yubico

WEBAUTHN WHITE PAPER SERIES: MAY 2020

The WebAuthn Standard: Why It Should Matter to the Public Sector and How It Works

Executive Summary

This paper is the second in a series of WebAuthn whitepapers published by Yubico. For an introduction to WebAuthn and why it is both more secure and easier to use, see the first paper, *Introducing WebAuthn: Enabling a Streamlined and More Secure User Authentication Experience.*

Most websites, services, and applications have difficulty providing secure, convenient authentication for users. Passwords are the problem. They tend to be either so simple they are easily guessed by hackers or so complex they are hard for users to remember. And all passwords, regardless of their complexity, are vulnerable to phishing and data breaches.

Fortunately, WebAuthn, a new web authentication standard approved in March 2019 by the World Wide Web Consortium (W3C), makes it easy for websites, services, and applications to offer strong authentication without relying on passwords. By replacing passwords with strong authentication based on public key cryptography, in which the private key never leaves the user's device, WebAuthn makes authentication both easier to use and more secure, benefitting users and service providers alike. The WebAuthn standard is already supported by all major browsers and most platforms including:

- Windows 10
- Android
- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Apple Safari
- Apple iOS

WebAuthn supports various models for account authentication, leveraging both external roaming authenticators, such as hardware security keys, and authenticators built into computing and mobile devices, such as fingerprint readers and facial recognition technology. Applications and web services can choose to implement WebAuthn for passwordless authentication, two-factor authentication (2FA), and multi-factor authentication (MFA). WebAuthn also supports step-up authentication, such as when a website, service, or application asks users for an additional factor before performing a high-value or high-risk transaction.

The WebAuthn standard makes use of these key terms:

- A **relying party** is a website, service, or application that wants to authenticate users.
- A **client/platform** is the underlying platform upon which the applicantion is running, e.g. a browser or operating system.
- An **authenticator** is the device that accepts the user's input to authenticate the user's credential for the relying party.

WebAuthn supports a wide range of authenticators, including external authenticators, such as hardware security keys, and internal authenticators such as built-in fingerprint readers, and built-in facial scan technology.

In a typical scenario, when a user wants to create an account for a relying party, she simply enters a username. In response, she is offered a choice of WebAuthn standard strong authentication methods including:

- Using an internal authenticator and doing one of the following:
  - Entering a PIN
  - Using a fingerprint scan
  - Using a facial scan
  - Using voice recognition
- Inserting and tapping a hardware security key

The best practice is to use a hardware security key when creating an account. Later, the user can optionally onboard the internal authenticator of a laptop, desktop, or some other device to also provide authentication credentials for the account.

The time has come for web and mobile app development teams to begin implementation of WebAuthn into upcoming software releases. Supporting WebAuthn promises to improve user experience, strengthen data security, and reduce costs for development, maintenance, and support.

Introduction

The use of passwords to access applications and services internally by employees and contractors, and externally by citizens puts public sector entities at risk of being hacked and their data and systems being compromised.

While Smart Cards are the de-facto authentication standard across the public sector, they aren't suitable for all use cases. Short term contract workers that aren't eligible for a Personal Identity Verification (PIV) credential or a Common Access Card (CAC) also require secure access to government services and systems. Some users may even require a token that works by itself without additional smart card hardware or software needs. Authenticating users in closed gap or isolated networks, authenticating first responders on the move, and even authenticating local constituents to customer-facing community services necessitates an authentication method that is stronger than the password.

Fortunately, the new web authentication standard WebAuthn, approved in March 2019 by the World Wide Web Consortium (W3C), makes it easy for websites, services, and applications to offer secure logins that are passwordless. By eliminating passwords, WebAuthn makes authentication both easier to use and more secure.

The WebAuthn standard opens the door to a new, streamlined, and more secure user experience, in which securely logging into websites, services, and apps takes just seconds from any device.

Problems with Web and Mobile Authentication Today

When users need to authenticate themselves to gain access to a website, SaaS application, or mobile app, the fast, easy world of internet computing hits a speed bump. Users are asked to enter complex passwords—often required to contain a mix of numbers, special characters, and upper and lowcase letters—before proceeding. In some cases, a two-factor authentication SMS code might be sent to the user's mobile device, requiring the user to also dutifully enter this code to proceed.

Why all this complexity? To protect online accounts from remote hacks, which continue to occur at an alarming rate, thanks to organized crime (responsible for about 39% of breaches) and nation states (about 23% of breaches[1]), sophisticated attack technologies, and lax defenses in cloud storage and IT.

The result: inadequate security and poor user experience.

The WebAuthn Standard

WebAuthn works in conjunction with the FIDO Client To Authenticator Protocol version 2 (CTAP2) to securely create and retrieve credentials on a security key. The two standards work in tandem making it easier for developers as they only need to concern themselves with interfacing with the WebAuthn specification.

Now websites, services, and applications have the opportunity to improve both data security and user experience, thanks to a new web authentication standard supported natively in all major browsers and most platforms.

The W3C, the primary standards body for the Web, has ratified the specification for WebAuthn, a new web authentication standard that strengthens the authentication of users to websites, services, and applications[2]. Significantly, WebAuthn takes web authentication beyond the limitations of passwords, improving data security while making user login faster and easier than ever before. All major browsers and most platforms are committed to supporting the WebAuthn standard.

[1] *2019 Verizon Data Breach Investigations Report,* https://enterprise.verizon.com/resources/reports/dbir/

[2] The W3C has published the protocol specifications here: https://www.w3.org/TR/webauthn/

The WebAuthn standard is already supported in:

- Windows 10
- Android
- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Apple Safari
- Apple iOS

Why WebAuthn Matters

For any website, service, or application that requires users to authenticate, the WebAuthn standard is important for several reasons:

- **Standardization**
  The approval of WebAuthn by the W3C enables the standardization of strong authentication across browsers and operating systems for the first time.
- **Improved Security**
  WebAuthn raises the bar for authentication, improving account security by enabling stronger authentication based on public key cryptography.
- **Streamlined User Experience**
  Web and mobile apps can now easily invoke strong authentication through the WebAuthn API, replacing the hassles of using passwords and SMS codes with the convenience of tapping a security key or using a fingerprint scan.
- **User Choice**
  WebAuthn gives users a broad range of choices for authenticating–everything from scanning a fingerprint using a built-in authenticator to tapping a button on a hardware security key.
- **Improved Productivity**
  WebAuthn also frees users from the time-consuming and frustrating tasks of hunting for passwords and resetting passwords. This time-savings extends to help desks and support centers who no longer have to devote time to helping users reset passwords.
- **Reduced Costs**
  WebAuthn reduces costs associated with passwords, including productivity costs, support costs, and financial penalties accruing from data breaches perpetrated by attackers using stolen or guessed passwords.
- **Accelerated Software Development**
  WebAuthn accelerates software development by enabling developers to implement best-in-class registration and authentication services with simple calls to the WebAuthn API supported by the browser or platform.

This new standard brings a new level of secure authentication to all users on the Web.

Use Cases: Authentication Made Fast, Easy, and Secure

WebAuthn makes fast and easy authentication a reality. Let us take a look at a few ways the new standard can be used for common IT operations.

WebAuthn supports various models for account authentication, including passwordless authentication, two-factor authentication (2FA), and multi-factor authentication (MFA). In this example, we will follow a user, called Sarah, using passwordless authentication to demonstrate the basic functionality of WebAuthn. Additional factors could be used to supplement the authentication described here.

User Registration

Sarah wants to create an account for a website, service, or application. Instead of entering a username and password, she enters simply a username. In response, she is offered a choice of WebAuthn standard strong authentication methods including:

- Using an internal authenticator and doing one of the following:
  - Entering a PIN
  - Using a fingerprint scan
  - Using a facial scan
  - Using voice recognition
- Inserting and tapping a hardware security key. The hardware security key can be protected with a PIN so that the user will have to enter a PIN while authenticating rather than simply tapping the key.

Taking any of these actions creates an authentication credential, which her WebAuthn-compliant browser or platform then submits to the website, service, or application, where it will be bound to her newly created account.

Thanks to the flexibility of WebAuthn, Sarah can choose to use whichever of the authentication method listed above she likes.

The best practice is to always register a hardware security key first, since it is external to the user's device and therefore can be used to quickly and easily bootstrap a new device if a device used earlier with an internal authenticator is lost or stolen. The website, service, or application might offer the user the opportunity to register additional authentication methods, such as a backup hardware security key, a PIN, etc. It is often recommended that the user register two hardware security keys, if possible.

Sarah chooses to follow best practice and first registers a security key, and then adds a fingerprint reader as a secondary authentication method. The next time Sarah logs in, she can be authenticated quickly and easily, using either the authentication methods she has set up during the registration process. And that's it. In less than a minute, Sarah has created a secure account and can quickly and securely authenticate any time she likes.

User Authentication

The next day Sarah wants to log in, so she simply enters her username.[3] She is prompted to insert her security key to authenticate herself, proving to the website that the login request is coming from her rather than a remote attacker or malicious script.

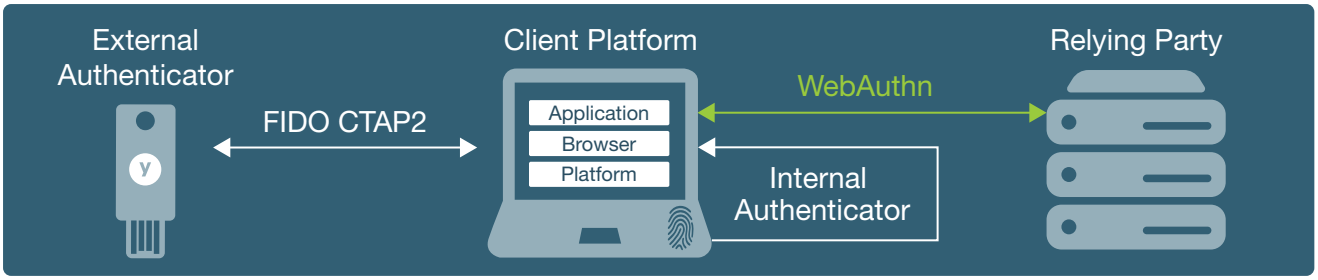
Once she inserts and taps her security key, and enters her PIN (if prompted), the website immediately logs her in.

Account Recovery

By enabling users to register multiple authenticators for each website, service, or application, WebAuthn makes it easy for users to recover access to accounts when devices are lost or stolen.

With WebAuthn, an external authenticator, such as a hardware security key, serves as a portable root of trust enabling users to access accounts, even if a primary device such as a laptop is lost, and to "bootstrap" or onboard new devices with access to specific accounts.

[3] It's possible to omit even this step. The service could receive the user id and credentials from the authenticator itself.



How WebAuthn Works

To understand how WebAuthn works, it is helpful to understand the key terms used in the specification.

Key Terms

A **relying party** is a website, service, or application that needs to authenticate users. For example, it could be a banking website, a SaaS application, a social media app like Facebook, or a mobile app such as a messaging app.

A **client/platform** is the software and hardware being used by a user to authenticate. For example, it could be Windows 10 running on a desktop PC or a Chrome web browser running on a Mac.

An **authenticator** is a mechanism that accepts the user's input to authenticate the user's credential for the relying party. WebAuthn supports a wide range of authenticators that adhere to the FIDO standard, including external roaming authenticators, such as hardware security keys, and internal authenticators such as built-in fingerprint readers and built-in facial scan technology.

All FIDO compliant authenticators include pre-loaded attestation certificates signed by a certificate authority, an organization trusted by internet standards bodies to issue digital certificates and to vouch for the authenticity of those certificates. For example, every Windows 10 PC includes a unique digital certificate signed by Microsoft, and every YubiKey security key has a unique certificate signed by Yubico.

Because each internal or external authenticator includes a digital certificate attesting to the authenticator's integrity, authenticators can be trusted to generate and sign new, unique key pairs for users and websites. (See the sidebar, "WebAuthn and the Benefits of Public Key Cryptography.")

WebAuthn and the Benefits of Public Key Cryptography

WebAuthn relies on public key cryptography. This type of cryptography is asymmetric, because different cryptographic keys[4] are used for encrypting and decrypting data.

For each website, service, or application a user registers with, cryptographic software creates two unique keys:

- a public cryptographic key, which can be shared broadly and which is used for encrypting data
- a unique private key, which is not shared with anyone else, and which is used for decrypting data

The two keys are mathematically related so that data encrypted with the public key can be decrypted only with the private key. But the mathematical relationship between the keys is complex so that the public key cannot mathematically derive the private key, even with the aid of very powerful computers.

Public key cryptography creates a convenient system for keeping data safe.

Websites, services, and applications store only users' public key credentials. The keys are useless for authenticating users on other sites, and they cannot be reverse-engineered to derive private keys.

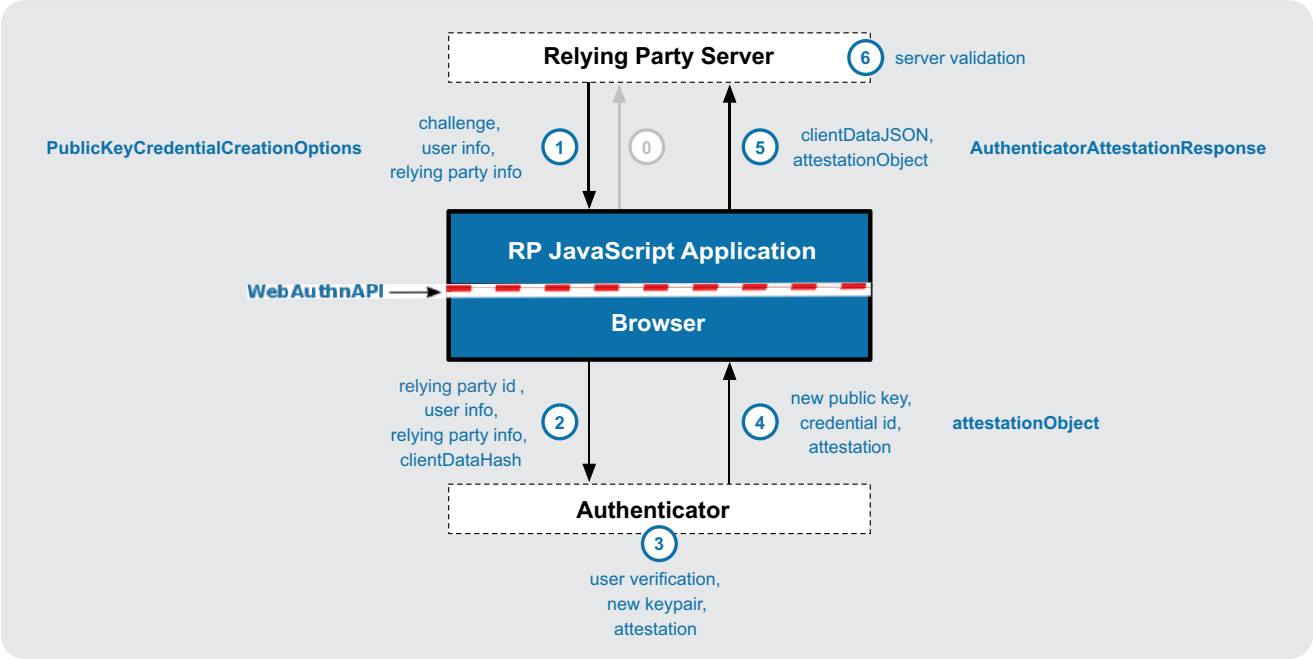
A user's public key is always safer than a password stored on the server. If hackers break into a website and steal its public keys, they still cannot access any data encrypted with the private key or gain access to any users' confidential data.

By relying on public key cryptography, WebAuthn eliminates many of the security vulnerabilities inherent in passwords, while providing a streamlined user experience for registering and authenticating accounts.

[4] "A cryptographic key is a string of bits used by a cryptographic algorithm to transform plain text into cipher text or vice versa." https://www.techopedia.com/definition/24749/cryptographic-key

Let's Get Started

When a user indicates that they want to create an account on the relying party, the relying party issues a challenge to the user's browser. The challenge includes the ID of the relying party. The ID includes the domain of the relying party (for example, yubico.com). It may also include a subdomain (for example, demo.yubico.com).



Including a domain in the challenge prevents other sites from spoofing the relying party. For example, if a phishing site were to send the user a challenge with demo.yub1co.com, the client/platform would recognize the discrepancy and authentication would fail.

Registration with WebAuthn — Step by Step

WebAuthn registration takes only a few seconds to complete, but takes advantage of multiple security features to make account registration highly secure. Here are the steps involved in registering (creating) a passwordless account with a WebAuthn relying party (a website, service, or application that supports WebAuthn).

1. **The user begins the registration process by entering a username to create an account.**
   - *Security benefit*: The user only has to create a username. There is no need to create and remember a password.
2. **The relying party generates a challenge, which is sent to the user's web browser but is not visible to the user.**
   - *Security benefit*: The challenge is unique. Attackers cannot replay it to spoof the relying party.
3. **The authenticator requires the user to take some action, guaranteeing that a human is present. The user takes this action.**
   - *Security benefit:* The authenticator ensures that the user is aware that the registration is taking place. For example, the user enters a PIN or taps the flashing button on a hardware security key. This action assures the relying party that the user's presence is not being faked by a script.

4. **The authenticator creates a key pair unique to the user and the relying party and sends the public key from this key pair to the relying party.**
   - *Security benefit:* The key pair is used to sign an attestation from the user, enabling the relying party to verify that the user really is who they say they are. Because the key pair is unique, it cannot be used to access the user's accounts on other websites.
5. **The relying party uses the public key to verify the attestation signature.**
   - *Security benefit:* This proves that the response is authentic and that the user's identity hasn't been faked. The relying party also learns the manufacturer of the authenticator, as well as which capabilities are supported by the authenticator.

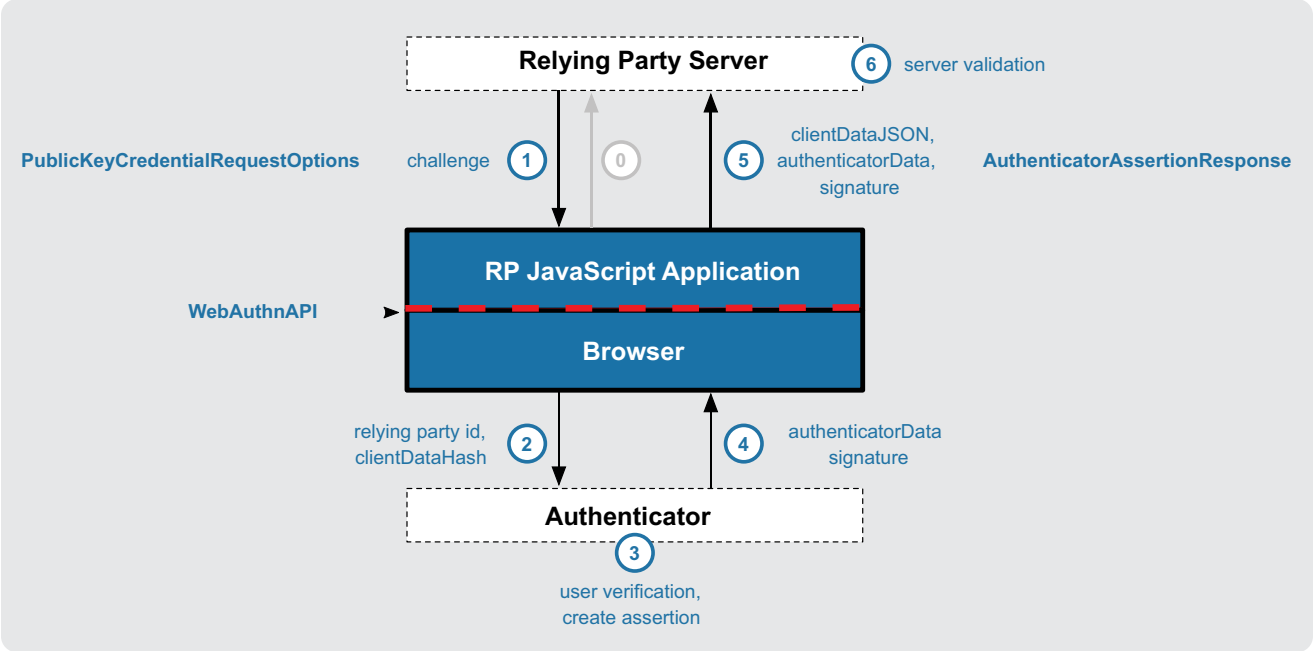
All these steps can be completed within a few seconds. Most of the process is invisible to the user, who simply creates a user name and responds to a prompt. Account registration has never been easier or more secure.

Authentication with WebAuthn — Step by Step

Once a user has created an account on a website, service, or application that supports WebAuthn, he can authenticate quickly and easily using an authenticator (either internal or external.)

Here are the steps involved in authenticating with WebAuthn:

1. **The user navigates to the site and enters his username.**[5]
   - *Security benefit:* The user does not have to enter a password.
2. **The relying party generates a challenge.**
   - *Security benefit*: The challenge is unique, which means a signed assertion from a previous authentication can't be replayed. This prevents attackers from being able to spoof a valid user to the relying party.
3. **The client validates the origin of the challenge.**
   - *Security benefit*: By verifying that the challenge came from the domain the user is really trying to authenticate with, the client prevents the user from accidentally submitting credentials to a phishing site.



[5] Optionally, in a first-factor authentication model, the username is stored along with the attestation credential, so no username needs to be specified. To login, the user simply navigates to the login page and takes an action with an authenticator, such as tapping the button on USB security key or entering a PIN on a Windows 10 PC.

4. **The authenticator requires the user to take some action, guaranteeing that a human is present.**
   - *Security benefit*: The authenticator ensures that the user is aware that the login is taking place and that the login attempt is not being performed by a malicious script.
5. **The authenticator verifies that the challenge came from the relying party and sends a signed attestation using the private key generated during registration.**
   - *Security benefit:* The private key is used to sign an attestation from the user, enabling the relying party to verify that the user really is who they say they are. The private key never leaves the authenticator device.
6. **The relying party uses the public key previously stored during the registration process to verify that the user's authentication is valid.**
   - *Security benefit:* The relying party can easily authenticate the user, since only the user's private key could have signed the attestation the relying party is now decrypting with the corresponding public key.

As with account registration, the authentication process itself is quick and easy. And like the account registration process, most of this process is hidden from the user.

For more details about the WebAuthn API, see the Yubico developer site for WebAuthn: https://developers.yubico.com/WebAuthn/

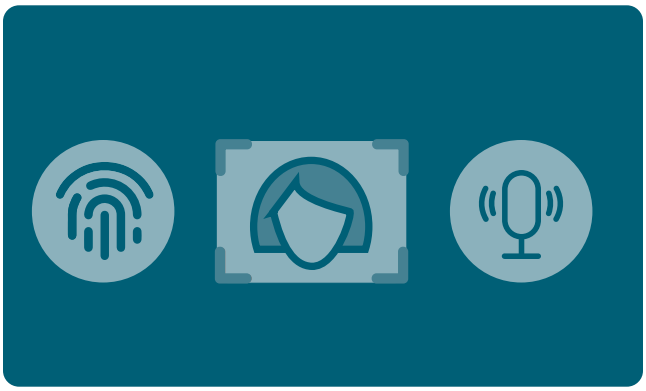
A Closer Look at Authenticators

WebAuthn works with two classes of authenticators: internal and external.

Internal Authenticators

Internal authenticators are built into the computing or mobile device. For example, internal authenticators could include:

- Biometrics built into a smartphone, tablet, laptop, or desktop, supporting:
  - A fingerprint reader
  - Other types of biometric recognition, including face recognition, iris recognition, and voice recognition[6]
  - A PIN, pattern (such as finger swipe pattern on a screen), or a passphrase[7]



When a user enters a PIN to authenticate, the PIN itself never crosses the internet. Instead, a special secure microcontroller in the device cryptographically compares the PIN entered by the user to a PIN the user entered earlier when setting up the device to support PIN authentication. If the PINs match, then the authenticator is unlocked and the login process is allowed to proceed.

[6] Biometric security is usually enforced through a Trusted Platform Module (TPM) or Trusted Execution Environment (TEE)/secure enclave, which provide a secure, tamper-proof execution environment for processes such as verifying biometric data.

[7] Unlike passwords, these forms of authentication are not stored on a server by the relying party, where they might be vulnerable to discovery. They are only stored in a secure repository on the device itself, and they never cross the internet.

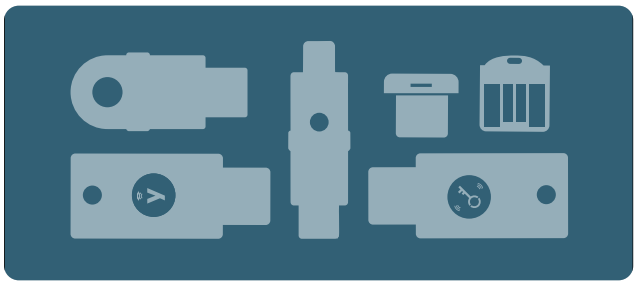
External Authenticators

External authenticators, sometimes known as roaming authenticators, include hardware devices such as security keys with a secure element for generating and storing cryptographic digital credentials.

External authenticators are valuable because they are portable and can be used across multiple devices. They thus act as a reliable “Root of Trust” for onboarding new devices, recovering accounts, and for high assurance step-up authentication.

External authenticators can connect to client/platforms in several ways, including:

- USB (USB-A, USB-C)
- NFC (Near Field Communication)
- Bluetooth



WebAuthn Operations: Economical and Powerful

Developers of websites, services, and applications implement all WebAuthn functionality via simple calls to the WebAuthn API supported by the browser or platform:

- Register
- Authenticate

WebAuthn enables developers to provide best-in-class user experiences and state-of-the-art security without:

- Long, resource-intensive development projects
- Custom coding to support specific browsers or end user devices
- Extensive ongoing maintenance and trouble-shooting of custom code

Download the WebAuthn Developer Guide: https://developers.yubico.com/WebAuthn/WebAuthn_Developer_Guide/

Conclusion

Already supported by all major browsers and most platforms, the WebAuthn standard is poised to make account registration and authentication faster and more secure for users of websites, services, and applications than ever before.

The power of WebAuthn lies in its simplicity and its breadth. With just two basic operations, Register and Authenticate, WebAuthn supports a broad range of authentication models, including passwordless authentication, 2FA, and MFA across all major browsers and most platforms. By accepting credentials submitted through internal authentication techniques such as biometrics, as well as through external authentication techniques such as hardware security keys, WebAuthn provides users with more control over authentication than they have ever had before. At the same time, WebAuthn eliminates not only the need for passwords, but also the need for extensive custom development and testing to support multiple authentication models on devices ranging from smartphones to desktops. With WebAuthn, the broadest support and the strongest security are combined in a highly flexible standard.

Now that the W3C has approved this standard and WebAuthn has been adopted by major browsers and operating systems, the time is right for web and mobile app developers to begin implementing WebAuthn in their own software releases. It is also time for internal IT teams to begin exploring the benefits of hardware security keys for employees and partners, so that convenient, passwordless authentication is available for all trusted users in all environments.

It is rare that a standard offers so many benefits to so many stakeholders. Public sector entities should seize the opportunity afforded by WebAuthn to radically improve authentication and digital experiences for users.

How to Get Started with WebAuthn

If you would like to learn more and even get some hands-on experience with WebAuthn, here are some free resources you can turn to.

- Read the complete WebAuthn specification on the W3C site: https://www.w3.org/TR/webauthn/
- Read the other white papers in the Yubico WebAuthn series:
  - *Introducing WebAuthn: Enabling a Streamlined and More Secure User Authentication Experience*
  - *Establishing a Secure Portable Root of Trust with WebAuthn*
  - *SIM Swap: Protecting against Account Takeovers with WebAuthn*
- Read an overview of WebAuthn on the Yubico website: www.yubico.com/webauthn
- Try out the Yubico WebAuthn demo site: https://demo.yubico.com/webauthn
- View the Yubico Technical Overview webinar: FIDO2 WebAuthn Server Validation
- View the Yubico webinar, FIDO2 Authentication Demystified
- Visit the Yubico developer site and access free developer resources for WebAuthn: https://developers.yubico.com/WebAuthn

yubico

About Yubico

Yubico sets new global standards for simple and secure access to computers, mobile devices, servers, and internet accounts.

The company's core invention, the YubiKey, delivers strong hardware protection, with a simple touch, across any number of IT systems and online services. The YubiHSM, Yubico's ultra-portable hardware security module, protects sensitive data stored in servers.

Yubico is a leading contributor to the FIDO2, WebAuthn, and FIDO Universal 2nd Factor open authentication standards, and the company's technology is deployed and loved by 9 of the top 10 internet brands and by millions of users in 160 countries.

Founded in 2007, Yubico is privately held, with offices in Sweden, UK, Germany, USA, Australia, and Singapore. For more information: www.yubico.com.