20 March 2025

# IETF 122 ACME

## This session is being recorded



# Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to pointyou in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79;please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- BCP 9 (Internet Standards Process)
- BCP 25 (Working Group processes)
- BCP 25 (Anti-Harassment Procedures)
- BCP 54 (Code of Conduct)
- BCP 78 (Copyright)
- BCP 79 (Patents, Participation)
- https://www.ietf.org/privacy-policy/(Privacy Policy)

# Note Really Well

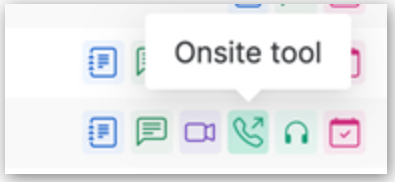
- IETF meetings, virtual meetings, and mailing lists are intended for professional collaboration and networking, as defined in the IETF Guidelines for Conduct (RFC 7154), the IETF Anti-Harassment Policy, and the IETF Anti-Harassment Procedures (RFC 7776). If you have any concerns about observed behavior, please talk to the Ombudsteam, who are available if you need confidentiality to raise concerns confident about harassment or other conduct in the IETF.
- The IETF strives to create and maintain an environment in which people of many different backgrounds and identities are treated with dignity, decency, and respect. Those who participate in the IETF are expected to behave according to professional standards and demonstrate appropriate workplace behavior.
- IETF participants must not engage in harassment while at IETF meetings, virtual meetings, social events, or on mailing lists. Harassment is unwelcome hostile or intimidating behavior—in particular, speech or behavior that is aggressive or intimidates.
- If you believe you have been harassed, notice that someone else is being harassed, or have any other concerns, you are encouraged to raise your concern in confidence with one of the Ombudspersons.

This session is being recorded

# IETF 122 Meeting Tips

**In-person participants**

- Make sure to sign into the session via Datatracker or the QR Code in this session.
- Use Meetecho (usually the “Meetecho lite”) client to:
  - join the mic queue
  - participate in shows of hands
- *Keep audio and video off if not using the onsite version.*




**Remote participants**

- Make sure your audio and video are off unless you are chairing or presenting during a session.
- Use of a headset is strongly recommended.

# Resources for IETF 122 Bangkok

- Agenda
  https://datatracker.ietf.org/meeting/agenda
- Meetecho and other information:
  https://www.ietf.org/how/meetings/preparation
- If you need technical assistance, see the Reporting Issues page:
  http://www.ietf.org/how/meetings/issues/

# Agenda

- Administrivia & agenda bashing
- Document Status
- Presentations:
  - ACME-RI & ACME-Profiles – Gable
  - DTN Node ID Validation – Sipos
  - ACME DNS Update - Li
  - ACME-RATS – Liu
  - JWTClaimConstraints – Wendt
  - ACME PK Challenges – Geng
- AOB

# Document Status (1/3)

- No new RFCs (since September 2023)
- ACME-ARI
  - Published revisions -07 and -08
  - IETF LC 22-Nov – 6-Dec
  - Approved by IESG 27-Feb-2025
  - In RFC Editor's queue
- ACME-Integrations for Device Certificate Enrollment
  - In RFC Editor's queue since 14-Jul-2023
  - Miss-Ref: anima-brski-cloud and lamps-rfc7030-csrattrs
    - Both have gone through IETF LC since IETF 121

# Document Status (2/3)

- ACME-Onion
  - Published revisions -05, -06, and -07 since IETF 121
  - IETF LC ended 26-Nov-2024
  - Approved by IESG 15-Jan-2025
  - In RFC Editor's queue
- ACME DTN NodeID
  - LC in 2021
  - Waited for DTN work
  - Have presentation today.

# Document Status (3/3)

- ACME-Client
  - Two new revisions since IETF 121
  - No ML engagement
- ACME DNS Account Label
  - Just one revision - -00.
  - Have presentation

# Presentations

# ACME-RI & Profiles

# ACME Renewal Information

draft-ietf-acme-ari-08

Aaron Gable, Let’s Encrypt
IETF 122, 2025-03-20

# Since IETF 121

- Draft -06
  - Sent to IESG for consideration as a Proposed Standard
- Draft -07
  - Rewrote client scheduling requirements and recommendations
  - Added security considerations re: clock skew
  - Many small editorial improvements
  - Sent to IESG ballot
- Draft -08
  - Further improvements to client scheduling
  - Changed some client SHOULDs to MUSTs
  - Many more small editorial improvements
  - Approved by the IESG as a Proposed Standard
  - Sent to the RFC Editors' queue

# Thank you for all the reviews and feedback!

- David Dong
- Deb Cooley
- Éric Vyncke
- Geoff Huston
- Jacob Hoffman-Andrews
- Mahesh Jethanandani
- Michael Tüxen
- Murray Kucherawy
- Paul Wouters
- Richard Barnes
- Sabrina Tanamal
- Shawn Emery
- Susan Hares
- Yoav Nir
- Zaheduzzaman Sarker

# ACME Profiles

draft-aaron-acme-profiles-00

Aaron Gable, Let’s Encrypt
IETF 122, 2025-03-20

# Since IETF 121

- No updates to draft-aaron-acme-profiles-00

- Deployed in Let's Encrypt's staging and production environments
  - Three profiles configured in each

- Implemented in ACME clients:
  - certbot (library / cli)
  - lego
  - eggsampler
  - acmez / certmagic / caddy
  - certifytheweb
  - posh-acme

# Next Steps

- Questions / comments / feedback?
- Call for adoption?

# ACME DTN NodeId Validation

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

# ACME DTN Node ID Validation

## IETF 122 ACME WG

Brian Sipos
JHU/APL

# Current Status

- Latest is draft-ietf-acme-dtnnodeid-16
- Dependency draft was published as RFC 9713
- Changes since -14:
  - Update validation method name to “bp-nodeid-00” (from “dtn-nodeid-01”)
  - Update dependency reference to published RFC
- Need re-confirmation from WG to pass last-call to IESG review

# ACME DNS Update

# Secure DNS RR Update for ACME DNS Based Challenges

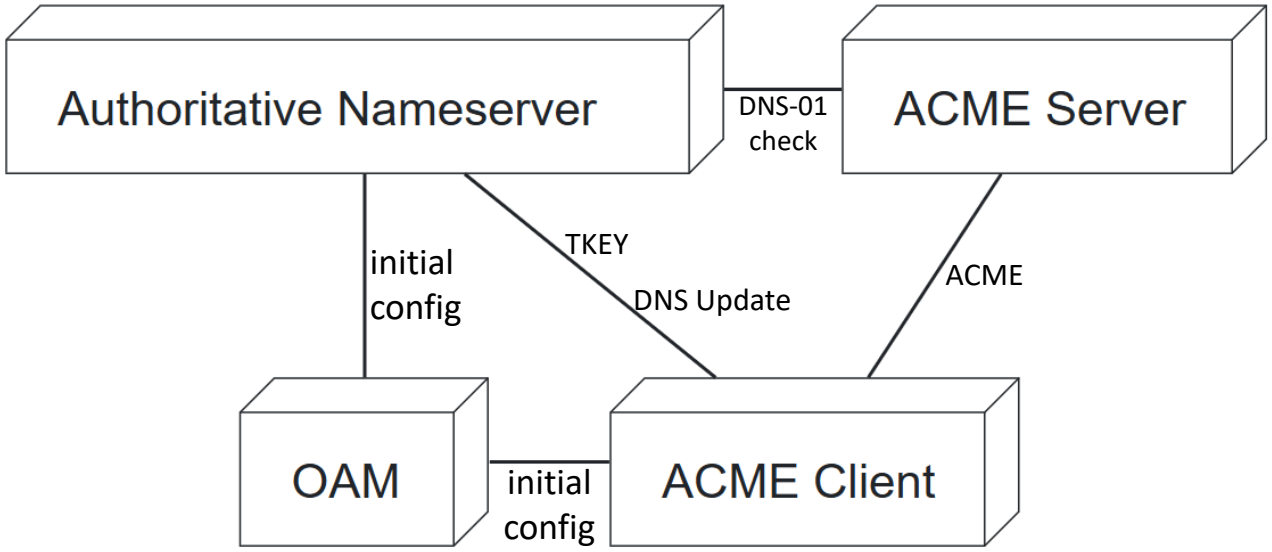
draft-li-acme-dns-update

**Ruochen Li**, Haiguang Wang, Zhongding Lei

# Problem Statement

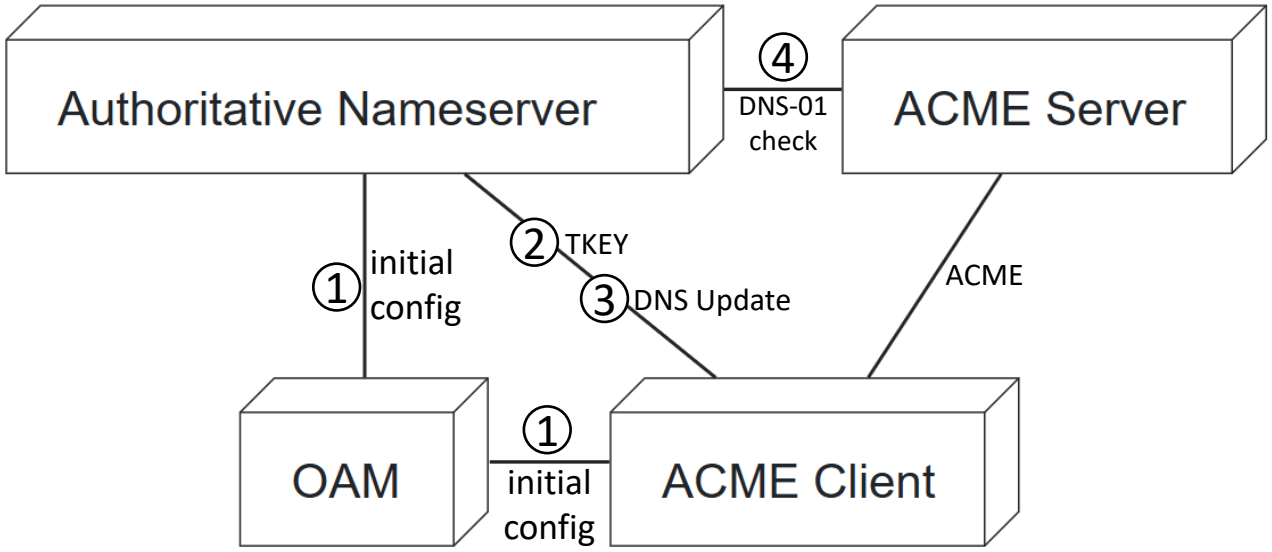
- **ACME DNS Challenges**: Require clients to prove domain control via DNS records (e.g., TXT entries).
- **Gap**: No recommended procedure for secure DNS record updates for ACME.
- **Solution**: Define a recommended DNS challenge procedure using ***DNS UPDATE*** (RFC 2136) with ***TSIG*** (RFC 8945) and/or ***SIG(0)*** (RFC 2931) for authentication.

# Architecture Overview



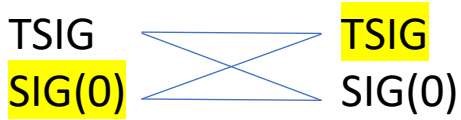

- **OAM** configures initial TSIG keys for clients.
- **ACME Client** uses keys to authenticate DNS updates.
- **Authoritative Nameserver** validates updates via access control.
- **ACME Server** verifies records for certificate issuance.

# Procedure Overview




1. Initial Setup
   - OAM configures domain and TSIG/SIG(0) keys on client & nameserver.
2. (Transaction Key)
   - Client establishes transaction TSIG key (via TKEY RFC 2930).
3. DNS Update
   - Client adds/removes ACME DCV records (TXT) with TSIG.
4. DNS Challenge Validation
   - ACME server fetches records to confirm domain control.

**Initial key:** **Txn key:**



# Things to Consider

- Access control of the keys on the DNS server (for DNS Update and transaction key establishment, restrict to specific domains/record types)
- DNS Update replay prevention
- Key rotation, short-lived keys
- Apply DoT/DoH to encrypt DNS Update traffic
- Clean up after use (DCV records & transaction keys)

# Thank You!

# ACME - RATS

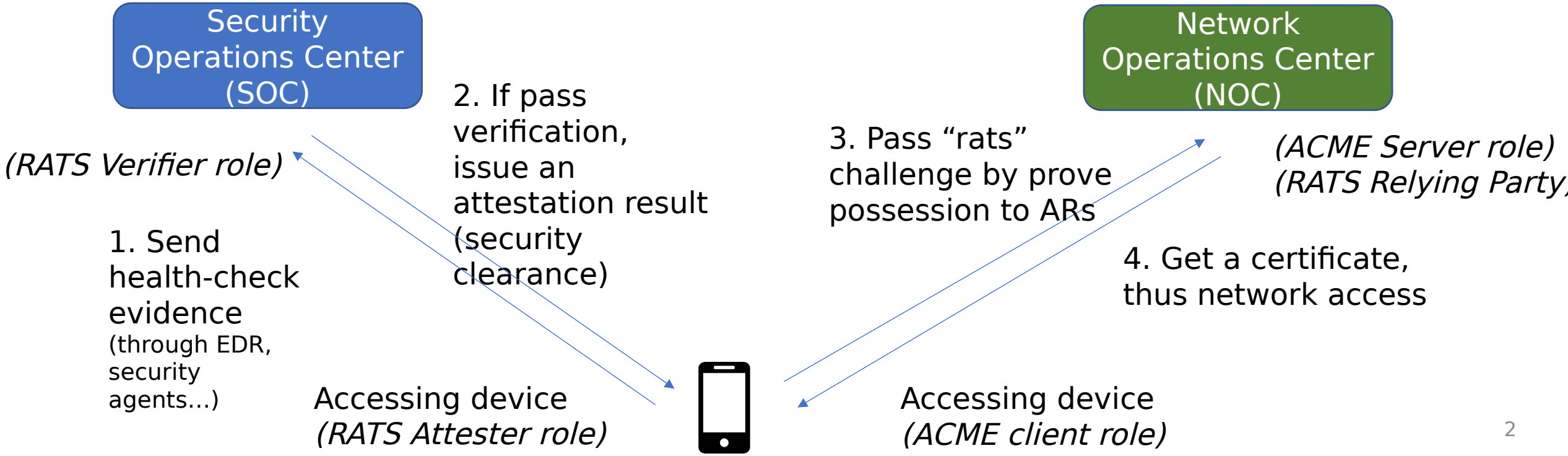
# draft-liu-acme-rats

Chunchi Peter Liu, Mike Ounsworth, Michael Richardson

IETF 122 Bangkok

# Why – client side certificates for mutual TLS
## use case: Grant internal **enterprise network access** (certificates) to devices that pass security checks

- Enterprise usually have 2 teams: Security ops team and Network ops team, where to draw the line is always hard. This work helps them collaborate.



# Other Motivating Usecases

- An ACME server (cloud domain host service provider) might put a policy on their ACME client (domain owner tenant)
- A CA (ACME server) may want to know certain attributes of the private key or of the device or application that will use it:
  - Private key resides in FIPS level 3 hardware and has `non-exportable=true`.
  - The policies that apply to certain (cloud) Key Management Service (KMS) instances.
  - TLS / OS / Docker stacks have been recently patched (ie <= 3 months old).
  - Etc.

# Changes from last IETF

1. Changed “rats” challenge type to “device-attest” challenge type
   - Referencing `draft-bweeks-acme-device-attest-01`
2. Deleted `http` challenge type
3. Changed how the response carry evidence or attestation result
   - `keyAuthorization = token || '.' || ~~base64url(attestationResult)~~ cmw`
4. Added ACME Attest Claims Hint Registry
   - To facilitate the Server requesting attestation of specific types claims or properties
   - Add IANA request for a new registry
5. Mike and Michael provided contents and reviews, so added them as co-authors.
   - Those who wish to collaborate please do the same? :)

But we think maybe we want to re-think the design

# Is Challenge the right place for remote attestation?

- draft-acme-device-attest-01 proves ownership of a permanent identifier, using device-attest-01 challenge.
- More **general** remote attestation may need to go in a different place in the ACME flow, not in the Challenge because:
  - Challenges are for proving ownership of identifiers, not system attributes.
  - Technical problem: The client is only required to respond to ONE challenge.
  - A separate system-level remote attestation could complement the identity challenge.

- So Challenge is not the right place.

# Design Space – Proposed direction

POST /acme/order/TOlocE8rfgo/finalize HTTP/1.1

```
{
 "protected": base64url({
  "alg": "ES256",
   … snip …
 }),
 "payload": base64url({
  "csr": "MIIBPTCBxAIBADBFMQ…FS6aKdZeGsysoCo4H9P",
  “attestation”: “…”,
 }),
 "signature": "uOrUfIIk5RyQ…nw62Ay1cl6AB"
}
```

Remote Attestation can go inside the CSR using draft-ietf-lamps-csr-attestation

OR in a new “payload.attestation” param so that attestation(s) can be carried next to the CSR.

# Design Space – Proposed Direction

The ACME server needs a way to:

1. Request that the ACME client **provides evidence/attestation results** of the requesting device.
   - For example in the POST /finalize.
2. Request **specific** remote attestation attributes / claims.
   - For example “proof-of-FIPS-level3”, “proof-of-OS-up-to-date” or “proof-of-application-stack”.
3. Provide an attestation **freshness nonce** to the ACME client.

- Idea (undeveloped): add this to the 201 CREATED response to POST /new-order.
  - Maybe in “authorizations” ??

# Next steps

- We would like to start an ACME WG affiliated design team meeting once a month. (Maybe? Tuesdays or Thursdays early-ish morning
  N. America time. TBD)
- If you want in, email:
  - Liuchunchi(Peter) <liuchunchi@huawei.com>
  - Mike Ounsworth mike.ounsworth@entrust.com

- Henk has reviewed this slides.

# JWTClaimConstraints

# IETF122 – Bangkok
# ACME

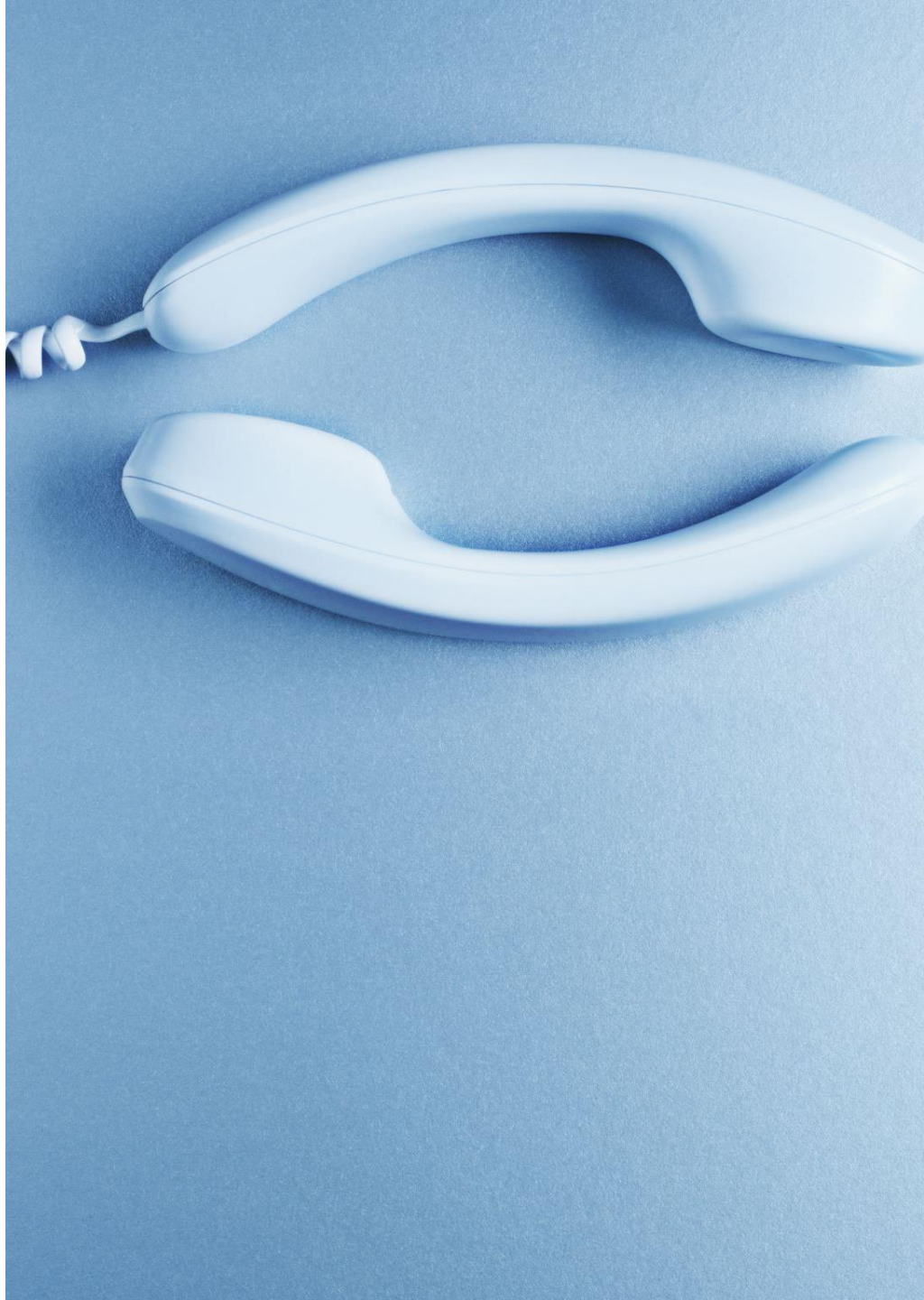
# JWTClaimConstraints Authority Token Profile

draft-ietf-wendt-acme-authority-token-jwtclaimcon-00

Chris Wendt, David Hancock

# Introduction to JWTClaimConstraints Authority Token Draft

- Defines an **Authority Token Profile** for ACME challenge validation specific to JWTClaimConstraints defined in STIR WG.
- Extends the existing Authority Token validation model [RFC9447] to include JWTClaimConstraints extensions.
- Enables authoritative validation necessary for Secure Telephone Identity (STI) including the ability to exclude not allowed claims and adding integrity constraints for Rich Call Data.



# Motivation and Background

- Current ACME **TNAuthList Authority Token Profile** (RFC9448) addresses TNAuthList extension as part of RFC8226
- *JWTClaimConstraints* extension also defined in RFC8226 and extended by EnhancedJWTClaimConstraints in RFC9118 is not addressed.
- *JWTClaimConstraints* define contraints for usage and values of claim for PASSporTs (RFC8225) used in stir authentication.

# Key Features of the Draft

- Defines new ACME Identifier Type: `"JWTClaimConstraints"`.
- Provides Authority Token Profile definition for:
  - Format used for Authority Token creation and acquisition from proper issuers of delegate certificates or associated Rich Call Data validated information.
  - ACME challenge specific to *JWTClaimConstraints* validation.

# How It Works (High-Level Flow)

1. ACME client constructs a CSR with set of JWTClaimConstraints associated with claims desired.
2. Request to Token Authority for issuing a validated JWTClaimConstraints Authority Token representing client's authority over specified claims.
3. ACME client fulfills token-based challenge with returned token (tkauth-01).
4. ACME server verifies the authority token claims, signature, and account key fingerprint.
5. CA issues STI certificate with verified JWTClaimConstraints extension.

# Next Steps

- Seeking WG review and adoption as a working item.
- Was presented at STIR working group and will commit to keep in loop in parallel.
- Aim to rapidly progress toward RFC assuming agreement this is fairly straight-forward.

# ACME Public Key Challenges

# ACME Extension for Public Key Challenges

draft-geng-acme-public-key-01

Feng Geng, Panyu Wu, Liang Xia
Huawei

IETF 122

# Summary

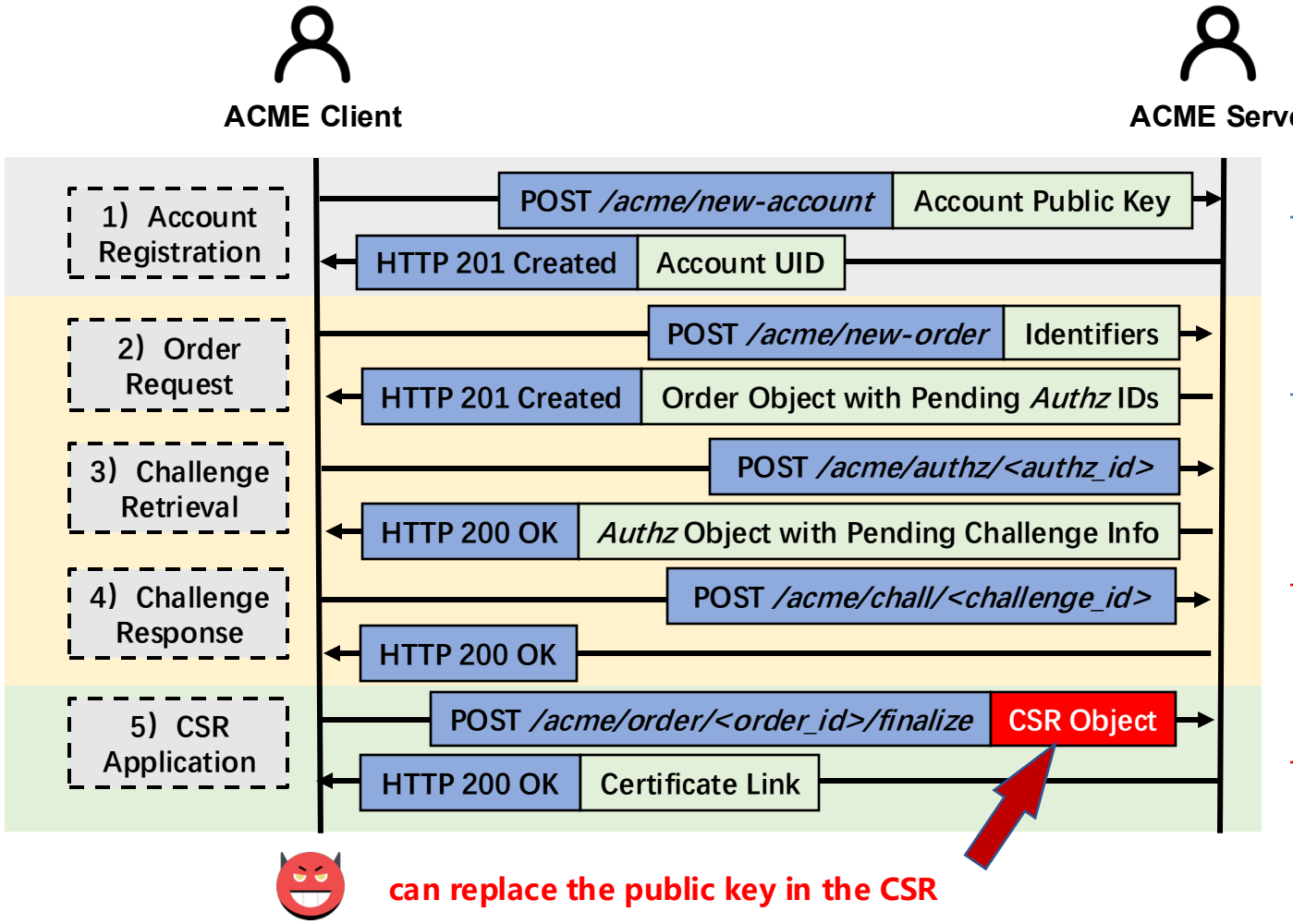
## Present ACME pk-01 challenge

- **Identify** the Public Key replacement risk of the ACME Phase 2 and Phase 3;
- e.g. Risks on **ACME client deployed on EMS**(element management system) to apply certificates for managed network element;
- **Defending against Public Key replacement attacks;**

- An optional process for removing CSR;
- For user/device to apply for their own certificates, it can be restricted by adding whitelisting measures;
- **Best Practices: ACME + OPAQUE** for issue short term certificates to people;

# Backgrounds – Identify risks and motivations

- Account Management phase
- Challenge phase (Identifier Validation)
- Certificate Issuance phase




All messages are protected by the public key signature of the account. It's end-to-end security.

PK replacement attack: Phase 2 just challenge the applying permissions of certificate, but not security binding the public key in CSR. It's not end-to-end security.

Motivation: the pk-01 can be used to verify that the client wants a given key to appear in its certificate and/or that it controls the corresponding private key.

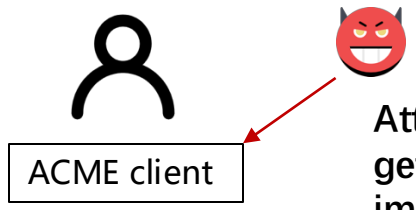
# Two categories and attack scenarios

Our work mainly focus

## resource category

proof of ownership of resources

| type | identifiers | rfc/draft | status |
|---|---|---|---|
| dns-01 | dns | rfc8555 | standard |
| http-01 | dns/ip | rfc8555/8737/8738 | standard |
| tls-alpn-01 | dns/ip | rfc8555/8737/8738 | standard |
| dns-account-01 | dns | draft-ietf-acme-dns-account-label-00 | draft active |
| onion-csr-01 | dns | draft-misell-acme-onion-07 | draft active |

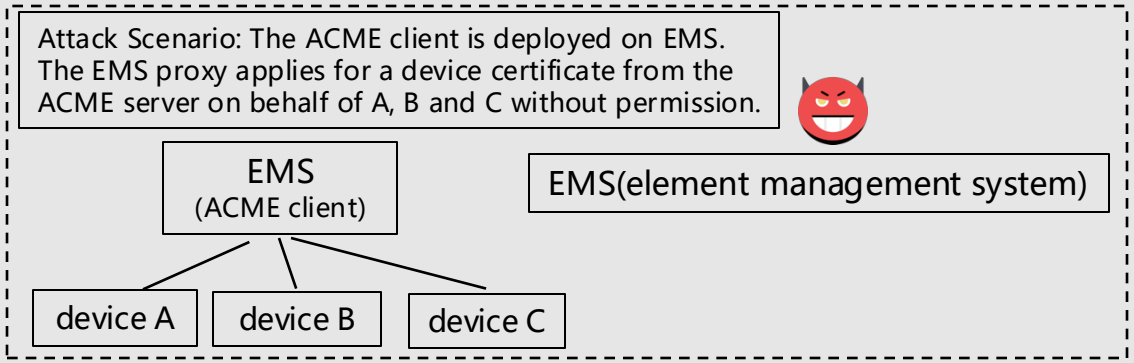



Attack Scenario:
get ACME account credentials
impersonates the victim's server

## user/device category

proof of ownership of digital identity

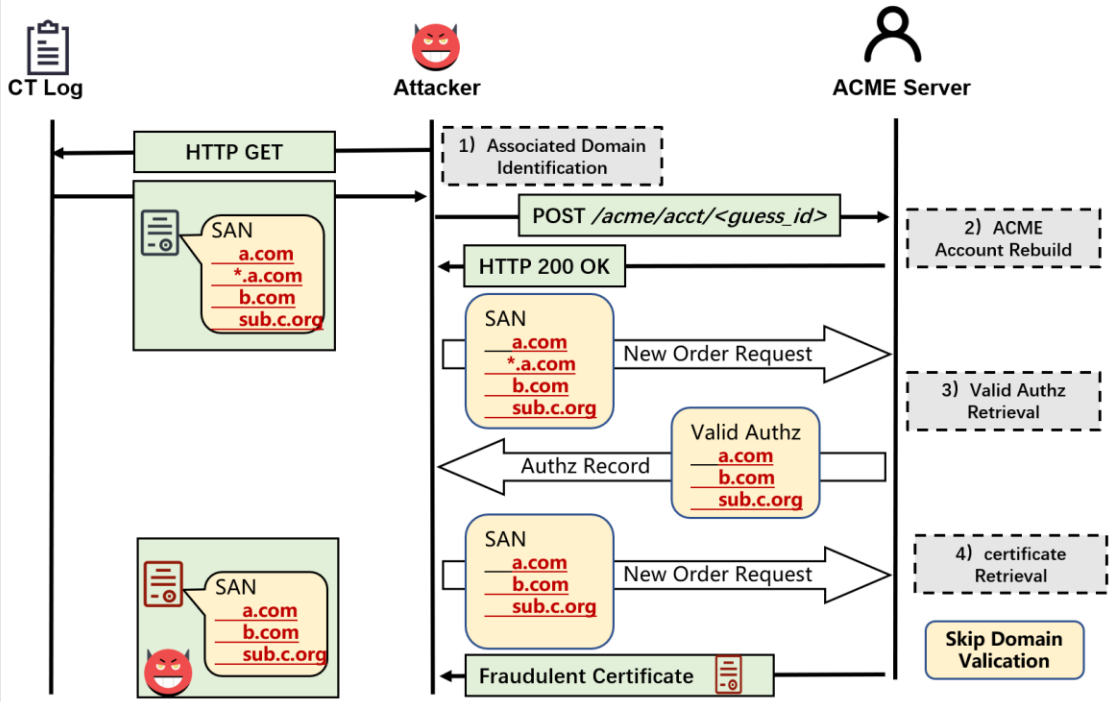
| type | identifiers | rfc/draft | status |
|---|---|---|---|
| sso-01 | email | draft-biggs-acme-sso-01 | draft expired |
| device-attest-01 | permanent-identifier | draft-acme-device-attest-03 | draft active |
| email-reply-00 | email | rfc8823 | standard |
| otp-01/hotp-01/totp-01 | - | draft-ietf-acme-client-09 | draft active |
| cert-01 | - | draft-ietf-acme-client-09 | draft active |
| ppkp-01 | - | draft-ietf-acme-client-09 | draft active |
| tkauth-01 | TNAuthList | rfc9447/9448 | standard |



# Comparison of two categories of attack models

## Existing work: ACME++ Attack path

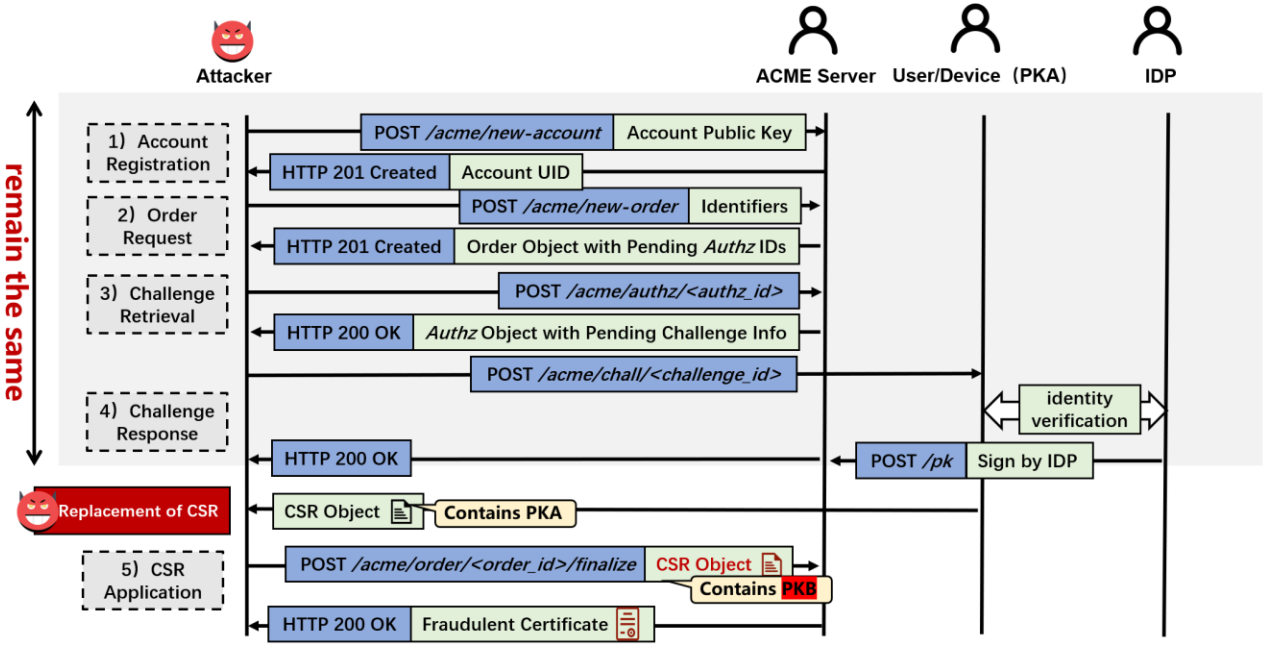
resource category



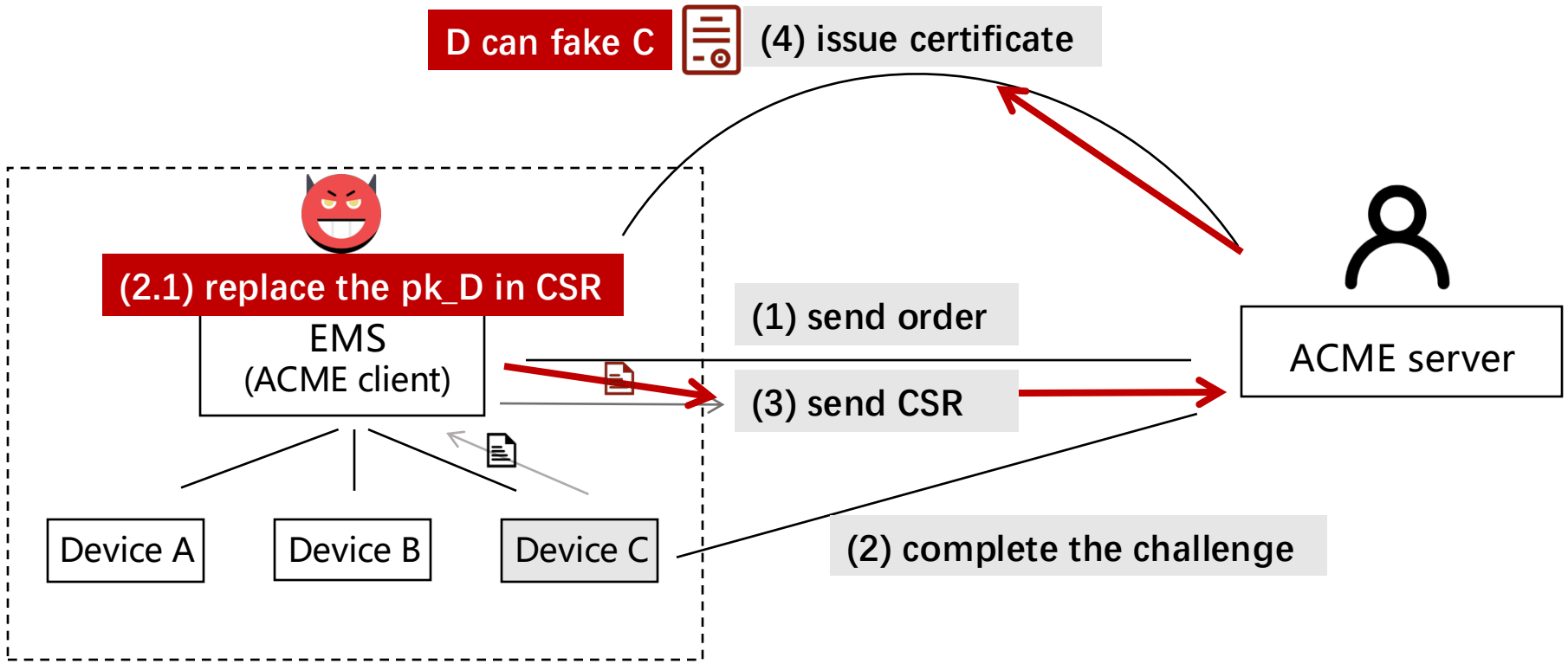

**Ref: ACME++: Secure ACME Client Verification for Web-PKI**

## Our work: Problems with similar attacks

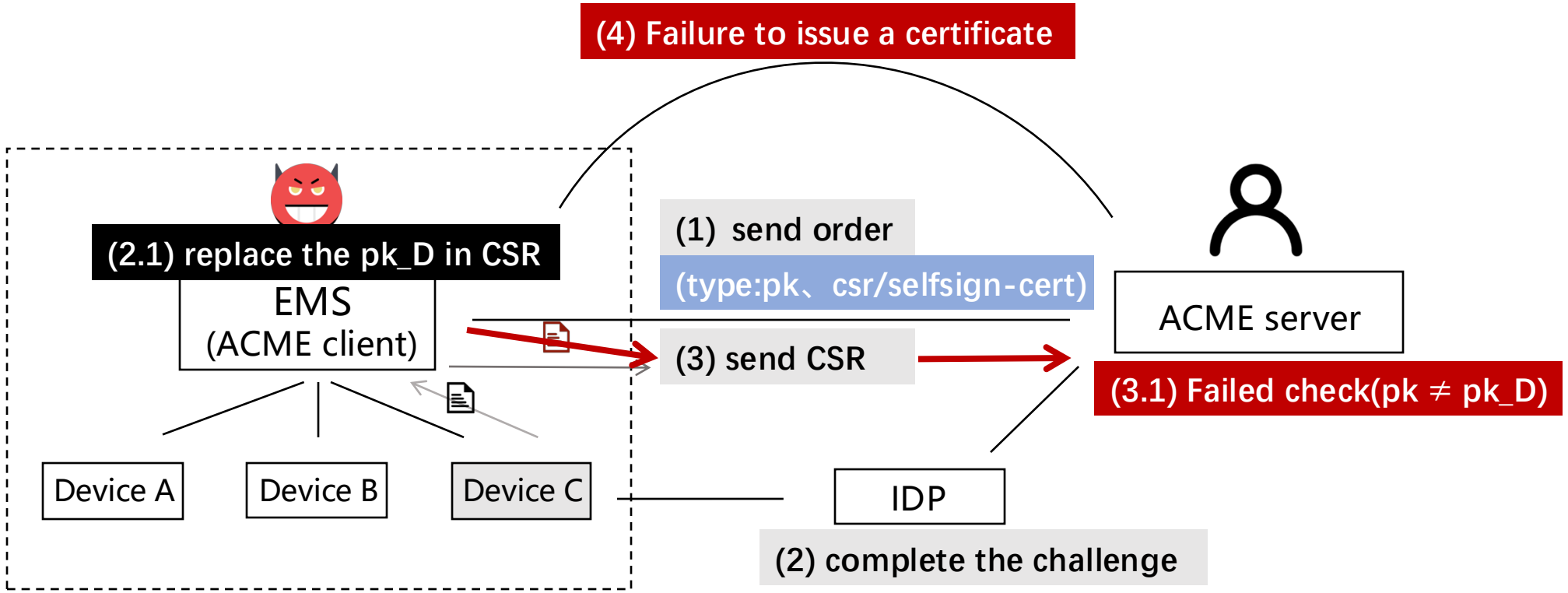
user/device category



# Security Model -- ACME client is deployed on EMS （ PK replacement attack ）



# pk-01--Defend against public key replacement attack



# Solution -- new ACME Extension for PK Challenges : ACME pk Identifier Type

```
"identifier": { "type": "pk", "value":"MIGfMA0GC***GbQIDAQAB"}
"identifier": { "type": "selfsign-cert", "value":"MIIHSDCC***AU1GH3xQ="}
"identifier": { "type": "csr", "value":"MIICljCCA***RL64+taHbP"}
```

**“pk” :**

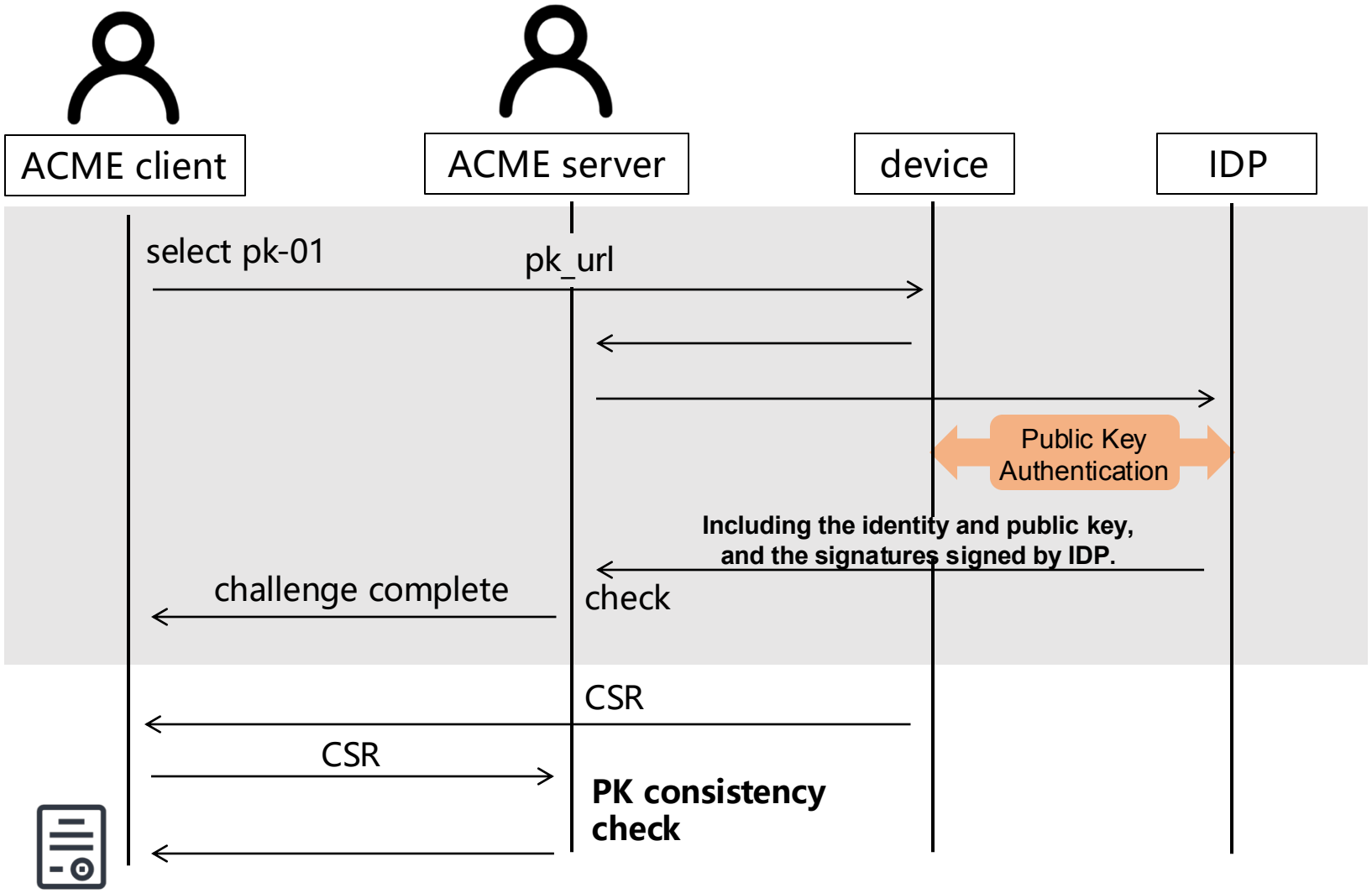
Used to request a certificate for a specific public key.
Example: requesting a certificate for a **device** that is tied to a user's identity.

**“csr” / “selfsign-cert” :**

Used to request certificates for applicants who need to be identified.
I.e., it requires **binding of specific identity information**.

# Solution -- ACME pk-01 Challenge Standard Process

- Account Management
- **Challenge phase (Identifier Validation)**
- **Certificate Issuance**

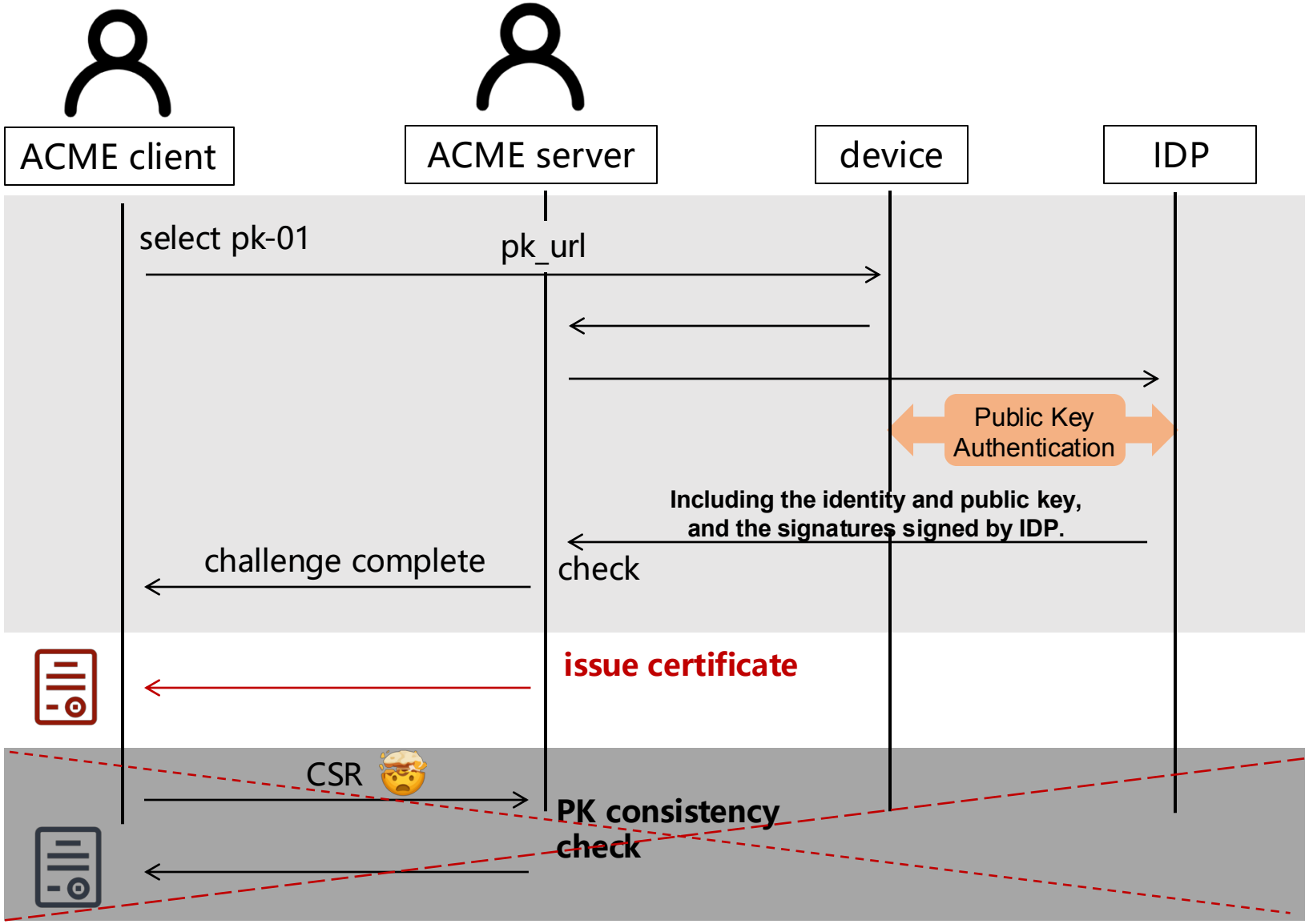


# Solution -- ACME pk-01 Challenge could Remove CSR (Optional)

```
{
  "type": "pk-01",
  "url": https://example.org/acme/chall/abc123_defg456,
  "status": "pending",
  "pk_url": "https://example.org/acme/start-pk",
  "pk_provider": "https://pk-identity-provider.org/",
  "standardization": "standard", "simplified"
}
```

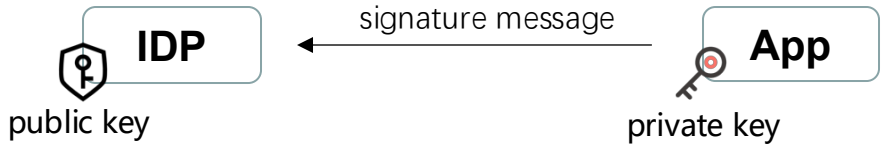
Optional

- Account Management
- **Challenge phase (Identifier Validation)**
- **Certificate Issuance**

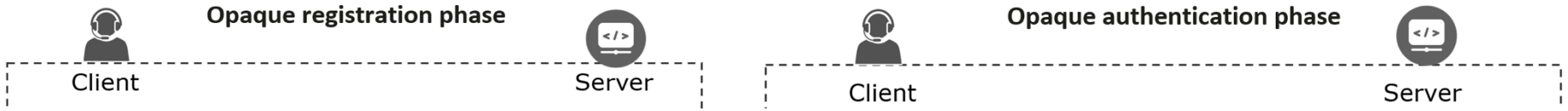


# Various Public Key Authentication Protocols

- challenge public key signature and verify signature (for example, **WebAuthn**)



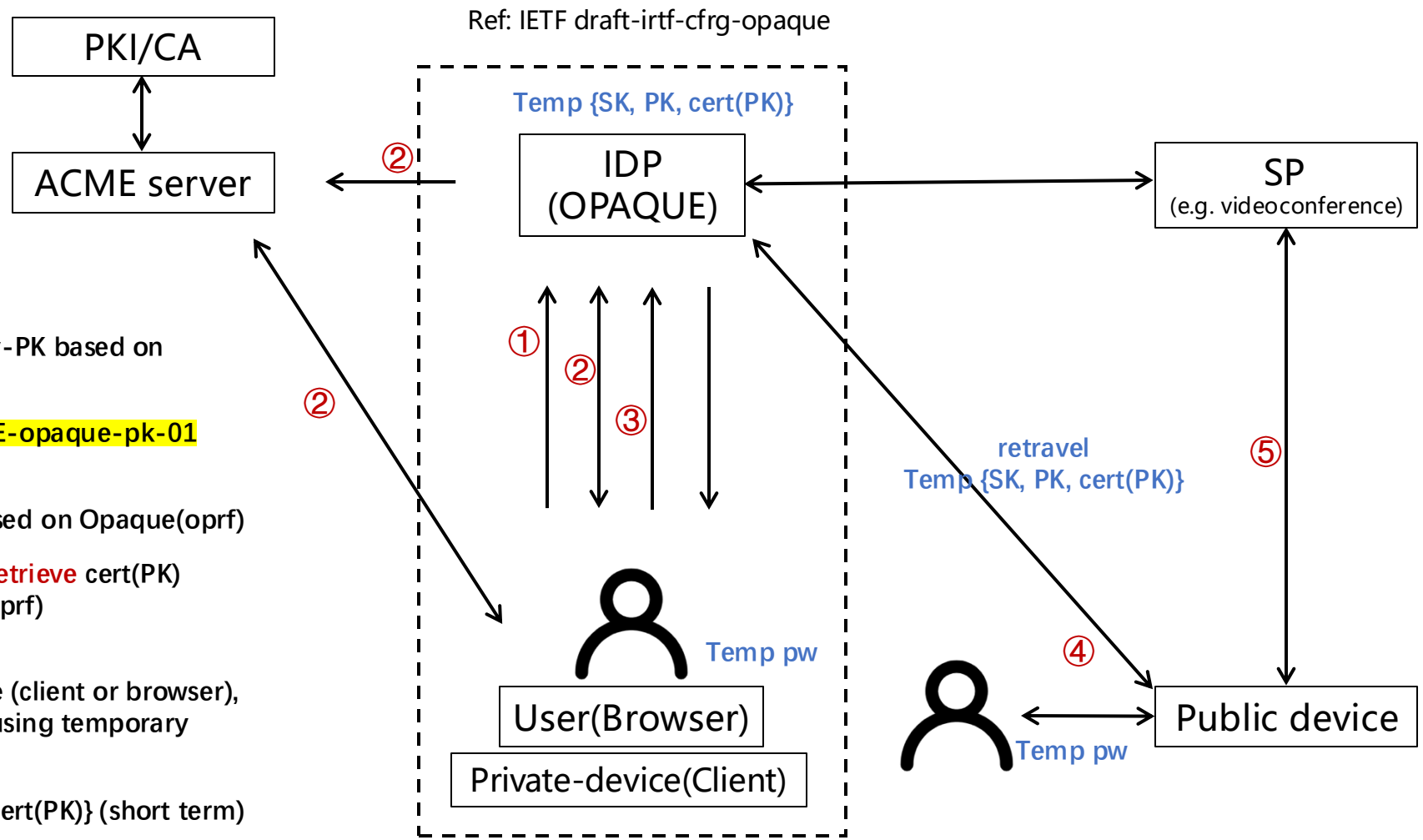

- Opaque/AKE (draft-irtf-cfrg-opaque-18)




- non-interactive zero-knowledge (NIZK) discrete logarithm equality (DLEQ) proof

.....

# ACME + OPAQUE -- Apply short term Certificate (One-time Key) for people to be temporary used in the public device




1. User account register temporary identity-PK based on Opaque(oprf)

2. Issue temporary cert(PK) based on ACME-opaque-pk-01
e.g. for a two-hour e2ee videoconference

3. User e2ee backup temporary cert(PK) based on Opaque(oprf)

4. When conference starts, User login and retrieve cert(PK) stored in public-device based on Opaque(oprf)

5. User access to SP.
e.g. e2ee videoconference via public-device (client or browser), complete two-way authentication with SP using temporary cert(PK).

6. User quit and delete temporary {SK, PK, cert(PK)} (short term)

# Future Work – comments and improvement

Thank Aaron point out:

we should discuss the merits of various challenge types that can be used to verify that the client wants a given key to appear in its certificate and/or that it controls the corresponding private key. I look forward to discussing this further at IETF 122 and into the future.

There is some ideas:

1. **tls-alpn-01** (how to use the existing tls-alpn-01 challenge, with the additional requirement that the presented certificate contain the requested public key and that the TLS handshake complete successfully)
2. **pk-tls-01** (a new method similar to the above, but without the acme-tls/1 ALPN protocol and without the acmeIdentifier extension, allowing the applicant to use their currently-valid TLS certificate if they plan to keep using the same key)
3. **pk-dns-01** (similar to the existing dns-01 challenge, with the change that the server expects to find a hash of the pubkey in the TXT record rather than a key authorization)
4. **pk-dane-01** (similar to the above, but searching for a TLSA 3 1 X or TLSA 1 1 X record rather than a TXT record)
5. **pk-jwk-01** (as mentioned by Richard Barnes upthread, use the private key to sign a JWS over the challenge token, and POST it to the challenge URL like the new onion-csr-01 challenge)

# AOB