



# PassKeys einrichten bei WordPress WooCommerce

**was muss ich als Wordpress-Nutzer (für Shop-Webaufritt) tun, damit meine Kunden sich auf meiner Website zusätzlich zur Passwortlegitimierung per Passkey einloggen können?**

Damit deine Kund:innen sich zusätzlich zur klassischen Passwort-Authentifizierung auch per **Passkey** (also WebAuthn / FIDO2 — z. B. Face ID, Touch ID, Sicherheitsschlüssel etc.) auf deiner WordPress-Shopseite einloggen können, sind ein paar technische Voraussetzungen und Konfigurationsschritte nötig. Ich führe sie hier auf — wenn du möchtest, kann ich auch ein Plugin vorschlagen und zeigen, wie man es Schritt für Schritt einrichtet.

## 1 Was sind Passkeys / WebAuthn

Kurz zur Einordnung:

- Passkeys sind moderne Authentifizierungs-Credentials, die auf kryptographischen Schlüsselpaaren beruhen (öffentlicher/privater Schlüssel).
- Der private Schlüssel bleibt beim Gerät oder in einer Hardware/Biometrie-Komponente, der öffentliche Schlüssel wird auf deinem Server registriert.
- Der Vorteil: Passwort-Diebstahl / Phishing wird deutlich erschwert.

### 1.1 Voraussetzungen

Damit Passkeys zuverlässig funktionieren, müssen ein paar Rahmenbedingungen erfüllt sein:

- HTTPS**  
Dein Shop muss über eine gesicherte Verbindung laufen (SSL/TLS). Ohne https funktioniert WebAuthn / Passkey nicht korrekt.
- PHP-Version & Server-Komponenten**  
Die meisten Plugins benötigen bestimmte PHP-Versionen (häufig mindestens PHP 7.4 oder höher). Außerdem sollten Erweiterungen wie mbstring, gmp, evtl. sodium installiert sein.
- Browser- & Geräteunterstützung**  
Die Nutzer:innen brauchen ein Gerät / Browser, der Passkeys / WebAuthn unterstützt (z. B. moderne Browser wie Chrome, Firefox, Safari; Geräte mit Biometrie oder Sicherheitsschlüssel).
- Kompatibilität mit deinem Shop-Setup**  
Wenn du z. B. WooCommerce nutzt, oder ein Membership-Plugin, musst du sicherstellen, dass das Passkey-Plugin dort mitspielt (z. B. Login-Formulare von WooCommerce).

### 1.2 Mögliche Plugins / Lösungen

Hier sind einige WordPress-Plugins, die du nutzen kannst:

Plugin	Funktionen / Vorteile	Hinweise
<b>WP-WebAuthn</b>	Ermöglicht Authentifizierung über FIDO2 und U2F (Passkeys, Gesicht, Fingerabdruck etc.). Unterstützt auch „namenlose Authentifizierung“. ( <a href="https://wordpress.com">WordPress.com</a> )	PHP-Erweiterungen wie gmp und mbstring erforderlich. HTTPS erforderlich. ( <a href="https://wordpress.com">WordPress.com</a> )
<b>Multidots Passkey Login</b>	Vollständige passkey-basierte Anmeldung (Touch ID, Face ID, Sicherheitsschlüssel), Benutzerregistrierung eines Passkeys, Admin-Steuerung. ( <a href="https://wordpress.com">WordPress.com</a> )	Funktioniert auch für bestehende WordPress-Nutzer, nach Passkey-Registration. ( <a href="https://chooseplugin.com">ChoosePlugin.com</a> )
<b>Secure Passkeys</b>	Integration von Passkeys / WebAuthn, inklusive Verwaltung durch den Nutzer im Profilscreen, unterstützt WooCommerce etc. ( <a href="https://wordpress.org">WordPress.org</a> )	

### 1.3 Schritt-für-Schritt: Wie du Passkeys zusätzlich aktivierst



Hier ein genereller Ablauf, wie du vorgehen kannst:

## 1. Backup & Testumgebung

Bevor du Änderungen machst, sichere dein WordPress + Datenbank. Wenn möglich, teste erst in einer Staging-/Entwicklungsumgebung.

## 2. Plugin auswählen & installieren

Beispiel: "WP-WebAuthn" oder "Multidots Passkey Login".

→ Im WordPress-Backend → Plugins → Neu hinzufügen → nach dem Namen suchen → installieren und aktivieren.

## 3. Plugin konfigurieren

- Prüfe in den Einstellungen des Plugins, ob alle Anforderungen erfüllt sind (HTTPS, PHP-Erweiterungen vorhanden).
- Lege fest, ob das Passkey-Login zusätzlich zum Passwort-Login dienen soll oder ob das Passwort abgeschafft wird (falls das Plugin das erlaubt).
- Anzahl der erlaubten Passkeys pro Nutzer/regeln ggf. festsetzen.

## 4. Benutzerregistrierung eines Passkeys

Nach Aktivierung sollte in den Profileinstellungen der Nutzer eine Möglichkeit erscheinen, einen Passkey zu registrieren (z. B. Gerät hinzufügen, Sicherheitsschlüssel verbinden etc.).

## 5. Frontend-Login anpassen

- Es wird ein Login-Formular nötig sein, das Passkeys unterstützt. Viele Plugins bieten Shortcodes oder Gutenberg-Blöcke, mit denen du Buttons wie „Login mit Passkey“ oder „Passkey registrieren“ einfügen kannst.
- Stelle sicher, dass auch WooCommerce Login-Formulare etc. angepasst sind.

## 6. Fallback / Sicherheit

- Falls ein Nutzer keinen Passkey nutzen kann oder will, sollte weiterhin das klassische Passwort verfügbar sein (oder zumindest ein anderer gesicherter Fallback).
- Denk an Sicherheitsmaßnahmen: Login-Versuche begrenzen, ggf. Zwei-Faktor-Authentifizierung, Logging etc.

## 7. Testlauf & Nutzer informieren

- Teste mit mehreren Geräten und Browsern, dass Anmeldung und Passkey-Registrierung gut funktionieren.
- Kommuniziere an deine Kund:innen, wie sie Passkeys nutzen können (z. B. Anleitungen / Hilfe auf der Website).

Wenn du willst, kann ich dir **ein Plugin empfehlen**, das besonders zuverlässig funktioniert für WooCommerce-Shops, und eine Schritt-für-Schritt-Anleitung speziell dafür. Möchtest du das?

**ja bitte**



Hier ist eine Plugin-Empfehlung + eine Schritt-für-Schritt-Anleitung, wie du Passkey / WebAuthn (z. B. Touch ID, YubiKey etc.) zusätzlich zur Passwortauthentifizierung in deinem WooCommerce-Shop einsetzt:

#### 1.4 Plugin-Empfehlung: **Secure Passkeys**

##### Warum Secure Passkeys?

- Es unterstützt **WooCommerce-Login-Seite** direkt. ([WordPress.org](https://WordPress.org))
- Es erlaubt, dass Nutzer:innen in ihrem Profil Passkeys registrieren und verwalten. ([WordPress.org](https://WordPress.org))
- Es ist relativ „jung“ und wurde zuletzt noch aktiv gepflegt. ([WordPress.com](https://WordPress.com))
- Es ist als freies Plugin verfügbar, sodass du erst testen kannst, bevor du evtl. auf Premiumfunktionen aufrüstest. ([WordPress.org](https://WordPress.org))

Ein bekanntes alternatives Plugin ist **WP-WebAuthn**. Dieses kann auch genutzt werden, aber beachte, dass es in der Vergangenheit Sicherheitslücken (Stored XSS via Shortcodes) hatte, die in neueren Versionen behoben wurden. ([wordfence.com](https://wordfence.com))

#### 1.5 Schritt-für-Schritt: Secure Passkeys im WooCommerce-Shop einrichten

Diese Anleitung geht davon aus, dass bereits WooCommerce läuft und du Zugriff auf das WordPress-Backend hast.

##### 1.5.1 1. Vorbereitungen

- **Sicherung** machen (Dateien + Datenbank), bevor du neue Plugins installierst.
- Stelle sicher, dass deine Seite via **HTTPS / SSL** erreichbar ist (WebAuthn / Passkey setzt sichere Verbindung voraus).
- Prüfe die PHP-Version und verfügbare Erweiterungen – Secure Passkeys verlangt in der Regel **PHP 7.4 oder höher**. ([WordPress.org](https://WordPress.org))

##### 2. Plugin installieren & aktivieren

- Im WordPress-Dashboard → **Plugins** → **Installieren**
- Suche nach „Secure Passkeys“
- Klicke „Installieren“ → „Aktivieren“

##### 3. Grundeinstellungen konfigurieren

- Nach Aktivierung findest du in den Einstellungen (z. B. **Einstellungen** → **Passkeys** oder **Sicherheit** → **Passkeys**) einen Bereich für das Plugin.
- Lege fest, ob Passkey-Login zusätzlich zum Passwort erlaubt ist oder ob manche Nutzergruppen (z. B. Kunden) Pflicht haben.
- Lege Limits fest: wie viele Passkeys darf ein Nutzer registrieren (z. B. mehrere Geräte, Sicherheits-Keys etc.).
- Falls das Plugin Funktionen anbietet wie „Passkey löschen / zurücksetzen“, aktiviere sie, damit Nutzer Geräte verwalten können.

##### 4. Integration in Login & Registrierungsseiten

- Secure Passkeys integriert sich laut Pluginbeschreibung in **WooCommerce Login-Seite** direkt. ([WordPress.org](https://WordPress.org))
- Falls nötig, füge Shortcodes oder Gutenberg-Blöcke des Plugins in deine Login- oder Registrierungsseiten ein, zum Beispiel „Passkey registrieren“ oder „Login mit Passkey“.
- Achte darauf, dass dein Theme / deine Login-Formulare kompatibel sind – ggf. CSS-Anpassungen, damit das Aussehen passt.

##### 5. Nutzer:innen die Registrierung eines Passkeys ermöglichen

- Sobald das Plugin läuft, sollte in jedem Benutzerprofil (für Kunden) ein Bereich erscheinen, um **Passkeys zu registrieren** (z. B. „Gerät hinzufügen / Sicherheitsschlüssel registrieren“).
- Der Nutzer klickt auf „Neues Authentifizierungsgerät hinzufügen“, wählt z. B. Fingerabdruck, Sicherheitsschlüssel etc., und bestätigt via WebAuthn API.



- 
- Danach steht der Passkey in seiner Liste sicherer Geräte.

#### 6. Testen

- Teste mit verschiedenen Geräten (Laptop, Smartphone) und Browsern (Chrome, Firefox, Safari), ob:
  1. Ein Passkey erfolgreich registriert werden kann.
  2. Der Login mit Passkey funktioniert (neben / zusätzlich zum Passwort).
  3. Fallback funktioniert (z. B. falls das Gerät nicht verfügbar ist).
  4. Das Löschen / Zurücksetzen eines Passkeys funktioniert.
- Teste auch im Incognito / Privatmodus, ob keine unerwarteten Fehler auftreten.

#### 7. Nutzerkommunikation & Support

- Informiere deine Kund:innen über diese neue Login-Option: was ist ein Passkey, wie registriert man einen, welche Vorteile (Sicherheit / Komfort).
- Erstelle ggf. eine Hilfeseite oder FAQ („Wie logge ich mich mit Passkey ein?“).
- Biete Support an, falls ein Nutzer seinen Passkey verliert – z. B. über E-Mail-Verifizierung oder Passwort-Fallback.

---

#### 1.6 Sicherheit und Fallstricke

- Halte das Plugin und WordPress stets auf aktuellen Versionen, damit bekannte Sicherheitslücken geschlossen sind. (WP-WebAuthn hatte beispielsweise XSS-Lücken in älteren Versionen) ([wordfence.com](https://wordfence.com))
- Biete **Fallback-Mechanismen**: Nicht alle Nutzer:innen werden Passkeys nutzen können (z. B. ältere Geräte). Stelle sicher, dass das Passwort-Login weiter existiert oder Alternativen (z. B. Zwei-Faktor via App) möglich sind.
- Begrenze Login-Versuche, benutze Logging / Monitoring, um Missbrauch zu erkennen.

---

Wenn du willst, kann ich dir eine **Video-Anleitung** oder **Screenshots** speziell für Secure Passkeys + WooCommerce herausuchen, damit du visuell geführt wirst – soll ich das tun?



**was kostet "Secure Passkeys" und wie lange benötigt der geübte Web-Admin zur Implementierung?**

Ich habe nachgeguckt — das Plugin *Secure Passkeys* wird aktuell **kostenlos** angeboten als Open-Source-Plugin über das WordPress-Plugin-Verzeichnis. ([WordPress.org](https://wordpress.org/plugins/secure-passkeys/))

- 1.6.1      ⚠ Hinweis zur Sicherheit
- Es gibt eine veröffentlichte Sicherheitslücke (CVE-2025-10305) für Versionen  $\leq 1.2.1$  von *Secure Passkeys*, durch die Benutzer mit Benutzerrolle Subscriber+ Passkeys sehen oder löschen konnten, ohne Berechtigung. Diese wurde in neueren Versionen des Plugins adressiert. ([CVE Details](#))
  - Wichtig: Stelle sicher, dass du die neueste Version nutzt.

1.6.2      ⌚ Wie viel Zeit braucht ein geübter Web-Admin zur Implementierung?

Abhängig von Erfahrung, der Komplexität deiner WordPress/WooCommerce-Installation und davon, wie viele Anpassungen du willst. Hier ein grober Richtwert:

Phase	Aufgaben	Geschätzter Zeitaufwand
Vorbereitung	Sicherung von Daten & Dateien, SSL prüfen, PHP & Server-Voraussetzungen checken	ca. <b>0,5–1 Stunde</b>
Plugin installieren & aktivieren	Plugin herunterladen/über Backend installieren	ca. <b>5–10 Minuten</b>
Grundeinstellungen konfigurieren	Optionen wie Rollen, Limits für Passkeys, Darstellung, Login-Formulare, Shortcodes prüfen	ca. <b>20–30 Minuten</b>
Integration in WooCommerce-Login & Profilseiten	Theme-Kompatibilität prüfen, ggf. CSS oder Templateanpassungen, Profilseite für Passkeys aktivieren	ca. <b>30 Minuten bis 1 Stunde</b>
Tests mit verschiedenen Geräten/Browsers	Registrierung und Login mit Passkey, Fallback testen, Profilverwaltung testen	ca. <b>30 Minuten</b>
Nutzerhinweise / Dokumentation intern oder öffentlich	Anleitung für Kunden, Kommunikations-Texte etc.	ca. <b>15–30 Minuten</b>

**Gesamtschätzung bei geübtem Admin:** etwa **1,5 bis 3 Stunden**, sofern keine größeren Anpassungen nötig sind oder Probleme auftreten.

Wenn du möchtest, kann ich schauen, ob es Premium-Features oder Add-ons gibt, die eventuell Kosten verursachen, damit du weißt, ob später weitere Ausgaben auf dich zukommen könnten?