YubiKey e TPM2 su Linux

Hardware Security: dalla linea di comando

Valerio Vaccaro

Satoshi Spritz Connect

23 Febbraio 2026

Valerio Vaccaro 👀

- 💻 Sviluppatore Bitcoin ed Esperto Hardware
- 🔥 Contributore a progetti Bitcoin open source
- ⚠️ Appassionato di hardware fai-da-te (DIY)
- Ingegnere Bitcoin e Liquid presso Blockstream

Social

- 🧑‍💼 **LinkedIn** linkedin.com/in/valeriovaccaro
- 🐙 **Github** github.com/valerio-vaccaro
- **Telegram** t.me/valeriovaccaro

Licenza

May this presentation inspire you to become more self-sovereign!

Meme


QUANDO TI DICONO CHE
SONO AL SICURO
PERCHE' HANNO UNA
CERTIFICAZIONE

Sommario

- Cos'è YubiKey e cosa può fare
- Cos'è TPM2 e cosa può fare
- Esempi pratici da linea di comando
- Confronto e casi d'uso

Introduzione

YubiKey

- **Hardware Security Module (HSM)** portatile, formato chiavetta USB
- Prodotto da Yubico
- Chiavi crittografiche mai escono dal dispositivo
- Supporta: **PIV**, **OpenPGP**, **FIDO2/WebAuthn**, **OATH** (TOTP/HOTP)

TPM2

- **Trusted Platform Module 2.0** — chip integrato nella scheda madre
- Standard TCG (Trusted Computing Group)
- Chiavi e dati "sigillati" allo stato dell'hardware/firmware
- Usato per: **secure boot**, **disk encryption**, **attestation**

Introduzione

Perché hardware?

- Chiavi non esposte in memoria o su disco
- ✓ Resistenza a malware e keylogger
- YubiKey: portabile tra PC | TPM2: legato alla macchina

YubiKey — Installazione e info

Installare ykman (YubiKey Manager CLI)

```
# Debian/Ubuntu
sudo apt install yubikey-manager

# Fedora
sudo dnf install yubikey-manager

# Via pip
pip install --user yubikey-manager
```

YubiKey — Installazione e info

Informazioni sul dispositivo

```
# Lista YubiKey collegati
ykman list --serials

# Info dettagliate
ykman info

# Info per dispositivo specifico
ykman --device 01234567 info
```

YubiKey — PIV (Smart Card)

Cos'è PIV

- **Personal Identity Verification** — standard NIST per smart card
- Slot per certificati: **9a** (auth), **9c** (signing), **9d** (key mgmt), **9e** (card auth)
- Usato per: login SSH, firma, cifratura

Generare chiave e certificato PIV

```
# Genera chiave ECC P-256 nello slot 9a
ykman piv keys generate --algorithm ECCP256 9a pubkey.pem

# Genera certificato auto-firmato
ykman piv certificates generate --subject "CN=SSH Key" 9a pubkey.pem

# Esporta chiave pubblica per SSH
ykman piv keys export 9a pubkey.pem
ssh-keygen -D /usr/lib/opensc-pkcs11.so -e
```

YubiKey — PIV: gestione PIN e chiave

Cambiare PIN e Management Key

```
# Cambia PIN (default: 123456)
ykman piv access change-pin --pin 123456 --new-pin NUOVO_PIN

# Cambia Management Key (genera random, protetta da PIN)
ykman piv access change-management-key --generate --protect

# Reset completo PIV (attenzione: cancella tutto!)
ykman piv reset
```

Verificare certificati

```
# Lista certificati negli slot
ykman piv certificates list
```

Configurare GPG sulla YubiKey

```
# Avvia configurazione scheda
gpg --card-edit

# Comandi utili in card-edit:
# admin     → opzioni amministrative
# name      → nome sul token
# url       → URL per recupero chiave
# key-attr  → algoritmo chiavi (es. rsa4096 o ed25519)
# generate  → genera chiavi sulla YubiKey
```

YubiKey — OpenPGP / GPG

Importare chiave esistente su YubiKey

```
# 1. Genera chiave su PC (o hai già keyid)
gpg --full-generate-key

# 2. Trasferisci subchiavi sulla YubiKey
gpg --edit-key KEYID
# gpg> key 1
# gpg> keytocard
# gpg> key 2
# gpg> keytocard
# ...
```

YubiKey — GPG: touch e PIN

Requisito touch e gestione PIN

```
# Richiedi touch per chiave di autenticazione
ykman openpgp keys set-touch aut on

# Richiedi touch per signing
ykman openpgp keys set-touch sig on

# Cambia PIN utente
ykman openpgp access change-pin --pin VECCHIO --new-pin NUOVO

# Cambia Admin PIN (min 8 caratteri)
ykman openpgp access change-admin-pin --admin-pin VECCHIO --new-admin-pin NUOV
```

YubiKey — GPG: touch e PIN

Usare GPG con YubiKey

```
# Firma
gpg --sign documento.pdf

# Decifra (con PIN/touch)
gpg --decrypt documento.pdf.gpg
```

YubiKey — SSH con PIV/GPG

SSH con PIV (PKCS#11)

```
# Aggiungi chiave PIV all'agent SSH
ssh-add -s /usr/lib/opensc-pkcs11.so

# Oppure in ~/.ssh/config
Host *
    PKCS11Provider /usr/lib/opensc-pkcs11.so
```

YubiKey — SSH con PIV/GPG

SSH con GPG (gpg-agent)

```
# Abilita supporto SSH in gpg-agent (~/.gnupg/gpg-agent.conf)
enable-ssh-support

# Restart agent
gpgconf --kill gpg-agent

# Lista keygrip per SSH
gpg --list-keys --with-keygrip
```

YubiKey — OATH (TOTP/HOTP)

Configurare OATH sulla YubiKey

```
# Aggiungi credenziale TOTP da URI
ykman oath add -t "Account:user@example.com" \
  "otpauth://totp/Account:user?secret=JBSWY3DPEHPK3PXP"

# Lista credenziali
ykman oath list

# Mostra codice TOTP (richiede touch)
ykman oath code "Account:user@example.com"

# Rimuovi credenziale
ykman oath delete "Account:user@example.com"
```

YubiKey — OATH (TOTP/HOTP)

FIDO2 / WebAuthn

- Login senza password su siti compatibili
- Resistente al phishing
- Configurabile da browser o `ykman fido`

TPM2 — Concetti base

Cos'è il TPM

- Chip dedicato sulla scheda madre
- 🔒 Chiavi generate e usate solo dentro il TPM
- **PCR** (Platform Configuration Registers): hash di firmware, bootloader, kernel
- **Sealing**: dati cifrati legati a valori PCR specifici

Perché è utile

- 💾 Sblocco automatico LUKS se boot integro
- 🔑 Secure Boot + measured boot
- 🛡️ Attestation: prova lo stato del sistema

TPM2 — Installazione e verifica

Installare tpm2-tools

```
# Debian/Ubuntu
sudo apt install tpm2-tools tpm2-abrmd

# Fedora
sudo dnf install tpm2-tools

# Verificare presenza TPM
tpm2_getcap properties-fixed
tpm2_getcap properties-variable
```

TPM2 — Installazione e verifica

Verificare che il TPM funzioni

```
# Test di base: genera numero random
tpm2_getrandom 32 | xxd

# Info sul TPM
tpm2_getcap manufacturers
```

TPM2 — Creare e usare chiavi

Gerarchia e chiavi primarie

```
# Crea contesto per chiave primaria (ECC)
tpm2_createprimary -C e -g sha256 -G ecc -c primary.ctx

# Crea chiave figlia (chiave da usare)
tpm2_create -C primary.ctx -g sha256 -G ecc -u key.pub -r key.priv

# Carica chiave nel TPM
tpm2_load -C primary.ctx -u key.pub -r key.priv -c key.ctx

# Persisti in slot NVRAM (opzionale)
tpm2_evictcontrol -C o -c key.ctx 0x81010002
```

TPM2 — Creare e usare chiavi

Firmare con chiave TPM

```
# Firma hash di un file
tpm2_sign -c key.ctx -g sha256 -o sig.bin file.txt
```

TPM2 — PCR e sealing

Cosa sono i PCR

- 24 registri (0–23) con hash di componenti di boot
- PCR 0–7: firmware, bootloader
- PCR 8: UEFI events
- PCR 9: bootloader
- Se il boot cambia → PCR cambiano → dati sealed non si decifrano

Seal/Unseal con PCR

```
# Crea policy basata su PCR 0,1,2
tpm2_createpolicy --policy-pcr -l sha256:0,1,2 -f policy.pcr

# Seal dati (cifra legata a PCR)
echo "secret" | tpm2_create -C primary.ctx -G keyedhash -u key.pub -r key.priv
  -L policy.pcr -i - -o sealed.dat
```

TPM2 — Clevis e LUKS

Cos'è Clevis

- Framework per decrittazione automatica
- **Pins**: tpm2, tang, sss, ...
- Lega segreti a policy (es. PCR, Tang server)

Clevis + TPM2: cifrare dati

```
# Cifra file legato a TPM (PCR default)
echo "dati sensibili" | clevis encrypt tpm2 '{}' > encrypted.jwe

# Decifra (funziona solo su stesso PC, stesso stato boot)
clevis decrypt < encrypted.jwe
```

TPM2 — Clevis e LUKS

Clevis + TPM2: policy PCR personalizzata

```
# Lega a PCR 0 e 7 (es. Secure Boot)
clevis encrypt tpm2 '{"pcr_ids":"0,7"}' < plaintext > encrypted.jwe

# Con algoritmo specifico
clevis encrypt tpm2 '{"hash":"sha256","key":"ecc"}' < plaintext > encrypted.jw
```

TPM2 — LUKS con Clevis (sblocco automatico)

Aggiungere slot Clevis a LUKS

```
# Partizione già cifrata con LUKS
sudo clevis luks bind -d /dev/sda3 tpm2 '{"pcr_ids":"0,1,2,3,4,5,6,7"}'

# Nuovo volume LUKS con Clevis
sudo cryptsetup luksFormat /dev/sda3 --type luks2
sudo clevis luks bind -d /dev/sda3 tpm2 '{}'
```

Sblocco automatico al boot

- Con `clevis luks bind`, il initramfs può sbloccare la partizione
- 💾 Nessuna password se boot non è alterato
- Se firmware/kernel cambiano → serve password o recovery key

```
# Verifica slot
sudo clevis luks list -d /dev/sda3
```

TPM2 — Esempio completo LUKS + Clevis

Setup su Debian/Ubuntu

```
# Installa pacchetti
sudo apt install clevis clevis-luks clevis-systemd

# Bind Clevis a LUKS esistente
sudo clevis luks bind -d /dev/nvme0n1p3 tpm2 '{}'

# Aggiorna initramfs
sudo update-initramfs -u -k all
```

Recupero

- Conserva una passphrase LUKS o recovery key
- Se TPM non sblocca (hardware/firmware cambiati):
  - Inserisci passphrase manualmente
  - Oppure usa `clevis luks unlock` con Tang/SSS se configurato

Confronto YubiKey vs TPM2

YubiKey

Pro	Contro
Portabile tra PC	Può essere persa
PIV, GPG, FIDO2, OATH	Costo (~50–100€)
Touch per conferma	Richiede USB/NFC
Molto usata per SSH/GPG	

Confronto YubiKey vs TPM2

TPM2

Pro	Contro
Integrato nel PC	Non portabile
Sblocco LUKS automatico	Legato a singola macchina
Nessun costo aggiuntivo	Meno flessibile per chiavi GPG/SSH
Measured boot, attestation	

Casi d'uso pratici

Quando usare YubiKey

- Chiavi GPG per firma/criptografia
- SSH da più workstation
- FIDO2/WebAuthn per login web
- TOTP/HOTP (2FA) su un solo dispositivo
- Lavoro da remoto con chiavi portatili

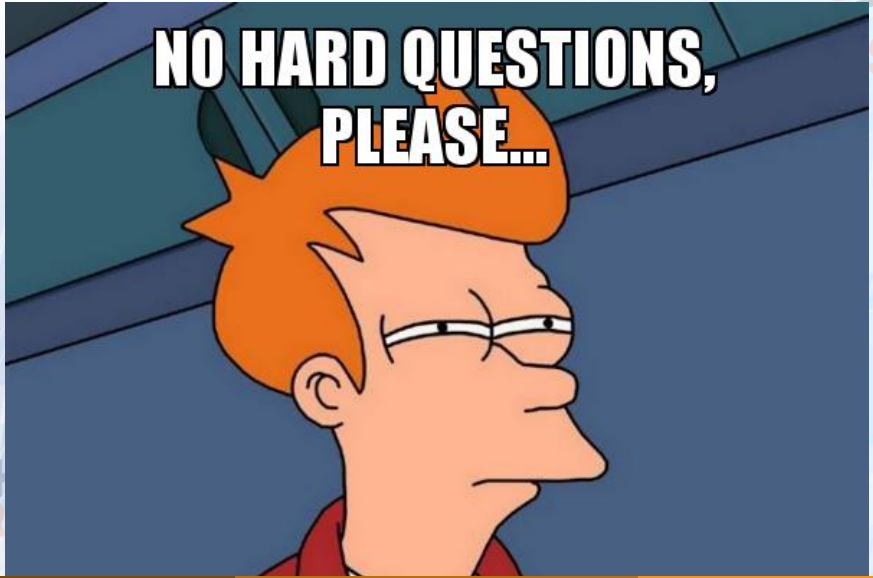
Quando usare TPM2

- Sblocco automatico disco cifrato
- Secure Boot + measured boot
- Server/VM senza interazione umana al boot
- Chiavi legate a integrità del sistema

Conclusioni

- 🔑 **YubiKey**: HSM portatile per GPG, SSH, FIDO2, OATH — ideale per chiavi che viaggiano con voi
- 💻 **TPM2**: chip integrato per sealing, LUKS automatico, attestation — ideale per proteggere la singola macchina
- 🛡️ Entrambi tengono le chiavi fuori da memoria e disco
- 🔗 Si possono usare insieme: YubiKey per identità, TPM2 per sblocco disco

? Domande



Bibliografia

- **Yubico**: docs.yubico.com — YubiKey Manager, PIV, OpenPGP
- **TCG TPM 2.0**: Specifiche Trusted Platform Module
- **Clevis**: github.com/latchset/clevis — Automated decryption
- **tpm2-tools**: github.com/tpm2-software/tpm2-tools

Risorse online

- developers.yubico.com
- tpm2-software.github.io
- blog.dowhile0.org - LUKS + TPM2

Progetto Satoshi Spritz

- Federazione di gruppi locali di Bitcoiner
- Eventi gratuiti e privacy oriented
- BITCOIN ONLY
- Satoshi Spritz Connect online settimanale
- Orientato all'apprendimento della self-sovereign

Links

- satoshispritz.it
- t.me/SatoshiSpritzConnect

Officine Bitcoin

- 🤝 Comunità Italiana di Bitcoiners, totalmente gratuita
- 🤖 BITCOIN ONLY
- 🎓 Focus su educazione e sviluppo di progetti
- 📋 Progetti: nodi Bitcoin, hardware wallet, open source, Debian, mnemoniche, ...

Links

- officinebitcoin.it