Messaging Layer Security
Internet-Draft
Intended status: Informational
Expires: 14 September 2023

R. Barnes
S. Nandakumar
Cisco

13 March 2023

UserInfo Verifiable Credentials as MLS Credentials
draft-barnes-mls-userinfo-vc-00

Abstract

This specification extends Message Layer Security (MLS) credentials framework with a new credential type, "UserInfoVC", based on the OpenID Connect UserInfo Verifiable Credential type "UserInfoCredential". A UserInfo Verifiable Credential encapsulates the UserInfo claims from the OpenID provider as a Verifiable Credential that can be presented to a third-party Verifier. These credentials can be easily provisioned to MLS clients using the OpenID Connect login flows, augmented with type "UserInfoCredential". The credential itself is an object associating identity attributes to the signature public key that the client will use in MLS, signed by the OpenID Provider. In situations where the OpenID Provider is distinct from the MLS Delivery Service, these credentials provide end-to-end secure identity assurance.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-barnes-mls-userinfo-vc/.

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

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

Status of This Memo

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 14 September 2023.

Table of Contents

1. Introduction

MLS provides end-to-end authenticated key exchange [@!I-D.ietf-mls-protocol]. As described in the MLS architecture, MLS requires an Authentication Service (AS) as well as a Delivery Service (DS) [@!I-D.ietf-mls-architecture]. The full security goals of MLS are only realized if the AS and DS are non-colluding. In other worlds, applications can deploy MLS to get end-to-end encryption (acting as MLS Delivery Service), but they need to partner with a non-colluding Authentication Service in order to achieve full end-to-end security.

OpenID Connect is widely used to integrate identity providers with applications, but its current core protocol doesn't provide the binding to cryptographic keys required for end-to-end security. When OpenID Connect is coupled with the "Verifiable Credentials"

framework, however, it can be used to provision clients with signed "UserInfo VC" objects that contain the critical elements of a credential to be used in MLS:

* Identity attributes for the user of a client

* A public key whose private key is held by a client

* A signature over the above by a trusted identity provider

The required updates to OpenID Connect are specfied in [OpenIDUserInfoVC]. That document defines a profile of the OpenID for Verifiable Credential Issuance protocol for issuing "UserInfo Verifiable Credentials". These credentials bind a signature key pair to the user attributes typically exposed through the OpenID Connect UserInfo endpoint.

In this document, we describe a "UserInfoVC" credential type for MLS that encapsulates a signed UserInfo object as Verifiable Credential, so that it can be used for authenticating an MLS client. We also describe the validation process that MLS clients use to verify UserInfoVC objects that they receive via MLS.

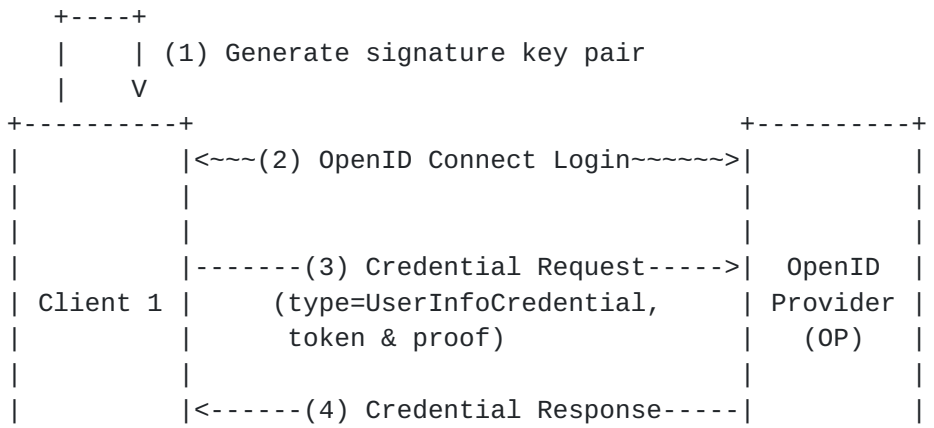
2. Conventions and Definitions

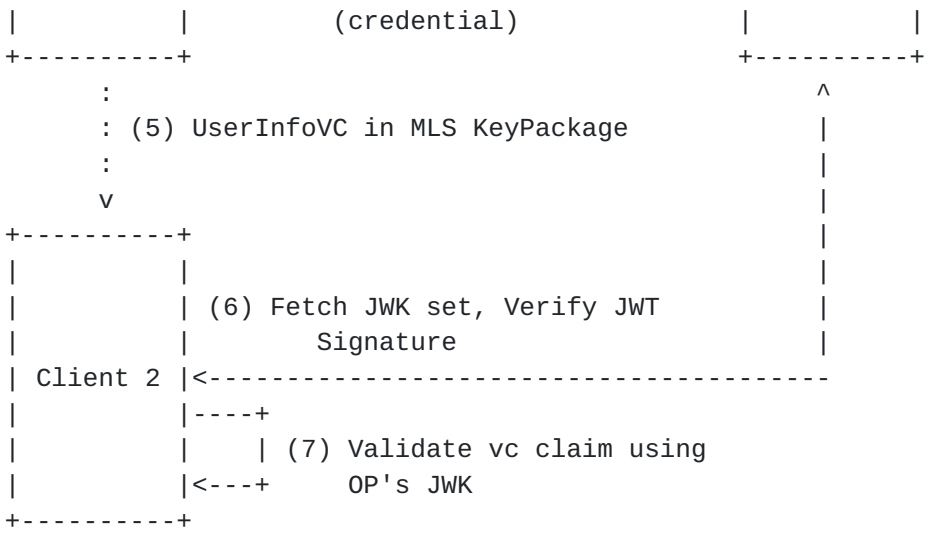
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2.1. Terminology

This specification uses terms from the MLS Protocol specification. In particular, we refer to the MLS Credential object, which represents an association between a client's identity and the signature key that the client will use to messages in the MLS key exchange protocol.

3. Concept





OpenID Connect UserInfo VC MLS Credential Flow

The basic steps showing OIDC Verifiable Credential based MLS credential flow are shown above.

Client 1 acts as an Holder (in the VC model) and as an MLS client. Client 2 is an MLS client and acts as Verifier (in the VC model) and implements certain OpenID Connect operations that enable it to verify signed UserInfo VC objects.

1. Client 1 generates a signature key pair using an algorithm that is supported by both MLS and UserInfo VC.

2. Client 1 performs an OpenID Connect login interaction with the scope "userinfo_credential" to obtain UserInfo VCs.

3. Client 1 sends a Credential Request specifying that it desires a UserInfo VC, together with a proof that it controls the private key of a signature key pair and the access token.

4. The OpenID Provider verifies the proof and create a Credential Response containing the UserInfo VC attesting the claims that would have been provided by the UserInfo endpoint and public key corresponding to the private key used to compute the proof in the Credential Request.

5. Client 1 generates a UserInfoVC MLS Credential object with the signed UserInfo VC JWT. Client 1 embeds the UserInfoVC in an MLS KeyPackage object and signs the KeyPackage object with the corresponding private key.

6. Client 1 sends the KeyPackage to Client 2, e.g., by posting it to a directory from which Client 2 fetches it when it wants to add Client 1 to a group.

7. Client 2 verifies the signature on the KeyPackage and extracts

the UserInfoVC credential. Client 2 uses OpenID Connect Discovery to fetch the OpenID Provider's JWK set.

8. Client 2 verifies the signed UserInfo VC using the the appropriate key from the OpenID Provider's JWK set.

If all checks pass, Client 2 has a high degree of assurance of the identity of Client 1. At this point Client 1's KeyPackage (including the VerifiableCredential) will be included in the MLS group's ratchet tree and distributed to the other members of the group. The other members of the group can verify the VerifiableCredential in the same way as Client 2.

4. UserInfoVC

A new credential type UserInfoVC is defined as shown below. This credential type is indicated with CredentialType userinfo_vc (see Section 7).

```
struct {
    opaque vc<0..2^32-1>;
} UserInfoVC;
```

The vc field contains the signed JWT-formatted UserInfo VC object (as defined in [OpenIDUserInfoVC]), encoded using UTF-8. The payload of object MUST provide iss and vc claims. The iss claim is used to look up the OpenID Provider's metadata. The vc claim contains authenticated user attributes and a public key binding. Specifically, the field vc.credentialSubject.id contains a did:jwk URI describing the subject's public key as a JWK.

4.1. Credential Validation

An MLS client validates a UserInfoVC credential in the context of an MLS LeafNode with the following steps:

* Verify that the jwt field parses successfully into a JWT [!@RFC7519], whose payload parses into UserInfo object as defined in Section 5.3.2 of [!@OpenID].

* Verify that an iss claim is present in the UserInfo VC payload and that "iss" value represents and issuer that is trusted according to the client's local policy.

* Verify the JWT signature:

  - Fetch the issuer metadata using OIDC Discovery [@!OpenID.Discovery].

  - Use the jwks_uri field in the metadata to fetch the JWK set.

  - Verify that the JWT signature verifies under one of the keys in

the JWK set.

* Verify the key binding:

  - Verify that a vc claim is present in the UserInfo VC payload.

  - Verify that the value of the claim is a JSON object that contains a credentialSubject field, as defined in Section 4 of openid-userinfo-vc.

  - Verify id field exists and it MUST be a a Decentralized Identifier with DID method jwk (W3c.did-core).

  - Verify that the jwk field parses as a JWK.

  - Verify that the signature_key in the LeafNode matches the key in the id field.

If all of the above checks pass, the client can use the signature key in the JWK for verifying MLS signatures using the signature scheme corresponding to the kty and crv parameters in the JWK. The identity attributes in the JWT should be associated with the MLS client that presented the credential.

4.2. Mapping between JWK Key Types and MLS Ciphersuites

Below table maps JWK key types (kty) and elliptic curves (crv) to the equivalent MLS signature scheme.

kty	crv	TLS/MLS signature scheme
EC	P-256	ECDSA with P-256 and SHA-256
EC	P-384	ECDSA with P-384 and SHA-384
EC	P-521	ECDSA with P-521 and SHA-512
EC	Ed25519	Ed25519
EC	Ed448	Ed448

Table 1

5. Security Considerations

The validation procedures specified verify that a JWT came from a given issuer. It doesn't veirfy that the issuer is authorative for the claimed attributes. The client needs to verify that the issuer is trusted to assert the claimed attributes.

6. Privacy Considerations

UserInfo can contain sensitive info such as human names, phone numbers, and using these credentials in MLS will expose this information to other group members, and potentially others if used in a prepublished KeyPackage.

7. IANA Considerations

7.1. MLS Credential Type

IANA is requested to register add the following new entry to the MLS Credential Type registry.

Value	Name	Recommended	Reference
0x0003	userinfo-vc	Y	RFC XXXX

Table 2

8. Normative References

[OpenIDUserInfoVC]
Ansari, M., Barnes, R., Kasselman, P., and K. Yasuda, "OpenID Connect UserInfo Verifiable Credentials 1.0", 15 December 2022, <https://openid.net/specs/openid-connect-userinfo-vc-1_0.html>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Richard Barnes
Cisco
Email: rlb@ipv.sx

Suhas Nandakumar
Cisco
Email: snandaku@cisco.com