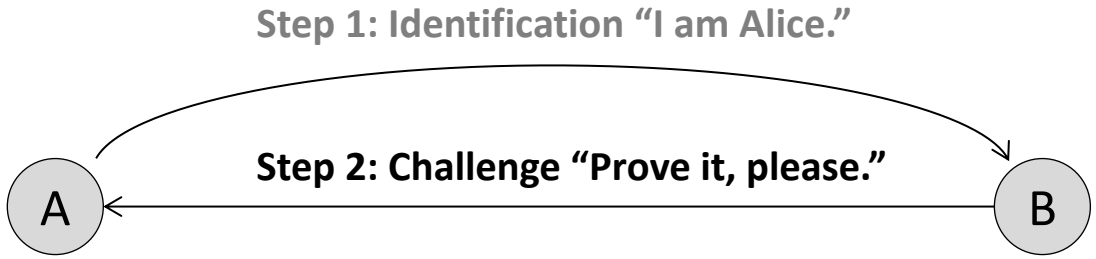
Week 5

One-Factor Authentication Schema



Week 5

One-Factor Authentication Schema

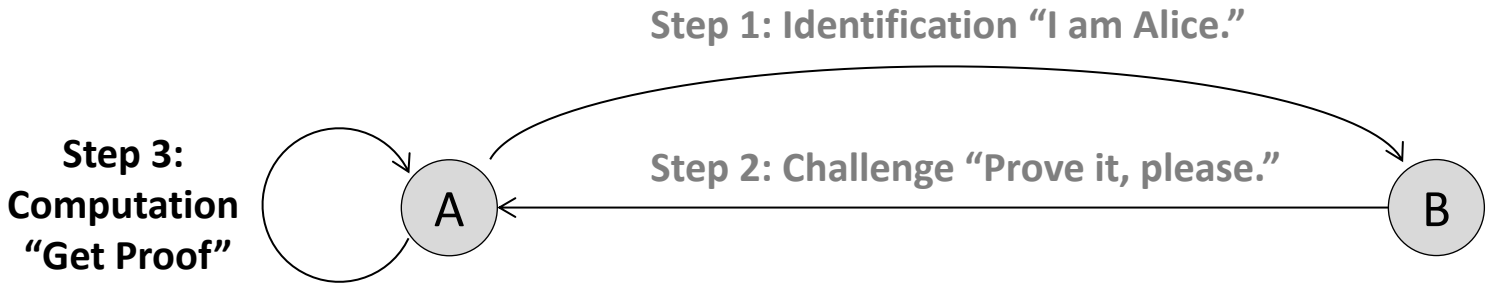


Challenge includes tangible domain value – possible “known plaintext” attacks

Challenge includes no tangible domain value – likely to restrict to “ciphertext–only attacks”

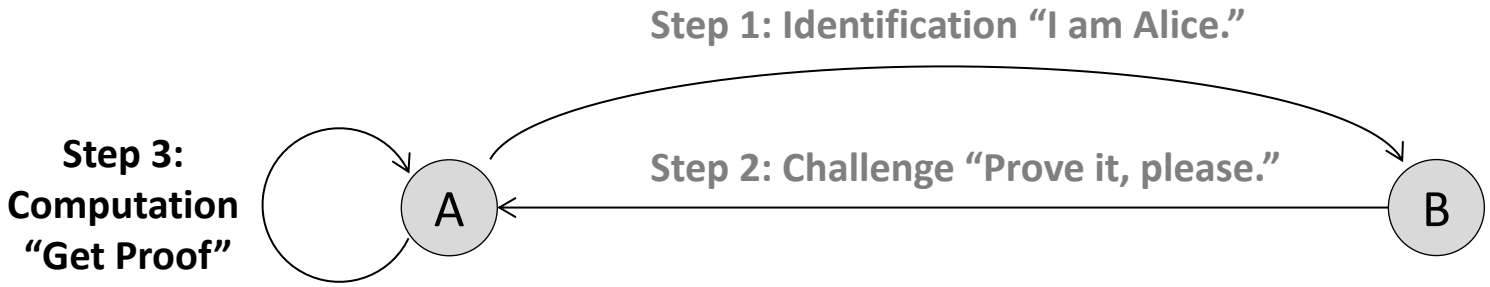
Week 5

One-Factor Authentication Schema



Week 5

One-Factor Authentication Schema

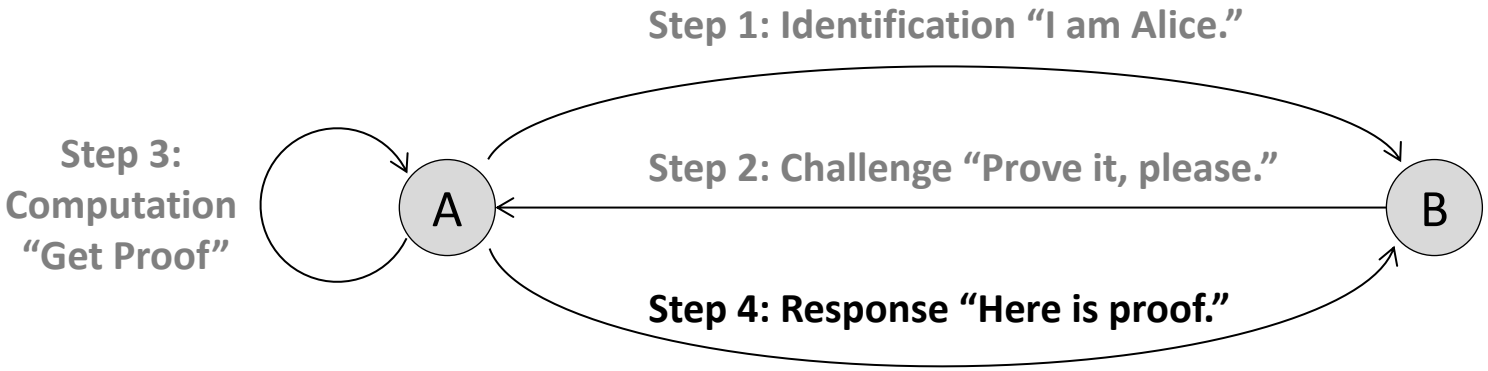


Computation might involve simple look-up/locate process (e.g., passwords)

Computation might be more deliberate mathematical operation on domain value

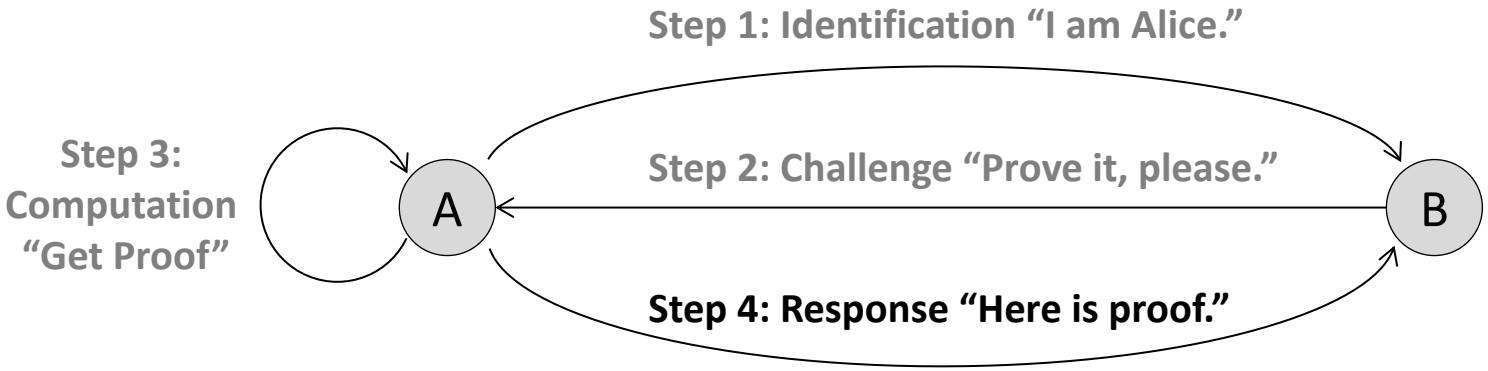
Week 5

One-Factor Authentication Schema



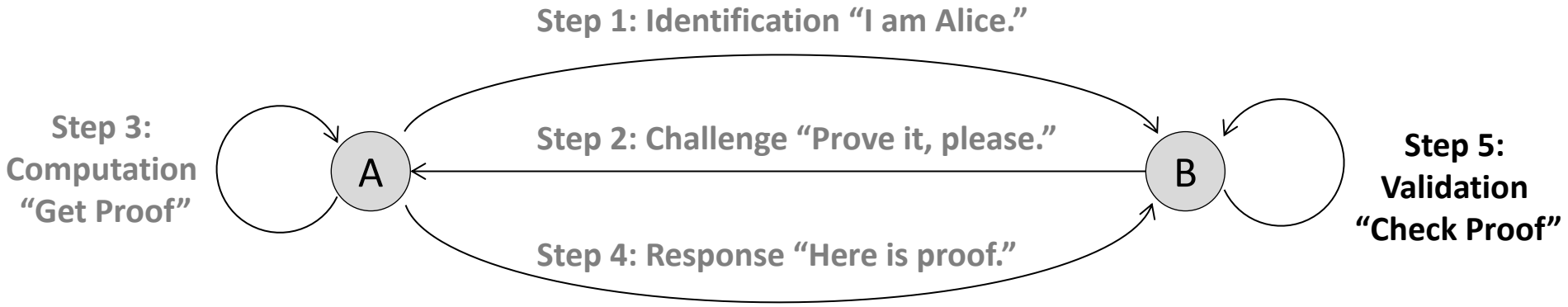
Week 5

One-Factor Authentication Schema



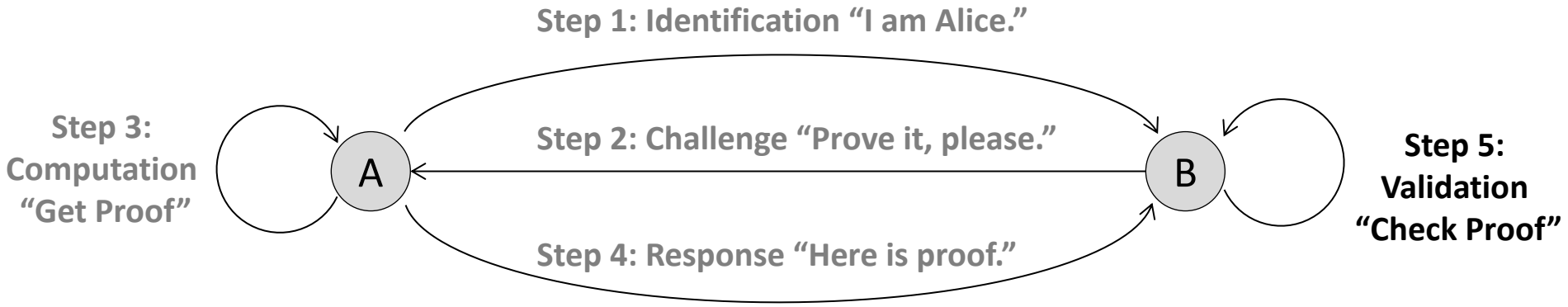
Week 5

One-Factor Authentication Schema



Week 5

One-Factor Authentication Schema

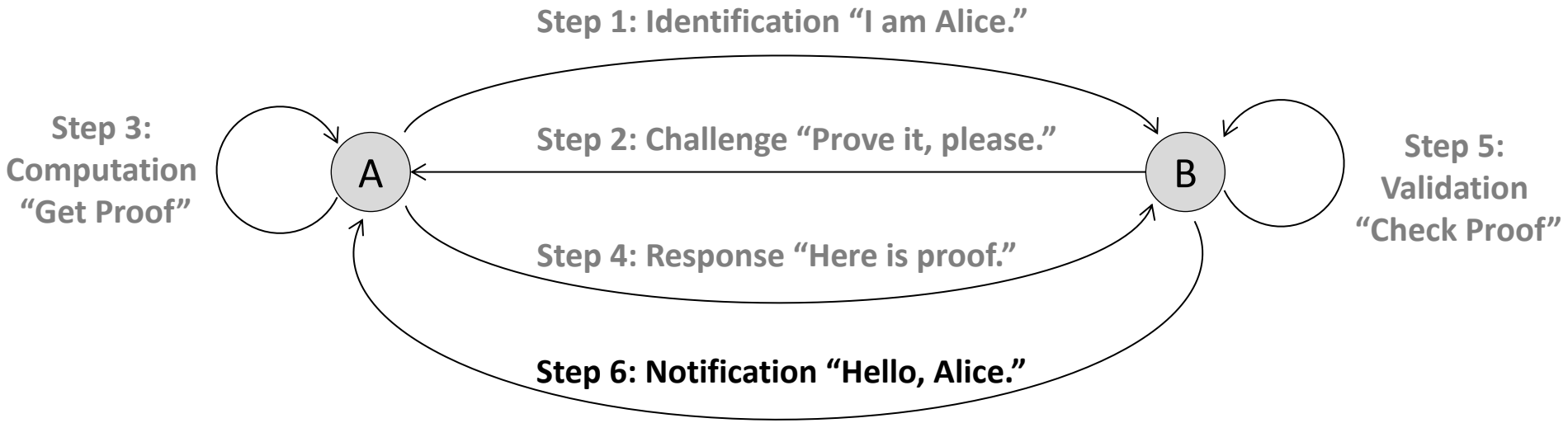


Validation might involve simple look-up/locate process (e.g., passwords)

Validation might be more deliberate mathematical operation on domain value

Week 5

One-Factor Authentication Schema

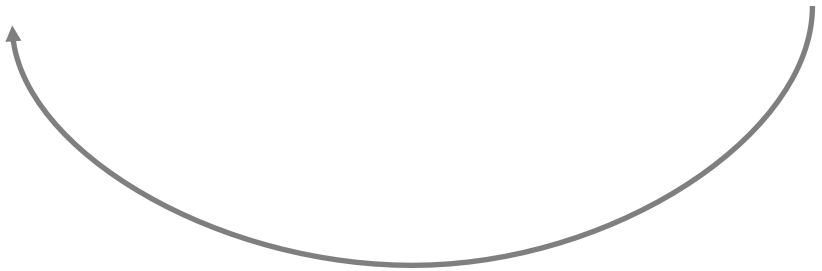


Week 5

Two-Factor Authentication Schema

Types of Proof:
"Something You Know" – Passwords
"Something You Are" – Biometrics
"Something You Have" – Token
"Somewhere You Are" – Location

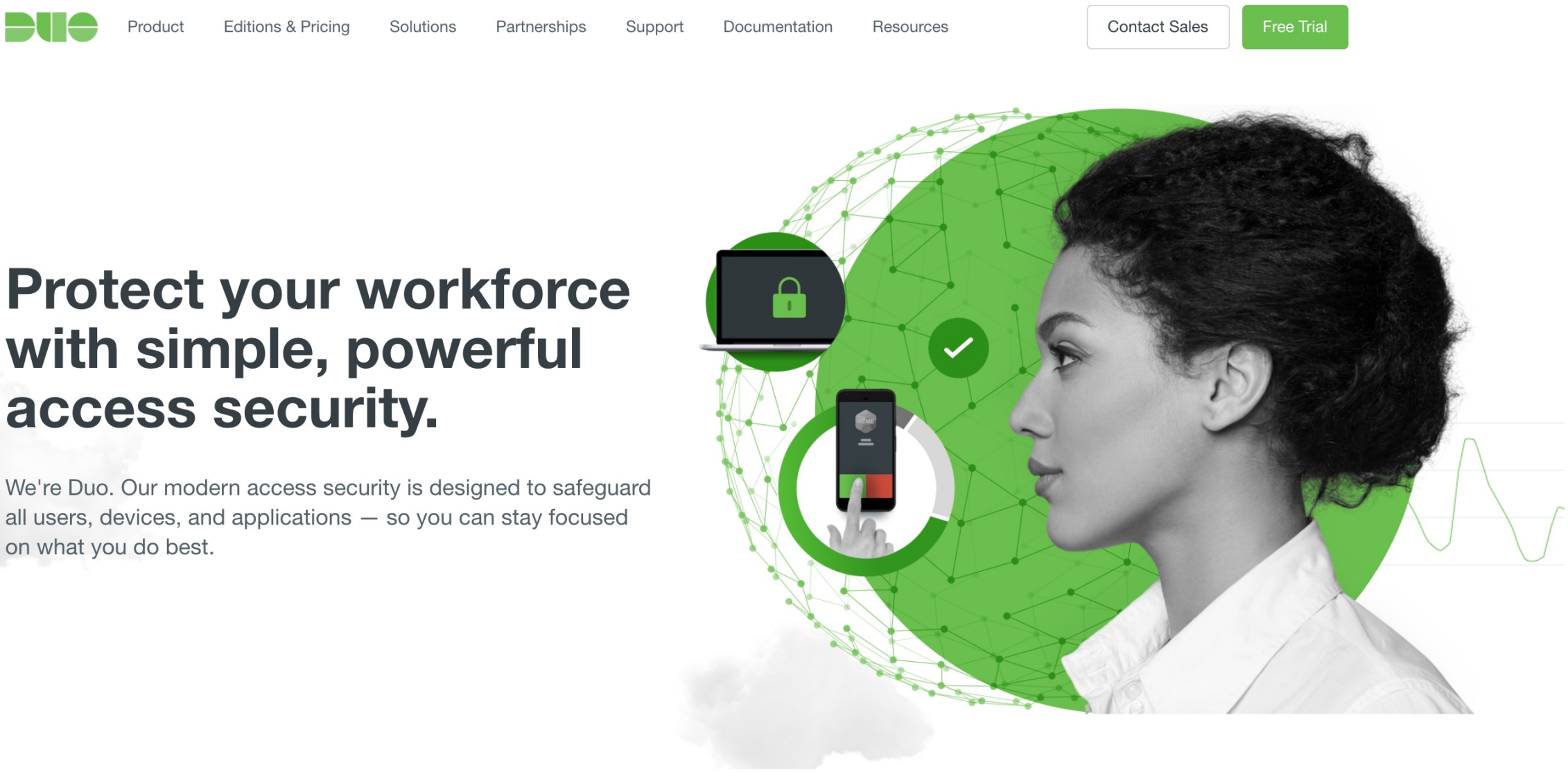
- ***Adaptive Authentication*** **considers context**
- ***Two-Factor Authentication*** **uses at least two types**



Choose Two Factors

Week 5

MyStevens Two-Factor Authentication



Week 5

ZenKey Mobile Carrier Authentication (Metadata)

ZENKEY

WHAT IS ZENKEY | BUSINESS | DEVELOPERS | RESOURCES | CONTACT | Get ZenKey

Unlock Trust

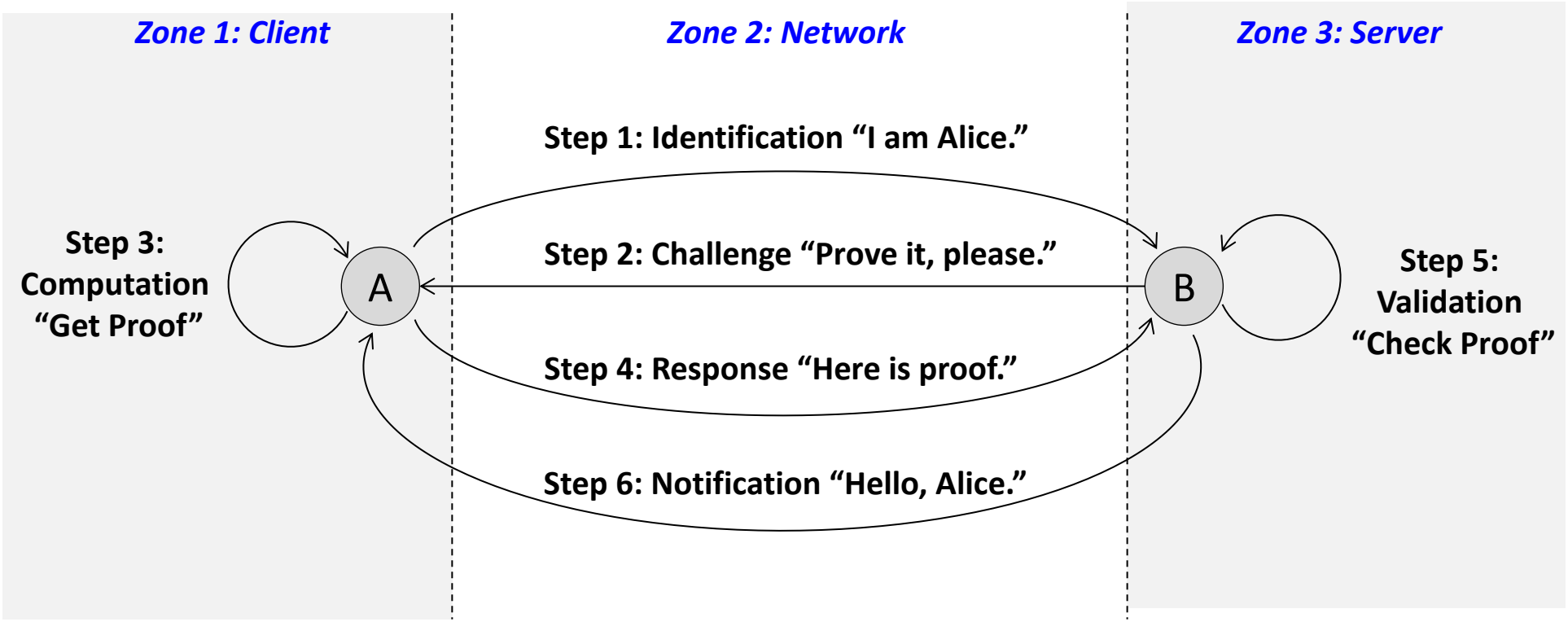
ZenKey gives you a fast and safe way to register for apps and websites. It also makes sure that the device you're using matches the device info you registered with your wireless carrier. This helps to secure your logins and transactions.

Be one of the first to try ZenKey and get the peace of mind you deserve.

How Do Hackers Try to Bypass Authentication?

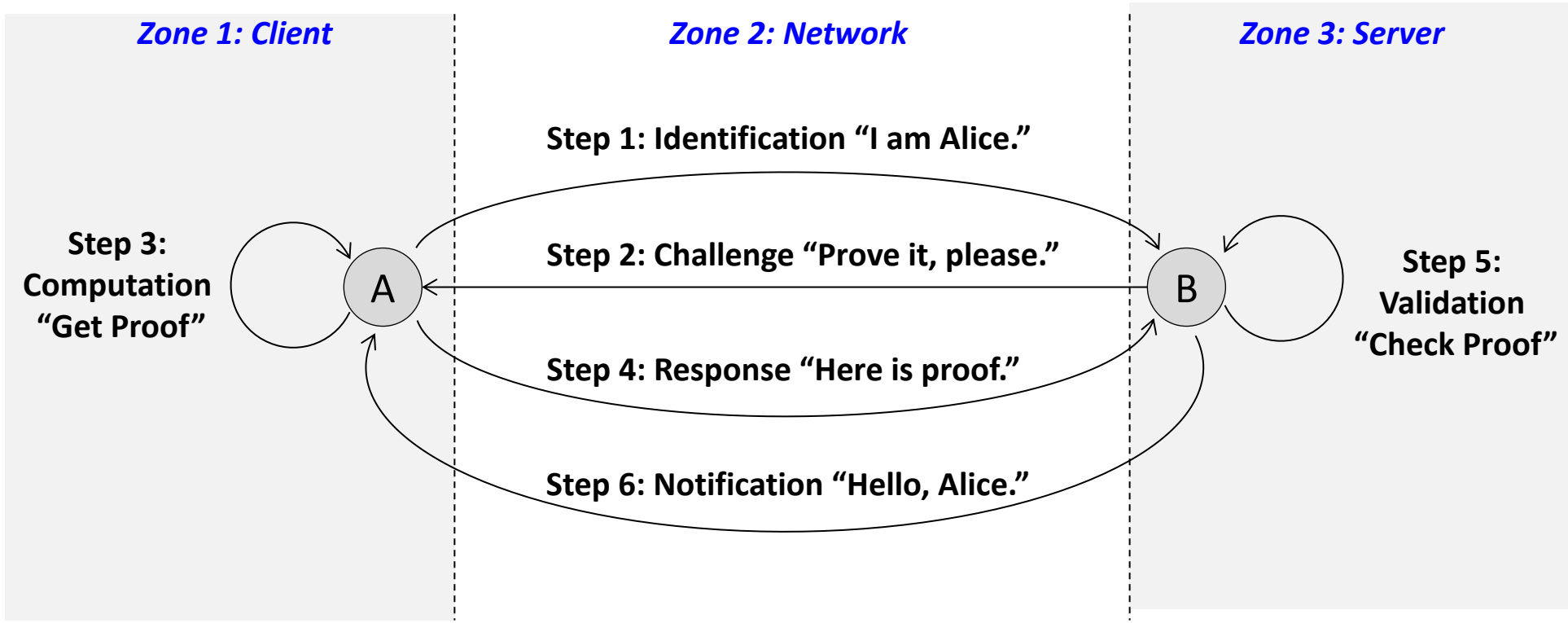
Week 5

Authentication Schema



Week 5

Authentication Schema

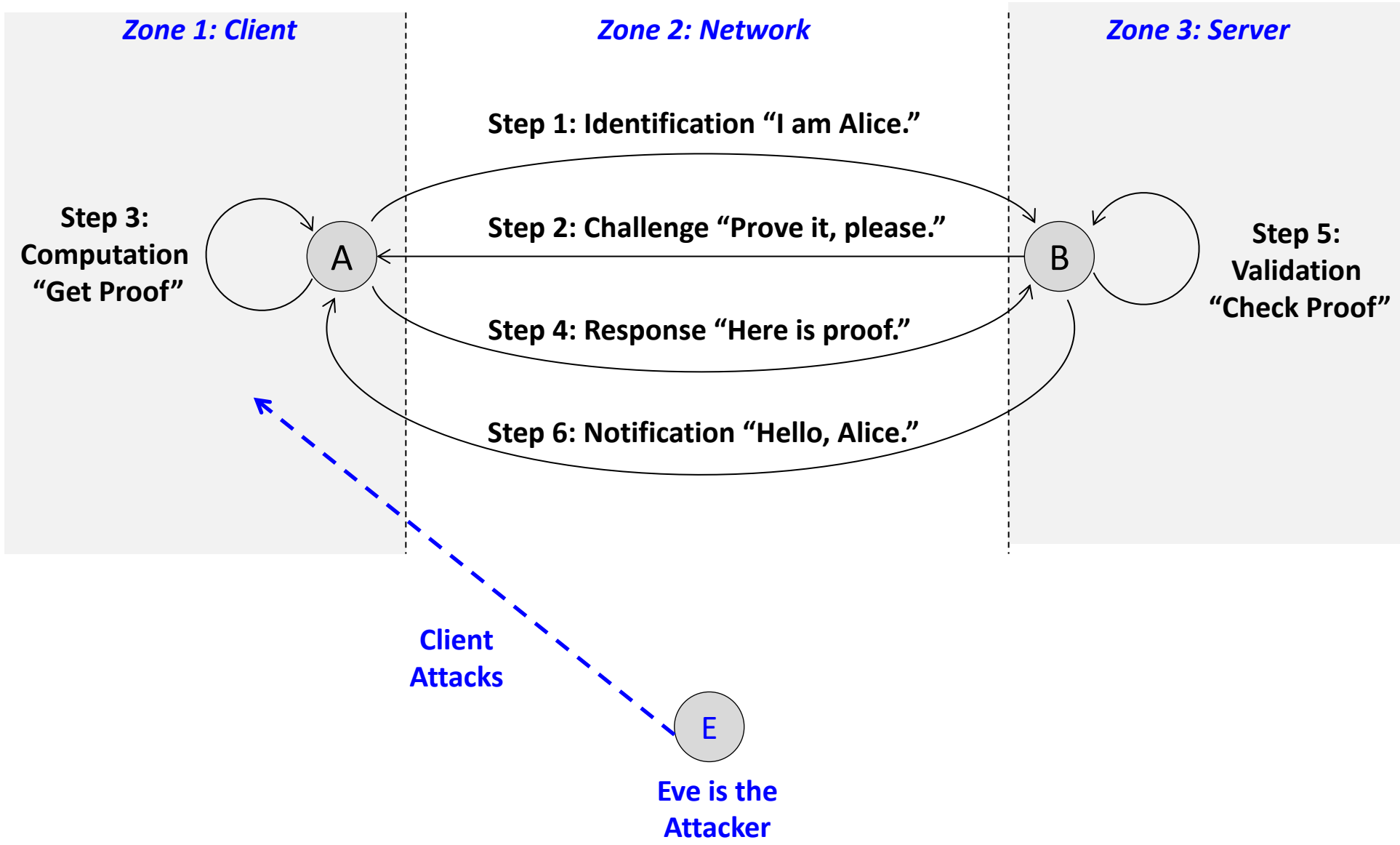


E

Eve is the Attacker

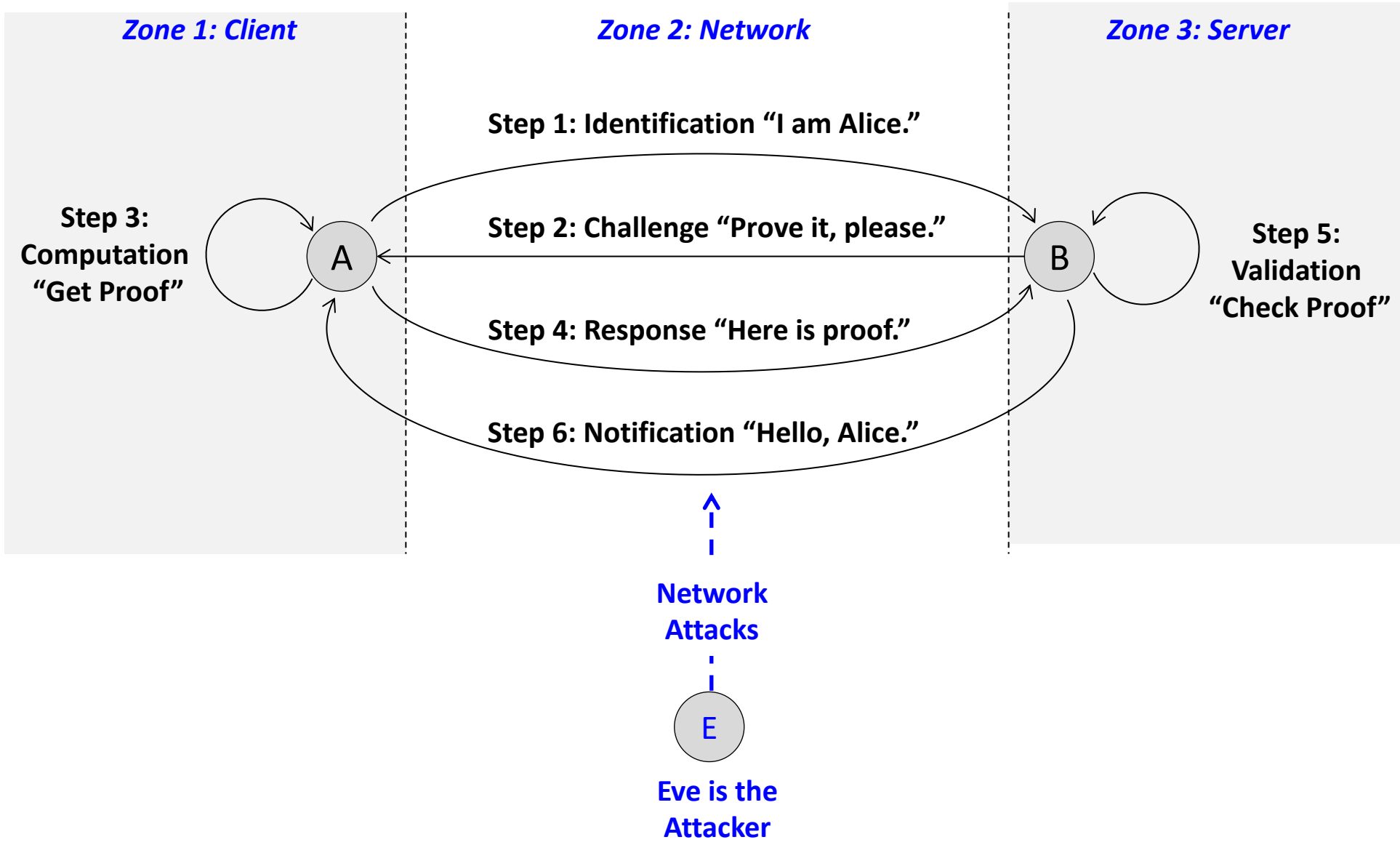
Week 5

Authentication Schema



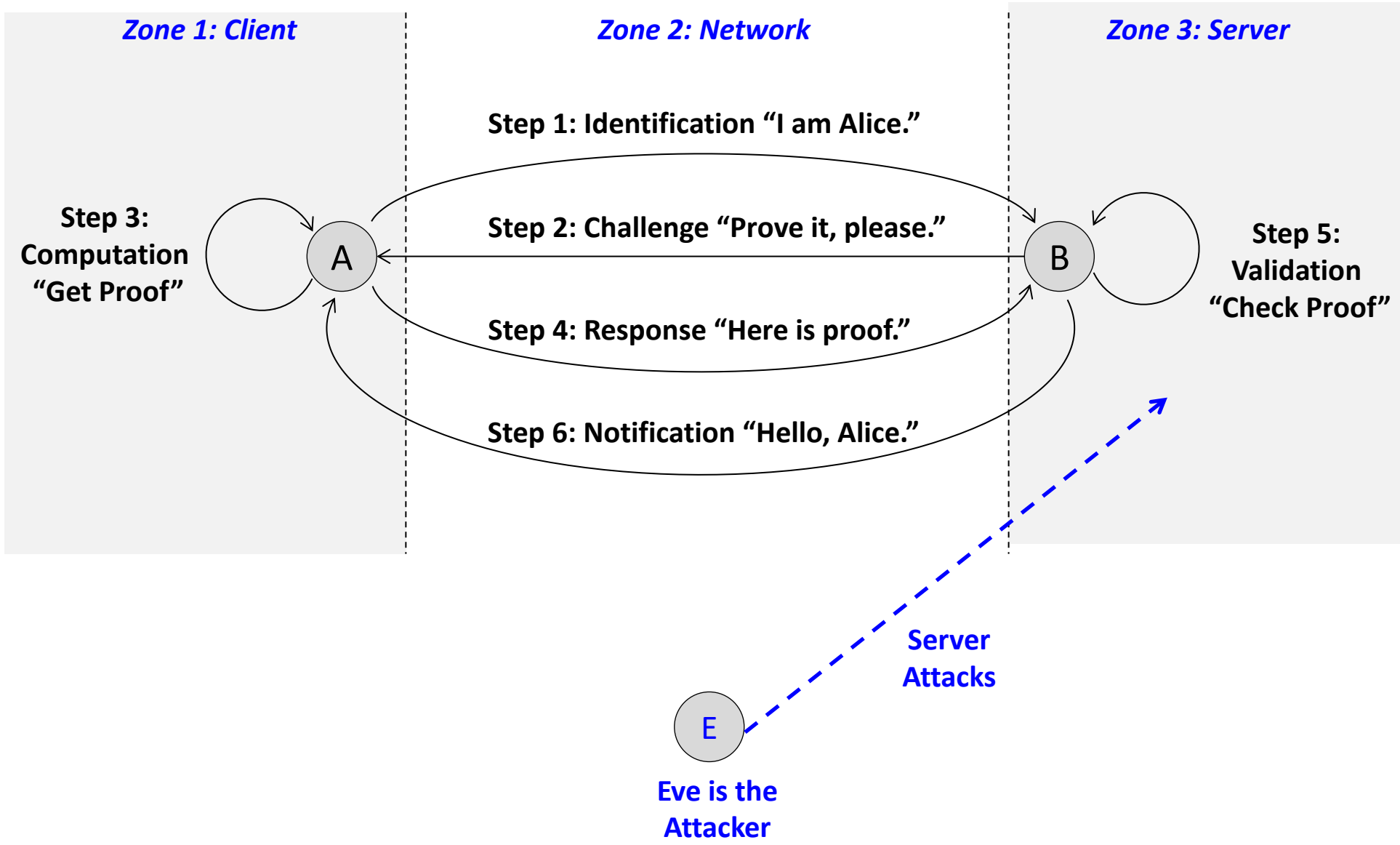
Week 5

Authentication Schema



Week 5

Authentication Schema



Are Passwords Acceptable for Authentication?

Week 5

Password-Related Incident: Democratic National Committee 2016

WikiLeaks Leaks News About Partners

> *From:* Google <no-reply@accounts.googlemail.com>
> *Date:* March 19, 2016 at 4:34:30 AM EDT
> *To:* john.podesta@gmail.com
> *Subject:* *Someone has your password*
>
> Someone has your password
> Hi John
>
> Someone just used your password to try to sign in to your Google Account
> john.podesta@gmail.com.
>
> Details:
> Saturday, 19 March, 8:34:30 UTC
> IP Address: 134.249.139.239
> Location: Ukraine
>
> Google stopped this sign-in attempt. You should change your password
> immediately.
>

Week 5

Password-Related Incident: Colonial Pipeline Ransomware 2021

Cybersecurity

Hackers Breached Colonial Pipeline Using Compromised Password

By William Turton and Kartikay Mehrotra
June 4, 2021, 3:58 PM EDT

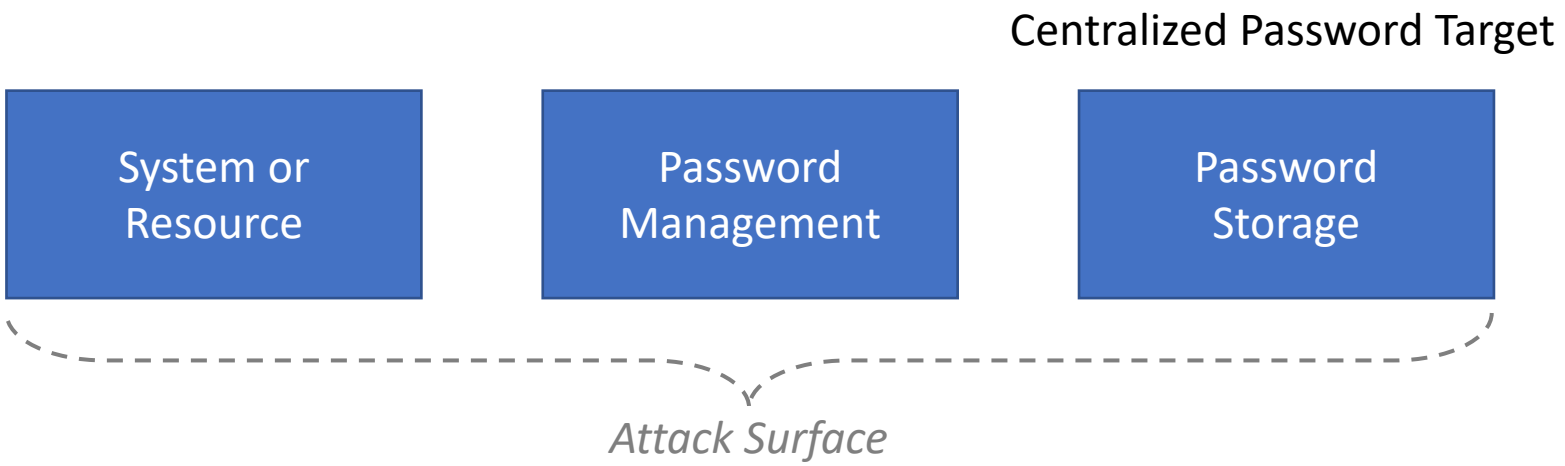
The account's password has since been discovered inside a batch of leaked passwords on the dark web. That means a Colonial employee may have used the same password on another account that was previously hacked, he said. However, Carmakal said he isn't certain that's how hackers obtained the password, and he said investigators may never know for certain how the credential was obtained.

The hack that took down the largest fuel pipeline in the U.S. and led to shortages across the East Coast was the result of a single compromised password, according to a cybersecurity consultant who responded to the attack.

Hackers gained entry into the networks of Colonial Pipeline Co. on April 29 through a virtual private network account, which allowed employees to remotely access the company's computer network, said Charles Carmakal, senior vice president at cybersecurity firm Mandiant, part of FireEye Inc., in an interview. The account was no longer in use at the time of the attack but could still be used to access Colonial's network, he said.

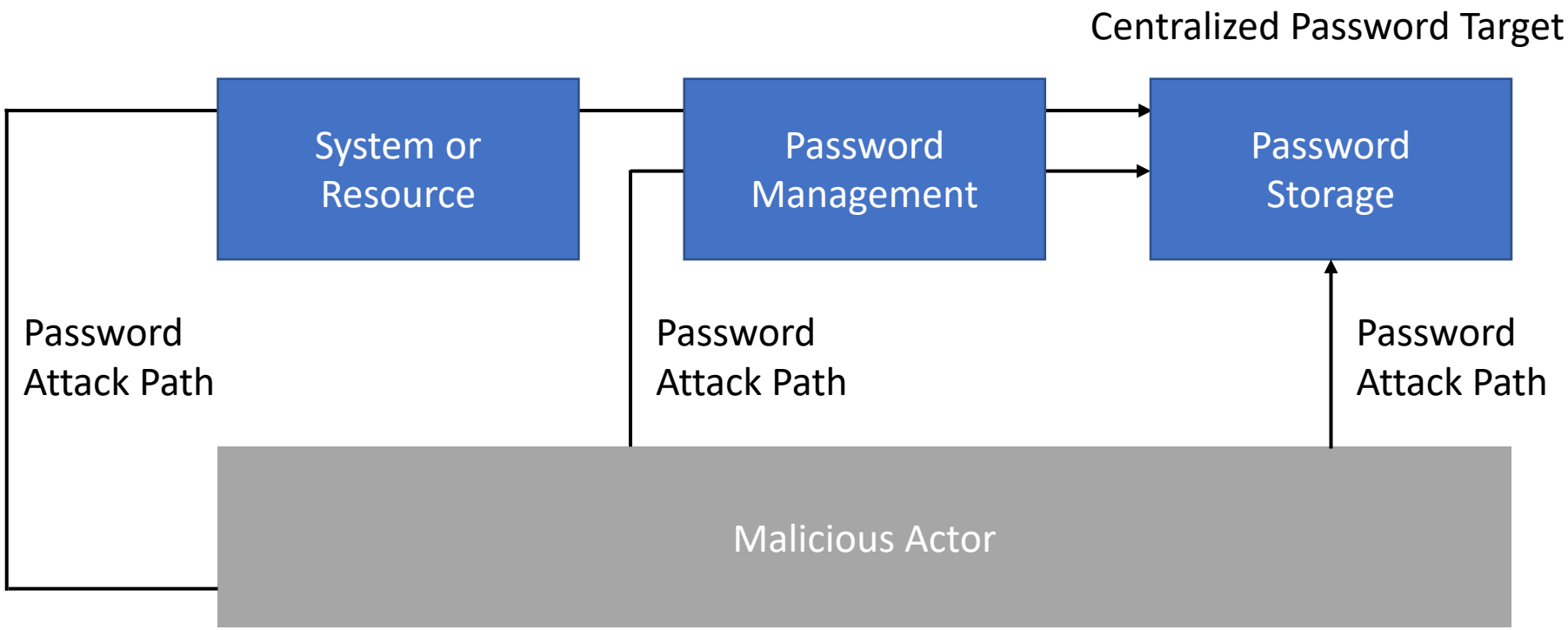
Week 5

Inherent Threat of Password Repositories



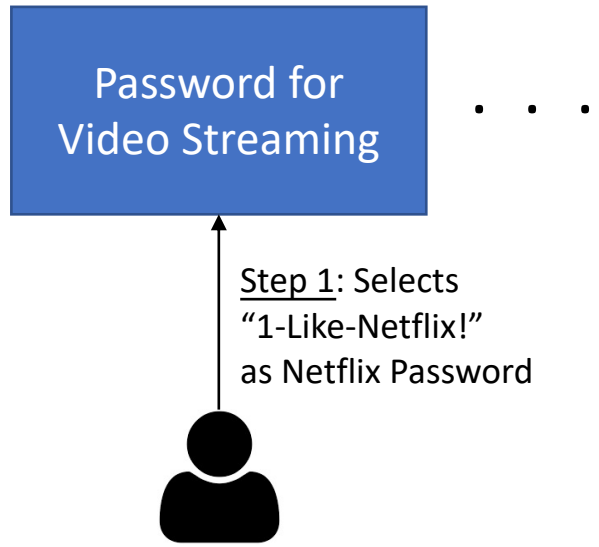
Week 5

Inherent Threat of Password Repositories



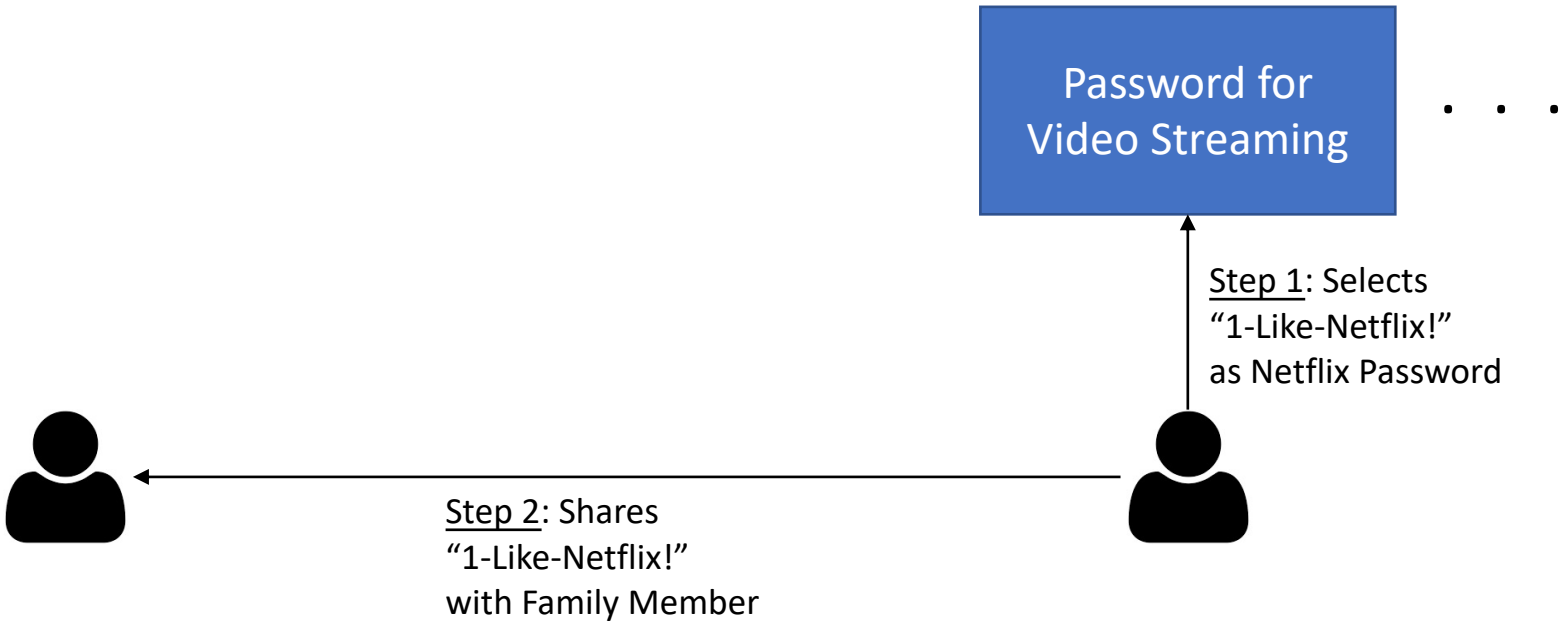
Week 5

Inherent Threat of Password Reuse



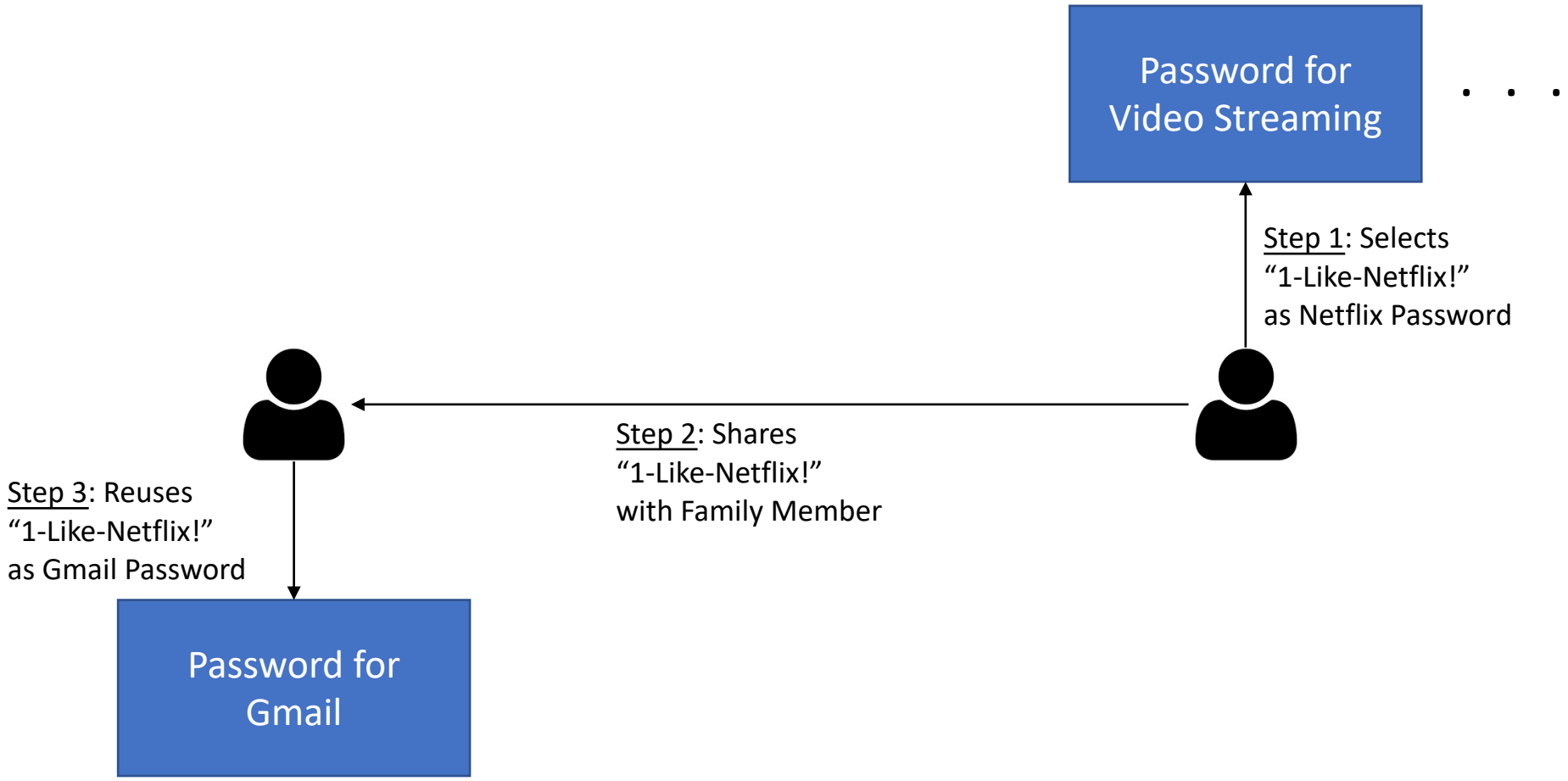
Week 5

Inherent Threat of Password Reuse



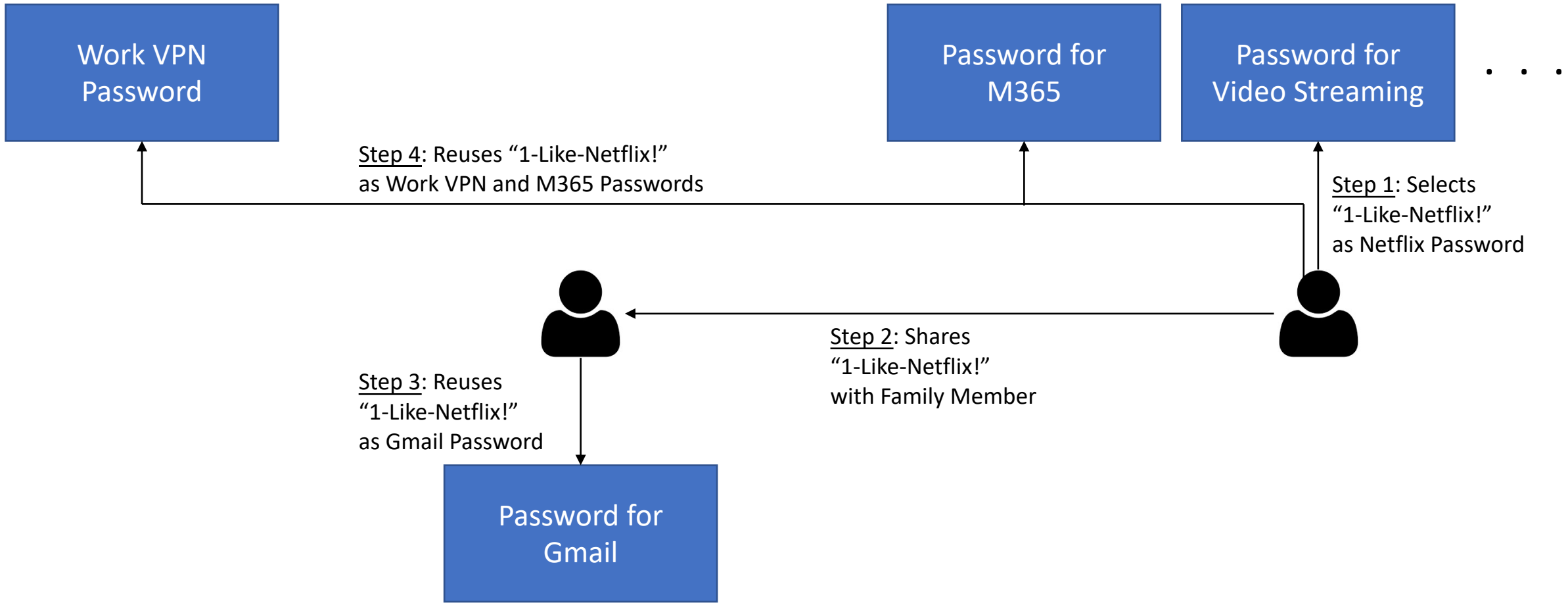
Week 5

Inherent Threat of Password Reuse



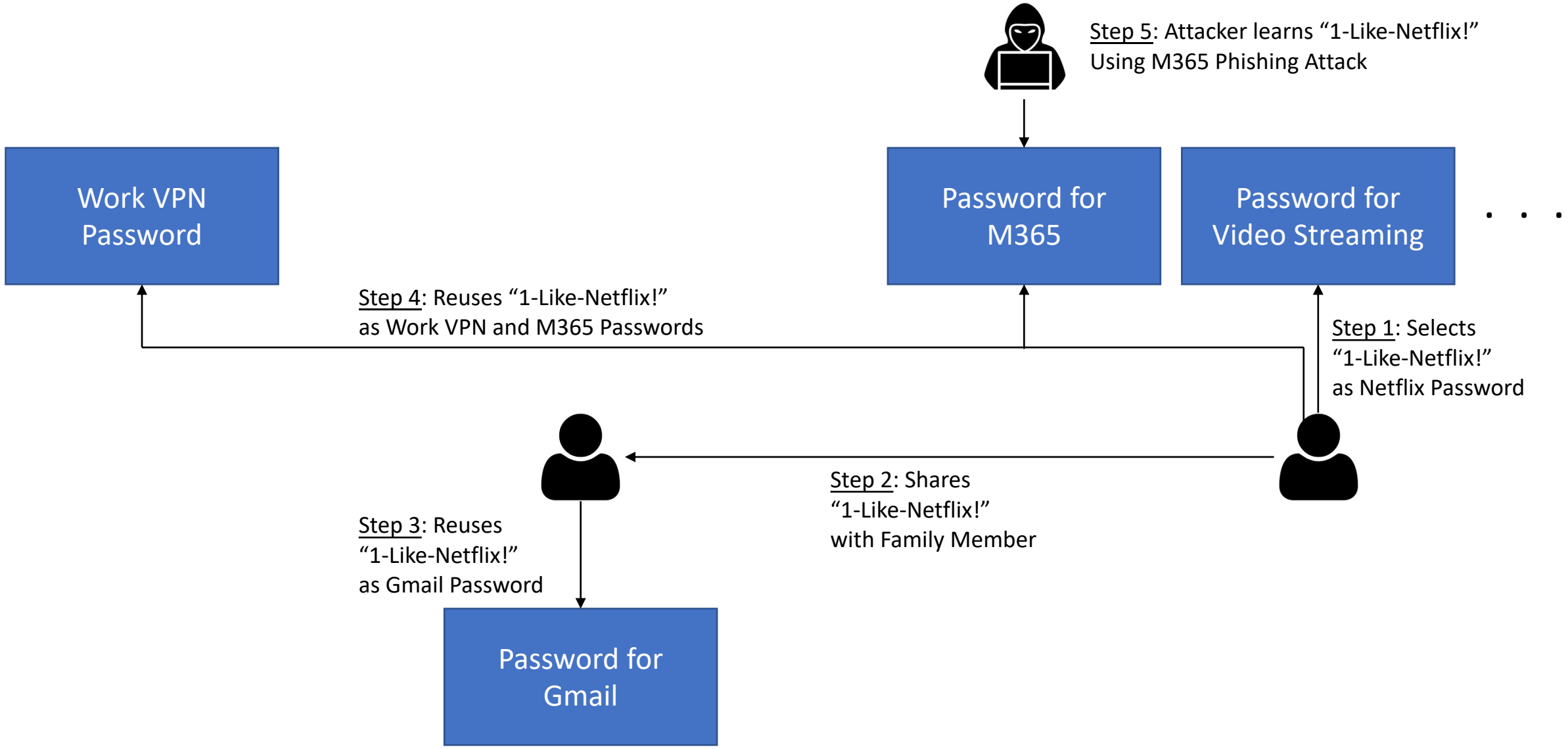
Week 5

Inherent Threat of Password Reuse



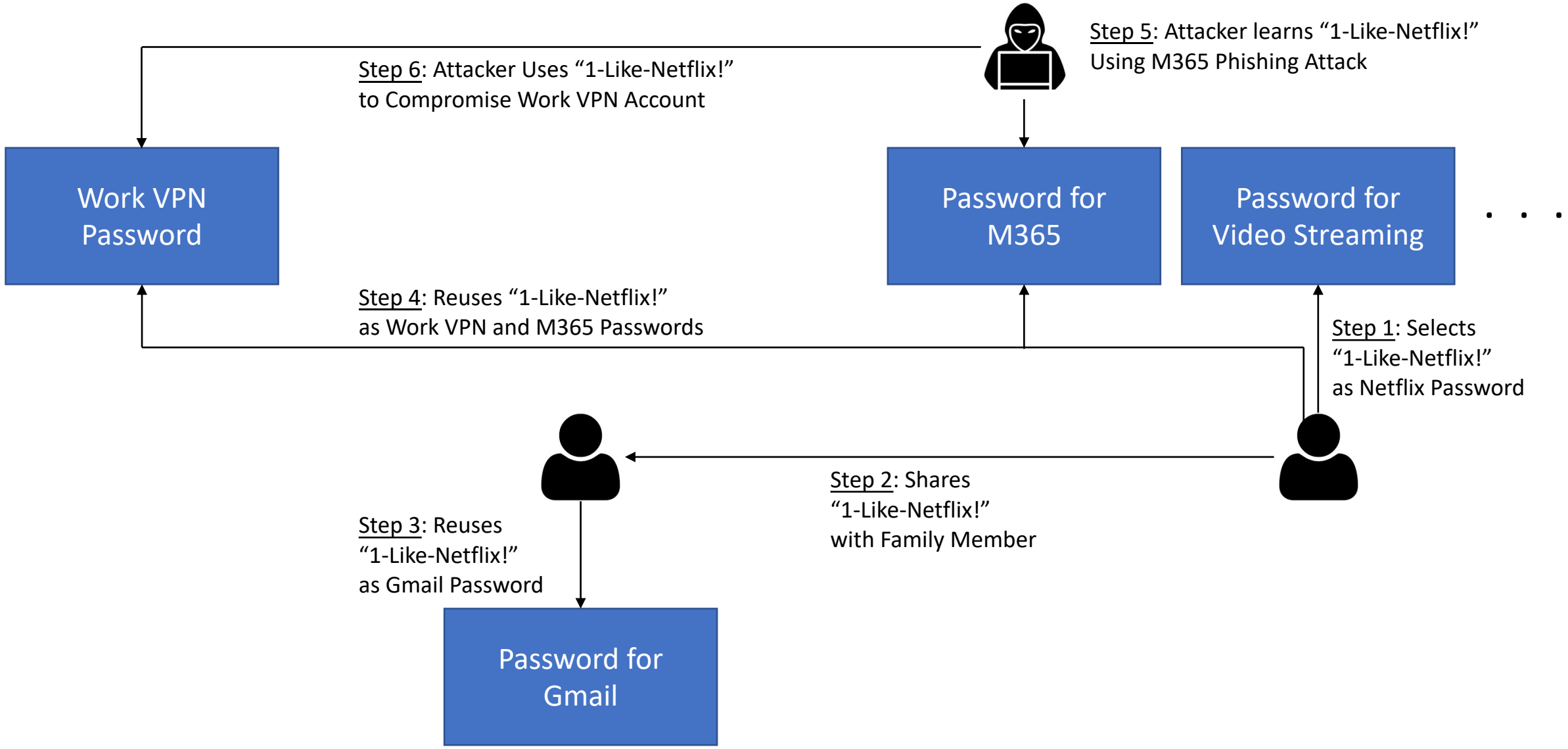
Week 5

Inherent Threat of Password Reuse



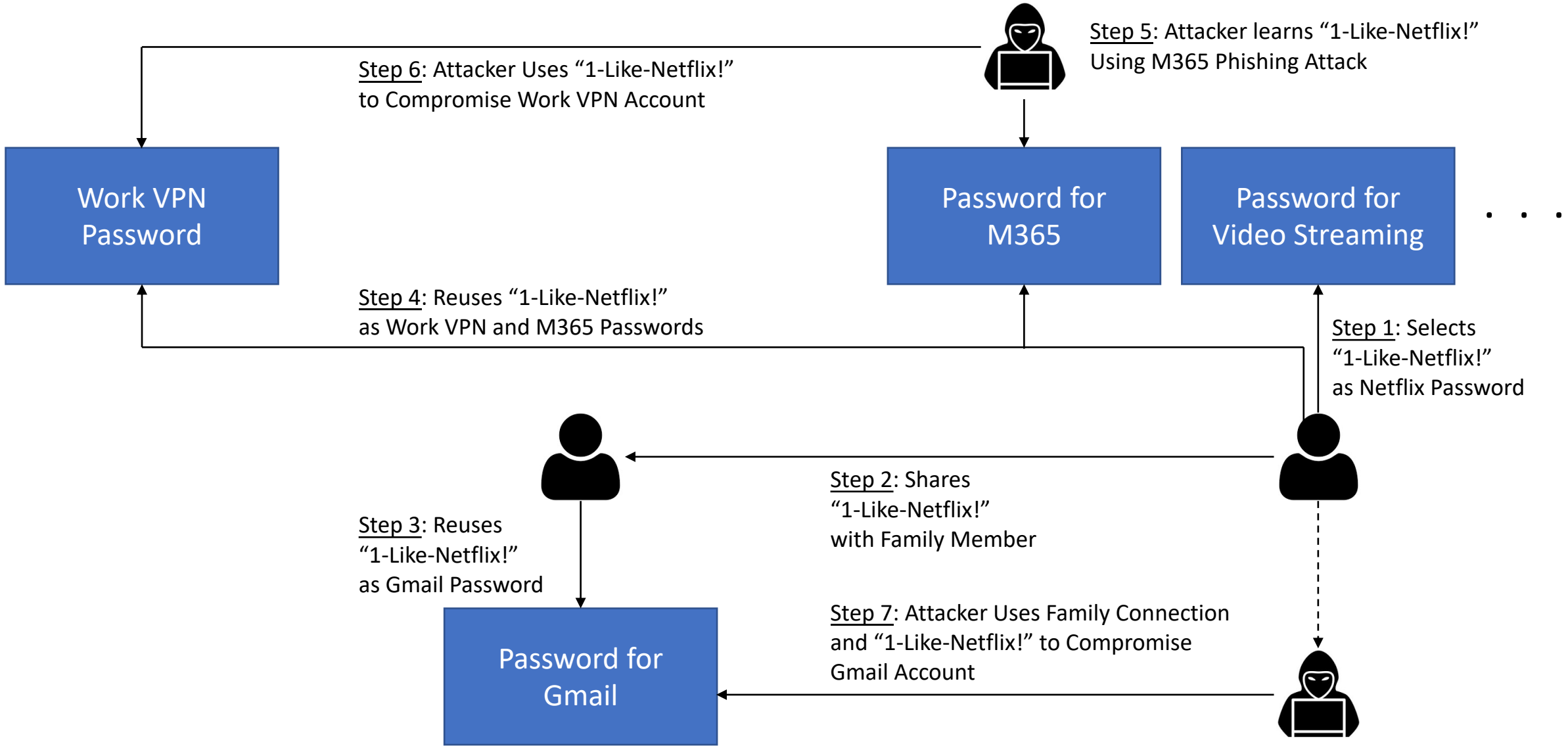
Week 5

Inherent Threat of Password Reuse



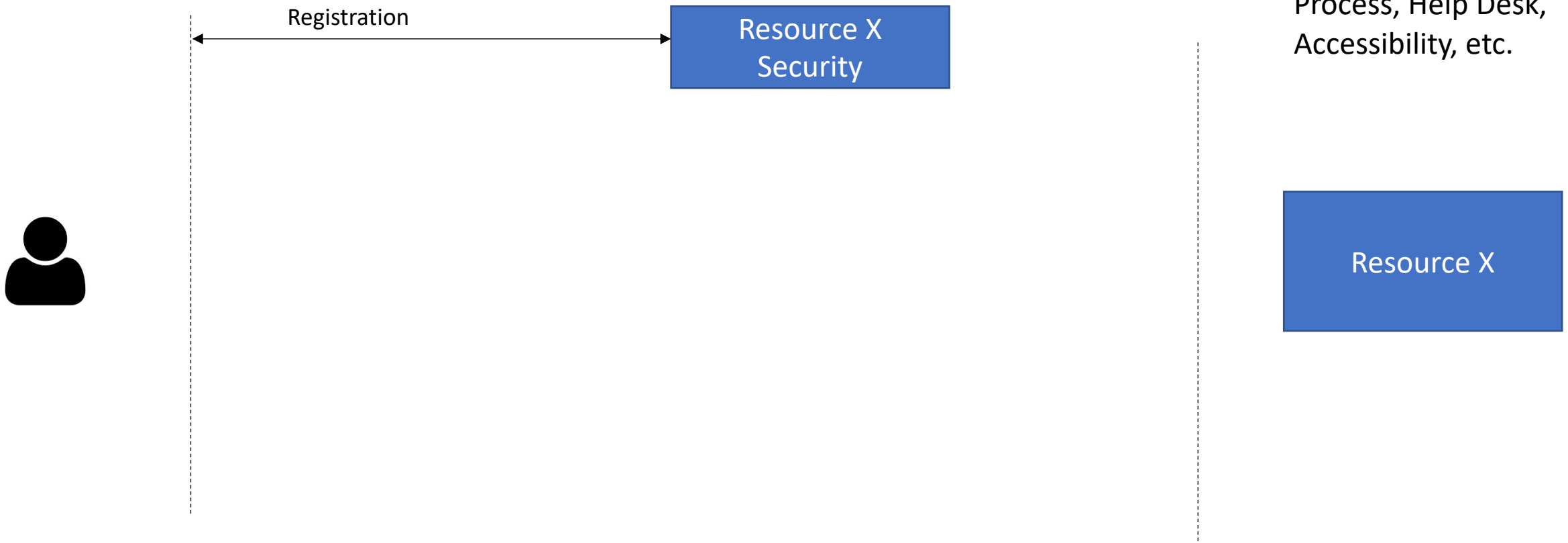
Week 5

Inherent Threat of Password Reuse



Week 5

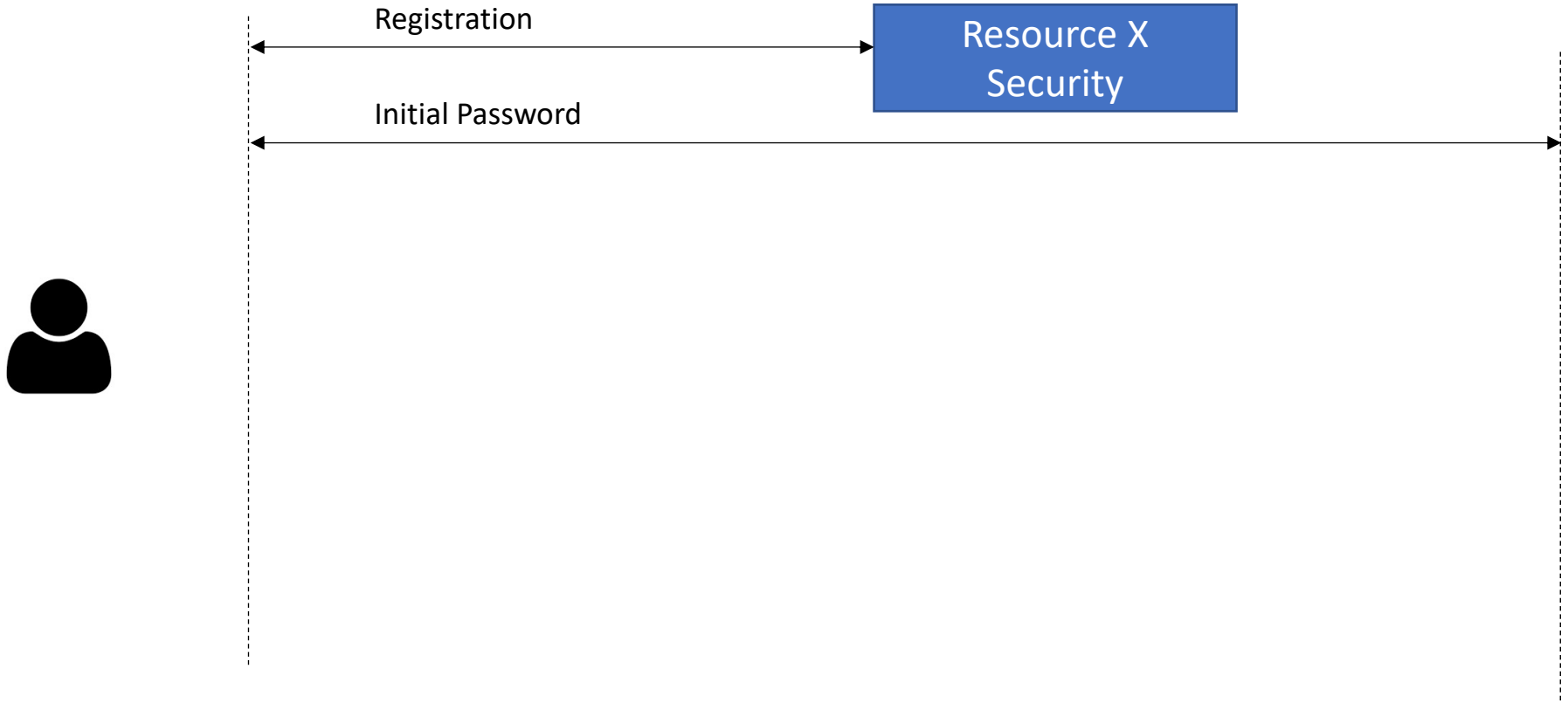
Inherent Friction from Password Usage



Friction:
Process, Help Desk, Accessibility, etc.

Week 5

Inherent Friction from Password Usage



Friction:
Machine Generated,
User Selected, etc.

Resource X

Week 5

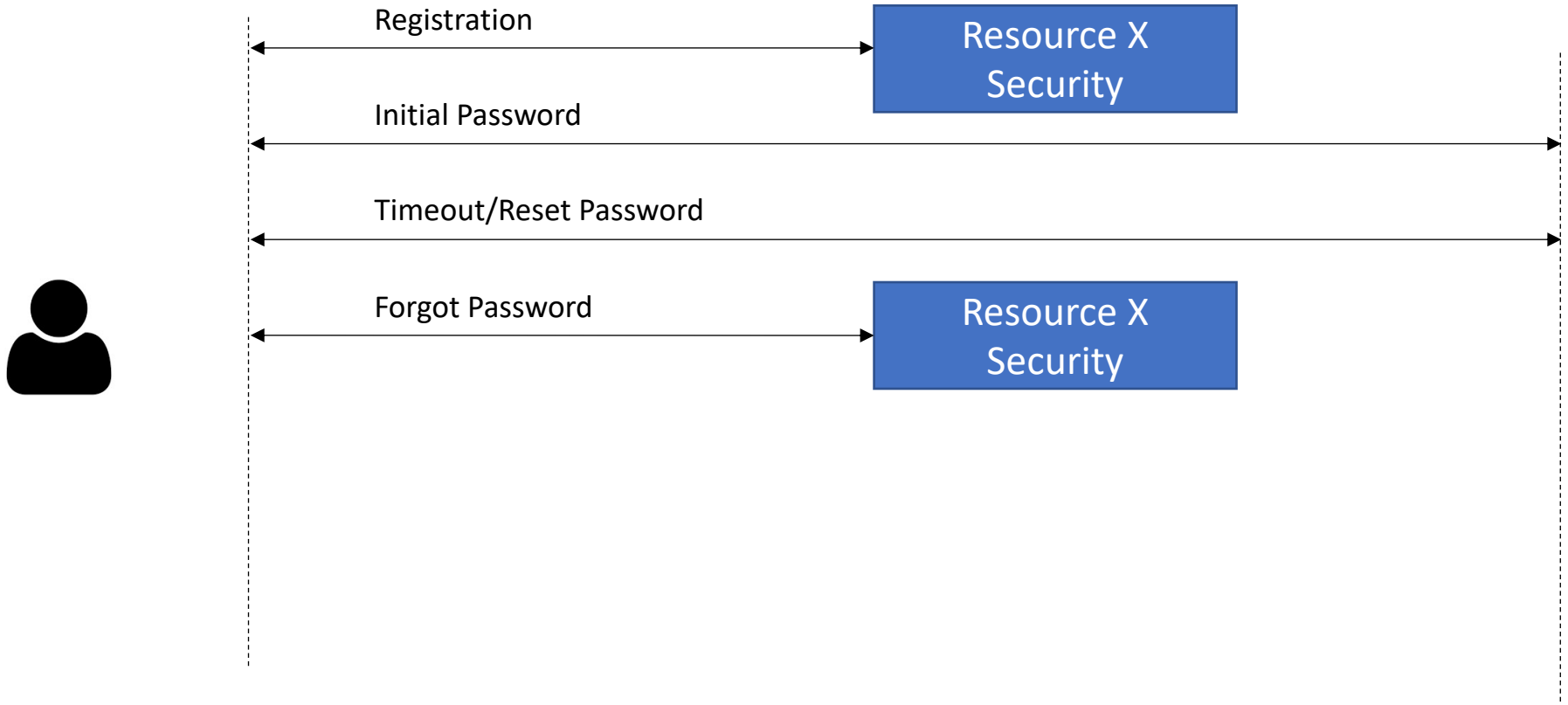
Inherent Friction from Password Usage

Friction:
Blocked Resource, Frustration, etc.

Resource X

Week 5

Inherent Friction from Password Usage

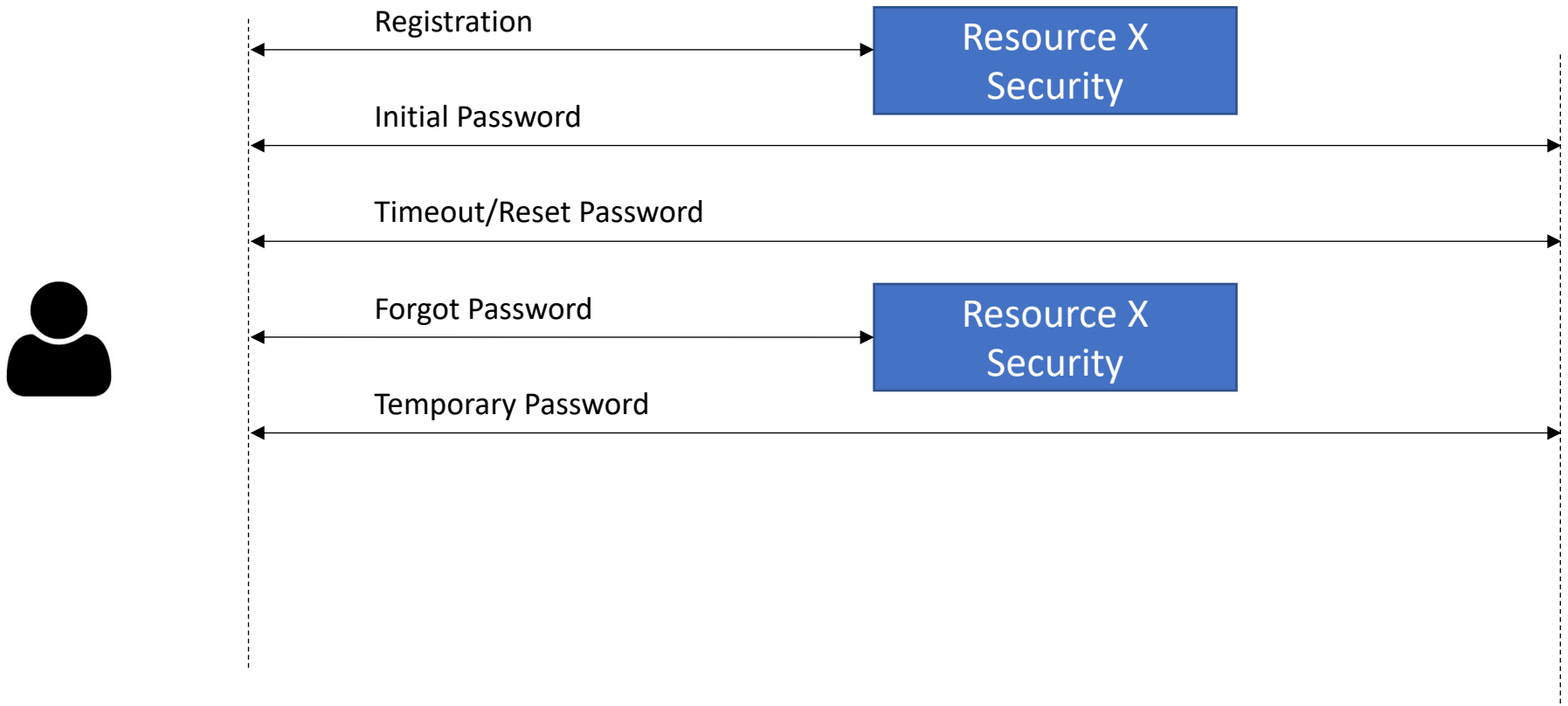


Friction:
Annoyance,
Frustration, etc.

Resource X

Week 5

Inherent Friction from Password Usage

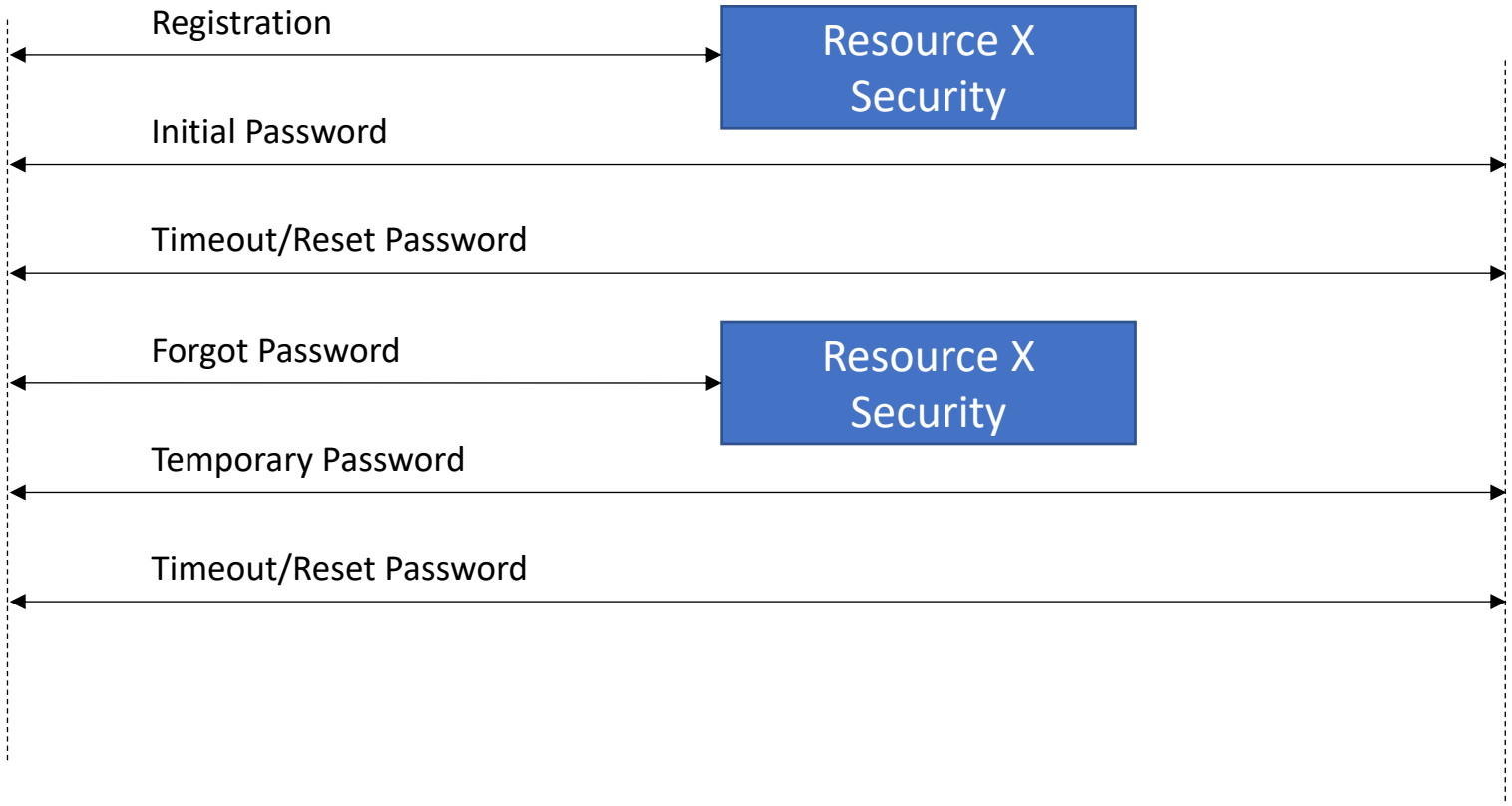


Friction:
Inconvenience,
New Password, etc.

Resource X

Week 5

Inherent Friction from Password Usage

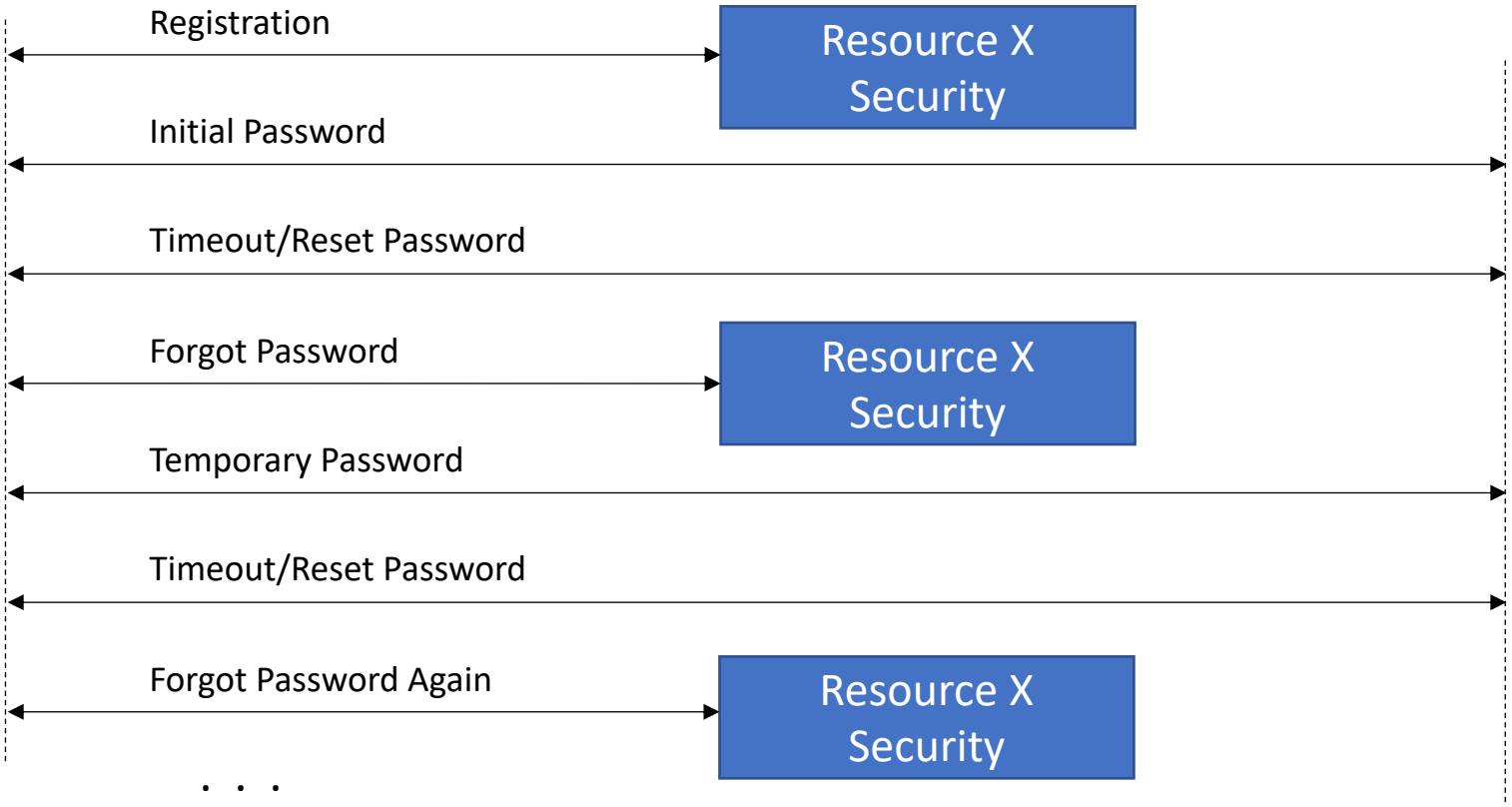


Friction:
Blocked Resource,
More Frustration, etc.

Resource X

Week 5

Inherent Friction from Password Usage



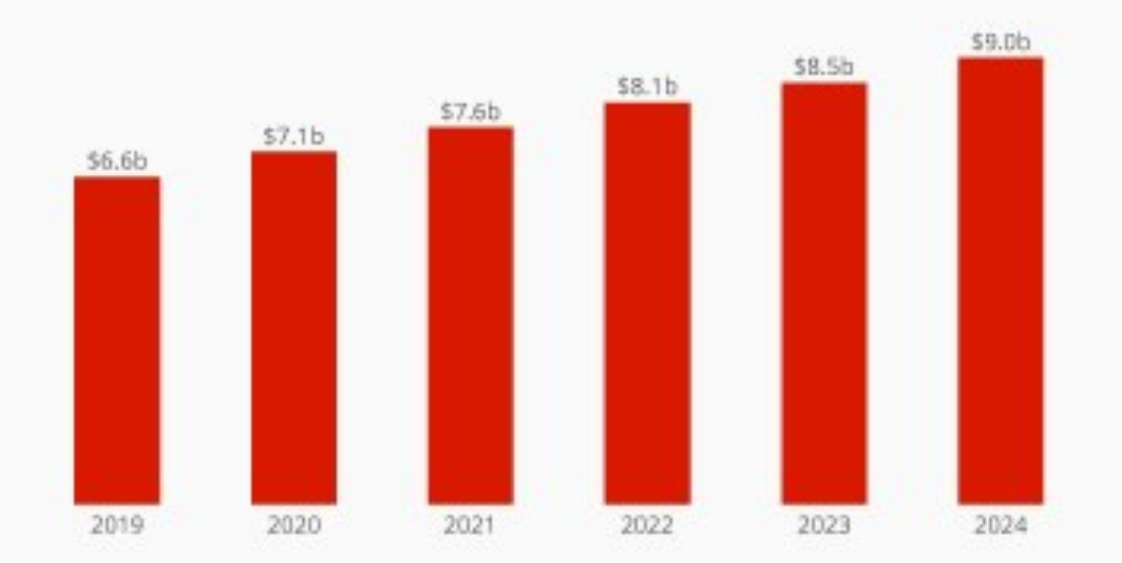
Friction:
Annoyance,
Frustration, etc.

Resource X

Week 5

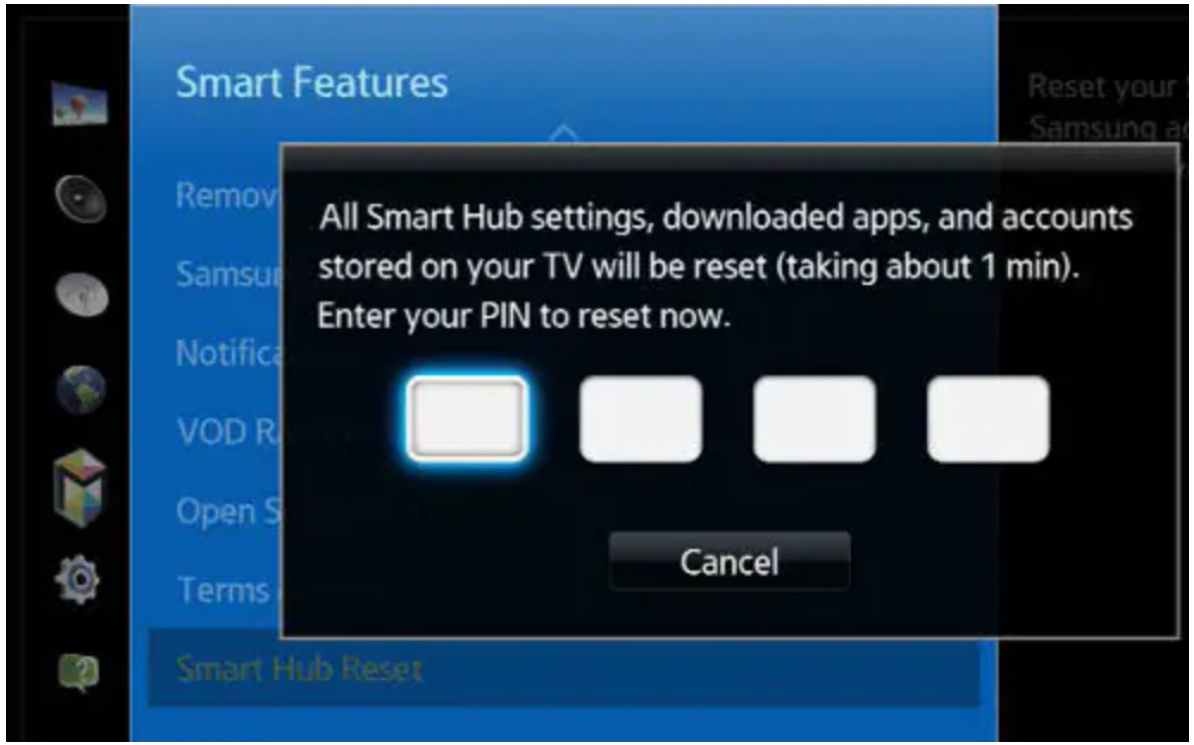
Password Issues with Smart TV/Streaming Channels

Estimated Revenue Losses for US Pay TV Industry from Piracy and Account Sharing



Source: Statistica

https://www.statista.com/chart/19914/estimated-revenue-loss-for-the-us-pay-tv-industry-from-piracy-and-account-sharing/



How Did Handheld Authenticators Work?

Week 5

Handheld Authentication Device

A

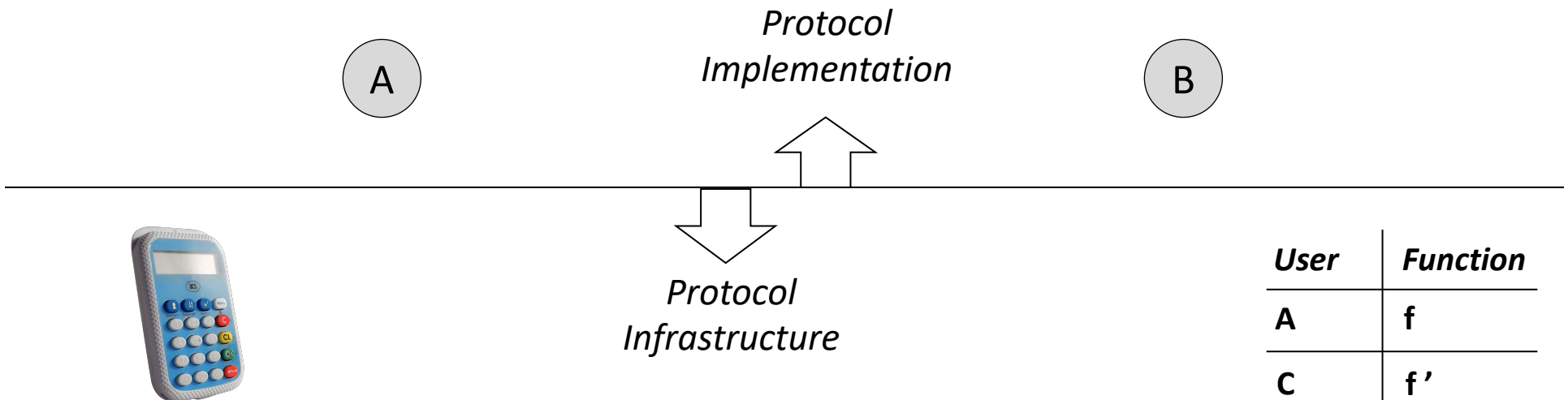
Embedded Function
f: integer -> integer

B

User	*Function*
A	f
C	f ’
G	f ’’
. . .	. . .

Week 5

Handheld Authentication Device



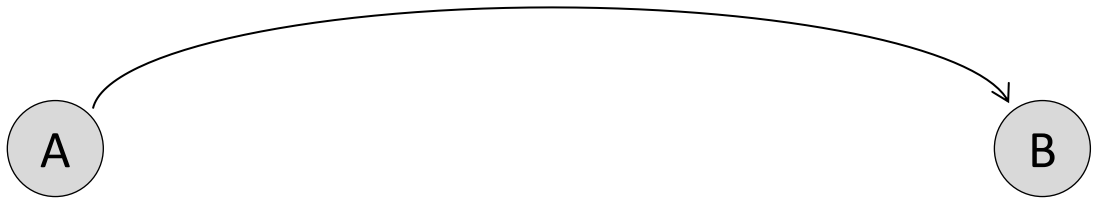
Embedded Function
f: integer -> integer

User	*Function*
A	f
C	f ’
G	f ’’
. . .	. . .

Week 5

Handheld Authentication Device

Step 1: I am Alice

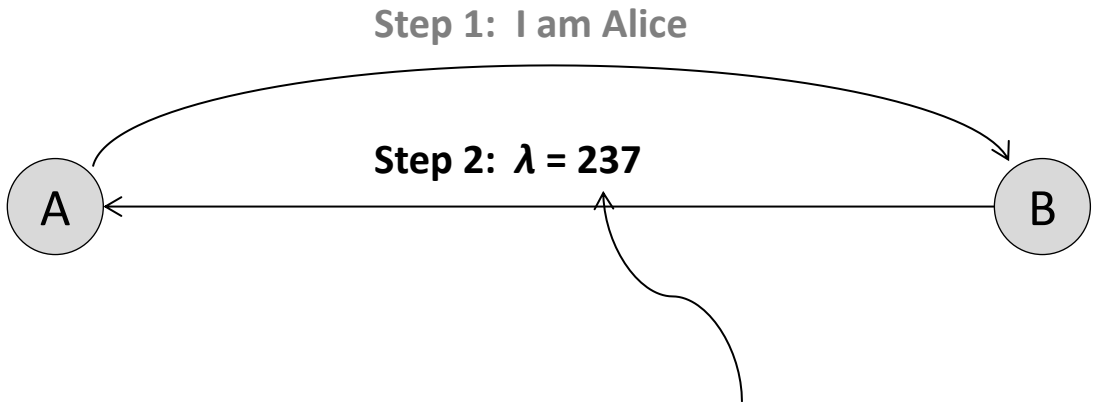


Embedded Function
f: integer -> integer

User	*Function*
A	f
C	f ’
G	f ’’
. . .	. . .

Week 5

Handheld Authentication Device

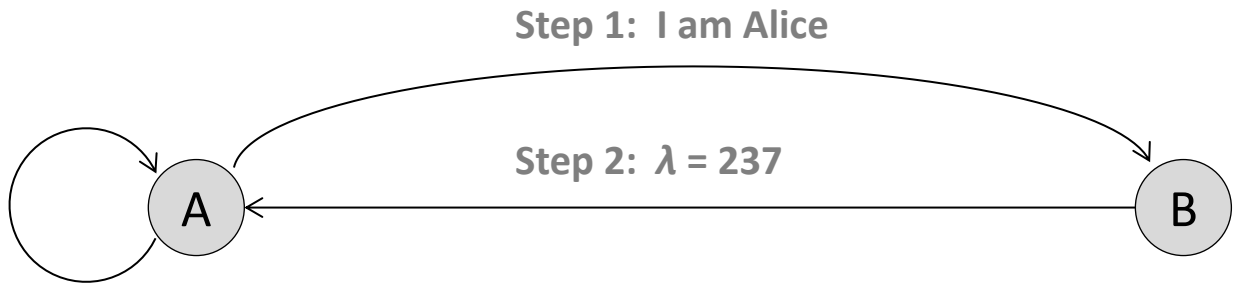


Embedded Function
f: integer -> integer

User	*Function*
A	f
C	f ’
G	f ’’
. . .	. . .

Week 5

Handheld Authentication Device

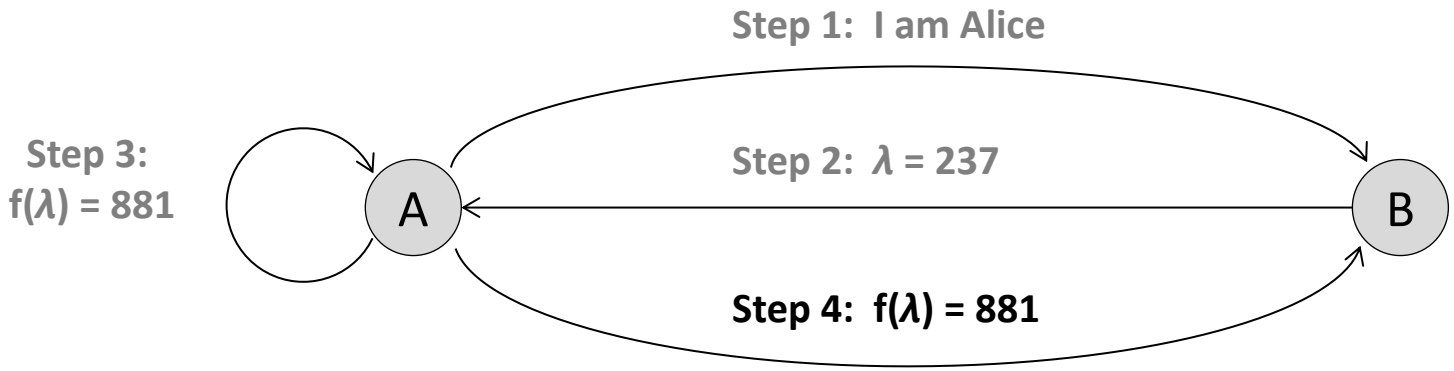


Embedded Function
f: integer -> integer

User	*Function*
A	f
C	f ’
G	f ’’
. . .	. . .

Week 5

Handheld Authentication Device

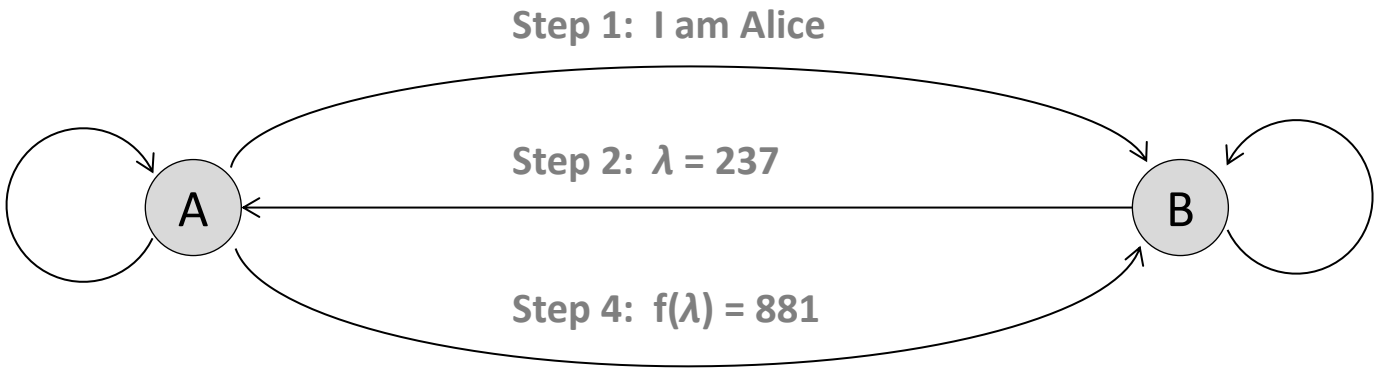


Embedded Function
f: integer -> integer

User	*Function*
A	f
C	f ’
G	f ’’
. . .	. . .

Week 5

Handheld Authentication Device



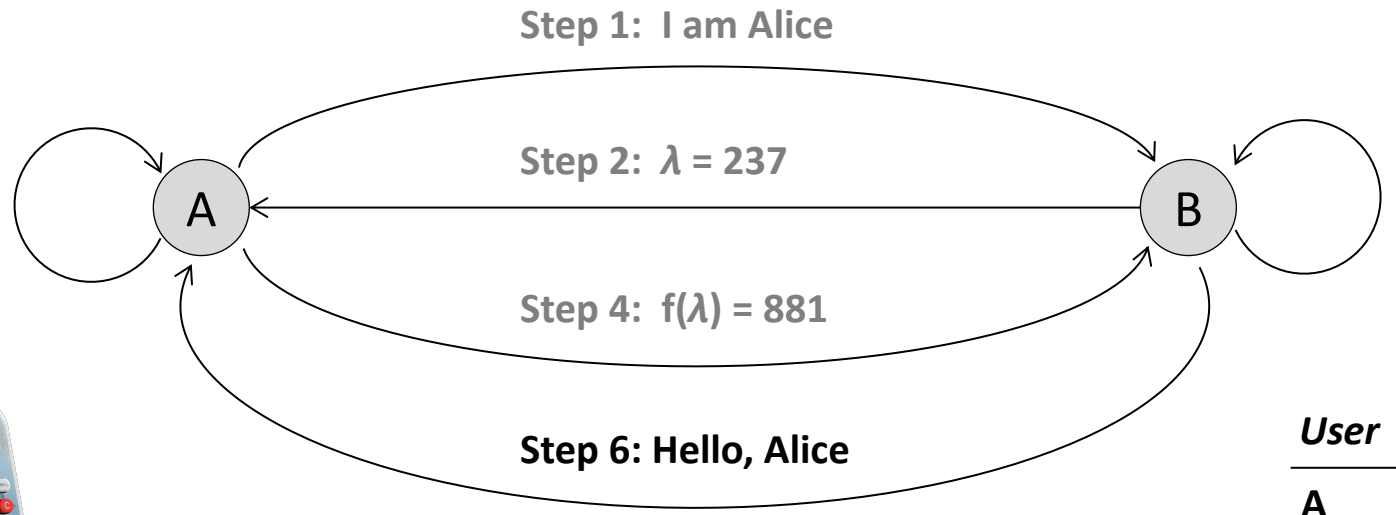
Step 5: Compute f(λ) = 881 locally

Embedded Function
f: integer -> integer

User	*Function*
A	f
C	f ’
G	f ’’
. . .	. . .

Week 5

Handheld Authentication Device

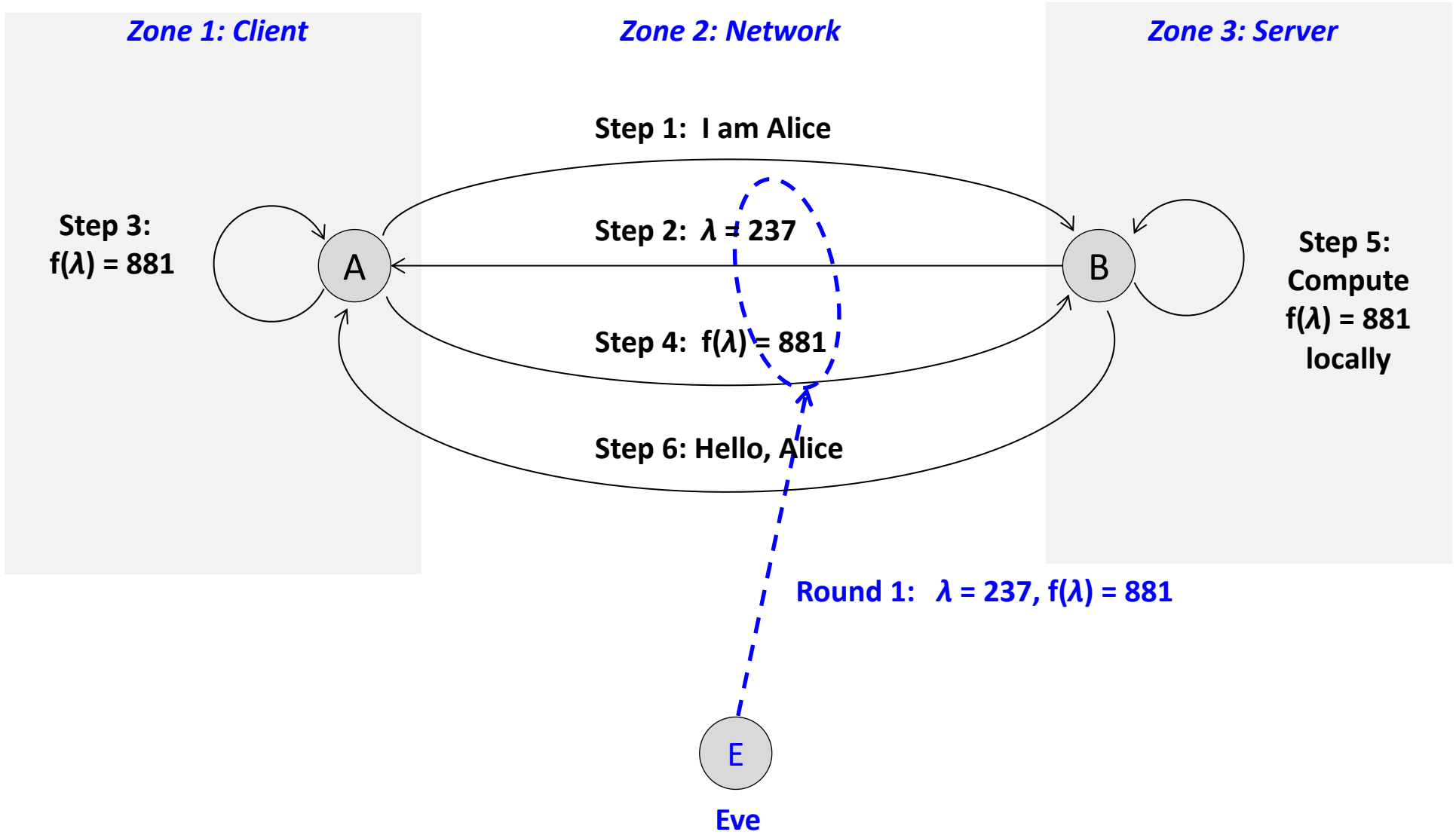


Embedded Function
f: integer -> integer

User	*Function*
A	f
C	f ’
G	f ’’
. . .	. . .

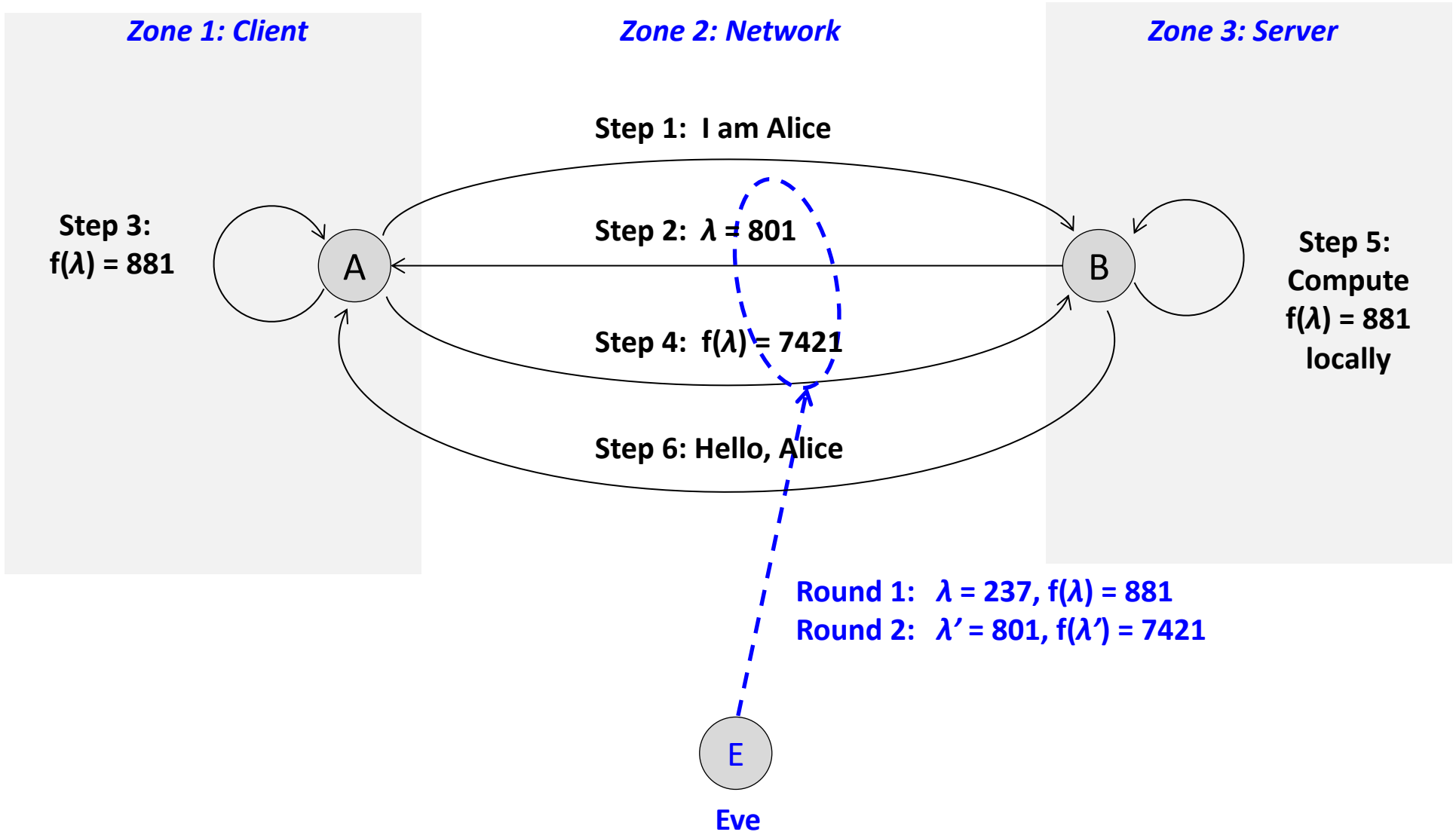
Week 5

Handheld Authentication Device Protocol



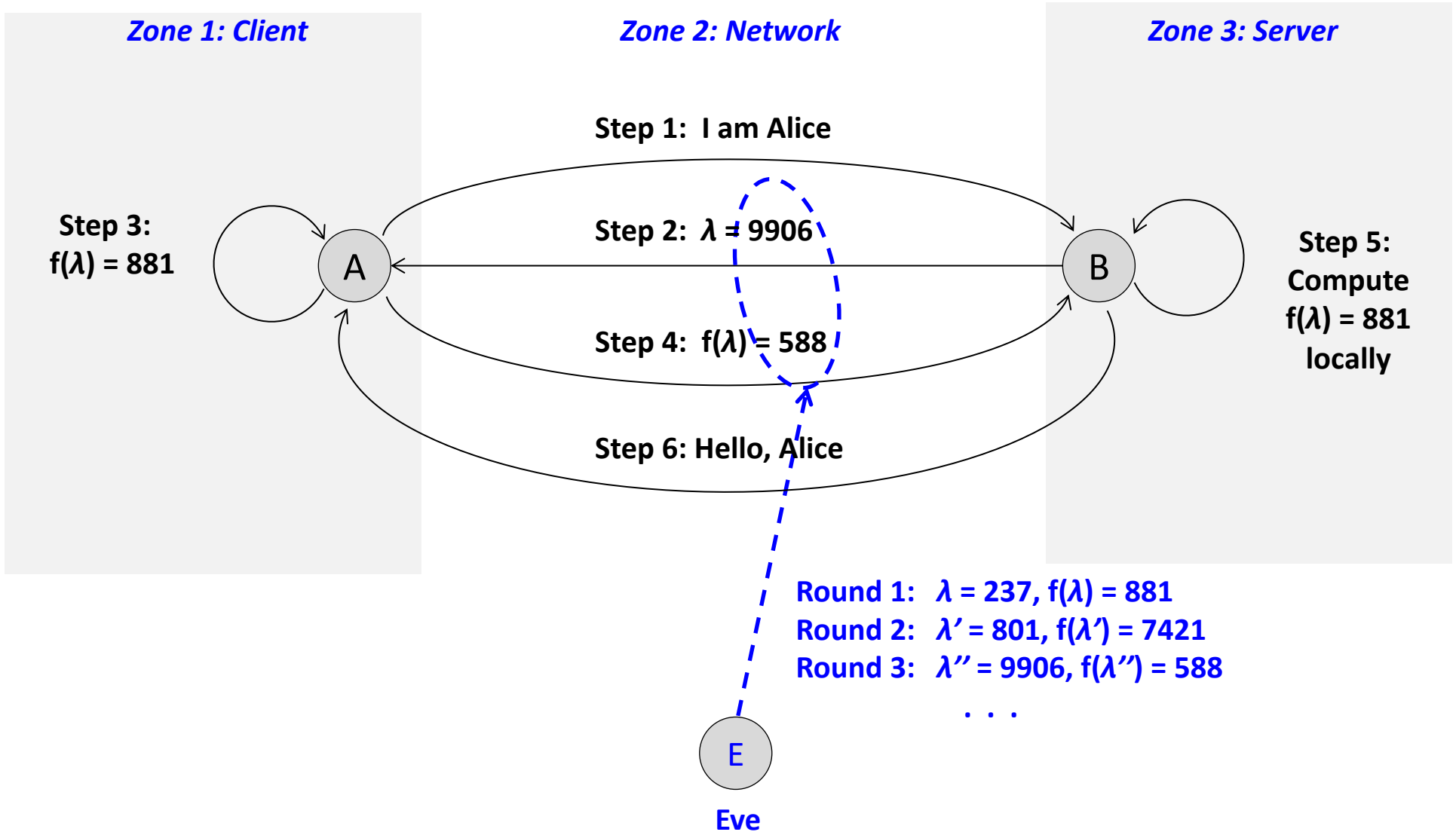
Week 5

Handheld Authentication Device Protocol



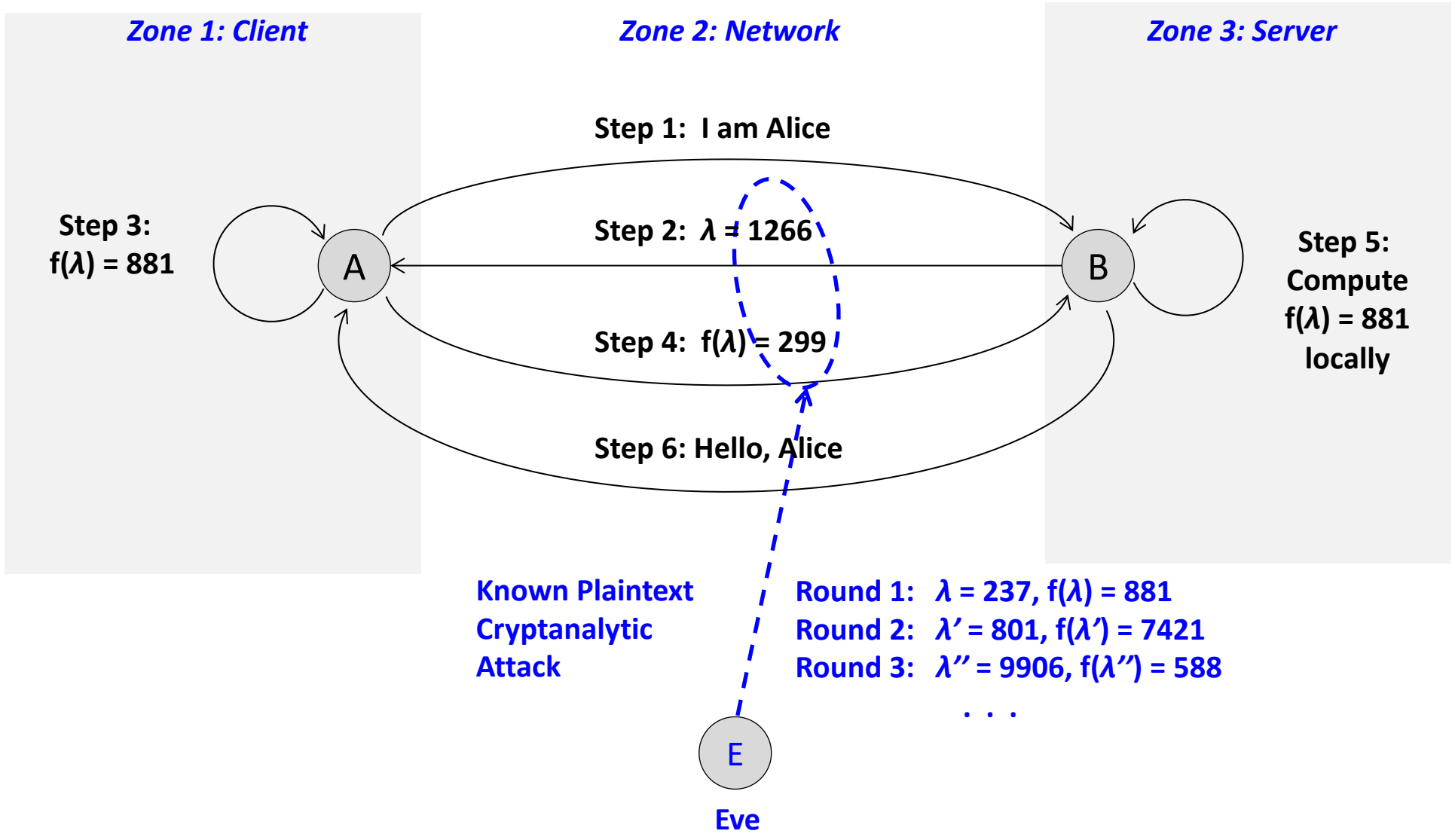
Week 5

Handheld Authentication Device Protocol



Week 5

Handheld Authentication Device Protocol

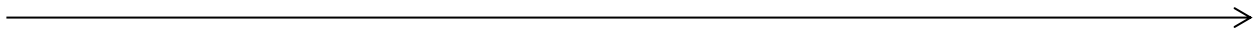


How Does RSA SecureID OTP Work?

Week 5

RSA SecurID One-Time Password (OTP) Algorithm

f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$



Week 5

RSA SecurID One-Time Password (OTP) Algorithm

f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$

Unique seed for each user

seed = λ

t_0 = 0 sec

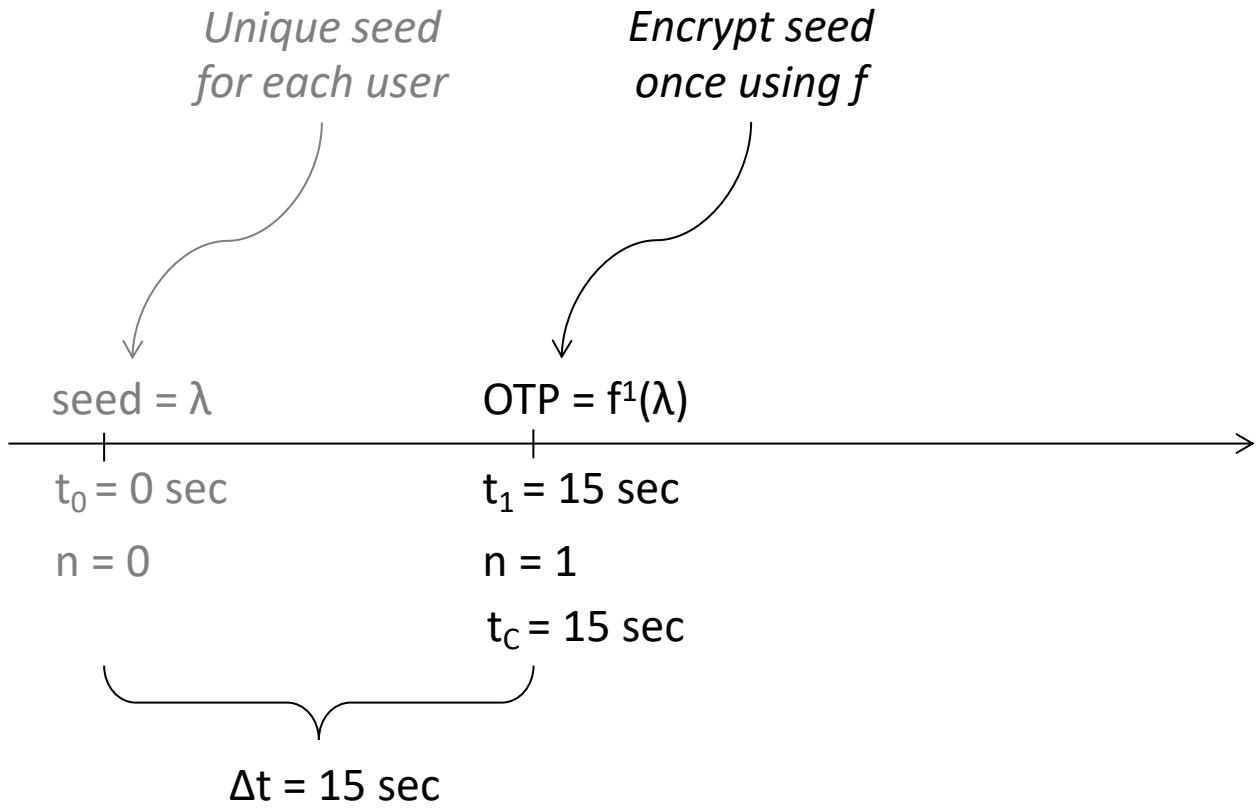
n = 0

t_C = 0 sec

Week 5

RSA SecurID One-Time Password (OTP) Algorithm

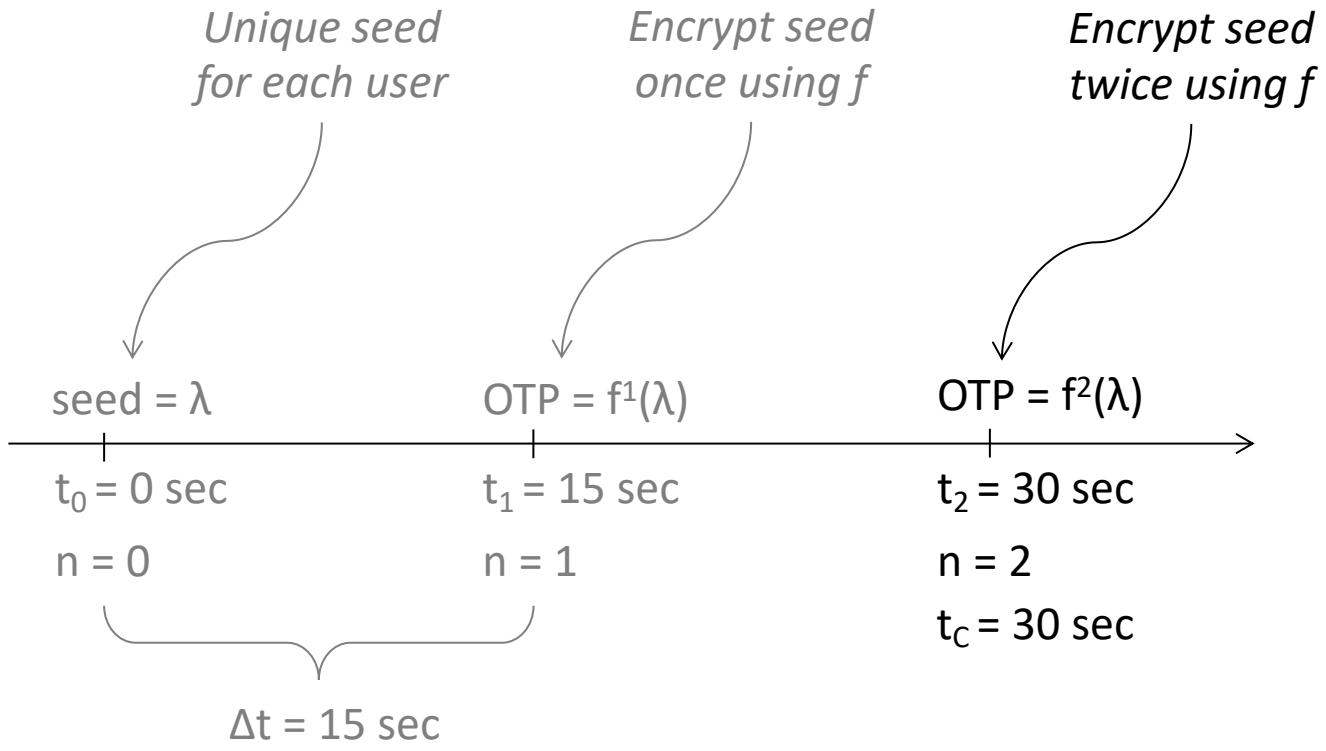
f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$



Week 5

RSA SecurID One-Time Password (OTP) Algorithm

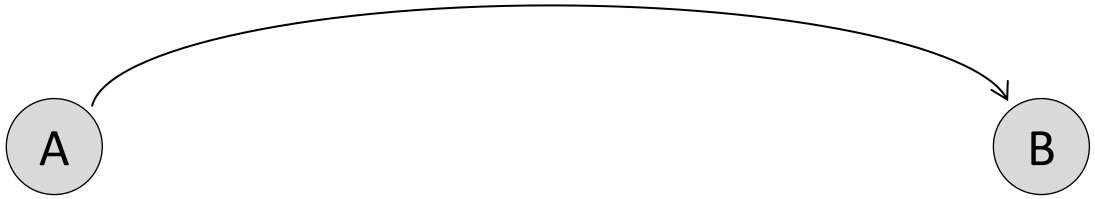
f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$



Week 5

RSA SecurID Protocol

Step 1: I am Alice

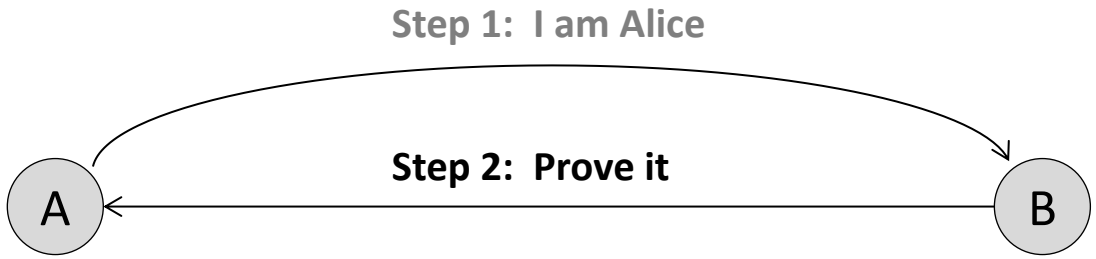


f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$

User	*Information*
A	f: integer -> integer λ: integer seed t_0: initial time t_C: current time Δt: time interval $n = (t_C - t_0) / \Delta t$

Week 5

RSA SecurID Protocol

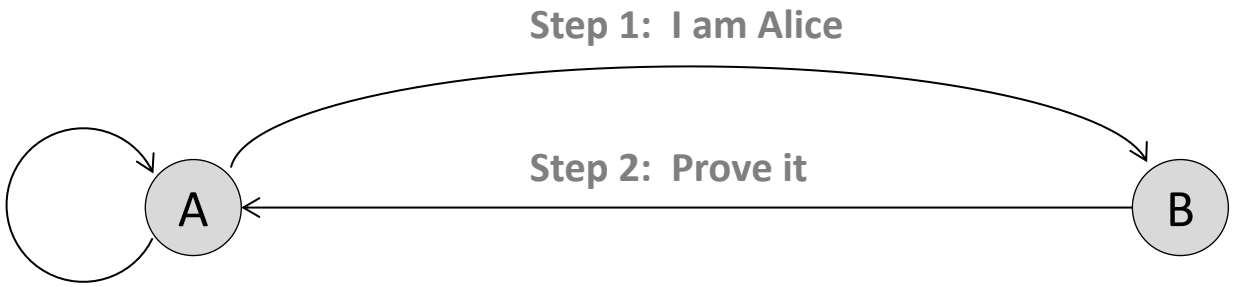


f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$

User	*Information*
A	f: integer -> integer λ: integer seed t_0: initial time t_C: current time Δt: time interval $n = (t_C - t_0) / \Delta t$

Week 5

RSA SecurID Protocol

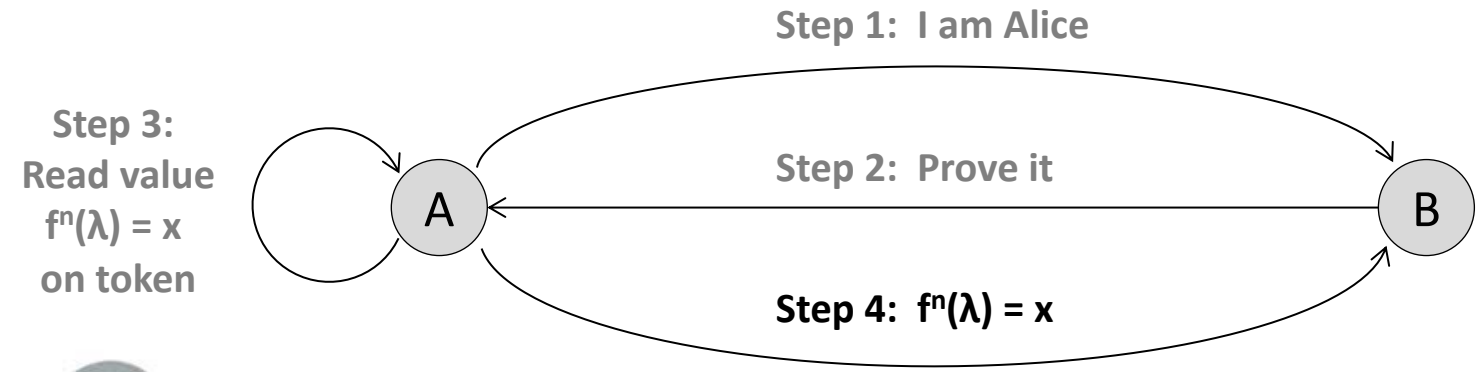


f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$

User	*Information*
A	f: integer -> integer λ: integer seed t_0: initial time t_C: current time Δt: time interval $n = (t_C - t_0) / \Delta t$

Week 5

RSA SecurID Protocol

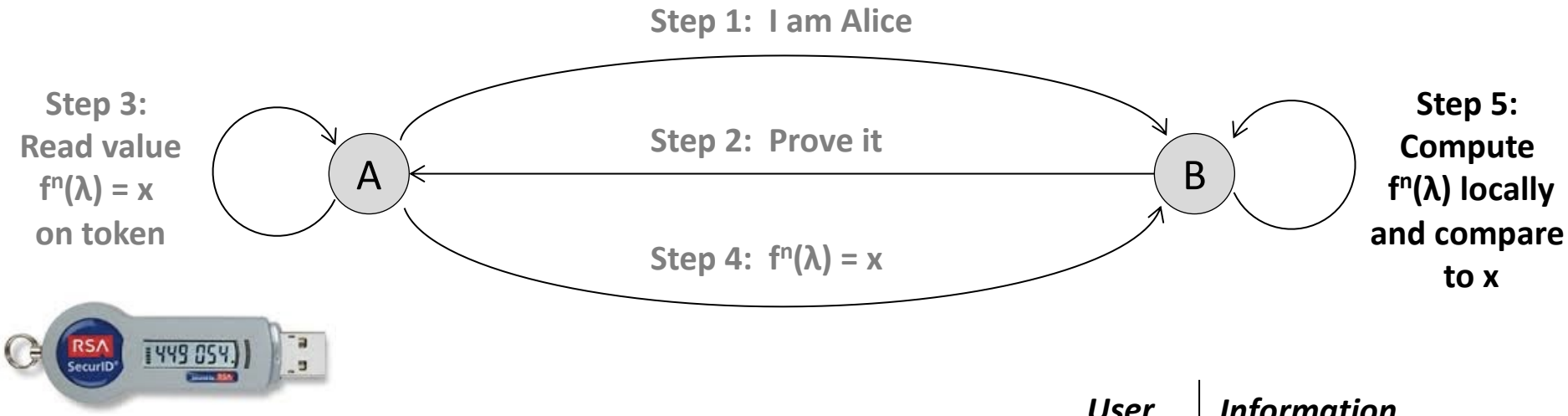


f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$

User	*Information*
A	f: integer -> integer λ: integer seed t_0: initial time t_C: current time Δt: time interval $n = (t_C - t_0) / \Delta t$

Week 5

RSA SecurID Protocol

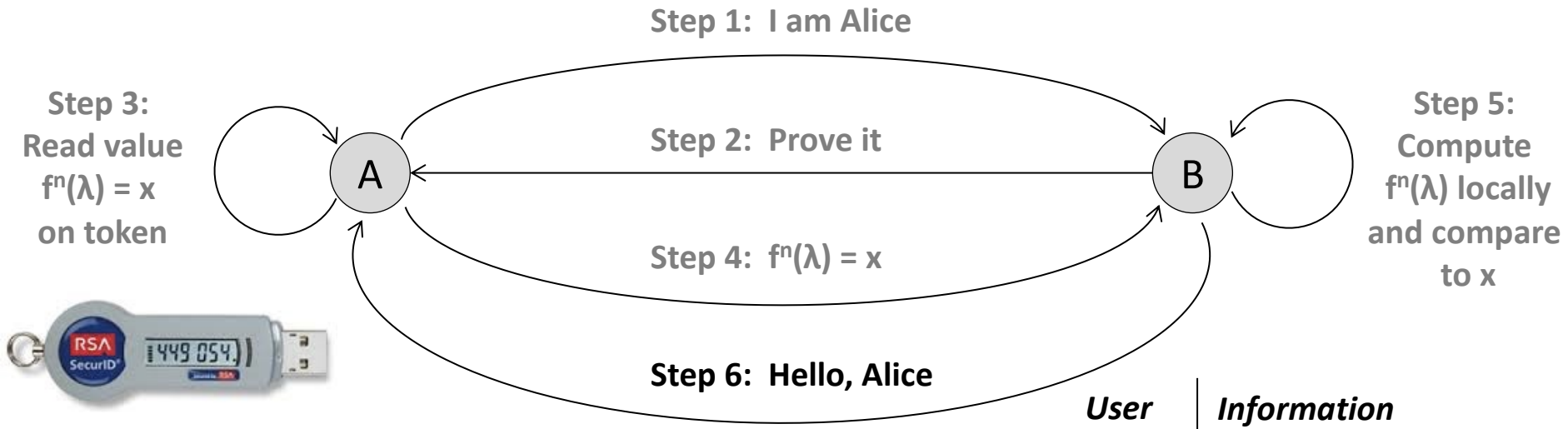


f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$

User	*Information*
A	f: integer -> integer λ: integer seed t_0: initial time t_C: current time Δt: time interval $n = (t_C - t_0) / \Delta t$

Week 5

RSA SecurID Protocol

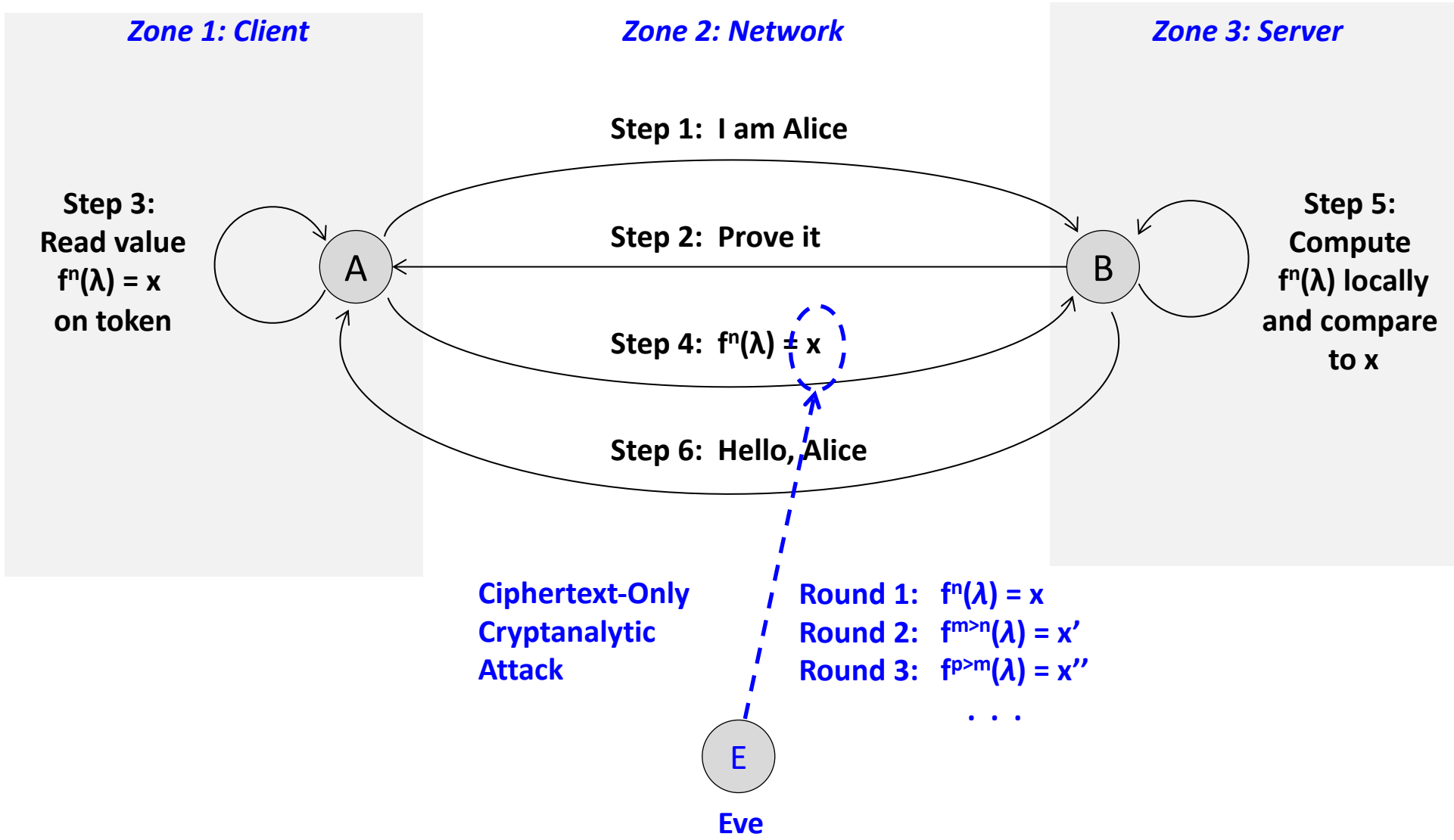


f: integer -> integer
λ: integer seed
t_0: initial time
t_C: current time
Δt: time interval
$n = (t_C - t_0) / \Delta t$

User	*Information*
A	f: integer -> integer λ: integer seed t_0: initial time t_C: current time Δt: time interval $n = (t_C - t_0) / \Delta t$

Week 5

RSA SecurID Protocol



Week 5

RSA SecurID App

Mac iPad iPhone Watch TV Music Support

App Store Preview

Open the Mac App Store to buy and download apps.

RSA SecurID Software Token 4+

RSA Security

Designed for iPhone

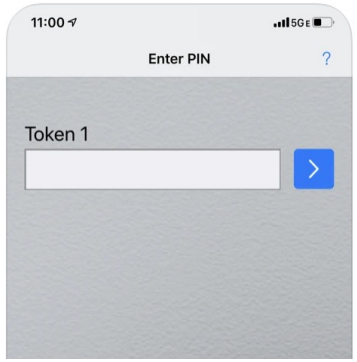
#69 in Business

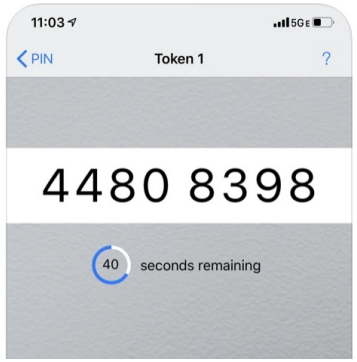
3.1 • 334 Ratings

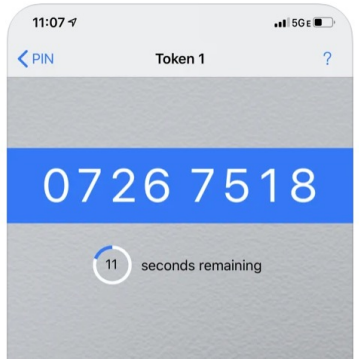
Free

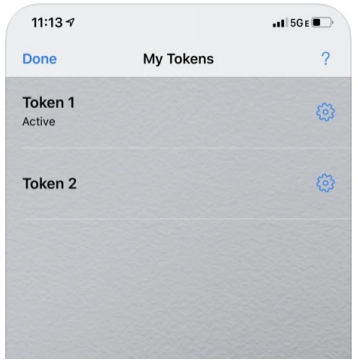
View in Mac App Store ↗

iPhone Screenshots



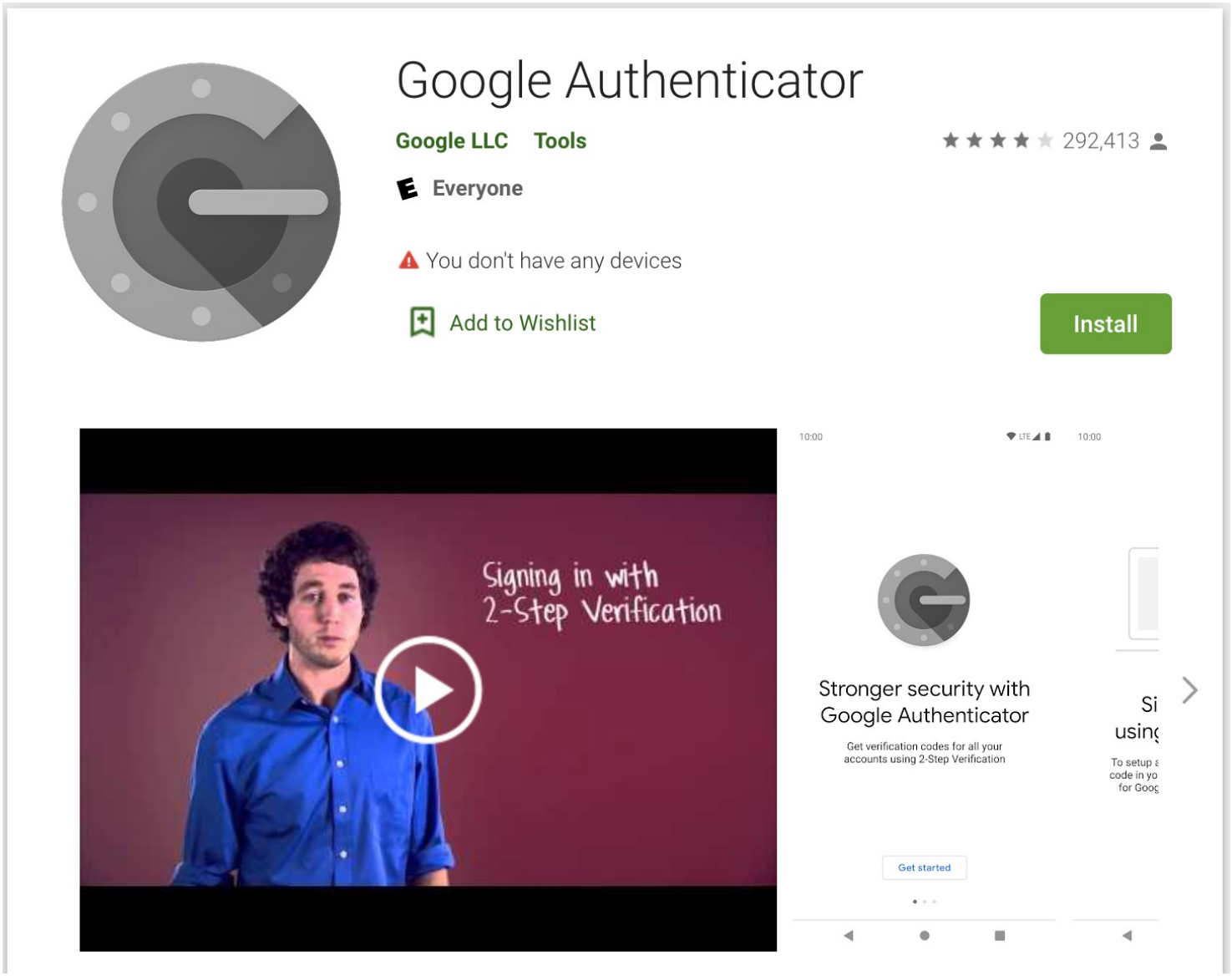






Week 5

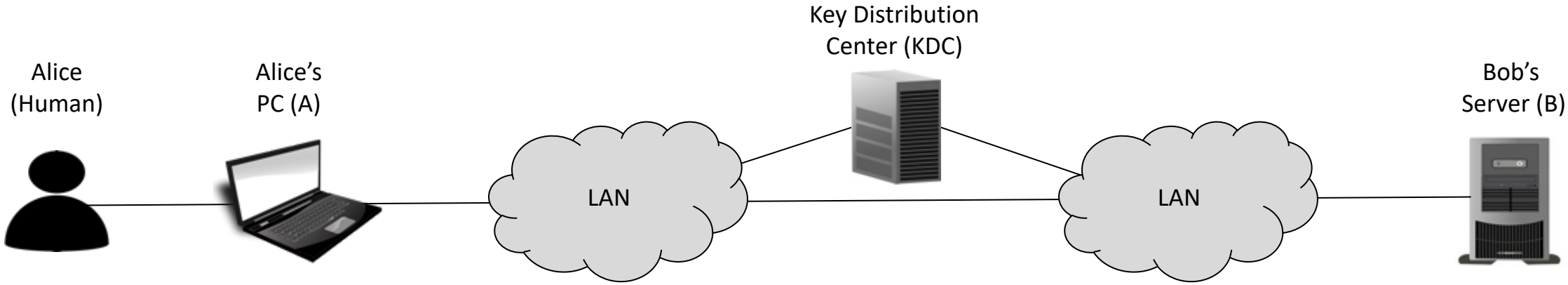
Google Authenticator App



How Does Kerberos Work?

Week 5

Kerberos: A Complex Solution to a Simple Password Problem



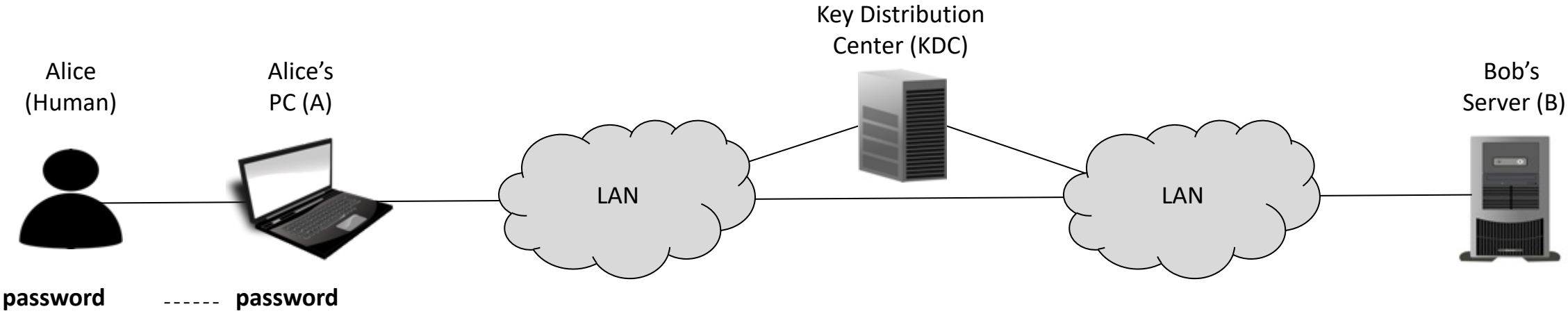
Basic Kerberos Concept:

- 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 5

Kerberos – Preconditions

Infrastructure Preconditions:

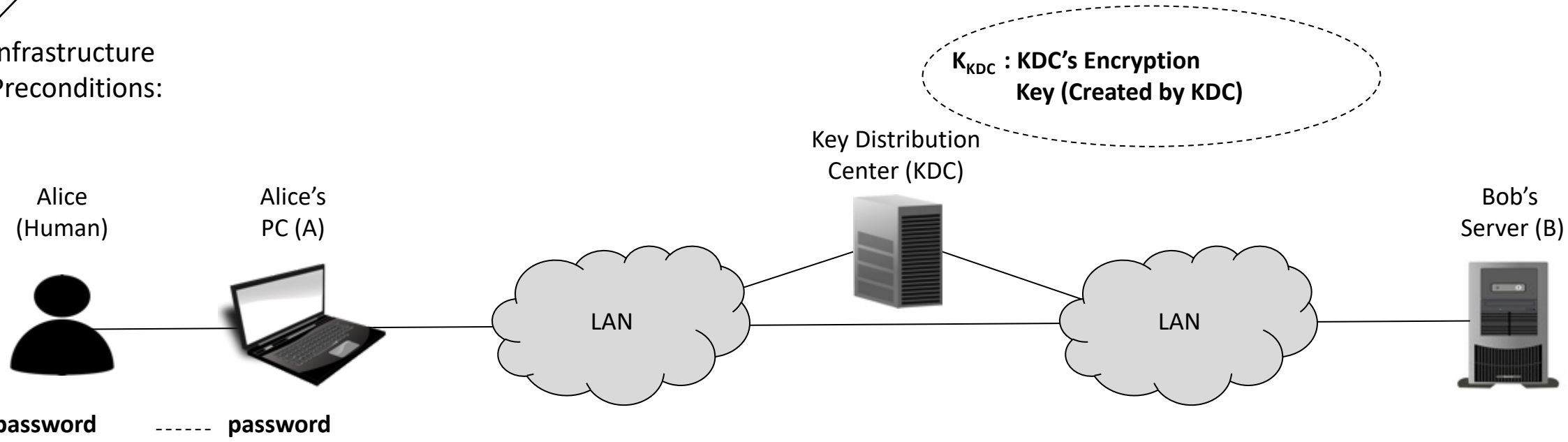


Kerberos password set up for Alice to log into her PC (Never used over any LAN)

Week 5

Kerberos – Keys

Infrastructure Preconditions:



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

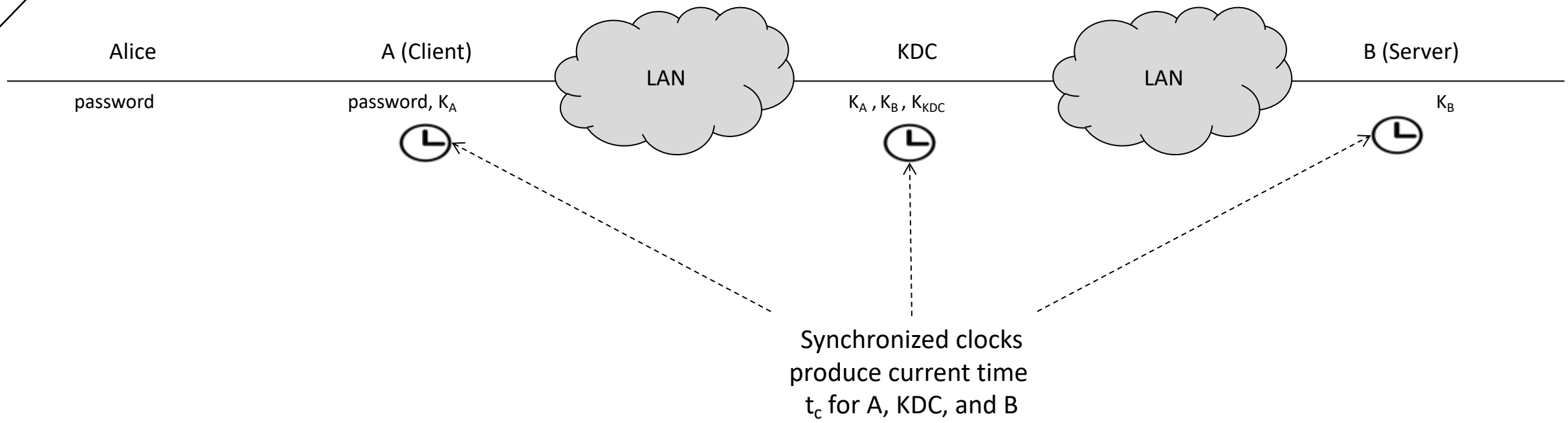
$\{\{ m \}_{K_A}\}_{K_A} = m$ $\quad \{\{ m \}_{K_B}\}_{K_B} = m$ $\quad \{\{ m \}_{K_{KDC}}\}_{K_{KDC}} = m$

Encrypt

Decrypt

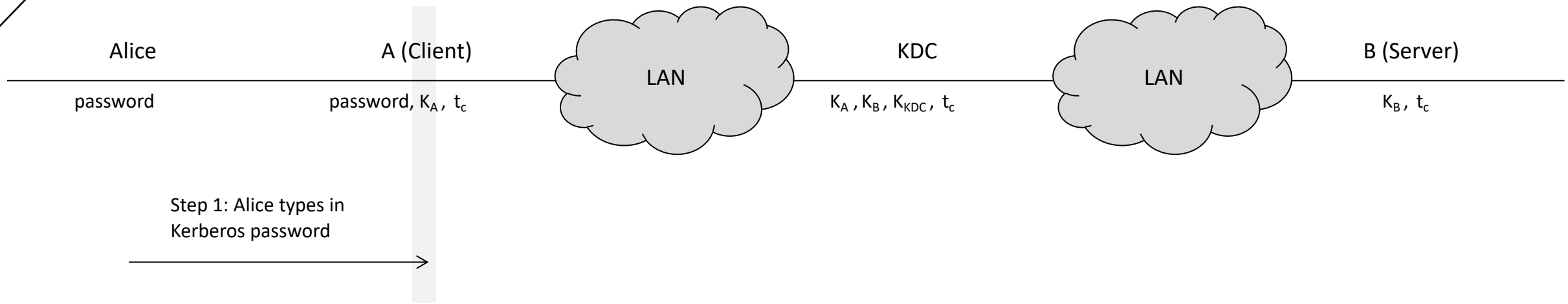
Week 5

Kerberos – Clocks



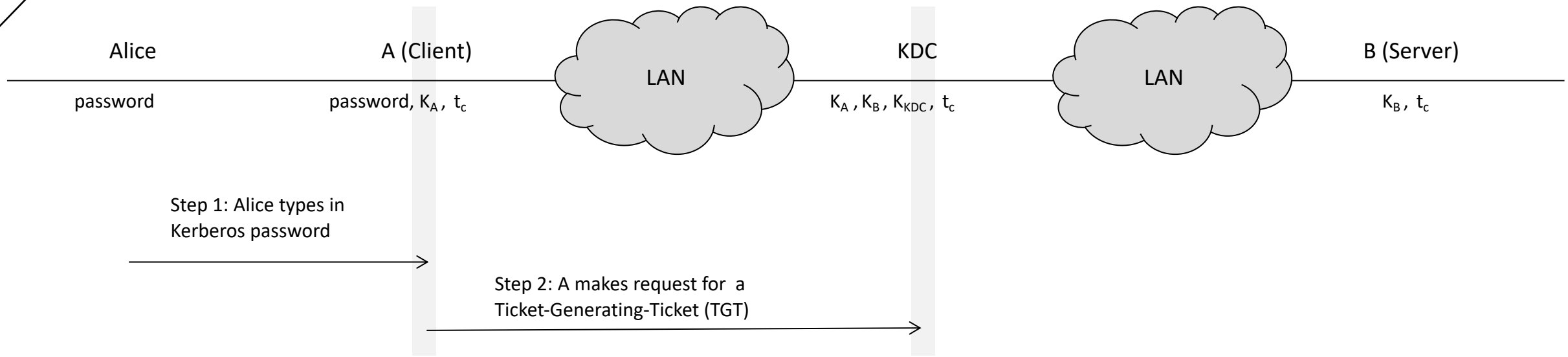
Week 5

Kerberos Step 1: Type in Local Password



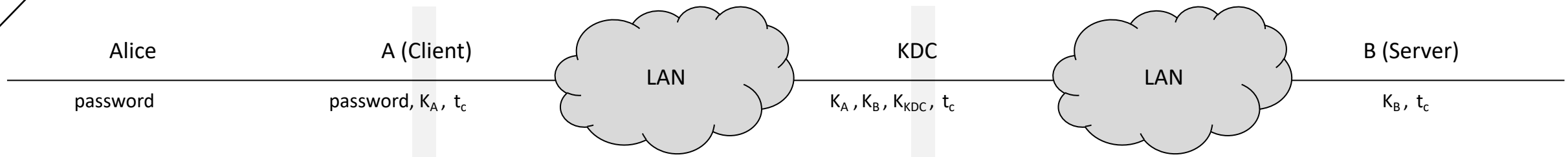
Week 5

Kerberos Step 2: Request TGT and Session Key



Week 5

Kerberos Step 3: Create Session Key and TGT



Step 1: Alice types in Kerberos password

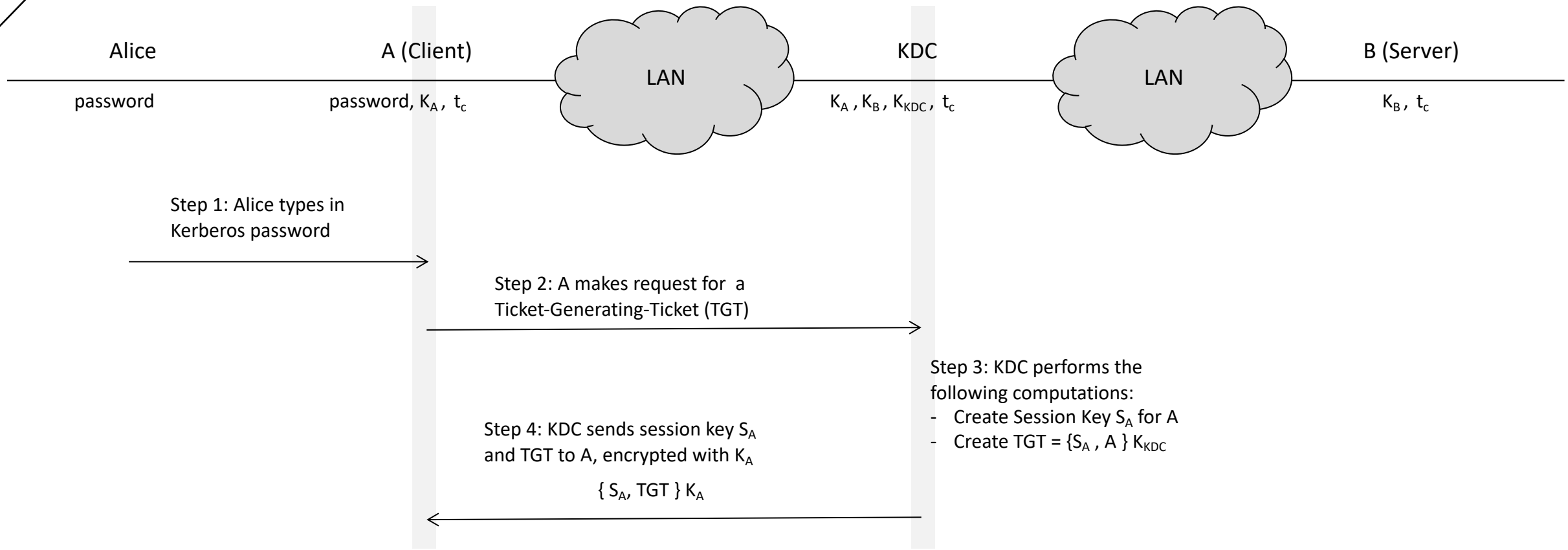
Step 2: A makes request for a Ticket-Generating-Ticket (TGT)

Step 3: KDC performs the following computations:

- Create Session Key S_A for A
- Create TGT = $\{S_A , A \} K_{KDC}$

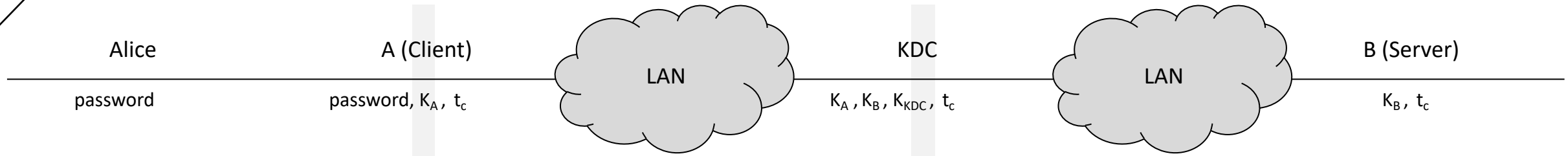
Week 5

Kerberos Step 4: Provide Session Key and TGT



Week 5

Kerberos Step 5: Decrypt Session Key and Store TGT



Step 1: Alice types in Kerberos password

Step 2: A makes request for a Ticket-Generating-Ticket (TGT)

Step 3: KDC performs the following computations:
- Create Session Key S_A for A
- Create TGT = $\{S_A , A \} K_{KDC}$

Step 4: KDC sends session key S_A and TGT to A, encrypted with K_A

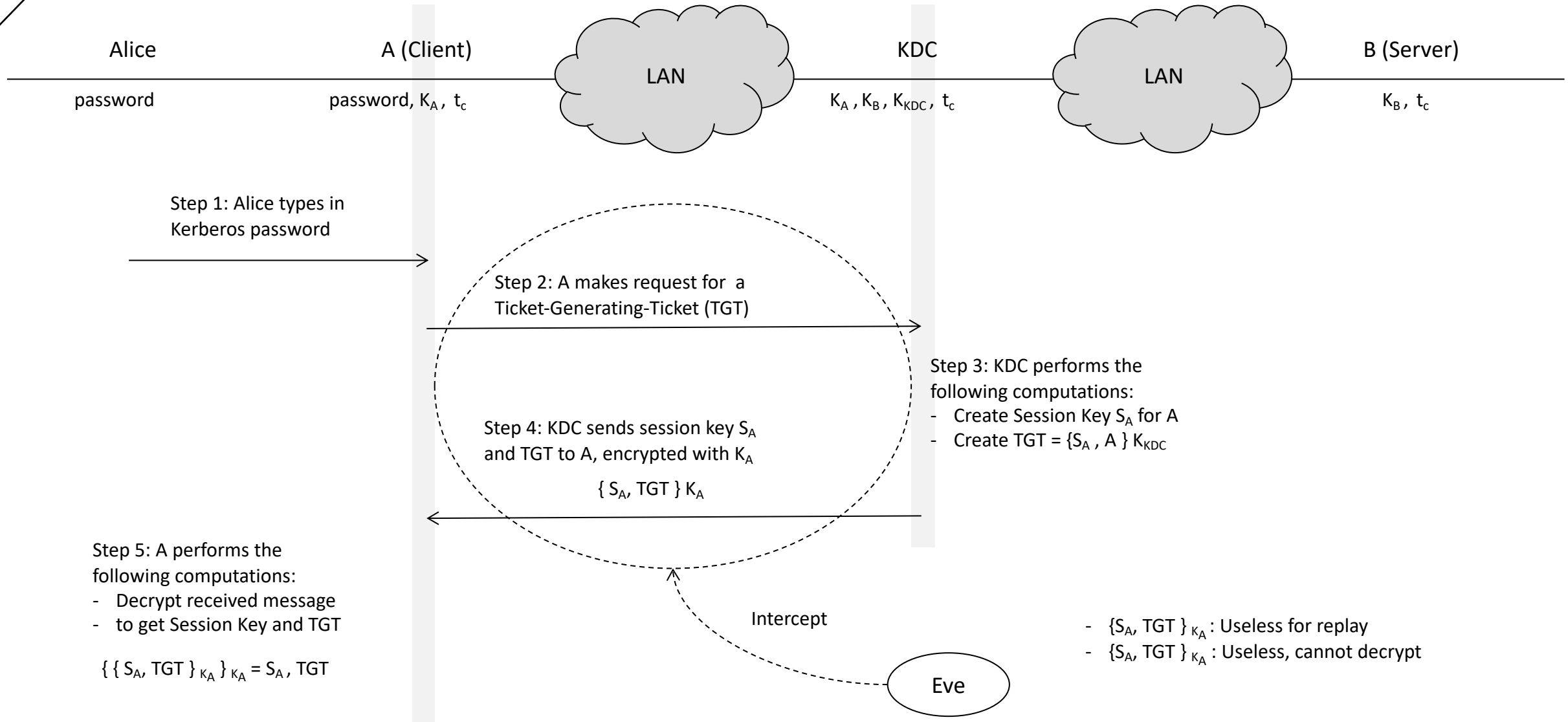
$\{ S_A, TGT \} K_A$

Step 5: A performs the following computations:
- Decrypt received message
- to get Session Key and TGT

$\{ \{ S_A, TGT \}_{K_A} \}_{K_A} = S_A , TGT$

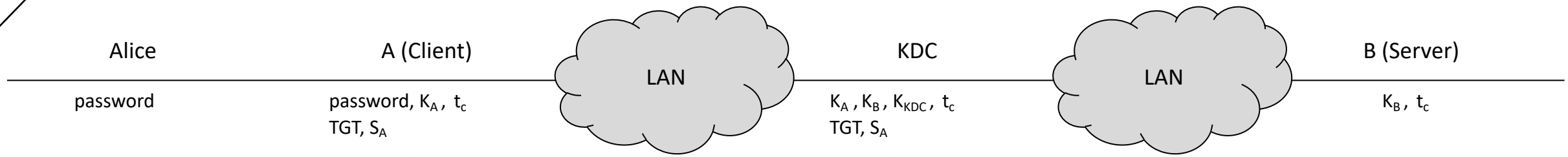
Week 5

Kerberos – Through Five Steps: Eve Cannot Hack



Week 5

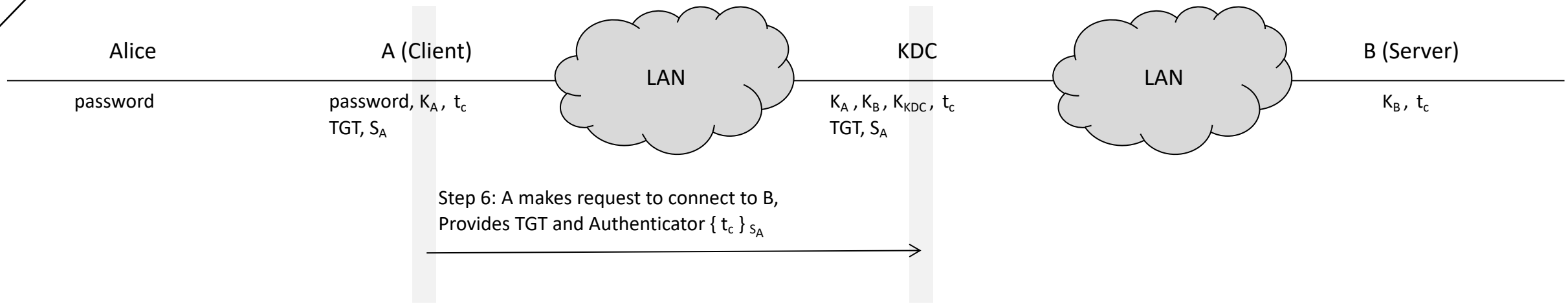
Kerberos – Result of Five Steps



Five step process for KDC to distribute TGT and S_A to A

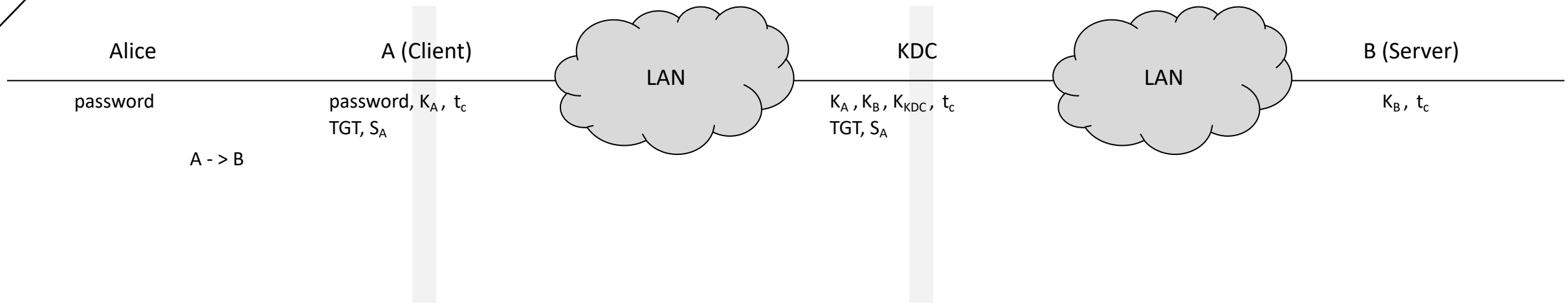
Week 5

Kerberos Step 6: Request Login to B



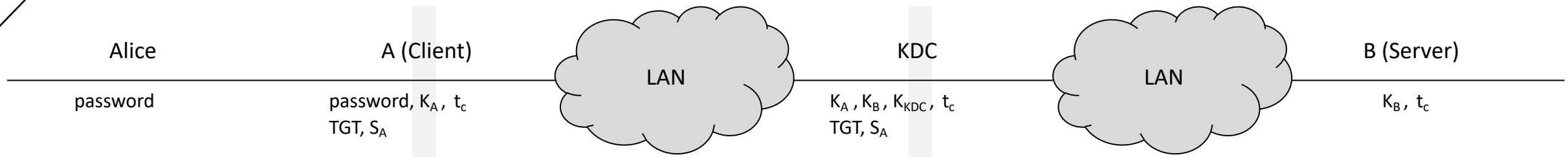
Week 5

Kerberos Step 6: Request Login to B



Week 5

Kerberos Step 7: Create Session Key and Ticket to Bob



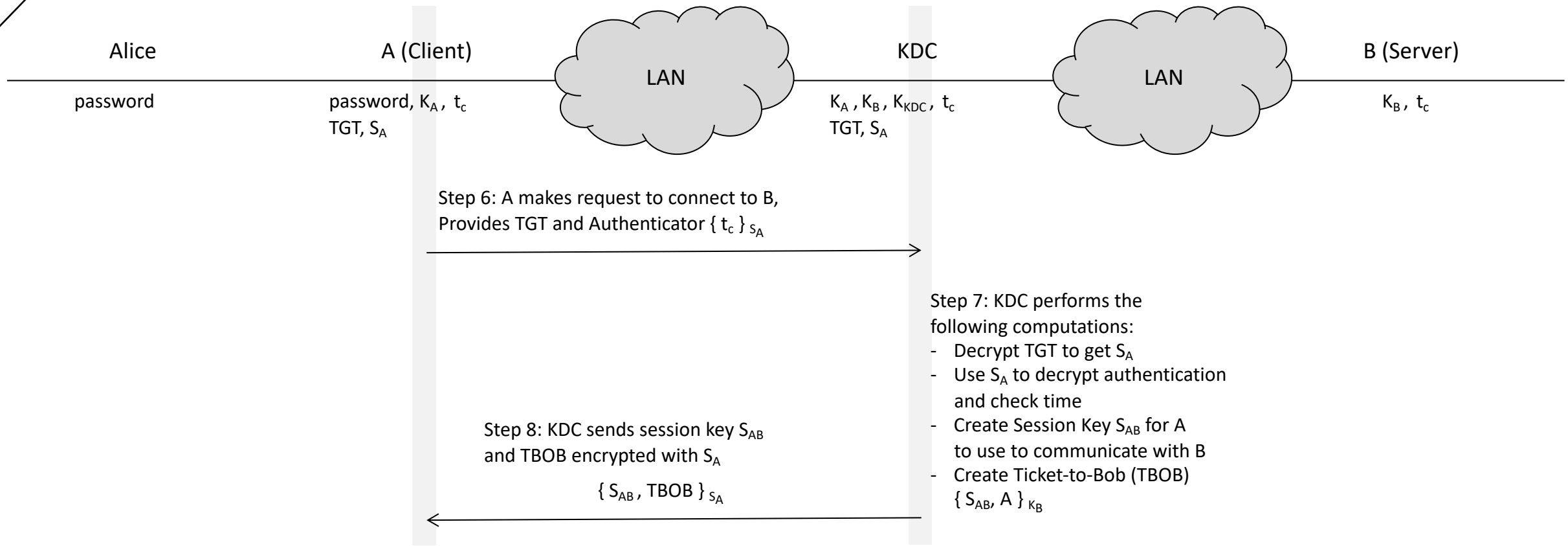
Step 6: A makes request to connect to B,
Provides TGT and Authenticator $\{ t_c \}_{S_A}$

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) $\{ S_{AB}, A \}_{K_B}$

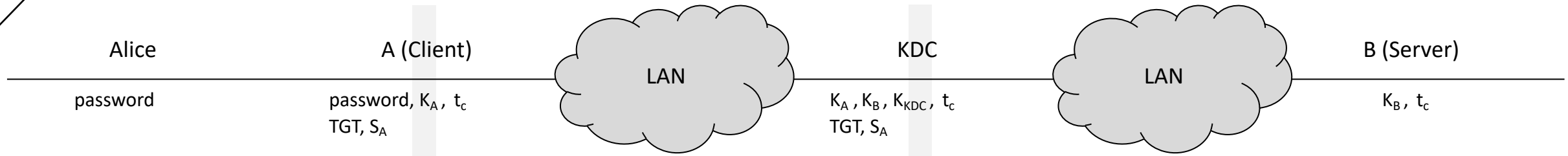
Week 5

Kerberos Step 8: Send Session Key and Ticket to Bob



Week 5

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



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

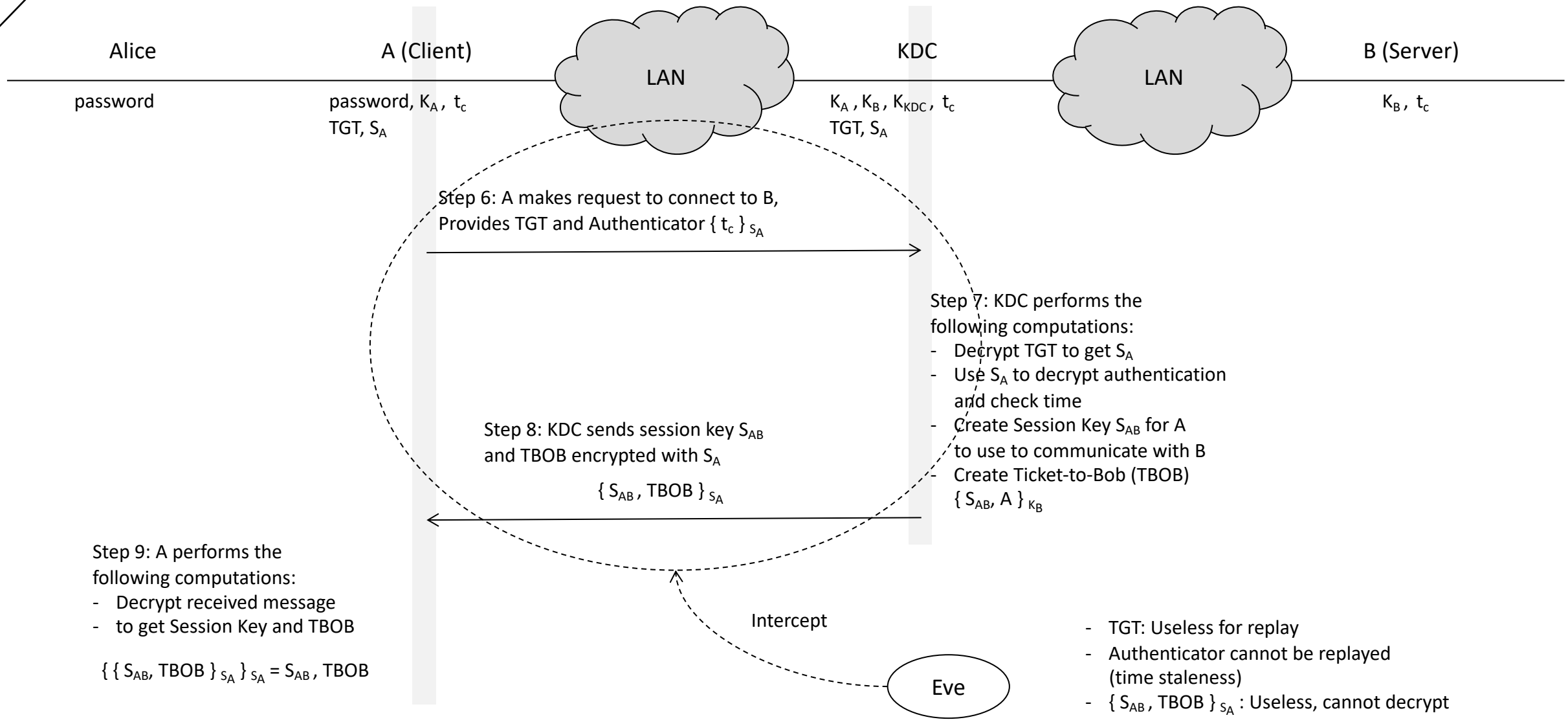
$\{ S_{AB} , TBOB \}_{S_A}$

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

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

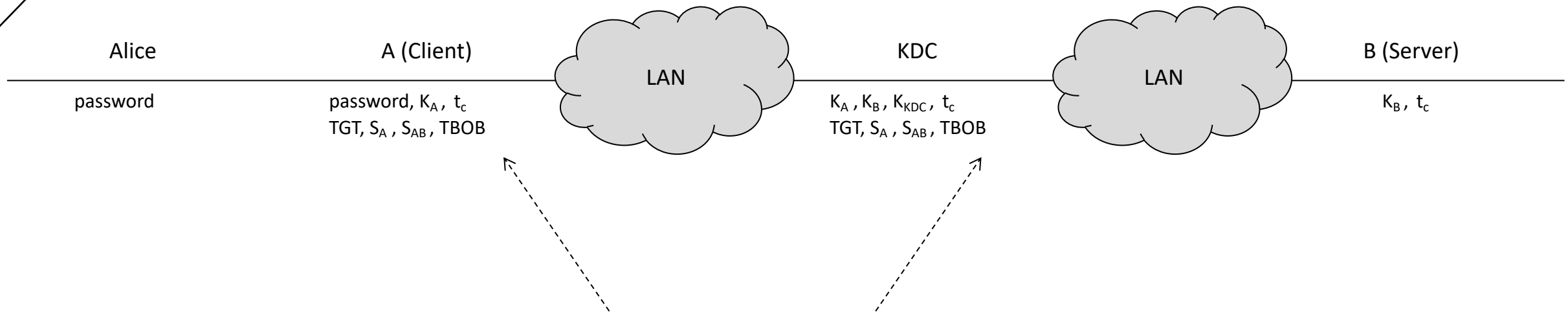
Week 5

Kerberos – Through Nine Steps: Eve Cannot Hack



Week 5

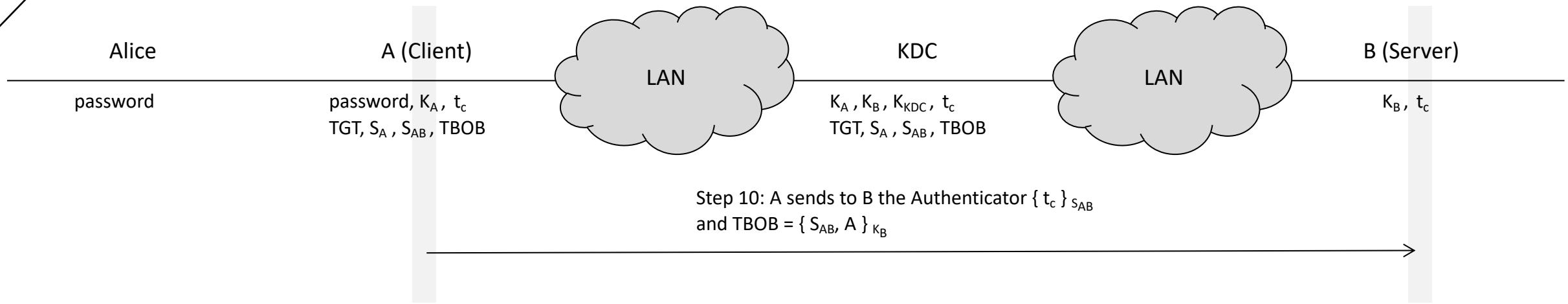
Kerberos – Result of Nine Steps



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

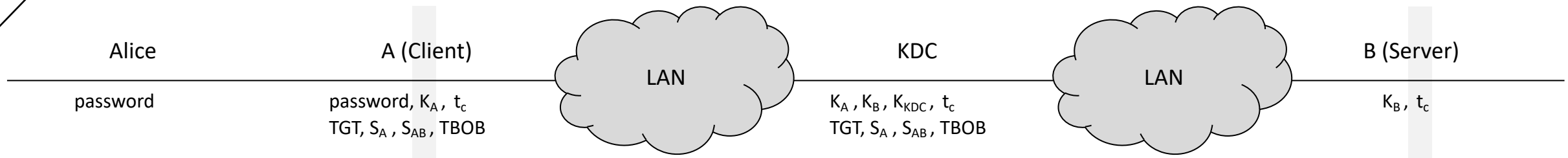
Week 5

Kerberos Step 10: Send Bob the Ticket to Bob



Week 5

Kerberos Step 11: Decrypt Ticket to Bob and Check Time



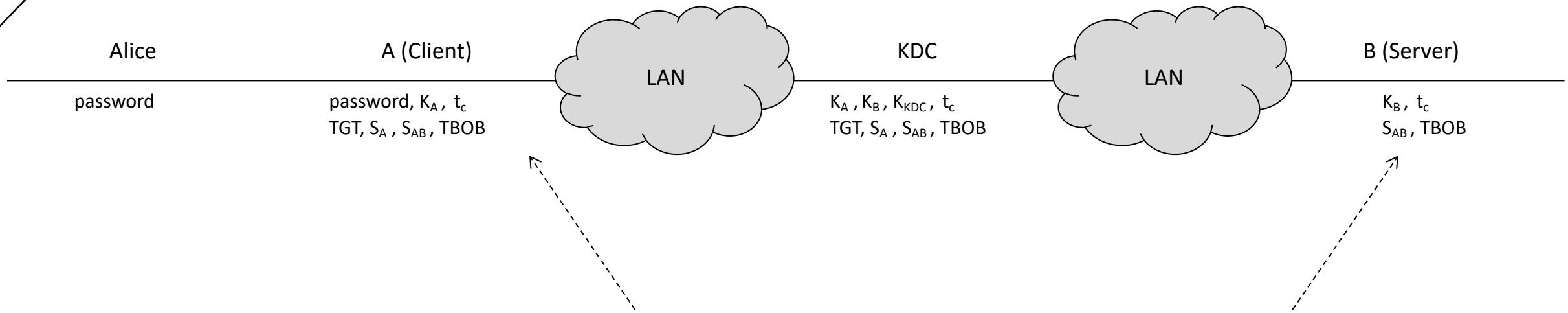
Step 10: A sends to B the Authenticator $\{ t_c \}_{S_{AB}}$
and TBOB = $\{ S_{AB}, A \}_{K_B}$

Step 11: B performs the following computations:

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

Week 5

Kerberos – Result of Eleven Steps



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

Week 5

Kerberos – Realms