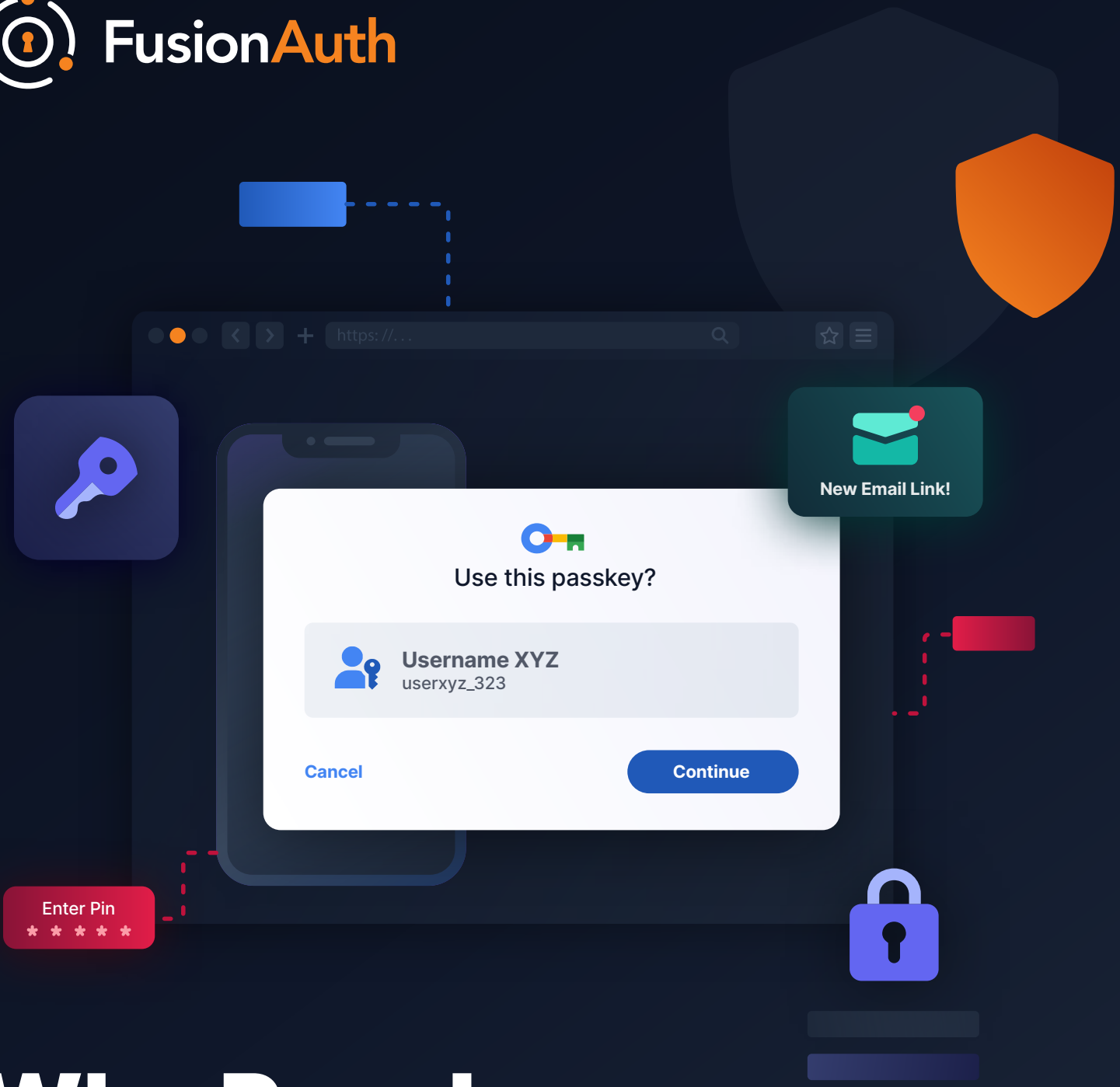
FusionAuth



Why Passkeys Improve User Security & How to Implement Them

WHITE PAPER

Table of Contents

Every user, whether at home or at work, must manage an overwhelming number of usernames and passwords. Unfortunately, passwords pose significant problems in today's digital landscape, including susceptibility to a variety of attack vectors and burdensome password requirements. Passkeys are emerging in the realm of digital security to transform how we manage online authentication, enhancing security and improving user experience in the digital space.

Moving Towards a Passwordless Web

According to the **2023 Verizon Data Breach Investigation Report**, poorly chosen and protected passwords are still one of the major sources of breaches in web application attacks, despite many warnings from security experts and the increased use of password management tools. Indeed, password management tools themselves are under attack, exposing millions of painstakingly generated and saved complex passwords to attackers and leaving their customers unsure of how to best protect their online accounts.

74%

Of all breaches are due to human error, privilege misuse, use of stolen credentials or social engineering.

Despite the challenges inherent in passwords, authentication has never been more important. Authentication enables access control and prevents unauthorized access to sensitive information and protected resources. This helps to mitigate security threats, such as identity theft and fraud, and facilitates accountability. And by tying access and activity to identified users, authentication helps organizations appropriately investigate and audit access following security incidents, something now required for public companies by new **Security and Exchange Commission rules**.

Robust authentication solutions also enable organizations to deliver the personalized experiences users in digital ecosystems have come to expect.

Allowing applications and services to offer customized access and functionality based on individual user identities and preferences.

2023 SEC Cybersecurity Reporting Requirements.

What to Report - Item 1.05

Must Report:
Nature + Scope + Timing

How to Report - Item 106

Registrants must describe their processes for assessing, identifying, and managing material risks from cybersecurity threats.

Recognizing the need for authentication that is more secure than passwords alone or passwords paired with multi factor authentication (MFA) and one time passcodes (OTP), the **Fast Identity Online (FIDO) Alliance** launched publicly in 2013 to reduce the world's dependence on passwords. The FIDO Alliance is an open industry association that promotes standards for authentication, supporting authentication technologies that are easier for consumers to use and service providers to deploy and manage.

These authentication standards are based on public key cryptography and enable login experiences in websites and apps to be replaced with a more secure and quick login experience. In other words, passkeys.

> Passkeys are a more user-friendly way of talking about **WebAuthn.**

which is a World Wide Web Consortium (W3C) **specification** that defines an API to create and use public-key credentials in web applications to allow users to authenticate in their browser the same way they unlock their device.

How Does WebAuthn Work?

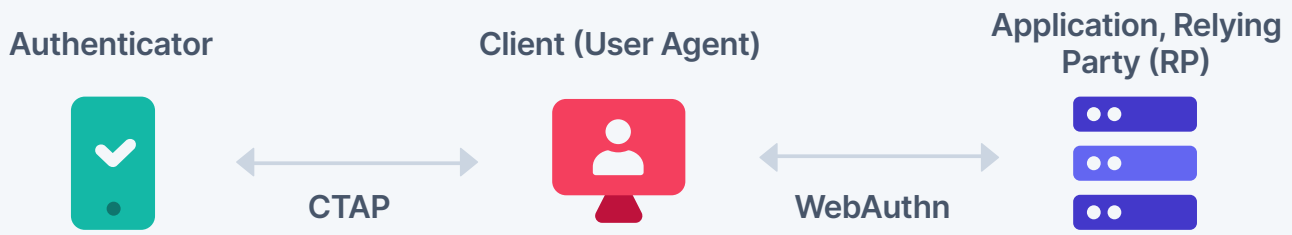
WebAuthn provides credentials and requires a piece of hardware or software to act as the authenticator. An important feature of an authenticator is that it connects with the client without using the Internet.

Relaying Party: is the application that performs the authentication of the user.

Client: is the software that implements the Web Authentication API.

Authenticator: is a device that creates and stores user credentials.

To connect to an authenticator, developers use the **Client to Authenticator Protocol (CTAP)**, which enables user-controlled cryptographic authenticators, such as hardware security keys or smartphones, to interoperate with a client platform, such as a laptop or a browser.



WebAuthn works because CTAP has been implemented for all major browsers.

The Mechanics of Passkeys

Passkeys offer a significant advancement in digital authentication, using the principles of public key cryptography to create a desperately needed authentication alternative to traditional passwords. Let's look more closely at how passkeys function, whose capability is heavily reliant on public key cryptography. Passkeys use a pair of keys to ensure secure communication. These two keys are:

Private Key:

which is kept secret and stored securely on the user's device.

Public Key:

which is shared with the online service or server.

The basic principle with public key cryptography is that any data encrypted with the public key can only be decrypted with the corresponding private key, and vice versa. When a user sets up a passkey for a service or website, their device generates this key pair. The public key is shared with the service and stored in the remote database, while the private key remains only on the user's device.

A key Pair

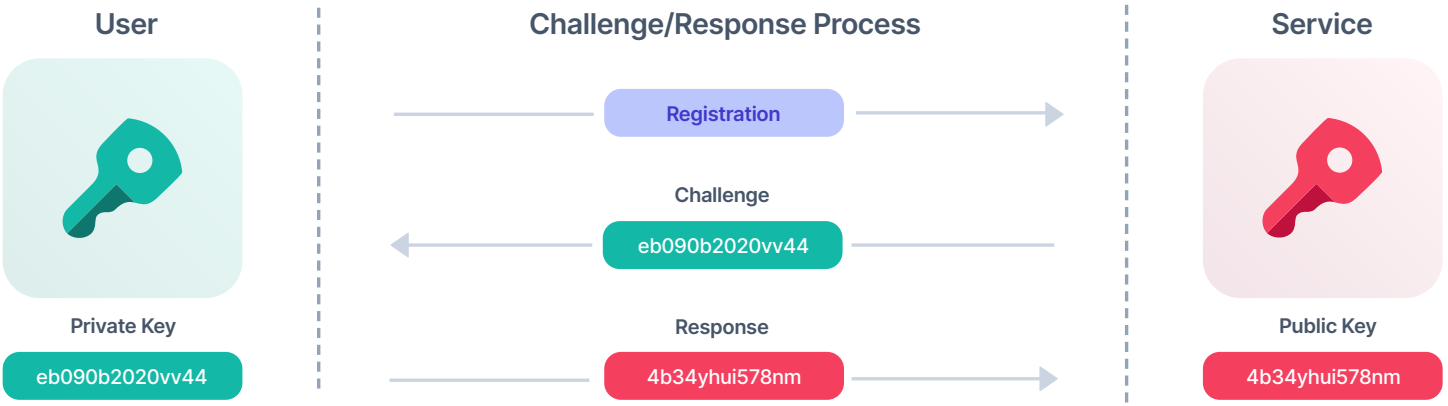
What one key does	The other will validate
Public: Encrypts	**Private:** Decrypts
Private: Digitally signs	**Public:** Verifies the signature
Private: Authenticates	**Public:** Verifies the authentication

Challenges of Passkeys

When the user wants to access the service again, the service sends a challenge to the user's device. The device uses the private key to sign this challenge, which creates a digital signature that is sent back to the service, which is commonly referred to as the relying party. Then the service uses the stored public key to verify the signature. If it matches, it confirms that the user has the correct private key and can authenticate the user's identity.

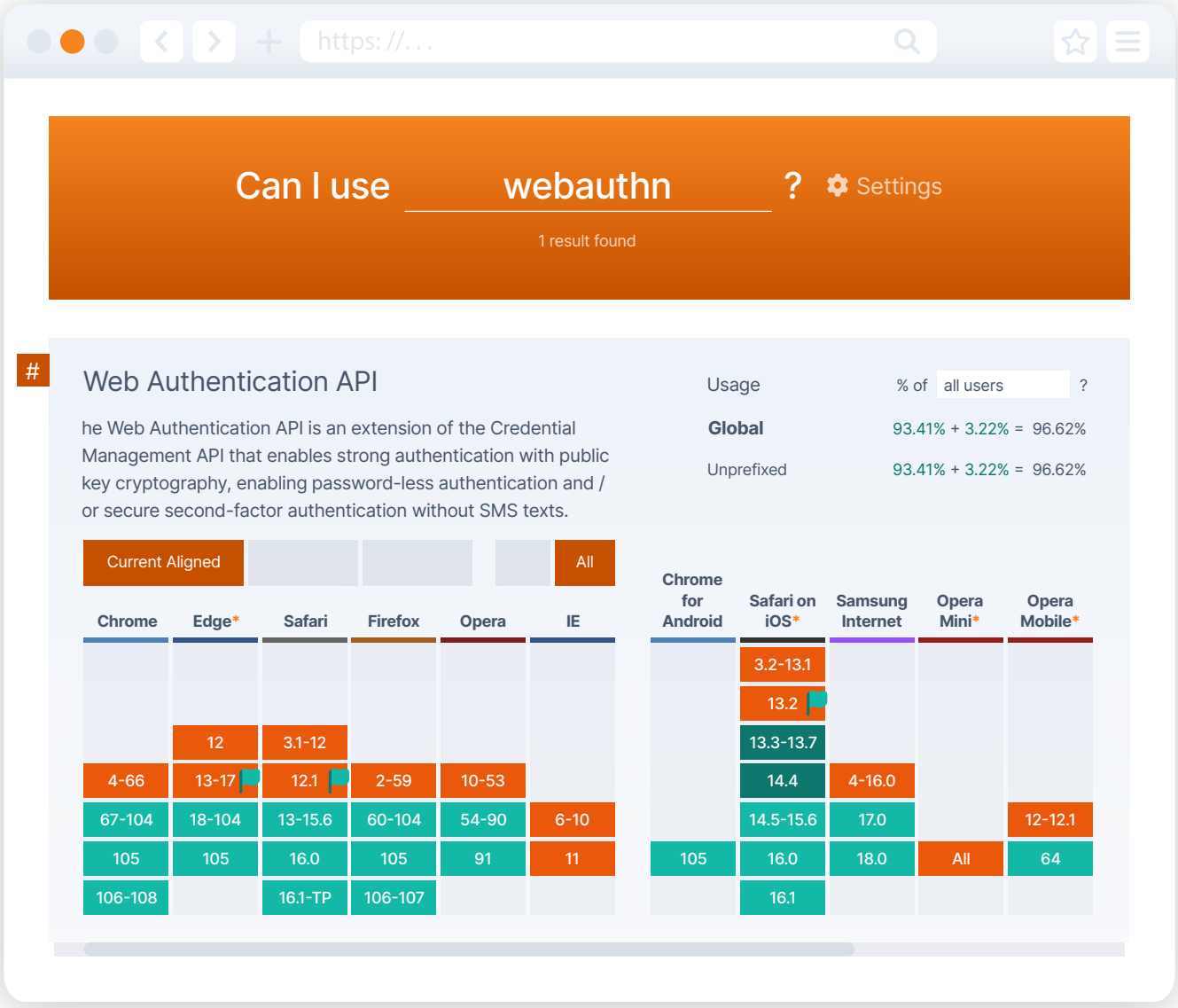
> Using passkeys offers several security benefits because it requires no exchange of passwords.

The private key never leaves the user's device, which significantly reduces the risk of many types of cyberattacks. And because each service has a different unique key pair, if one service is breached, it does not compromise any other services. Authentication no longer requires the user to remember the right login id for the service, plus the paired password, and, often, an OTP sent by text message, authentication app, or email. Instead, users can authenticate with a secure, private passkey that they never have to remember. A challenge signed by a private key for one service at a given hostname can never be presented to a service at a different hostname, leading to greater protection from phishing.



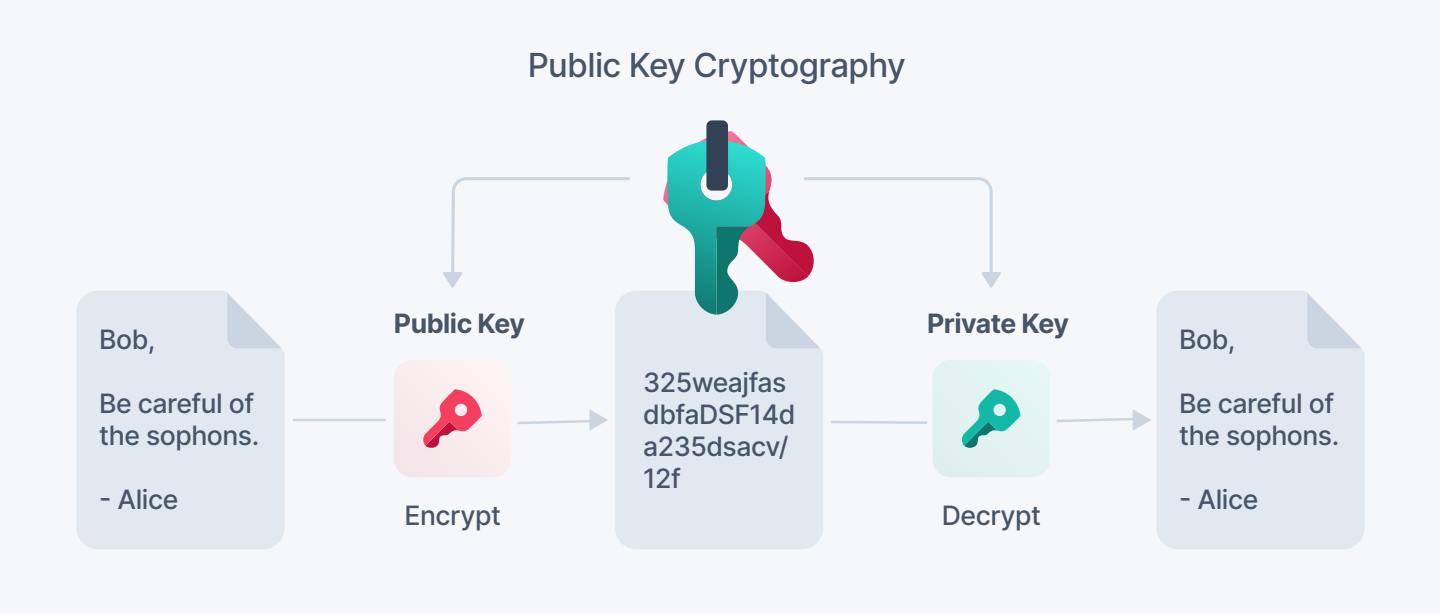
The Passkey Authentication Process

Passkeys are stored directly on your device, and supported by Apple, Microsoft, Google, and many other companies, using the authentication mechanisms built into the devices users already have. You can create a passkey with any compatible hardware, including laptops, desktops, mobile phones, and tablets, as well as any hardware security key that supports the **FIDO2 protocol**. It is compatible with nearly every modern browser, so you don't have to push your users to use one browser over another for it to work.



For users, the authentication process is simple:

1. Visit a website that supports passkeys, create a new account, and use a passkey to secure it.
2. The site will confirm your authenticator, which may be a smartphone, another mobile device or hardware authenticator, or a password manager that supports passkeys.
3. The authenticator requires an additional form of verification, which could be a master password or biometrics, such as a face or fingerprint.
4. The public and private key are generated by your authenticator; the public key is stored on the company's website and the private key is stored on your device.
5. When you log in, the site's server sends a challenge to your authenticator, which your private key uses to create a signature and result that it sends to the server.
6. The server verifies the challenge was signed by the corresponding private key.
7. Once this process is complete, you can access the account using your passkey.



A few things to note: if you're re-authenticating, you need to use the same passkey, and a passkey registered with a platform authenticator on one browser (Chrome) won't work in a different browser (Safari or Edge). However, users can complete another WebAuthn registration ceremony in a different browser to generate and use a new passkey.

Passkeys vs. Traditional Passwords

Traditional password systems are inherently vulnerable. Perhaps because passwords have been the primary form of authentication for decades, cyber attackers have found multiple ways to take advantage of their inherent weaknesses.

Phishing attacks involve tricking users into divulging their passwords to malicious actors, often through deceptive emails or websites that mimic legitimate services, and phishing works.

IBM's 2023 Cost of a Data Breach Report showed that phishing was the initial attack vector in 16% of all breaches. Authentication processes that rely on traditional password systems are highly vulnerable to phishing because they require users to input their credentials, and it's not difficult to trick users into entering their passwords into a fraudulent site.

Fake websites, emails, and even QR codes are easy to make and difficult for the average person to detect. Similarly, attackers may use other social engineering techniques to lead people to reveal their passwords, allowing unauthorized access to sensitive information.

The cost of a data breach in 2023 rose to $4.45 million, according to IBM.

Phishing - the fraudulent practice of sending emails or other messages purporting to be from reputable companies in order to induce individuals to reveal personal information, such as passwords and credit card numbers.

One troubling fact about traditional passwords is the tendency for people to reuse passwords across multiple sites.

Maybe for convenience or use the same basic password and just change it slightly for use on another site. That means that if a malicious actor gets a password for one site, they may be able to access other accounts easily.

DID YOU KNOW?

Only 3 in 10 employees create strong passwords for their work accounts, according to LastPass.

Read more: **Psychology of Passwords 2022**

Remembering many passwords is difficult, so people often create passwords that are easy to remember, which makes them easy to guess or crack. **Have I Been Pwned** allows users to check whether their email address has been exposed in a data breach, and it contains nearly 13 billion accounts (exposed username and password combinations) that attackers may use to gain access to other accounts.

Both simple and commonly used passwords can also be used in brute force attacks, when attackers try many username/password combinations using automated tools.

Another issue with passwords is that they are typically stored on servers and may be vulnerable if the server is compromised, particularly if the passwords aren't appropriately protected. MFA provides an additional layer of security, but it can be cumbersome to set up and use — and attackers have found ways to redirect or intercept some MFA challenges, reducing the effectiveness of MFA.

Passkeys offer a much simpler way for users to authenticate securely.

Since there's no need to input anything on a website, the opportunity for phishing attacks drops dramatically. And because users no longer need to remember or manually enter credentials, passkeys eliminate many potential human risks, including use of weak or reused passwords, social engineering techniques, and reliance on complex passwords.

Passkeys are ideal because they're effective even when individual users don't have strong security practices — the security is built into the technology and stored on individual's devices, requiring minimal user interaction, such as a simple biometric check or device prompt.

Real-world Passkey Implementations

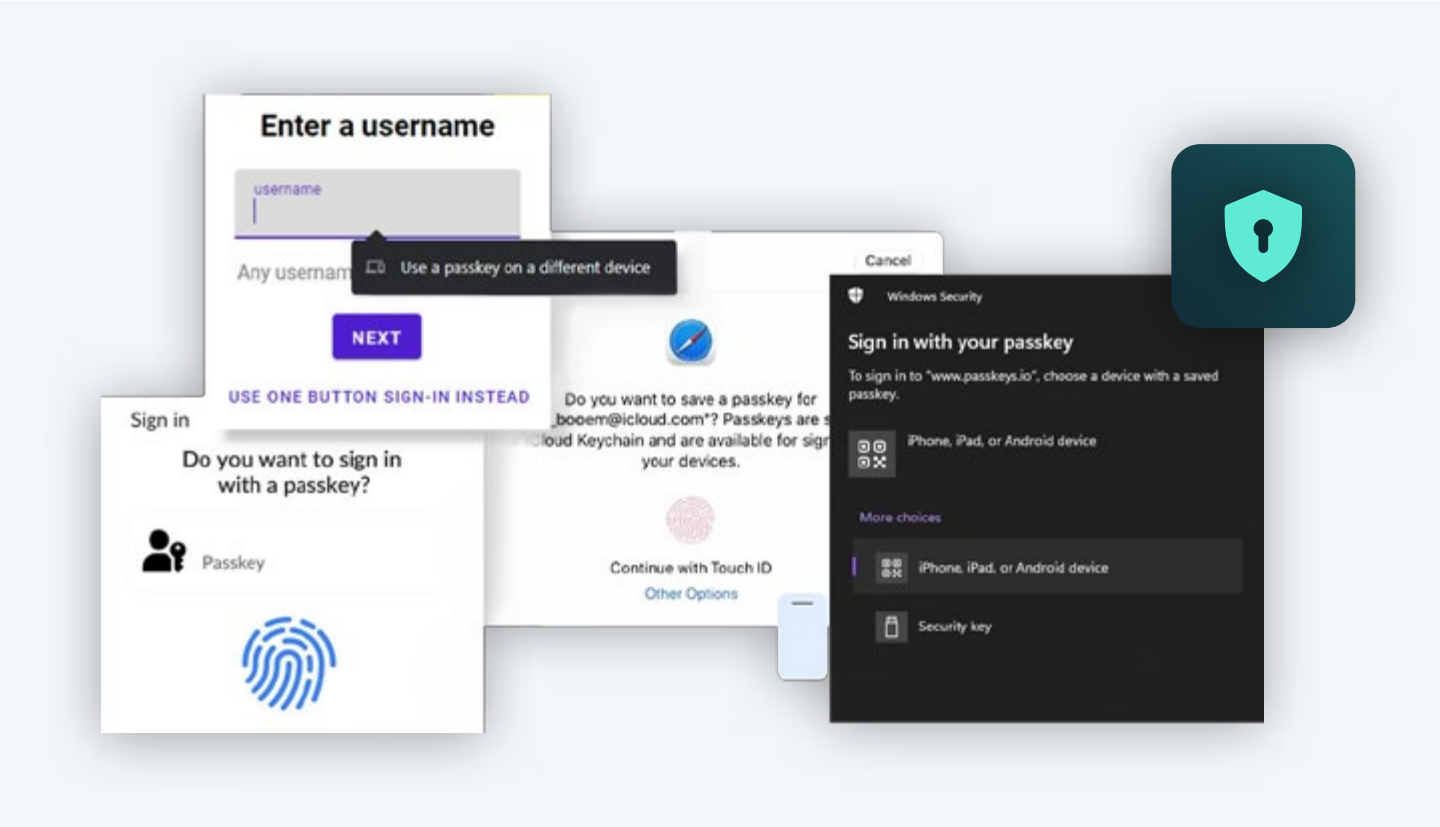
How Apple Uses Passkey Systems

Apple has been at the forefront of implementing **passkey systems**, integrating the authentication technology into their ecosystems to increase security and improve user experience. Passkeys work smoothly across various Apple devices, including **iPhones**, iPads, and Macs. This integration provides a positive and seamless user experience moving from one device to another. For example, a Watch or iPhone paired with a Mac can automatically unlock it, or a user can unlock their device with a fingerprint or using Face ID in seconds.

In Apple's ecosystem, passkeys are stored in the iCloud Keychain, which synchronizes across all devices signed in with the same Apple ID.

This enables Apple to maintain high security standards while still providing convenience and accessibility to its users.

Users can generate passkeys with just a few taps, authenticating using face or fingerprint recognition. When creating a passkey, the system automatically uses strong, unique passkeys. Users do not need to create and remember them. Passkeys are protected with end-to-end encryption and are stored locally on the device and securely backed up in iCloud.



Apple also works to ensure that its passkeys are compatible in both its own ecosystem and with other platforms, aligning with broader industry efforts to standardize passkey technology. Users can authenticate with their passkeys on any of their Apple devices, which makes the process flexible and user-friendly. It provides mechanisms for account recovery in case a device is lost or inaccessible, including using another trusted device or account recovery information. Apple also provides **APIs and tools for developers** to integrate passkey support into their apps and websites to encourage adoption and increase the overall security of their ecosystem and beyond.

Practical Implementations & Challenges

Implementing passkey systems, while offering significant benefits in terms of security and user experience, also may present some practical challenges. Passkeys require adoption of newer and perhaps unfamiliar technologies, user adoption, and may raise some concerns regarding compatibility. A few potential challenges include:

1. **A relatively new standard:** Users are still accustomed to password-based authentication, and WebAuthn is not yet a widely used standard.
2. **Passkeys are device-specific:** If a user registers a passkey to an account using their Apple laptop and TouchID, they can't log in to their account on their mobile device using biometric Face ID. Users need to set up individual passkeys for each device, except when the provider offers proprietary cross-device synchronization, as Apple does. Many password managers and operating systems are working to support the synchronization of passkeys.
3. **Many websites and apps don't yet support passkeys:** Password and email address pairs are still an easy way for developers to create and authenticate accounts, even though they do not offer the same level of security as passkeys. For this reason, support for passkeys is increasing rapidly. Developers can implement passkeys for applications using **WebAuthn** developer resources.
4. **Lost device access:** Because passkeys are stored on user devices, it may be difficult to recover account access if a user loses access to the device. Apple and Google both offer solutions to help with lost devices and enable users to transfer passkeys from an old device to a new one. More capabilities to help users recover account access are under development.

Developers can use in house resources and open libraries to implement passkeys or turn to a third-party service to provide passkey functionality.

While in-house implementations are more time- and resource-intensive, they also allow developers to create customized solutions and put controls in place for unique use cases.

Third party services provide pre-built biometric authentication services and are simpler to implement, while also providing additional security benefits by offering a third-party audit.

Best Practices for Implementing Passkeys

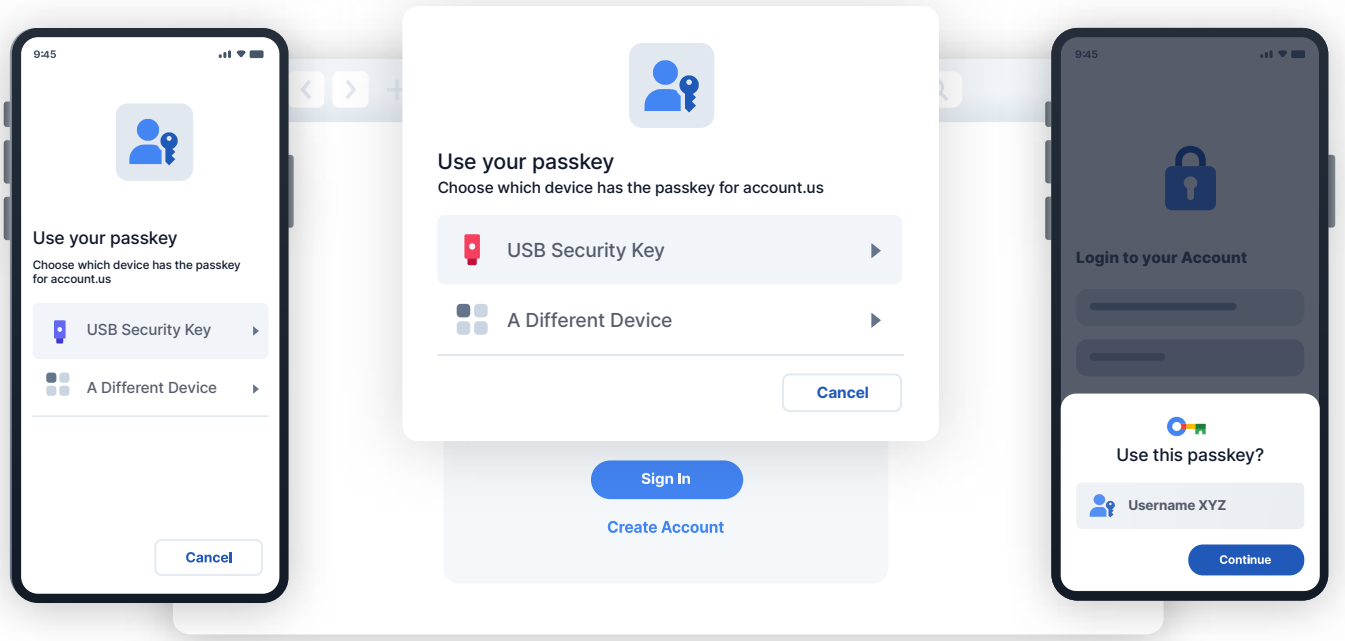
Implementing passkeys in different systems involves several steps that span technical, operational, and user-experience domains. The following are a few things to consider as you implement passkeys and suggestions for how to facilitate the process in different systems.

Prioritize User Experience

Encourage adoption by setting passkeys as the preferred login method during account creation and when users are logging into an existing account. Use simple, familiar language to explain the benefits of passkeys and guide users through the setup process. Some users may prefer to use passwords or **federated identities**, so make it easy for them to choose the option that works best for them. The FIDO Alliance provides helpful **user experience guidelines and best practices** for implementing a positive authentication experience.

Ensure Smooth Integration

Use existing, secure libraries and open-source software tools, such as the WebAuthn API libraries or FIDO2 server solutions, and implement appropriate server-side validation to ensure the authenticity and signature of the passkey before granting access. Research potential passkey behavior variations between different browser and operating system combinations and take steps to minimize confusion. Test your passkeys across various devices and platforms to ensure seamless functionality.



Promote Security and Accessibility

Use asymmetric cryptography, where the public key is stored on the server and the private key remains on the user's device. Adhere to standards, particularly FIDO2, WebAuthn, and CTAP, which provide frameworks for secure authentication. Implement secure backup mechanisms for passkeys in case of device loss or damage. Securely manage session tokens post-authentication to prevent session hijacking and verify user identity thoroughly during the initial setup or registration process and periodically re-verify the user's identity to maintain security integrity.

Make sure your passkey implementation is compatible with assistive technologies, including screen readers, to ensure accessibility for users with disabilities, or provide alternatives.

Additional Passkey Implementation Resources

There are a number of useful guides for implementing passkeys available; here are a few options to help you get started:

- **Apple Passkeys Overview for Developers**
- **FIDO Alliance Passkey Resources**
- **Google Developers Passkey Guide for Relying Parties**
- **WebAuthn Explained**
- **WebAuthn APIs for Passwordless Authentication on Windows**
- **Yubico Passkey Implementation Guide**

The Future of Digital Security: Beyond Passkeys

The future of digital security, while currently focusing on passkeys and leveraging public and private keys, extends far beyond them, including a range of emerging technologies.

These innovations are driven by a rapidly evolving cyber threat landscape and the increasing need for secure, efficient, and user-friendly authentication solutions in the digital realm.

While passkey adoption may not be moving as fast as many would prefer, new capabilities in the following areas will help ensure the adoption and security of passkeys in the future:

1. **Biometric Evolution:** Expect more sophisticated biometric technologies, such as heart rate patterns, gait analysis, and even brainwave recognition, to play a role in user identification and authentication.

2. **Behavioral Analytics:** Continuous analysis of user behavior patterns (such as typing rhythm, gait analysis and mouse movements) can help detect anomalies indicative of unauthorized access, allowing authentication requirements to adjust automatically based on the risk profile associated with the user's current behavior.

4. **Quantum Computing and Cryptography:** Development of **quantum-safe security keys** before quantum computers are able to crack current standards.

5. **Artificial Intelligence and Machine Learning:** AI/ML algorithms are becoming increasingly available and sophisticated, which may help predict and even prevent security breaches before they occur.

6. **Decentralized Identity Models:** Some countries are **using blockchain technology** to create decentralized and tamper-evident identity verification systems, which may help users own and control their own identity credentials rather than relying on centralized authorities.

7. **Internet of Things (IoT) Security:** Expect more robust security protocols using passkeys for IoT devices, focusing on securing edge computing environments.

8. **Increasing Regulatory and Ethical Frameworks:** As cyberattacks and breaches become more common and the implications of exposure of sensitive data more severe, countries are likely to enact stricter regulations to protect personal data, establishing ethical guidelines for the use of AI, biometrics, and other technologies.

As passkeys adoption grows, advancement of these new capabilities will help ensure secure authentication and protect electronic transactions and communications.

Stay Ahead in Digital Security with Passkeys

Read more about passkey implementation in our docs:

Authentication With WebAuthn & Passkeys

WebAuthn is a W3C specification that defines an API to create and use public-key credentials in web applications that provides the ability for users to authenticate...

Add WebAuthn Passkey

WebAuthn is disabled by default on a tenant. To toggle navigate to Tenants → Edit Tenant → WebAuthn.

The increasing adoption of passkeys signifies a critical step forward in improving digital security, changing well, we protect digital identities and assets. In an era of increasingly sophisticated cyber threats, passkeys address many of the vulnerabilities inherent in traditional password systems, particularly by reducing the risks of phishing, social engineering, and brute force attacks. The convenience of passkeys make it easier for users to comply with security protocols by creating a secure, user-friendly authentication process.

The question is not whether you should implement passkeys, but when and how to do it to bring a safer, more secure, and easy-to-use authentication method to your users.

Thank You

FusionAuth is the authentication and authorization platform built for developers, by developers. For technical leaders creating products for external users, it solves the problem of building essential user security without distracting from the primary application.

Learn more at FusionAuth.io