

2021 TAG CYBER SECURITY ANNUAL



TIME TO BREAK UP THE RSA CONFERENCE

Harold Greene presided over a consent vote up the Bell System. While you might debate the innovations that followed. Just a year later, for example, we all watched as Steve Jobs on stage to demonstrate the new iPhone. I never invention, and many other advances called by the break-up.

me to the RSA Conference. I first started annual event in the mid-1990's, and believe it reference during that period was both relevant made real news, held real fights (remember accepted real technical papers by real have one of those iconic RSA Conference the mid-90's showing RSA as the only listens to its customers." Awesome.

er, the RSA Conference has devolved into a for mid-lifers with booth-after-booth-after-same-old, same-old. And just as with AT&T in tion was not caused by bad leadership, but terribly unavoidable corporate condition: -curve. I believe the RSA Conference has the top part of that scary curve, which is to take action.

ledge that the RSAC corporate ownership, m committees, and its expert conference d will explain that they've evolved the point to the new programs, sessions, and on and on. But look - AT&T said o's back in 1983. They were just as overve suspect RSA leadership will be to my tion. No one likes change, really.

will be bad for business. Steve Martin, for his stand-up act back in the 70s when he couple of new empty seats in the back d recommend that RSAC leadership act while the conference is still strong. If they on now, then RSAC will continue its slide using the cyber security equivalent of (ohem) an Show in Vegas.

recommend RSAC leadership should sunset advisory board (no offense to my friends). It

The RSA Conference has devolved into a routine event for mid-lifers with booth-after-booth-after-booth of the same-old, same-old.



should then create five new program committees with no member over twenty-nine and at least two-thirds women. These five new committees should then caucus over beers outside Whaler's to reinvent five crazy-interesting conferences with themes that are meaningful and edgy. They should push the envelope.

Then the PCs should reinvent how these five new S-curves are physically held. It could be something cool like those crowdsourced, simulcast, conference-BNB things. Maybe it could involve using the headquarters of security companies from around the world. Instead of having physical booths at Missone, vendors could host concurrent RSAC three-day parties for anyone who chooses to come to their venue. Or whatever. It would be fun.

Look - I know this would be a jolt, but if RSAC continues on its present path, then here is my prediction: Within three years, the RSA Conference will book less than 20K paid attendees, and it will start to lose its grip on the vendor community. Perhaps worse, the current show is really turning into a BoomerCon. Just like Spot the Fed at DFCON, RSAC could initiate a Spot the Non-Boomer contest. It would be quite a challenge.

By the way, Black Hat is the new RSA Conference. Just look at this sponsorship page for a conference that started as anti-establishment. Rich Powell and I developed a cartoon to lampoon this inevitable transition. You see, Black Hat is riding up the middle of its S-Curve. It is still somewhat edgy, and still somewhat relevant. In a few years, it probably be whining that they please stop kicking their conference can down the road.

Oh - and there's this: RSAC 2020 attendance looked to our TAG Cyber team to be about 50% down. This had nothing to do with the conference and everything to do with the virus. But it is precisely such random events that can trigger a downfall. Some security vendor or enterprise team might notice, for example, that the earth continues to rotate despite not having been at RSAC. This leads to a decision next year to maybe... well, you get the idea.

I believe that breaking up RSAC into five new conferences is good business for the owners and healthy for our industry. Even the venerable AT&T, where I spent most of my adult life, thrived mightily post-divestiture despite decades of fighting the courts. If RSAC ownership wants to protect its investment, then they will listen to my advice. If they don't - well, at least RSAC 2025 will be easier to navigate, because no one will be there.

I hope they listen.



Required Additional Reading: <https://www.tag-cyber.com/advisory/annuals>

Case Study: Brute Force Cryptanalytic Attack

Plaintext:

Loren ipsum is a pseudo Latin
text used in typography

*Classic Caesar Cipher
(Shift +2 in Alphabet)*



Ciphertext:

Nqtgp kruwo ku c rugwfq Ncvjp
Vgzv wugf kp varqitcrjb

Case Study: Brute Force Cryptanalytic Attack

Plaintext:

Loren ipsum is a pseudo Latin
text used in typography

*Classic Caesar Cipher
(Shift +2 in Alphabet)*

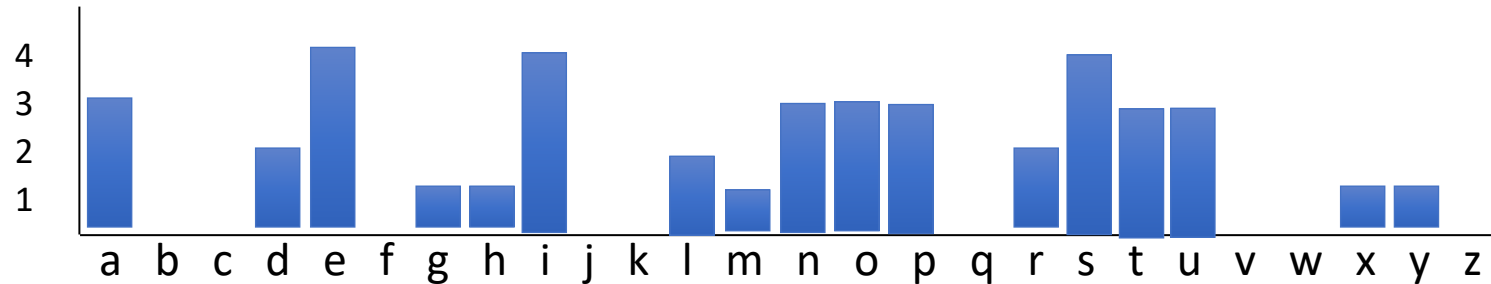


Ciphertext:

Nqtgp kruwo ku c rugwfq Ncvjp
Vgzv wugf kp varqitcrjb

Plain Text Character Distribution:

(Approximates Frequency Distribution of English Language with More Data)



Case Study: Brute Force Cryptanalytic Attack

Plaintext:

Loren ipsum is a pseudo Latin
text used in typography

*Classic Caesar Cipher
(Shift +2 in Alphabet)*

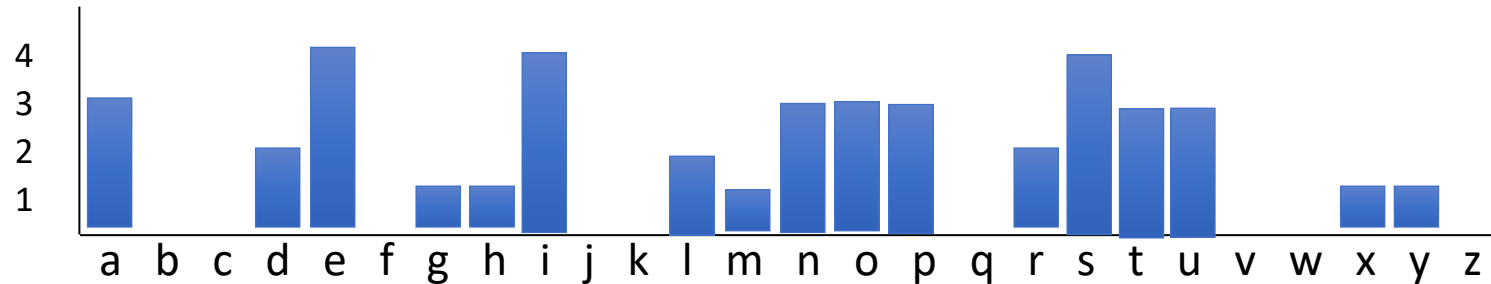


Ciphertext:

Nqtdgp kruwo ku c rugwfg Ncvjp
Vgzv wugf kp varqitcrjb

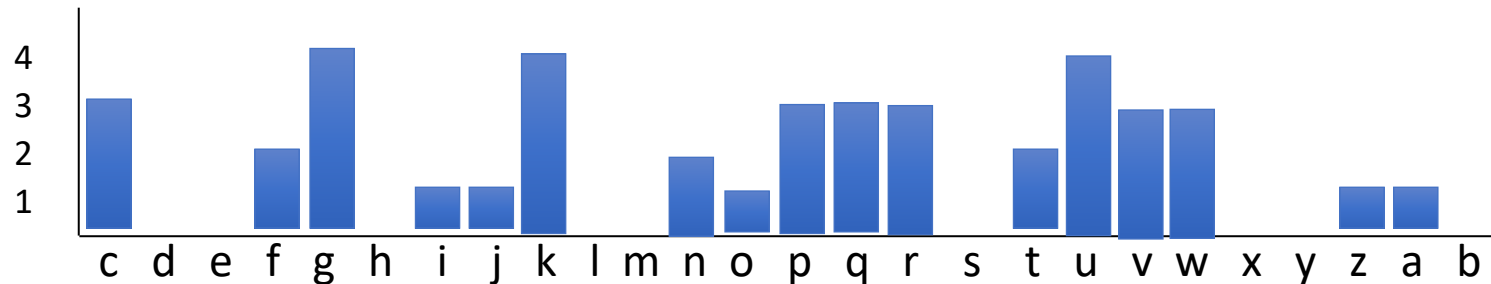
Plain Text Character Distribution:

(Approximates Frequency Distribution of English Language with More Data)

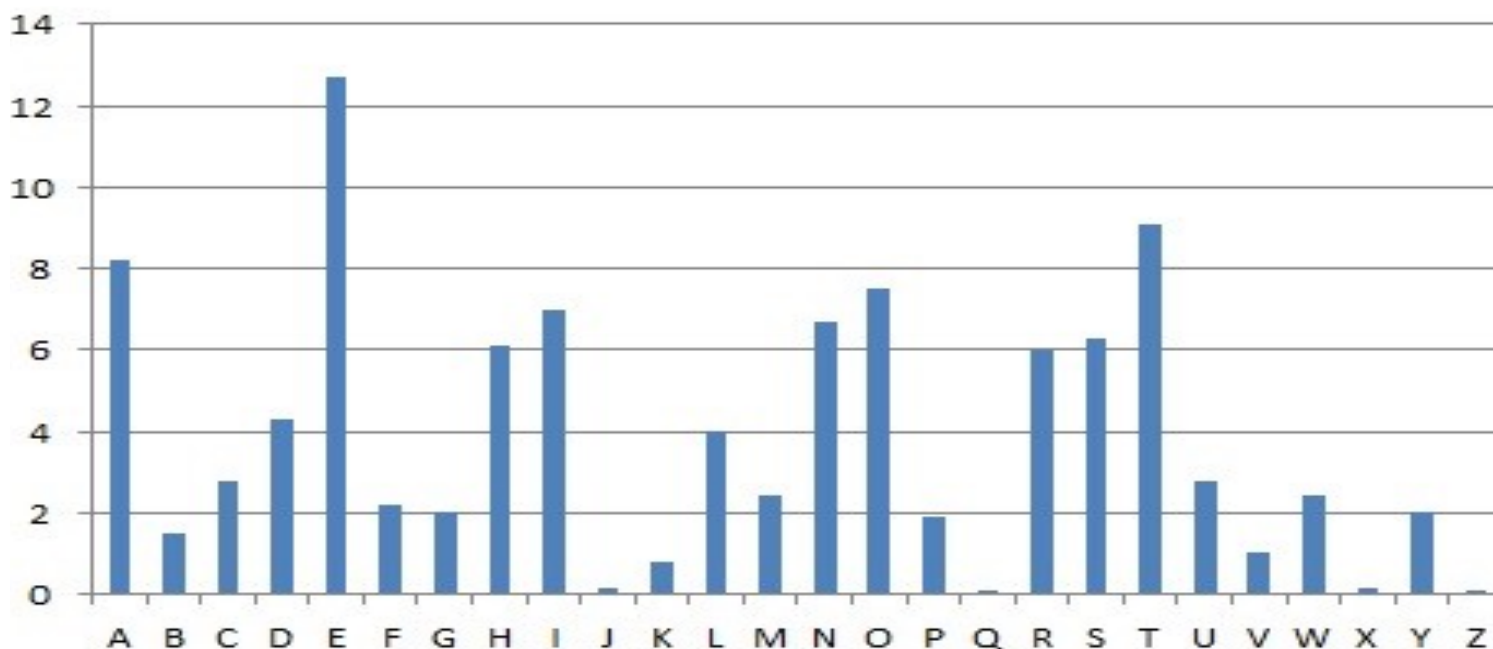


Ciphertext Character Distribution:

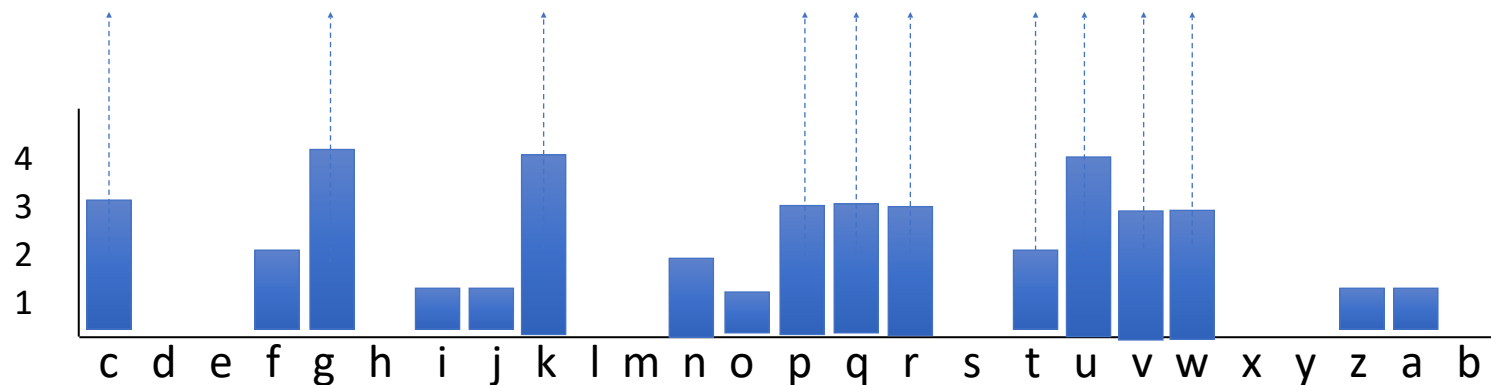
(Frequency Distribution Exposes Caesar Cipher Character Mapping)



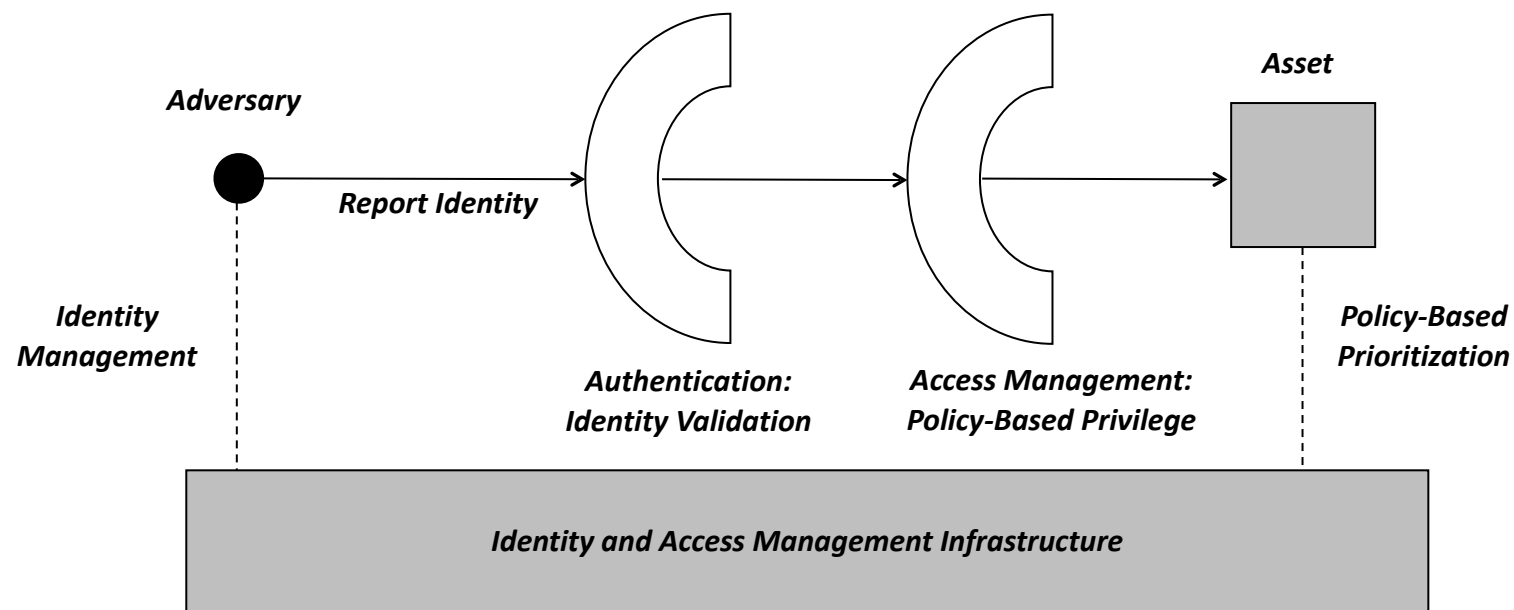
Case Study: Brute Force Cryptanalytic Attack



*Actual
Frequency
Distribution*

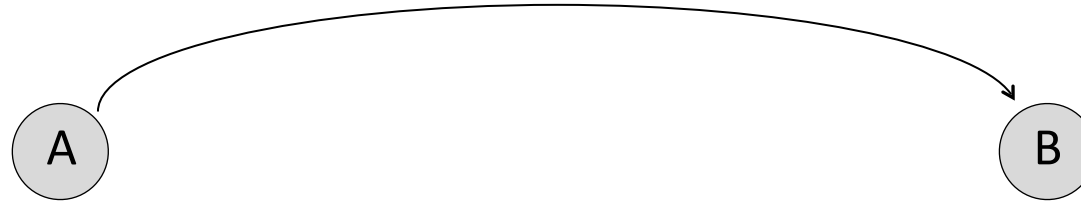


Safeguard: Authentication



Authentication Schema

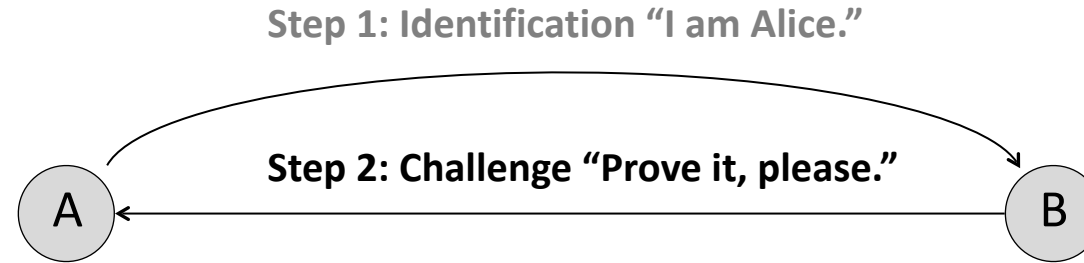
Step 1: Identification “I am Alice.”



Client A – Server B: “Client Authentication”

Client B – Server A: “Server Authentication”

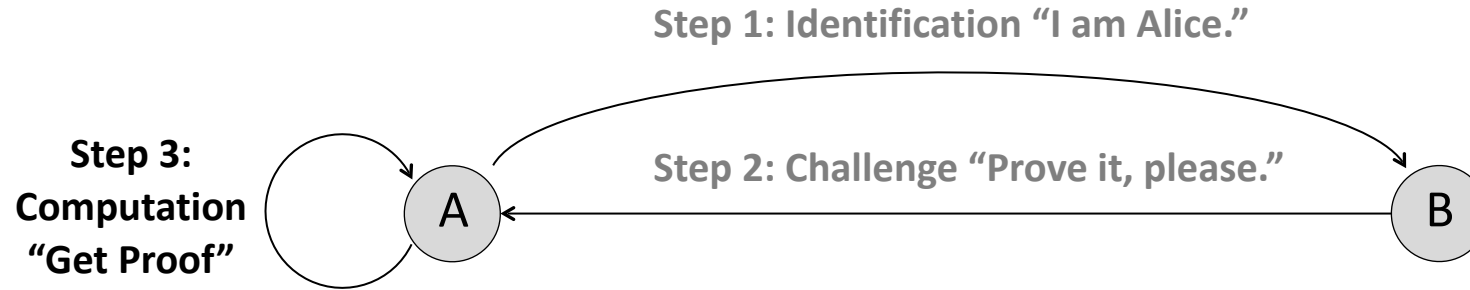
Authentication Schema



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

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

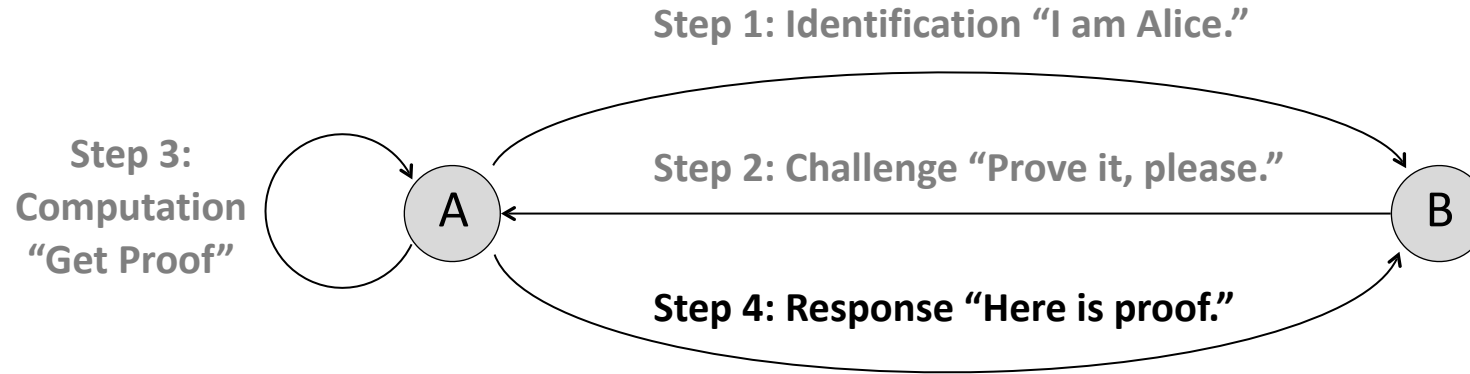
Authentication Schema



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

Computation might be more deliberate mathematical operation on domain value

Authentication Schema



Types of Proof:

"Something You Know" – Passwords

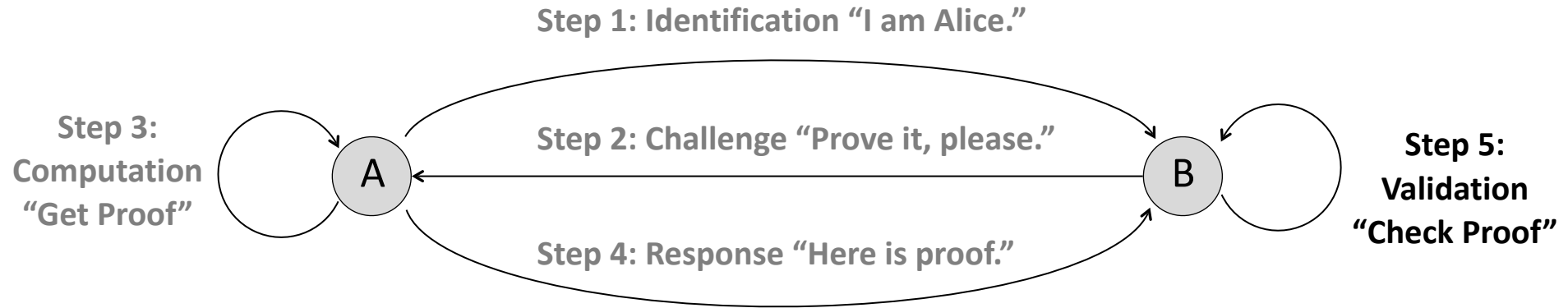
"Something You Are" – Biometrics

"Something You Have" – Token

"Somewhere You Are" – Location

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

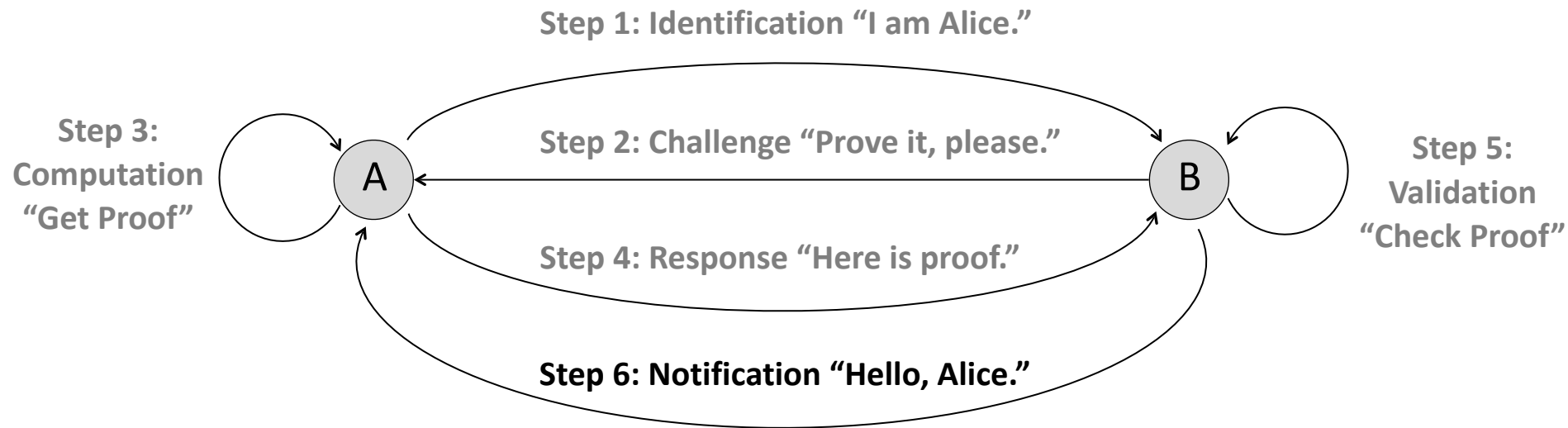
Authentication Schema



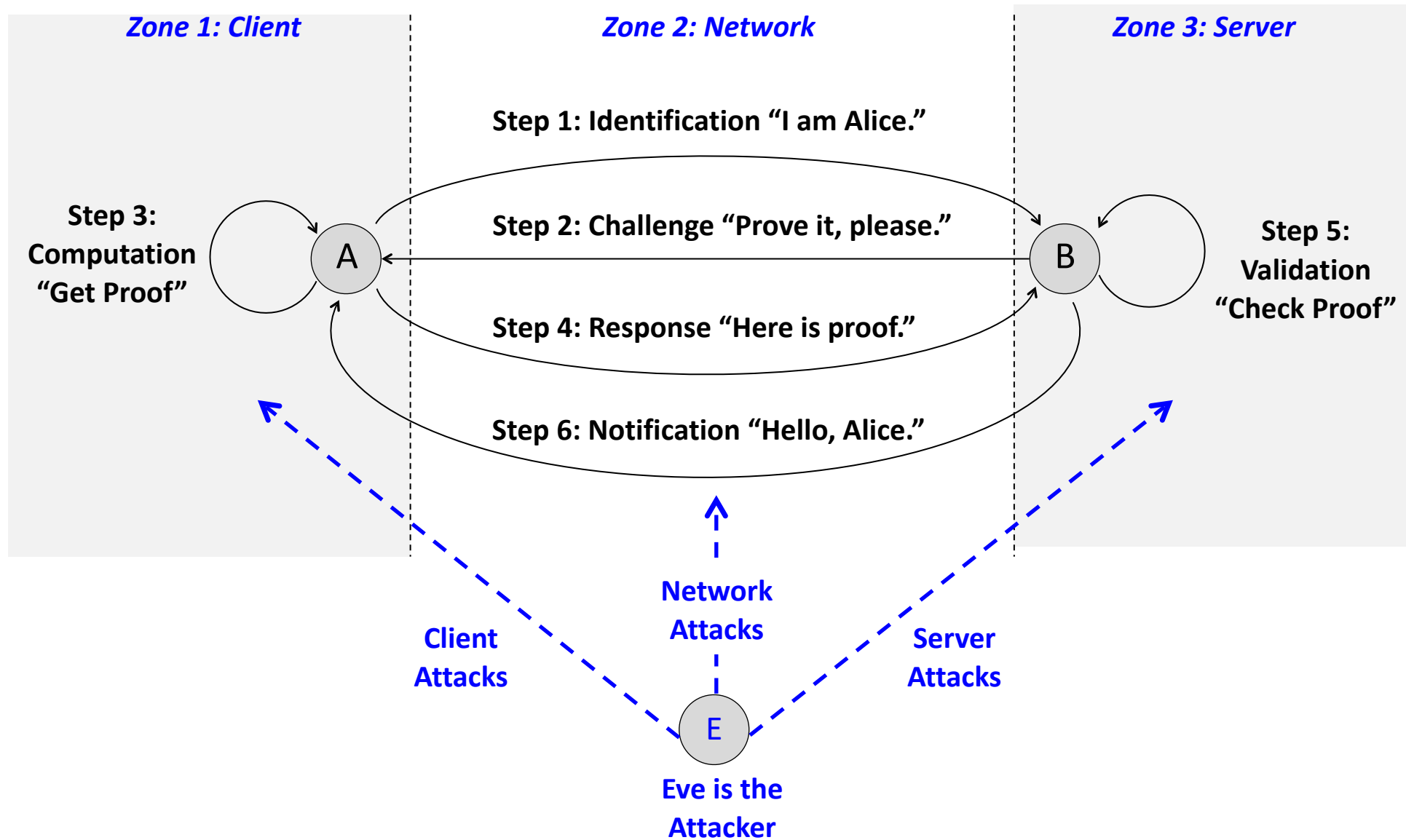
Validation might involve simple look-up/locate process (e.g., passwords)

Validation might be more deliberate mathematical operation on domain value

Authentication Schema



Authentication Schema



Handheld Authentication Device

A

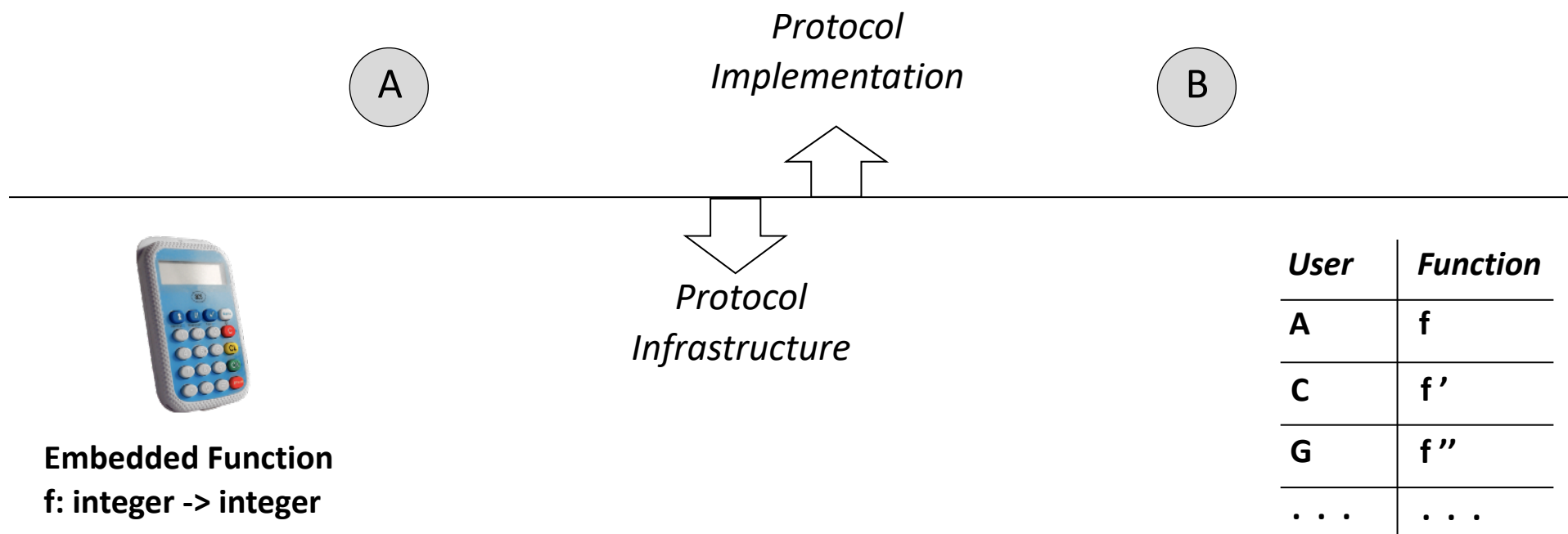
B



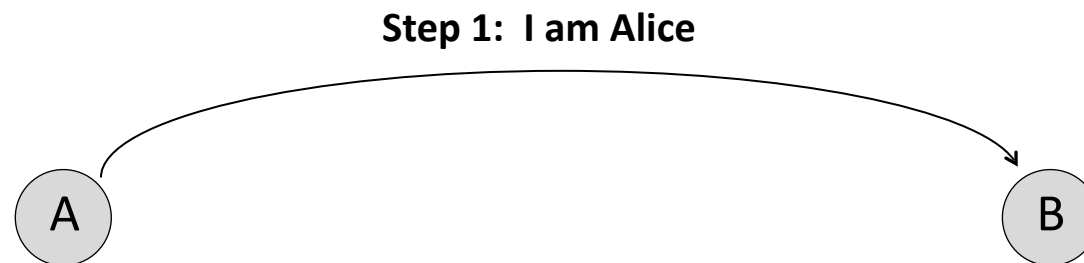
Embedded Function
 $f: \text{integer} \rightarrow \text{integer}$

<i>User</i>	<i>Function</i>
A	f
C	f'
G	f''
.

Handheld Authentication Device



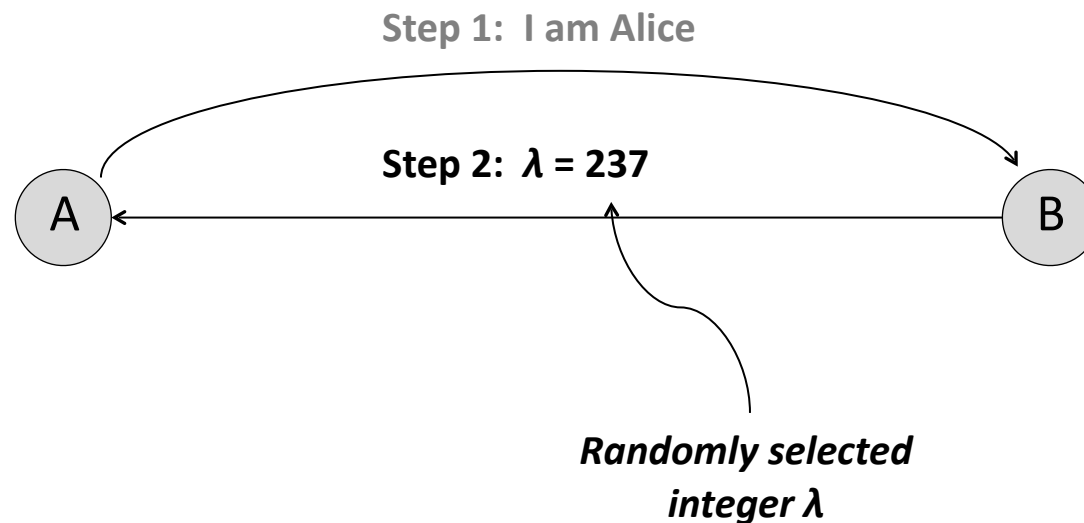
Handheld Authentication Device



Embedded Function
 $f: \text{integer} \rightarrow \text{integer}$

<i>User</i>	<i>Function</i>
A	f
C	f'
G	f''
...	...

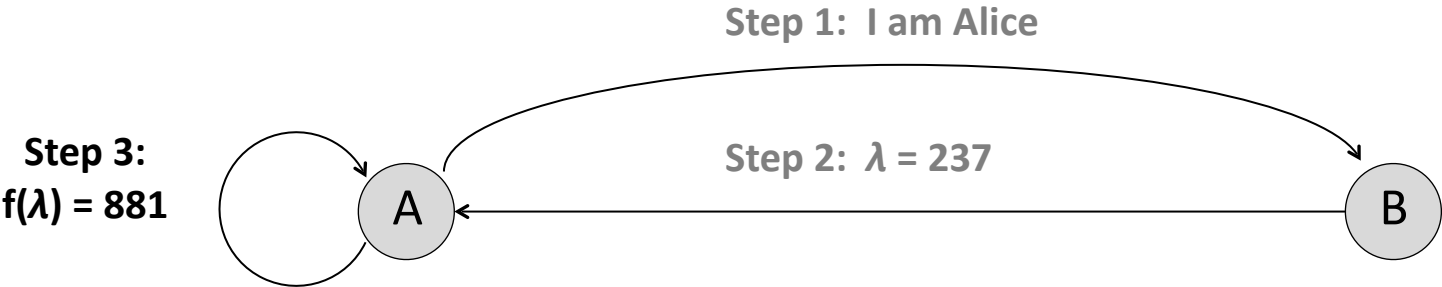
Handheld Authentication Device



Embedded Function
 $f: \text{integer} \rightarrow \text{integer}$

<i>User</i>	<i>Function</i>
A	f
C	f'
G	f''
...	...

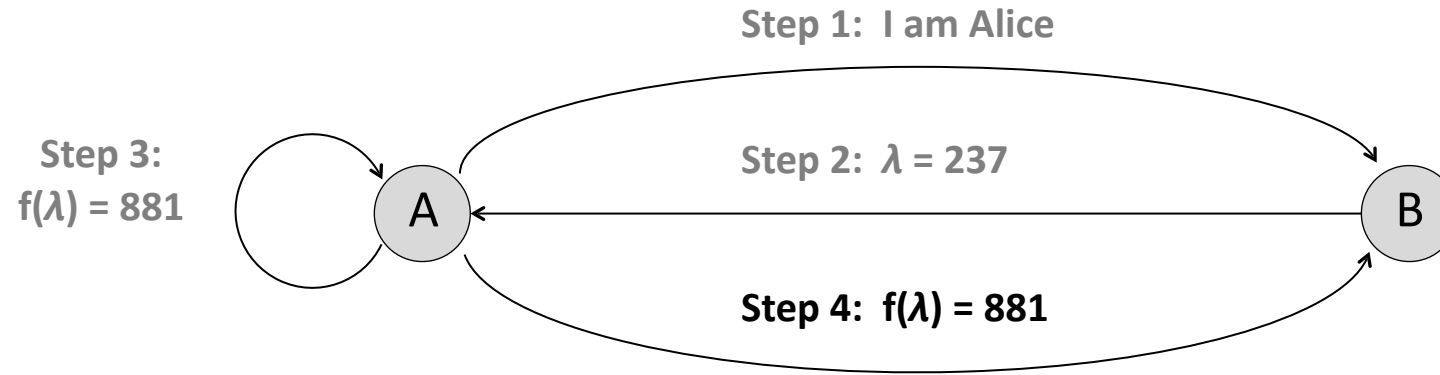
Handheld Authentication Device



Embedded Function
 $f: \text{integer} \rightarrow \text{integer}$

<i>User</i>	<i>Function</i>
A	f
C	f'
G	f''
...	...

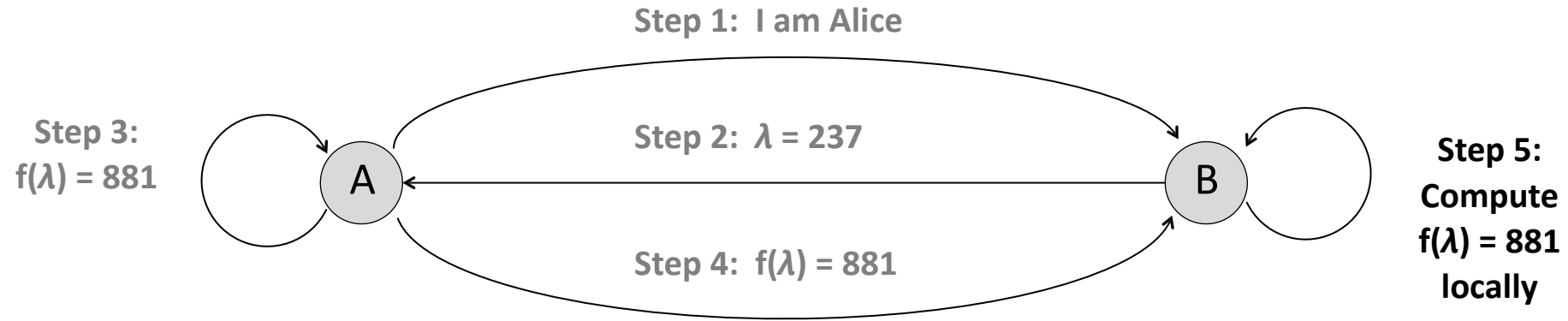
Handheld Authentication Device



Embedded Function
 $f: \text{integer} \rightarrow \text{integer}$

<i>User</i>	<i>Function</i>
A	f
C	f'
G	f''
...	...

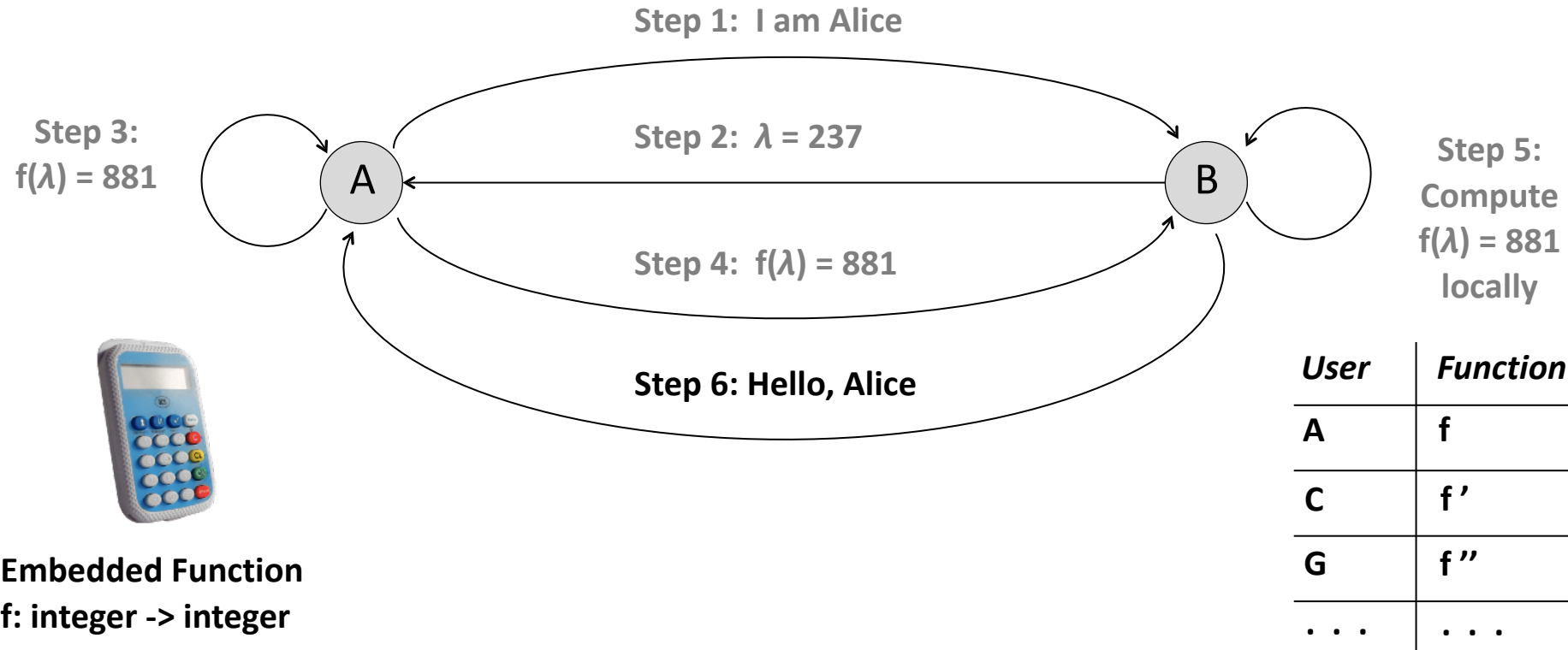
Handheld Authentication Device



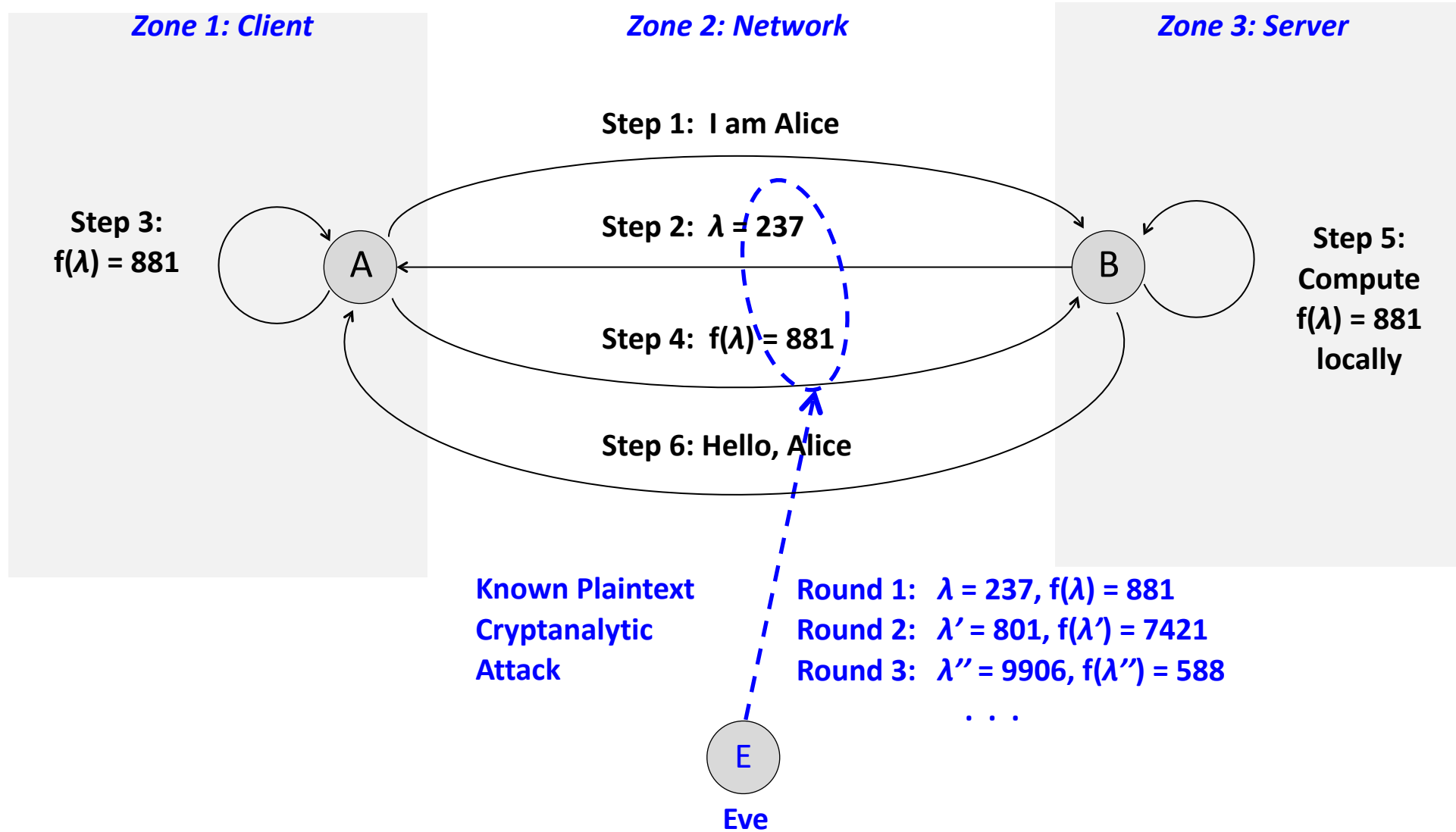
Embedded Function
 $f: \text{integer} \rightarrow \text{integer}$

<i>User</i>	<i>Function</i>
A	f
C	f'
G	f''
...	...

Handheld Authentication Device



Handheld Authentication Device Protocol



RSA SecurID One-Time Password (OTP) Algorithm



f : integer \rightarrow integer

λ : integer seed

t_0 : initial time

t_c : current time

Δt : time interval

$n = (t_c - t_0) / \Delta t$



RSA SecurID One-Time Password (OTP) Algorithm



f : integer \rightarrow 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

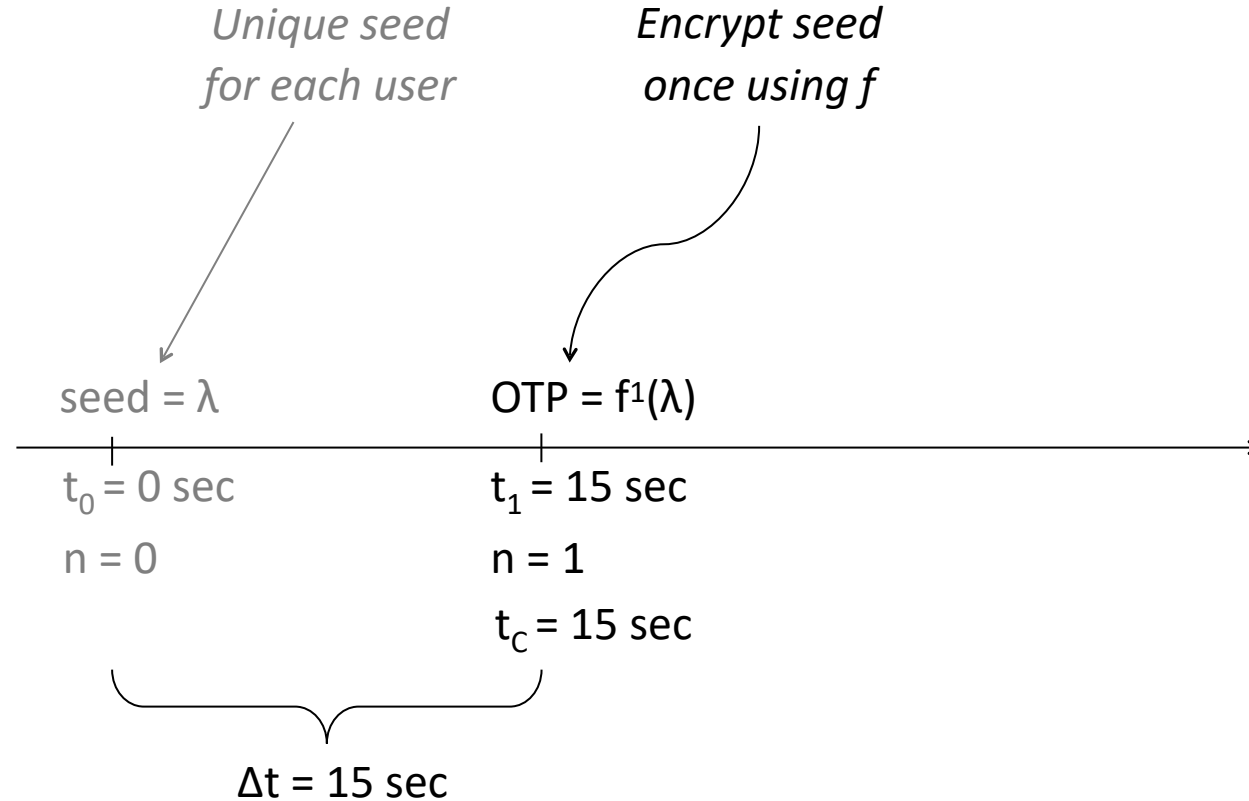
$n = 0$

$t_c = 0$ sec

RSA SecurID One-Time Password (OTP) Algorithm



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



RSA SecurID One-Time Password (OTP) Algorithm



f : integer \rightarrow integer

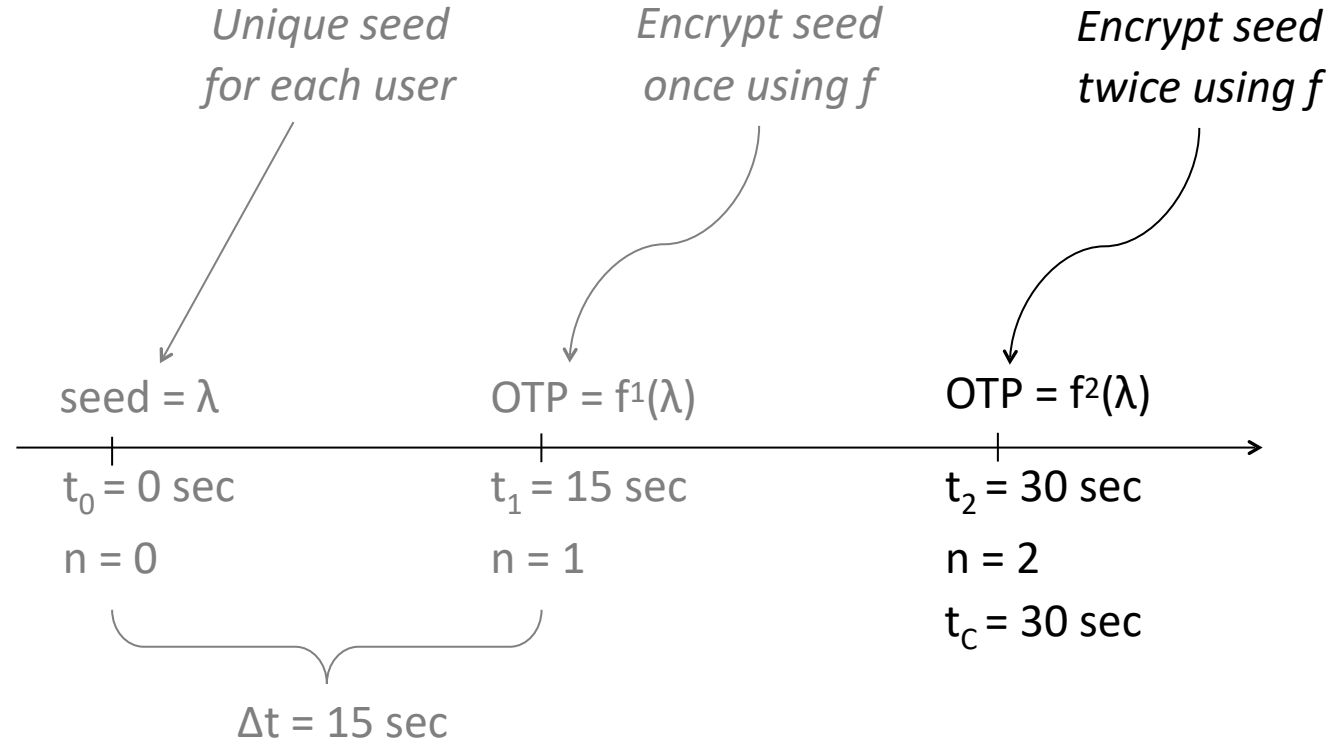
λ : integer seed

t_0 : initial time

t_c : current time

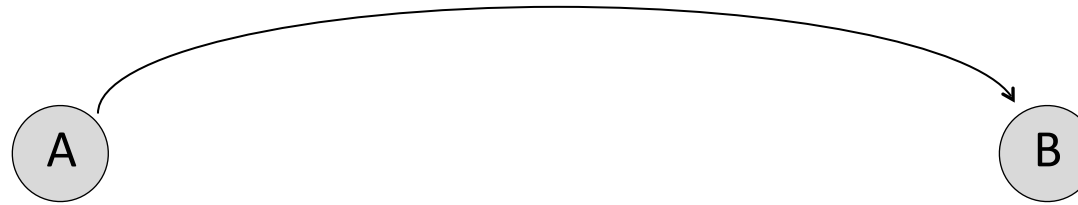
Δt : time interval

$n = (t_c - t_0) / \Delta t$



RSA SecurID Protocol

Step 1: I am Alice



f : integer \rightarrow integer

λ : integer seed

t_0 : initial time

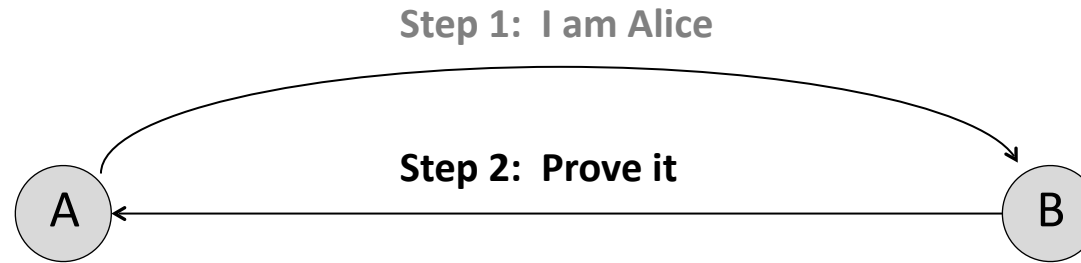
t_c : current time

Δt : time interval

$n = (t_c - t_0) / \Delta t$

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

RSA SecurID Protocol



f : integer \rightarrow integer

λ : integer seed

t_0 : initial time

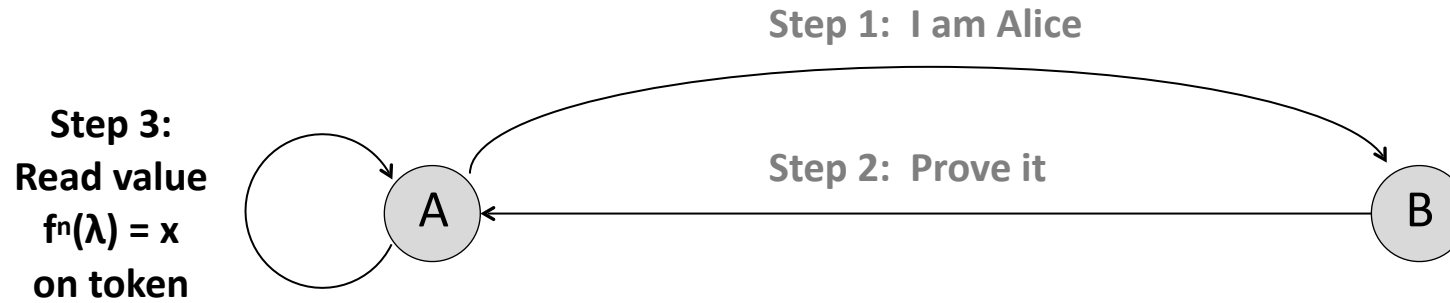
t_c : current time

Δt : time interval

$n = (t_c - t_0) / \Delta t$

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

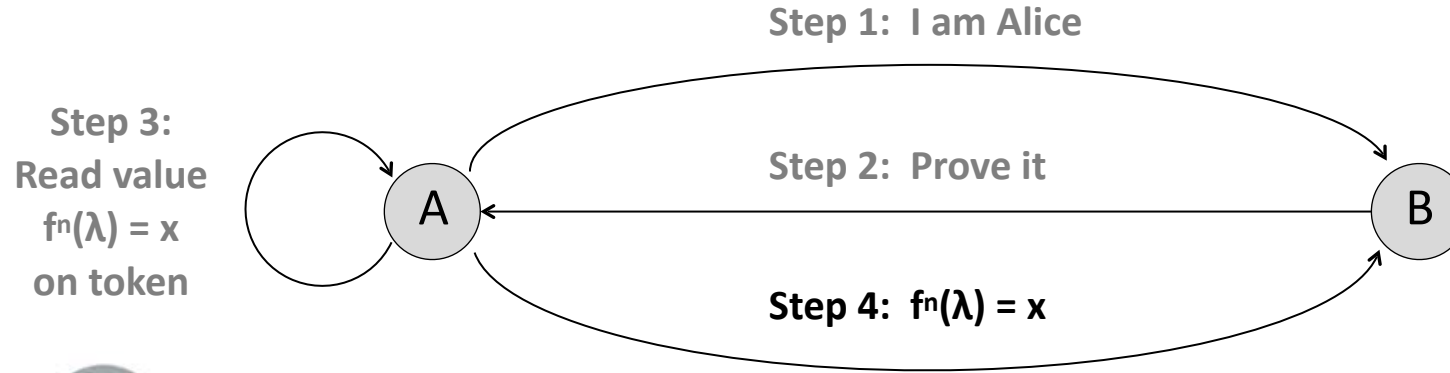
RSA SecurID Protocol



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

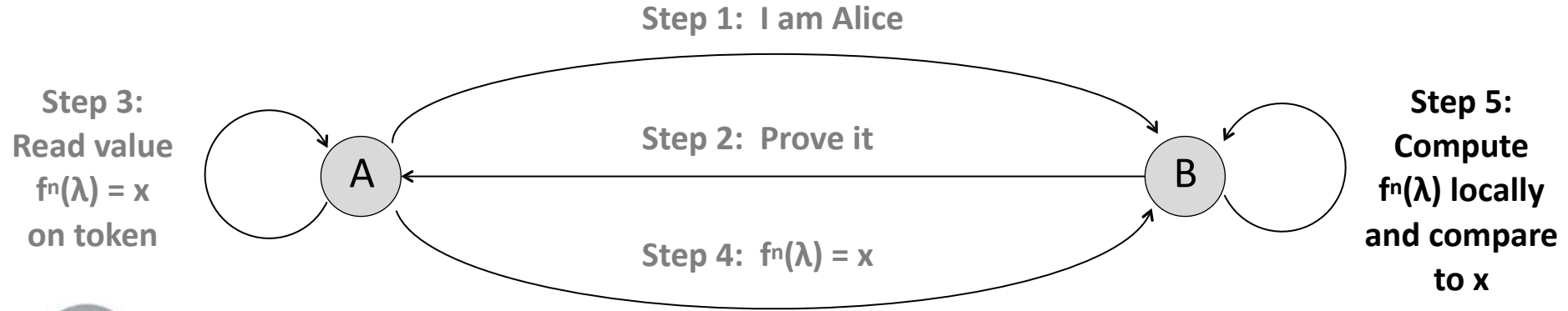
RSA SecurID Protocol



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

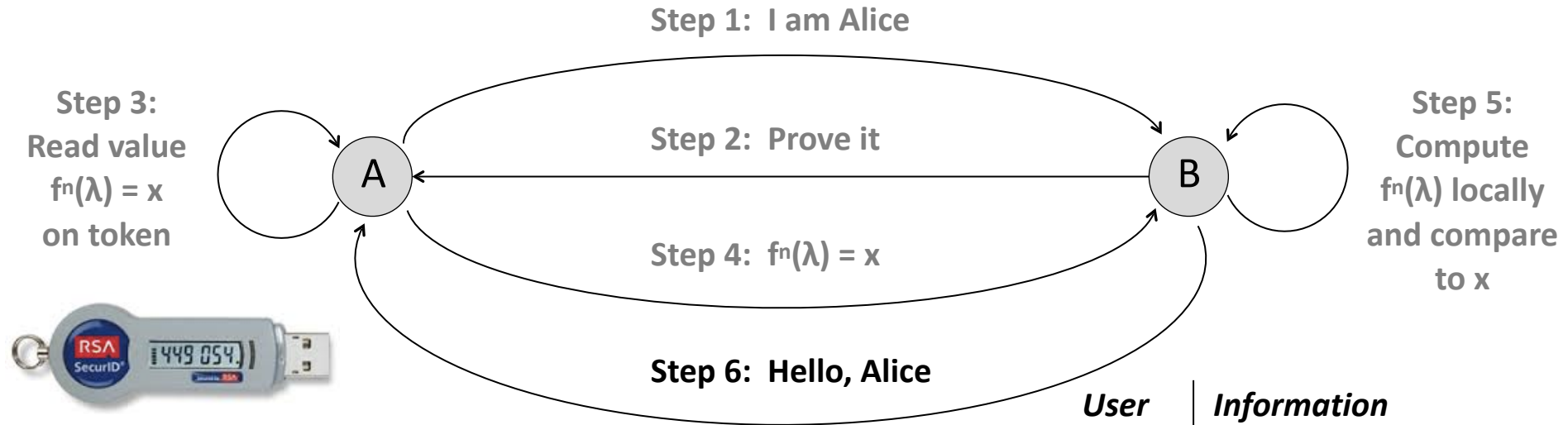
RSA SecurID Protocol



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

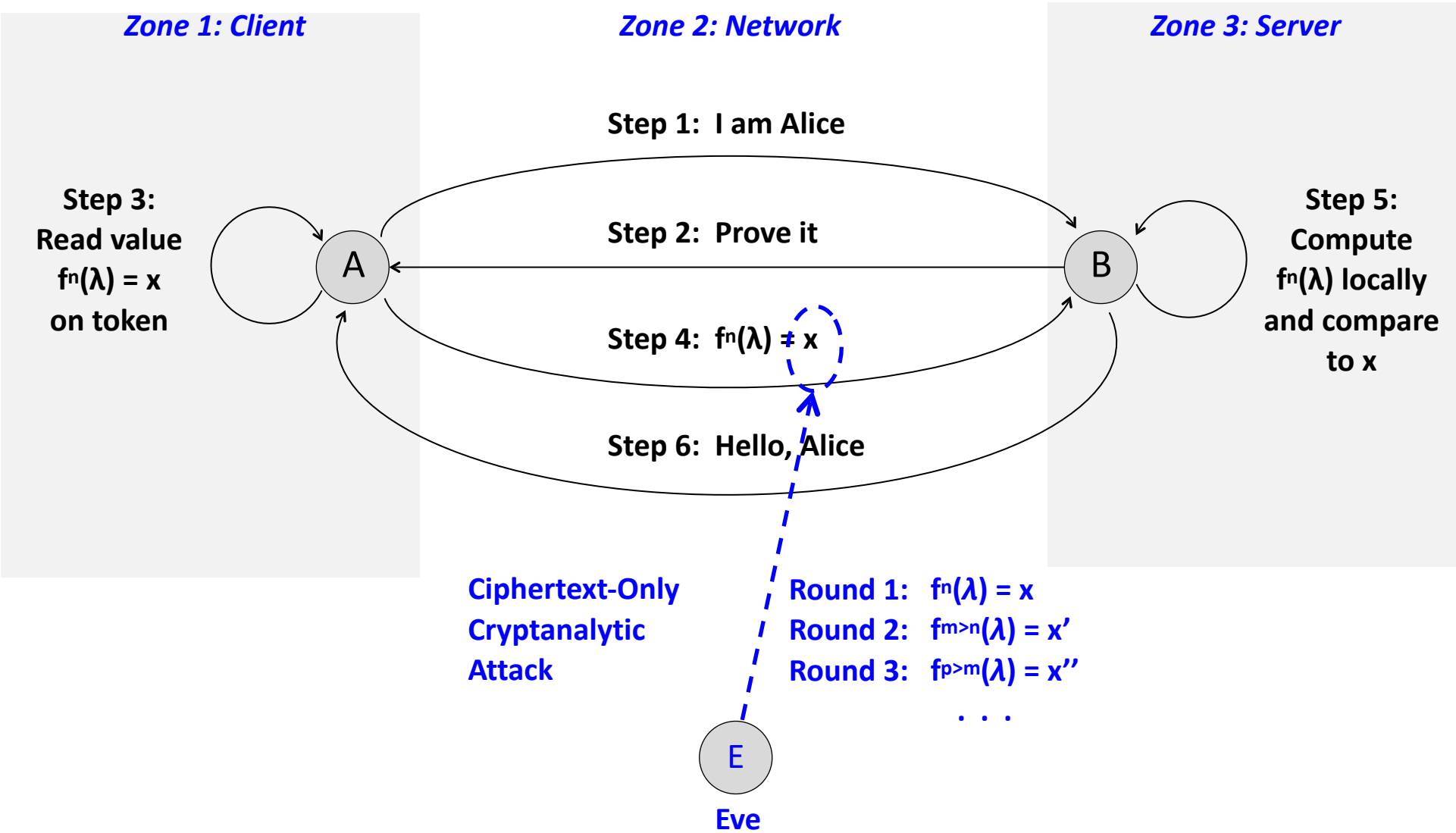
RSA SecurID Protocol



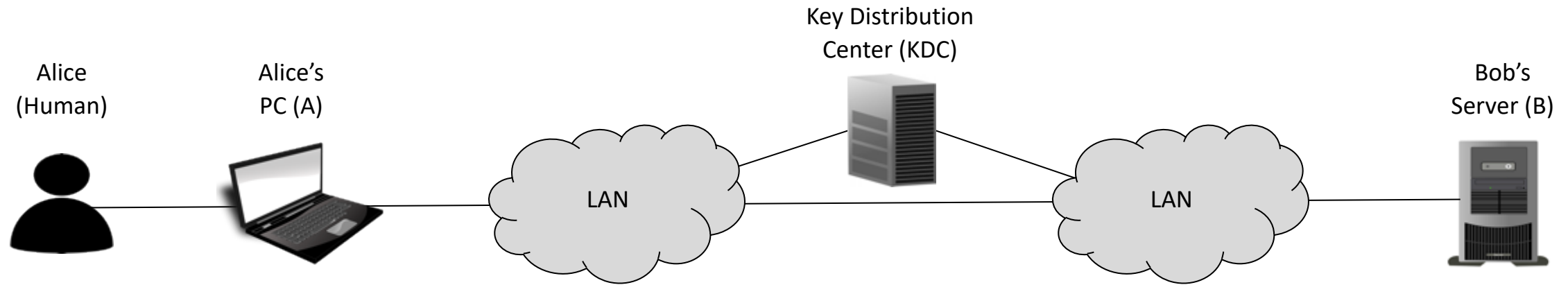
f : integer \rightarrow integer
 λ : integer seed
 t_0 : initial time
 t_c : current time
 Δt : time interval
 $n = (t_c - t_0) / \Delta t$

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

RSA SecurID Protocol



Kerberos: A Complex Solution to a Simple Password Problem



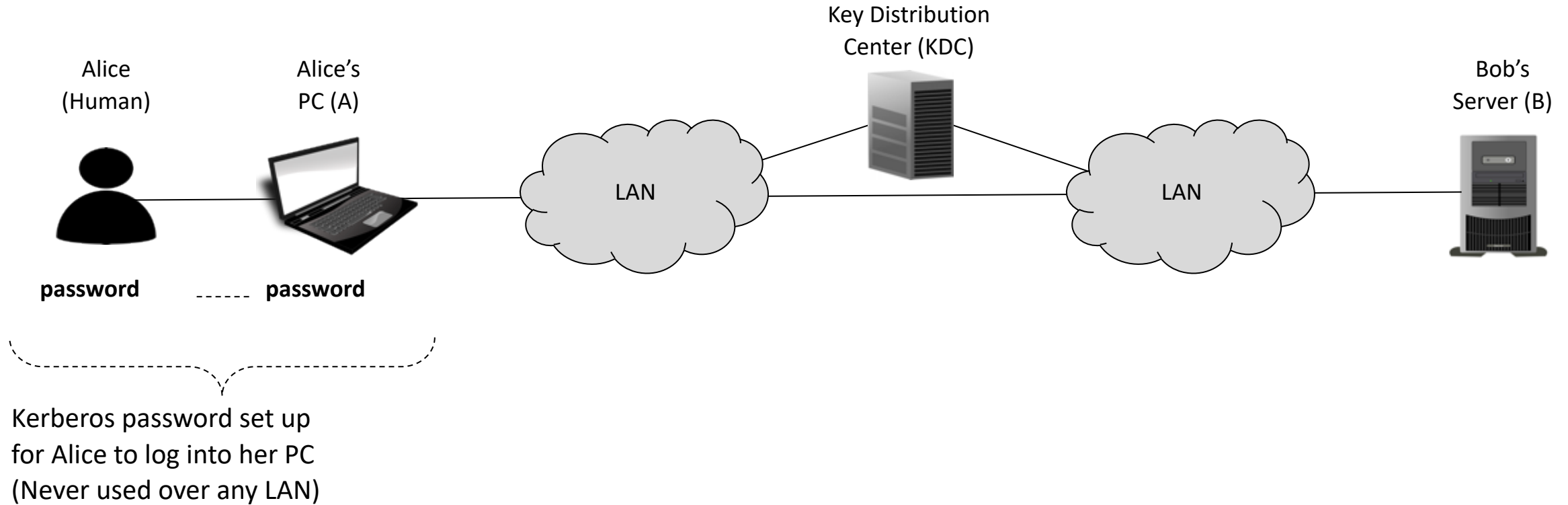
Basic Kerberos Concept:

- 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)



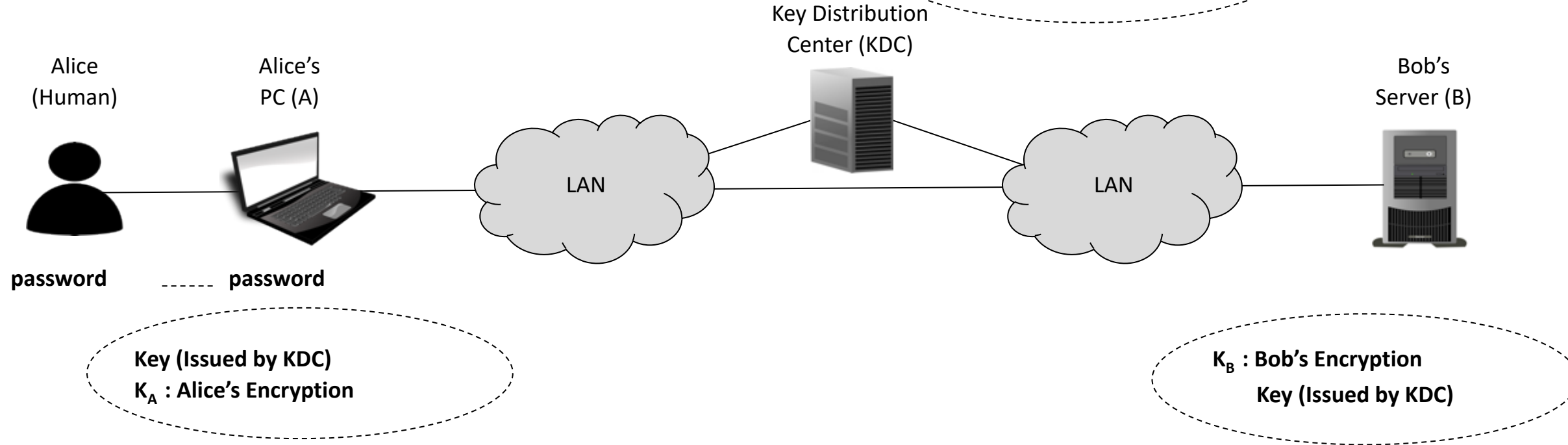
Kerberos – Preconditions

Infrastructure
Preconditions:



Kerberos – Keys

Infrastructure
Preconditions:



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

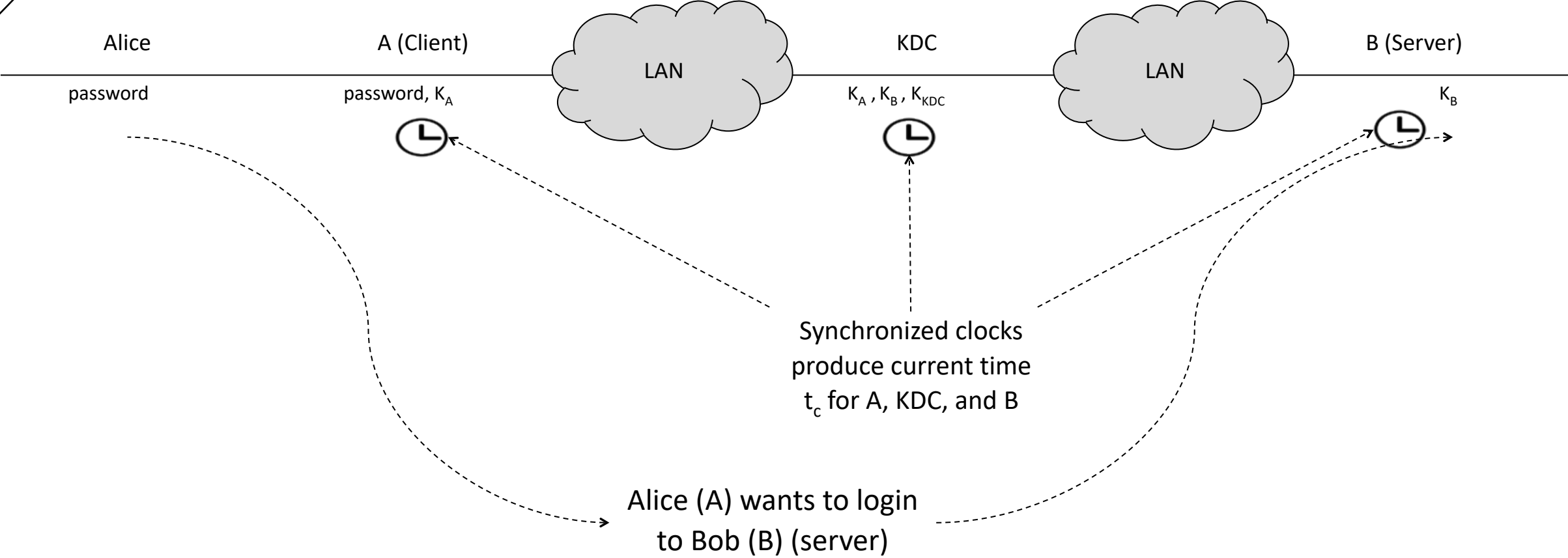
$$\{\{m\}_{K_A}\}_{K_A} = m$$

↑ Encrypt
 ↑ Decrypt

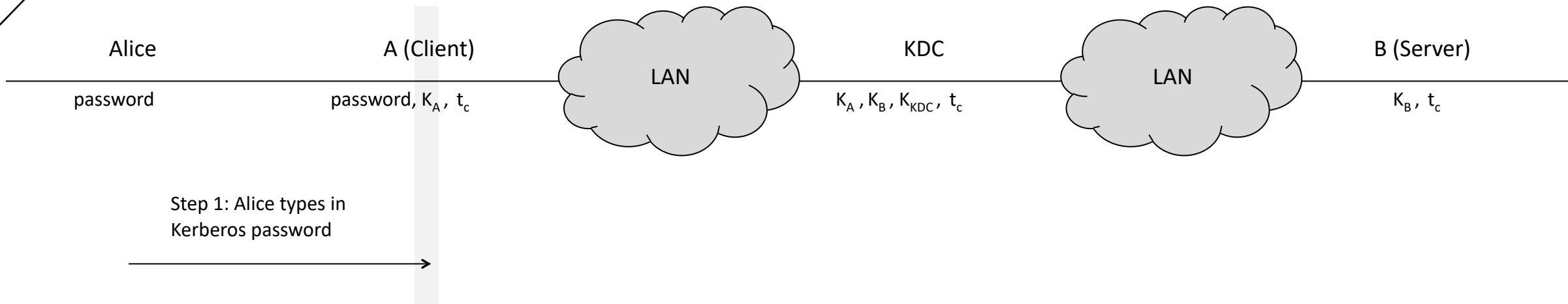
$$\{\{m\}_{K_B}\}_{K_B} = m$$

$$\{\{m\}_{K_{KDC}}\}_{K_{KDC}} = m$$

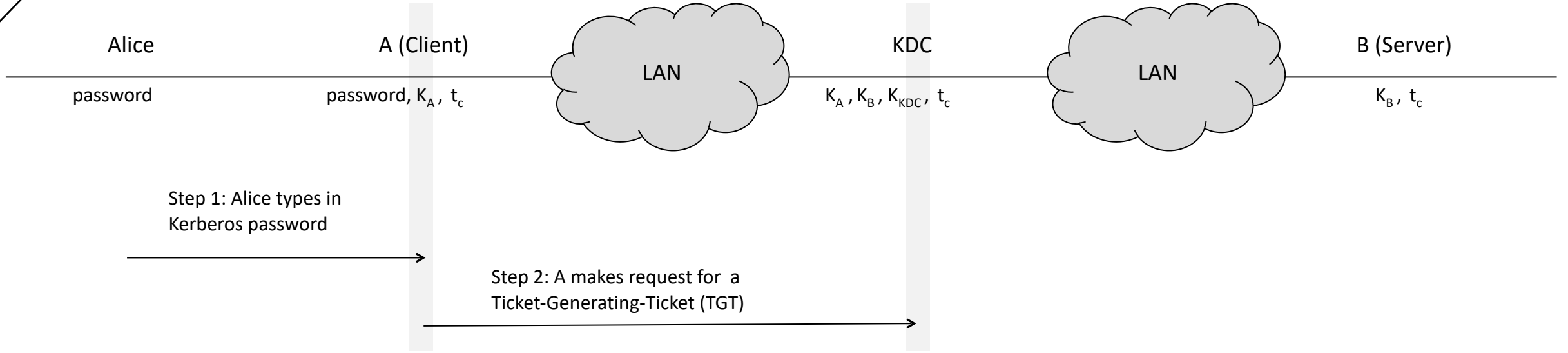
Kerberos – Clocks



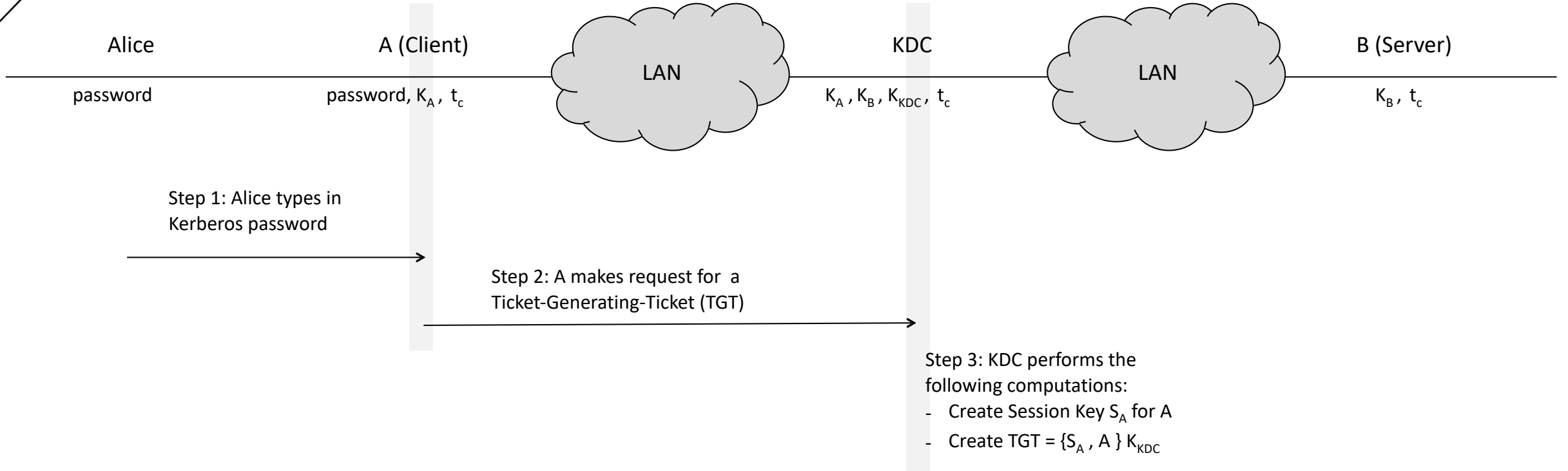
Kerberos Step 1: Type in Local Password



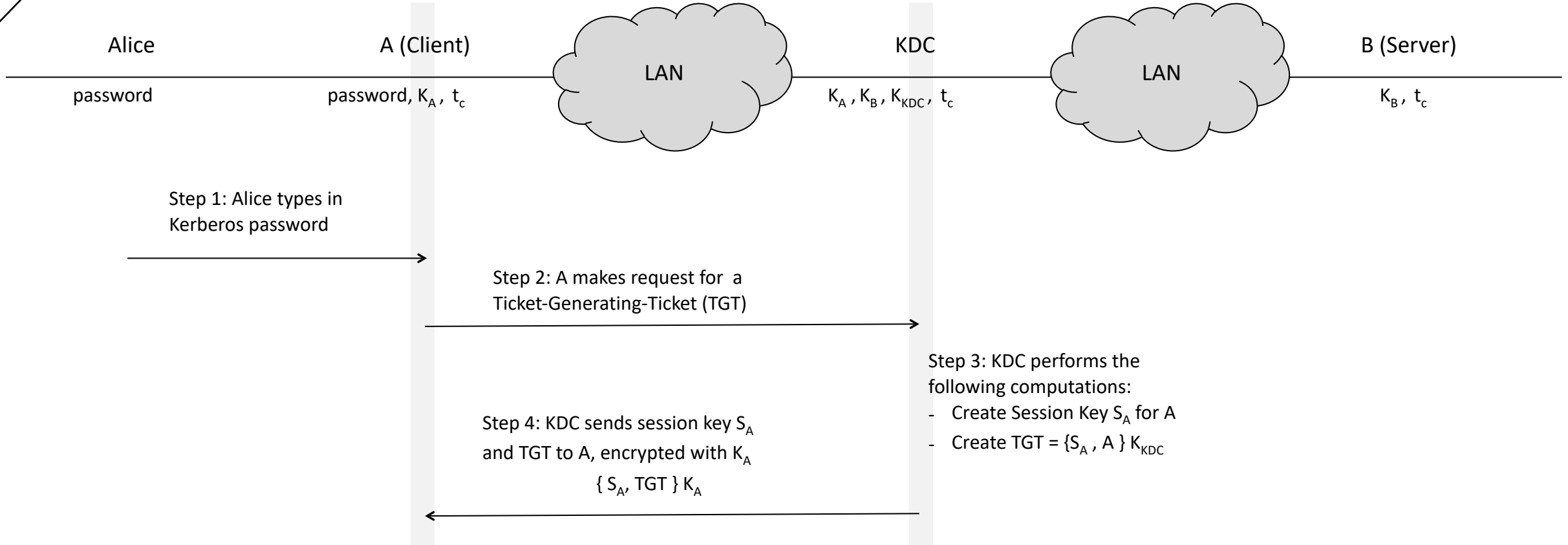
Kerberos Step 2: Request TGT and Session Key



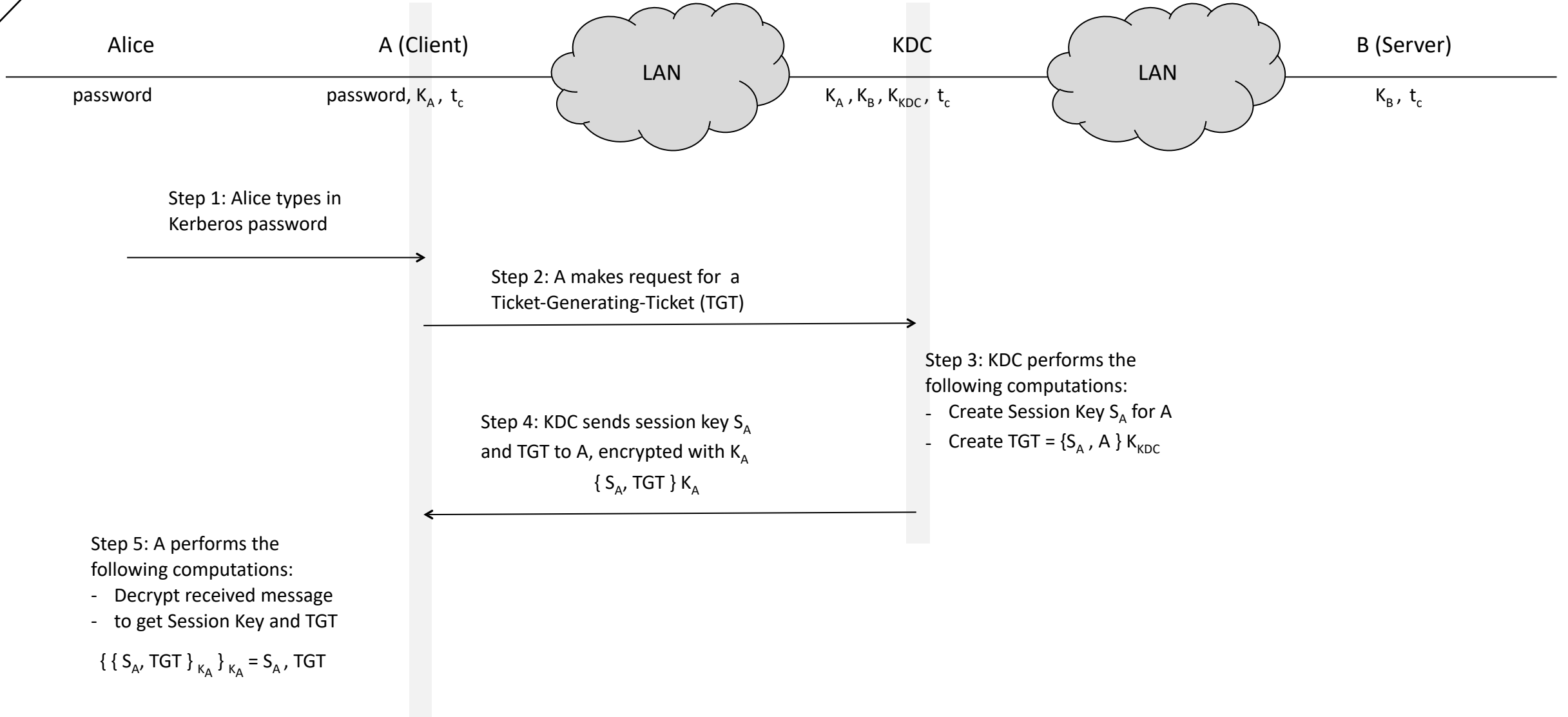
Kerberos Step 3: Create Session Key and TGT



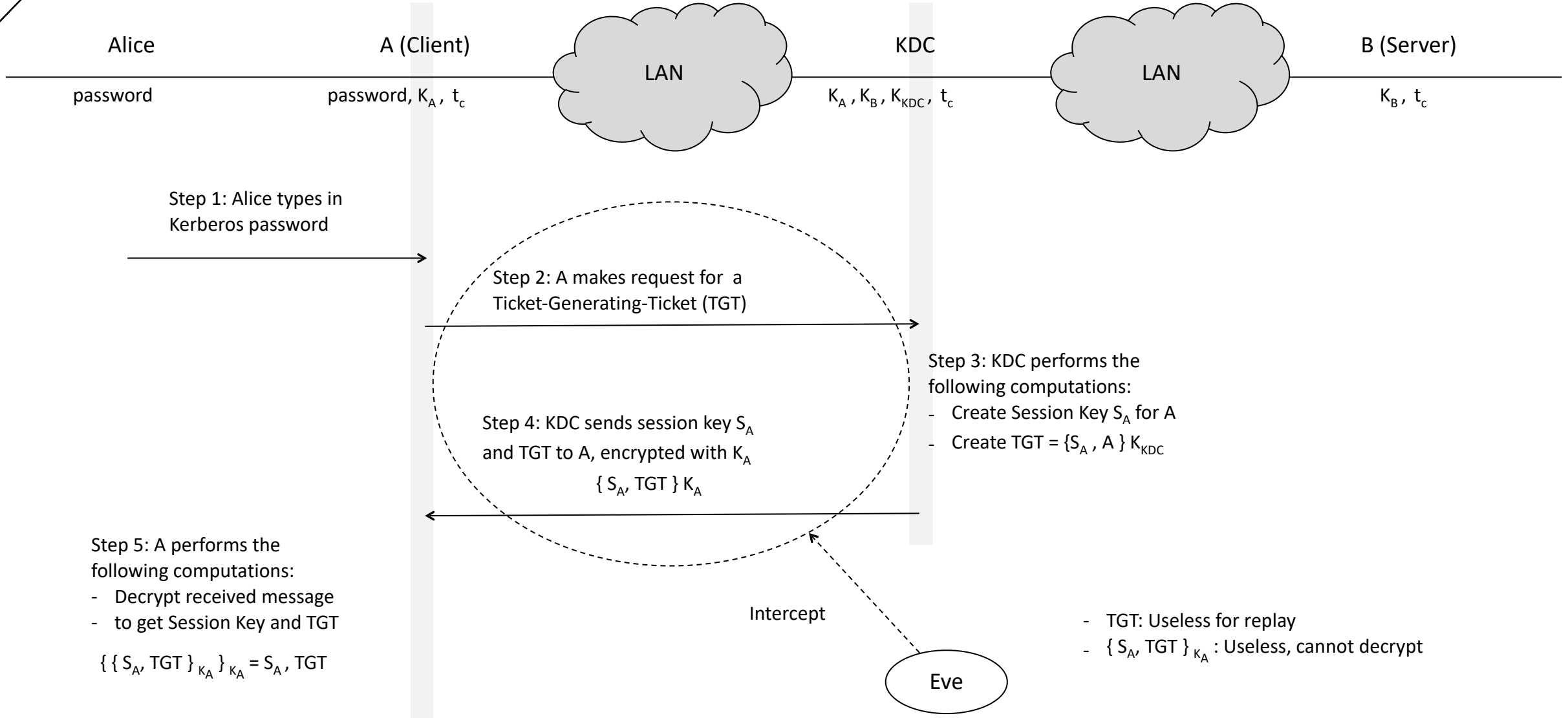
Kerberos Step 4: Provide Session Key and TGT



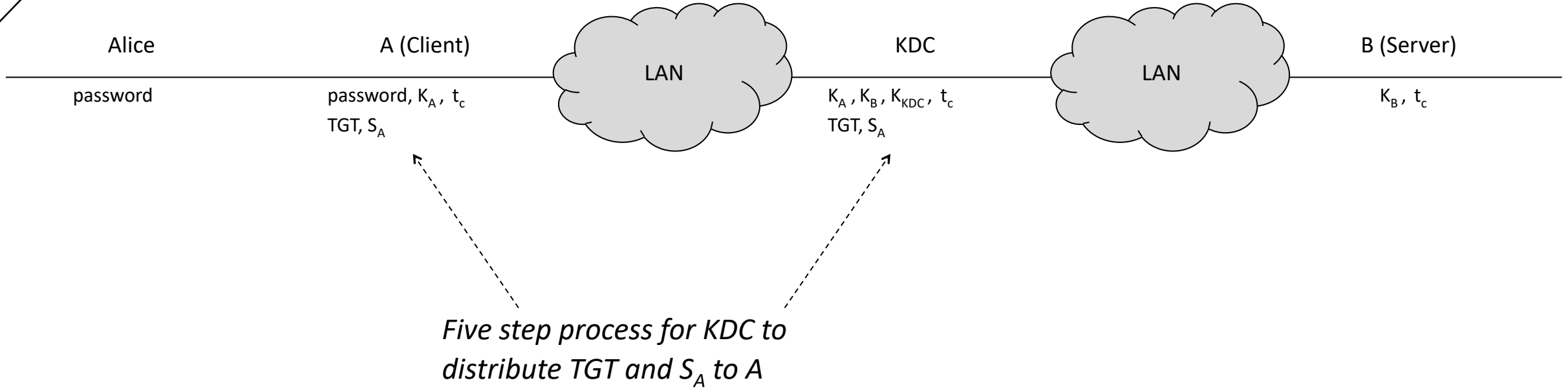
Kerberos Step 5: Decrypt Session Key and Store TGT



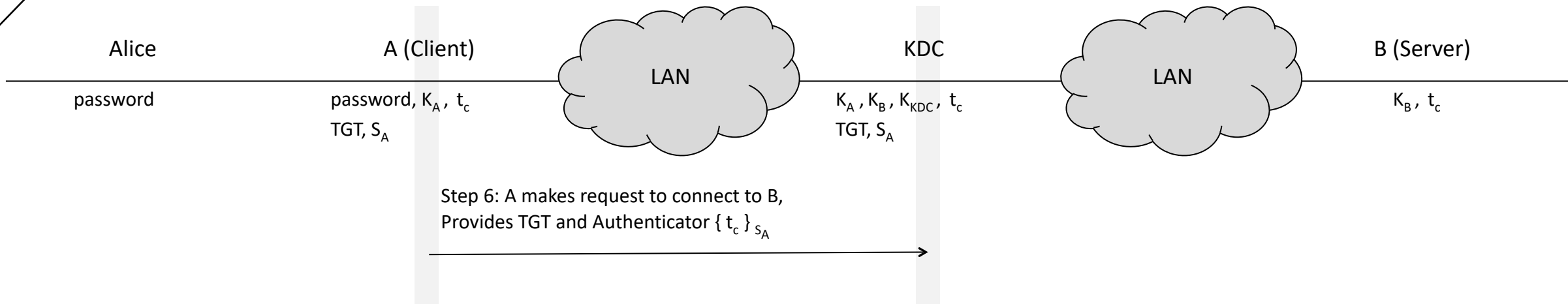
Kerberos – Through Five Steps: Eve Cannot Hack



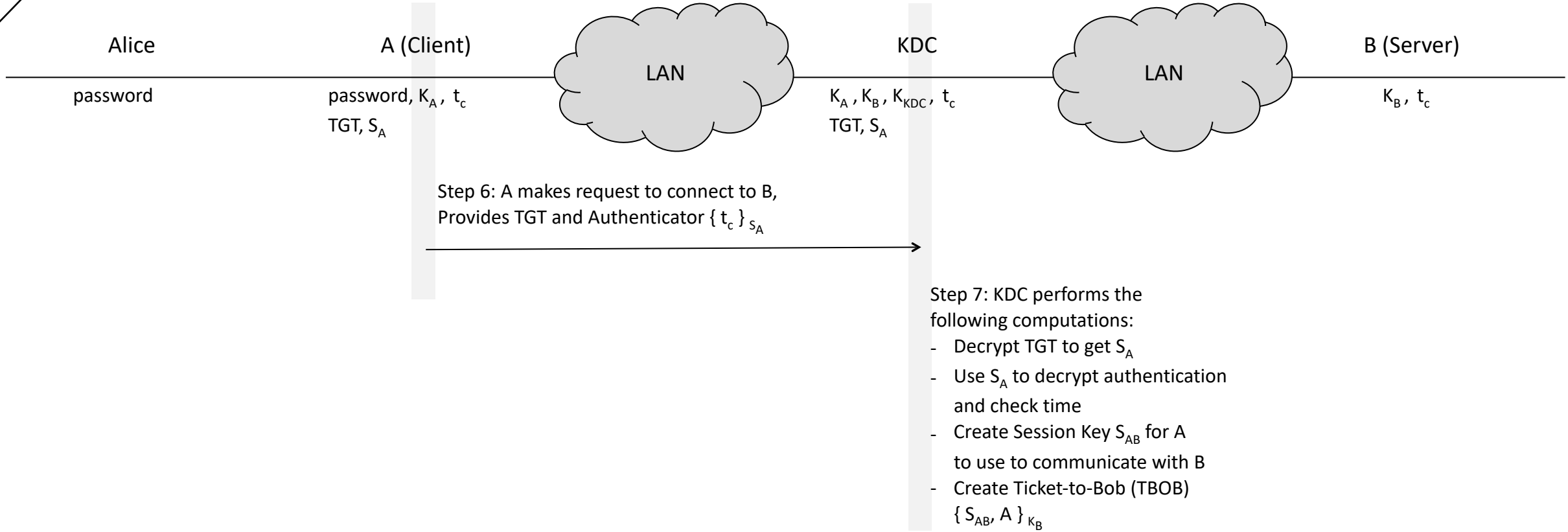
Kerberos – Result of Five Steps



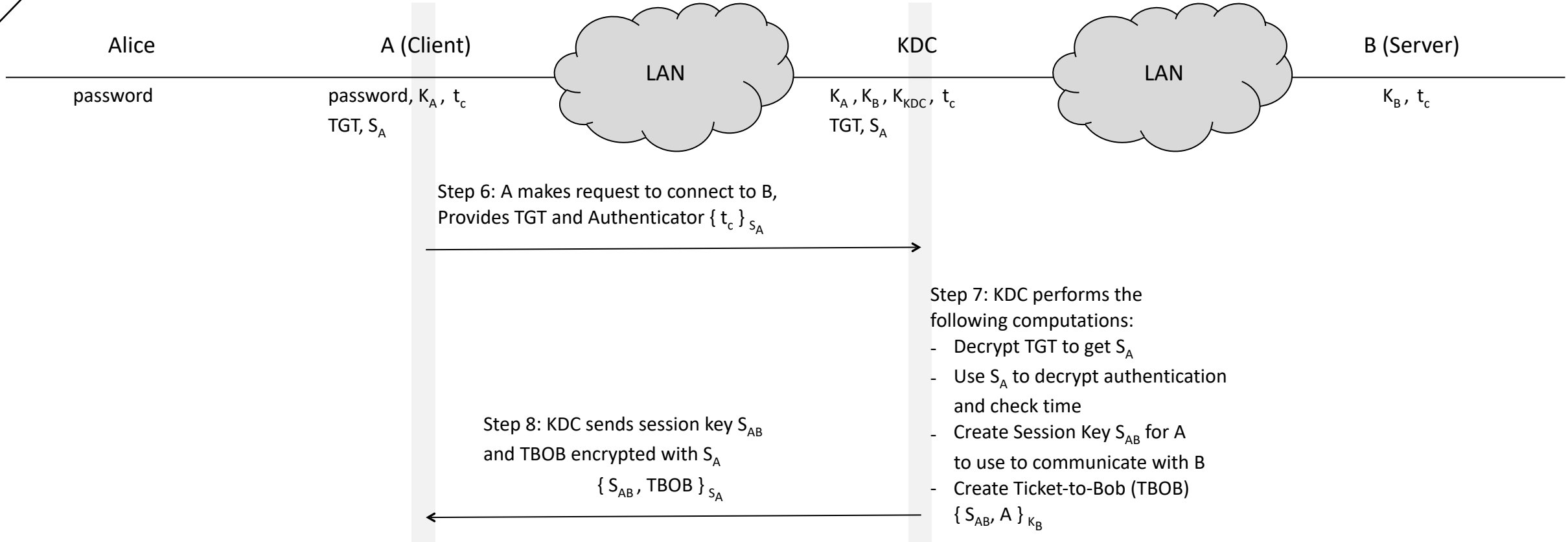
Kerberos Step 6: Request Login to B



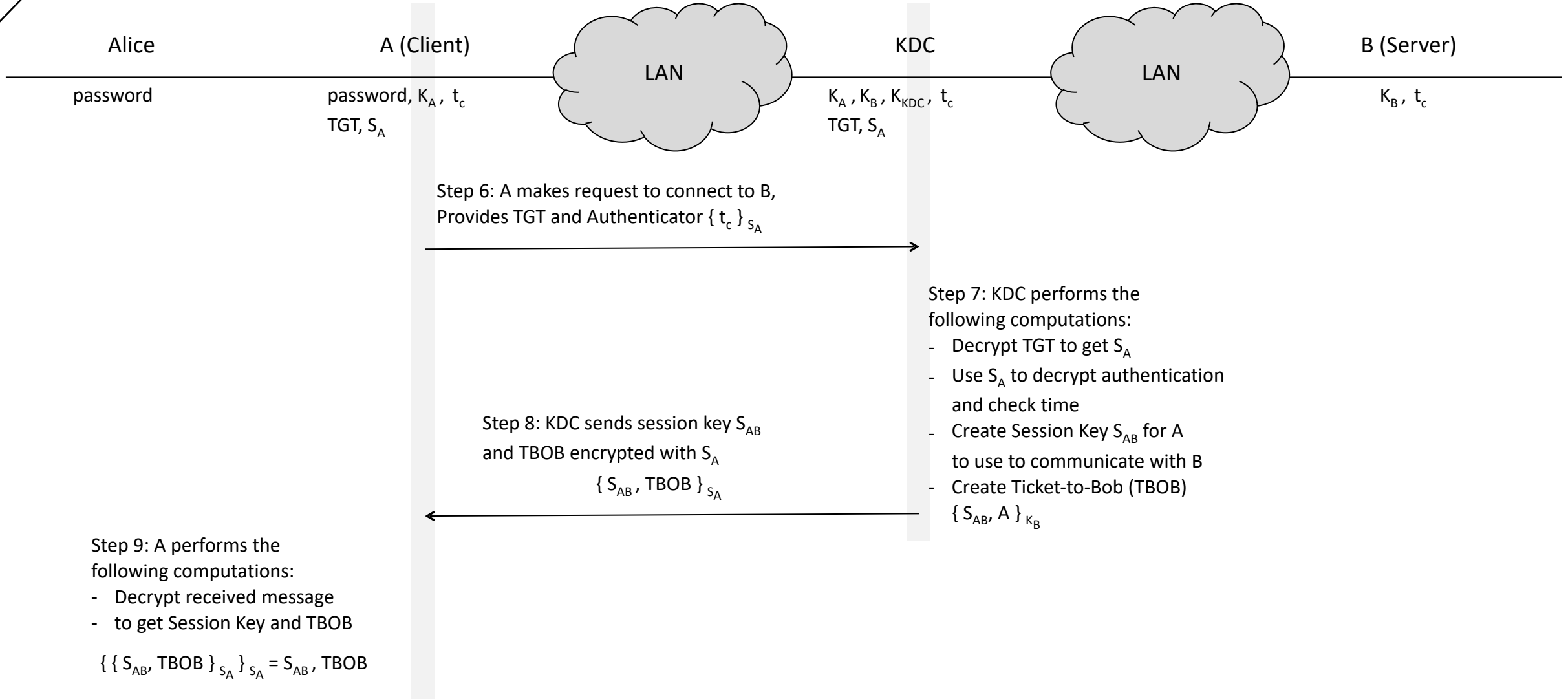
Kerberos Step 7: Create Session Key and Ticket to Bob



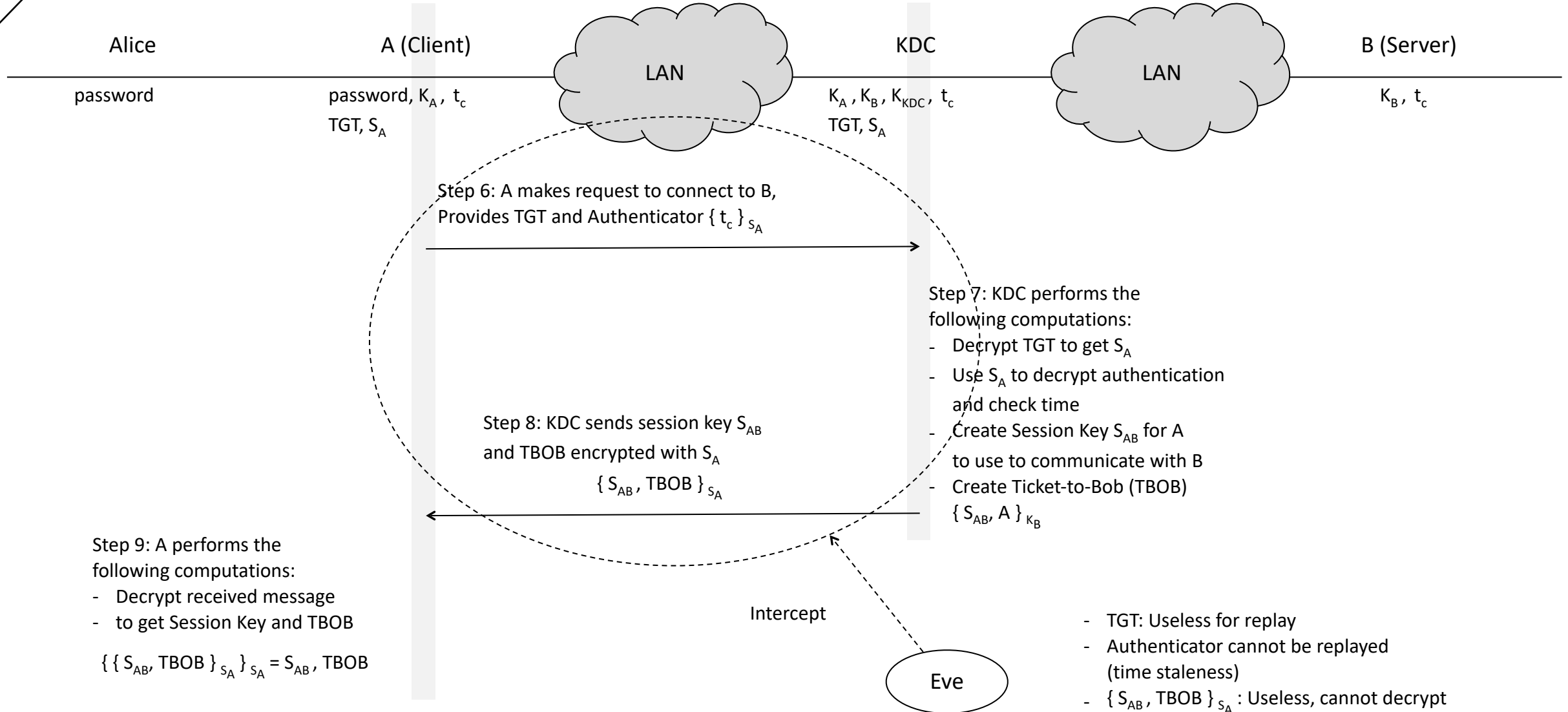
Kerberos Step 8: Send Session Key and Ticket to Bob



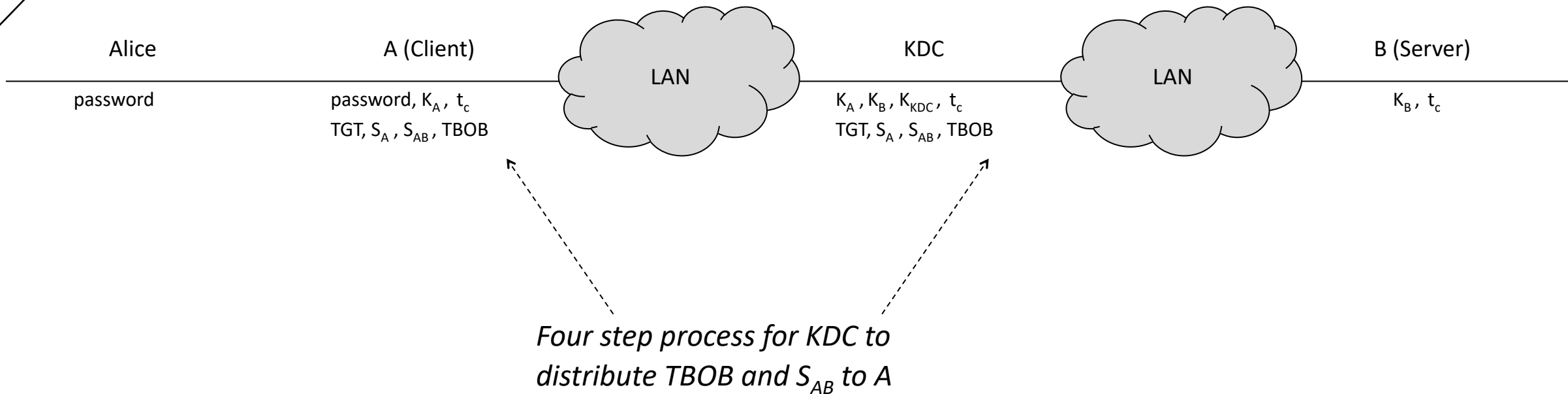
Kerberos Step 9: Decrypt Session Key and Store Ticket to Bob



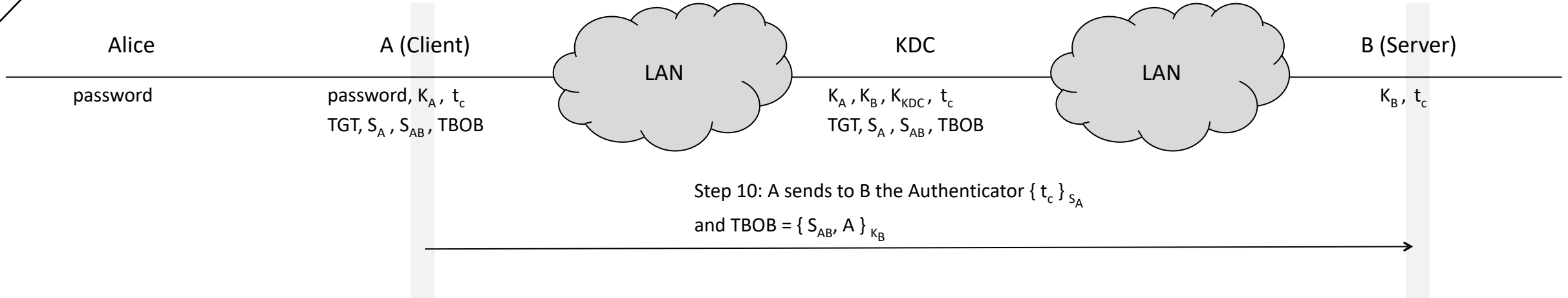
Kerberos – Through Nine Steps: Eve Cannot Hack



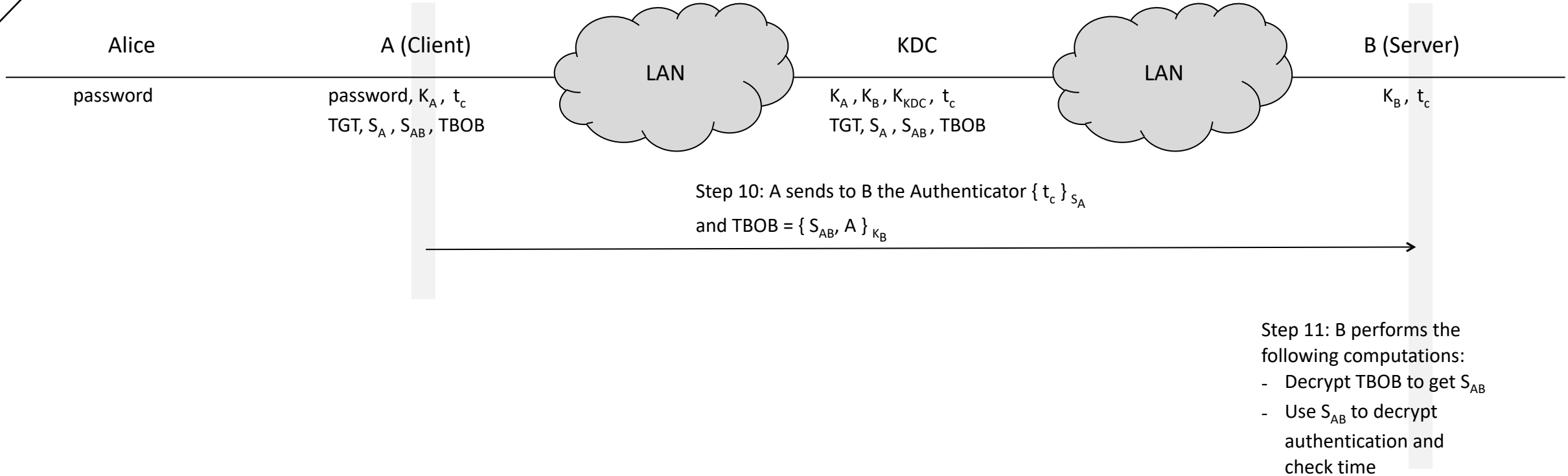
Kerberos – Result of Nine Steps



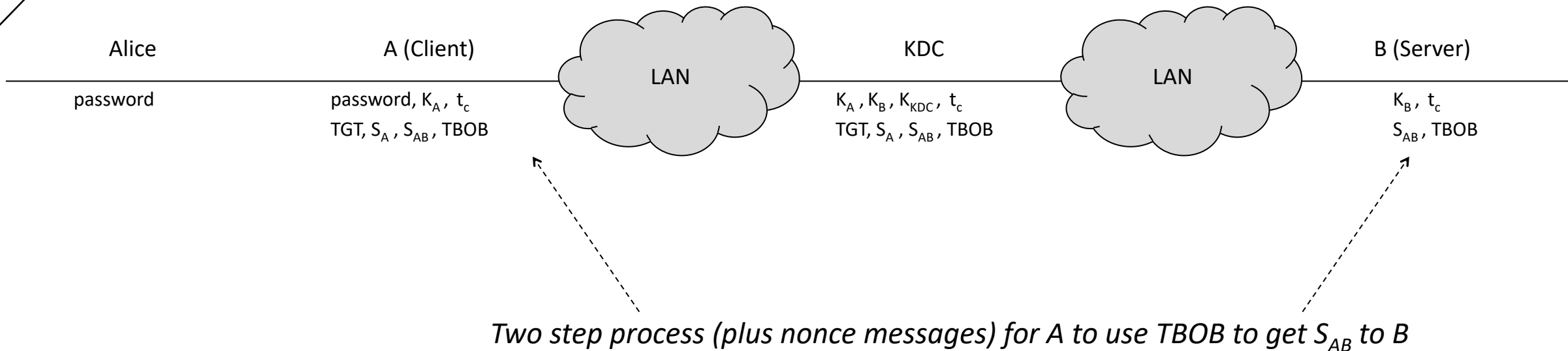
Kerberos Step 10: Send Bob the Ticket to Bob



Kerberos Step 11: Decrypt Ticket to Bob and Check Time



Kerberos – Result of Eleven Steps



Kerberos – Realms

