Automated Certificate Management Environment Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

F. Geng
P. Wu
L. Xia
Huawei Technologies

7 July 2025

Automated Certificate Management Environment (ACME) Extension for Public Key Challenges

draft-geng-acme-public-key-02

Abstract

This document specifies an extension to the ACME protocol [RFC8555] that enables Acme server to use the public key authentication protocol to verify that the real certificate applicant behind the Acme client has control over the identity and to ensure strong consistency between the identity in the challenge phase and the identity of the final certificate issued. This document also proposes a process extension for removing CSR at the certificate application phase.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 January 2026.

Copyright Notice

Table of Contents

1. Introduction

When ACME was first introduced, its main focus was on the challenge of verifying control of identifiers such as domain names. ACME later expanded to support verification of user/device ownership and issuance of the certificate for user/device. Based on the sso-01 [I-D.biggs-acme-sso] for verifying email identifiers, this document proposes a more general ACME challenge type oriented towards verifying user/device ownership. The challenge enables the Acme server to verify the control of the identity of the certificate applicant behind the Acme client via a public key authentication protocol, 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 challenge avoids public key replacement attacks.

Typically, the Acme client acts as a unified entry point for automated certificate requests and communicates with the Acme server. In the process of requesting a certificate for the user/device, the Acme server does not verify whether the public key in the Certificate Signing Request (CSR) is owned by the applicant, although it can verify the applicant's control over an identifier through the challenge phase. This mechanism has the risk of public key replacement, i.e., an untrustworthy or compromised Acme client may replace the public key in the CSR after completing the challenge, so that the applicant who completes the challenge is not the same as the user corresponding to the public key of the issued certificate. This ultimately allows certificates to be issued to unrelated or malicious entities.

This document proposes a new ACME challenge type pk-01 that verifies the identity of the applicant by means of public key authentication and is directly bound to the authorization process. The Acme server requires that the public key in the challenge phase be consistent with the public key in the final CSR submission, eliminating the risk of public key replacement 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. 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 user/device, ensuring that the certificate clearly identifies the legitimacy of the holder of the identifier.

Note that this document requires an IdP that records public keys. 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 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 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. Security Considerations

Problem Statement

The standard ACME automated certificate application process can be summarized in three phases: ACME account management, challenge phase and certificate application phase. The current decoupled design of the three-part framework brings many benefits, but there are also security issues. The public key of the applicant who completes the challenge in phase 2 and the public key in the issued certificate in phase 3 is not consistently bound. Problems can occur where the users in the challenge phase are not the same as the users who actually receive certificates.

The content of ACME application certificates can be categorized into resources and user/device. The resource category is mainly server-side resources (e.g., domain names), such as dns-01[RFC8555], http-01[RFC8555][RFC8738], tls-alpn-01[RFC8737][RFC8738] and so on. The user/device category is mainly such as sso-01 [I-D.biggs-acme-sso], device-attest-01 [I-D.acme-device-attest], and e-mail-reply-00 [RFC8823] and so on. Public key replacement attacks are more likely to occur in scenarios where certificates are issued for the user/device.

3.1. Issuing certificates for resource server

In that case, the applicant is required to demonstrate resource control (e.g., dns-01, http-01, etc.). This is a typical use case for the initial focused application scenario of Web PKI domain name certificates presented in the Acme WG Charter. In this use case, the applicant controls the resource and needs to grant Acme client privileges to directly prove its ownership of the resource to complete the interrogation, in exchange for Acme server's trust, so as to obtain the certificate. Thus both Acme server and applicant need to trust Acme client.

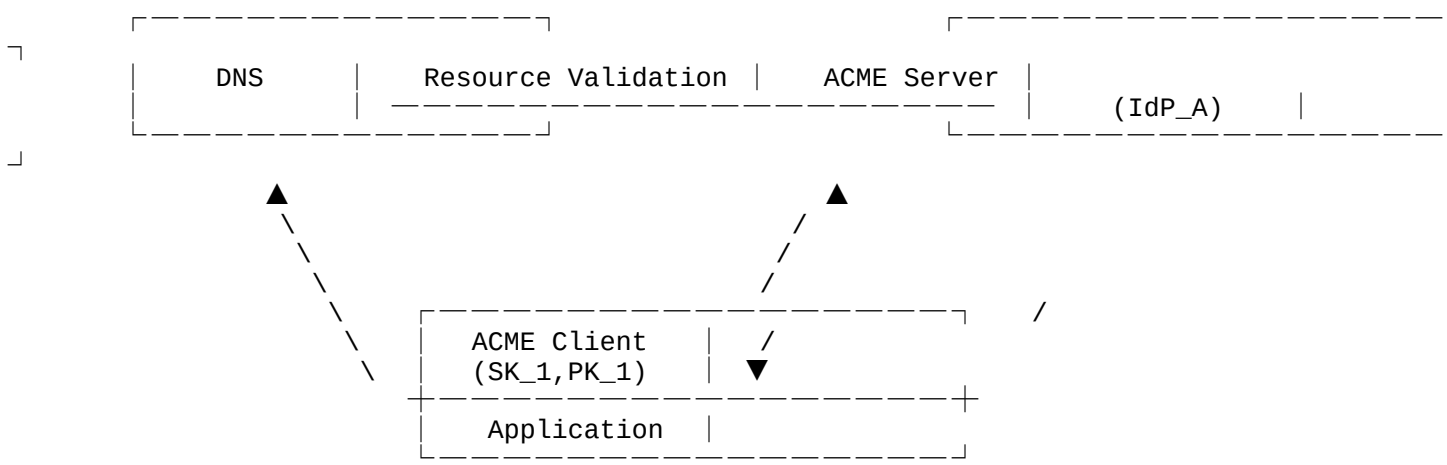

Figure 1: Issuing certificates for resource server

3.1.1. Issuing certificates for clients with existing digital identities

In this case a third party IdP_B exists to provide identity assertion for the applicant. This is commonly seen in situations where certificates are issued for user/device within an organization, where the user/device already hold their own public-private key pairs.

Applicant has registered the identity public key PK_2 under IdP_B to get the provided identity assertion of IdP_B. Acme server trusts Acme client, but applicant can't fully trust Acme client and holds private key privileges by itself without granting them to other. As a result, when the applicant completes the challenge, it cannot trust whether the Acme client will apply for certificates for other malicious users with the help of its own challenge success record, i.e., the Acme client may be attacked so as to replace the public key information in the CSR to apply for certificates to the Acme server. This causes an adversary to impersonate a legitimate user/device. And Acme server can't "see" the applicant behind the Acme client.

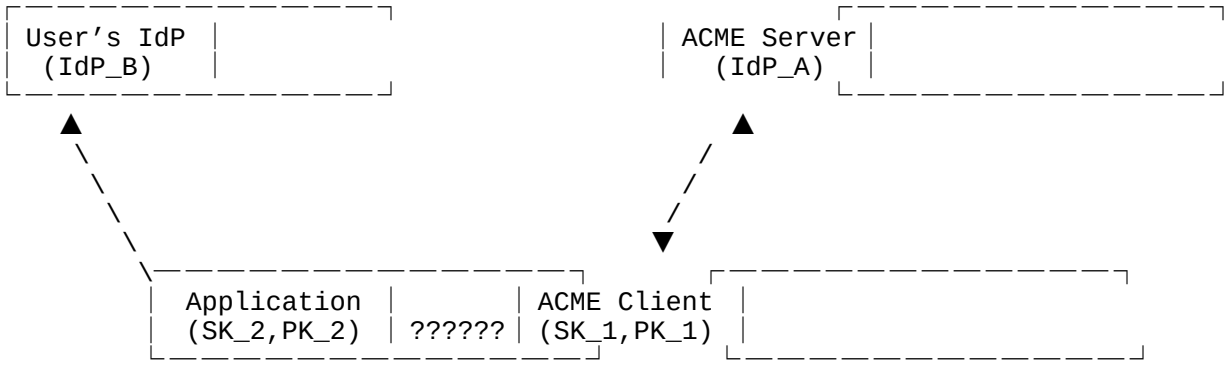

Figure 2: Issuing certificates to clients with existing digital identities

3.2. Defend against Threats

This document proposes the pk-01 challenge to construct a complete chain of trust from the Acme server to the applicant, so that certificates can be issued to the applicants (users/devices) that actually pass the challenge. Acme server indirectly verifies the authenticity of PK_2 behind the Acme client proxy and who holds control of SK_2 by trusting IdP_B. A complete chain of trust from the Acme server to the applicant is constructed so that the Acme server verifies the control of the identity by the applicant behind the Acme client through a public key authentication protocol.

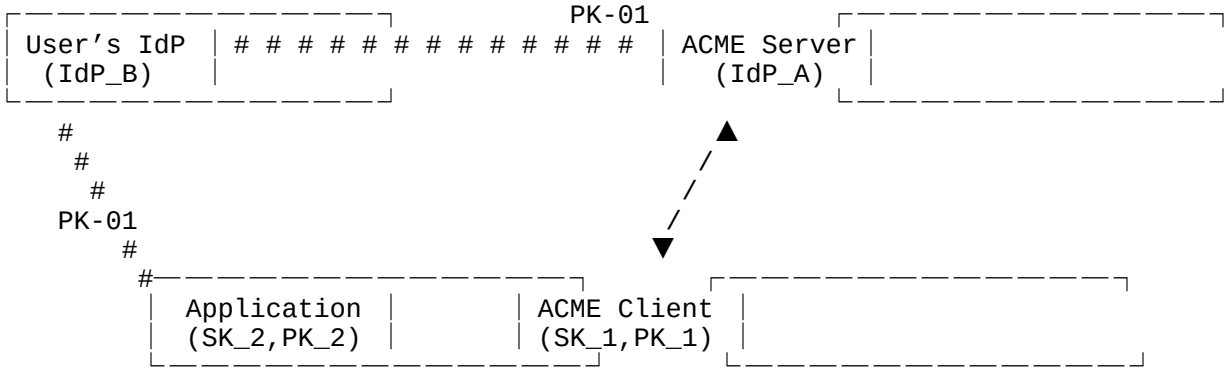

Figure 3: Trust chain from Acme server to applicant

Under the security model of Section 3.1.2, because of the presence of User's IdP, instead of granting the Acme client privileges and trust equivalent to resource control as it did in Section 3.1.1, the

applicant only needs to give the Acme client the minimum required trust to require the Acme client to fulfill its proxy responsibility for requesting certificates. PK-01 provides end-to-end security when crossing trust domains from Acme server to applicant (user/device), preventing attacks from both external and internal sources, constructing a stronger generalized security model. It is also applicable and relevant for authentication scenario use cases between cross-trust domains.

The pk-01 challenge presented in this document prevents public key replacement attacks on the other hand (especially in 8.2 scenarios). With the proposed new identifiers, the consistency of the PK in the challenge phase and the pk in the final issued certificate is guaranteed against public key replacement attacks.

4. Extensions -- Identifier Types

New identifier types are proposed in this document to ensure 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.

The identifier pk can be used in lightweight binding cases, such as user-bound devices, where the Acme server needs to verify the consistency of the pk identifier in the final certificate issuance phase and in the challenge phase. The identifiers csr and selfsign-cert can be used in cases where specific identity information is bound (including pk, subject, etc.). The Acme server needs to verify the consistency of this identifier value during the final certificate issuance phase to prevent the identity information from being tampered with midway. Especially when csr is specified as the identifier, the applicant can send the csr when submitting the certificate order without submitting the csr file again in the final certificate application phase, which corresponds to the simplified process of revocation of csr in Section 7. Implementations MUST NOT use the challenge of type pk-01 as options to validate a type identifier other than the following.

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

5. Extensions -- pk-01 Challenge Types

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.

process options (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", "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

The general process of the PK 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 the certificate request, which carries the public key information at the start. The server responds to the client that it must satisfy authentication. The client can select one of the various authentication methods it supports and inform the server. The server returns what needs to be accomplished for this authentication method. This content will contain the "start URL".

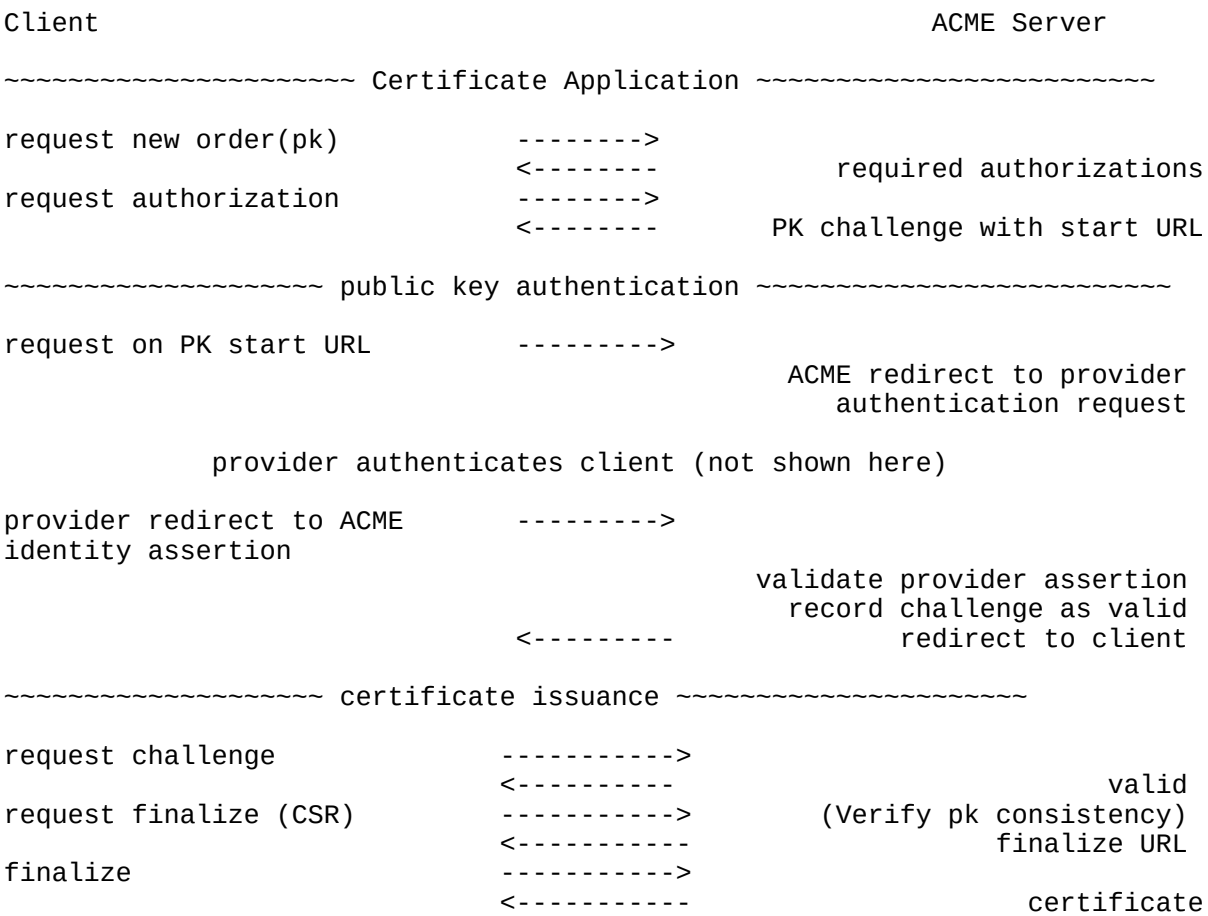

Figure 4: Overview of the pk-01 Challenge Flow

In the second phase, the client performs a request for one of the authentication methods and the server responds to it. Among these may be one or more challenge of type pk-01, which can be used to start the process of authentication based on the web through different "start URL". The request redirects the client to the applicant and the IdP, which conducts the public key authentication protocol with the applicant. After successfully providing validation to the IdP, the Acme server records the associated challenge as validated (or not). Among them, the whitelisting mechanism can be supplemented here. Legitimate public key identities will be registered in advance at the Acme server to form a trusted list, and after the public key authentication protocol is passed, the public key identity will be signed by the IdP and then passed to the Acme server to verify the application authority of the public key (whether it is in the trusted list), thus restricting unauthorized public key identities.

In the process of authentication, the process of this document is quite different from the way of sso. The authentication process is primarily authenticated through public key authentication protocols (e.g. aPAKE/Opaque [I-D.irtf-cfrg-opaque]). The IdP here also needs to support the appropriate public key authentication protocols.

The redirection to the client then indicates that the authentication process has been completed, at which point it can be demonstrated whether the applicant has the private key corresponding to the public key in the application order. The user’s private key information is always private in this process and does not need to be provided to the agent. After completing the authentication, it redirects the client to the certificate request process.

In the last phase, after completing authentication the client eventually submits the CSR. This document requires that the public key contained in the CSR must match the public key in the starting order. In this way, tampering with public key information can be avoided. Other processes are consistent with the standard process.

6.2. 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 GET requests for pk_url. Upon receiving such a request:

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, (2) Opaque/AKE and (3) non-interactive zero-knowledge (NIZK) discrete logarithm equality (DLEQ) proof, 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 identifiers of type csr and selfsign-cert, identity consistency checks are also required. The challenge is successful if the check passes.

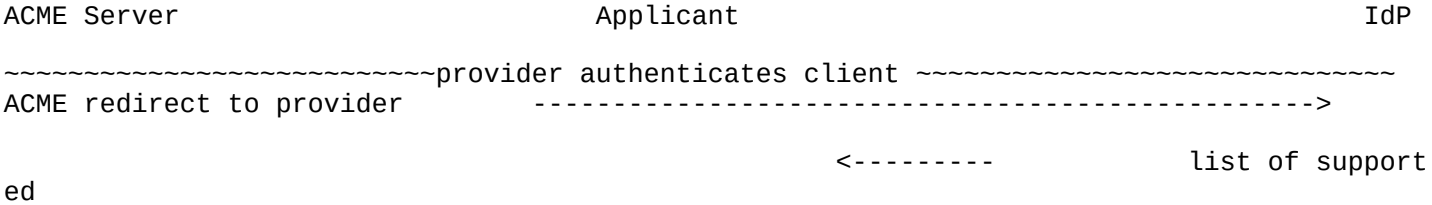

Figure 5: Overview of the Public Key Authentication Flow

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.

In the case of public key verification, the IdP acts as the asserting party and conducts a public key authentication agreement with the applicant to obtain public key information and identity information about the subject (applicant). The Acme server acts as a relying party that receives identity information from the IdP. The Acme server verifies the consistency of the public key and the order public key after receiving the public key and identity information. For the identifiers of type csr and selfsign-cert, the identity consistency needs to be further checked on the basis again.

The standard process, as defined in section 7.4 of [RFC8555]. Once the client has been authorized in all respects, it can proceed with the completion of the order. The client SHOULD use the public key declared in the order as the public key in the CSR. If the order identifier type is csr or selfsign-cert, the commonName, subjectAltName, etc. fields should be filled in the CSR. Then the CSR, encapsulated as specified, is submitted to the ACME server.

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. Changes to the Finalize Request

Since the Acme server has already obtained the user/device’s authenticated public key during the challenge phase, there is no need to send the CSR in the final phase, since the CSR no longer carries parts that are not present in other parts of the ACME protocol.

The process after removing the CSR, starting from the client issuing a new order request to the ACME server validating the challenge, is consistent with the process in Section 6. It is only in the final certificate issuance phase that the ACME server updates the challenge record to verified (or unverified) and automatically requests a certificate for the applicant (verified public key) when the result is verified. There is no need to send the CSR file again.

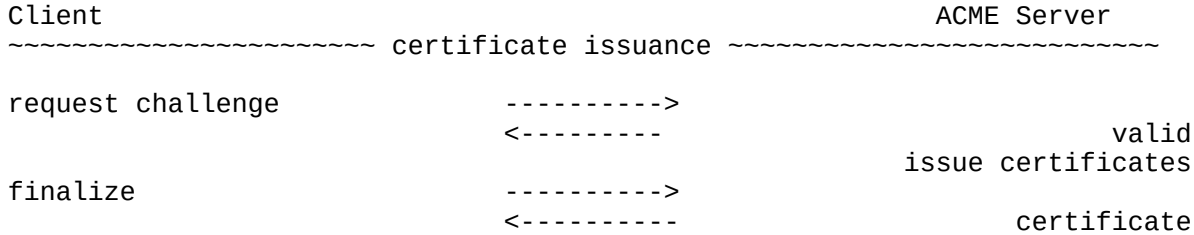

Figure 6: Overview of the PK Challenge Flow(remove CSR)

8. Further Use Cases

Temporary Certificate with One-time Key

The public key authentication protocol in pk-01 can be chosen OPAQUE protocol to realize the temporary certificate with one-time key. For example, in an enterprise cooperation scenario, each company has its own private CA, and a member of company A needs to have a temporary meeting with a member of company B, Bob, and needs to authenticate and authorize Bob. The most suitable method is that the CA of company A signs a temporary certificate for him for this meeting.

Alice, a legitimate user in company A, generates a temporary ID and password pw for Bob (pw performs OPAQUE protocol for registration), passes the ID and pw to Bob via out-of-band, and Bob generates a temporary public-private key pair (pk,sk) for this purpose. Bob registers the temporary public key pk at the ACME server after authentication through the OPAQUE protocol with the help of the password pw, and then applies for a temporary certificate issued by the CA of Company A that is bound to pk and ID.

8.1. 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.

8.2. 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.

9. IANA Considerations

9.1. ACME Identifier Types

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

Label	Reference
pk	RFC XXXX
csr	RFC XXXX
selfsign-cert	RFC XXXX

9.2. 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/csr/selfsign-cert	Y	RFC XXXX

10. References

10.1. 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>.

[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>.

[RFC8738] Shoemaker, R.B., "Automated Certificate Management Environment (ACME) IP Identifier Validation Extension", RFC 8738, DOI 10.17487/RFC8738, February 2020, <https://www.rfc-editor.org/info/rfc8738>.

[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>.

[RFC8823] Melnikov, A., "Extensions to Automatic Certificate Management Environment for End-User S/MIME Certificates", RFC 8823, DOI 10.17487/RFC8823, April 2021, <https://www.rfc-editor.org/info/rfc8823>.

10.2. Informative References

[I-D.biggs-acme-sso]
Biggs, A., Barnes, R., and R. Moynihan, "Automated Certificate Management Environment (ACME) Extension for Single Sign On Challenges", Work in Progress, Internet-Draft, draft-biggs-acme-sso-01, 8 April 2021, <https://datatracker.ietf.org/doc/html/draft-biggs-acme-sso-01>.

[I-D.acme-device-attest]
Weeks, B., "Automated Certificate Management Environment (ACME) Device Attestation Extension", Work in Progress, Internet-Draft, draft-acme-device-attest-04, 20 June 2025, <https://datatracker.ietf.org/doc/html/draft-acme-device-attest-04>.

[I-D.irtf-cfrg-opaque]
Bourdrez, D., Krawczyk, H., Lewi, K., and C. A. Wood, "The OPAQUE Augmented PAKE Protocol", Work in Progress, Internet-Draft, draft-irtf-cfrg-opaque-18, 21 November 2024, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-opaque-18>.

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