Decentralized Backup & Recovery of TOTP Secrets

Conor Gilsenan
conorgilsenan@berkeley.edu
@conorgil

Background

- Time-based One-Time Passwords (TOTP) is a widely deployed 2FA method
  - Client uses a shared secret to generate a unique OTP, which the server validates
- People lose access to the secret and risk locking themselves out of their accounts:
  - lost/new phones
  - uninstall the app
- Backup/recovery in prevalent 2FA apps have security, privacy, and usability issues

Our solution

- User can s*elf-recover* if one of many personal devices is lost
  - Full TOTP secrets are synced across paired personal devices
- User can *socially recover* from Trusted Friends when all personal devices are lost
  - Devices of Trusted Friends receive encrypted TOTP secrets and a share of the corresponding keys (i.e. Shamir's)
- Messages between devices are end-to-end encrypted and authenticated

Pilot study design

- We designed an in-lab, survey driven study
- Users are asked to enable TOTP 2FA on a Dropbox account and log in/out during several controlled scenarios

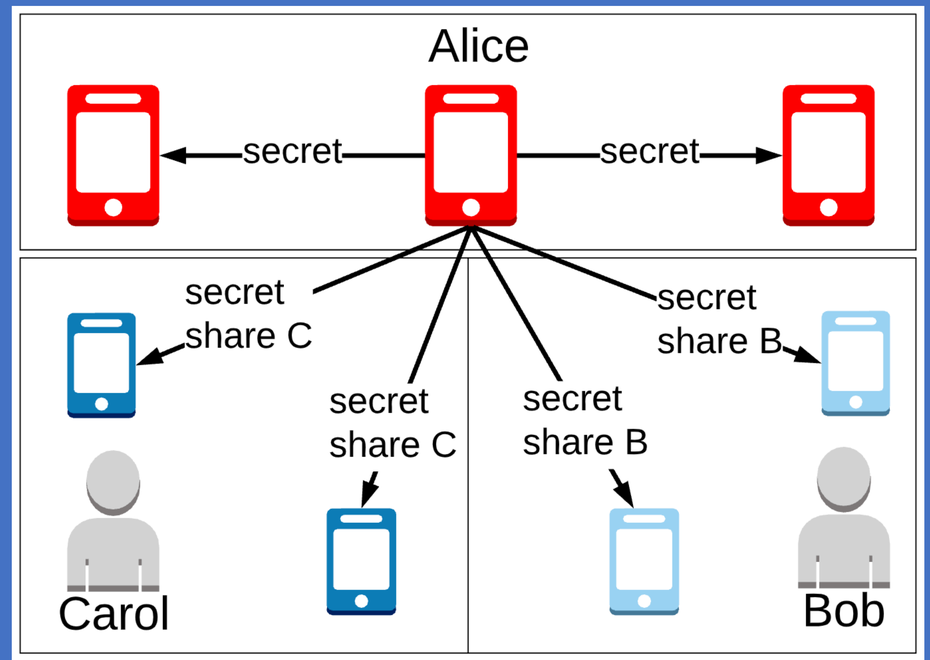
Future work

- Security analysis of prevalent 2FA TOTP apps to detail risks of backup methods
- Update prototype based on learnings from pilot study and replicate study at full scale
- Design and execute longitudinal study focused on *social recovery*

We designed a decentralized backup and recovery scheme for TOTP secrets that provides stronger security and privacy guarantees than existing 2FA apps.

1) self recovery

2) social recovery



Adding a new TOTP secret; Alice has three paired personal devices (red) and her Trusted Friends, Carol and Bob, each have two paired personal devices (blues).

Presenter Notes

Design requirements

- R1: Backup & Recovery
- R2: No accounts on central server
- R3: High-entropy keys
- R4: Server cannot access plaintext
- R5: No centralized long-term storage

TABLE I
DESIGN REQUIREMENTS APPLIED TO 2FA AUTHENTICATOR APPS

	R1	R2	R3	R4	R5
Google Authenticator	–	–	–	–	–
Authy 2FA Authenticator	●	–	–	–	–
LastPass Authenticator	●	–	–	–	–
Duo Mobile	●	◐	–	–	–
Microsoft Authenticator	●	–	●	◐	–
BLUES 2FA Authenticator	●	●	●	●	◐

● = satisfies requirement; ◐ = partially satisfies requirement; – = does not satisfy requirement;

Krypton secure pairing protocol

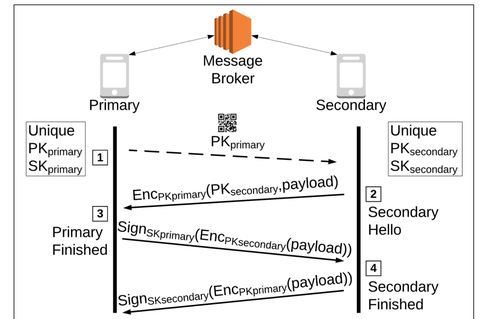


Fig. 1. The Krypton secure pairing protocol for exchanging public keys between two devices.

Other

- Generating TOTP secret shares

```
let kt = random_256_bit_key()
let cipher_text = encrypt(kt, s, metadata)
let shares = shamirs_secret_sharing(kt)
```

  - Each device of TF[i] receives cipher_text and shares[i]
- Server can verify message deletion via a simple commitment scheme
- Heartbeats can provide confidence in availability of all secret shares
- Even a user with a single personal device can use *social-recovery*
- Device revocation only protects future TOTP secrets; rotate existing secrets

Conor Gilsenan
Noura Alomar
Andrew Huang
Serge Egelman

HOTSOS 2020
7TH ANNUAL HOT TOPICS in the SCIENCE OF SECURITY
APRIL 7-8, 2020 | LAWRENCE, KANSAS

BLUES BERKELEY LABORATORY FOR USABLE AND EXPERIMENTAL SECURITY