Desktop Passwordless Deployment Guide: Windows Login, Shared Workstations, and Clean Rooms

Desktop & Endpoints / Last updated 2026-06-11 / https://www.scrambleid.com/learn/desktop-deployment-guide

In one sentence: Replace workstation passwords with device-bound keys (Windows Hello today; Touch ID as macOS support lands) while keeping AD/LDAP governance, supporting shared workstations, and maintaining SIEM-grade audit trails.

Goal: eliminate workstation passwords while keeping AD/LDAP governance, strong cryptographic proof, and SIEM-grade audit.

TL;DR (canonical)

- Desktop binds a device keypair to a user (SUID) and device (ZID) using the OS keystore (TPM/Windows Hello, Secure Enclave/Keychain).
- Login can be same-device (Windows Hello) or cross-device assist (signed QR QID scanned by a trusted device).
- Shared workstations should target fast swaps (seconds) and deterministic states.
- Offline access windows are possible but should be time-boxed and tightly audited.

What platforms does this guide cover?

Windows workstations today; macOS support is planned, and web login with the mobile app covers Mac users in the meantime. Linux desktops are not supported; web login covers them too. If your environment requires FIPS-validated cryptography, ask your ScrambleID account team for the current cryptographic module validation status before you commit to a deployment design.

When is ScrambleID Desktop the right choice?

Deploy ScrambleID Desktop when you need passwordless login and:

- contact centers / clean rooms restrict mobile devices
- shift-based shared workstations need rapid tap-in/tap-out

- regulated environments want strong identity without typing passwords

---

How does ScrambleID Desktop work?

1. Desktop app generates a device-bound keypair and stores the private key in the OS-protected keystore.
2. Public key registers with ScrambleID and binds to the user (SUID) and device (ZID).
3. A realtime channel (e.g., WebSocket) coordinates login session state.

Two login patterns:

Pattern A – Same-device platform authentication

> **Key lifecycle detail:** enrollment, rotation overlap windows, revocation propagation SLAs, and the difference between Windows Hello (local) vs Windows Hello for Business (enterprise-managed, AD-bound) are specified in Device key lifecycle in the architecture reference.

- Windows Hello unlocks the desktop key.
- Desktop signs a ScrambleID challenge and exchanges an assertion.
- Workstation login completes (domain session can start without passwords).

Pattern B – Cross-device assist (QR)

- Login screen shows a signed QID that encodes a DID and certificate thumbprint.
- Trusted mobile device scans, verifies signature, and confirms.
- Desktop receives completion via realtime channel.

What does a successful Desktop deployment look like?

Use measurable targets, calibrated against your own pilot baseline:

- successful workstation logins: a near-perfect success rate, tracked on a moving window
- login time: fast enough that users don't reach for workarounds; track median and p95 against the pilot baseline
- helpdesk password reset tickets: material reduction for pilot cohorts
- shared stations: swap time fast enough for shift handoffs; track the median

What is the fleet-scale deployment checklist?

1) Prerequisites

Windows

- Windows Hello configured (if using Pattern A)
- domain sign-in expectations validated (Kerberos)

If you use Windows Hello for Business (WHfB) at scale, validate WHfB policy and enrollment during your pilot, Desktop Pattern A depends on the same underlying platform authenticators.

macOS (planned; validate when support lands)

- Touch ID / local user verification policy validated
- MDM posture confirmed

Identity

- confirm provisioning source for users (directory/SCIM/HR feed)
- confirm device lifecycle rules (enroll/revoke)

2) Decide policy upfront

- allow cross-device QR assist? (yes/no)
- allow offline login window? (yes/no, TTL)
- shared workstation mode? (yes/no)

3) Silent install examples

> Exact installer flags vary by release; the examples below show standard patterns.

Windows MSI (example)

```
msiexec /i ScrambleIDDesktop.msi /qn ORG_ID="org_123" ENROLLMENT_CODE="enroll_abc" AUTO_UPDATE=1
```

macOS PKG (example, for the planned macOS release)

```
sudo installer -pkg ScrambleIDDesktop.pkg -target /
```

4) MDM distribution (patterns)

- Intune: Win32 app + detection rule + assignment to device group
- Jamf: package + configuration profile

Microsoft Intune (Windows), Win32 app packaging quickstart

If you manage Windows endpoints with Intune, treat ScrambleID Desktop like any other Win32 line-of-business (LOB) app:

1. Package into **.intunewin**
2. Configure **install/uninstall** commands (silent)
3. Configure **detection rules**
4. Assign to **device groups** (pilot → staged rollout)

Detection rules matter because they prevent reinstall loops.

Typical detection rule types Intune supports for Win32 apps:

Detection type	Good for	Example
File	apps that install a stable binary	`C:\Program Files\ScrambleID\Desktop\scramble.exe` exists
Registry	MSI or apps with stable keys	`HKLM\Software\ScrambleID\Desktop\Version`
MSI	MSI-based installs	product code present
Script	complex or per-user installs	PowerShell checks path + version

Tip: pick one stable signal (file path or MSI product code) and include a version check if you want controlled upgrades.

Official reference: Microsoft Intune, Add and assign Win32 apps (includes detection rules): https://learn.microsoft.com/en-us/intune/intune-service/apps/apps-win32-add

Jamf Pro (macOS, planned), package + policy quickstart

For Jamf-managed fleets:

1. Upload the **PKG** to Jamf Pro and a distribution point
2. Create a **Policy** that installs the package
3. Scope the policy (pilot smart group → progressive rollout)
4. Use **configuration profiles** for any required permissions or system settings

Tip: keep enrollment and device binding separate from package install so you can rebind a device without reinstalling.

Official reference: Jamf Pro, Package Deployment: https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/Package_Deployment.html

5) Enrollment and binding

Common patterns:

- user-driven enrollment during first run
- helpdesk issues a one-time enrollment code (TTL enforced)

How do you handle shared workstation authentication?

Shared stations should behave like badge systems:

- deterministic states: verified / denied / timeout
- fast swaps (quick enough for shift handoffs; track the median)
- clear audit per login with SUID and ZID

Operational tips:

- pin the login method (Pattern A in clean rooms)
- ensure time sync (clock skew breaks TTLs)
- instrument swap time and failure reasons

When should you allow offline access?

If you enable offline login:

- keep window time-boxed
- require local user verification (Windows Hello)
- limit attempts and log every attempt
- sync audit events once connectivity returns

State the core implication plainly in your policy: revocation and deprovisioning do not reach an offline workstation during the window. A terminated or compromised user can still log in locally until the machine reconnects or the window expires, so keep windows short in high-churn or high-risk environments.

How do you troubleshoot Desktop deployment issues?

Symptom: stuck at "waiting for confirmation"

- verify realtime connectivity to ScrambleID endpoints
- validate device time sync
- confirm policy allows the selected pattern (A vs B)

Symptom: biometric prompt fails repeatedly

- validate Windows Hello / Touch ID enrollment

- check policy for PIN fallback

Symptom: wrong device scanned in QR flow

- teach users intent binding: compare on-screen cues before confirming

Symptom: Intune keeps reinstalling / reports "not installed"

- verify detection rules match the actual install path and version
- prefer MSI product code or a stable file path if available
- confirm 32-bit vs 64-bit registry views if you use registry detection

Symptom: high timeout rate

- shorten prompts, reduce steps
- check if users are hitting network captive portals

Symptom: Intune says "installed" but app doesn't appear

- confirm the **detection rule** is actually validating the correct file/registry path
- verify install context (system vs user)
- check device reboot requirements (some credential provider scenarios require a restart)

What metrics should you track for Desktop?

- login success rate
- median and p95 time-to-login
- biometric fail/cancel rate
- offline login rate
- shared station swap time
- helpdesk password reset ticket reduction

(See: Metrics + ROI Playbook)

Key Takeaway

Desktop passwordless replaces workstation passwords with device-bound keys (Windows Hello today; Touch ID as macOS support lands) while preserving AD/LDAP governance. Shared workstations use tap-in/tap-out session binding. Offline login should be time-boxed and audited, it's a policy decision, not a default.

FAQ

When should we deploy ScrambleID Desktop?

When passwords at the endpoint are a major risk or operational drag (contact centers, clean rooms, shared workstations) and you want deterministic cryptographic proof.

Can Desktop work without mobile devices?

Yes. Pattern A uses platform authenticators (Windows Hello / Touch ID) to unlock device-bound keys.

Does Desktop replace Active Directory?

No. It replaces passwords at workstation login while letting AD/LDAP governance and group policy remain in effect.

Is QR login required?

No. QR cross-device assist is optional and policy-controlled.

Can we allow offline login?

Yes, but keep it time-boxed and audited. Offline is a business policy decision, not a default.

References (public)

- NIST Zero Trust Architecture (identity and context at decision time): https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf
- Microsoft Intune: Add and assign Win32 apps (detection rules): https://learn.microsoft.com/en-us/intune/intune-service/apps/apps-win32-add
- Jamf Pro: Package Deployment: https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/Package_Deployment.html
- Windows Hello for Business overview: https://learn.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/

Related reading

- Phishing-Resistant Web Authentication
- Dynamic Identifiers (DID/QID)
- Metrics + ROI Playbook