FIDO Authentication and EMV 3-D Secure: Using FIDO for Payment Authentication

September 2020

Editors:

Brian Piel, Mastercard
Rolf Lindemann, Nok Nok Labs

Contents

Introduction

The FIDO Alliance defines standards that enable strong consumer authentication and seeks to use those standards to improve security on the internet. EMV 3-D Secure (EMV 3DS) is a payment industry standard for performing consumer verification and authentication within the context of online payments via credit cards. EMV 3DS also standardizes payment transaction information which is sent from a merchant to the issuing bank and includes data about the cardholder account, payment environment, and actions taken during payment. Using this data, the card issuing bank or a party operating on their behalf can perform transaction risk assessment and minimize the need to apply unnecessary friction to a payment transaction when it is deemed low risk. This is also known as "frictionless authentication" within the EMV 3DS standard.

FIDO Authentication standards are specified in terms of how a person uses an authenticator with a client platform to authenticate to a specific relying party. For example, the client platform could be a web browser and the relying party a website that a user is logging into. However, in the context of credit cards, the relying party is not a single entity: users interact with merchants to make a purchase, but the ultimate approver is the issuing bank of the credit card being used.

The issuing bank would receive the strongest assurance if it took the role of the FIDO/WebAuthn relying party itself, using the merchant as a conduit. It could issue challenges to authenticators, judge the signed responses, and potentially use transaction authentication to ensure that specific messages were displayed in a web browser or on the authenticator itself. The drawback in this instance is that the existing FIDO standards partition different relying parties. From the point of view of the client platform, the merchant is the counterparty, so credentials from an issuing bank would not be usable in the current iteration of the specifications. These issues could be addressed using redirects in the web context or dedicated support in other client platforms, but that introduces other costs and a significant amount of complexity for issuing banks to take on the role of relying party.

This document focuses on the role of the merchant as the FIDO or WebAuthn relying party and defines the methods for the merchant to leverage EMV 3DS as the conduit to report FIDO Authentication Data to the issuing bank. This data, along with the other transaction details sent using EMV 3DS messaging via the 3DS *Authentication Request* message, can help ensure minimized friction through risk-based authentication at the time of online payment. Although the resultant assurance level is reduced using this method, as opposed to an issuer-managed credential, and it will need to be viewed within the context of the entire EMV 3DS message, it can provide an approach that can be more easily deployed at scale than issuer-managed FIDO Authentication methods.

EMVCo and EMV 3-D Secure

EMVCo is a standards body with members representing the six major global credit card payments networks: American Express, Discover, JCB, Mastercard, Union Pay, and Visa, as well as other major payment industry contributors including banks, major electronic commerce merchants, payment service providers, and other payment industry participants. Many card payments standards have been created and distributed by EMVCo including physical payments standards pertaining to chip cards, contactless card payment, mobile phone payments, etc. One of the standards published by the organization is EMV 3-D Secure (EMV 3DS), which defines a messaging framework and ecosystem for digital merchants to share relevant transactional information to the card issuing banks via payment networks, as well as standardized cardholder authentication performed by issuing banks within ecommerce payment flows. Note that EMV 3DS also covers many other payments and non-payment authentication use cases, although these are out of scope of this document.

The term merchant as used in this document may consist of many parts. For example, businesses engaged in online commerce may use the services of a payment service provider (PSP) to perform transactions on their behalf or may outsource other activities related to payments as well. The transaction processing may be split into several systems depending on the configuration. Ultimately, in the terminology of EMV 3DS, a *3DS Requestor* begins a transaction by sending an *Authentication Request (AReq)* to the issuing bank's *Access Control Server (ACS)* via the payment network Directory Server. The EMV 3DS *AReq* is formatted as a JSON message, the fields of which (see EMV 3DS specification section B.1 for a list) include a field *3DS Requestor Authentication Information* (see EMV 3DS specification section A.7.3) and also specifies the format of another subfield, *3DS Requestor Authentication Data* which is in a string format and can carry relevant authentication data. This subfield will carry the FIDO Authentication Data as defined in this document. The current EMV 3DS version also reserves the *3DS Requestor Authentication Method* string "06" for "Login to the cardholder account at the 3DS requestor system using a FIDO Authenticator". Also note that the method "06" is a two-character string in the 3DS JSON and not the number six, and that method "07" is also reserved for FIDO-derived information but is not discussed in this document.

The full EMV 3-D Secure Protocol and Core Functions Specification version 2.2.0 is publicly available for download here: https://www.emvco.com/wp-content/plugins/pmpro-customizations/oy-getfile.php?u=/wp-content/uploads/documents/EMVCo_3DS_Spec_v220_122018.pdf

FIDO Authentication Data in EMV 3DS Messaging

The FIDO Authentication Data as defined in this section for use within EMV 3DS messaging is generated by a participating FIDO server and will require the data to be sent to a 3DS Requestor in the EMV 3DS *Authentication Request* within the *3DS Requestor Authentication Data* field. It is critical that the FIDO server can ensure consistency of the FIDO Authentication Data provided to a 3DS Requestor since the EMV 3DS messaging used to carry this data will be leveraged by the card issuing bank's ACS to assess risk. There are several defined data elements extracted from the FIDO server that must align with the data definition.

FIDO Authentication Data

The FIDO Authentication Data generated by the 3DS Requestor's FIDO server is a JSON object. The EMV 3DS specification defines that the `3DS Requestor Authentication Data` subfield must be a string, therefore the FIDO Authentication Information must be serialized using JSON and inserted as a string in the `3DS Requestor Authentication Information` JSON object of the EMV 3DS Authentication Request message.

The FIDO Authentication Data object has the following members:

Key	Type	Description
`authTime`	string (required)	ISO 8601 encoded time of when the FIDO Authentication was performed.
`FIDOAuthenticatorReferences`	array of `FIDOAuthenticatorReference` objects (required)	The user's authenticators registered with this merchant. See member data fields in the table below.
`rpId`	string (present if, and only if, `appId` is not)	RP ID for U2F or FIDO2 authenticators
`appId`	string (present if, and only if, `rpId` is not)	AppID for UAF authenticators

The `FIDOAuthenticatorReferences` object, used above, has the following members:

Key	Type	Description
`publicKey`	string (required)	Base64url encoding of the public key raw format (e.g. ALG_KEY_ECC_X962_RAW or ALG_KEY_RSA_2048_RAW, see https://fidoalliance.org/specs/common-specs/fido-registry-v2.1-ps-20191217.html#public-key-representation-formats) related to the FIDO credential registered on this authenticator for this merchant.
`aaguid`	string (present if, and only if, `aaid` is not)	The AAGUID of the FIDO2 authenticator (or the encoding of 16 zero bytes if not available). In the case of a U2F authenticator it will be the Base64 encoded serialized JSON encoded attestationCertificateKeyIdentifiers list (see FIDO Metadata Statements).
`aaid`	string (present if, and only if, `aaguid` is not)	The AAID of the UAF authenticator.
`usedForThisTransaction`	boolean (required)	True if this authenticator was used to authenticate the current session at the merchant.
`up`	boolean (never present if `usedForThisTransaction` is false)	("User Presence".) True if the authentication involved some affirmative user gesture (such as touching the authenticator)
`uv`	boolean (never present if `usedForThisTransaction` is false)	("User Verification".) True if the authenticator collected proof that an authorized human was present during the authentication, e.g. via a PIN or biometric.

The fields `publicKey`, `aaguid`, `aaid` shall be treated as blobs, i.e. no specific assumptions on a character subset (to UTF8) and on the internal structure shall be made.

FIDO Authentication Data Example

The following is an example of FIDO Authentication Data. It is based on a user having a single authenticator of model "0132d110-bf4e-4208-a403-ab4f5f12efe5" that was used for this transaction. User presence check "up" and user verification "uv" have been performed. The user authenticated to "merchant.com".


```
{
    "rpId": "https://merchant.com",
    "authTime": "2020-08-08T07:42:17Z",
    "FIDOAuthenticatorReferences": [
        {
            "publicKey":
"BJDLvkKkuXRpn3bWh_hiJQb8lJNd9kTXEmQgEwoHezkNI_VPGd3vrjrWyZTfgxVDl9_TpixrVjmEAEmegmfL2WI",
            "aaguid": "0132d110-bf4e-4208-a403-ab4f5f12efe5",
            "uv": true,
            "up": true,
            "usedForThisTransaction": true
        }
    ]
}
```


Recommended Issuer ACS Processing of the FIDO Authentication Data

When an ACS first receives the FIDO Authentication Data via EMV 3DS Authentication Request, the issuer should:

1. Verify the cardholder via preferred method of authentication through an existing risk assessment or EMV 3DS challenge method (see the EMV 3DS Core specification for more details).
2. Check whether the `rpId/appID` identify an entity (merchant) is trusted for this transaction.
3. Check whether the properties of the authenticator that has `usedForThisTransaction` set are satisfactory, if specified by the `aaid` member, or if the `aaguid` member is not sixteen zero bytes.
4. Ensure the ACS risk engine can leverage the newly received FIDO Authentication Data for subsequent transactions. Store the array of `FIDOAuthenticatorReference` objects, linked to the cardholder and merchant combination, minimally the `publicKey` and `aaguid` *or* `aaid` per authenticator. Additionally, the Issuer should maintain the identity verification status of the user using these authenticators.

Any subsequent transactions where the issuer ACS receives the FIDO Authentication Data for a transaction from a previously assessed account, the issuer ACS should:

1. Check whether the `rpId` or `appID` identified an entity trusted for this delegated authentication operation and matches to a known merchant ID.
2. Verify whether the `publicKey` of the authenticator included in the FIDO Authentication Data that has the flag `usedForThisTransaction` set to true is already known.
3. Check whether the authenticator model as indicated by `aaguid` and `aaid` of the authenticator in step #2 is acceptable by verifying:
    a. whether either entry `uv` is present and set to true.
    -or-
    b. FIDO Metadata for the given `aaguid` or `aaid` specifies that the authenticator *always* performs user verification.

4. Synchronize data as required for the issuer ACS risk engine; for example by performing the following actions:
    a. Remove all authenticators that have been remembered but are not included in the new FIDO Authentication Data (look for identical `publicKeys`).
    b. Add all authenticators that are included in the new FIDO Authentication Data but are not present in the risk model data store (look for identical `publicKeys`).

FIDO Alliance Metadata Service (MDS)

The issuer ACS may choose to make use of the FIDO Metadata Service to verify the authenticator data that is received in the FIDO Authentication Data via EMV 3DS. For example, the issuer could use this data to verify that the authenticator used for cardholder authentication has been tested and is compliant with FIDO testing standards. When `uv` and `up` flags are not present, the issuer could use this data to check whether FIDO Metadata for the given `aaguid/aaid` specifies that the authenticator always performs user verification.

Privacy Considerations

The FIDO Authentication Data members `rpId` and `appId` identify the merchant. As part of the EMV 3DS transaction data, the issuer can identify the merchant with existing fields, thus disclosing no new information. The fields `aaguid` and `aaid` represent the authenticator make and model. Since the FIDO Alliance requires a large number of authenticators of the same make and model, this information is not personal identifiable information (PII). It is important to note that the sum of this information may be greater than its parts. For example, while a specific *aaguid* is not PII, the set of the different authenticators that a person uses, and the times of day when they switch between them, for example, may be uniquely identifying. These fields are included to provide evidence to the issuer that the legitimate user submitted the payment credentials – which is the purpose of this approach.

See the FIDO Alliance Privacy Principles Whitepaper for more information on the privacy aspect of FIDO Authentication.

FIDO Server Support of EMV 3DS Data

For FIDO servers, the ability to support the EMV 3DS FIDO Authentication Data is considered optional and is generally only relevant for relying parties that also participate in the EMV 3DS ecosystem. The technical details for extracting and sending the specified FIDO Authentication Data from a FIDO server implementation are not in scope for the FIDO specification itself. Each participating FIDO server must comply with the definition and format as described in this document. Any incorrect implementation of the EMV 3DS FIDO Authentication Data will cause inconsistency and could impact the performance of the EMV 3DS transactions and have inconsistent outcomes.

The data sourced from a FIDO server and shared to the 3DS Requestor/3DS Server is handled within the 3DS Requestor environment as defined by the EMV 3DS specifications. Handling of this data shall meet the data handling criteria of the EMV 3DS specification and as defined by EMVCo.

Conclusion

EMV 3-D Secure can provide higher levels of security and better overall approval rates for ecommerce merchants by providing relevant FIDO Authentication data to payment card issuing banks at the time of the EMV 3DS transaction. FIDO Authentication, when completed by a consumer during an ecommerce checkout, provides a high level of security as well as an excellent user experience. Through data sharing with EMV 3DS messaging, the FIDO Authentication Data can positively influence overall risk assessment by a card issuing bank and provide consistent frictionless authentication when leveraged by an ACS. This additional FIDO Authentication Information provides evidence of cardholder presence and may reduce issuer step-up authentication being required to complete a purchase.

While this document covers one method for data sharing of FIDO Authentication Data via EMV 3DS for transaction risk assessment, FIDO and other standards organizations such as EMVCo and W3C may continue to contribute additional methods of authentication for payment transactions that may become available in future publications.