Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Defence, Civil Protection and Sport DDPS

National Cyber Security Centre NCSC

September 2025

Technology brief

Passkeys

1 Introduction

Passwords remain the most widely used form of authentication, despite their many well-known and well-documented security flaws. None of the alternative authentication methods proposed in the past have gained broad acceptance. By contrast, the FIDO2 standard and its passkey implementation are seeing growing support. This document provides an overview of the FIDO2 standard and passkeys, and discusses their security properties and practical implications.

2 FIDO2 standard

The Fast Identity Online (FIDO) Alliance is an industry consortium whose partners work together to develop and standardise passwordless authentication methods as an alternative to traditional passwords.[1] Based on the earlier FIDO Universal Second Factor (U2F) and FIDO Universal Authentication Framework (UAF) standards, the FIDO Alliance has developed the Client to Authenticator Protocol version 2 (CTAP2). Alongside the Web Authentication (WebAuthn) protocol from the World Wide Web Consortium (W3C), CTAP2 forms the core of the FIDO2 standard.[2]

Figure 1 illustrates the authentication process according to the FIDO2 standard, and the interaction between the WebAuthn and CTAP2 protocols. Essentially, FIDO2 allows users to authenticate to a server via any client device. Rather than using a password, authentication relies on a dedicated hardware component called an 'authenticator', which can be either built into the client device or connected externally (typically in the form of a token). External authenticators can be connected via a cable – for example, through a USB interface – or wirelessly, using Bluetooth or Near Field Communication (NFC). In all cases, authentication actually takes place between the authenticator and the server. The client merely acts as a passive intermediary, forwarding the messages exchanged between the authenticator and the server. Figure 1 shows the WebAuthn protocol in blue, which defines the exchange of

[1] https://fidoalliance.org
[2] https://www.w3.org

messages between the client and the server. The green area represents the CTAP2 protocol, which defines communication between the client and the authenticator.

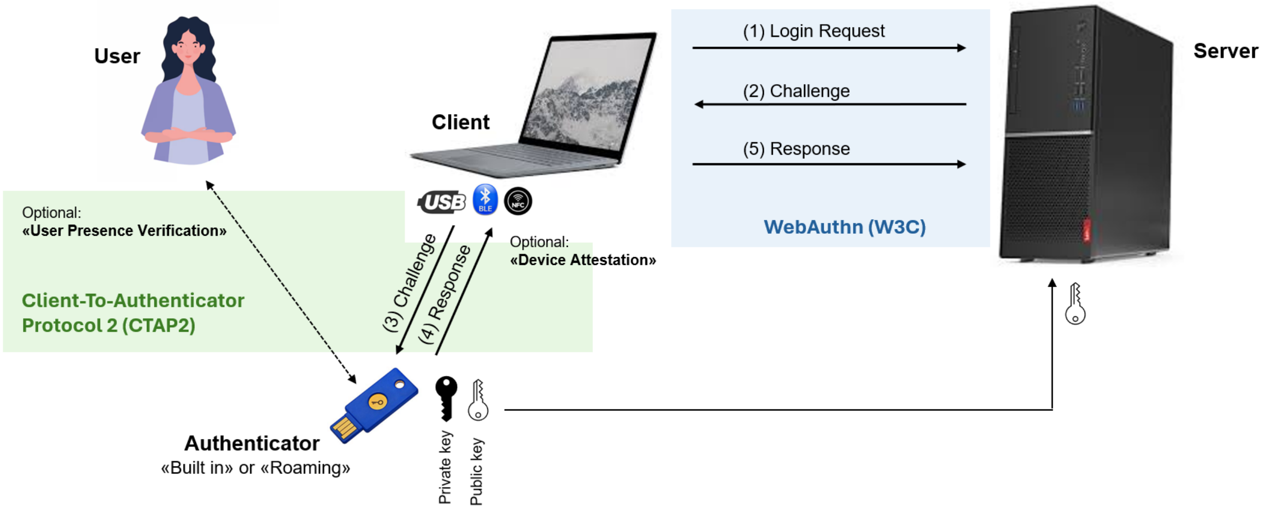

Figure 1: Authentication according to the FIDO2 standard

Before an authenticator can be used to verify a user's identity for a server identified by a URL, it must first be initialised. During this process, the authenticator generates a random asymmetric key pair (that consists of a private and a respective public key) for the user and associates it with the server's URL. The private key is stored on the authenticator and used to create digital signatures, while the public key is shared with the server and used to verify those signatures.

Once initialisation is complete, the user can authenticate to the server using a simple challenge–response protocol consisting of five steps. In step (1), the client sends a login request to the server. In step (2), the server sends the client a challenge – a randomly generated piece of data that must be digitally signed with the private key to prove possession of that key and produce a valid response. This signature can only be created by the authenticator using the private key. In step (3), the client forwards the challenge to the authenticator, and in step (4), the authenticator responds with the corresponding signature. Finally, in step (5), the client sends the response back to the server. Using the public key, the server can verify that the response is valid. A valid response is confirmation of the user's authenticity.

As the server's URL is included in the challenge and verified by the authenticator prior to signature generation, the response can only be used to authenticate to the intended server. This means that the WebAuthn protocol effectively protects users from phishing and Mallory-in-the-middle (also called man-in-the-middle, MITM) attacks.[3] During the protocol run, the client can verify the authenticity of the device (device attestation), and the device can verify the user's presence and consent (user presence verification). These checks are optional and may be implemented differently by different manufacturers or software providers. In the simplest and most common setup, device attestation ensures that a specific token is used

[3] In an MITM attack, the attacker positions themselves between the client and the server. Both the client and the server communicate with the MITM, but believe that they are communicating directly with each other. As an alternative to, or in addition to, the security mechanism that relies on embedding the target server's URL, the Token Binding protocol can be used. Defined as a Transport Layer Security (TLS) extension in RFC 8471 and 8472, it is not widely supported by standard web browsers.

and user presence verification is achieved by requiring the user to press a button on the token or client device.

3 Passkeys

The original aim of the FIDO2 standard was to securely store a user's private key in an unreadable format inside an authenticator for a specific application. This means that the user's identity is bound to that authenticator; this is known as a device-bound authenticator. While this is desirable from a security perspective, it causes difficulties in practice when several different authenticators are used, especially if they are built into client devices. It would be easier if users' authenticators could synchronise the private keys they store and manage, for example via cloud storage, resulting in synced authenticators. This would make authentication less dependent on individual devices. The FIDO Alliance has developed technology for this purpose, known as passkeys. Currently, the term 'passkey' refers to both the technology and the private keys that can be used for authentication instead of passwords.

Passkeys can be implemented in various ways, offering considerable flexibility, although this usually comes at the expense of security. As previously mentioned, a user's passkeys can be synchronised across devices. Some systems also support the Credential Exchange Format (CXF) and the corresponding Credential Exchange Protocol (CXP).[4] This enables passkeys to be exported individually and imported onto other devices, and in some cases even transferred between users. Cross-Device Authentication (CDA) enables devices that do not support passkeys themselves to be integrated into the passkey ecosystem when using authenticators that are physically nearby. Simplifications are also possible in relation to the hardware protection originally envisaged by FIDO2. For example, password managers can manage passkeys even without dedicated hardware protection, safeguarding the stored passkeys in software instead. Therefore, for any concrete implementation of passkeys, it will be crucial to examine how it departs from the original FIDO2 standard to prioritise user convenience, and what security implications this has. In principle, the FIDO Alliance defines three security levels for authenticators and their Restricted Operating Environment (ROE).[5] However, these three levels are not sufficiently granular for effective security analysis and design.

4 Assessment

From a security perspective, the FIDO2 standard is a positive development. Rather than relying on passwords, it uses modern cryptographic methods that reduce the burden on users. However, the originally intended hardware protection – specifically, preventing the corresponding key material from being read – has been relaxed to enable broader market adoption of passkeys. As a result, passkeys now offer greater flexibility and can be used across a wider range of applications.

---

[4] https://fidoalliance.org/specs/cx/

[5] According to https://fidoalliance.org/certification/authenticator-certification-levels/ the first level (L1) allows the use of any hardware and software components. Level 2 (L2) requires a Restricted Operating Environment (ROE) in the form of a Secure Element (SE) or Trusted Execution Environment (TEE), and Level 3 (L3) requires an Allowed ROE (AROE) approved by the FIDO Alliance.

From a security standpoint, the details matter – and since these depend on the specific implementation, the security of passkeys can vary in practice. Nevertheless, passkeys are more secure than passwords for at least three reasons:

(1) Unlike passwords, passkeys are unique to each target server and would therefore have to be obtained separately by an attacker.[6]

(2) Unlike passwords, passkeys are generated pseudo-randomly and cannot easily be guessed.

(3) Passkeys are not transmitted over data communication channels, so they cannot be intercepted from network traffic or reused in replay attacks.

Unfortunately, in many cases, passkeys will still allow or even require the continued use of passwords (for example, as a backup option for password resets). Consequently, weak passwords and poor password management practices will continue to be attack vectors. This must be taken into account when assessing security mechanisms that build on passkeys.

The National Cyber Security Centre (NCSC) expects the FIDO2 standard, in the form of various passkey implementations, to become established in the market, given its security advantages and strong industry support. Major technology providers such as Google, Apple and Microsoft are already moving in this direction, replacing passwords with passkeys wherever possible. This paves the way for passwordless authentication, a development that the NCSC welcomes and supports.

[6] By contrast, passwords are designed – and typically used – in such a way that the same password is often used for multiple servers, or that different passwords differ only slightly from one another.

Abbreviations

AROE	Allowed ROE
CDA	Cross Device Authentication
CTAP	Client to Authenticator Protocol
CXF	Credential Exchange Format
CXP	Credential Exchange Protocol
FIDO	Fast IDentity Online
MITM	Mallory-in-the-Middle (man-in-the-middle).
NCSC	National Cyber Security Centre
NFC	Near Field Communication
RFC	Request for Comments
ROE	Restricted Operating Environment
SE	Secure Element
TEE	Trusted Execution Environment
TLS	Transport Layer Security
U2F	Universal Second Factor
UAF	Universal Authentication Framework
URL	Uniform Resource Locator
USB	Universal Serial Bus
W3C	World Wide Web Consortium
WebAuthn	Web Authentication