UAF Authenticator Metadata

Specification Set: fido-uaf-v1.0-rd-20140209 REVIEW DRAFT

Editors:

Davit Baghdasaryan, NokNok Labs Inc
Brad Hill, PayPal Inc

Contributors:

Abstract:

FIDO Authenticators may have many different form factors, characteristics and capabilities. This document defines a standard means to describe the relevant pieces of information about an Authenticator in order to interoperate with it, or to make risk-based policy decisions about transactions involving a particular authenticator.

This document is a REVIEW DRAFT and subject to updated requirements from the FIDO Alliance Certification Working Group.

Status:

This Specification has been prepared by FIDO Alliance, Inc. **This is a Review Draft Specification and is not intended to be a basis for any implementations as the Specification may change**. Permission is hereby granted to use the Specification solely for the purpose of reviewing the Specification. No rights are granted to prepare derivative works of this Specification. Entities seeking permission to reproduce portions of this Specification for other uses must contact the FIDO Alliance to determine whether an appropriate license for such use is available.

Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The FIDO Alliance, Inc. and its Members and any other contributors to the Specification are not, and shall not be held, responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

THIS FIDO ALLIANCE SPECIFICATION IS PROVIDED “AS IS” AND WITHOUT ANY WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTY OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

1 Notation

Type names, attribute names and element names are written in *italics*.

String literals are enclosed in “”, e.g. “UAF-TLV”.

In formulas we use “|” to denote byte wise concatenation operations.

UAF specific terminology used in this document is defined in [FIDOGlossary].

1.1 Key Words

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

1.2 Revision History

This revision history may be subsumed by the SVN checkin comments and/or JIRA comments once that is integrated.

In any case, I would expect this section to disappear as part of the publication process.

Revision	Date	Author	Summary
0.1	May 7, 2013	Davit Baghdasaryan	Draft
0.2	Dec 6, 2013	Brad Hill	Update to coincide with FIDO Registry of Predefined Values
0.3	Jan 7, 2014	Brad Hill	Update to Open Office format.
0.4	01/10/14	Brad Hill	Metadata updates
0.5	02/01/14	Brad Hill	Removed references to "unsigned long" as these are not valid JSON types, fixed bibliography, other clarifying text, esp. regarding CWG impact on the future of this document.
0.6	02/02/14	Brad Hill	Fixed type of AssertionScheme

			and AuthenticationAlgorithm

2 Overview

The FIDO family of protocols enable simpler and more secure online authentication utilizing a wide variety of different devices in a competitive marketplace. Much of the complexity behind this variety is hidden from Relying Party applications, but in order to accomplish the goals of FIDO, Relying Parties must have some means of discovering and verifying various characteristics of Authenticators. Relying Parties can learn a subset of verifiable information for Authenticators certified by the FIDO Alliance with an Authenticator Metadata statement.

For definitions of terms, please refer to the FIDO Glossary [FIDOGlossary].

2.1 Scope

This document describes the format of and information contained in an Authenticator Metadata statement. For a definitive list of possible values for the various types of information, refer to the FIDO Registry of Predefined Values [FIDORegistry].

This document does not describe the processes and methods by which Authenticator Metadata statements are created, certified, signed, verified or distributed. This information is under definition by the FIDO Alliance Certification Working Group, and the scope and values for the metadata statements defined in this document are also subject to change by that group.

2.1.1 Audience

The intended audience for this document includes:

- **FIDO Authenticator Vendors** who wish to produce metadata statements for their products.
- **FIDO Server Implementers** who need to consume metadata statements to verify characteristics of authenticators and attestation statements, make proper algorithm choices for protocol messages, create policy statements or tailor various other modes of operation to authenticator-specific characteristics.
- **FIDO Relying Parties** who wish to
    - create custom policy statements about which authenticators they will accept
    - risk score authenticators based on their characteristics
    - verify attested authenticator IDs for cross-referencing with third party metadata

2.2 Architecture

Authenticator Metadata statements are used directly by the FIDO Server at a Relying Party, but the information contained in the authoritative statement is used in several other places. How a server obtains these metadata statements is out of scope for this document.

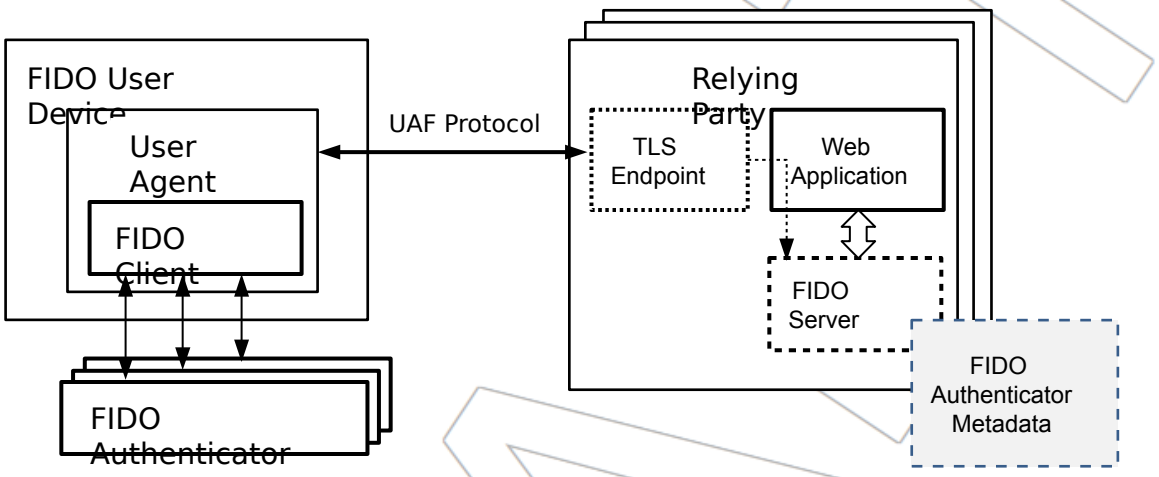

Figure 2.1: The UAF Architecture

The workflow around an authenticator metadata statement is as follows:

1. The authenticator vendor produces a metadata statement describing the characteristics of an authenticator.
2. Following a certification process yet to be defined, the metadata statement is distributed to FIDO Servers.
3. A Relying Party configures their registration policy to allow authenticators matching certain characteristics to be registered.
4. The FIDO Server sends a registration challenge message with this policy statement.
5. The FIDO Client receives the policy statement as part of the challenge message. It queries available authenticators for their self-reported characteristics and (with the user's input) selects an authenticator to register that matches the policy.
6. The FIDO Client processes and sends a registration response message to the server. This message contains the AAID for the authenticator and, optionally, a signature made with the authenticator's attestation certificate.
7. The FIDO Server looks up the metadata statement for the authenticator using its AAID. If the metadata statement lists an attestation certificate(s), it verifies that an attestation signature is present and from a certificate that chains to one of the listed certificates.

8. The FIDO Server next verifies that the authenticator meets the originally supplied registration policy based on its authoritative metadata statement. This prevents a faulty, modified or compromised FIDO Client from registering authenticators that are out of policy.
9. *Optionally*, a FIDO Server may, with input from the Relying Party, assign a risk or trust score to the authenticator based on its metadata, including elements not selected for by policy.
10. *Optionally,* a FIDO Server may cross-reference the attested AAID of the authenticator with other metadata database published by third parties. Such third party metadata might, for example, inform the FIDO Server if an authenticator has achieved certifications relevant to certain markets, industry verticals or whether it meets application-specific regulatory requirements.

3 Metadata Values

Name	JSON Type	Description
AAID	String	The Authenticator Attestation ID
AttestationRootCertificate	String[]	Each element of this array represents a PKIX [RFC5280] trust root X.509 certificate that is valid for this AAID. Multiple certificates might be used for different batches without distinct AAIDs. The array does not represent a certificate chain, but only the trust anchor. A certificate listed here is a trust root. It might be the actual certificate presented by the authenticator, or it might be an issuing authority certificate from the vendor that the actual certificate in the authenticator chains to. Each array element is a Base64-encoded [RFC4648] DER [ITU.X690.2008] PKIX certificate value. Each element MUST be dedicated for Authenticator attestation. If this value is blank, it indicates that the authenticator does not provide an attestation.
Description	String	A human-readable short description of the Authenticator
UserVerificationMethods	Number	A 64 bit number representing the bit fields defined by the USER_VERIFY constants in the FIDO Registry of Predefined Values. [FIDORegistry] Any number of the bits not defined as mutually exclusive may be set.
ValidAttachmentTypes	Number	A 64 bit number representing the bit fields defined by the ATTACHMENT_HINT constants in the FIDO Registry of Predefined Values. [FIDORegistry] The connection state and topology of an authenticator may be transient and cannot be relied on as authoritative by a Relying Party, but the metadata field should have all the bit flags set for

Name	JSON Type	Description
		the topologies possible for the authenticator. For example, an authenticator instantiated as a single-purpose hardware token that can communicate over bluetooth should set ATTACHMENT_HINT_EXTERNAL but not ATTACHMENT_HINT_INTERNAL.
KeyProtection	Number	A 64 bit number representing the bit fields defined by the KEY_PROTECTION constants in the FIDO Registry of Predefined Values. [FIDORegistry]
SecureDisplay	Number	A 64 bit number representing the bit fields defined by the SECURE_DISPLAY constants in the FIDO Registry of Predefined Values. [FIDORegistry]
SecureDisplayContentTypes	String[]	List of supported MIME content types [RFC1341] for the Secure Display, such as “text/plain” and “image/png”.
SecondFactorOnly	Boolean	Indicates if the Authenticator is designed to be used only as a second factor.
Logo	String	A Base64 [RFC4648] encoded PNG [PNG] logo for the Authenticator.
AssertionScheme	String	The assertion scheme supported by the Authenticator. Must be set to one of the enumerated Strings defined in the FIDO UAF Registry of Predefined Values. [FIDORegistry]
AuthenticationAlgorithm	Number	The Authentication algorithm supported by the authenticator. Must be set to one of the UAF_ALG_* constants defined in the FIDO UAF Registry of Predefined Values. [FIDORegistry]
AttestationType	String[]	The supported attestation type(s). (e.g. "Basic") See the UAF Protocol Specification for more information. [UAFProtocol]
UPV	Number[][]	The version(s) of the UAF protocol sup-

Name	JSON Type	Description
		ported. This is an array of 2-element arrays. The first index is the set of all protocol versions supported, each of which is indicated by a major version number at [*n*][0] and a minor version number at [*n*][1].

4 Metadata Statement Format

A FIDO Authenticator Metadata Statement is a JSON Web Signature (JWS) [JWS] where the JWS Payload is a JSON document containing the values defined in section [3] of this document.

Valid signature algorithms and key formats are to be determined by the FIDO Alliance Certification Working Group.

Bibliography

FIDO Alliance Documents:

[FIDOGlossary] Rolf Lindemann, Davit Baghdasaryan, Brad Hill, John Kemp. FIDO Technical Glossary. Version v1.0-rd-20140209, FIDO Alliance, February 2014. See http://fidoalliance.org/specs/fido-glossary-v1.0-rd-20140209.pdf

[UAFProtocol] Rolf Lindemann, Davit Baghdasaryan, Eric Tiffany. FIDO Universal Authentication Framework Protocol. Version v1.0-rd-20140209, FIDO Alliance, February 2014. See http://fidoalliance.org/specs/fido-uaf-protocol-v1.0-rd-20140209.pdf

[FIDORegistry] Rolf Lindemann, Davit Baghdasaryan, Brad Hill. FIDO Universal Authentication Framework Registry of Predefined Values. Version v1.0-rd-20140209, FIDO Alliance. February 2014. See http://fidoalliance.org/specs/fido-uaf-reg-v1.0-rd-20140209.pdf

Other References:

[ITU.X690.2008] X.690: Information technology – ASN.1 encoding rules: Specifications of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), (T-REC-X.690-200811)International Telecommunications Union, November 2008

[JWS] JSON Web Signature (JWS) draft-jose-json-web-signature, M. Jones. Work in progress.

[**PNG**] **Portable Network Graphics (PNG) Specification (Second Edition)** Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification. ISO/IEC 15948:2003 (E), D. Duce, Ed., World Wide Web Consortium, November 2003

[RFC1341] MIME (Multipurpose Internet Mail Extensions) (RFC1341), N. Borenstein et al, June 1992

[RFC4648] The Base16, Base32, and Base64 Data Encodings (RFC 4648), S. Josefsson, October 2006

[RFC5280] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC5280), D. Cooper et al, May 2008