Automated Certificate Management Environment Working Group F. Geng
Internet-Draft P. Wu
Intended status: Standards Track L. Xia
Expires: 3 September 2026 Huawei Technologies
2 March 2026

# Automated Certificate Management Environment (ACME) Extension for Public Key Challenges
draft-geng-acme-public-key-04

## Abstract

This document implements automated certificate provisioning through "public key identity challenge + private key ownership verification" by introducing the pk-01 challenge to the ACME protocol [RFC8555]. It serves as a valuable complement to existing external resource verification challenge types such as DNS/HTTP, extending the ACME protocol's applicability beyond Web-PKI to other scenarios. This enables automated certificate issuance for devices and accounts. The core design objective of this document's extension to ACME's pk-01 challenge is to introduce a trusted identity provider (IdP) during the digital certificate application process. This provider verifies the certificate applicant's identity and obtains the corresponding identity public key. It enables the ACME server to use public key identity authentication protocols (e.g., WebAuthn and Opaque [RFC9807]) to verify whether the genuine application behind the ACME client controls the public key. It ensures strong consistency between the public key used during the challenge phase and the public key ultimately used to sign the certificate, preventing tampering with the public key during the CSR submission phase. This enhances the security of the digital certificate issuance process. Similar related work can be found in [RFC9883].

This document also defines an optional process extension that allows removal of the CSR under the pk-01 challenge, enabling the ACME server to issue a certificate directly after successful public key verification.

This document provides an example of practical application at the end, illustrating the integration of the OPAQUE [RFC9807], strong asymmetric password authenticated key exchange (saPAKE) protocol with the pk-01 challenge.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 3 September 2026.

## Copyright Notice



## Table of Contents

## 1. Introduction

The ACME protocol ensures the accuracy and security of automated certificate issuance through various challenge types. Early external resource verification challenges such as DNS/HTTP [RFC8555] have brought much-needed flexibility to Web-PKI, enabling comprehensive automation of web certificates. This document proposes a more general public key identity challenge type for verifying an applicant's (especially an end-entity's or account's) control over certificate ownership. It supports automated certificate issuance for devices and accounts based on public-key identity verification challenges.

The challenge enables the ACME server to verify the control of the identity of the certificate applicant behind the ACME client via the public key authentication protocol (e.g., WebAuthn and Opaque [RFC9807]), truly realizing what the ACME WG Charter describes as "The processing must also confirm that the requesting party has access to the private key that corresponds to the public key that will appear in the certificate". On the other hand, the proposed challenge mitigates public key substitution attacks. In typical deployments, the ACME client serves as the automated interface

between the real applicant and the ACME server. During certificate issuance, while the ACME server verifies the applicant' s control over an identifier through the challenge process, it does not verify that the public key contained in the CSR is actually controlled by the applicant. It is observed that untrusted or compromised ACME clients might replace the public key in the CSR after completing the challenge. Particularly in PKI/CA reseller scenarios, the resellers may submit CSR requests with replaced public key to the ACME server. This results in a mismatch between the applicant's identity and the identity associated with the public key used to issue the certificate. This ultimately allows certificates to be issued to unrelated or malicious entities.

### 1.1. Document Contributions

This document proposes a new ACME challenge type pk-01 that verifies the applicant's control over their identity by means of public key authentication. The public key used in the challenge phase must consistently match the public key in the CSR submission, eliminating the risk of public key substitution attacks. Considering the complicated task of CSR parsing, this document supplements a simplified process of removing CSR (see Section 7), which directly issues a certificate after successful challenge phase, realizing the consistency of the public key in the challenge phase and in the issued certificate.

More critically, the challenge is designed for the ACME server to work in coordination with a trusted identity provider (IdP), which endorses the relationship between the applicant and its public key. This enables ACME server to recognize the real applicant behind through ACME client, and establishes a complete chain of trust from ACME server to applicant. This trust binding mechanism is particularly important in the scenario of issuing certificates for end-entity, ensuring that the certificate clearly identifies the legitimacy of the holder of the identifier. It is precisely this mechanism enabling the ACME server to collaborate with trusted identity providers that supports automated certificate issuance for devices and accounts based on public-key authentication verification challenges. This extends ACME's applicability beyond Web-PKI scenarios.

### 1.2. Some notes

Note that this document requires an IdP that records public keys, achieved by supporting the corresponding public key authentication protocols (e.g., WebAuthn and Opaque). The IdP is the trust center for the automated certificate request process, which stores the trusted user’s public key and possibly the identity information corresponding to the public key. Specific identity information can be selected based on the actual operations. Before the certificate request process, the user’s public key should have been entered into the IdP in some way, such as self-registration by the user or LDAP-based import.

It is worth noting that the existing external account binding (EAB) in ACME [RFC8555] is primarily used to authenticate ACME client account identitie and is not the same as the validation of certificate applicant identities proposed in this document; the two do not conflict. EAB enables the ACME client to authenticate to ACME server using an identifier and key value, which the server can verify against its maintained repository. Through EAB, ACME servers can filter which users can access their services, making it highly useful for commercial certificate authorities. This differs from Let’s Encrypt’s model, which allows anyone to request certificates.

The main purpose of this document is to issue certificates to "applicant who actually pass the challenge", ensuring that the public key of the applicant who completes the challenge and the public key of the certificate issued are consistent. This document strictly corresponds to the concept of “Prohibit the reuse of ACME client public key to request certificate” in the standard ACME process. If the ACME client’s public key is used to apply for a certificate, it is equivalent to an identity impersonation/injection attack by the ACME client on the certificate applicant. The pk-01 challenge presented in this document avoids this very problem by allowing the ACME server to "see" the actual applicant behind the ACME client, enhancing security.

## 2. Terminology

The key words "*MUST*", "*MUST NOT*", "*REQUIRED*", "*SHALL*", "*SHALL NOT*", "*SHOULD*", "*SHOULD NOT*", "*RECOMMENDED*", "*NOT RECOMMENDED*", "*MAY*", and "*OPTIONAL*" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

# 3. Motivation and Threat Model

## 3.1. Motivation

The real world presents typical practical use cases that ACME currently struggles to adequately address. For instance: http-01 requires ports 80/443, yet many devices cannot open them; dns-01 requires permission to modify DNS settings, which is unfeasible for internal networks or IoT devices; tls-alpn-01 [RFC8737] has limited support, and no ACME challenge currently supports “pure public-key identity”. However, extending the pk-01 public key challenge can perfectly address the certificate automation needs in these real-world scenarios, such as:

1. Automated Certification for IoT Devices/Industrial Internet of Things (IoT)

   These devices lack domain names and public IP addresses, cannot open ports 80/443, and cannot modify DNS settings. However, each device possesses a unique identity public key. Through the pk-01 extension, it proves “I am a legitimate device” and “I possess the private key”, device certificate are issued fully automatically with zero configuration.

2. Automated Certification for 5G/6G Edge Nodes, Base Stations, and Small Cells

   These massive edge nodes cannot be manually deployed or configured with DNS, but they come equipped with factory certificates from the device supplier. With the pk-01 extension, they can prove “I am a legitimate device” and “I possess a factory certificate”, enabling certificate automation.

3. Automated Certification for V2X/TSP Platforms

   Automatic certificate renewal for vehicle components (ECU, T-Box, IVI, etc.) cannot rely on DNS/HTTP. Remote vehicle control via the owner’s app requires the verified account to request remote control certificates from the TSP. The pk-01 challenge proves “I am a legitimate component” and “I am a legitimate account”, enabling certificate automation.

4. Automation Workload Certificate in Zero Trust Architecture (ZTA)

The zero-trust architecture explicitly requires public keys as the basis for identity, rather than IP/domain names. The pk-01 challenge can directly issue identity certificates to containers, Pods, microservices, and virtual machines using the workload's public key.

5. Automated Certificate provisioning for Internal Network Devices (Government and Enterprises)

   Achieve automated certificate management for internal network devices through pk-01 challenge based on an internal network IdP combined with a private CA.

6. Anonymous/Pseudonymous Identity Certificates (corresponding to privacy-enhanced PKI)

   With pk-01, only public key identity and private key ownership need to be verified, protecting users' real identities. It can be widely used in scenarios such as private communications, anonymous conferences, pseudonymous identities, and data sovereignty.

## 3.2. Practical Risk Scenarios

The ACME protocol generally assumes by default that the ACME Client represents the real applicant and is capable of correctly submitting a CSR containing the applicant's public key. However, in many deployment models, the ACME client is not operated directly by the end entity. Instead, it may be provided by a reseller, managed security service, IoT platform operator, or other intermediary. In such scenarios, the ACME server verifies control over the identifier during the challenge phase, but does not verify that the public key contained in the CSR is actually controlled by the entity that completed the challenge.

The core issue is the lack of strong cryptographic binding between:

* The entity that completes the challenge, and

* The holder of the private key corresponding to the issued certificate.

The ACME server relies on an assumption of trust in ACME client without enforcing a strong cryptographic binding between the challenging entity and the public key of the final certificate.

#### 3.2.1. Reseller-Based Certificate Issuance

When issuing certificates for resources, the applicant is required to demonstrate resource control (e.g., dns-01, http-01, etc.). The applicant controls the resource and needs to grant ACME client privileges to directly prove its ownership of the resource to complete the challenge, in exchange for ACME server's trust, so as to obtain the certificate.

In reseller-based PKI deployment models, an intermediary may operate the ACME client and interact directly with the CA on behalf of multiple end entities. Such intermediaries often assume responsibilities comparable to those of a Registration Authority (RA), including key management or CSR submission. In these architectures, the CA validates control over an identifier during the ACME challenge phase but does not independently verify that the public key included in the CSR is controlled by the same entity that completed the challenge.

If the intermediary replaces or generates a different public key after successful validation—whether due to compromise, operational error, or malicious intent—the resulting certificate may bind the validated identifier to a key unrelated to the original applicant. This architectural trust assumption introduces a potential public key substitution risk.

#### 3.2.2. IoT and Device Enrollment

In IoT deployment models, certificate issuance is often performed at large scale through automated enrollment systems. An IoT platform or device management service may operate as the ACME client on behalf of thousands or millions of devices. In such architectures, devices typically:

* Possess hardware-bound key pairs (e.g., generated in a Secure Element or TPM), or

* Rely on a provisioning service to request certificates during manufacturing or onboarding.

While ACME identifier validation ensures control over an identifier (e.g., device domain name or device identifier), it does not guarantee that the public key included in the CSR corresponds to the actual hardware device that completed the validation process. If private key ownership is not verified during the challenge phase, the device's identity integrity cannot be enforced.

Device Identity Hardware Private Key Issued Certificate

#### 3.2.3. Cross-Domain Vehicle Control with External IdP

In remote control of vehicles scenarios, it is necessary to apply for a remote control certificate for the device in order to control the automobile. In certain vehicle ecosystems, user authentication is delegated to an external Identity Provider (IdP), such as a charging service provider or mobility platform. In this model: The IdP authenticates the vehicle owner; The Original Equipment Manufacturer (OEM) vehicle cloud relies on the IdP' s identity assertion; An ACME client (operated by the OEM cloud or the service provider) requests a certificate from a CA; The vehicle ultimately trusts certificates issued by the CA.

This architecture spans multiple trust domains:

* Trust Domain A: External IdP and user-facing application

* Trust Domain B: OEM vehicle cloud

* Trust Domain C: CA/ACME server

In such cross-domain scenarios, the CA validates the certificate request through the ACME client but does not directly verify that the authenticated end user possesses the private key corresponding to the public key included in the CSR. If the ACME client or an intermediary component substitutes the public key after identity validation, the issued certificate may bind the authenticated identity to an attacker-controlled key. The vehicle, trusting the CA, may accept remote control commands from the attacker. This risk is amplified in cross-domain deployments because identity assertion, certificate request, and certificate validation occur in different trust domains.

The “pk-01” challenge is therefore necessary to ensure that the authenticated end entity demonstrates proof-of-possession of the certificate key, thereby enforcing strong identitykey binding across trust boundaries.

### 3.3. Core Security Issues

The core issue is the lack of strong cryptographic binding between:

* The entity that completes the challenge, and

* The holder of the private key corresponding to the issued certificate.

While ACME ensures identifier control, it does not inherently guarantee that the validated entity and the certificate subject key are cryptographically linked beyond the trust assumption placed on the ACME client. This gap may lead to identitykey inconsistency and certificate misbinding.

Binding the public key to the challenge is valuable regardless of whether you trust the ACME client.

### 3.4. Threat Model

#### 3.4.1. System Model

The following entities are involved in cross-domain ACME deployment scenarios:

* *End-Entity (EE)*: The ultimate certificate subject that generates a key pair and intends to obtain a certificate (e.g., IoT device, vehicle control application, customer system).

* *Intermediary / ACME Client*: An entity that interacts directly with the ACME server on behalf of the End-Entity.

* *ACME Server / CA (AS)*: The certificate authority that validates challenges and issues certificates.

* *Identity Provider (IdP)* (optional): Provides identity assertions to the intermediary.

The ACME Server trusts the ACME Client to complete the challenge procedure, while the End-Entity relies on the intermediary to request a certificate corresponding to its key material. These entities may reside in different trust domains. The ACME Server/CA (AS) needs to establish trust with the identity provider.

#### 3.4.2. Trust Assumptions

* The ACME Server fully trusts the ACME Client to correctly represent the End-Entity.

* The End-Entity does not necessarily trust the ACME Client with its private key.

* The ACME Server cannot directly observe the End-Entity behind the ACME Client.

* Identity assertions (if any) are not cryptographically bound to the certificate public key by default.

### 3.4.3. Attack Capabilities and Core Threats

The adversary is capable of:

* Modifying or replacing the CSR public key.

* Submitting a substituted key for certificate issuance.

* Rebinding identity assertions to attacker-controlled key material.

The core threat lies in the fact that the Challenge executor ≠ CSR private key holder, and the CA cannot prove their identity matches.

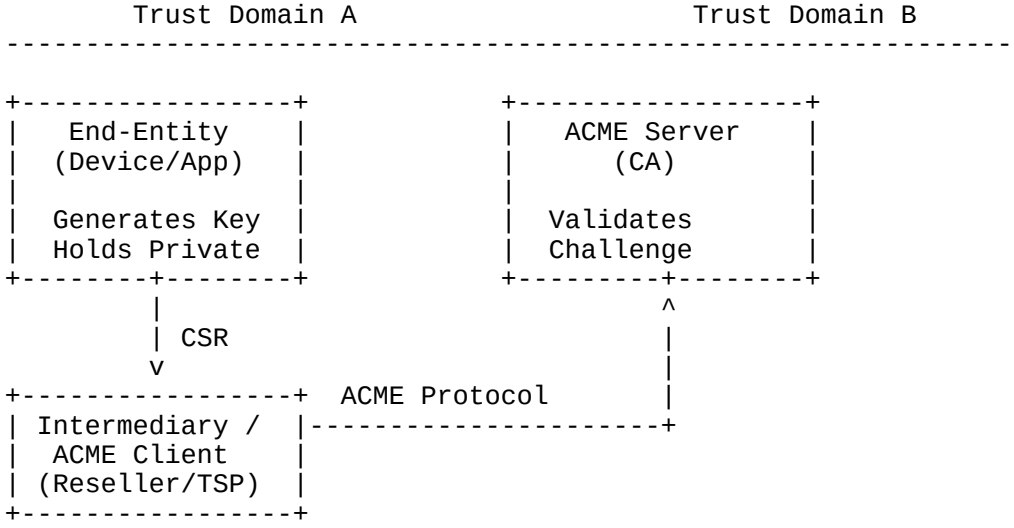


Figure 1: Threat Model

## 4. Extensions - Identifier Types

This document introduces an extension that enables validation of end-entity ownership of the public key included in the certificate request. It ensures strong consistency between the identity corresponding to the final certificate issued and the identity of the certificate applicant. And the public key authentication protocol is utilized to provide strong security for the authentication process.

This extension does not replace existing identifier types. Instead, it augments the ACME validation model by enabling cryptographic binding between:

* The identifier being validated, and

* The public key to be certified.

This document defines a new identifier type:

```
"identifier": {
     "type": "pk",
     "value":"MIGfMA0GC***GbQIDAQAB"
}
```

The “pk” identifier represents a public key to be certified. The ACME server MUST ensure that the public key included in the final certificate matches the value of the “pk” identifier. This identifier type is intended for lightweight key-binding scenarios, such as device enrollment or user-device association, where the public key itself represents the certified subject.

### 4.1. Binding Model

In certain deployments, stronger binding between identity information and key material may be required. This document defines optional binding modes for the pk identifier:

#### 4.1.1. Self-Signed Certificate Binding Mode

The client MAY provide a self-signed certificate containing the public key and subject information. If used, the ACME server MUST verify that:

* The public key matches the “pk” identifier.

* The subject information in the final certificate matches the self-signed certificate.

The self-signed certificate serves solely as a binding container and does not imply trust in its issuer.

#### 4.1.2. CSR Binding Mode

The client MAY include a base64url-encoded CSR in the order object. If present, the ACME server MUST ensure that:

* The public key in the final certificate matches the CSR.

* The subject and extensions in the final certificate are consistent with the CSR.

When CSR binding mode is used, the client is not required to resubmit the CSR during the finalize step. This mechanism prevents modification of identity information between order creation and certificate issuance.

```
{
  "identifiers": [
     { "type": "pk",
       "value": "MIGfMA0GC..."
       "bindingMode": { \n
            "mode": "none" | "csr" | "selfsigned",
            "value": "<base64 data if applicable>"
       }
     }
   ],
}
```

## 5. Extensions -- pk-01 Challenge Types

### 5.1. Challenge Definition

This specification defines a new ACME challenge type pk-01. pk-01 challenge is used to prove an applicant’s control over the private key corresponding to a given public key. Unlike traditional identifiers (e.g., DNS, IP), the PK identifier itself represents a public key value. Therefore, the goal of the pk-01 challenge is:

To prove that the entity initiating the challenge actually holds the private key matching the public key specified in the identifier.

### 5.2. Challenge Object

The pk-01 challenge type requires the client to access the specified “pk-url” to start the challenge and complete the verification of the corresponding private key control of the declared public key. A challenge of this type MUST include all required fields described in section 8 of [RFC8555]. In addition, the following fields are defined for this specific type of challenge:

* pk_url (required, string): The url to start the pk-01 challenge type process. The server must include enough information in the URL to allow the request to be associated with a specific challenge and to be able to point to a specific PK provider or public key server.

* pk_provider (optional, string): The domain of the PK provider relied upon for this challenge. ACME server MAY rely upon any number of PK providers and public key servers, however each MUST

be represented as a different entry in the challenge array. The applicant can use this field to differentiate the list of providers and select the most appropriate one. If this field does not exist, the ACME server's default identity provider is used. The server MUST NOT present more than one pk-01 challenge with the same pk_provider value in a single authorization, including values for unprovided fields.

* standardization (optional, string): Indicate options for the ACME process. The Acme server provides a choice between the standard ACME protocol flow (standard) or a removed CSR file (simplified). If this field is not exist, the ACME standard process is executed by default.

The server MUST sets the status of the challenge to processing after receiving a response from the client within the validity period. If the client completes the proof of ownership of the private key corresponding to public key and the generated identity assertion validates the declared identifier, then the status of the challenge is set to "valid". If the server fails to validate the public key against the private key control or fails to validate the declared identifier, the status of the challenge will be set to "invalid".

```
{
      "type": "pk-01",
      "url": "https://example.org/acme/chall/abc123_defg456",
      "status": "pending",
      "token": "LoqXcYV8...uZg",
      "pk_url": "https://example.org/acme/start-pk",
      "pk_provider": "https://pk-identity-provider.org/",
      "standardization": "standard" | "simplified"
}
```

# 6. Identifier Validation Challenges

## 6.1. Protocol Overview

When the identifier type is "pk", the ACME server MUST validate proof-of-possession of the corresponding private key using the pk-01 challenge. The bindingMode parameter associated with the identifier defines additional validation requirements that apply during order finalization, but it does not alter the semantics of the pk-01 challenge.

The validation process consists of two logically distinct phases:

1. Challenge Validation Phase: Proves possession of the private key corresponding to the identifier public key.

2. Order Finalization Phase: Enforces consistency constraints defined by the selected bindingMode.

The ACME server MUST ensure that the public key used to complete the challenge is identical to the public key bound to the issued certificate.

## 6.2. Protocol Details

The general process of the pk-01 challenge is illustrated by the standard ACME certificate issuance sequence. For convenience, it is divided into three phases: certificate application, public key authentication and certificate issuance phase.

In the first phase, the client submits a newOrder request containing one or more identifiers, including public key information for the pk identifier. The ACME server responds with a list of authorizations, each containing one or more available challenges for proving control of the identifier. The client selects a challenge it can complete and performs the required proof-of-possession operation.

In the second phase, the client selects a challenge from the list of available challenges provided by the ACME server, which may be of the pk-01 type. Selecting such a challenge triggers a public key authentication protocol between the applicant and their corresponding IdP. The ACME client and IdP perform the authentication protocol (e.g., aPAKE or Opaque [RFC9807]) to prove possession of the private key corresponding to the identifier public key. The IdP may optionally provide a signed assertion of the public key identity if the ACME server maintains a whitelist of trusted public keys. Upon successful validation, the ACME server marks the associated challenge as valid. The applicant's private key is never disclosed to any intermediate agent during this process.

After completing the challenge, the client proceeds to finalize the order by submitting the CSR (if not submitted earlier). The ACME server MUST verify that the public key in the CSR matches the public key in the identifier that was validated during the challenge phase. Furthermore, ACME server MUST verify whether the bindingMode value matches that specified in the CSR.

This ensures that any tampering with the public key is detected. Other aspects of order finalization and certificate issuance follow the standard ACME process.

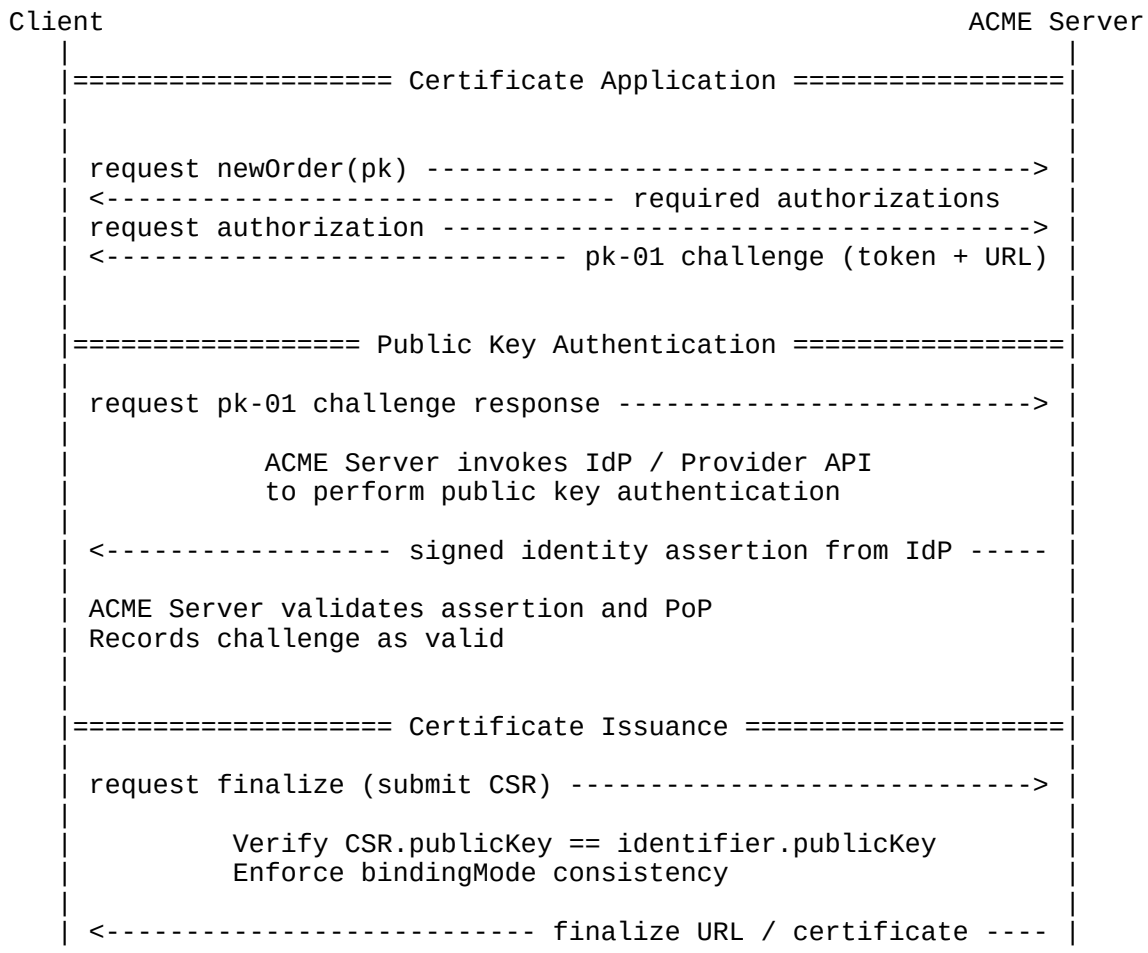


Figure 2: Overview of the pk-01 Challenge Flow

### 6.3. Public key authentication & Order fulfillment

Public key authentication is essentially authenticating the control of the corresponding private key of a public key and pk_url allows the client to initiate the public key authentication process. The server must accept requests for pk_url.

1. The ACME server receives the request and redirects it to the IdP. IdP instance holds the public key, e.g. IdP instances supporting the aPAKE/Opaque protocols.

2. The IdP requires the requesting party to perform authentication to verify that it holds the private key corresponding to the public key. The IdP will include supported public key verification protocols in the verification request, protocols that include, but are not limited to (1) challenge public key signature and verify signature (WebAuthn), (2) saPAKE (Opaque [RFC9807]) and (3) zero-knowledge proof authentication (DLEQ or zk-SNARKs), etc. The client selects one of the protocols to perform the authentication process.

3. After successfully authenticating the identity, the IdP returns the user's information and the logged-in device public key information to the ACME server. When the Acme server receives the request, it checks whether the device public key is consistent with the public key in the order. When the ACME server receives the request, it MUST check whether the device public key is consistent with the public key in the order. For csr and selfsign-cert bindingMode types, identity consistency checks are also required. The challenge is successful if the check passes.

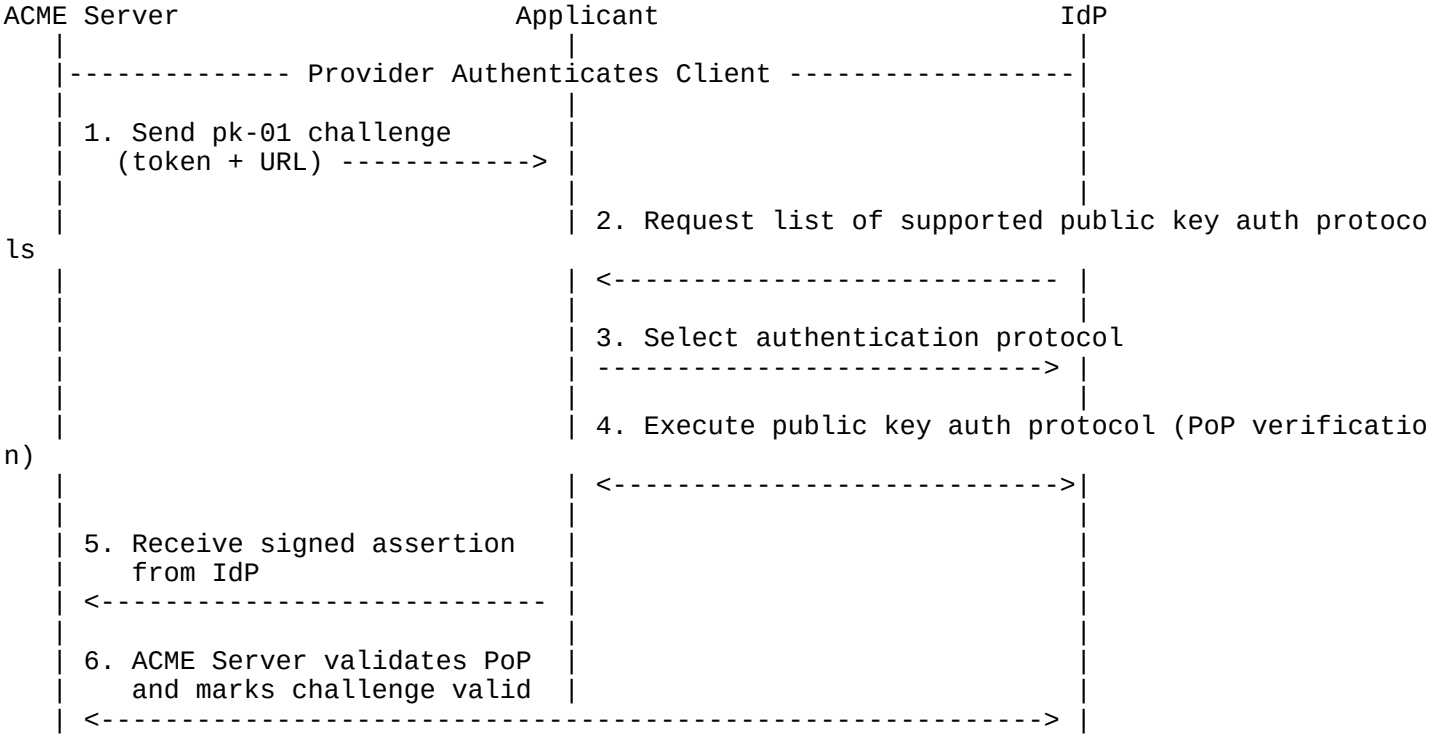


Figure 3: Overview of the PK Authentication Flow

When the ACME server receives a response from IdP, it must validate it and update the appropriate challenge status. The server updates the challenge status to "valid" if the provider validation is successful, or to "invalid" if it is unsuccessful. The server needs to validate the order against the identifier after receiving a request from the client to complete the order. The client's request for order fulfillment can only be continued under the condition that all checks have been passed.

## 7. Certificate Issuance without CSR

Since the ACME server has already obtained and validated the applicant' s public key during the pk-01 challenge phase, submitting a CSR in the finalization phase is optional. In cases where the CSR does not contain any additional information beyond what is already established in the ACME protocol (e.g., identifier and public key), the final phase can proceed without transmitting a separate CSR file.

The process from the client submitting a new order to the ACME server validating the challenge remains consistent with Section 6. During the final certificate issuance phase, the ACME server updates the challenge status to valid (or invalid if verification fails) and automatically issues a certificate bound to the verified public key. There is no need to submit the CSR again.

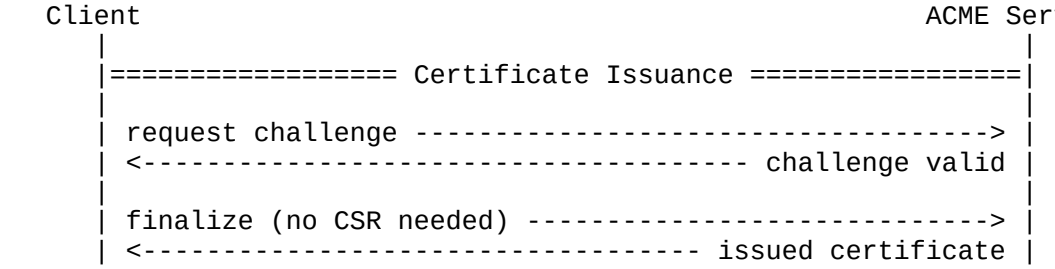


Figure 4: Overview of the pk-01 Challenge Flow (remove CSR)

## 8. Integration with OPAQUE Protocol

### 8.1. Overview

This section specifies the integration of the OPAQUE asymmetric Password-Authenticated Key Exchange (aPAKE) protocol with the pk-01 ACME challenge.

The synergy between these two protocols addresses the "Identity-to-Key" binding problem in Zero Trust environments. While OPAQUE provides a cryptographically strong, password-based mutual

authentication where the server never sees the plaintext password, the pk-01 challenge provides a mechanism for an ACME server to verify that a Public Key (PK) is indeed owned by the authenticated entity. By combining them, an IdP can issue an endorsement for a transient public key based on a successful OPAQUE exchange.

## 8.2. OPAQUE-based Public Key Authentication Flow

The following diagram illustrates the interaction between the ACME Client, the ACME Server, and the OPAQUE-enabled IdP.

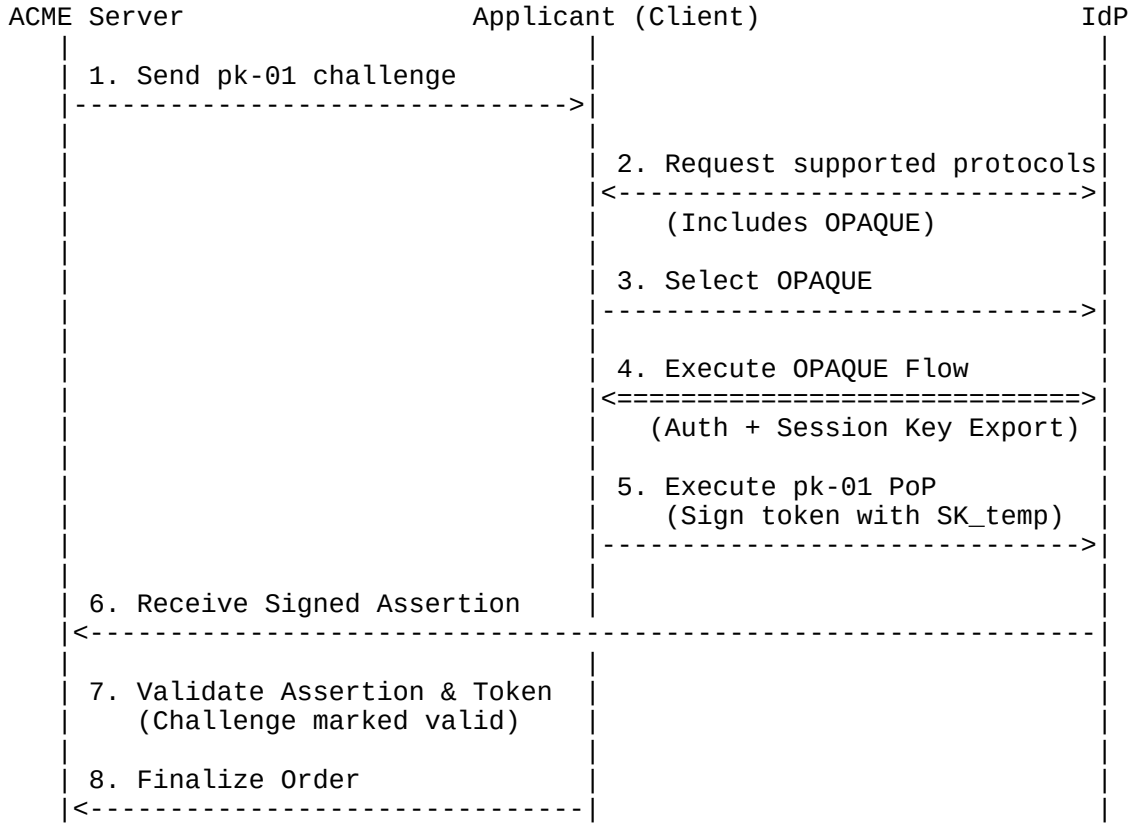


Figure 5: OPAQUE-based Public Key Authentication Flow

## 8.3. Step-by-Step Integration Details

1) Challenge Initiation: The ACME Server provides a token which MUST be unique and a url pointing to the OPAQUE-enabled IdP. 1) Protocol Selection and OPAQUE Execution: Upon contacting the IdP, the Applicant selects OPAQUE from the supported list.

* The OPAQUE exchange provides Mutual Authentication: The Applicant verifies the IdP, and the IdP verifies the Applicant via their password.

* Key Export: Both parties derive a shared secret Kexport. This key is used to secure the subsequent PoP verification.

3) PoP Verification (The Key Binding): The Applicant generates a transient key pair (PKtemp, SKtemp) and proves possession to the IdP.

* The Applicant signs the ACME token using SKtemp.

* The entire request to the IdP is authenticated using the OPAQUE session key (Kexport), binding the identity to the specific public key and ACME session.

4) Assertion Submission: The IdP generates a Signed Assertion containing:

* The validated Public Key (PKtemp).

* The ACME token.

* The Applicant’s identity metadata (e.g., alice@company-a.com).

The IdP sends this directly to the ACME Server (or via the Applicant). The ACME Server verifies the IdP’s signature and the token freshness.

### 8.4. Implementation Scenarios: Cross-Domain Trust

In a managed guest access scenario (e.g., Alice from Company A joining Company B’s session), cross-domain temporary certificate requests can be achieved by combining pk-01 and OPAQUE.

* Out-of-band Provisioning: An administrator at Company B pre-registers Alice in Company B’s OPAQUE IdP and securely transmits a temporary password to her.

* Challenge Redirection: When Alice initiates the ACME process with Company B’s Server, she is directed to Company B’s IdP.

* Local Trust: The ACME Server is configured to trust its local OPAQUE IdP. The IdP acts as the "Validator" of the pk-01 challenge.

* Credential Lifecycle: Since the password was pre-registered for a specific meeting, the IdP can automatically expire the OPAQUE record and the associated pk-01 validation status once the session concludes.

# 9. Further Use Cases

## 9.1. Various Public Key Authentication Protocols

The certificate applicant can pick a suitable public key authentication protocol according to the specific usage scenario. It can be WebAuthn authentication, OPAQUE, private key signature checking, non-interactive zero-knowledge (NIZK) discrete logarithm equality (DLEQ) proof, and so on.

## 9.2. Revocation of Certificates

When a certificate is revoked, it is also necessary to prove whether the user is authorized to revoke it. Thus, the PK-01 challenge proposed in this document can also be used to prove that the user applying for revocation does have the ownership of the corresponding private key of the certificate, so as to realize a more reliable revocation.

# 10. Security Considerations

The security of the pk-01 challenge depends on the integrity of the tripartite trust model between the ACME Server, the Applicant, and the Identity Provider (IdP).

## 10.1. Trust in the Identity Provider (IdP)

In this model, the ACME Server delegates the verification of public key possession to an external IdP. Therefore, the ACME Server MUST establish a strong trust relationship with the IdP. If an IdP is compromised, an attacker could obtain fraudulent certificates by generating malicious assertions for arbitrary public keys. ACME Servers SHOULD limit the scope of certificates (e.g., specific identifiers or short validity periods) issued based on assertions from any single IdP.

## 10.2. Integrity of the Assertion

The assertion issued by the IdP acts as the proof of validation. It MUST be protected against tampering and replay attacks.

* Cryptographic Binding: The assertion MUST be digitally signed by the IdP using a key verifiable by the ACME Server.

* Freshness: The assertion MUST contain a timestamp and the ACME token to ensure it is bound to a specific, current challenge and cannot be re-used in a different context.

* Public Key Pinning: The IdP MUST include the exact subject public key (PK) in the assertion to prevent "Public Key Substitution" attacks, where an attacker intercepts a valid identity proof but replaces the public key with their own.

### 10.3. Credential Security and OPAQUE

When OPAQUE is used as the authentication protocol:

* Password Hardening: OPAQUE provides protection against offline dictionary attacks even if the IdP's database is breached. However, the initial provisioning of temporary passwords (e.g., in guest access scenarios) remains a point of vulnerability. Such passwords SHOULD be transmitted via out-of-band, encrypted channels.

* Session Binding: The IdP SHOULD ensure that the public key PoP (Proof of Possession) is performed within the secure session established by the OPAQUE handshake to prevent session hijacking.

### 10.4. Privacy Considerations

The IdP SHOULD only share the minimum necessary identity metadata with the ACME Server. While the ACME Server needs to verify that the applicant is authorized, it may not require the user' s full profile. Implementing parties SHOULD adhere to the principle of least privilege regarding data disclosure in assertions.

## 11. IANA Considerations

### 11.1. Identifier Types

The "ACME Identifier Types" registry is to be updated to include the following entries:

| Label | Reference |
|---|---|
| pk | RFC XXXX |

### 11.2. ACME Validation Method

The "ACME Validation Methods" registry is to be updated to include the following entries:

| Label | Identifier Type | ACME | Reference |
|---|---|---|---|
| pk-01 | pk | Y | RFC XXXX |

## 12. Normative References

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

[RFC9807] Bourdrez, D., Krawczyk, H., Lewi, K., and C. A. Wood, "The OPAQUE Augmented Password-Authenticated Key Exchange (aPAKE) Protocol", RFC 9807, DOI 10.17487/RFC9807, July 2025, <https://www.rfc-editor.org/info/rfc9807>.

[RFC9883] Housley, R., "An Attribute for Statement of Possession of a Private Key", RFC 9883, DOI 10.17487/RFC9883, October 2025, <https://www.rfc-editor.org/info/rfc9883>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8737] Shoemaker, R.B., "Automated Certificate Management Environment (ACME) TLS Application-Layer Protocol Negotiation (ALPN) Challenge Extension", RFC 8737, DOI 10.17487/RFC8737, February 2020, <https://www.rfc-editor.org/info/rfc8737>.

## Authors' Addresses

Feng Geng
Huawei Technologies
Email: gengfeng@huawei.com

Panyu Wu
Huawei Technologies
Email: wupanyu3@huawei.com

Liang Xia
Huawei Technologies
Email: frank.xialiang@huawei.com