

Automated Certificate Management Environment
Internet-Draft
Intended status: Standards Track
Expires: 28 March 2026

C. P. Liu
Huawei
M. Ounsworth
Entrust Limited
M. Richardson
Sandelman Software Works Inc
24 September 2025

Automated Certificate Management Environment (ACME) rats Identifier and
Challenge Type
draft-liu-acme-rats-02

Abstract

This document describes an approach where an ACME Server can challenge an ACME Client to provide a Remote Attestation Evidence or Remote Attestation Result in any format supported by the Conceptual Message Wrapper.

The ACME Server can optionally challenge the Client for specific claims that it wishes attestation for.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://liuchunchi.github.io/draft-liu-acme-rats/draft-liu-acme-rats.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-liu-acme-rats/>.

Discussion of this document takes place on the Automated Certificate Management Environment Working Group mailing list (<mailto:acme@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/acme/>. Subscribe at <https://www.ietf.org/mailman/listinfo/acme/>.

Source for this draft and an issue tracker can be found at <https://github.com/liuchunchi/draft-liu-acme-rats>.

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 28 March 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Attestation Results only	5
1.2. Related work	5
2. Extensions -- trustworthy identifier	6
2.1. Step 1: newOrder Request Object	6
2.2. Step 2: Order Object	7
2.3. Step 3: Authorization Object	7
2.4. Step 4: Obtain Attestation Result	8
2.5. Step 5: Perform other challenges	9
2.6. Step 6: Finalize Order, retrieve certificate	9
3. ACME Extensions -- attestation-result-01 challenge type	9
3.1. attestation-result-01 Challenge	9
3.2. attestation-result-01 Response	10
4. ACME Extensions -- attestation-evidence-01 challenge type	10
4.1. attestation-evidence-01 Challenge	10
4.2. attestation-evidence-01 Response	11
5. ACME Attest Claims Hint Registry	11
6. Example use cases	11
6.1. Conflicting duties	12
6.2. Enterprise WiFi Access	12

6.3. BYOD devices	13
6.4. Private key in hardware	13
7. Security Considerations	13
8. IANA Considerations	14
8.1. ACME Attest Claims Hint Registry	14
9. References	15
9.1. Normative References	15
9.2. Informative References	15
Acknowledgments	17
Authors' Addresses	17

1. Introduction

ACME [RFC8555] is a standard protocol for issuing and renewing certificates automatically. When an ACME client needs a certificate, it connects to the ACME server, providing a proof of control of a desired identity. Upon success, it then receives a certificate with that identity in it.

These identities become part of the certificate, usually a Fully Qualified Domain Name (FQDN) that goes into the Subject Alt Name (SAN) for a certificate. Prior to ACME, the authorization process of obtaining a certificate from an operator of a (public) certification authority was non-standard and ad-hoc. It ranged from sending faxes on company letterhead to answering an email sent to a well-known email address like `hostmaster@example.com`, evolving into a process where some randomized nonce could be placed in a particular place on the target web server. The point of this process is to prove that the given DNS FQDN was controlled by the client system.

ACME standardized the process, allowing for automation for certificate issuance. It has been a massive success: increasing HTTPS usage from 27% in 2013 to over 80% in 2019 [letsencrypt].

While the process supports many kinds of identifiers: email addresses, DTN node IDs, and can create certificates for client use. However, these combinations have not yet become as popular, in part because these types of certificates are usually located on much more general purpose systems such as laptops and computers used by people.

The concern that Enterprises have with the use of client side certificates has been the trustworthiness of the client system itself. Such systems have many more configurations, and are often considered harder to secure as a result. While well managed mutual TLS (client and server authentication via PKIX certificate) has significant advantages over the more common login via username/password, if the private key associated with a client certificates is disclosed or lost, then the impact can be more significant.

The use case envisioned here is that of an enterprise. A Network Operations Center (NOC) (which may be internal or an external contracted entity) will issue (client) certificates to devices that can prove via remote attestation that they are running an up-to-date operating system as well as the enterprise-required endpoint security software.

This is a place where Remote Attestation can offer additional assurance [RFC9334]. If the software on the client is properly designed, and is up to date, then it is easier to assure that the private key will be safe.

This can be extended to Bring Your Own Device (BYOD) by having those devices provide an equivalent Attestation Result.

In this document, we propose an approach where ACME Server MAY challenge the ACME Client to produce an Attestation Evidence or Attestation Result in any format that is supported by the RATS Conceptual Message Wrapper [I-D.ietf-rats-msg-wrap], for instance, an EAT (entity attestation token). The ACME Server then verifies the attestation result against an appraisal policy as required by the requested certificate profile.

ACME can presently offer certificates with multiple identities. Typically, in a server certificate situation, each identity represents a unique FQDN that would be placed into the certificate as distinct Subject Alt Names (SAN). For instance each of the names: example.com, www.example.com, www.example.net and marketing.example.com might be placed in a single certificate for a server that provides web content under those four names.

This document defines a new identity type, trustworthy that the ACME client can ask for. A new attestation-result-01 challenge is defined as a new method that can be used to authorize this identity using a RATS Passport model. The encrypted-evidence-02 challenge is also defined, enabling a background check mechanism.

In this way, the Certification Authority (CA) or Registration Authority (RA) issues certificates only to devices that can provide an appropriate attestation result, indicating that the device from which the ACME request originates has passed the required security checks.

Attested ACME requests can form an essential building block towards the continuous monitoring/validation requirement of Zero-Trust principle when coupled with other operational measures, such as issuing only short-lived certificates.

For ease of denotation, we omit the "ACME" adjective from now on, where Server means ACME Server and Client means ACME Client.

1.1. Attestation Results only

This document currently only defines the a mechanism to carry Attestation Results from the ACME client to the ACME server. It limits itself to the Passport model defined in [RFC9334].

This is done to simplify the combinations, but also because it is likely that the Evidence required to make a reasonable assessment includes potentially privacy violating claims. This is particularly true when a device is a personal (BYOD) device; in that case the Verifier might not even be owned by the Enterprise, but rather the device manufacturer.

In order to make use of the background check that Evidence would need to be encrypted from the Attesting Environment to the Verifier, via the ACME Server -- the Relying Party. Secondly, in order for the ACME Server to be able to securely communicate with an Enterprise located Verifier with that Evidence, then more complex trust relationships would need to be established. Thirdly, the Enterprise Verifier system would then have to expose itself to the ACME Server, which may be located outside of the Enterprise. The ACME Server, for resiliency and loading reasons may be a numerous and dynamic cluster of systems, and providing appropriate network access controls to enable this communication may be difficult.

Instead, the use of the Passport model allows all Evidence to remain within an Enterprise, and for Verifier operations to be more self-contained.

1.2. Related work

[CSRATT] define trustworthy claims about the physical storage location of a key pair's private key. This mechanism only relates to how the private key is kept. It does not provide any claim about the rest of the mechanisms around access to the key. A key could well be stored in the most secure way imaginable, but in order to use the key some command mechanism must exist to invoke it.

The mechanism created in this document allows certification authority to access the trustworthiness of the entire system. That accessment goes well beyond how and where the private key is stored. ACME uses Certificate Signing Requests, so there is no reason that [CSRATT] could not be combined with the mechanism described in this document.

[RATSPA] defines a summary of a local assessment of posture for managed systems and across various layers. The claims and mechanisms defined in [RATSPA] are a good basis for the assessment that will need to be done in order to satisfy the trustworthiness challenge detailed in this document.

2. Extensions -- trustworthy identifier

This is a new identifier type.

type (required, string): The string "trustworthy".

value (required, string): The constant string "trustworthy"

The following sections detail the changes.

2.1. Step 1: newOrder Request Object

During the certificate order creation step, the Client sends a /newOrder JWS request (Section 7.4 of [RFC8555]) whose payload contains an array of identifiers.

The client adds the trustworthy identifier to the array.

This MUST NOT be the only identifier in the array, as this identity type does not, on its own, provide enough authorization to issue a certificate. In this example, a dns identity is chosen for the domain name client01.finance.example.

An example extended newOrder JWS request:

```
POST /acme/new-order HTTP/1.1
Content-Type: application/json
{
  "protected": base64url({
    "alg": "ES256",
  }),
  "payload": base64url({
    "identifiers": [
      { "type": "trustworthy", "value": "trustworthy" },
      { "type": "dns", "value": "client01.finance.example" },
    ],
  }),
  "signature": "H6ZXtGjTZyUnPeKn...wEA4Tk1Bdh3e454g"
}
```

2.2. Step 2: Order Object

As explained in [RFC8555], Section 7.1.3, the server returns an Order Object.

An example extended Order Object that includes

```
POST /acme/new-order HTTP/1.1
...

HTTP/1.1 200 OK
Content-Type: application/json
{
  "status": "pending",

  "identifiers": [
    { "type": "trustworthy", "value": "trustworthy" },
    { "type": "dns",          "value": "0123456789abcdef" },
  ],

  "authorizations": [
    "https://example.com/acme/authz/PAniVnsZcis",
    "https://example.com/acme/authz/Cluq5Dr+x8GSEJTSKW5B",
  ],

  "finalize": "https://example.com/acme/order/T..fgo/finalize",
}
```

Note that the URLs listed in the authorizations array are arbitrary URLs created by the ACME server. The last component is a randomly created string created by the server. For simplicity, the first URL is identical to the example given in [RFC8555].

2.3. Step 3: Authorization Object

The Server has created an Authorization Object for the trustworthy and dns identifiers.

The client accesses each authorization object from the URLs given in the Order Object. In this example, the PAniVnsZcis authorization relates to the dns identifier, and it is not changed from [RFC8555], Section 8.

The Cluq5Dr+x8GSEJTSKW5B authorization is a new authorization type, trustworthy, it is detailed in Section 3 and Section 4.

Here is an example:

```
GET https://example.com/acme/authz/C1uq5Dr+x8GSEJTSKW5B HTTP/1.1
..

HTTP/1.1 200 OK
Content-Type: application/json
{
  "status": "pending",
  "expires": "2025-09-30T14:09:07.99Z",

  "identifier": {
    "type": "trustworthy",
    "value": "trustworthy"
  },

  "challenges": [
    {
      "type": "trustworthy",
      "status": "pending",
      "token": "yoW1RL2zPBzYEHBQ06Jy",
      "url": "https://example.com/acme/chall/prV_8235AD9d",
    }
  ],
}
```

2.4. Step 4: Obtain Attestation Result

The client now uses the token yoW1RL2zPBzYEHBQ06Jy as a fresh nonce. It produces fresh Evidence, and provides this to the Verifier.

The details of this step are not in scope for this document. As an example, it might use TPM-CHARRA [RFC9684], or X, or Y (XXX: insert more options)

The format result is described in Section 3.2 and Section 4.2. (An example from [I-D.ietf-rats-ar4si] would be good here)

This result is sent as a POST to https://example.com/acme/chall/prV_8235AD9d

```
POST https://example.com/acme/chall/prV_8235AD9d HTTP/1.1
..
```

```
HTTP/1.1 200 OK
Content-Type: application/cmw+cbor
```

```
yePAuQj5xXAnz87/7ItOkDTk5Y4syow1RL2zPBzYEHBQ06JyUvZDYPYjeTqwlPszb9Grbxw0UAEFx5DxObV1
```

(EDIT: change to cwm+jws example)

The Server decodes the provided CMW [I-D.ietf-rats-msg-wrap]. The Attestation Results found within will be digitally signed by the Verifier.

The Server MUST verify the signature. The signature MUST be from a Verifier that the ACME Server has a trust anchor for. The list of trust anchors that a Server will trust is an attribute of the ACME Account in use. The details of how these trust anchors are configured is not in scope for this document.

At this point, if the client were to retrieve the authorization object from step 3, it would observe (if everything was accepted, verified) that the status for this challenge would now be marked as valid.

2.5. Step 5: Perform other challenges

The client SHOULD now perform any other challenges that were listed in the Order Object from step 2. ACME provides no ordering constraint on the challenges, so they could well have occurred concurrently.

2.6. Step 6: Finalize Order, retrieve certificate

At this point, the process continues as described in [RFC8555], Section 7.4. This means that the finalize action is used, which includes a CSR. If all is well, it will result in a certificate being issued.

3. ACME Extensions -- attestation-result-01 challenge type

A attestation-result-01 challenge type asks the Client to prove provide a fresh Attestation Result. This section describes the challenge/response extensions and procedures to use them.

3.1. attestation-result-01 Challenge

The attestation-result-01 Challenge works with Passport Model of RATS.

The corresponding Challenge Object is:

type (required, string): The string "attestation-result-01".

token (required, string): A randomly created nonce provided by the server which MUST be included in the Attestation Results to provide freshness.

attestClaimsHint (optional, list of string) If the Server requires attestation of specific claims or properties in order to issue the requested certificate profile, then it MAY list one or more types of claims from the newly-defined ACME Attest Claims Hints registry defined in Section 5.

Once fresh Attestation Results have been obtained from an appropriate RATS Verifier, then this result is posted to the URL provided in the url attribute.

3.2. attestation-result-01 Response

The data sent SHOULD be Attestation Results in the form of of a CMW [I-D.ietf-rats-msg-wrap], Section 5.2 tagged JSON encoded Attestation Results for Secure Interactions (AR4SI) [I-D.ietf-rats-ar4si]. The CM-type MUST include attestation-results, and MUST NOT include any other wrapped values. Other formats are permitted by prior arrangement, however, they MUST use the CMW format so that they can be distinguished.

4. ACME Extensions -- attestation-evidence-01 challenge type

A attestation-evidence-01 challenge type asks the Client to send fresh Evidence to the Server. The Server will use the RATS background model to connect to a Verifier, obtaining Attestation Results.

4.1. attestation-evidence-01 Challenge

The attestation-evidence-01 Challenge works with Background Model of RATS.

The corresponding Challenge Object is:

type (required, string): The string "attestation-evidence-01".

token (required, string): A randomly created nonce provided by the server which MUST be included in the Evidence to provide freshness.

verifierEncryptionCredential (optional, base64 encoded) This mode is for cases where the evidence of a device contains specific identifiers that could be linkable to a person and therefore qualify as Personally Identifiable Information. In these cases, the Server MAY opt to pass the evidence encrypted to the Verifier so that it never needs to handle to decrypted PII. The verifierEncryptionCredential can be of any type that is compatible with JWE encryption.

4.2. attestation-evidence-01 Response

Once fresh Evidence has been collected, then it is posted to the URL provided in the url attribute.

The data sent SHOULD be Evidence in the form of of a CMW [I-D.ietf-rats-msg-wrap], Section 5.2 tagged JSON encoded Evidence. The CMW-type MUST include Evidence, and MUST NOT include any other wrapped values. Other formats are permitted by prior arrangement, however, they MUST use the CMW format so that they can be distinguished.

If a verifierEncryptionCredential was provided by the Server, then the Client MUST encrypt the evidence by placing the entire CMW as the payload of a JWE encrypted for the verifierEncryptionCredential.

5. ACME Attest Claims Hint Registry

(EDIT: unclear if this is still important)

In order to facilitate the Server requesting attestation of specific types claims or properties, we define a new registry of ACME Claims Hints. In order to preserve flexibility, the Claim Hints are intended to be generic in nature, allowing for the client to reply with any type of attestation result that contains the requested information. As such, these values are not intended to map one-to-one with any specific remote attestation evidence or attestation result format, but instead they are to serve as a hint to the ACME Client about what type of attestation it needs to collect from the device. Ultimately, the CA's certificate policies will be the authority on what evidence or attestation results it will accept.

The ACME Attest Claims Hint Registry is intended to help clients to collect evidence or attestation results that are most likely to be acceptable to the Server, but are not a guaranteed replacement for performing interoperability testing between a given attesting device and a given CA. Similarly, an ACME attestation hint may not map one-to-one with attestation functionality exposed by the underlying attesting device, so ACME clients might need to act as intermediaries mapping ACME hints to vendor-specific functionality on a per-hardware-vendor basis.

See Section 8.1 for the initial contents of this new registry.

6. Example use cases

6.1. Conflicting duties

(EDIT: This text might be stale)

1. Integration/compatibility difficulty: Integrating SOC and NOC requires plenty of customized, case-by-case developing work. Especially considering different system vendors, system versions, different data models and formats due to different client needs... Let alone possible updates.
2. Conflict of duties: NOC people do not want SOC people to interfere with their daily work, and so do SOC people. Also, NOC people may have limited security knowledge, and SOC people vice versa. Where to draw the line and what is the best tool to help them collaborate is a question.

6.2. Enterprise WiFi Access

In enterprise access cases, security administrators wish to check the security status of an accessing end device before it connects to the internal network. Endpoint Detection and Response (EDR) softwares can check the security/trustworthiness statuses of the device and produce an Attestation Result (AR) if the check passes. ACME-RATS procedures can then be used to redeem a certificate using the AR.

With that being said, a more specific use case is as follows: an enterprise employee visits multiple campuses, and connects to each one's WiFi. For example, an inspector visits many (tens of) power substations a day, connects to the local WiFi, download log data, proceed to the next and repeat the process.

Current access solution include: 1. The inspector remembers the password for each WiFi, and conduct the 802.1X EAP password-based (PAP/CHAP/MS-CHAPv2) authentication. or 2. an enterprise MDM receives the passwords and usernames over application layer connection from the MDM server, and enter them on user's behalf. While Solution 1 obviously suffer from management burdens induced by massive number of password pairs, and password rotation requirements, the drawback of Solution 2 is more obscure, which include:

- a. Bring Your Own Device (BYOD) situation and MDM is not available.
- b. Password could risk leakage due to APP compromise, or during Internet transmission. Anyone with leaked password can access, without binding of trusted/usual devices.
- c. The RADIUS Client/Access Point/Switch is not aware of the identity of the accessing device, therefore cannot enforce more fine-grained access policies.

An ideal user story is: 1. When the inspector is at base (or whenever the Remote Attestation-based check is available), he get his device inspected and redeem a certificate using ACME-RATS. 2. When at substation, the inspector authenticate to the WiFi using EAP-TLS, where all the substations have the company root CA installed. 2*. Alternatively, the Step 2 can use EAP-repeater mode, where the RADIUS Client redirects the request back to the RADIUS Server for more advanced checks.

6.3. BYOD devices

Another example is issuing S/MIME certificates to BYOD devices only if they can prove via attestation that they are registered to a corporate MDM and the user they are registered to matches the user for which a certificate has been requested.

In this case, the Server might challenge the client to prove that it is properly-registered to the enterprise to the same user as the subject of the requested S/MIME certificate, and that the device is running the corporate-approved security agents.

6.4. Private key in hardware

In some scenarios the CA might require that the private key corresponding to the certificate request is stored in cryptographic hardware and non-extractable. For example, the certificate profile for some types of administrative credentials may be required to be stored in a token or smartcard. Or the CA might be required to enforce that the private key is stored in a FIPS-certified HSM running in a configuration compliant with its FIPS certificate -- this is the case, for example, with CA/Browser Forum Code Signing certificates [CABF-CSBRs] which can be attested for example via [RATSKA].

It could also be possible that the requested certificate profile does not require the requested key to be hardware-backed, but that the CA will issue the certificate with extra assurance, for example an extra policy OID or a longer expiry period, if attestation of hardware can be provided.

7. Security Considerations

The attestation-result-01 challenge (the Passport Model) is the mandatory to implement. The encrypted-evidence-01 challenge (the background-check model) is optional.

In all cases the Server has to be able to verify Attestation Results from the Verifier. To do that it requires appropriate trust anchors.

In the Passport model, Evidence -- which may contain personally identifiable information (PII)) -- is never seen by the ACME Server. Additionally, there is no need for the Verifier to accept connections from ACME Server(s). The Attester/Verifier relationship used in the Passport Model leverages a pre-existing relationship. For instance if the Verifier is operated by the manufacturer of the Attester (or their designate), then this is the same relationship that would be used to obtain updated software/firmware. In this case, the trust anchors may also be publically available, but the Server does not need any further relationship with the Verifier.

In the background-check model, Evidence is sent from the Attester to the ACME Server. The ACME Server then relays this Evidence to a Verifier. The Evidence is encrypted so that the Server it never able to see any PII which might be included. The choice of Verifier is more complex in the background-check model. Not only does ACME Server have to have the correct trust anchors to verify the resulting Attestation Results, but the ACME Server will need some kind of business relationship with the Verifier in order for the Verifier to be willing to appraise Evidence.

The trustworthy identifier and challenge/response is not an actual identifier. It does not result in any specific contents to the certificate Subject or SubjectAltName.

8. IANA Considerations

8.1. ACME Attest Claims Hint Registry

IANA is requested to open a new registry, XXXXXXXX

Type: designated expert

The registry has the following columns:

- * Claim Hint: the string value to be placed within an ACME device-attest-02 or device-attest-03 challenge.
- * Description: a description of the general type of attestation evidence or attestation result that the client is expected to produce.

The initial registry contents is shown in the table below.

Claim Hint	Description
FIPS_mode	Attestation that the device is currently booted in FIPS mode.
OS_patch_level	Attestation to the version or patch level of the device's operating system.
sw_manifest	A manifest list of all software currently running on the device.

Table 1

9. References

9.1. 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/rfc/rfc8555>>.
- [RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <<https://www.rfc-editor.org/rfc/rfc9334>>.
- [I-D.ietf-rats-msg-wrap] Birkholz, H., Smith, N., Fossati, T., Tschofenig, H., and D. Glaze, "RATS Conceptual Messages Wrapper (CMW)", Work in Progress, Internet-Draft, draft-ietf-rats-msg-wrap-17, 12 August 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-msg-wrap-17>>.
- [I-D.ietf-rats-ar4si] Voit, E., Birkholz, H., Hardjono, T., Fossati, T., and V. Scarlata, "Attestation Results for Secure Interactions", Work in Progress, Internet-Draft, draft-ietf-rats-ar4si-09, 15 August 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-ar4si-09>>.

9.2. Informative References

- [CSRATT] Ounsworth, M., Tschofenig, H., Birkholz, H., Wiseman, M., and N. Smith, "Use of Remote Attestation with Certification Signing Requests", Work in Progress, Internet-Draft, draft-ietf-lamps-csr-attestation-20, 7 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-csr-attestation-20>>.
- [RATSPA] Moriarty, K., Wiseman, M., Stein, A. J., and C. Nelogal, "Remote Posture Assessment for Systems, Containers, and Applications at Scale", Work in Progress, Internet-Draft, draft-ietf-rats-posture-assessment-03, 7 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-posture-assessment-03>>.
- [RATSKA] Ounsworth, M., Fiset, J., Tschofenig, H., Birkholz, H., Wiseman, M., and N. Smith, "PKIX Evidence for Remote Attestation of Hardware Security Modules", Work in Progress, Internet-Draft, draft-ietf-rats-pkix-key-attestation-01, 7 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-pkix-key-attestation-01>>.
- [I-D.draft-bweeks-acme-device-attest-01]
Weeks, B., "Automated Certificate Management Environment (ACME) Device Attestation Extension", Work in Progress, Internet-Draft, draft-bweeks-acme-device-attest-01, 7 August 2022, <<https://datatracker.ietf.org/doc/html/draft-bweeks-acme-device-attest-01>>.
- [letsencrypt]
Electronic Frontier Foundation, "Celebrating Ten Years of Encrypting the Web with Let's Encrypt", 20 August 2025, <<https://www.eff.org/deeplinks/2023/08/celebrating-ten-years-encrypting-web-lets-encrypt>>.
- [CABF-CSBRs]
CA/BROWSER FORUM, "Baseline Requirements for Code-Signing Certificates", n.d., <<https://cabforum.org/working-groups/code-signing/documents/>>.
- [RFC9684] Birkholz, H., Eckel, M., Bhandari, S., Voit, E., Sulzen, B., Xia, L., Laffey, T., and G. C. Fedorkow, "A YANG Data Model for Challenge-Response-Based Remote Attestation (CHARRA) Procedures Using Trusted Platform Modules (TPMs)", RFC 9684, DOI 10.17487/RFC9684, December 2024, <<https://www.rfc-editor.org/rfc/rfc9684>>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Chunchi Peter Liu
Huawei
Email: liuchunchi@huawei.com

Mike Ounsworth
Entrust Limited
Email: mike.ounsworth@entrust.com

Michael Richardson
Sandelman Software Works Inc
Email: mcr+ietf@sandelman.ca