FIDO Alliance White Paper:

Multiple Authenticators for Reducing Account-Recovery Needs for FIDO-Enabled Consumer Accounts

June 2020

Editors:
Wataru Oogami, Yahoo! JAPAN
Max Hata, NTT DOCOMO

Introduction

When a service using FIDO authentication is deployed, a secure account recovery process must be implemented to address lost, damaged, or stolen FIDO authenticators. FIDO's Recommended Account Recovery Practices for FIDO Relying Parties white paper [1] recommends two strategies:

- **Strategy-1: Multiple authenticators per account (reduction of account-recovery needs)**
- **Strategy-2: Re-run identity proofing or user onboarding mechanisms (actual execution of account recovery)**

The paper recommends adopting Strategy-1, then utilizing Strategy-2 for cases where Strategy-1 cannot be used. Examples of Strategy-1 failure include:

- User never bothered to set up a second authenticator.
- User's primary and backup authenticators were lost at the same time, e.g. "My backpack with my phone and computer were stolen."
- A primary authenticator has failed on a business trip and the backup is at home.

Strategy-1 plays a very important role for FIDO-enabled consumer-facing accounts, where authentication like passwords is disabled after FIDO credentials are registered, or where passwords and FIDO credentials are registered for two-step authentication. These are typical deployment models where a FIDO credential is registered after authenticating with a password for an existing account. This relieves users from annoying password problems and enables them to enjoy FIDO's strong authentication while protecting their online account from password testing attacks. Without multiple authenticators, however, users are taken to Strategy-2 if an authenticator is lost. Strategy-2 will put the users through an onerous experience compared to the routine use of FIDO.

Hence there are use cases where multiple authenticators are critical, i.e., where passwords are disabled. This paper assumes that the first authenticator is already registered (how this can be done is out of the scope of this paper) and focuses on how to enable additional authenticators.

Eventually it is expected that passwords will be completely replaced by FIDO, but in the near term, passwords may be kept active in parallel with FIDO credentials. In this model, Strategy-1 may not be as important in terms of account recovery since users can authenticate themselves using passwords and register a new credential even if FIDO authenticators are lost. Having multiple authenticators, of course, is convenient, but this model still suffers from password problems since passwords are still usable. In addition, FIDO may make the "forgot password" problem more frequent because a user that consistently uses FIDO will likely have more trouble remembering a rarely used password.

In enterprise use cases, multiple authenticators are also very useful and recommended. However, under the governance of enterprises, there are additional ways to deal with account recovery problems, e.g., in-house help desk, HR departments, and this paper does not discuss these use cases. Of course, these additional approaches may require increased operational costs and overhead. In consumer-facing use cases, the fact that the number of account recovery options can be limited makes multiple authenticators very important for providing simple and secure recovery.

As such, Strategy-1, multiple authenticators, is extremely important for FIDO-enabled consumer accounts. This paper focuses on these use cases by providing guidance on how to deploy Strategy-1. It discusses how to register new authenticators bound to an already-registered authenticator, security considerations, coverage/authenticator options, usability, and policy, based on state-of-the-art FIDO-enabled browsers and platforms. It provides recommendations for registration methods and policy examples for deploying the solution.

Contents

Tables

Figures

1. What RPs Should Consider

We assume that the first FIDO authenticator has already been registered ("already-registered" authenticator). We now consider how to register an additional authenticator using the FIDO credential of the already-registered authenticator[1].

RPs that offer registration of multiple authenticators to users should evaluate the following issues:

- Security:
  - Adding multiple authenticators to an authentication system should not degrade the overall security. The total security of a system becomes the lowest level of any part of the system. Note that any FIDO authenticator will be stronger than passwords.
  - Each registration operation of an additional authenticator using an authentication session with a registered authenticator should be completed securely. RPs need a process that prevents attackers from hijacking a user's account during these registrations.
- Usability:
  - RPs should simplify the registration process for additional authenticators to make the process easy for consumers and to reduce customer costs.
  - Users have multiple devices that can be used as FIDO2 authenticators. FIDO2 is built into smartphones, tablets, PCs, and security keys. This can provide a convenient way to introduce additional authenticators because users are already accustomed to these devices. Users can avoid the cost of acquiring additional authenticators by using one of the devices that they already own as an additional authenticator, e.g., smartphone or PC.
- Coverage: RPs need to provide FIDO2 authentication services for as many consumers as possible. RPs need to understand the state-of-the-art implementation, particularly stability, in this early phase of FIDO2 rollout.
- Policy: An RP needs to have a policy to support multiple authenticators that matches their security requirements and the needs of their users. Such a policy will likely differ from one RP to another. See section 3 for details.

2. Registration

2.1 Authenticator Variations

RPs may support their users' utilization of (i) platform authenticators and (ii) roaming authenticators for multiple authenticators.

Platform authenticator

Platform authenticator is a FIDO authenticator that is implemented on fixed to the device. For example, Windows 10 and Android 7.0 and above provide platform authenticators that cannot be removed from the device.

[1] The assurance of authenticating with FIDO credentials themselves is high. But the overall assurance of the first FIDO credential itself is subject to the weakest link security design principle, as determined by (i) the assurance of the identity proofing or the user onboarding mechanism that the RP has used to create the user account, and (ii) the assurance of the binding of the FIDO credential and the account. The assurance of additional credentials to be created is equal to or less than that of the first one.

Roaming authenticator

Roaming authenticator is a FIDO authenticator that is implemented off-device and can be accessed over a transport such as Universal Serial Bus (USB), Bluetooth Low Energy (BLE), or Near Field Communications (NFC). Roaming authenticators can be used across multiple devices. For example, a USB authenticator can be used on an office PC and a home PC.

2.2 Registration Overview

A multiple-authenticator solution requires registering an additional authenticator using an authenticator that has already been registered.

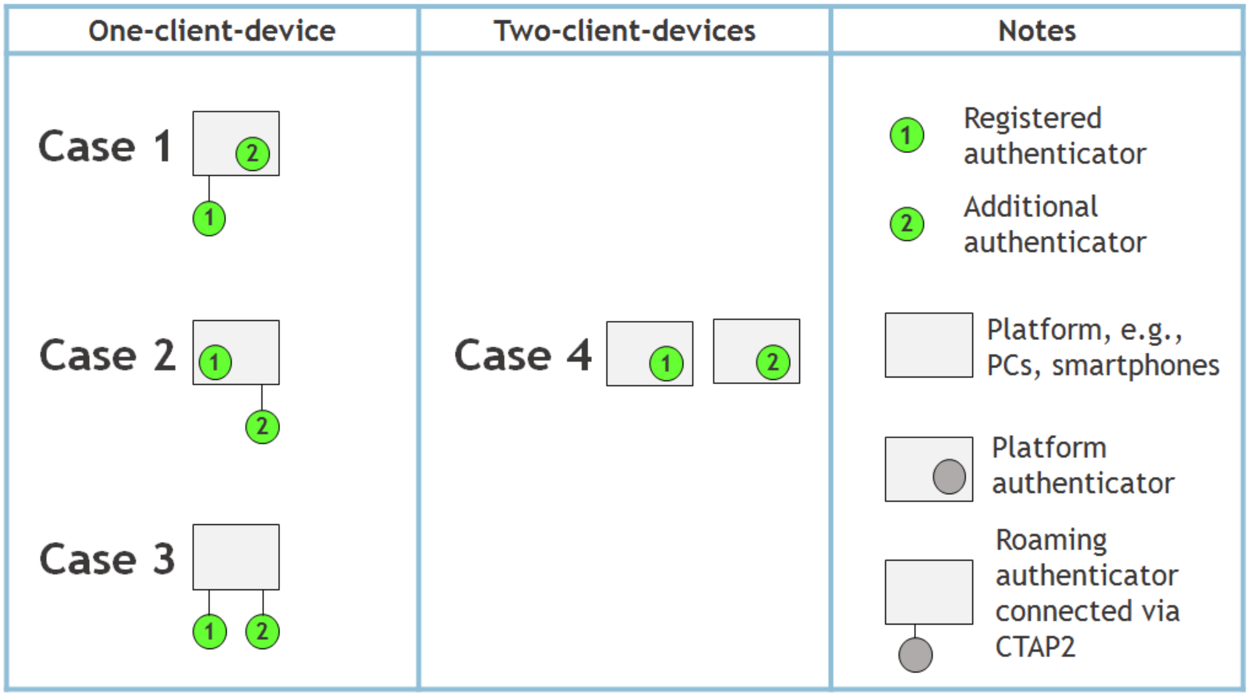
After registration, multiple authenticators and their respective credentials must meet the following requirements:

- The credential created by this registration operation is newly created and the server binds it with the already-registered credential.
- The new credential can be used for authentication on the same account based on the trust from the already-registered credential.
- The security and usability requirements outlined in section 1 should be met regardless of the authenticator option.

The user needs one or two devices that run clients to register additional authenticators. The use cases of authenticator and client device combinations are shown in Table 1. Cases 1, 2, and 3 represent one-client-device use cases and Case 4 represents a two-client-device use case.

For example, Case 1 shows a case where the platform authenticator of a PC or a smartphone is registered using an already-registered roaming authenticator connected via CTAP2 (FIDO Client to Authenticator Protocol) [4]. Case 4 shows a two-client-device use case where a platform authenticator in a PC or smartphone is registered by using another already-registered platform authenticator in another PC or smartphone.

Table 1: Client device & authenticator use cases for registration



2.3 Registering a New Authenticator and its Security

One-client-device use cases

Registration of an additional authenticator can be securely completed for all the one-client-device use cases, as shown in Figure 1.

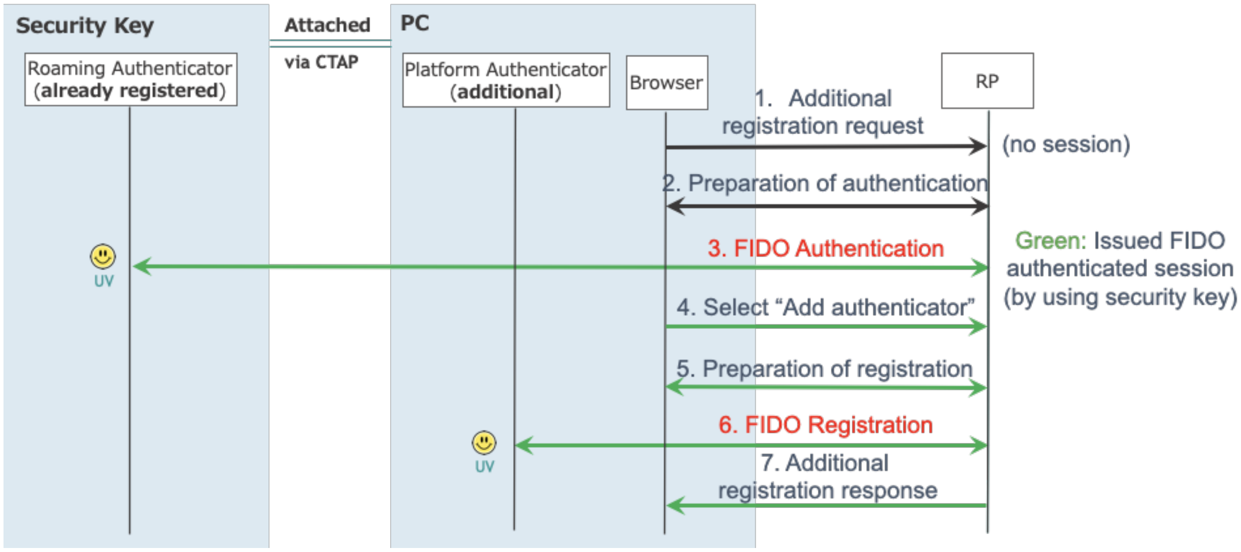

Security Key
Roaming Authenticator (already registered)
Attached
via CTAP
PC
Platform Authenticator (additional)
Browser
RP
1. Additional registration request
(no session)
2. Preparation of authentication
3. FIDO Authentication
UV
Green: Issued FIDO authenticated session (by using security key)
4. Select "Add authenticator"
5. Preparation of registration
6. FIDO Registration
UV
7. Additional registration response

Figure 1: Additional registration operation in one-client-device use cases

The overall procedure consists of usual FIDO authentication (1 to 3) followed by registration (4 to 7). These steps connect the authenticator to the RP account. Note that the RP should register an additional authenticator by using the same session created in step 3.

If the RP has developed a trust score for the already-registered authenticator based on its history or other factors, the score could be applied to the new authenticator. This transfer of trust allows the new authenticator to be treated with the same confidence as the already-registered authenticator.

Two-client-devices use cases

Two-client-device use cases often become necessary when registering a FIDO platform authenticator using an already-registered FIDO platform authenticator e.g., registering a PC's platform authenticator using the credential that is already registered in the platform authenticator of a smartphone. The three available two-client-device methods are shown in Figure 2.

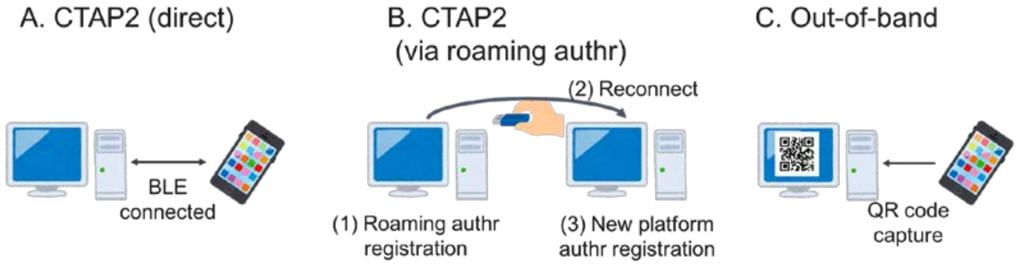

A. CTAP2 (direct)
BLE connected
B. CTAP2 (via roaming authr)
(2) Reconnect
(1) Roaming authr registration
(3) New platform authr registration
C. Out-of-band
QR code capture

Figure 2: Examples of methods addressing two-client-device use cases

A. CTAP2 (direct):

This method can be implemented based on CTAP2 using a transport such as USB, BLE or NFC. It must be properly and securely implemented. In order to enable this method, you need to implement CTAP2 specification on your PCs or smartphones integrating with the platform authenticators. If the two platform authenticators in case 4 are connected via CTAP2, registration can be achieved as in case 1 or 2.

Since this approach is not widely available yet, RPs may need to consider the solutions outlined in (B) or (C) below.

In order to ensure security and interoperability, it is recommended to use only CTAP2 authenticators certified by FIDO Alliance's Authenticator Security Certification [7].

B. CTAP2 (via roaming authenticator):

This method is secure based on CTAP2. (i) The user connects and registers a roaming authenticator with the already-registered platform authenticator. (ii) The user reconnects the registered authenticator to the device of the authenticator the user wants to register. (iii) The user bootstraps the new platform authenticator with the connected already-registered roaming authenticator. This method is secure as it is the same as the one-client-device use cases, but the user needs a roaming authenticator.

C. Out-of-band (e.g., QR code):

This method is generally not secure. It is not recommended because it may be vulnerable to man-in-the-middle (MITM) attacks. However, careful design and operation, where no other solutions are available, may provide adequate security depending on the types of services. For example, on a PC, a single use QR code is displayed by the RP containing a token with a very short lifetime. This QR code is scanned by an RP's application on a smartphone with an integrated QR code reader. The QR code should not be read by open-market QR code scanners as they may open MITM attacks. This integration will mitigate the risk of MITM attacks and provide reasonable security and usability for many practical applications. RPs should carefully evaluate the risks if they wish to deploy this type of solution.

Security evaluation

Table 2 summarizes the security for the use cases that are outlined in Table 1.

Table 2: Security evaluations of use cases

Number of client devices	Security		Depends only on FIDO protocols?	Examples
1	**Secure.**		**yes**	Registration using an already-registered authenticator.
2	A) CTAP2 (direct)	**Secure**. One device needs to connect to the other via CTAP2. NB: Security may be subject to the implementation.	**yes**	Google's implementation on Android [2].
	B) CTAP2 (via roaming authenticator)	**Secure.** Another roaming authenticator is required.	**yes**	Use of a roaming authenticator.
	C) Out-of-band	**Generally, NOT secure.** RPs need to prevent MITM attacks.	**no**	QR code.

3. Policies

Policies vary among RPs. Generally, policies depend on (i) the levels of security for protecting assets that the RP services manage and (ii) the convenience that RPs want to offer to their users. This principle also applies to the selection of policies for multiple authenticators.

Although any type of multiple authenticator can be registered, RPs may want to register only those that meet their requirements. RPs can reflect such policies in the registration flow. Figure 3 shows the generic points at which RPs may control authenticators based on their policies in the FIDO flow. Preliminary information at control point 1 includes FIDO-specific information (e.g., *isUserVerifyingPlatformAuthenticatorAvailable()*) and a user's local information (e.g., a browser's cookies).

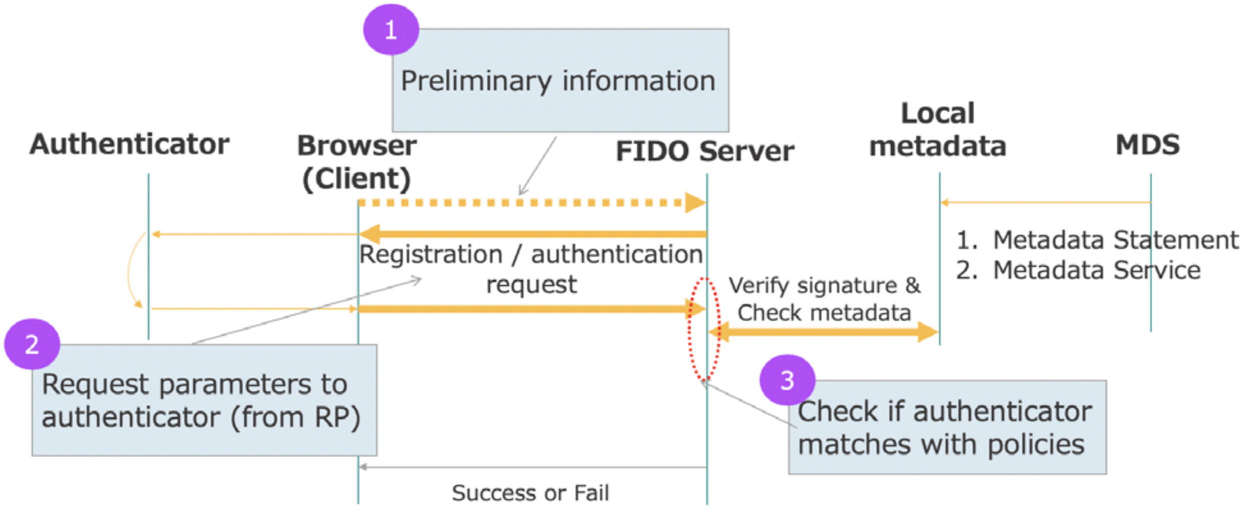

1
Preliminary information
Authenticator
Browser (Client)
FIDO Server
Local metadata
MDS
1. Metadata Statement
2. Metadata Service
Registration / authentication request
Verify signature & Check metadata
2
Request parameters to authenticator (from RP)
3
Check if authenticator matches with policies
Success or Fail

Figure 3: Control points of RP's policy

3.1 Typical Policy Examples

This section describes policy examples that RPs can deploy today using WebAuthn Specs [3] and the current implementations.

Policy #1: Pre-approved authenticator list

Security-conscious RPs, e.g., those managing high value assets for users, may use this policy to control the authenticators to be accepted. RPs need to scrutinize candidate authenticators in advance to create and keep updating a list of acceptable authenticator's models. RPs can compare the AAGUID (Authenticator Attestation Globally Unique IDentifier) of an authenticator and the list at control point 3 to determine if the authenticator can be accepted or not. The advantage is that this policy is deployable today on major browsers and platforms. The disadvantages are that (i) users will know their authenticators cannot be accepted only after they have provided User Verification (UV) gestures, and (ii) RPs need to keep updating the list manually. As this model requires manual updates, it may not scale. In order to improve user experience, RPs should present a link to a list of acceptable authenticators for the user in advance. Alternatively, RPs can have a list of authenticators that are not suitable.

Example: RPs specify *"AttestationConveyancePreference"* = *"direct"* at control point 2 to get AAGUID.

Policy #2: Accept any authenticator

This is the simplest of all policies. This policy will let RPs accept any FIDO authenticators that users attempt to register. It leverages the promise that any FIDO authenticators are better than passwords by design.

Example 1: At control point 2, RPs specify the following policy: *"AttestationConveyancePreference"* = *"none."* This means the RP does not receive any attestation from any authenticator.

Example 2: RPs may want to specify *"AttestationConveyancePreference"* = *"indirect"* or *"direct"*, to get the AAGUID. The AAGUID is not for selecting authenticators but for future reference that RPs want to record in case any trouble arises with the authenticator model.

Policy #3: Check User Presence (UP) / User Verification (UV)

This policy may not be applicable for two-step verification (2SV) where User Presence (UP) is the main user authorization. A typical passwordless account uses two-factor authentication (2FA). Confirmation of only User Presence decreases authentication security if User Verification is required. High value services should consider these risks because confirming only User Presence makes friendly fraud easier.

Example: At control point 2, if the RP specifies "*UserVerificationRequirement*" = "*required*," the authenticator should execute authentication with UV (e.g., fingerprint matching or client PIN). Also, at control point 3, RPs need to verify the authenticator's response. RPs can verify this by whether the "*UV*" flag in "*AuthenticatiorAttestationResponse.authenticatorData*" or in "*AuthenticatorAssertionResponse.authenticatorData*" is "*1*" (which means the user is verified).

Note: Since "*AuthenticatorAttestationResponse.authenticatorData*" is a response that is returned only if attestation is requested, RPs must set "*AttestationConveyancePreference*" = "*direct*" (which means request appropriate attestation) at control point 2 if RPs adopt this policy. In addition, this enables RPs (i) to verify the attestation signature using the public keys and (ii) to check the metadata of the authenticator from the AAGUID if such metadata is available (offline or FIDO MDS (Metadata Service)).

Policy #4: Require resident credentials

WebAuthn/FIDO2 defines two types of credentials: (i) resident credentials and (ii) non-resident credentials. Both are optional and some authenticators support both types, but others support only one type. From the perspective of user experience (UX) for authentication, resident credentials offer an additional UX, where a user does not have to provide their username when requesting authentication. If an RP supports this UX, the authenticator to be accepted should be supporting resident credentials.

An RP can only learn a generated credential is a resident credential if and only if the RP explicitly sets the "requireResidentKey" option as "true" at registration time. If an RP does not so specify, the RP does not have any way to know the type of credential that is generated. It is up to the authenticator which type of credentials to generate if the option is not set to true. Thus, it is recommended to set "*requireResidentKey*" = "*true*" if this policy is adopted.

Policy #5: Check certification levels

RPs may want to check the security level of an authenticator to confirm whether the authenticator meets their security requirements. FIDO Alliance has a program to certify the security levels of FIDO authenticators. The FIDO authenticator certification levels are good indicators for RPs to know the levels of the authenticators' security. Information on the FIDO authenticator certification levels is available from FIDO Alliance's Website [6] for offline uses or FIDO MDS [5] for automatic uses. RPs need to know the authenticator's AAGUID for an authenticator that is requesting registration to check its FIDO certification levels as well as other metadata.

Example: AAGUID is included in "*AuthenticatorAttestationResponse.authenticatorData*." Thus, RPs should specify "*AttestationConveyancePreference*" = "*direct*" at control point 2. In addition, RPs should acquire and retain the latest metadata from offline sources or FIDO MDS. At control point 3, RPs should verify the authenticator's attestation, as well as checking the certification level corresponding to the AAGUID.

Policy #6: Time Window

An attacker with physical access to a user's system could simply attach a new roaming authenticator to a user's account when they leave their PC unattended for a few minutes. RPs should limit the time between the use of an already-registered authenticator and the connection of a new authenticator to 60 seconds or less. The user should be asked to re-authenticate with the already-registered authenticator if the window has expired. This ensures explicit confirmation by the legitimate user that they are connecting a new authenticator rather than a well-timed imposter.

Advanced Policies

There are a variety of policies depending on the needs of each RP. As we gain more experience, we will publish more details about best practices. The following are examples of features that may be applicable for each control point.

Control point 1: RPs may want to execute "*isUserVerifyingPlatformAuthenticatorAvailable()*" to check if a platform authenticator is available.

Control point 2: RPs may want to accept only external authenticators or platform authenticators. They should specify *AuthenticatorAttachment* to indicate this requirement.

Control point 3:

- RPs may want to confirm the authenticator's modalities (e.g., fingerprint or PINs) that were used for user verification. They should request "*uvm*" (UVM Extension) = "*true*" at control point 2 to receive a report on the modality.
  Note: Support of UVM extension by major browsers / platforms is currently very limited.
- Various metadata for FIDO authenticators is defined in addition to the certification levels. RPs may want to scrutinize the metadata to determine certain criteria for acceptance.

4. Recommendations

We strongly recommend that RPs support features to enable their users to utilize multiple authenticators per account to reduce account-recovery needs.

Registration

We recommend the following registrations:

- "one-client-device"

  This method is secure because it can be implemented within one client device and only with FIDO protocols.

- "two-client-devices"

  The method called CTAP2 (direct or via roaming authenticator) in Table 2. These methods are secure because they are deployed only with FIDO protocols, while enabling registration of a client device from a separate client device.

The out-of-band for two-client-devices, e.g., a QR code, is not generally recommended. Nonetheless, if RPs wish to deploy this type of solution for a specific situation where no other solutions are applicable, RPs should carefully evaluate risks such as MITM attacks.

Policies

The typical policies described in section 3 are a good starting point for RPs, but RPs should augment and customize these policies for their specific needs and users.

5. Glossary of Terms

2FA	Two-factor authentication
2SV	Two-step verification
AAGUID	Authenticator Attestation Globally Unique IDentifier
BLE	Bluetooth Low Energy
CTAP	Client to Authenticator Protocol
MDS	Metadata Service
MITM	Man-in-the-middle
NFC	Near Field Communications
RP	Relying Party
UP	User Presence
USB	Universal Serial Bus
UV	User Verification
UVM	User Verification Method
UX	User experience

6. Acknowledgments

The authors acknowledge the following people (in alphabetical order) for their valuable feedback and comments:

- Jason Burnett, Raonsecure
- Norio Fujita, Soft Giken
- Hidehito Gomi, Yahoo Japan Corporation
- Jeff Hodges, Google
- Bill Leddy, Visa
- Giridhar Mandyam, Qualcomm
- Aiki Matsushita, DDS
- Vy Phạm Hoàng Trúc, Soft Giken
- Dario Salice, Facebook
- Dean H. Saxe, Amazon
- David Treece, Yubico
- Shane Weeden, IBM

7. References

[1] Recommended Account Recovery Practices for FIDO Relying Parties, February 2019, https://fidoalliance.org/recommended-account-recovery-practices/

[2] Android phones transformed into anti-phishing security tokens, April 2019, https://nakedsecurity.sophos.com/2019/04/12/android-phones-transformed-into-anti-phishing-security-tokens/

[3] Web Authentication: An API for Accessing Public Key Credentials Level 1, W3C Recommendation, March 2019, https://www.w3.org/TR/webauthn/

[4] Client to Authenticator Protocol (CTAP), Proposed Standard, January 2019, https://fidoalliance.org/specs/fido-v2.0-ps-20190130/fido-client-to-authenticator-protocol-v2.0-ps-20190130.html

[5] Metadata Service, FIDO Alliance, https://fidoalliance.org/metadata/

[6] FIDO Certified Products, https://fidoalliance.org/certification/fido-certified-products/

[7] Authenticator Certification, https://fidoalliance.org/certification/functional-certification/

Note: URL references to specific definitions can also be embedded in the document. Usually the cross-reference or link to a document or definition is inserted at the first occurrence of the term.