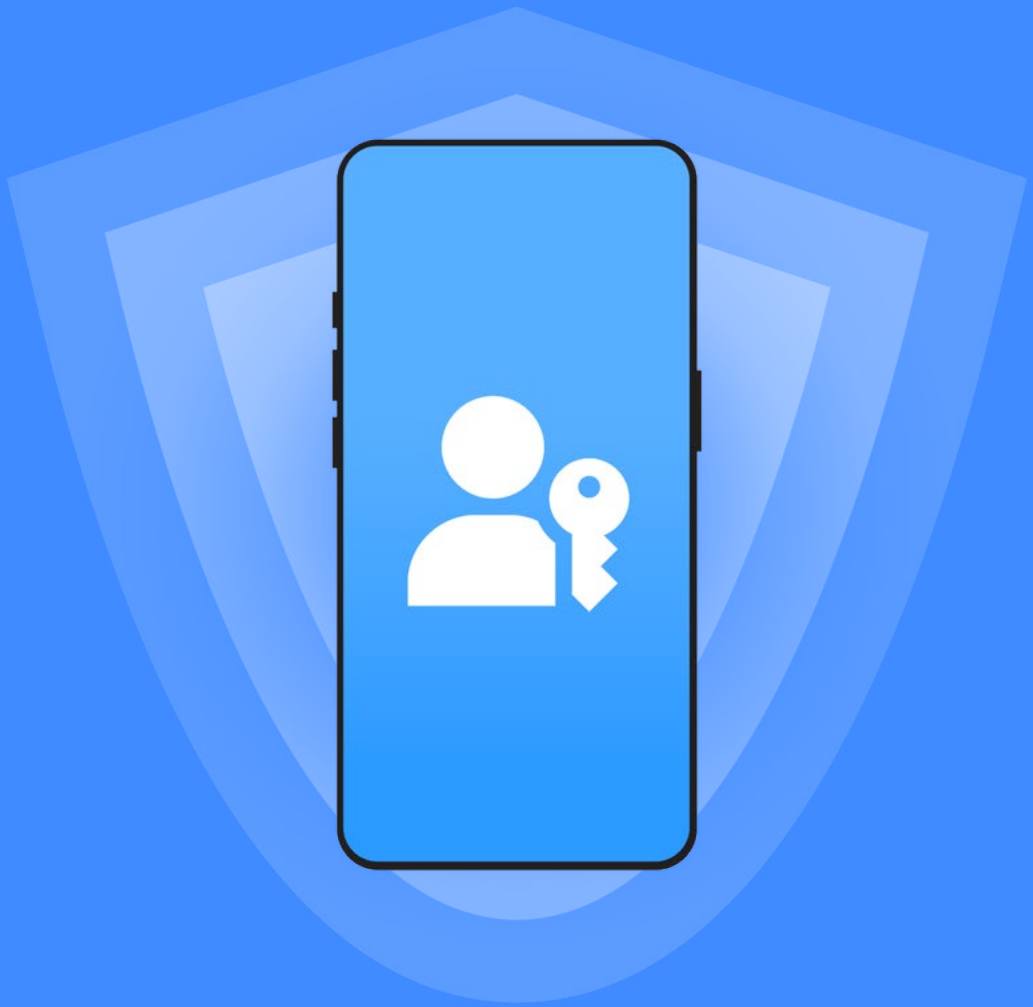
OneSpan

Passkey Security



WHITE PAPER

Contents

Introduction

The transition from traditional password-based authentication to modern, phishing-resistant methods has become a critical priority for organizations facing increasingly sophisticated cyber threats. Passwords and legacy multi-factor authentication (MFA) mechanisms, such as SMS-based one-time passwords (OTPs), have proven vulnerable to large-scale attacks, credential theft, and social engineering. These weaknesses have driven the adoption of stronger, user-friendly alternatives that can withstand scalable and automated attack vectors.

Passkeys, built on the FIDO (Fast IDentity Online) standards, represent a paradigm shift in authentication security. They eliminate shared secrets, leverage asymmetric cryptography, and bind credentials to specific domains, making them inherently resistant to phishing and credential replay attacks. By design, passkeys store private keys securely on user devices and never transmit them to servers, significantly reducing the risk of mass credential compromise.

This white paper explores the security properties of passkeys, their resilience against common attack classes, and the implications of different deployment models such as device-bound and synced passkeys. It also examines industry endorsements, formal security analyses, and practical considerations for organizations adopting passkeys as part of their authentication strategy.

Executive summary

Passkeys, based on FIDO standards, represent a significant advancement in authentication security by eliminating shared secrets and providing strong phishing resistance. Unlike traditional credentials such as passwords or one-time passwords (OTPs), passkeys leverage asymmetric cryptography, ensuring that private keys remain securely stored on user devices while only public keys are registered with the relying party. This design mitigates large-scale credential theft from servers and renders common attack vectors like credential stuffing and phishing ineffective.

While any passkey is more secure than passwords and OTPs, the security of passkeys depends on the authenticator type and key management model.

Device-bound passkeys offer the strongest protection because keys never leave the device and are typically stored in secure hardware elements or trusted execution environments. In practice, they are tied to the device lifecycle, requiring secure migration processes as part of deployment planning.

Synced passkeys improve usability by enabling multi-device access through cloud backup and synchronization. While convenient, this model introduces potential risks, such as reliance on the passkey provider's security measures for cloud storage, recovery flows, and export/import functionality. These risks, however, can be mitigated by the relying party by adding device binding to synced passkeys.

Passkeys also provide resilience against scalable attacks. They neutralize server-side credential breaches and phishing attempts by binding credentials to specific domains (relying party IDs). Protection against client-side malware and session hijacking depends on the implementation of user verification and platform security. Physical attacks remain possible but are not scalable and require sophisticated techniques.

Government bodies such as CISA, ENISA, and NIST endorse FIDO-based authentication as the gold standard for phishing-resistant MFA. Formal security analyses further validate the robustness of the FIDO protocols. Compared to passwords and OTPs, passkeys deliver superior security characteristics with improved usability, making them a compelling choice for organizations seeking to strengthen their authentication posture.

Authentication in context

When allowing remote users to access a system, the identity of that user matters. Is that user the legitimate account holder or an attacker?

Consider these scenarios:

Sign-up

In typical systems, a Know-Your-Customer (KYC) process is performed. That process starts with identity proofing to verify the relation of the submitted name and potentially other identity attributes to the "person at the other end" – the user. This process is often followed by additional background checking to determine whether the user should even get an account.

Sign-in

Since these steps are not very convenient for the user and costly for the company, a credential is issued to the user to be recognized in subsequent attempts to access the system (authentication). It is important that the process of binding the credential to a user is appropriate for the assurance of the identity proofing and the assurance level of the authentication. In practice, the binding method could be a browser session, a physical letter with an OTP, the shipment of a dedicated hardware token plus a PIN letter, or even an in-person hand-over.

Credential lifetime

In an ideal world, the user would keep the credential forever and could also use that credential on any device. However, users might lose their credentials. Sometimes credentials are bound to physical devices in order to prevent uncontrolled replication. Those devices (e.g., dedicated hardware tokens, smartphones, or smart cards) could be lost or stolen and they might need to be replaced after some years.

In this document, we focus on the authentication or sign-in process, ignoring the security aspects of the ID proofing and the process of binding the credential to the user, as they are not specific to passkeys.

Threat landscape

Not all attacks are created equal. When authenticating to web services over the internet, we especially care about **scalable attacks**[1]. These are attacks that essentially only require the "investment" of a fixed amount of money, independent of the number of attack targets.

For example, stealing passwords from 100 million users needs a framework to break into the server, whereas stealing smartphones from 100 million people would need 100 million times the effort of stealing the smartphone from one user. Because of their scale, scalable attacks affect our economy.

- **Class 1 attacks** target the server to steal credentials for impersonating users, or operate as an adversary-in-the-middle to intercept the credential and finally own the authenticated session.
- **Class 2, 3, and 4 attacks** can be automated to affect a large number of client devices.
  - Class 2 attacks attempt to extract the credential from the client device (e.g., key logger malware or similar).
  - Class 3 attacks don't steal the credential, but attempt to misuse the credential to then gain access to an authenticated session that can be exploited.
  - Class 4 attacks wait until the user has authenticated a session and then misuse that session, for example by stealing the session cookie or exploiting the session directly.
- **Class 5 and 6 attacks** assume physical access to the client device.
  - Class 5 attacks attempt to extract the credential from the device, for example by decapping[2] chips and running differential power attacks or other lab-style attacks.
  - Class 6 attacks attempt to misuse the credential (without extracting it) by forging the user gesture (e.g., brute-forcing the PIN or running a spoofing attack on the biometrics).

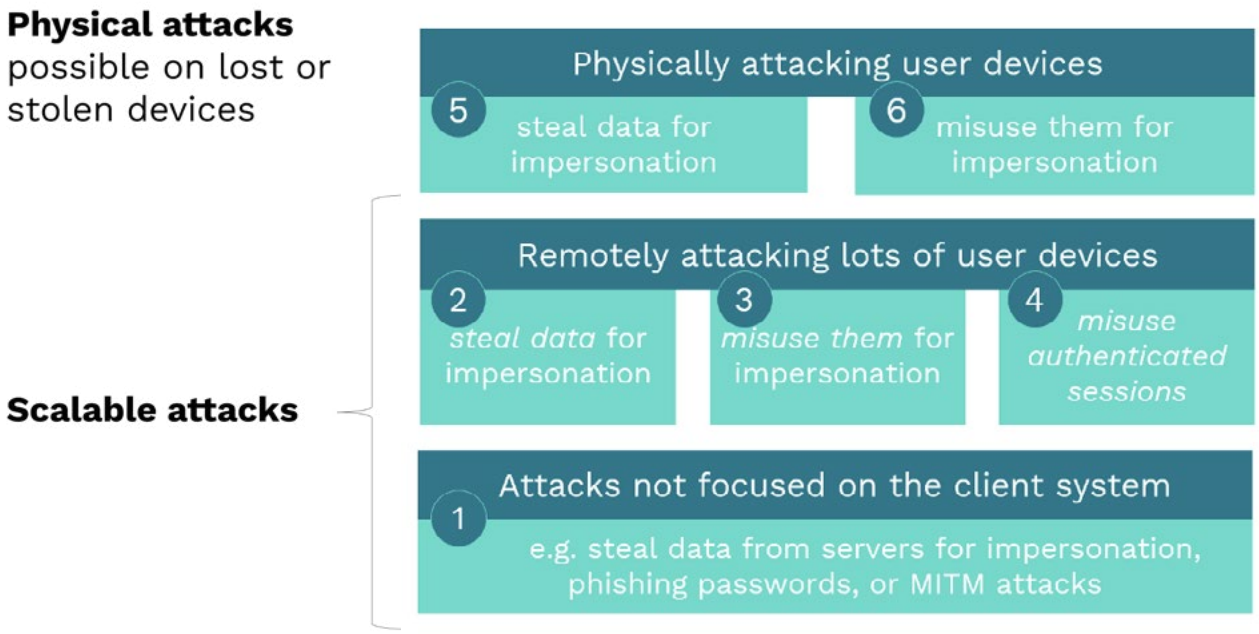


Figure 1: Attack classes

Security properties of passkeys

Passkey authentication in general has the following security properties:

(1) It is a two-factor authentication method. The user gesture represents one factor and the possession of the cryptographic key represents the other factor. As a result, biometrics or PINs cannot be misused without access to the authenticator.

(2) The cryptographic keys are generated with high entropy. Unlike passwords, they cannot be brute-forced.

(3) No secrets are stored on the server. That means that attacking the server to steal passkeys won't be sufficient to impersonate users.

(4) Phishing doesn't work since only the legitimate app (not a phishing website) can trigger the use of the relying party's passkey.

Looking at the attack classes in Figure 2, passkeys have the following characteristics:

- Passkeys **protect against class 1 attacks** as no secrets are stored on the server and passkey authentication protects against phishing.
- Depending on the authenticator implementation, specifically the protection of the keys in the authenticator, they also **protect against attempts to steal the private keys** (see earlier description of class 2 attacks).
- Depending on the implementation of the user gesture that is required to unlock the key and the platform's mechanism (to ensure that only the legitimate app can trigger the use of the passkey), this provides **protection against class 3 attacks**.
- For **protection against class 4 attacks**, systems could use device-bound session credentials (DBSC) or, in the case of transactions, support of transaction confirmation[3].
- Class 5 and 6 attacks assume physical access to the authenticator and hence are not scalable. The robustness of the authenticator implementation determines the protection level[4].

The FIDO Alliance operates a FIDO Certification program for assessing and certifying the security of FIDO authenticators.

Passkey authentication in general looks like this:

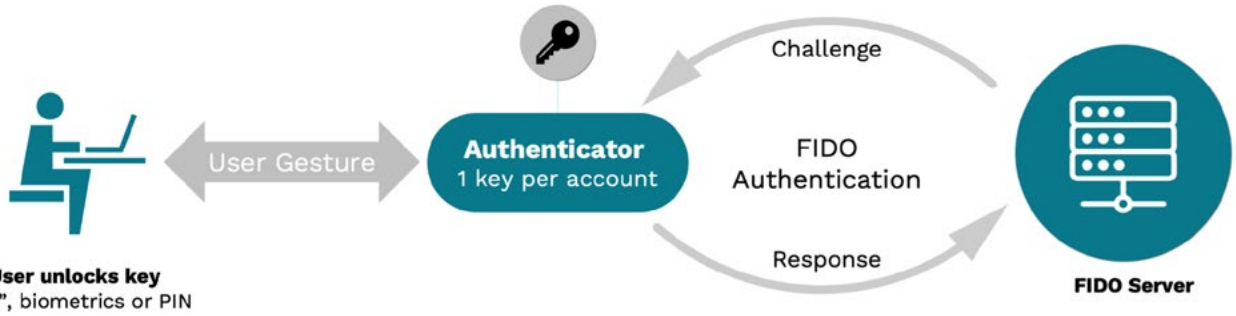


Figure 2 Passkey authentication overview

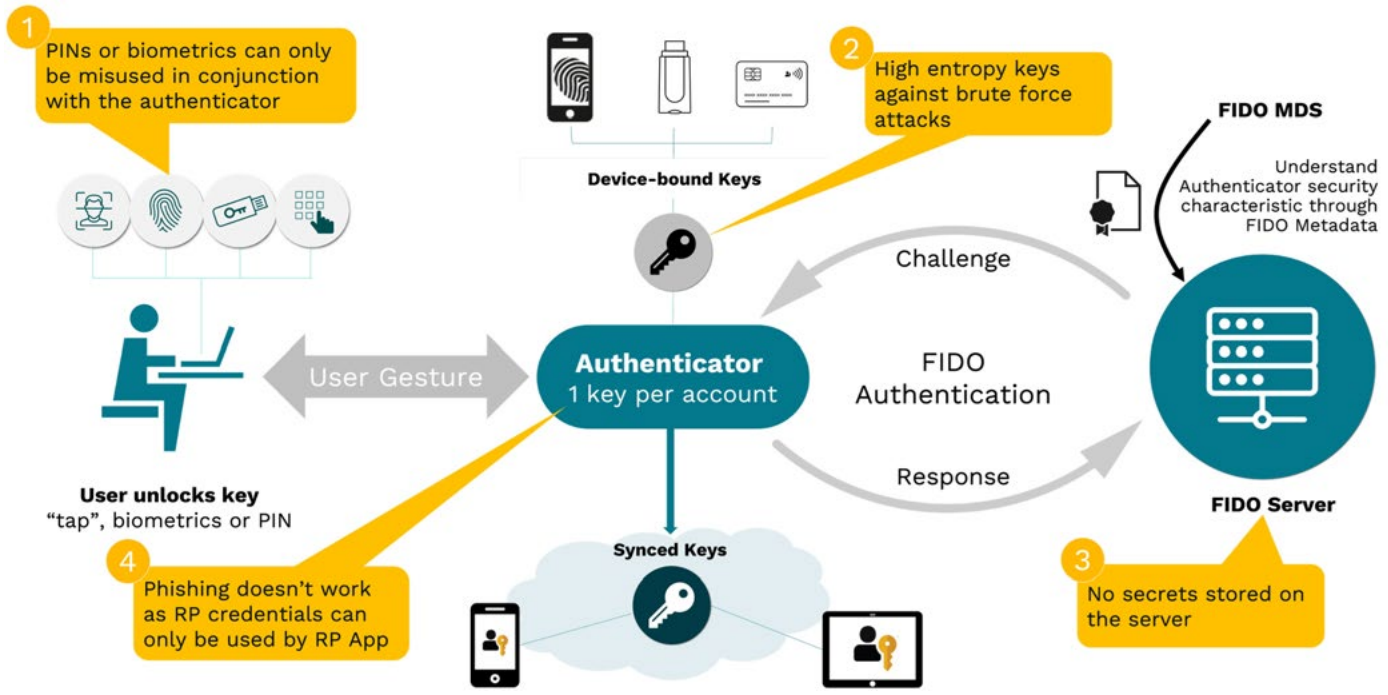


Figure 3: Passkey authentication details

The following is a high-level description:

1. The **app** fetches a random challenge from the relying party's (RP) FIDO server.
2. The platform determines the relying party identity (RP ID) from the server URL / app.
3. The **authenticator** receives the challenge and the relying party identity and waits for a user gesture.
4. The **authenticator** then signs the response using the specific key for that relying party identity. That response is sent back to the server.

No shared secrets

Traditional authentication methods like passwords and OTP use shared secrets. The relying party knows the password (or a hashed version of it) to compare against the one the user entered and the relying party also has access to the OTP or at least the seed that was used to compute the OTP in order to verify the entered OTP.

However, billions of passwords have been stolen from servers[5]. Additionally, OTP seeds have been stolen, putting millions of OTP tokens at risk[6].

In the case of passkeys, there is no shared secret. Passkeys are based on public key cryptography. The authenticator is the only entity that knows the private key. The relying party server only stores the related public key. Deriving a private key from a public key is computationally infeasible and considered impossible with current computing technology, as it requires solving complex mathematical problems.

Phishing resistance

When entering passwords, looking into cameras or touching surfaces, users are often expected to understand which app gets access to the information. If passwords are entered into a malicious app or web page, the adversary-in-the-middle (AitM) will get access to the credential.

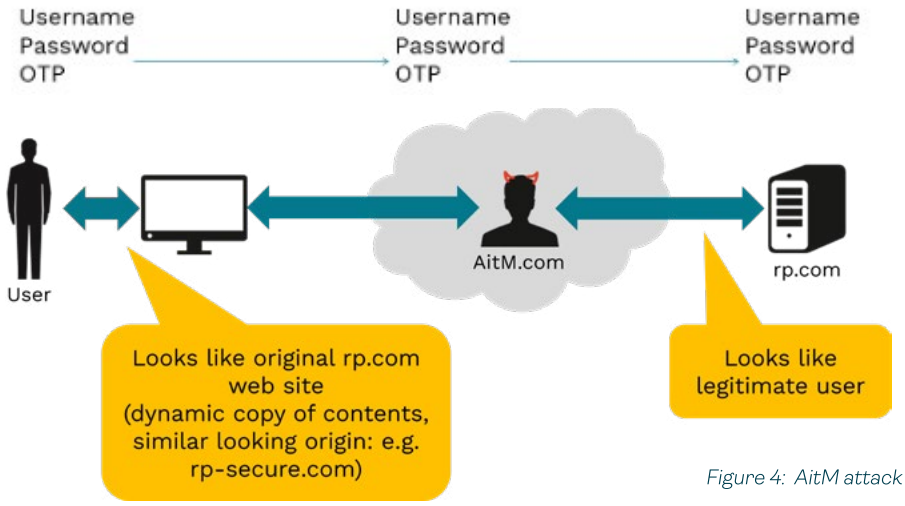


Figure 4: AitM attack

When using passkeys, the authentication works differently. The platform determines the RP ID for which a passkey needs to be used (rp.com in this example). The authenticator will then use the appropriate passkey for that RP ID, if it exists. In the case of web apps, the web browser determines the RP ID by looking at the origin of the current web page. In the case of native apps, the operating system determines the app publisher and looks up the RP ID that is related to that app publisher. If the web browser has loaded the phishing web page from aitm.com, the authenticator would attempt to use the passkey for aitm.com – not the one for rp.com. Hence it would fail.

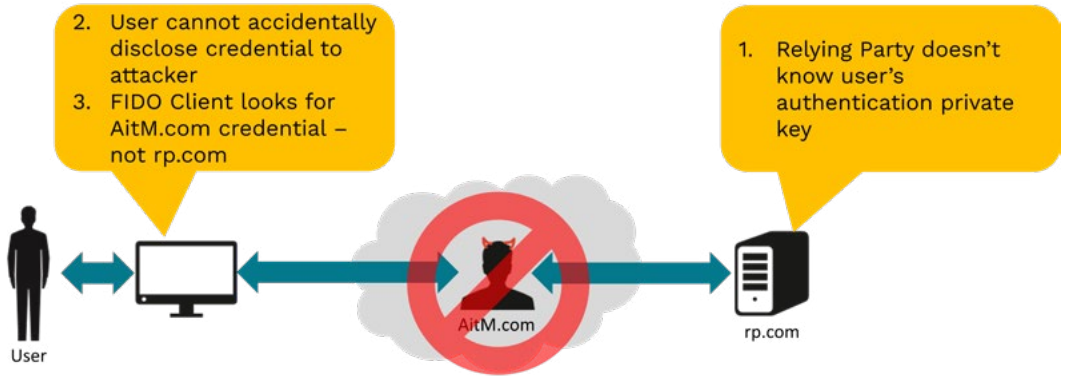


Figure 5: Passkeys protect against MitM attacks

Synced vs. device-bound passkeys

Device-bound passkeys were introduced in the first FIDO specifications. They are an effective replacement for legacy MFA. Device-bound passkeys need to be registered on each device.

Synced passkeys were introduced in 2022 as an effective method to replace passwords. Unlike device-bound passkeys, synced passkeys can be used across multiple devices much like a password. Standardized passkey export and import are recent developments[7]. Both device-bound and synced passkeys can be used together to achieve the desired security and user experience goals.

Authenticator types and security levels

Authenticator types

In typical cases, a security key would leverage a single chip "secure element" that implements sophisticated security measures; for example, against side-channel attacks, differential power analysis (DPA), decapping, etc.

On a multi-purpose computing device there might be many different options to maintain cryptographic keys. For example, in a Trustlet running in a Trusted Execution Environment[8], but also by a TPM[9] that is part of a laptop or desktop computer, either a dedicated chip or part of the main CPU[10]. Some devices even have secure elements[11] which could be used for maintaining cryptographic key material. But even when no hardware-backed security mechanisms are available, platform authenticators could use whitebox encryption to make it harder to misuse or steal cryptographic material[12].

Authenticator certification programs

There are many different FIDO authenticators available in the market. Security certification programs exist to provide transparency regarding the specific authenticator security characteristics.

The form factor (dedicated security key or part of a multi-purpose computing device) is typically less relevant than the concrete protection methods of the cryptographic material. The FIDO Authenticator Certification program specifies security requirements[13] independently of the form factor.

The FIDO Alliance operates the FIDO Metadata Service. This is a directory of known FIDO authenticators and their security characteristics[15]. The FIDO server can implement access to the FIDO Metadata Service in order to access and process authenticator security characteristics.

The FIDO protocol supports the use of authenticator attestation. This is a method to ask the authenticator for a cryptographic proof (attestation) regarding the authenticator model (Authenticator Attestation ID [AAID], Authenticator Attestation Globally Unique ID [AAGUID]). The relying party server can leverage the FIDO Metadata Service to retrieve a list of all known FIDO authenticators, their attestation root certificates, and their security characteristics. The attestation root certificates are used to verify the attestation. This approach is only meaningful if the authentication credential is always tied to that authenticator (and cannot be exported and imported to another device).

	HW & SW Requirements	Defend against	Implementation examples
L1	Any device HW or SW	L1 prevents against phishing and the majority of scalable attacks with software and security best practices.	L1 is the by-default security level required for any functional certification.
L2	Device must support allowed ROE (e.g. TEE, Secure Element...), as listed here.	L2 authenticators with a hardware protected border (AROE), protecting against remote software attacks.	Within the list of Allowed ROE: • Security Key (BLE/NFC/USB) • TEE based on ARM Trustzone • TEE Based on Intel VT HW...
L3	Device supported by an AROE with security resistance against physical attacks.	L3 authenticators with a hardware protected border (AROE), protecting against remote software attacks and local hardware attacks.	GlobalPlatform certified TEE (L3 GlobalPlatform Companion Program) CC certified Secure Element (L3 CC Companion Program)

Figure 6: FIDO-certified authenticator levels[14]

Risk analysis

Passkeys in general are significantly more secure than passwords and OTPs. The use of any passkey provides protections against many scalable attacks, including server-side credential stealing and phishing attacks. However, there are security differences in the different passkey types and authenticator implementations.

Synced passkeys

Synced passkeys are classified by NIST SP 800-63B as AAL2[16]. (section B.3).

The passkey provider, which facilitates the ability to backup and restore passkeys from and to the device, handles the device management and provides a credential with an indefinite lifetime. The device management is essentially outsourced to the passkey provider.

Consider the following:

(1) Protection at rest in the cloud. As a result, passkeys are typically also stored in a cloud system, so the effective protection level for the key at rest in the cloud is relevant[17].

(2) Identity assurance for recovery. When users lose access to their account at the passkey provider (e.g., password manager), the passkey provider might support recovery flows. A recovery flow could be as simple as sending an email with a magic link to the email address the user configured, or involve an SMS-OTP, or even require the legitimate user to present a valid passport or ID card (or verifiable credential). However, the method varies by passkey provider.

(3) Protection in transit. Syncable passkeys can be backed up and restored to a new device. That typically means that the private keys will be transmitted over the internet. The passkey provider could send the private keys over an encrypted transport layer security (TLS) connection, they could also encrypt the private key before it is being sent over TLS, and they could implement sophisticated schemes to control the exchange of the encryption/decryption keys[18]. Alternatively, the passkey provider could always keep the keys in a secured cloud system[19] – avoiding any transit of private keys.

(4) Protection at rest on the device. When passkeys are stored on the device, different protection methods for the private keys could be used. The key could be stored in the clear, it could be imported into a key store leveraging the trusted execution environment (TEE), or it could even be imported into a secure element. The method depends on the passkey provider and is often not even published.

(5) Sharing with friends. In some cases, it is possible to share a passkey with a friend[20]. This is conceptually similar to sharing passwords with friends. However, it typically means that the friend can now also use, potentially export, or even share the passkey with other friends. For the relying party it means that a direct or indirect friend could now authenticate using their own device.

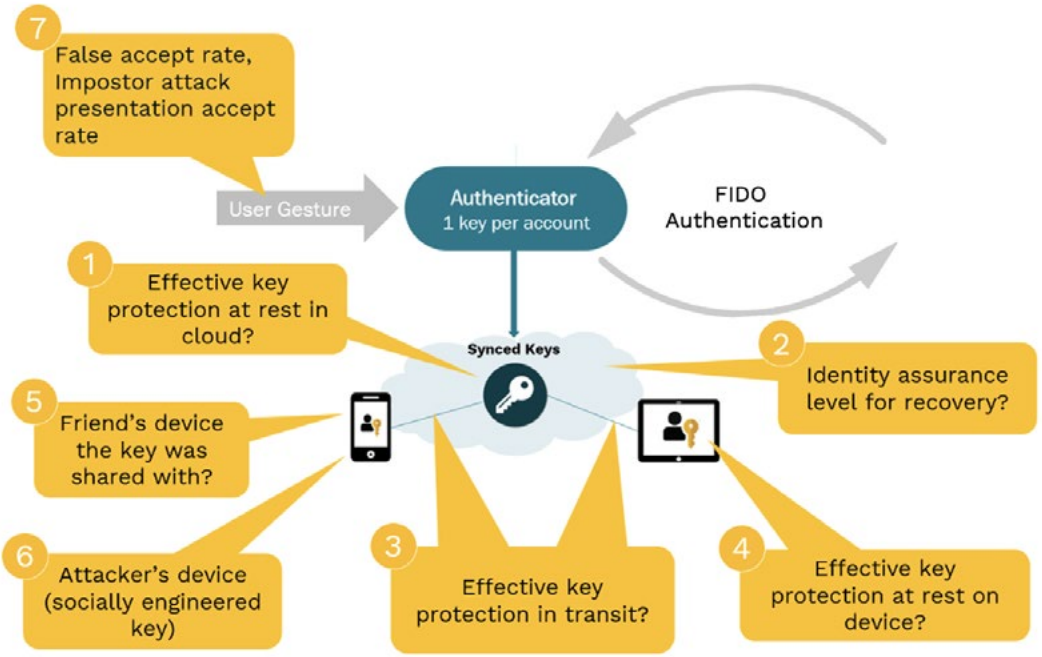


Figure 7: Synced passkey attack surface

(6) Social engineering. Many passkey providers also support the export and import of passkeys into other passkey providers. This function allows users to move their credentials to another passkey provider if they want. However, it also opens the door for social engineering attacks. An attacker could try to convince users to share the exported passkey with them.

(7) Impostor attack. Sometimes phones, tablets, or laptops are left unattended for a while, get lost, or are stolen. The only method that prevents an adversary from misusing the passkey is the user gesture. If that gesture is just a test of user presence, for example a button to press, that is easy to do. But if user verification through a PIN or a biometric is required, that hurdle is much higher. Especially when retry counters and sophisticated anti-spoofing methods are implemented.

Device-bound passkeys

Device-bound passkeys can meet the NIST SP 800-63B requirements for AAL3[21], when having FIPS 140 certification at the appropriate level and when user verification is performed.

The attack surface of device-bound passkeys is significantly smaller:
Since keys are generated and maintained in the authenticator (meaning there is no need/ability to create backups), they are typically exclusively stored in trusted execution environments, often through a key store or in secure elements.

Note that the lifetime of device-bound passkeys is limited by the lifetime of the related device. The relying party is responsible for managing multiple credentials for multiple devices with appropriate binding and potentially even ID proofing processes (in the case of account recovery).

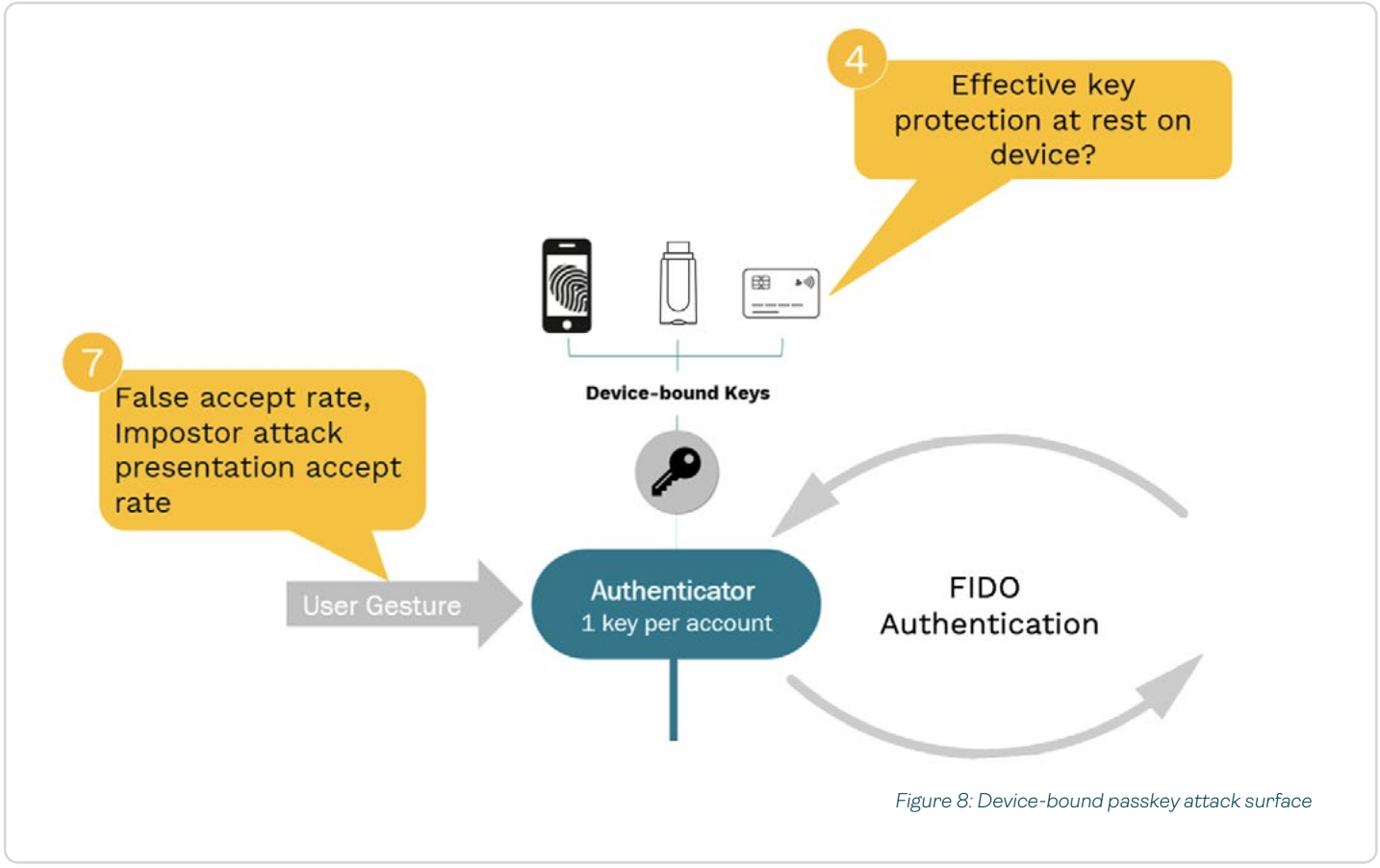


Figure 8: Device-bound passkey attack surface

Endorsements and research

FIDO specifications and certification programs have been designed with security and usability in mind. The security has evolved over time and many government agencies and security researchers have investigated the security of passkeys.

Examples of government agencies that endorse FIDO

1. CISA: Implementing Phishing Resistant MFA[22].

a. "Phishing-resistant MFA is the gold standard for MFA."

b. "The only widely available phishing-resistant authentication is FIDO/WebAuthn authentication."

c. "Push bombing, SS7, and SIM swap attacks are not applicable [to FIDO."

2. CISA, FBI, EPA, and DOE: Primary Mitigations to Reduce Cyber Threats to Operational Technology[23]**: Secure remote access to OT networks**

a. "Many critical infrastructure entities, or contractors working on their behalf, make risk-based tradeoffs when implementing remote access to OT assets. These tradeoffs deserve careful reevaluation. If remote access is essential, upgrade to a private IP network connection to remove these OT assets from the public internet and use virtual private network (VPN) functionality with a strong password and **phishing-resistant multifactor authentication** (MFA) for user remote access."

3. CISA: Mobile Communications Best Practice Guidance[24].

a. "Apply these best practices to your devices and online accounts:

i. [...]

ii. Enable Fast Identity Online (FIDO) phishing-resistant authentication. FIDO authentication uses the strongest form of MFA and is effective against MFA bypass techniques. Where feasible, hardware based FIDO security keys, [...], are the most effective; however, FIDO [syncable] passkeys are an acceptable alternative."

4. US Executive Office of the President: Memorandum M-22-09[25]**:**

a. "For agency staff, contractors, and partners, phishing-resistant MFA is required."

b. "For public users, phishing-resistant MSA must be an option."

5. ENISA : NIS2 Technical Implementation Guideline[26]**. (pages 141-146)**

a. 11.6.2

i. "The use of phishing-resistant MFA is recommended. Below is a list of currently available solutions ordered from strongest to weakest.

1. 'Strong':

a. phishing-resistant:

i. - no shared secrets, not vulnerable to attacker-in-the-middle;

ii. - protected cryptographic private key that can be securely registered to:

1. a domain, in accordance with Fast Identity Online (FIDO) and W3C WebAuthn standards;

2. a trust provider, following public key infrastructure and International Telecommunication Union X.509 standards.

2. 'Medium' MFA, for example:

a. push notification, number matching or application based.

3. 'Last resort' MFA, for example:

a. text message or email OTP"

b. 11.7.1 Guidance "Select appropriate MFA methods and continuous authentication mechanisms based on the entity's security needs [...]. It is also a good practice to consider user convenience when selecting an implementing a solution:

i. [...]

ii. [syncable] Passkeys

iii. Fast Identity Online 2 security keys

iv. [...]"

c. 11.7.2 Guidance: [...] "Wherever possible, use phishing-resistant MFA."

Security research

Multiple security researchers have analyzed the security of the FIDO protocol using formal methods. Examples:

1. Formal Analysis of the FIDO 1.x Protocol[27]**:**
a. "We show that even if we corrupt the Server, or the Client (but never both of them), there is no possible attack."
b. "Our ProVerif analysis shows that, when the verification [application identity of the caller (RP ID)] is performed, the expected authentication properties are satisfied."

2. Formal Verification of the W3C Web Authentication Protocol[28]**:**
"[...] the W3C Web Authentication protocol is secure and so ensures strong user authentication"

3. Provable Security Analysis of FIDO2[29]**:**
"[...] our proof confirms the authentication security of WebAuthn".

Comparison with legacy methods

Authentication is often based on passwords and sometimes an OTP is required as a second factor. Both methods are subject to phishing (AitM/MitM) attacks.

Characteristics	Passkeys	Passwords	OTPs
Resistant against stealing credentials from servers	Yes – only public keys stored on the server	No – billions of passwords have been leaked[30].	No – the seeds for OTP tokens have been stolen from the server[31].
Resistant against phishing	Yes	No	No
Security characteristic known	Yes, if authenticator attestation is used and it is a device-bound key. That works in BYOD scenarios and scenarios in which the passkey is pre-provisioned on the security key. Provider indication can be used to understand the provider of a synced passkey.	No. Approximately 50% of users use either password managers[32], their web browsers to manage their passwords, or write their passwords down in electronic documents or on paper. So it is even unclear whether the password can be considered a knowledge factor.	No, in the case of a bring-your-own (OTP) device, the security characteristic of the OTP token or the authenticator app is unknown. In the case of SMS-OTP, the mobile-network operator (MNO) manages the devices that have access to the text messages (SMS). There is no easy way to determine the MNO, nor do MNOs publish their security measures.
Dependency on third parties	**Device-bound keys:** No. Keys are generated in the authenticator and never exported. **Synced passkeys:** Yes. The passkey provider ("Sync-Fabric"). They implement ID proofing when users want to restore their passkeys, they implement measures to protect the keys at rest and in transit.	Yes. The user might share password with friends or use third-party password managers.	**SMS-OTP:** Yes, the MNO. SIM-swap attack protection relies on the MNO. SS7 attack protection relies on the MNO. **OTP tokens:** Yes, seeds could be stolen from the provider

OneSpan products

Digipass® S3 Authentication Software supports authenticator attestation and provider indications. Its **Intelligent Passwordless Authentication** feature adapts the strength of the authenticator and the context of the authentication.

For example:

1. Detecting whether an authenticator that was used meets NIST SP 800-63 AAL2 or even AAL3 requirements.

2. Triggering step-up authentication when a passkey that was created in a passkey provider that doesn't meet the security requirements is used the first time on a new device. (It could have been shared with a friend, socially engineered by an adversary, etc.)

3. Determining whether it is a good time to suggest passkey creation via **Digipass Smart Sense** to a user who signed in using a legacy authentication method.

With **Digipass S3 Authentication Software**, organizations can combine synced and device-bound passkeys, simplifying secure device migration while ensuring strong device binding.

At the core, the **Digipass S3 Server** leverages FIDO Metadata Statements to verify authenticator attestation and apply adaptive authentication policies based on the security characteristics of each authenticator.

OneSpan's **Digipass FIDO2 security keys** are FIDO certified and provide the highest security level through the implementation of device-bound passkeys protected by dedicated security hardware. Additionally, they may be connected to multiple devices, from smartphones to laptops and desktop PCs.

References

1. "Why Scalable Attacks Matter", publication in DuD 4/2016, January 2016, https://www.springerprofessional.de/en/avoiding-the-tsunami/10162734
2. https://en.wikipedia.org/wiki/Decapping
3. https://fidoalliance.org/wp-content/uploads/2020/08/FIDO-Alliance-Transaction-Confirmation-White-Paper-08-18-DM.pdf
4. https://fidoalliance.org/certification/authenticator-certification-levels/
5. Billions of Password Leaked, https://time.com/7296254/passwords-leaked-data-breach/
6. Full Story of RSA Hack, https://www.wired.com/story/the-full-story-of-the-stunning-rsa-hack-can-finally-be-told/
7. Credential Exchange Protocol, https://fidoalliance.org/specs/cx/cxp-v1.0-wd-20241003.html
8. TrustZone for Cortex A, https://www.arm.com/technologies/trustzone-for-cortex-a/tee-and-smc
9. Trusted Platform Module (TPM), https://trustedcomputinggroup.org/work-groups/trusted-platform-module/
10. What you always wanted to know about TPM, https://www.tuxedocomputers.com/en/Infos/Help-Support/Frequently-asked-questions/What-you-always-wanted-to-know-about-TPM.tuxedo
11. What is a secure element and why should you care, https://www.tropicsquare.com/blogs/what-is-a-secure-element-and-why-should-you-care
12. Whitebox Cryptography, https://www.whiteboxcrypto.com/
13. FIDO Security Requirements, https://fidoalliance.org/specs/fido-security-requirements/fido-authenticator-security-requirements-v1.5-fd-20211102.html
14. Table sourced from https://fidoalliance.org/certification/authenticator-certification-levels/
15. FIDO Metadata Service, see https://fidoalliance.org/metadata/
16. NIST SP 800-63B-4, https://pages.nist.gov/800-63-4/sp800-63b.html#appB
17. "Why Scalable Attacks Matter", publication in DuD 4/2016, January 2016, https://www.springerprofessional.de/en/avoiding-the-tsunami/10162734
18. Apple Platform Security, see https://help.apple.com/pdf/security/en_US/apple-platform-security-guide.pdf
19. Dashlane Cloud Passkeys, https://www.dashlane.com/de/blog/cloud-passkeys
20. Air-drop passkeys, https://support.apple.com/en-am/guide/iphone/iph0dd1796bb/ios
21. NIST SP 800-63B-4, https://pages.nist.gov/800-63-4/sp800-63b.html#appB
22. CISA Implementing Phishing Resistant MFA, https://www.cisa.gov/sites/default/files/2023-01/fact-sheet-implementing-phishing-resistant-mfa-508c.pdf
23. https://www.ic3.gov/CSA/2025/250506.pdf
24. CISA 'Mobile Communications Best Practice Guidance' https://www.cisa.gov/sites/default/files/2024-12/guidance-mobile-communications-best-practices.pdf
25. US Executive Office of the President, Memorandum M-22-09, https://www.whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf
26. ENISA 'NIS2 Technical Implementation Guideline' https://www.enisa.europa.eu/publications/nis2-technical-implementation-guidance
27. Formal Analysis of the FIDO 1.x Protocol, https://www.researchgate.net/publication/323234868_Formal_Analysis_of_the_FIDO_1x_Protocol
28. Formal verification of the W3C web authentication protocol, https://dl.acm.org/doi/10.1145/3190619.3190640
29. Provable Security Analysis of FIDO2, https://eprint.iacr.org/2020/756.pdf
30. Billions of Password Leaked, https://time.com/7296254/passwords-leaked-data-breach/
31. Full Story of RSA Hack, https://www.wired.com/story/the-full-story-of-the-stunning-rsa-hack-can-finally-be-told/
32. https://www.security.org/digital-safety/password-manager-annual-report/

About OneSpan

OneSpan is a global leader in digital security, trusted by thousands of enterprises across 100+ countries—including more than 60% of the world's 100 largest banks—to safeguard digital accounts, secure financial transactions, and prevent fraud. Our award-winning solutions provide passwordless authentication, digital transaction security, and advanced mobile application protection, helping organizations meet the highest security standards and global compliance requirements. As cyber threats grow more sophisticated, OneSpan delivers cutting-edge technology to safeguard customers, mitigate risks, and ensure trust in every digital interaction.

Learn more at
OneSpan.com/security

Contact us at
OneSpan.com/contact-us