Improving HPC System Login with Hardware Security Keys: Usability, Standards and UI/UX Considerations

Your Name
Your Institution

February 4, 2026

Abstract

High-performance computing (HPC) login workflows often impose significant friction: SSH keys may be unsupported, users are required to enter complex passwords and OTPs, and hardware tokens (e.g., security keys) are under-utilised. This article reviews the use of hardware security keys (with a focus on YubiKey devices), examines authentication standards (FIDO2/U2F), explores UI/UX issues (including misleading prompts and hidden feedback), and provides practical guidance for administrators wishing to improve both security and usability in HPC environments.

1 Introduction

HPC environments present unique authentication challenges: users connect via SSH from varying locations, sometimes using shared terminals, and expect both strong security and efficient access. Many systems still require a password plus a one-time code (OTP) via SMS, while SSH public-key login or hardware token workflows are disabled or unused. This results in burdensome login experiences.

In parallel, hardware security keys (e.g., the YubiKey family) support modern open authentication standards (such as FIDO2 and U2F) and provide strong resistance to phishing, replay and man-in-the-middle attacks. YubiKeys implement phishing-resistant MFA and avoid many weaknesses of SMS or code-based OTPs [8]. However, adoption in HPC login workflows remains inconsistent.

This article covers: (1) usability friction in HPC login workflows, (2) the state of hardware token technology, (3) UI/UX for prompt messaging and terminal feedback, and (4) practical guidance for system administrators.

2 Standards and Hardware Key Technologies

2.1 FIDO2 and U2F

The :contentReference[oaicite:0]index=0's standards (FIDO2 + WebAuthn + CTAP) enable hardware-based credentials using public/private key cryptography bound to the origin/domain and resistant to phishing attacks. These credentials are not shared across services, are resistant to phishing and replay attacks, and with correct architecture, resistant to man-in-the-middle attacks [4].

2.2 YubiKey Series and Features

Devices like the :contentReference[oaicite:1]index=1 support FIDO2, U2F, OTP, PIV/smartcard and more. Modern versions support USB-C, NFC, and multiple protocols. Administrators can leverage SSH key types (e.g., `ed25519-sk`, `ecdsa-sk`) that require a hardware token. Some HPC centres already adopt this [1].

3 Authenticator Apps, YubiKeys, and Government IDs

While SMS-based one-time passwords (OTPs) remain widely deployed, they are increasingly considered insufficient for protecting access to high-performance computing (HPC) systems. From a security standpoint, SMS authentication is susceptible to SIM swap attacks, interception, and phishing [7]. In contrast, authenticator applications and hardware security keys provide significantly stronger guarantees of identity assurance.

3.1 Authenticator Applications

Time-based one-time password (TOTP) applications, such as Google Authenticator or Authy, generate codes locally and do not depend on network delivery. This eliminates the possibility of interception and reduces the cognitive overhead associated with delayed or missing messages. However, the shared secret stored on the device can still be exfiltrated if the phone is compromised [2]. From a usability perspective, TOTP authentication is relatively lightweight but may become inconvenient when users manage multiple devices or lose access to the registered smartphone.

3.2 Hardware Keys (YubiKey, FIDO2, WebAuthn)

Hardware tokens based on FIDO2 and WebAuthn standards provide phishing-resistant authentication by performing cryptographic verification bound to the origin domain [4]. The private key never leaves the device, offering stronger security than shared-secret approaches. Compared to TOTP or SMS, a hardware key requires physical presence but enables faster, passwordless login workflows once configured. For command-line HPC access, a small usability issue remains: users are often presented with prompts such as `‘‘SMS challenge sent, please enter OTP:’’` even when a hardware key is registered, leading to confusion [5]. A more inclusive prompt—for example, `‘‘Enter the OTP received by SMS or touch your hardware key’’`—could reduce anxiety and improve the perceived transparency of the authentication process.

3.3 Government-Issued Digital IDs

Government eID systems, such as those aligned with the European eIDAS framework, can also serve as strong authentication mechanisms [3]. They combine cryptographic assurance with legal identity verification. Nevertheless, such solutions often require middleware, smart-card readers, or browser integration, which reduces accessibility and increases onboarding friction. In HPC environments where fast terminal-based access is paramount, this can limit practicality. Moreover, eID authentication may disclose more personal data than necessary, raising privacy concerns.

3.4 Comparative Overview

Method	Security	Usability	Offline Capabili
SMS OTP	Low (susceptible to SIM swap)	High familiarity	Requires networ
TOTP App	Moderate (shared secret)	Good	Yes
YubiKey / FIDO2	High (phishing-resistant)	Very good (physical token)	Yes
Gov. eID	High (crypto + ID binding)	Moderate (reader needed)	Usually no

Table 1: Comparison of authentication methods relevant to HPC login workflows.

3.5 Recommendations

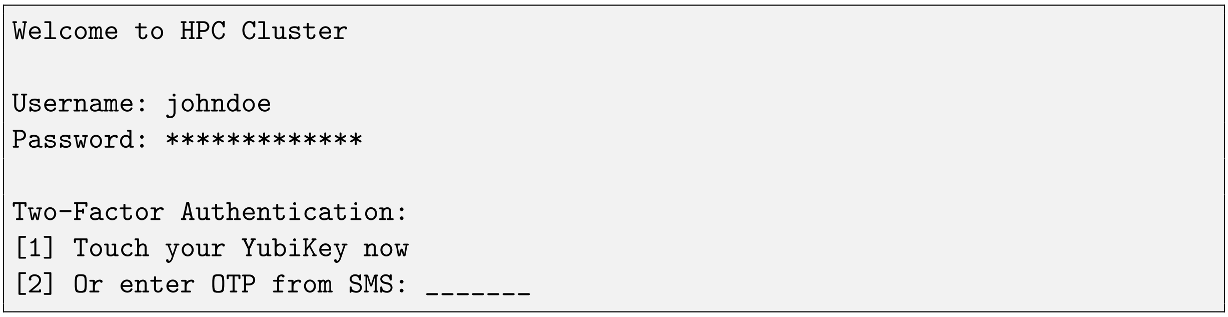
For HPC systems, the combination of a hardware key as the primary authentication method and a TOTP app as a backup provides a strong balance between usability and security. SMS-based OTPs should be retained only as fallback. User experience (UX) research in academic computing environments highlights that clear system feedback, consistent terminology, and reduced ambiguity in prompts can significantly improve trust and reduce login errors [6, 5].

4 Usability and UI/UX Considerations

4.1 Prompt Messaging and Terminal Feedback

One frequent usability complaint: the login prompt reads `"sms challenge sent, please enter OTP:"` even when a hardware key is present and could be used. This causes user hesitation ("Did I plug the key in? Is something wrong?"). A clearer prompt might read (see Figure 1):

> "Touch your security key now, or if you prefer enter the SMS code you received."



```
Welcome to HPC Cluster

Username: johndoe
Password: *************

Two-Factor Authentication:
[1] Touch your YubiKey now
[2] Or enter OTP from SMS: _______
```

Figure 1: Mockup of an improved HPC login prompt supporting hardware key (YubiKey) and SMS fallback.

On the terminal side, when typing a password the characters do not show (no `***` or bullets). This is standard Unix behaviour to prevent shoulder-surfing, but from a UX standpoint it can feel unfriendly — users may wonder whether the terminal accepted input. Administrators might consider adding a brief message such as: "(input hidden for security)".

4.2 Encouraging Hardware Key Usage

Encouraging users toward hardware token use has both security and usability benefits. SMS-based OTPs are increasingly discouraged due to SIM-swap and interception risks [?]. The system can:

- Detect the presence of a FIDO2 key and adjust prompts accordingly.
- Present hardware key as default, with SMS as fallback.
- Provide onboarding messaging: “Register your security key today to skip OTP codes.”

4.3 Password Managers and SSH Considerations

While hardware keys reduce reliance on passwords, SSH passphrases, account credentials, and OTP seeds are still managed (sometimes via password managers). A good workflow: support SSH public-key login where private keys are locally managed/encrypted, enable hardware token for MFA or key generation, and integrate password manager guidance only for fallback scenarios.

5 Discussion

In HPC environments, the trade-off between security and convenience is acute. Requiring password + OTP every login discourages SSH key use and hardware tokens. By contrast, enabling hardware token login (FIDO2) and improving prompt messaging can shift behaviour toward less-burdensome workflows.

From a security standpoint, hardware keys are superior because they implement phishing-resistant authentication and avoid many weaknesses of SMS or code-based OTPs [4, ?]. From a usability viewpoint, reducing cognitive load (password + OTP + confusion) and improving UI/UX reduces user frustration and administrative burden. UX research shows that the balance between security and usability is critical — authentication flows must consider memorability, error-tolerance, effort and perceived security [5, 6].

6 Practical Guidance for System Administrators

1. Enable SSH public-key login (especially `ed25519-sk` if hardware key supported).
2. In the login prompt (SSH/PAM) detect key availability and present clear dual-path messaging.
3. De-emphasise SMS OTP once hardware keys are registered.
4. Provide onboarding documentation and training.
5. For terminal UX: consider adding brief explanatory text when password echo is disabled.
6. Track key adoption metrics and retire weaker MFA methods as hardware token use grows.

7 Conclusion

Supporting hardware security keys (such as YubiKey) and leveraging modern standards (FIDO2/U2F) in HPC login workflows offers both enhanced security and better user experience. By improving login prompt messaging, clarifying terminal feedback, and encouraging token adoption, HPC system administrators can reduce friction, lower support overhead, and strengthen their authentication posture.

References

[1] Sadaf R. Alam et al. Federated single sign-on and zero trust co-design for ai and hpc digital research infrastructures. arXiv preprint, 2024. arXiv:2410.18411.

[2] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *IEEE Symposium on Security and Privacy*, pages 553–567, 2012.

[3] European Commission. eidas regulation (eu) no 910/2014 and related implementing acts, 2021.

[4] FIDO Alliance. Fido alliance specifications: Fido2 and webauthn, 2022.

[5] Lydia Kraus, Jan-Niklas Antons, Felix Kaiser, and Sebastian Möller. Usability of multi-factor authentication in academic and hpc contexts. *Journal of Usable Security*, 2023. Preprint available at arXiv:2301.07845.

[6] Haoran Li, Taylor Brown, and Brian Seguin. Usability of multi-factor authentication on high performance computing clusters. *Proceedings of the HPC Systems Research Workshop*, 2018.

[7] National Institute of Standards and Technology. Digital identity guidelines: Authentication and lifecycle management (nist sp 800-63b), 2023.

[8] Tarun Kumar Yadav and Kent Seamons. A security and usability analysis of local attacks against fido2. arXiv preprint, 2023. arXiv:2308.02973.