Blind Certificate Authorities

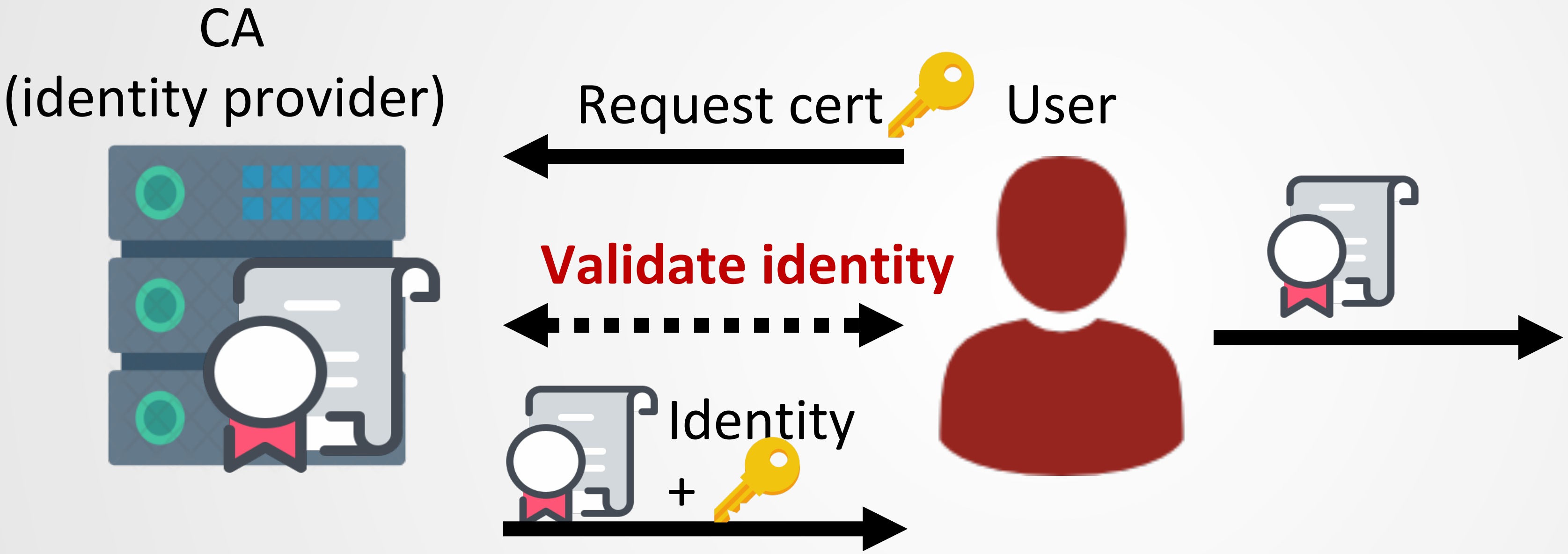
Liang Wang[1], Gilad Asharov[2], Rafael Pass[2], Thomas Ristenpart[2], abhi shelat[3]

[1] Princeton University [2] Cornell Tech [3] Northeastern University

Motivation

Certificate Authorities (CA) issue certificates

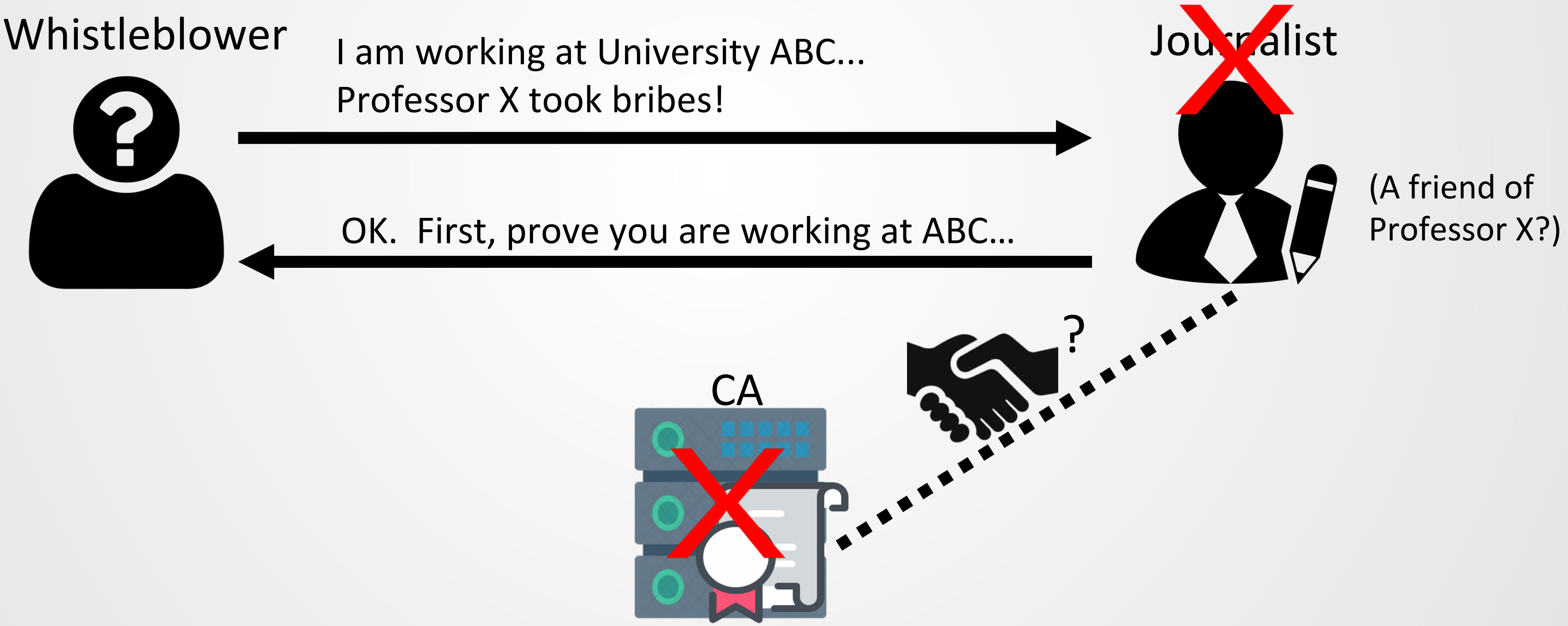
Certificates bind public keys to identities



- Email
- Website login
- **Anonymous credential systems**
- ….

The user must reveal true identity to the CA during identity validation

Identity is sensitive



Whistleblower

I am working at University ABC…
Professor X took bribes!

OK. First, prove you are working at ABC…

Journalist

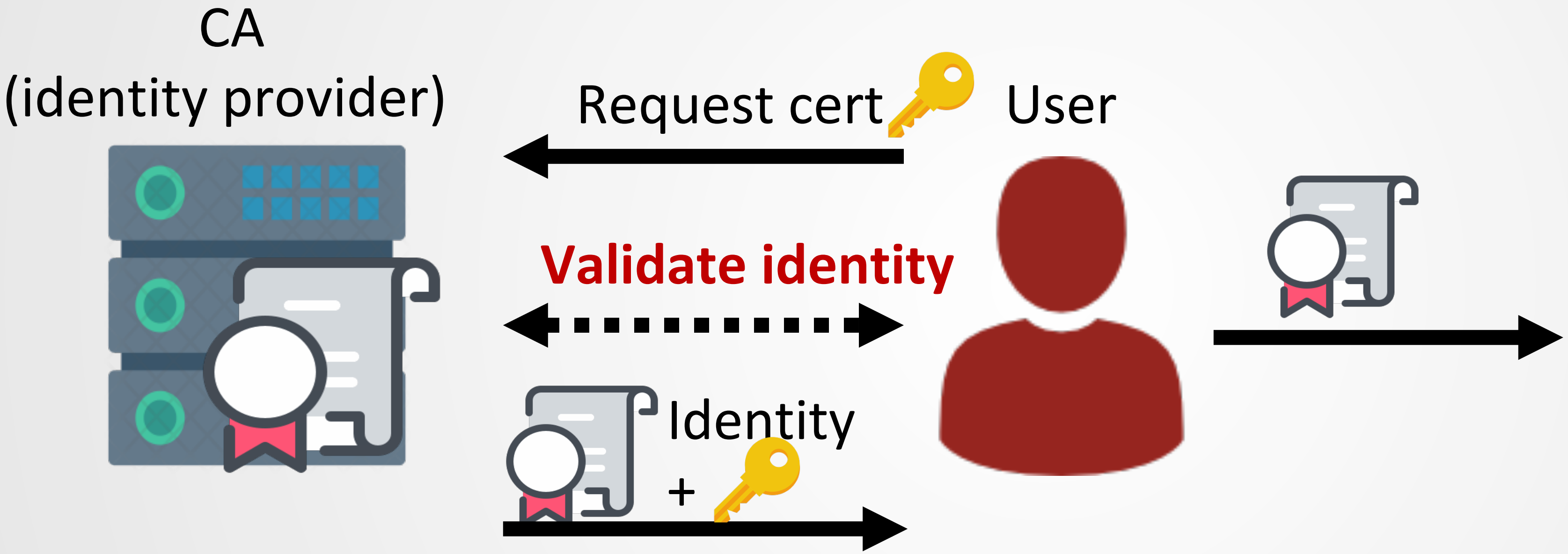
(A friend of
Professor X?)

?

CA

Third-party or from University ABC

CA: single point of privacy failure



- PGP
- Website login
- **Anonymous credential systems**
- ….

alice@domain.com: cert1
bob@gmail.com: cert2
…..

Can we make CA “blind”?

Main challenge:

Validate an identity while not learning it

YES!!!

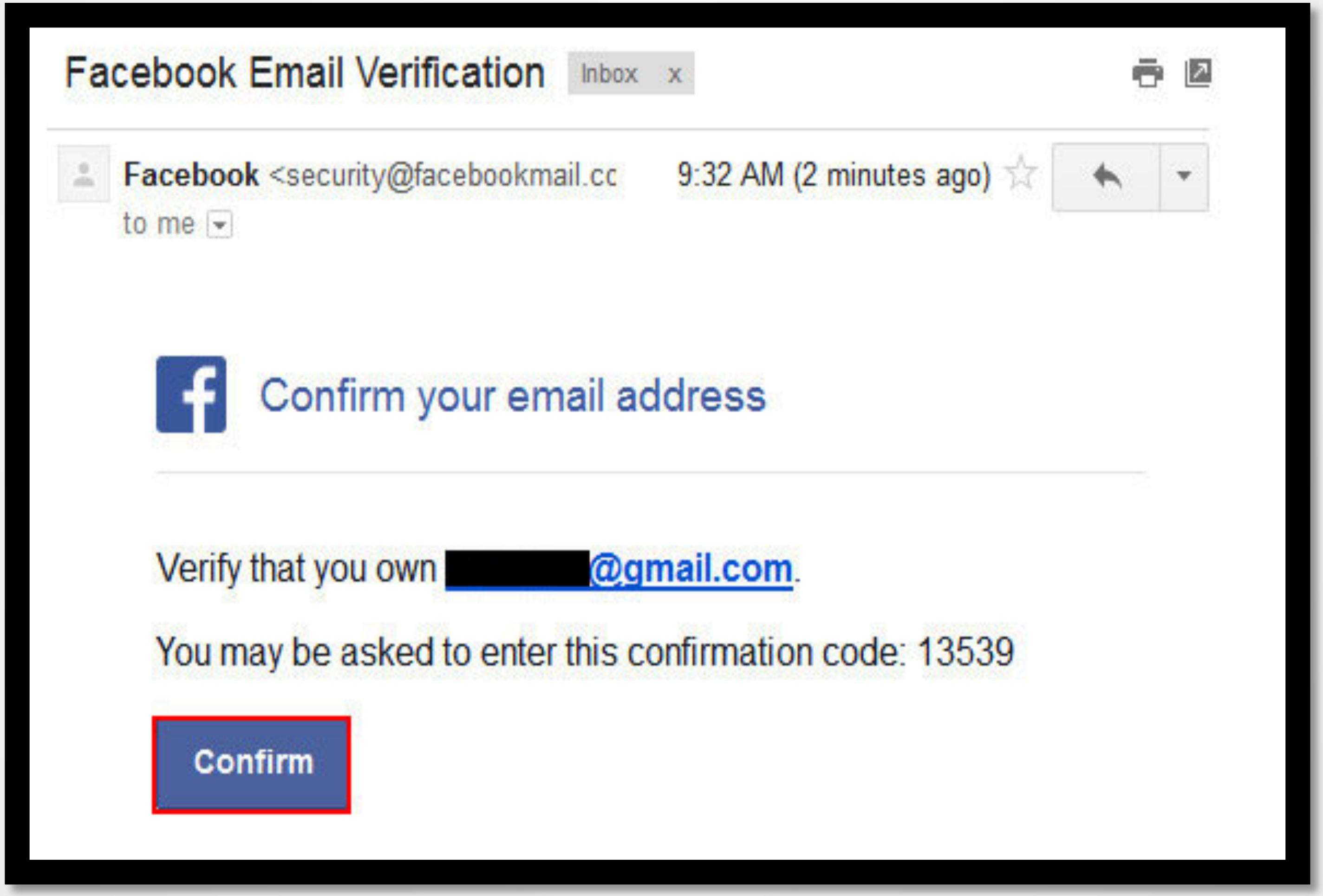
Contributions

- **Secure Channel Injection (SCI):**
  - A primitive allows a party to inject a small amount of information into a secure connection between two parties
  - (SCI-TLS) An efficient, special-purpose MPC protocol for two parties to compute a TLS record

- **Anonymous Proof of Account Ownership (PAO):**
  - Validate one owns some email accounts from a given organization without knowing which account

- **BlindCA:**
  - Validate ownership of an account alice@domain.com and issue a X.509 certificate binding “alice” to a public key, without learning the account and the key

Email is the most common identity



Facebook Email Verification

Facebook <security@facebookmail.cc

9:32 AM (2 minutes ago)

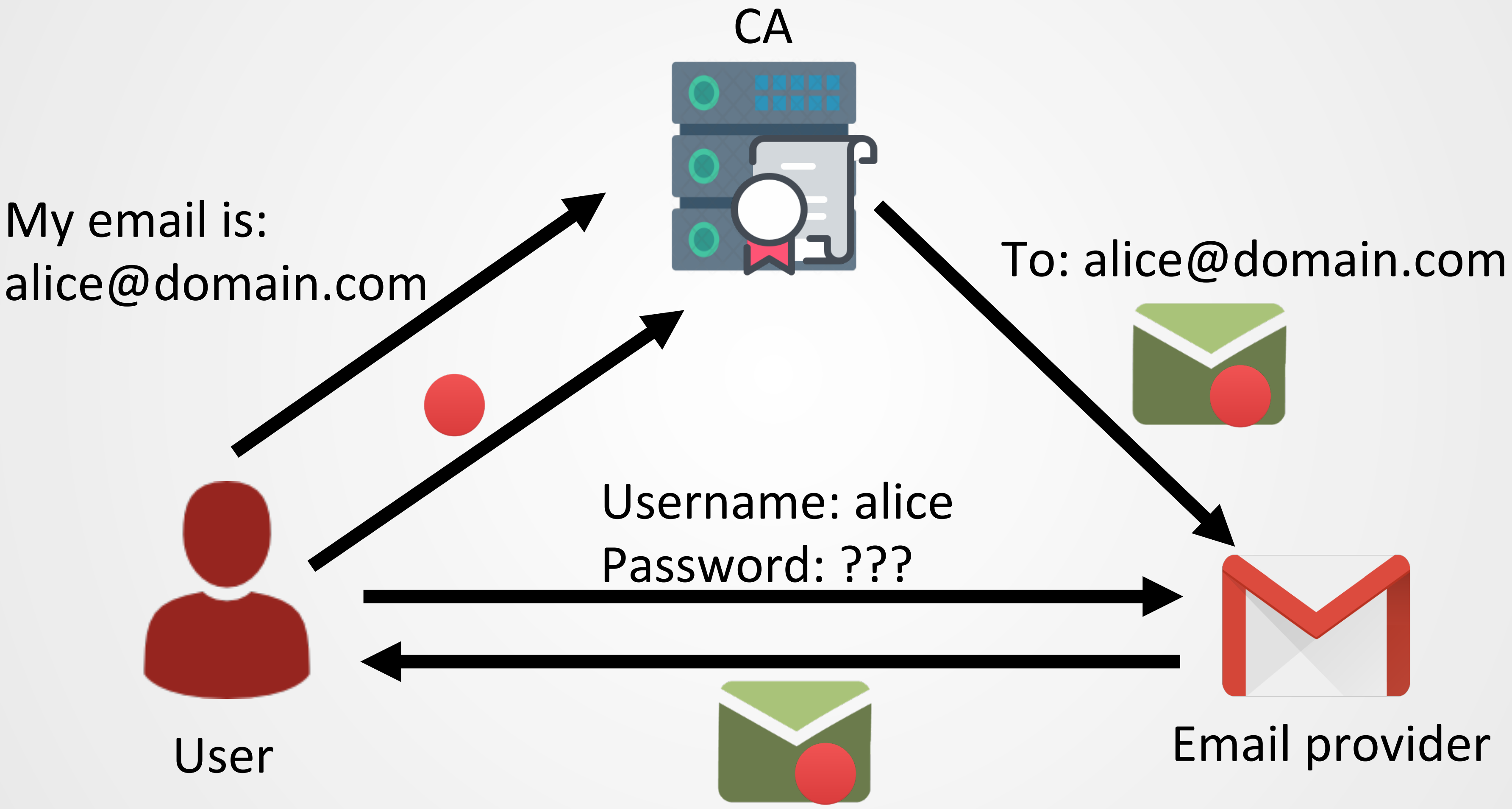
to me

Confirm your email address

Verify that you own [redacted]@gmail.com.

You may be asked to enter this confirmation code: 13539

Conventional email verification



CA

My email is:
alice@domain.com

To: alice@domain.com

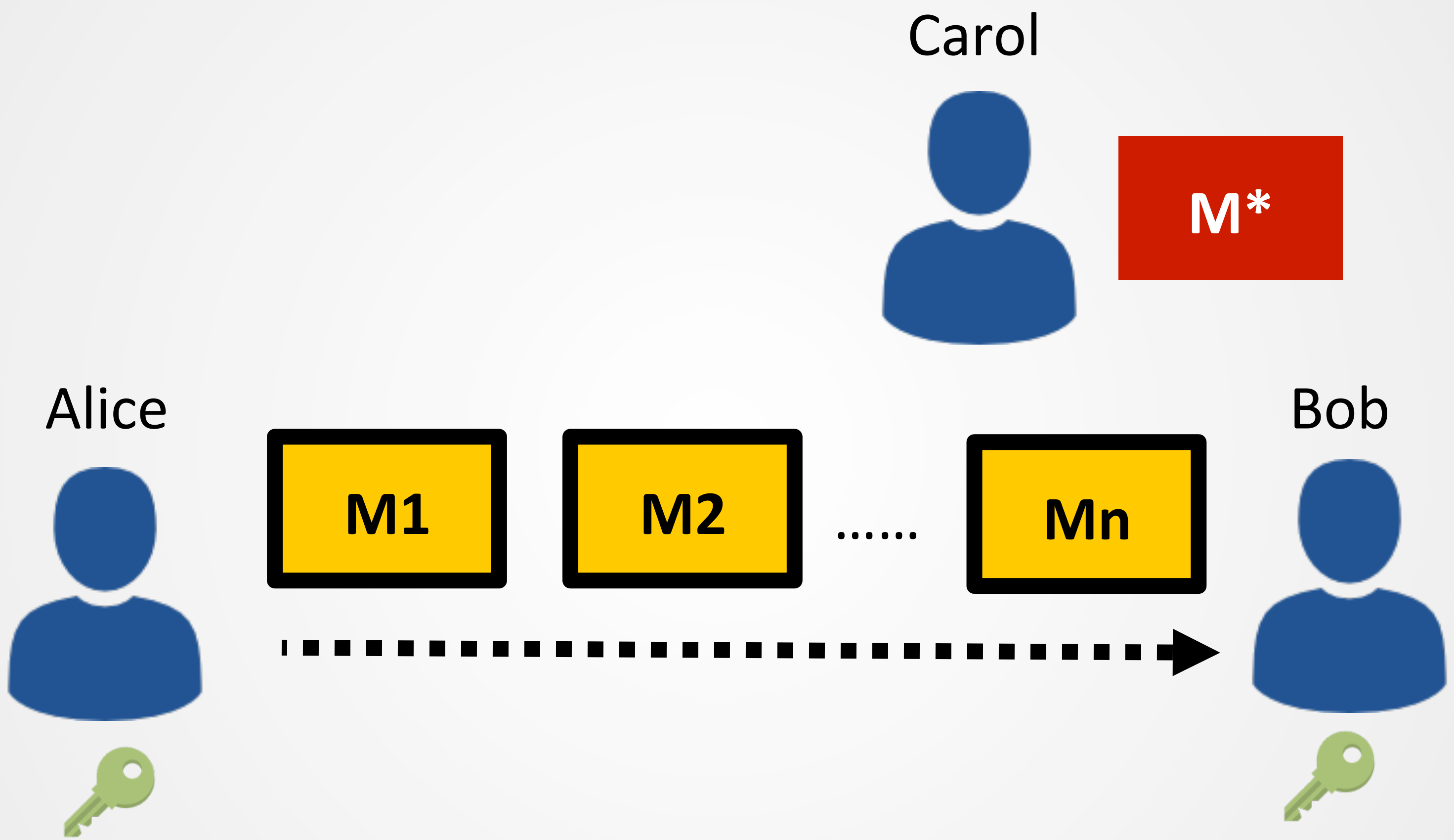
Username: alice
Password: ???

User

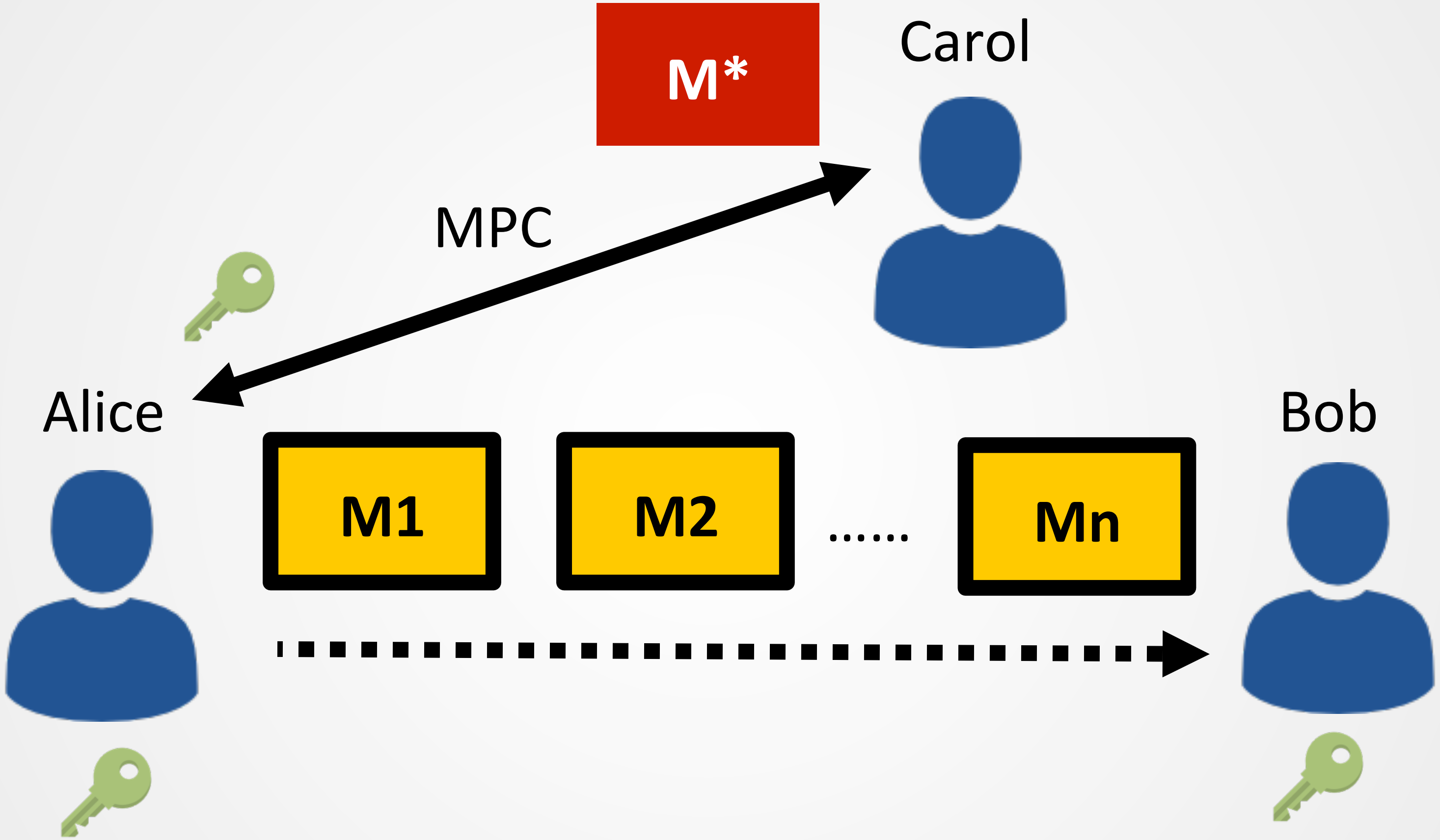
Email provider

Prove account ownership by showing the ability to READ an email from an account

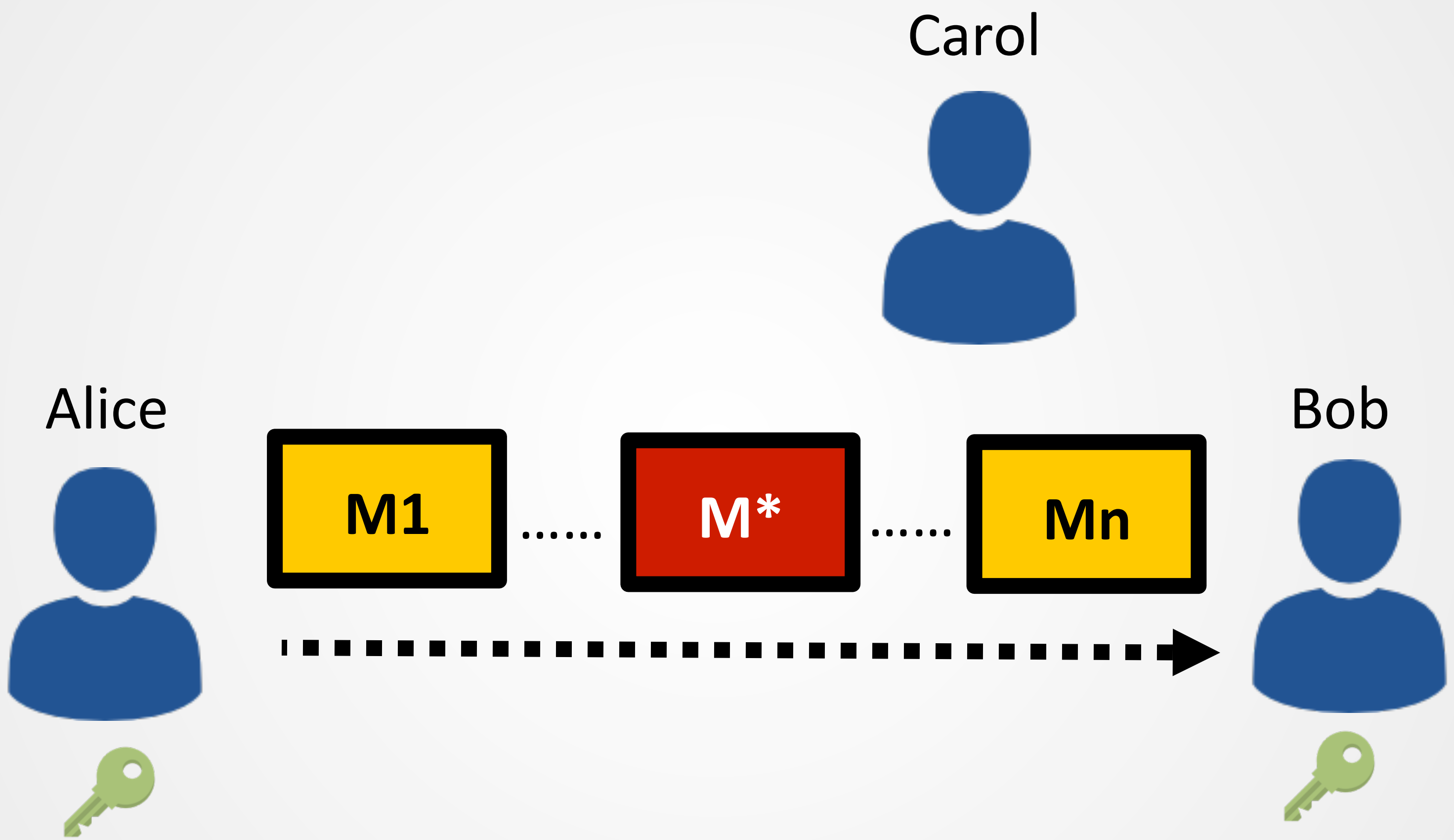
Secure Channel Injection (SCI)



Secure Channel Injection (SCI)



Secure Channel Injection (SCI)

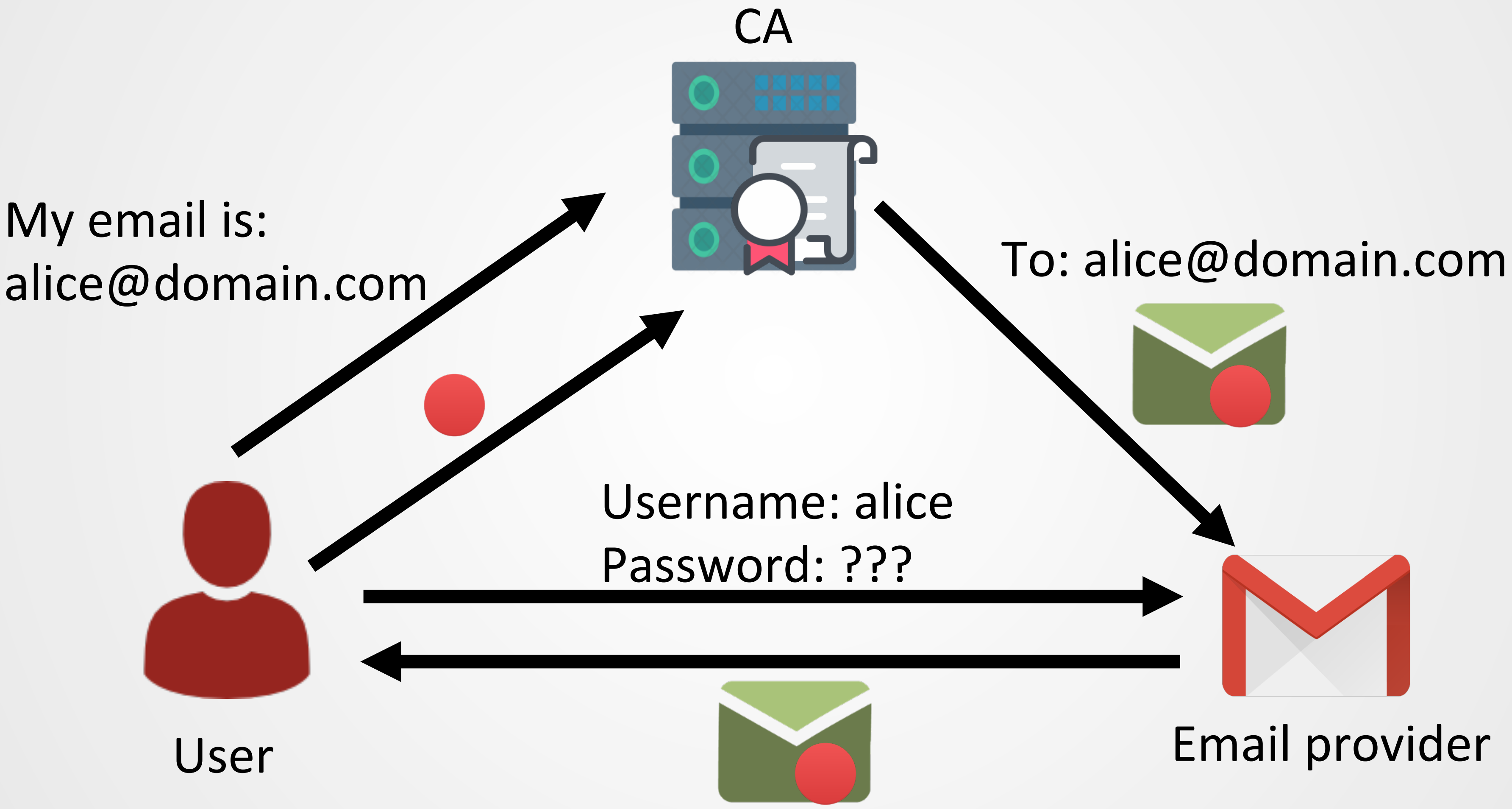


Alice: Learns nothing about M*

Bob: Doesn’t know M* is from Carol

Carol: Learns nothing about other messages from Alice

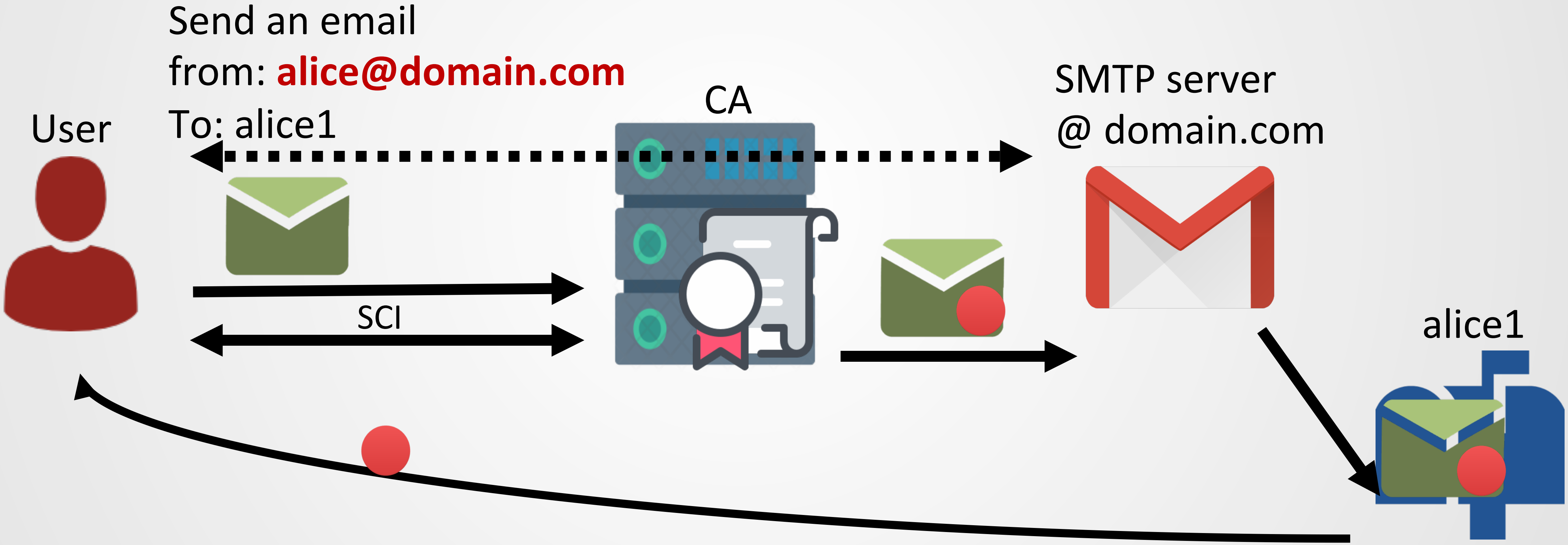
Conventional email verification



Prove account ownership by showing the ability to READ an email from an account

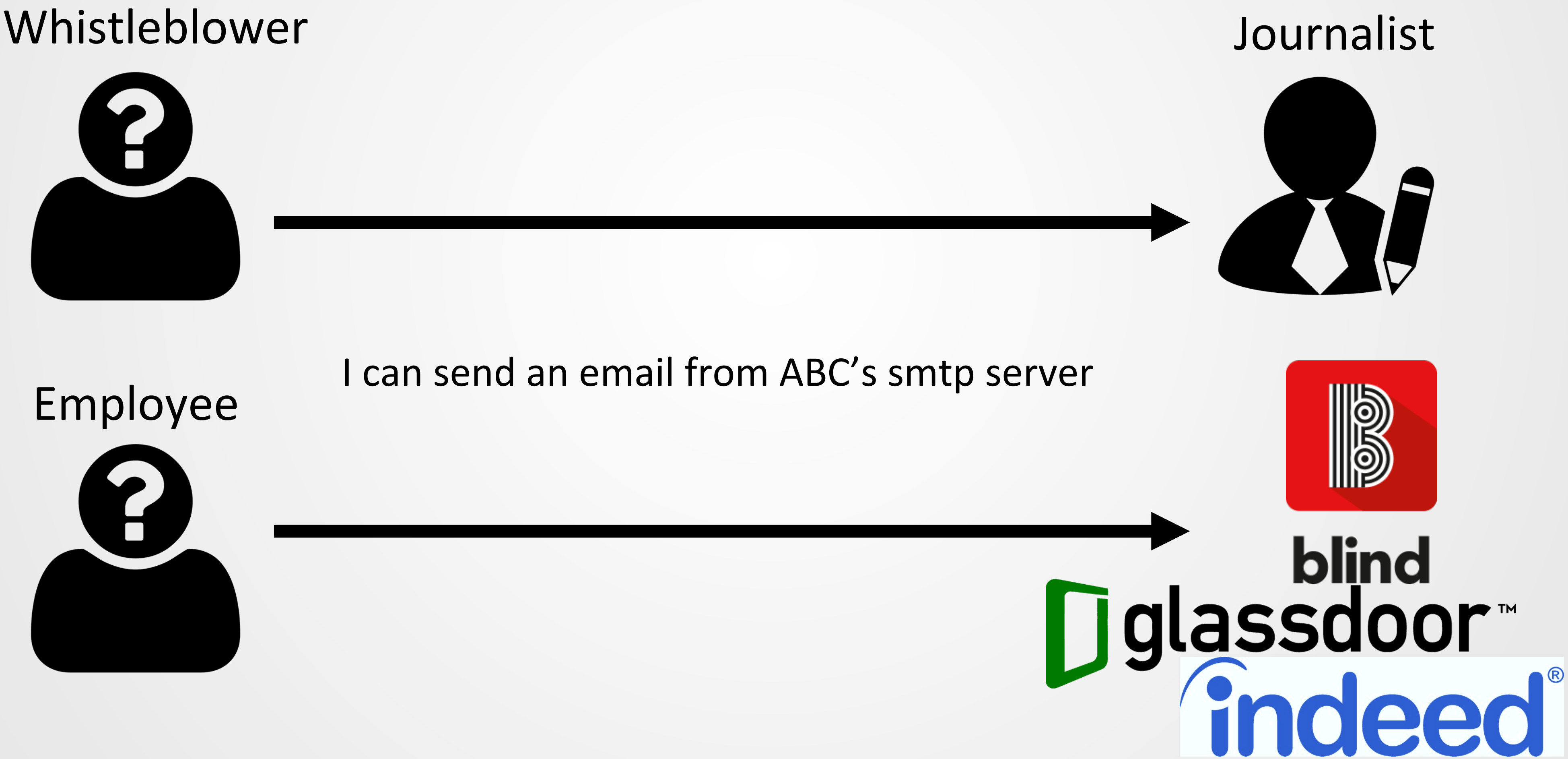
Anonymous proof of account ownership (PAO)

Goal: Validate Alice owns some email accounts from domain.com



Prove account ownership by showing the ability to SEND an email from an account

PAO use cases



Whistleblower

Journalist

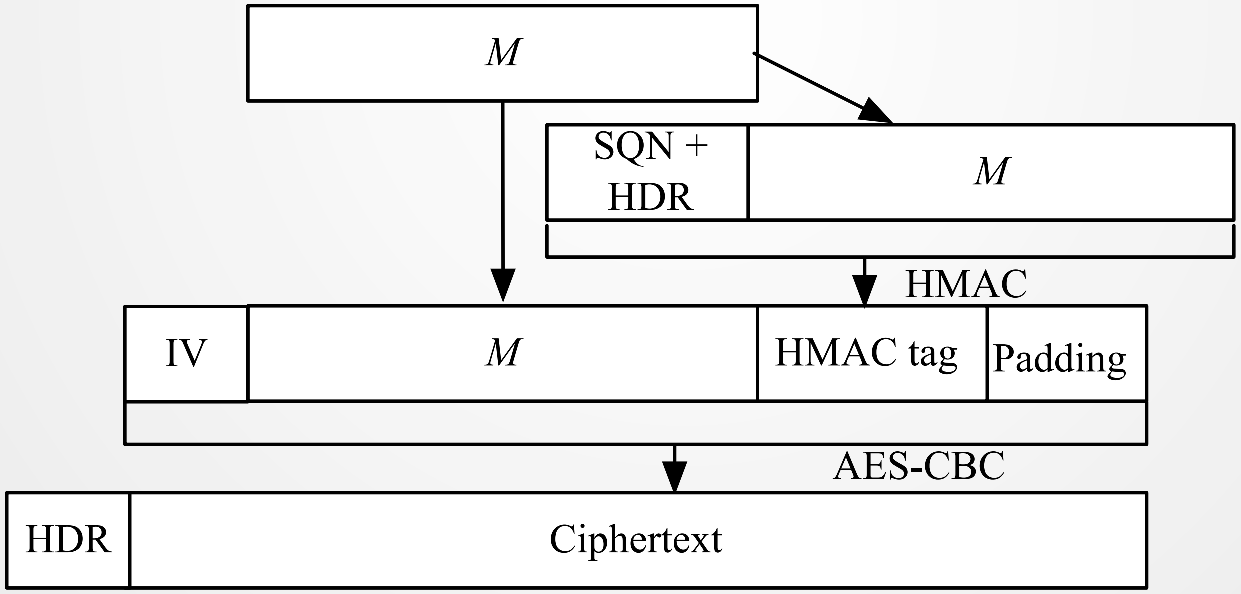
Employee

I can send an email from ABC’s smtp server

Anonymous PAO needs to use MPC to compute TLS records

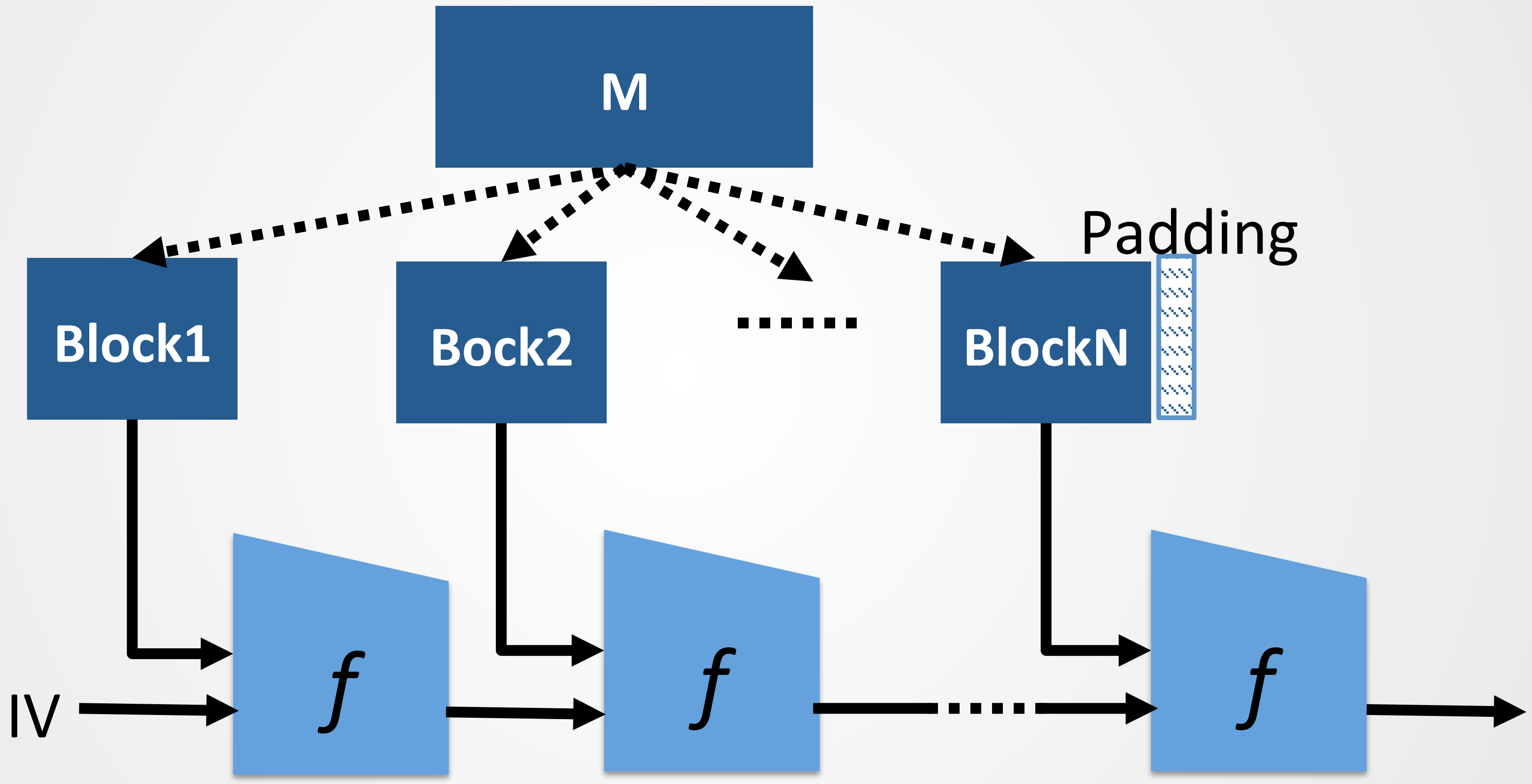
For a 512-byte email and 16-byte challenge

- Generic MPC: 32 AES and 8 SHA256 operations → 0.94M+ AND gates

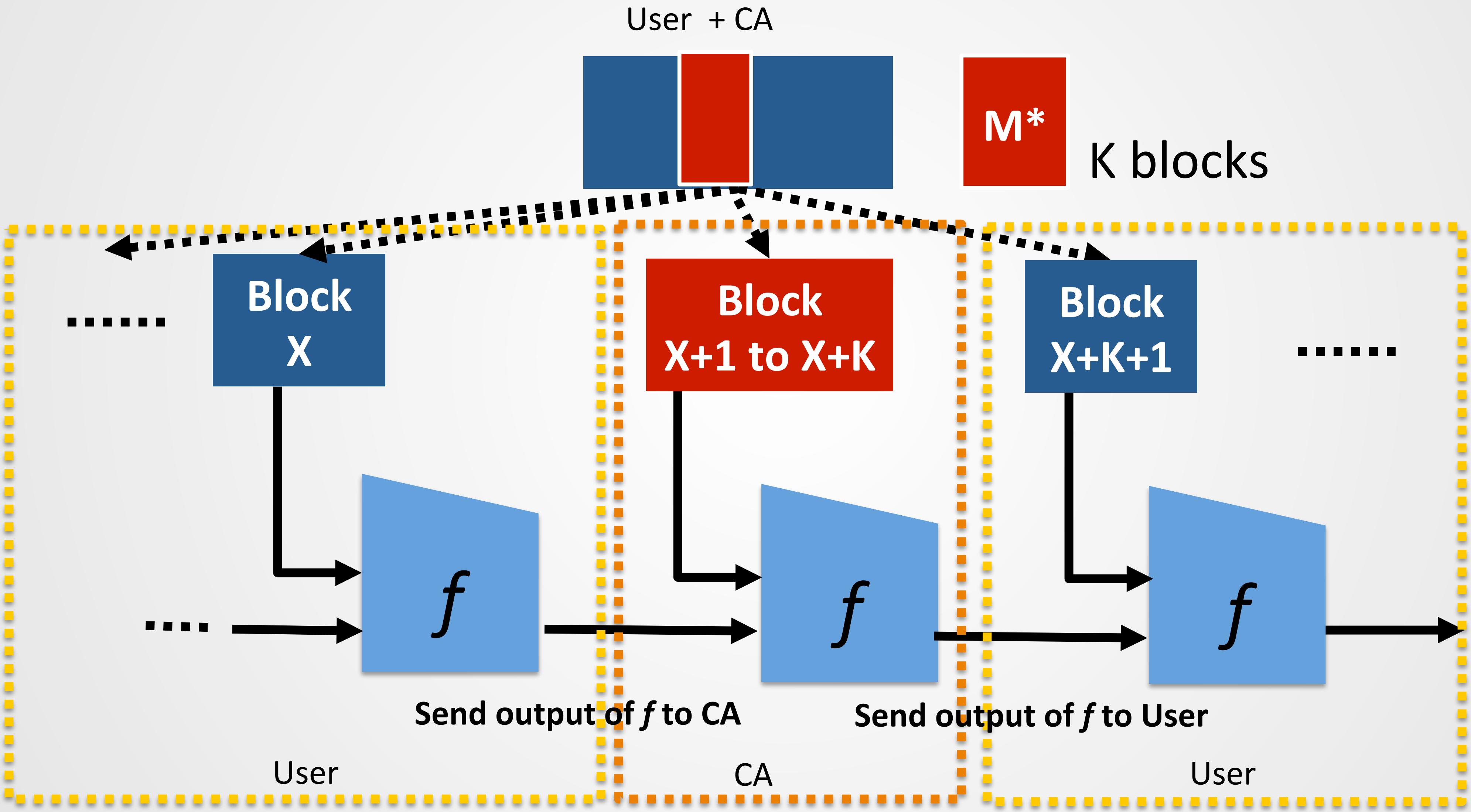


TLS AES-CBC with SHA256

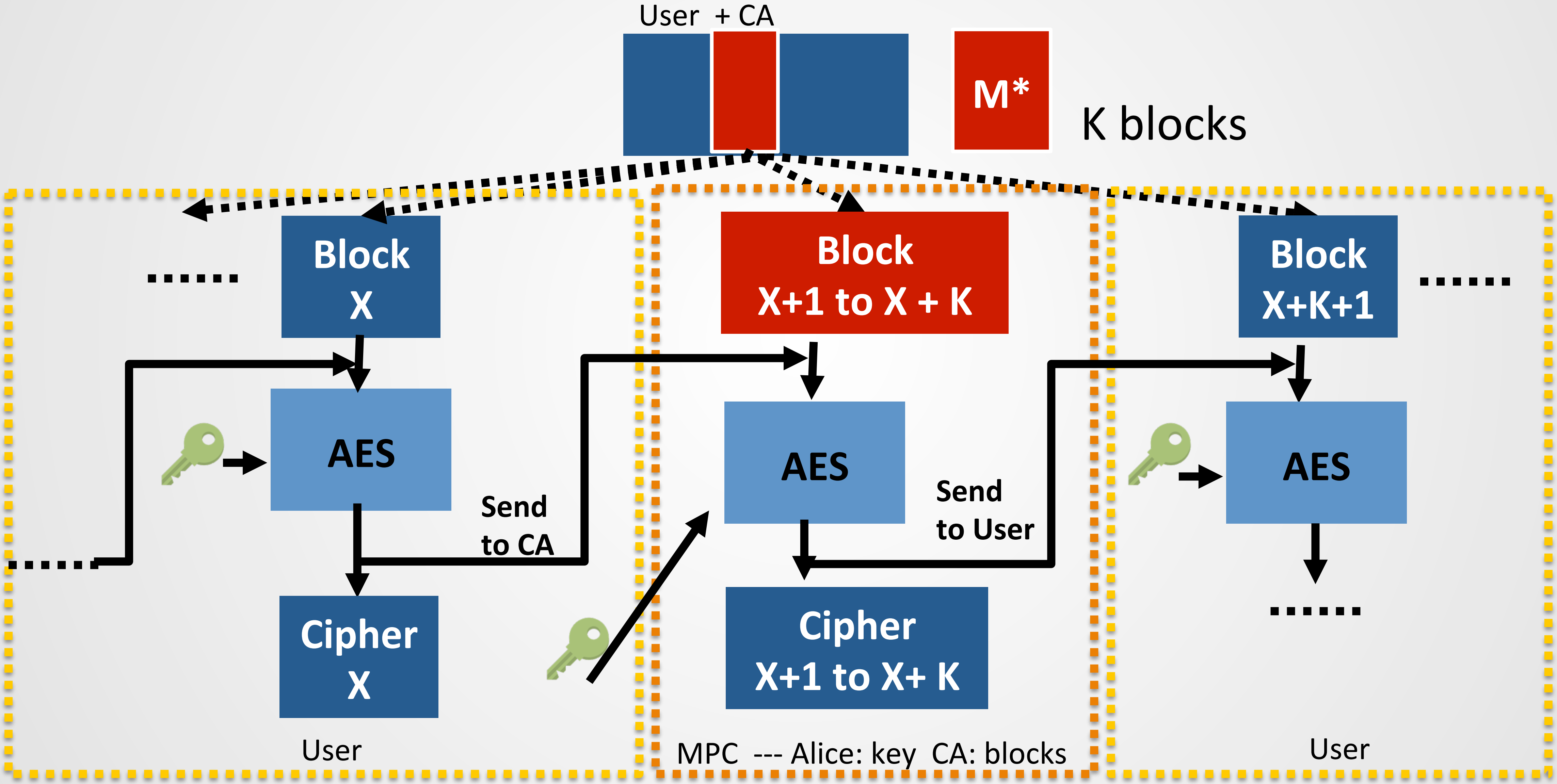
Merkle–Damgård Construction



Two-party SHA: "Outsource" SHA computation



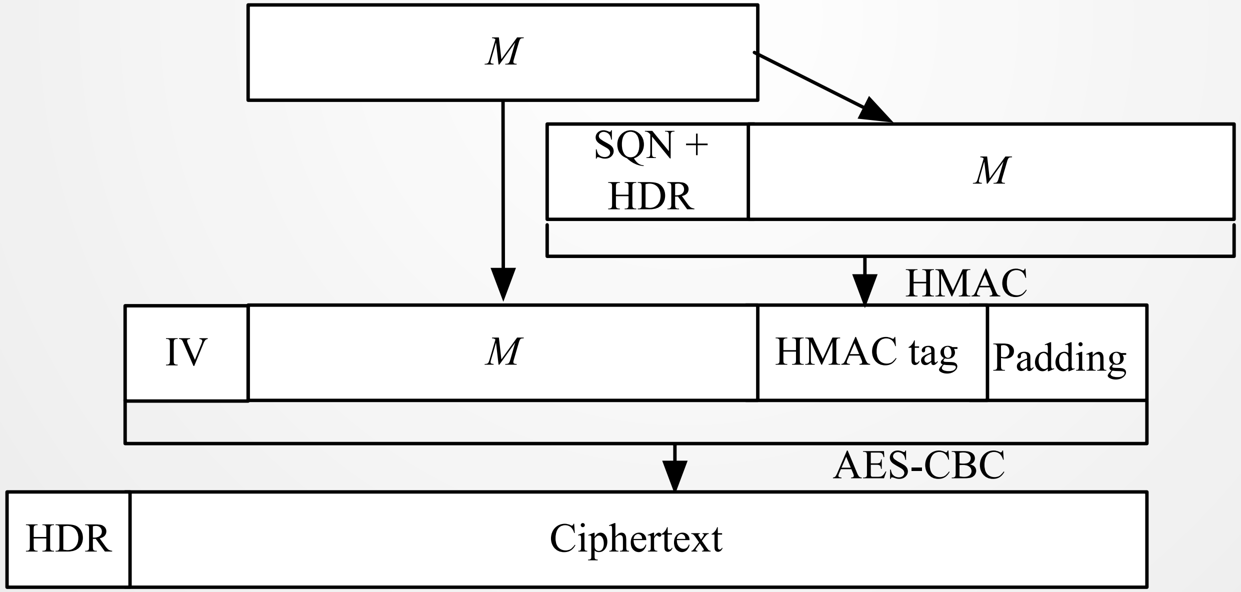
Two-party AES CBC



Anonymous PAO needs to use MPC to compute TLS records

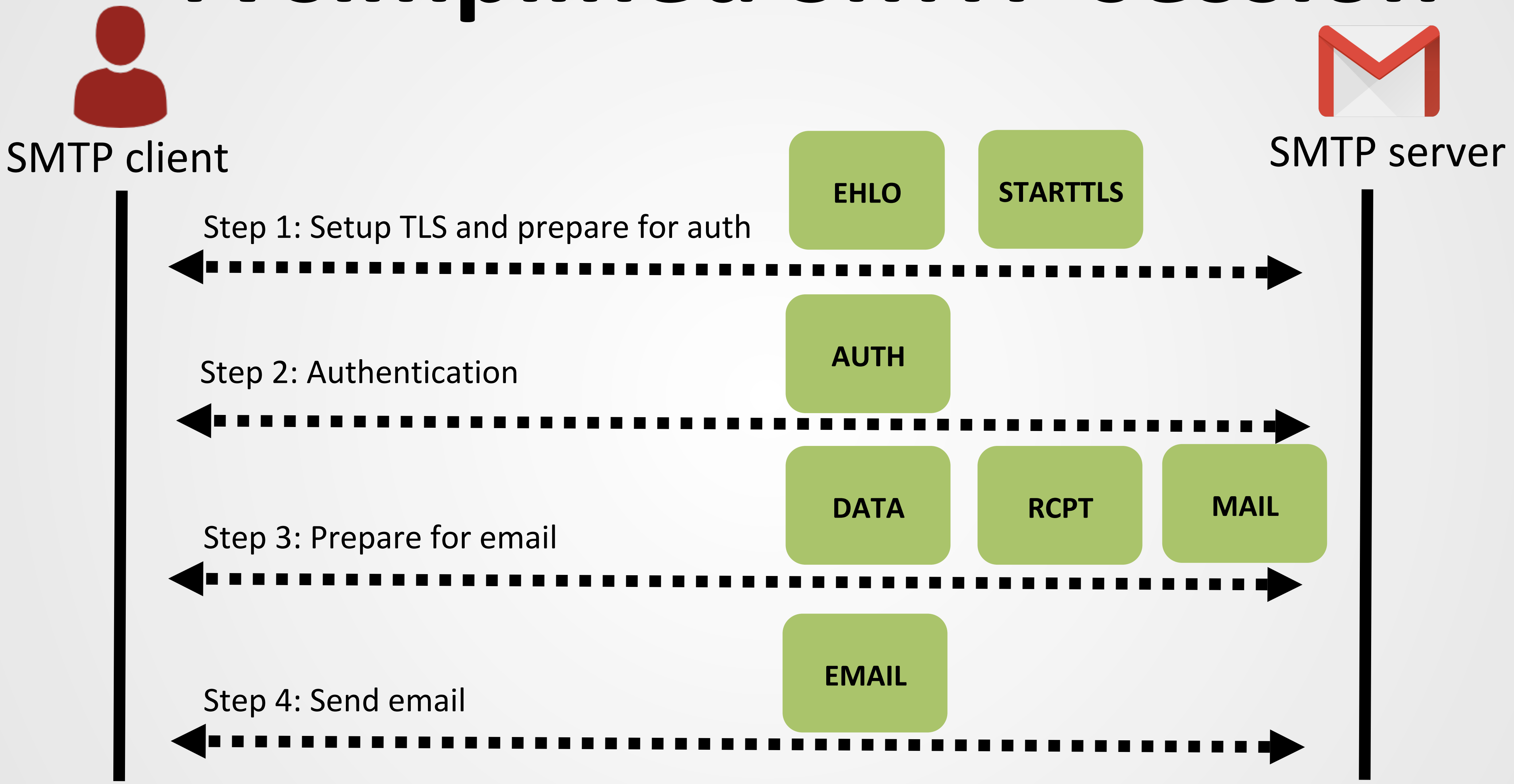
For a 512-byte email and 16-byte challenge

- Generic MPC: 32 AES and 8 SHA-256 operations → 0.94M+ AND gates
- **Our protocol: 4 AES operations → 27K+ AND gates; NO MPC for HMAC**



TLS AES-CBC mode

A simplified SMTP session



SMTP client

SMTP server

Step 1: Setup TLS and prepare for auth

EHLO

STARTTLS

Step 2: Authentication

AUTH

Step 3: Prepare for email

DATA

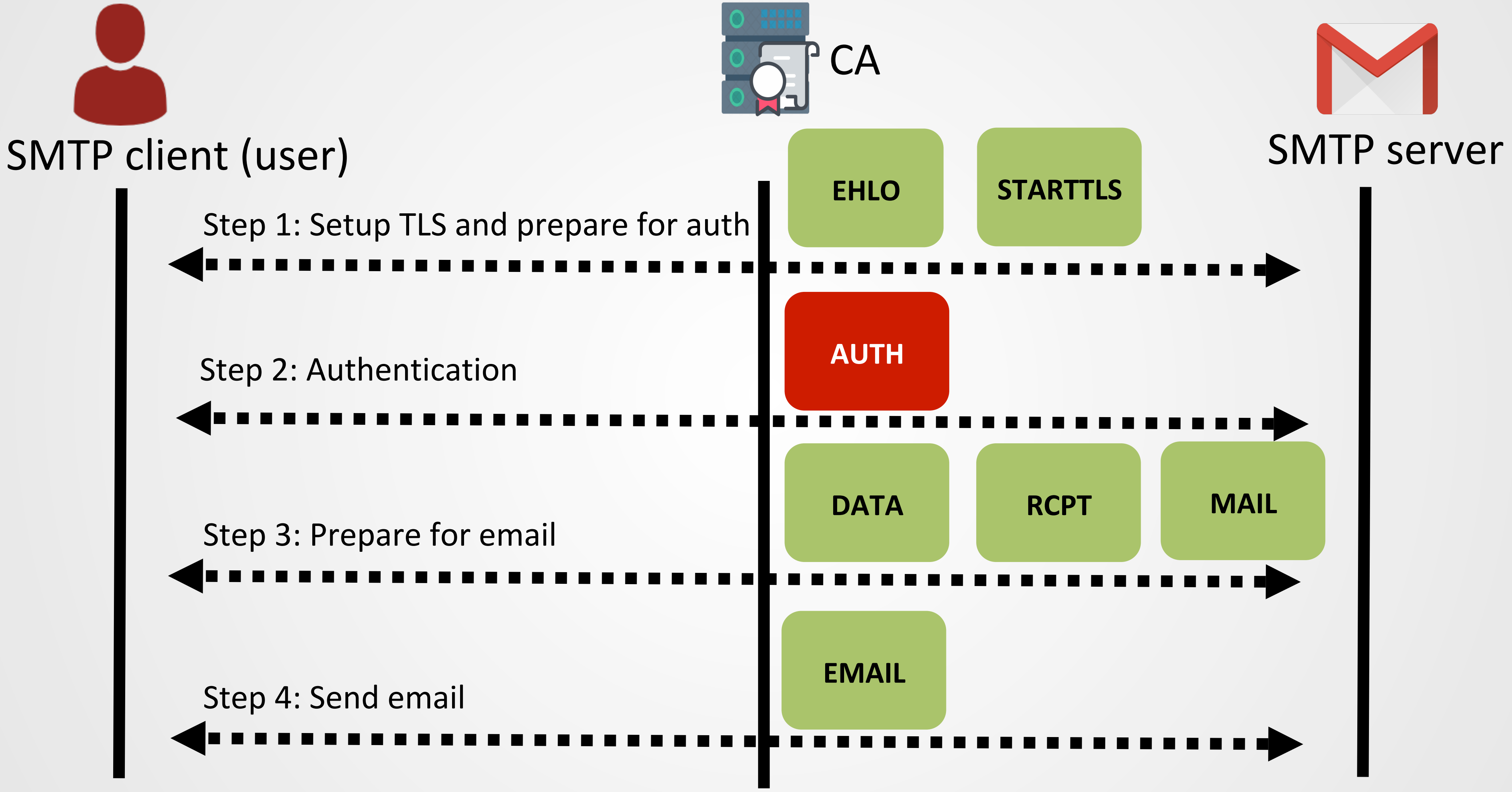
RCPT

MAIL

Step 4: Send email

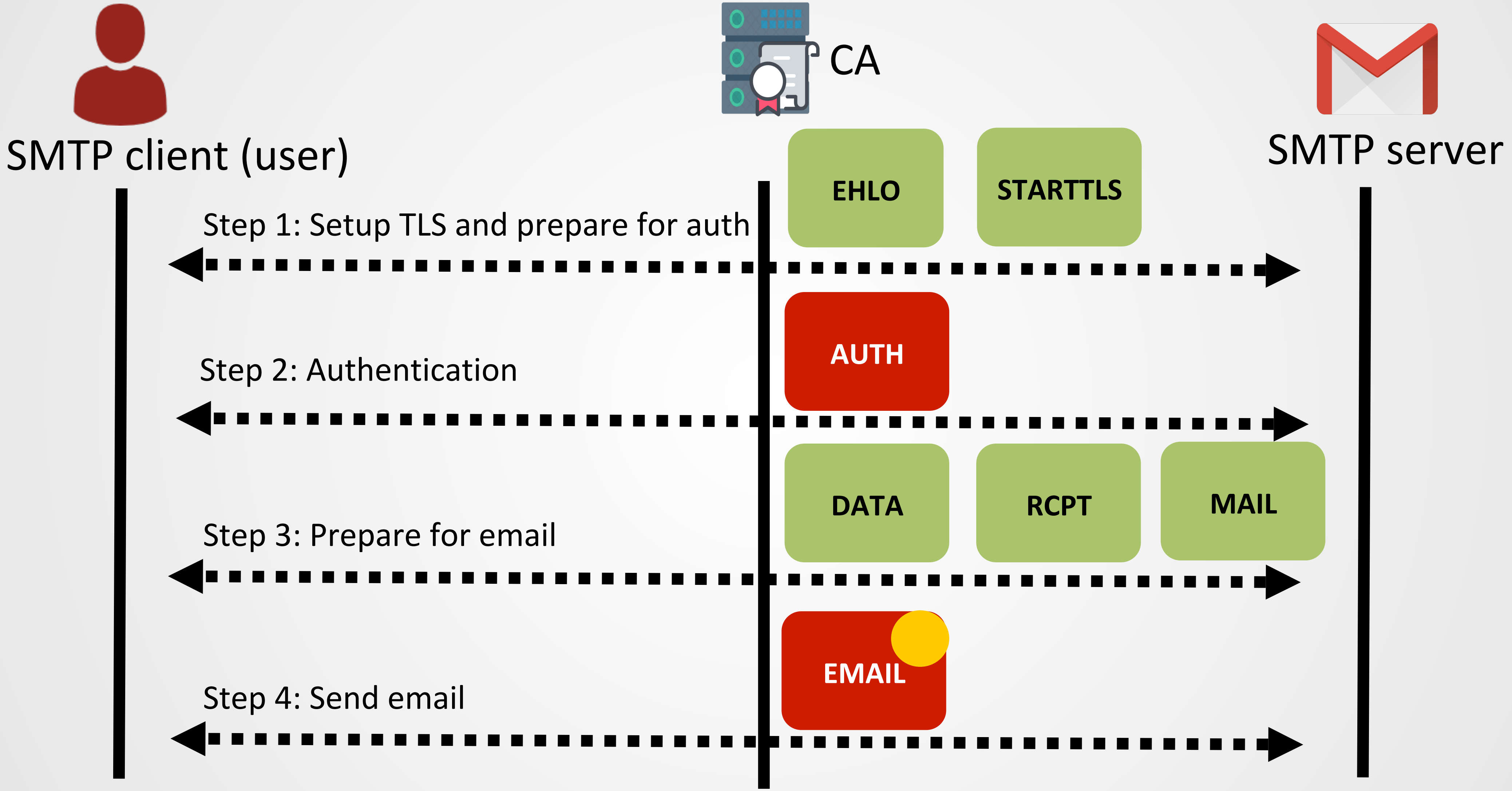
EMAIL

BlindCA: TLS record as commitment

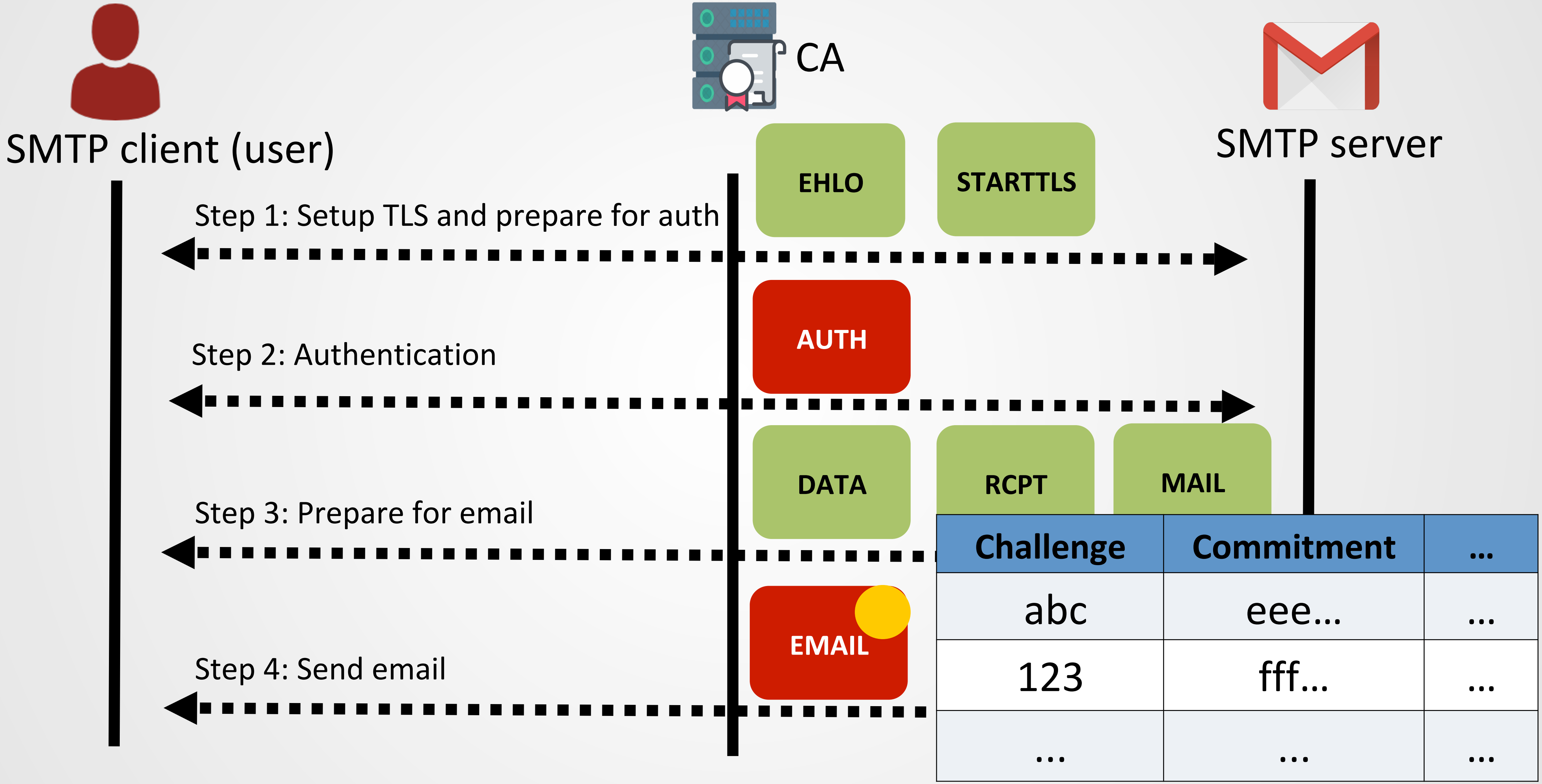


The SMTP AUTH message contains email account (user identity)

BlindCA: Anonymous PAO



BlindCA: Anonymous PAO



CA

SMTP client (user)

SMTP server

EHLO

STARTTLS

Step 1: Setup TLS and prepare for auth

AUTH

Step 2: Authentication

DATA

RCPT

MAIL

Step 3: Prepare for email

EMAIL

Step 4: Send email

Challenge	Commitment	…
abc	eee…	…
123	fff…	…
…	…	…

Prover produces a ZKBoo proof

Issuer: BlindCA
Subject: ?@abc
Public key: ?
Version: …

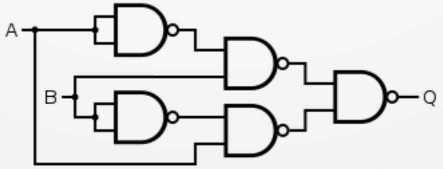
CA: Shares a certificate template with the user

- ○ All fields are known except for subject and public key

User: Fills in missing info, produces the hash of the cert;
Generates a zkboo proof to show the knowledge of:

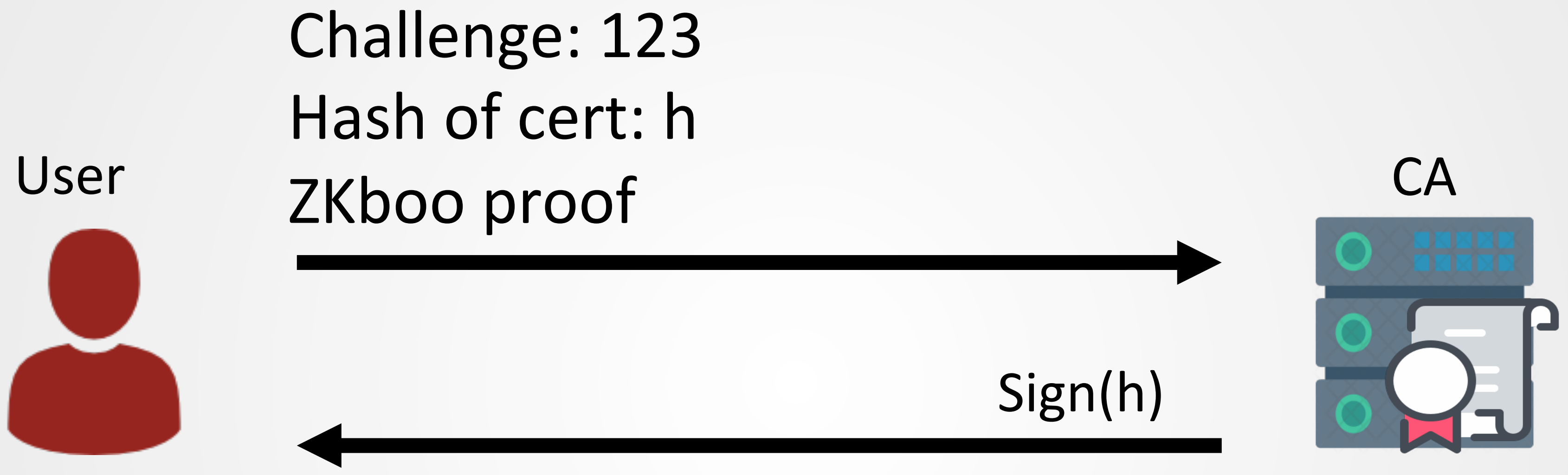
- The email account (e1) and public key for forming the certificate
- The opening of the TLS commitment:
  - ○ secret keys, email account (e2) and password
- e1 = e2

→ **Single Boolean circuit!**



Giacomelli, Irene, Jesper Madsen, and Claudio Orlandi. "Zkboo: Faster zero-knowledge for boolean circuits." USENIX Security 2016.

CA verifies proofs and signs



User

Challenge: 123
Hash of cert: h
ZKboo proof

Sign(h)

CA

Challenge	Commitment	…
abc	eee…	…
123	fff…	…
…	…	…

BlindCA overhead

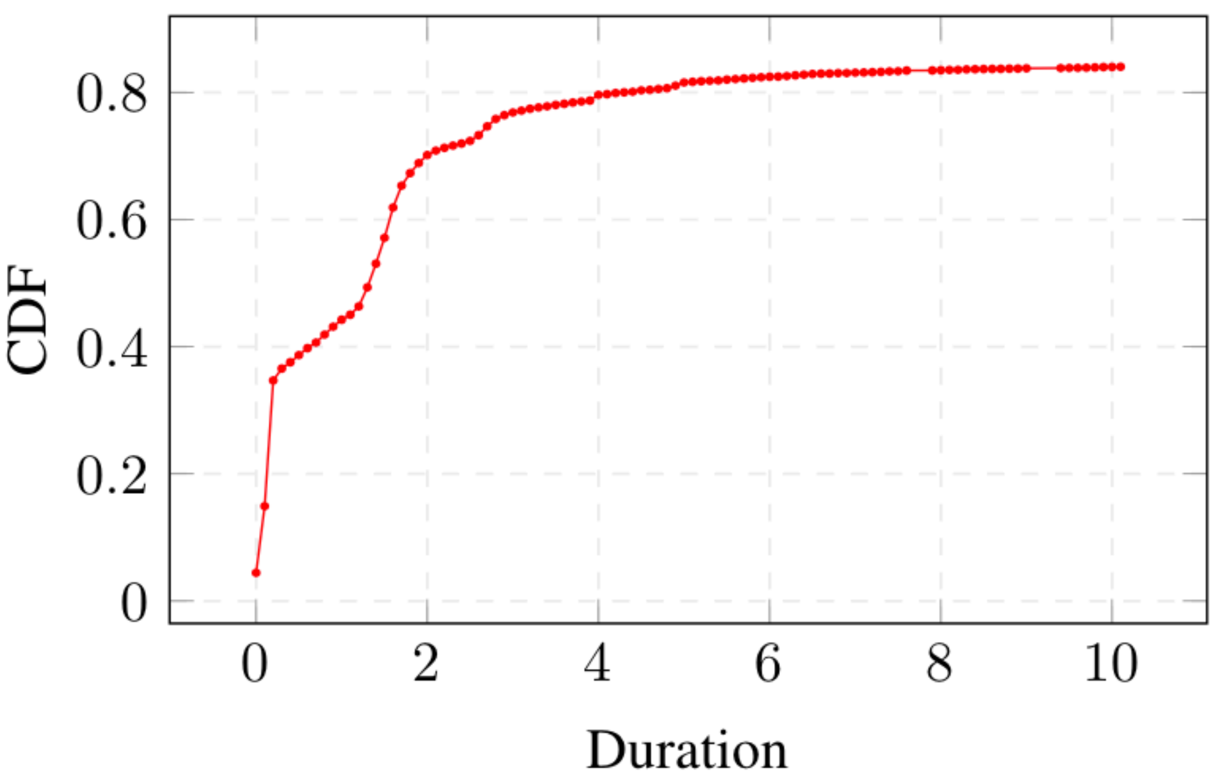
	Loc 1 (No Tor)	Loc2 (No Tor)	Loc1 (With Tor)
2P-HMAC	0.01	0.03	0.31
2P-CBC	0.20	0.35	0.36
PAO	0.76	1.68	4.31
SMTP Baseline	0.31	0.77	3.33

The median time (seconds) to complete the 2P-HMAC, 2P-CBC (without offline), PAO (without offline) and normal SMTP-TLS

- PAO Test with Gmail, UW-Madison, and Cornell SMTP servers:
  - PAO (without offline): 1.01s, 1.64s, 1.53s
  - Without PAO: 0.44s, 0.94s, 0.79s

- BlindCA proof (136 ZKBoo proofs):
  - Size: 85M+
  - Generation: 2.9s
  - Verification: 2.3s

Session duration is not a good detector



15% > 10s!

The distribution of the SMTP durations is long-tailed (based on 8K+ SMTP-TLS sessions).

Summary

- We design the first “blind” CA: a CA that can validate identities and issue certificates without learning the identity
  - SCI for TLS AES-CBC and AES-GCM (see paper)
- Participation privacy: does not disclose to any party the identities of users
- Please see our paper for more details (security proofs, security analysis, etc.)!

Thank you!

Title