Web Authorization Protocol
Internet-Draft
Intended status: Informational
Expires: 11 October 2024

R. L. Barnes
Cisco
S. Goldberg
BastionZero, Inc.

9 April 2024

Proof of Issuer Key Authority (PIKA)
draft-barnes-oauth-pika-00

Abstract

A relying party verifying a JSON Web Token (JWT) needs to verify that the public key used to verify the signature legitimately represents the issuer represented in the "iss" claim of the JWT. Today, relying parties commonly use the "iss" claim to fetch a set of authorized signing keys over HTTPS, relying on the security of HTTPS to establish the authority of the downloaded keys for that issuer. The ephemerality of this proof of authority makes it unsuitable for use cases where a JWT might need to be verified for some time. In this document, we define a format for Proofs of Issuer Key Authority, which establish the authority of a key using a signed object instead of an HTTPS connection.

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://bifurcation.github.io/redistributable-jwks/draft-barnes-oauth-redistributable-jwks.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-barnes-oauth-pika/.

Discussion of this document takes place on the Web Authorization Protocol Working Group mailing list (mailto:oauth@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/oauth/. Subscribe at https://www.ietf.org/mailman/listinfo/oauth/.

Source for this draft and an issue tracker can be found at https://github.com/bifurcation/redistributable-jwks.

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 11 October 2024.

Table of Contents

1. Introduction

A relying party verifying a JSON Web Token (JWT) [RFC7519] needs to verify that the public key used to verify the signature legitimately represents the issuer represented in the "iss" claim of the JWT.

Today, relying parties commonly use the iss claim to fetch a set of authorized signing keys over HTTPS, relying on the security of HTTPS to establish the authority of the downloaded keys for that issuer. For example, in OpenID Connect Discovery [OIDC-Discovery], the iss claim is used to form a URL from which issuer metadata is downloaded over HTTPS. The issuer's JWK set is linked via the jwks_uri field in the metadata. The SD-JWT-VC specification describes a similar HTTPS-based mechanism for discovering the valid keys for an issuer (see Section 5 of [I-D.ietf-oauth-sd-jwt-vc]).

These HTTPS-based authority mechanisms are "live", in the sense that they can only prove the authority of a key to someone who does an HTTPS transaction with the relevant server. The fact that the server needs to be reachable and responsive at the time the JWT is being validated is a serious limitation in some use cases, two examples of which are given below.

In this document, we define Proofs of Issuer Key Authority (PIKA), a format for a redistributable proof of authority for an issuer key. As in OIDC and SD-JWT-VC, we assume that issuers are identified by HTTPS URLs, or at least by domain names. A PIKA is then simply a JWT whose payload contains the issuer key in question, and whose header contains an X.509 certificate proving that the PIKA-signing key is authoritative for the issuer's domain name.

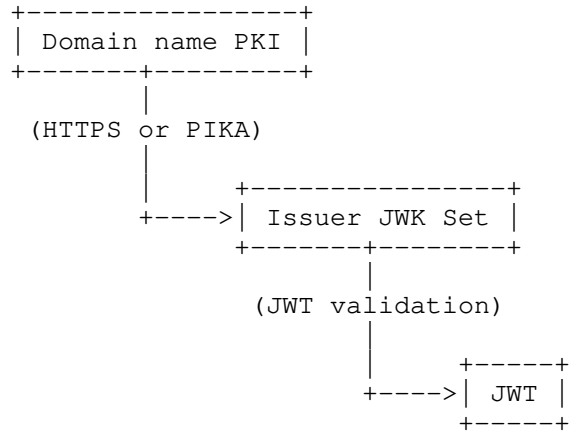

Figure 1: Trust model for PIKA or HTTPS-based discovery

This design preserves the same trust model as in the HTTPS-based proof of authority; it just swaps the signature in the TLS handshake underlying HTTPS for an object signature. PIKAs are thus "redistributable" in the same sense that an intermediate certificate would be, so that they can be verified without the issuer being online and reachable.

We also define a simple syntax for referencing PIKAs keys in metadata documents such as OIDC Discovery metadata and SD-JWT-VC issuer metadata.

1.1. Use Case: End-to-End Security

In applications using MLS for end-to-end security, endpoints can authenticate to each other using Verifiable Credentials (VCs) [I-D.barnes-mls-addl-creds]. These VCs are formatted as JWTs. In such applications, HTTPS-based proof of authority is an availability risk to the application and to the VC issuer.

The risk to the application is clear: A client joining an MLS group needs to validate the credentials of their peers. If part of that process entails making an HTTPS query to validate the authority of the keys used to sign their peers' credentials, and the relevant HTTPS server is down, then the client will not be able to join the group and use the application. Worse, since different peers may have credentials from different issuers, an outage at any one of those issuers can cause downtime for the application.

The use of HTTPS to validate authority also creates unnecessary load on the VC issuer. Consider, for example, an MLS-based video conference with 1,000 participants presenting credentials from 10 different issuers, all of whom join at the start of the meeting. This situation would create a spike of around 10,000 HTTPS requests to the VC issuer.

With PIKAs, the clients in a meeting can bundle the proof of authority along with their VC, avoiding the need for any HTTPS interaction with the issuer at all.

1.2. Use Case: Verifying Stored Signatures

Some applications are interested in verifying historical signatures. For example, a container registry might wish to demonstrate that a container was signed by its author at some time in the past.

Live HTTPS-based proofs of authority are fundamentally incompatible with these applications, since the proof of authority they produce cannot be preserved and reused later. With PIKAs, a trusted timestamping authority is all that is needed to achieve the desired properties.

Suppose the registry stores the following information for each container:

* A signature by the container author over the container

* A JWT attesting to the container author's identity and public key, e.g., a Verifiable Credential or an OpenPubkey PK Token [OpenPubkey]

* A PIKA providing the JWT issuer's key and proving its authority for the issuer

* An assertion by the timestamping authority that all of the above artifacts existed at a time in the past when they were all valid

Based on the timetamping authority's assertion, a relying party can validate that at the specified time, the container was signed by an author with the specified identity, and that the identity was asserted by the specified issuer.

1.3. Alternatives

An alternative design discussed in Section 3.5 of [I-D.ietf-oauth-sd-jwt-vc] is to simply sign the based JWT with an X.509 certified keys. This design has a few drawbacks relative to the design described here:

First, it changes the trust model relative to HTTPS-based proof of authority. The issuer JWT-signing key is removed as an intermediate step. This makes it more difficult for this design to coexist with HTTPS-based proof of identity.

Second, it removes flexibility that allows for efficiency. The extra data of the X.509 certificate chain has to be sent every time the base JWT is sent. Allowing the two to be decoupled allows for more flexible caching schemes.

2. Conventions and Definitions

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. Proof of Issuer Key Authority Format

Because the requirements for PIKAs are similar to those for OpenID Federation [OIDC-Federation], we re-use the Federation Historical Keys Response format as a base format for PIKAs.

A PIKA is a JWT meeting the requirements of the Historical Keys Response format in [OIDC-Federation]. In particular, the JWT Claims in a PIKA MUST contain iss, iat, and keys claims. Each JWK in the JWK Set in the keys claim MUST contain kid and exp claims, and SHOULD contain an iat claim.

A PIKA MUST also satisfy the following additional requirements:

* The iss field in the JWT Claims MUST be formatted as an HTTPS URL or a domain name.

* The JOSE Header of the PIKA MUST contain an x5c field. The contents of this field MUST represent a certificate chain that authenticates the domain name in the iss field. The domain name MUST appear as a dNSName entry in the subjectAltName extension of the end-entity certificate.

* The alg field of the PIKA MUST represent an algorithm that is compatible with the subject public key of the certificate in the x5c parameter.

* The JWT Claims in a PIKA SHOULD contain an exp claim. If an exp claim is not present, then a relying party MUST behave as if the exp field were set to the notAfter time in the end-entity certificate in the x5c field.

Figure 2 shows the contents of the JWT header and JWT payload for an example PIKA, omitting the full certificate chain:

```
JWT Header:
{
  "alg": "ES256",
  "typ": "JWT",
  "x5c": ["MII..."]
}

JWT Payload:
{
  "iss": "https://server.example.com",
  "iat": 123972394272,
  "exp": 124003930272,
  "keys":
    [
      {
        "kty": "EC",
        "crv": "P-256",
        "alg": "ES256"
        "x": "qiGKLwXRJmJR_AOQpWOHXLX5uYIfzvPwDurWvmZBwvw",
        "y": "ip8nyuLpJ5NpriZzCVKiG0TteqPMkrzfNOUQ8YzeGdk"
        "kid": "2HnoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs",
        "iat": 123972394872,
        "exp": 123974395972
      },
      {
        "kty": "RSA",
        "n": "ng5jr...",
        "e": "AQAB",
        "kid": "8KnoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMJJr",
        "iat": 123972394872,
        "exp": 123974394972
        "revoked": {
          "revoked_at": 123972495172,
          "reason": "keyCompromise",
          "reason_code": 1
        }
      }
    ]
}

JWT Signature:
// Signature over JWT Header and Claims, as defined in RFC 7519
```

Figure 2: An example Proof of Issuer Key Authority

A Verifier that receives such a PIKA validates it by taking the following steps:

1. If this PIKA was looked up using an iss value, verify that the value of the iss claim in the PIKA is identical to the one used to discover it.

2. Verify that the PIKA is currently valid, according to its iat and exp claims.

3. Verify that the certificate chain in the x5c field is currently valid from a trusted certificate authority (see [@!RFC5280][@!RFC6125]).

4. Verify that the end-entity certificate matches the iss field of the PIKA.

5. Verify the signature on the PIKA using the subject public key of the end-entity certificate

Before using a key in a PIKA to validate a JWT, a Verifier MUST verify that the time at which the JWT was signed (e.g., as expressed by its iat claim) is within the signing interval for the key. This interval is expressed by the iat and exp fields within the key attested to in the PIKA.

4. Referencing Proofs of Issuer Key Authority

JWT issuers commonly advertise their JWK Sets using mechanisms such as OpenID Connect Discovery or SD-JWT-VC Credential Issuer Metadata [OIDC-Discovery] [I-D.ietf-oauth-sd-jwt-vc]. These discovery mechanisms could be extended to also provide PIKAs, using one of a few approaches.

Current discovery mechanisms typically present the issuer's JWK set as a value or link embedded in the metadata object. Similarly, the Federation Historical Keys endpoint in OpenID Federation provides a link from which the issuer's historical keys may be downloaded (see Section 5.1.1 of [OIDC-Federation]). These mechanisms are illustrated in Figure 3.

```
{
    // Other metadata...

    // Current mechanisms for unsigned JWKS
    "jwks_uri": "https://example.com/jwks",
    "jwks": { "keys": [ ... ] },

    // OpenID Federation historical keys
    "federation_historical_keys_endpoint": "https://example.com/historical_key
s",
}
```

Figure 3: Current mechanisms for provided an issuer JWK Set

A similar field could be defined to provide a single set of issuer keys expressed as a PIKA, either by reference or by value. Such a mechanism requires the issuer to list all of the keys that are currently valid in one PIKA, requiring a Relying Party to download the whole PIKA even if they are only interested in one key.

An alternative design would allow for more specific PIKAs, covering individual keys and referencing them by kid. With such a design, an issuer metadata object would contain a map like the following (showing three keys with kid values "us-east-2024-01", "us-west-2024-01", and "us-east-2024-04"):

```
{
  // Other metadata...

  "signed_jwks": {
    "us-east-2024-01": "https://example.com/signed_jwks/us-east-2024-01",
    "us-west-2024-01": "https://example.com/signed_jwks/us-east-2024-01",
    "us-east-2024-04": "https://example.com/signed_jwks/us-east-2024-01",
  }
}
```

Figure 4: Referencing individual PIKAs by Key ID

5. Security Considerations

The main difference between establishing the authority of issuer keys via PIKA vs. via HTTPS is that where HTTPS is ephemeral, a PIKA can be redistribted and verfied for some period of time (until its exp time). Issuers should exercise care in choosing the exp value they populate in a PIKA, in order to avoid a key being used beyond its intended lifetime.

An issuer may wish to revoke a key, in the sense of instructing verifiers that they should no longer use the key to validate JWTs from the issuer. PIKAs provide both implicit and explicit revocation. With implicit revocation, the issuer simply removes the key from PIKAs it publishes. With explicit revocation, the issuer adds a revoked field to the key, as described in [OIDC-Federation]. In either case, the key will no longer be used by verifiers once all PIKAs positively authorizing the key have expired.

The above properties imply an operational trade-off for issuers. On the one hand, having shorter PIKA validity times means that the issuer can revoke keys more quickly. On the other hand, having short PIKA validity times will require PIKAs to be signed more often, and result in higher load on endpoints by which PIKAs are distributed.

6. References

6.1. Normative References

[OIDC-Federation]
"OpenID Federation 1.0 - draft 33", 23 February 2024, <https://openid.net/specs/openid-federation-1_0.html>.

[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/rfc/rfc2119>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/rfc/rfc7519>.

[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/rfc/rfc8174>.

6.2. Informative References

[I-D.barnes-mls-addl-creds]
Barnes, R. and S. Nandakumar, "Additional MLS Credentials", Work in Progress, Internet-Draft, draft-barnes-mls-addl-creds-01, 4 March 2024, <https://datatracker.ietf.org/doc/html/draft-barnes-mls-addl-creds-01>.

[I-D.ietf-oauth-sd-jwt-vc]
Terbu, O., Fett, D., and B. Campbell, "SD-JWT-based Verifiable Credentials (SD-JWT VC)", Work in Progress, Internet-Draft, draft-ietf-oauth-sd-jwt-vc-03, 4 March 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-sd-jwt-vc-03>.

[OIDC-Discovery]
"OpenID Connect Discovery 1.0 incorporating errata set 2", 15 December 2023, <https://openid.net/specs/openid-connect-discovery-1_0.html>.

[OpenPubkey]
"OpenPubkey", n.d.,
<https://www.bastionzero.com/openpubkey>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Richard L. Barnes
Cisco
Email: rlb@ipv.sx

Sharon Goldberg
BastionZero, Inc.
Email: goldbe@bastionzero.com