Blind Certificate Authorities

Liang Wang\textsuperscript{1}, Gilad Asharov\textsuperscript{2}, Rafael Pass\textsuperscript{2}, Thomas Ristenpart\textsuperscript{2}, abhi shelat\textsuperscript{3}

\textsuperscript{1}Princeton University \quad \textsuperscript{2}Cornell Tech \quad \textsuperscript{3}Northeastern University
Motivation

Certificate Authorities (CA) issue certificates
Certificates bind public keys to identities

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

Third-party or from University ABC

(A friend of Professor X?)
CA: single point of privacy failure

- PGP
- Website login
- Anonymous credential systems

User

Validate identity

Request cert

Identity +

CA (identity provider)

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

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

• **BlindCA:**
  o 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
My email is: alice@domain.com

To: alice@domain.com

Username: alice
Password: ???

User

Email provider

Conventional email verification

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

My email is: alice@domain.com

To: alice@domain.com

Username: alice
Password: ???

User

CA

Email provider

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

Employee

I can send an email from ABC’s smtp server

Journalist

blind

glassdoor™

indeed®
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 $\rightarrow$ 0.94M+ AND gates

TLS AES-CBC with SHA256
Merkle–Damgård Construction

Block1 → f → f → f → Padding

Block2 → f → f → f → Padding

BlockN → f → f → f → Padding

M → f → f → f → Padding

IV → f → f → f → Padding
Two-party SHA: “Outsource” SHA computation

- Block X
- Block X+1 to X+K
- Block X+K+1

User + CA

- M* K blocks

User

CA

Send output of f to CA

Send output of f to User

User
Two-party AES CBC

MPC --- Alice: key  CA: blocks

User + CA

Block X

AES

Cipher X

Send to CA

Block X+1 to X + K

AES

Cipher X+1 to X + K

Send to User

Block X+K+1

AES

User

K blocks

M*
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
A simplified SMTP session

Step 1: Setup TLS and prepare for auth

Step 2: Authentication

Step 3: Prepare for email

Step 4: Send email
BlindCA: TLS record as commitment

1. Setup TLS and prepare for auth
   - EHLO
   - STARTTLS
2. Authentication
   - AUTH
3. Prepare for email
   - DATA
   - RCPT
   - MAIL
4. Send email
   - EMAIL

The SMTP AUTH message contains email account (user identity)
BlindCA: Anonymous PAO

Step 1: Setup TLS and prepare for auth

Step 2: Authentication

Step 3: Prepare for email

Step 4: Send email
Step 1: Setup TLS and prepare for auth

Step 2: Authentication

Step 3: Prepare for email

Step 4: Send email

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Commitment</th>
</tr>
</thead>
<tbody>
<tr>
<td>abc</td>
<td>eee...</td>
</tr>
<tr>
<td>123</td>
<td>fff...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Prover produces a ZKBoo proof

**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

---

CA verifies proofs and signs

Challenge: 123
Hash of cert: h
ZKboo proof

User

CA

| Challenge | Commitment | ...
|-----------|------------|...
| abc       | eee...     | ...
| 123       | fff...     | ...
|           |            | ...
|           |            | ...

Sign(h)
BlindCA overhead

<table>
<thead>
<tr>
<th></th>
<th>Loc 1 (No Tor)</th>
<th>Loc2 (No Tor)</th>
<th>Loc1 (With Tor)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2P-HMAC</td>
<td>0.01</td>
<td>0.03</td>
<td>0.31</td>
</tr>
<tr>
<td>2P-CBC</td>
<td>0.20</td>
<td>0.35</td>
<td>0.36</td>
</tr>
<tr>
<td>PAO</td>
<td>0.76</td>
<td>1.68</td>
<td>4.31</td>
</tr>
<tr>
<td>SMTP Baseline</td>
<td>0.31</td>
<td>0.77</td>
<td>3.33</td>
</tr>
</tbody>
</table>

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

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

15% > 10s!
Summary

• We design the first “blind” CA: a CA that can validate identities and issue certificates without learning the identity
  o 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