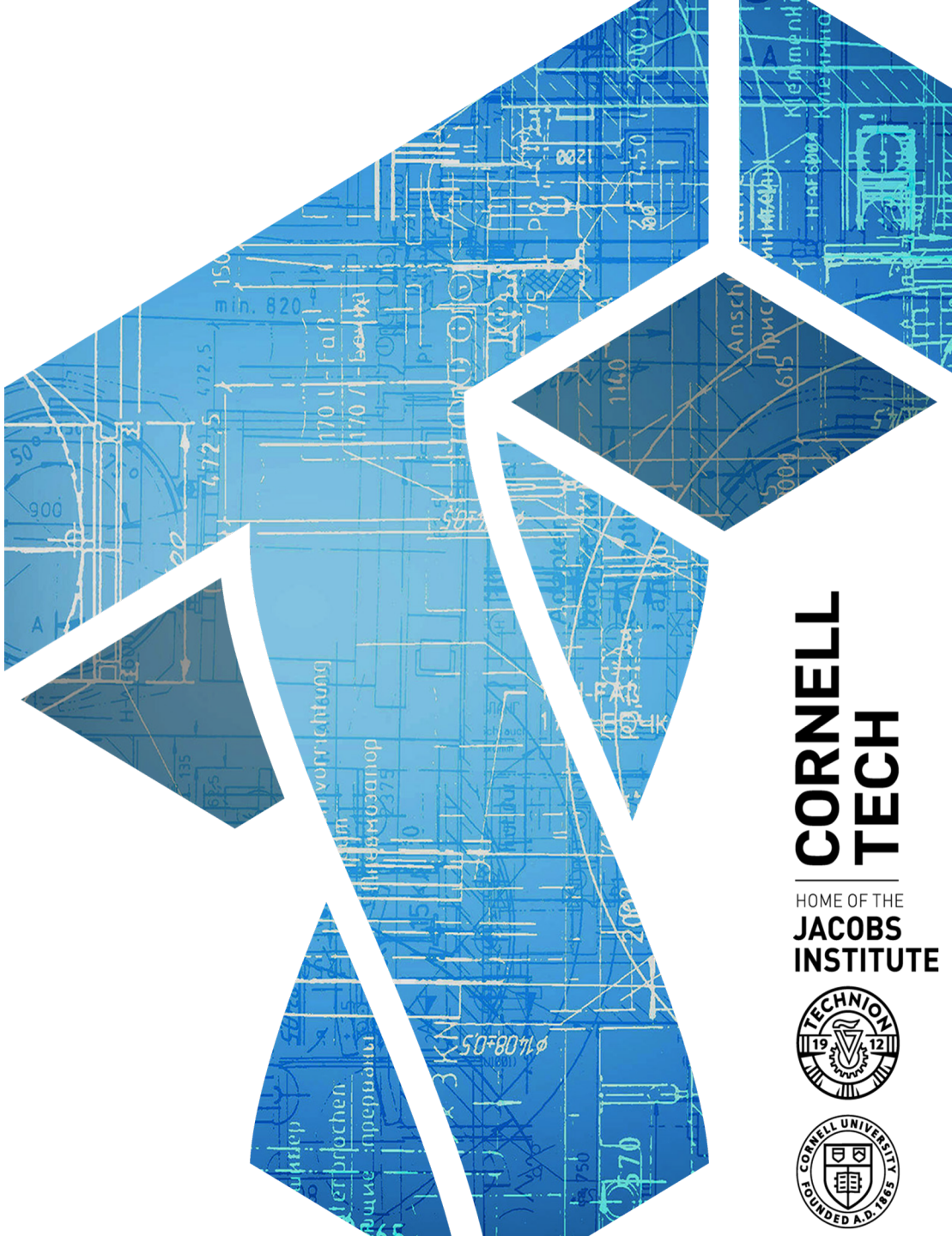
CS 5439:
Abuser-aware HCI

Tom Ristenpart



CORNELL TECH
HOME OF THE
JACOBS INSTITUTE

Four categories of common attacks

Ownership-based

- Abuser owns device/account
- Shared account/device
- Buying children device

- Prevent use / destroy device
- Digitally control access
- Track location, monitor usage

Account/device compromise

- Physical access to unlocked device
- Force password / pin revelation
- Remotely “hack” via security questions / passwords

- Install spyware / “dual-use” app
- Track location, monitor victim
- Steal or delete info
- Lock victim out of account
- Impersonate victim

Harmful messages or posts

- Call/text/message victim (from spoofed account)
- Post harmful content (e.g., threaten violence)
- Harass victim’s friends/family
- Proxy harassment

Exposure of private information

- Blackmail by threat of exposure
- “Doxxing” victim
- Non-consensual intimate images
- Fake profiles/advertisements of sexual services

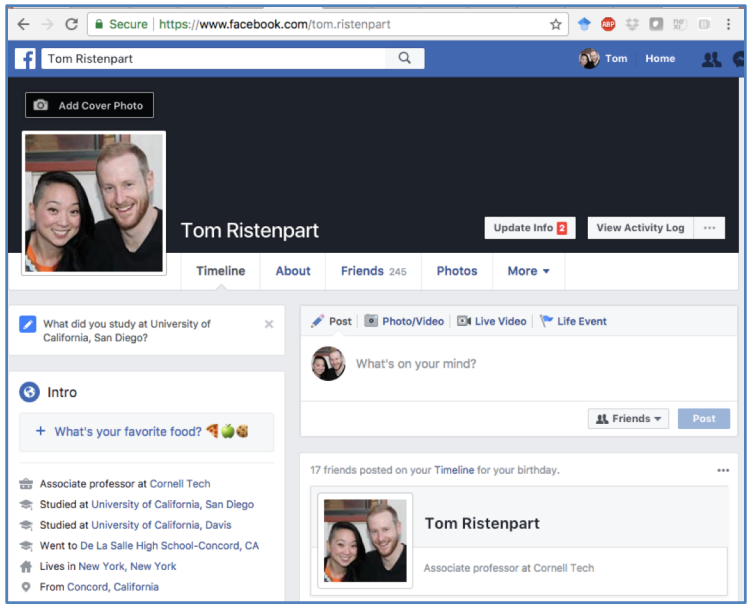
"They'll hack into their phones and they'll hack into their accounts. Especially with intimate partner victimization." *– Case manager*

Many abusers exploit normal user interfaces (UIs)

Revisit HCI to take into account IPV threat models

UI-bound adversary threat model:

- Authenticated but adversarial user
- Limited to standard user interfaces



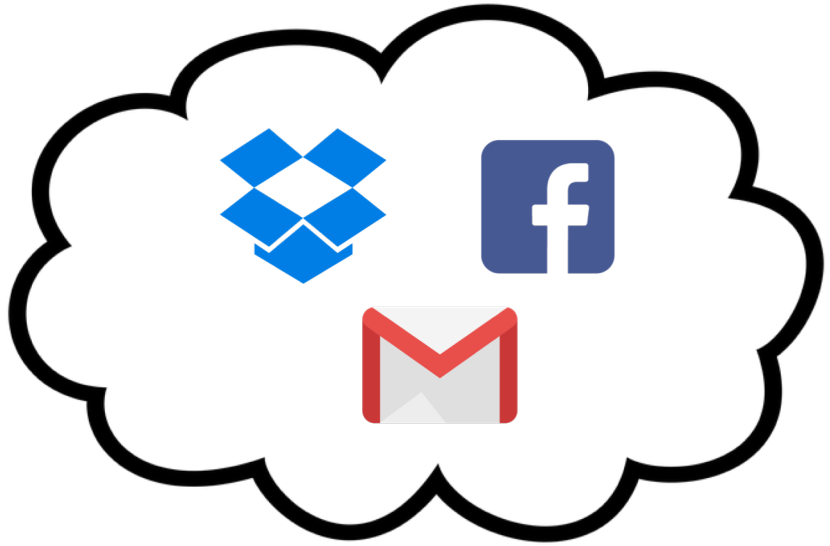
Example: covert entry

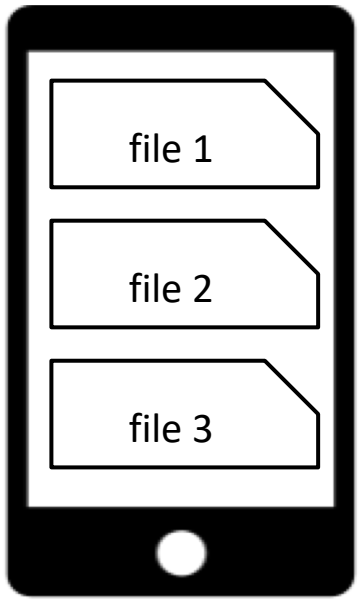
How do we build covert entry points to an app?
(Secret app useful to victim despite UI-bound adversary)

How do we know UI-bound or other adversary?

1. “Rubber-hose” cryptography & deniable file systems
   - Encrypted volumes on system
   - Enter real password to decrypt content
   - Enter decoy password to reveal decoy content
   - Can change bootloader on system to have hidden OS
   - Examples: TrueCrypt, rubber-hose FS

Compelled Access Setting: traditional viewpoint





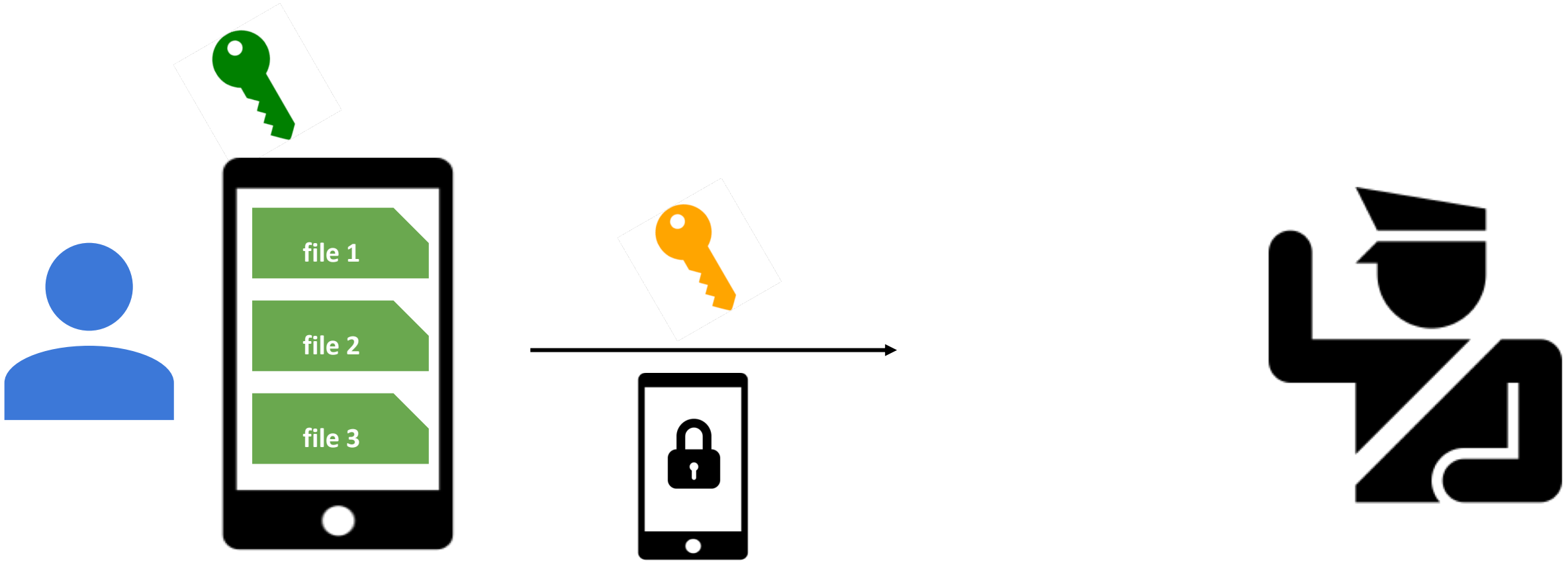
e.g., journalists, dissidents, activists

Ability to apply forensics tools

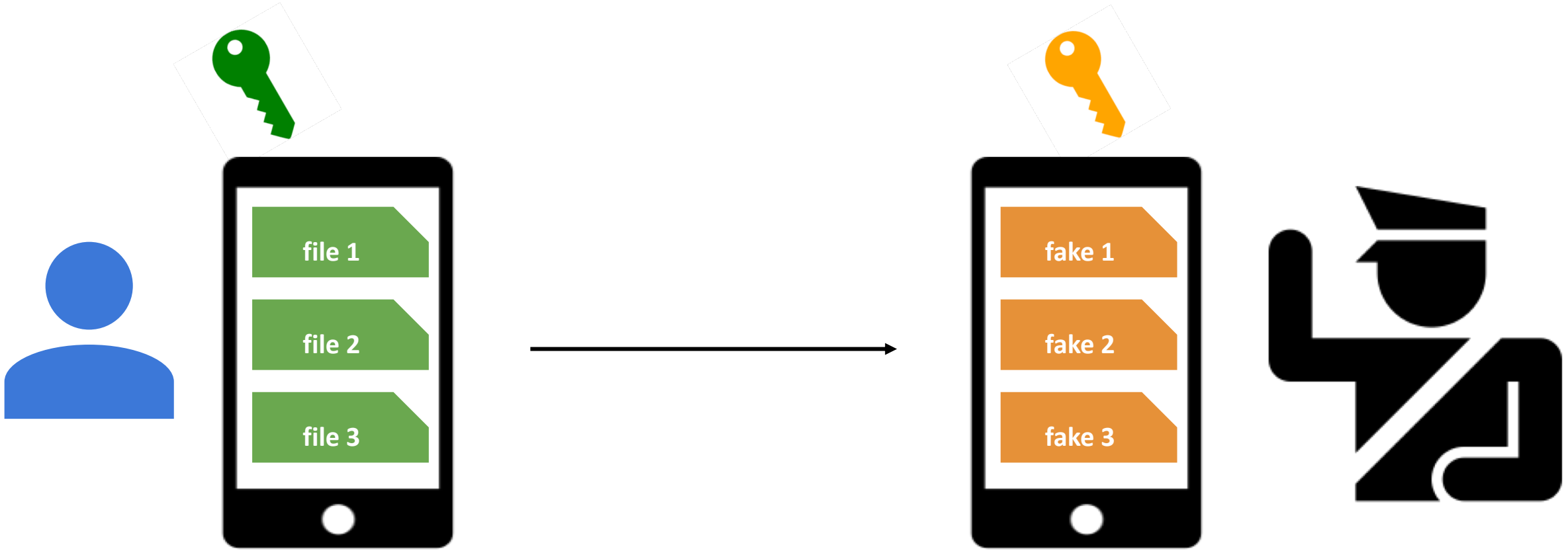
e.g., border crossing, airport security, police checkpoints

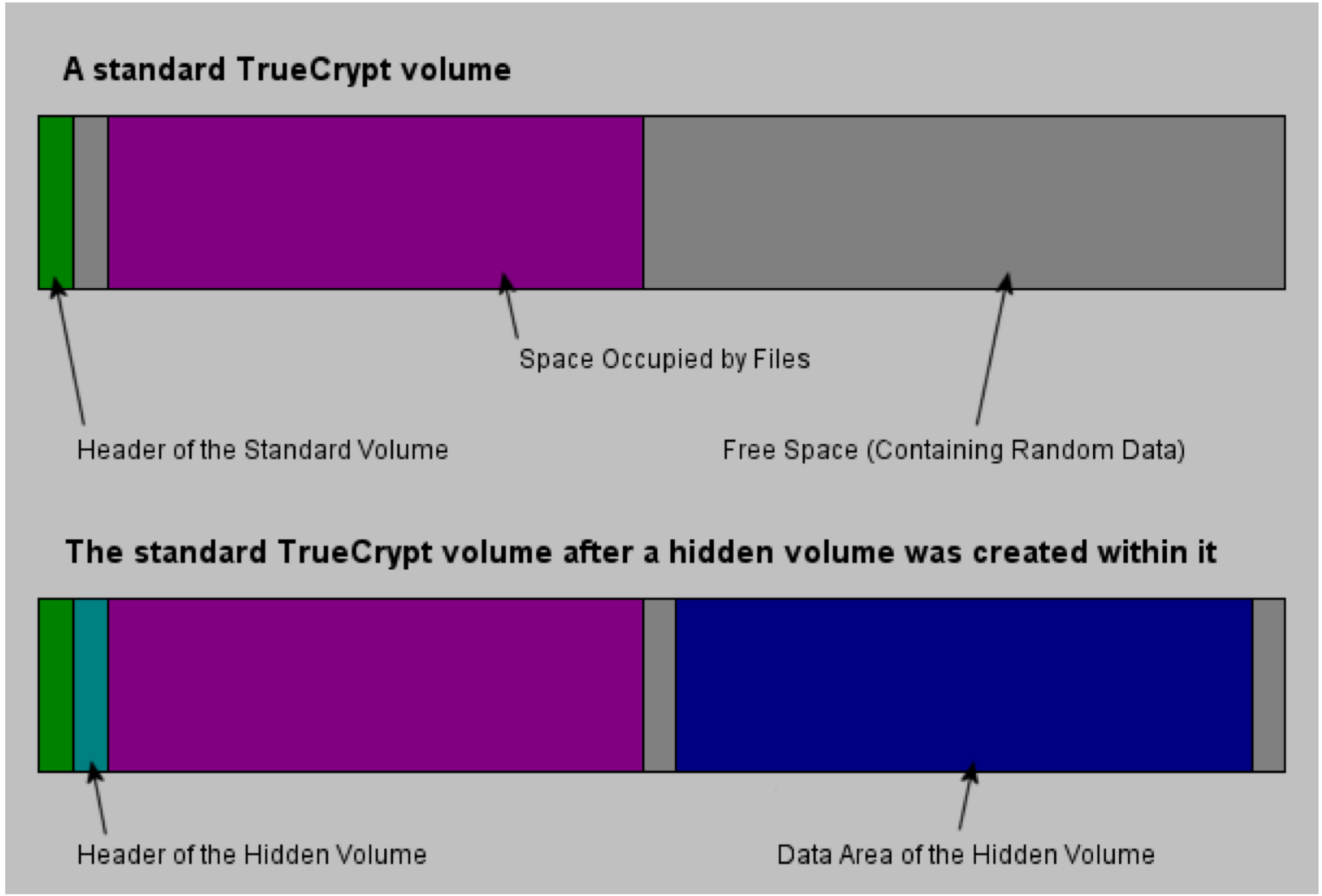
Deniable/Steganographic file systems hide files by deceiving

Deniable/Steganographic file systems hide files by deceiving

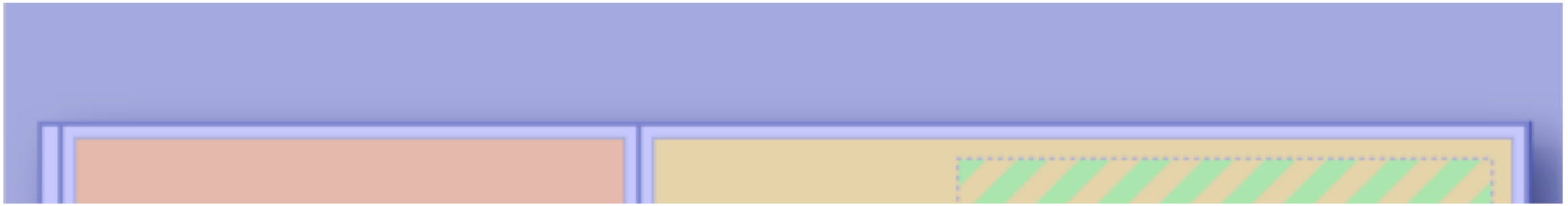


Deniable/Steganographic file systems hide files by deceiving





https://www.truecrypt71a.com/documentation/plausible-deniability/hidden-volume/



You should use the decoy operating system as frequently as you use your computer. Ideally, you should use it for all activities that do not involve sensitive data. Otherwise, plausible deniability of the hidden operating system might be adversely affected (if you revealed the password for the decoy operating system to an adversary, he could find out that the system is not used very often, which might indicate the existence of a hidden operating system on your computer). Note that you can save data to the decoy system partition anytime without any risk that the hidden volume will get damaged (because the decoy system is *not* installed in the outer volume — see below).

Example: covert entry

How do we build covert entry points to an app?
(Secret app useful to victim despite UI-bound adversary)

How do we know UI-bound or other adversary?

1. "Rubber-hose" cryptography & deniable file systems
   - Encrypted volumes on system
   - Enter real password to decrypt content
   - Enter decoy password to reveal decoy content
   - Examples: TrueCrypt, rubber-hose FS

No focus on hiding presence of tool

Requires lying when forced to enter access credential

Not even close to usable

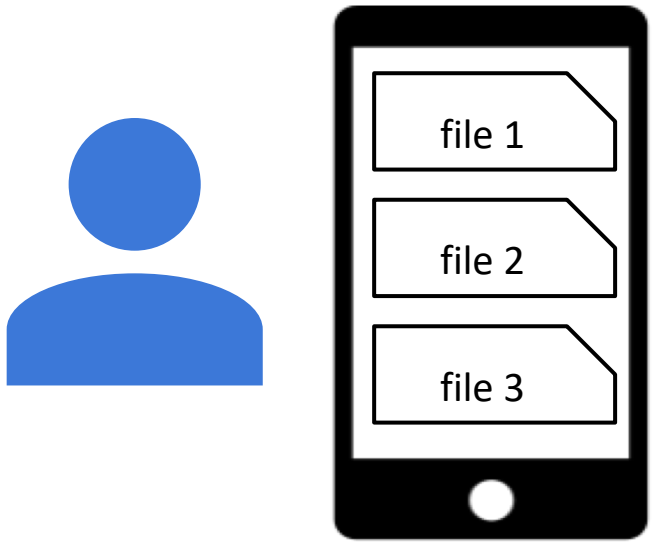
(Weaknesses to forensic tools:
https://www.schneier.com/academic/paperfiles/paper-truecrypt-dfs.pdf)

BurnBox: Confidentiality while honestly complying

- Not designed for IPV setting (digression!)
- Allow users to temporarily revoke access to cloud-backed storage before compelled search
  - Revocation and deletion indistinguishable
  - Later can recover with help of secret key stored elsewhere (e.g., at home)
- Confidentiality targeted even when disclose all true secrets to authority
  - No decoys required

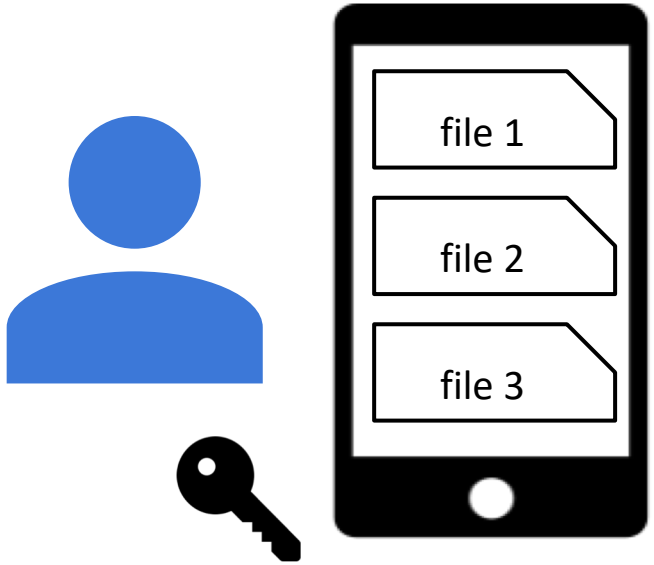
https://www.usenix.org/conference/usenixsecurity18/presentation/tyagi

BurnBox Overview



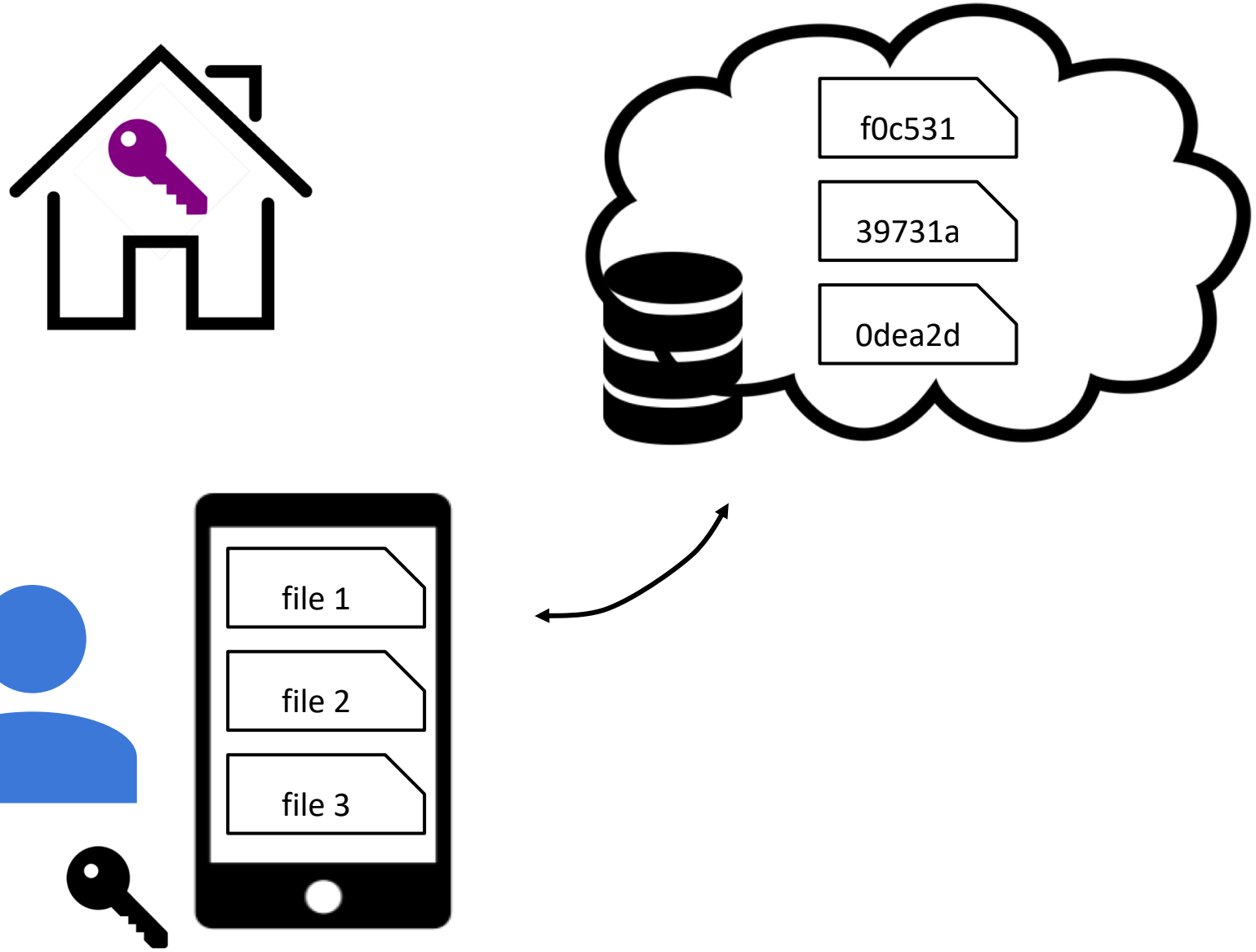
BurnBox Overview

Offline Restoration Cache

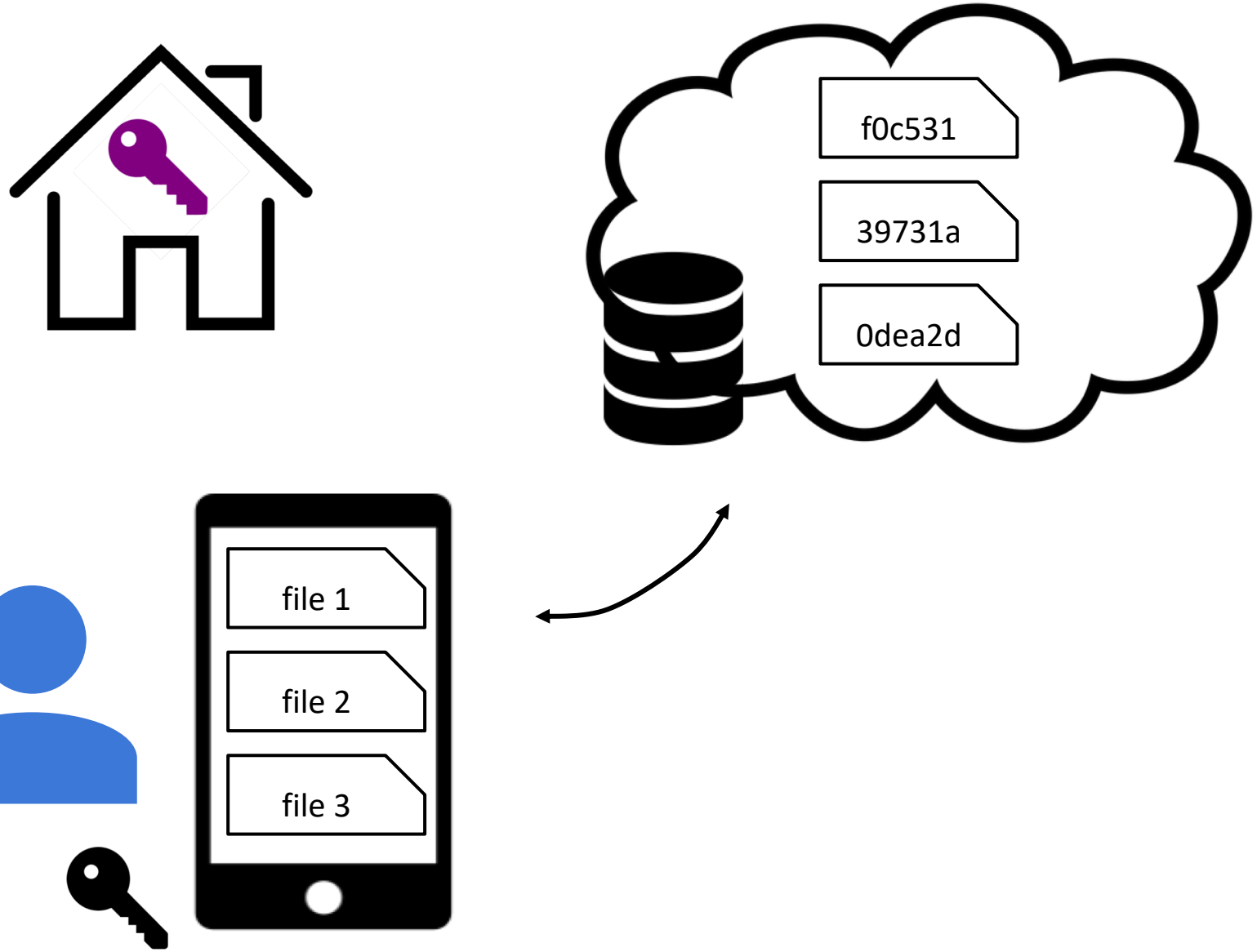


Local Device

BurnBox Overview



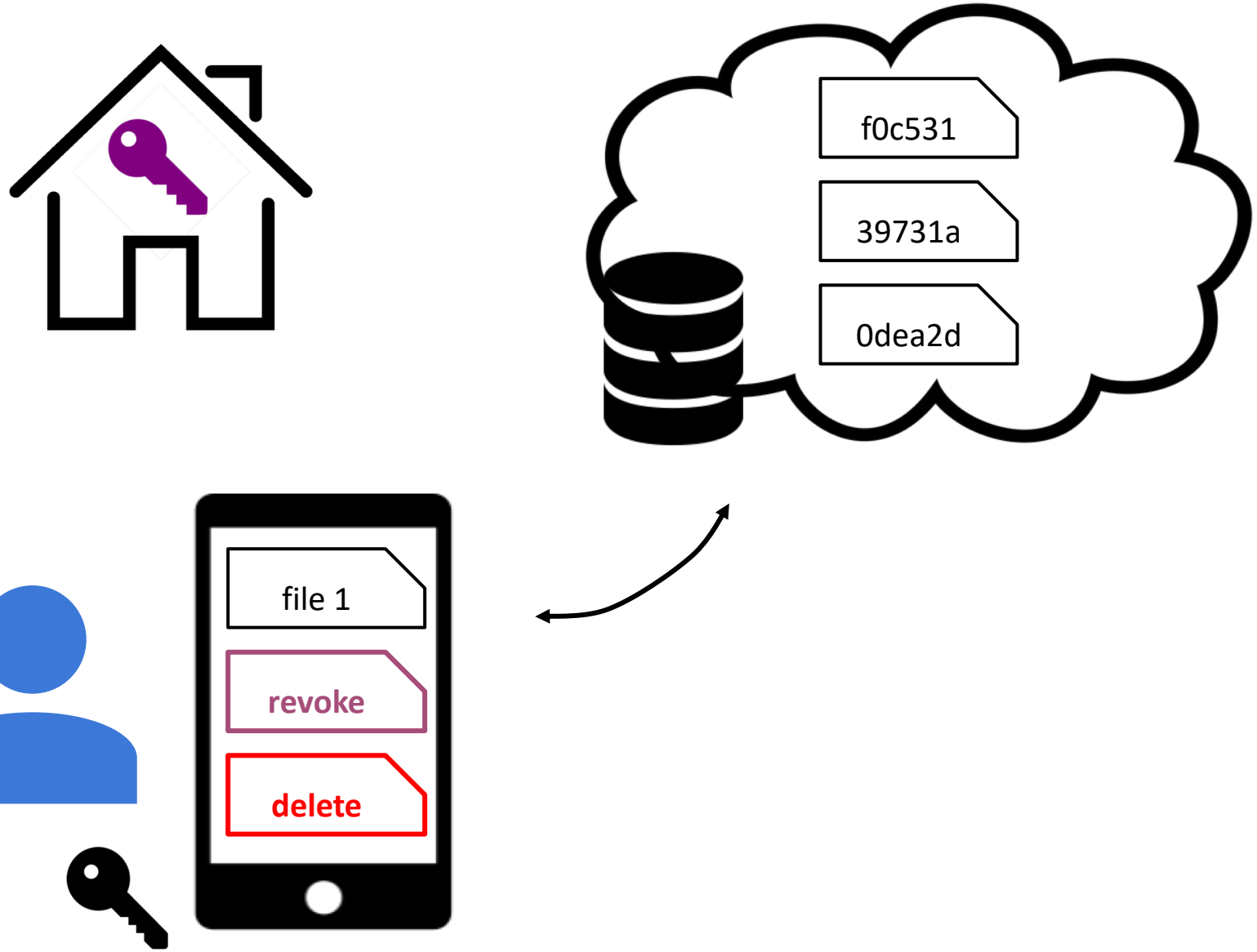
BurnBox Overview



Before Compelled Access

User selectively deletes and revokes sensitive files

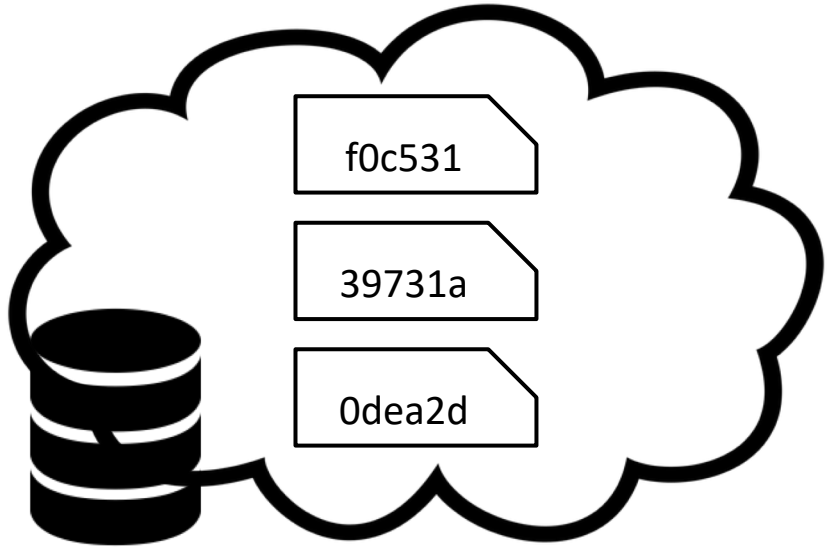
BurnBox Overview

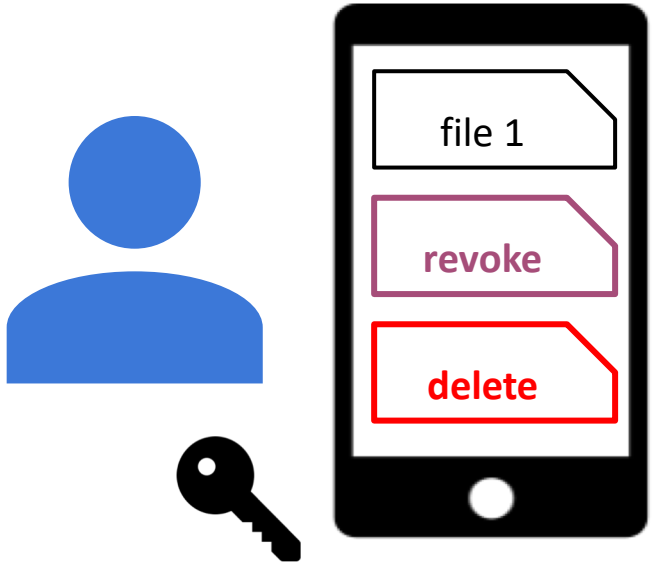


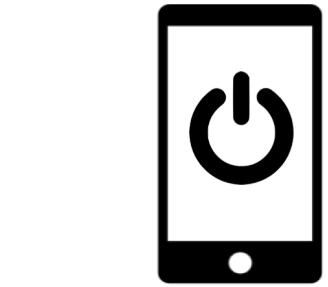
Before Compelled Access

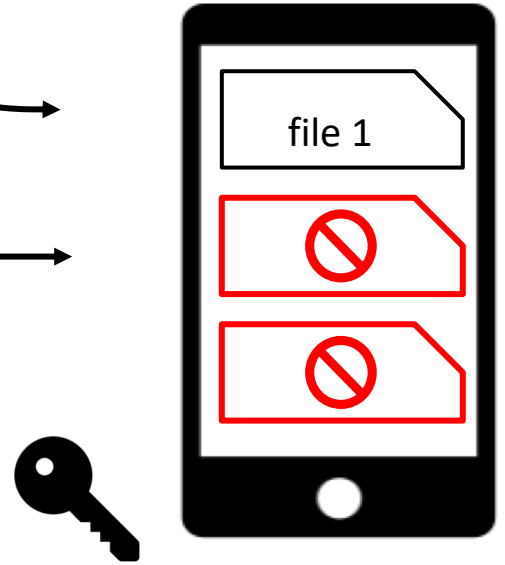
User selectively deletes and revokes sensitive files

BurnBox Overview









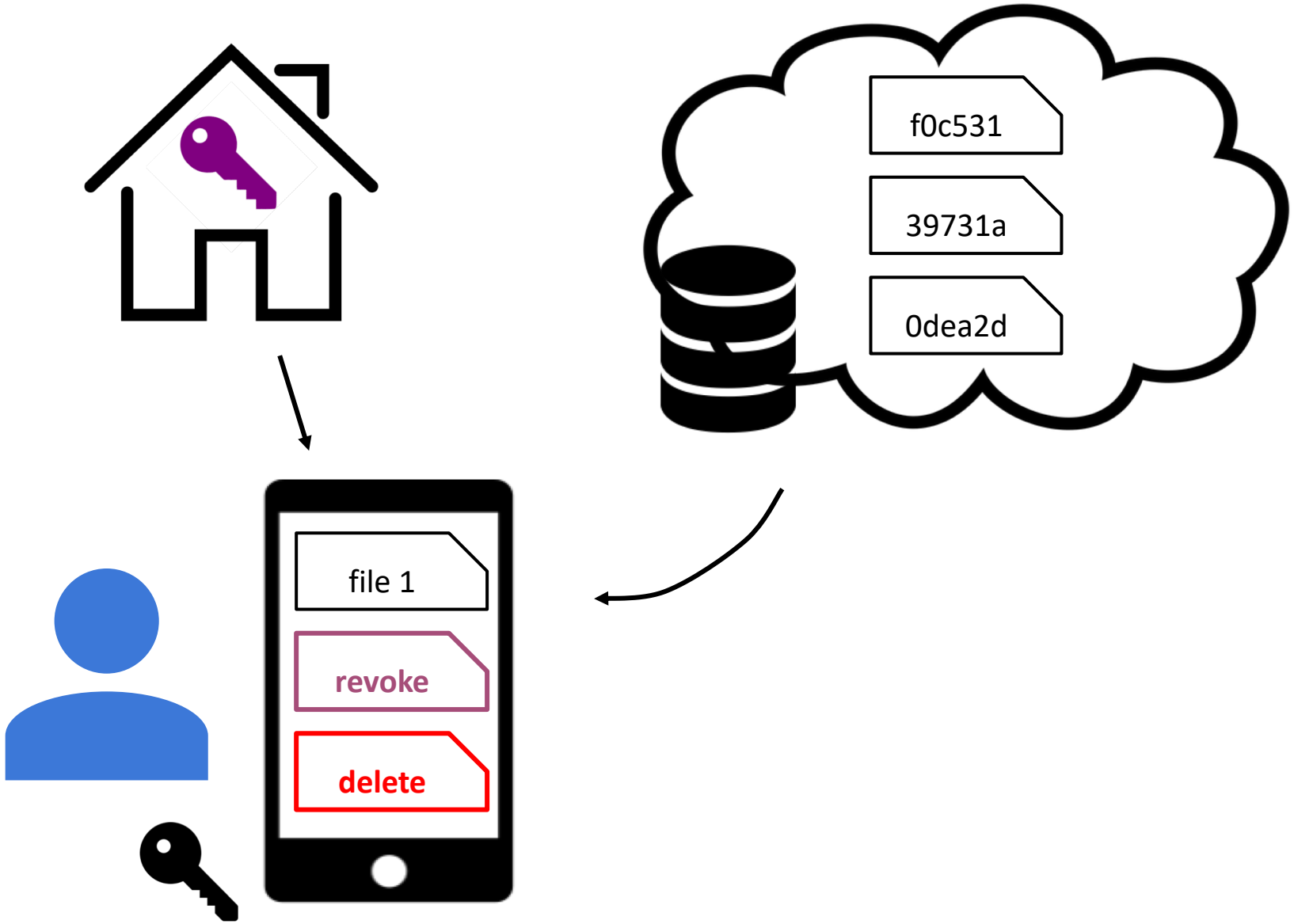
Before Compelled Access

User selectively deletes and revokes sensitive files

During Compelled Access

Deleted files and revoked files are inaccessible and are cryptographically indistinguishable

BurnBox Overview



Before Compelled Access

User selectively deletes and revokes sensitive files

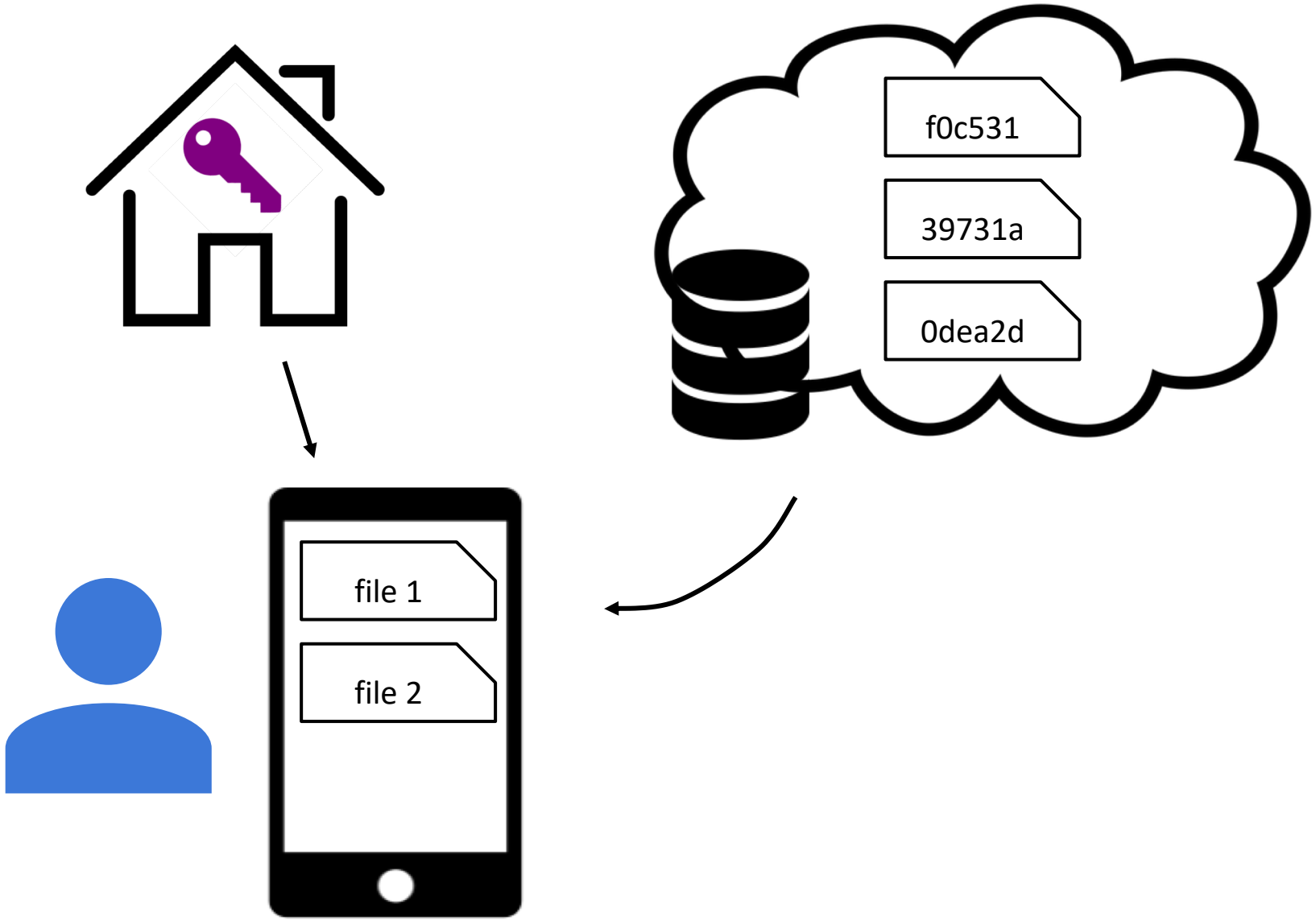
During Compelled Access

Deleted files and revoked files are inaccessible and are cryptographically indistinguishable

After Compelled Access

User restores access to revoked files with access to restoration key

BurnBox Overview



Before Compelled Access

User selectively deletes and revokes sensitive files

During Compelled Access

Deleted files and revoked files are inaccessible and are cryptographically indistinguishable

After Compelled Access

User restores access to revoked files with access to restoration key

Example: covert entry

How do we build covert entry points to an app?
(Secret app useful to victim despite UI-bound adversary)

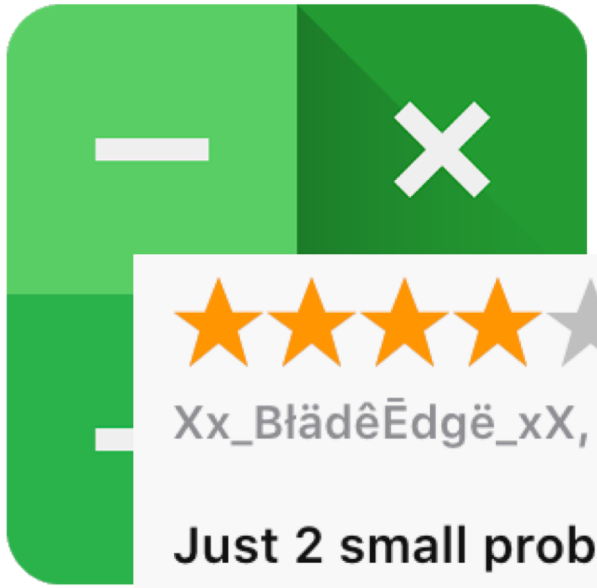
2. BurnBox
   - Disclose all secrets, cooperate with authority
   - Hides whether deleted or revoked
   - Otherwise no covertness

No focus on hiding presence of app
Not even close to usable (research prototype)
Not designed for IPV context either

Example: covert entry

How do we build covert entry points to an app?
(Secret app useful to victim despite UI-bound adversary)

3. Steganographic apps
   - Fake calculator
   - Enter special pin into keypad to enter vault
   - Otherwise works as regular
   - Lots of examples if search for “fake calculator”



Calculator Vault : App Hider - Hide Apps

★★★★☆

Xx_BłädêĒdgë_xX, 07/06/2018

Just 2 small problems

I absolutely love this app! The way it uses a calculator to disguise itself is very clever, and the password system is extremely smart. I do have two minor nitpicks however. First of all, the app is disguised as “Calculator+”, but when first opening the app, the actual calculator is almost a complete copy of the actual built in calculator app. This could very easily cause suspicion. Seeing that the app is nicknamed Calculator+, adding a few extra calculator features would prevent anybody from batting an eye at the app. If you added some new things, then maybe I would actually use this as my calculator app instead of the normal one. Secondly, it would be nice to have some sort of password/data recovery system. I do realize that this may defeat the entire point of the app if done incorrectly, but if implemented correctly, it would be very useful. There have been some occasions where I forget my password or acadentaly deleted a file.

Example: covert entry

How do we build covert entry points to an app?
(Secret app useful to victim despite UI-bound adversary)

3. Steganographic apps
   - Fake calculator
   - Enter special pin into keypad to enter vault
   - Otherwise works as regular
   - Lots of examples if search for “fake calculator”

Focus on hiding presence of app

- Security through obscurity?

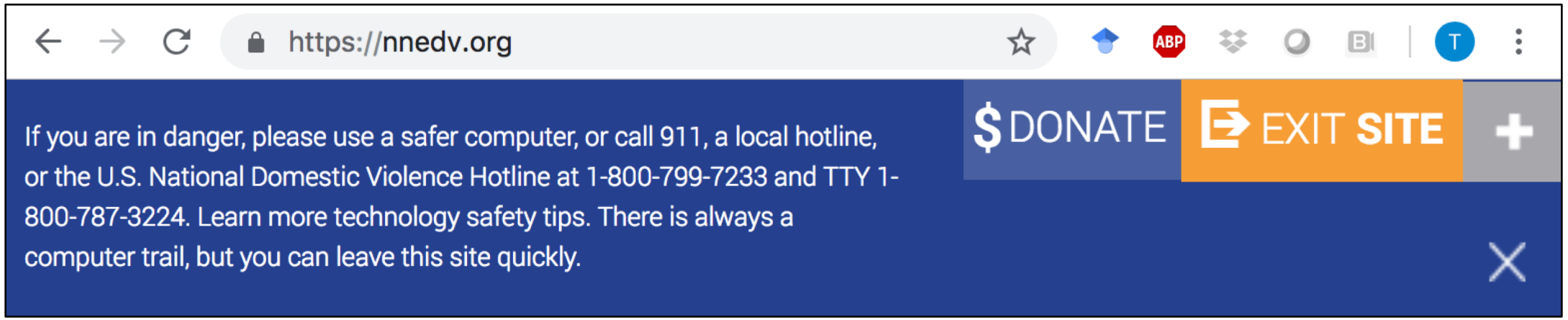
Usability questions:

- How easy to use and configure?
- How easy to use securely? (Eg: guessable PIN codes?)

Example: covert exit

How do we build covert exit for an app?
Ex: browsing ENDGBV website and abuser walks in

1. Exit buttons
   - Prominent button for leaving a website / app
   - Usually navigates to some standard website (google.com, weather.com)
   - Used on many DV support websites



Example: covert exit

How do we build covert exit for an app?
Ex: browsing ENDGBV website and abuser walks in

1. Exit buttons
   - Prominent button for leaving a website / app
   - Usually navigates to some standard website (google.com, weather.com)
   - Used on many DV support websites

Does not modify browser history --- back button returns to website

- Need to use incognito mode and close window
- Another approach masters team looked at: entry point embedded on innocuous webpage to iframe

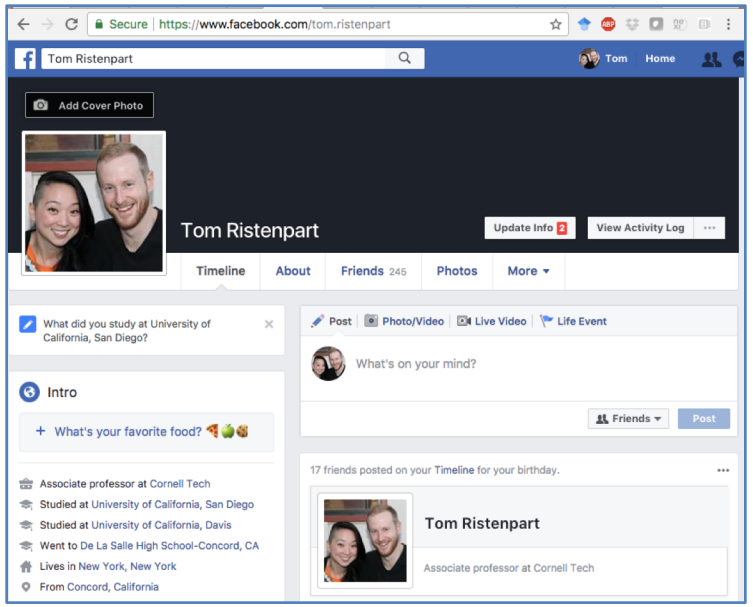
Closing an app does not remove it from app tabs list

Many abusers exploit normal user interfaces (UIs)

Revisit HCI to take into account IPV threat models

UI-bound adversary threat model:

- Authenticated but adversarial user
- Limited to standard user interfaces



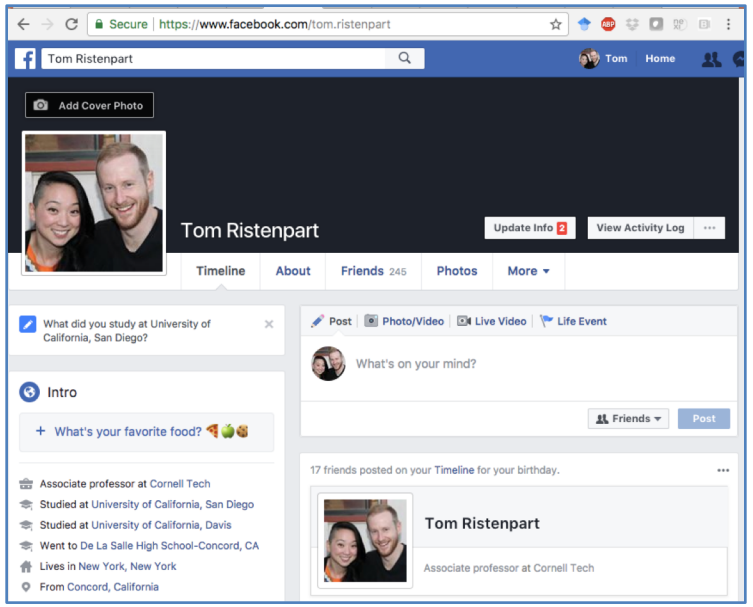
Detecting abuser access to accounts?

Can we detect abuser access to devices or accounts?

Devices: role of biometrics?

Online accounts:

- Can we detect abuser accesses via behavioral cues?



Abuser-aware HCI

Need new methods for UI design that account for UI-bound adversaries

- User studies (standard, but need to avoid WEIRD traps)
- IPV safety reviews
- Measurement studies with / at companies of abuser behaviors
- Other new methods???

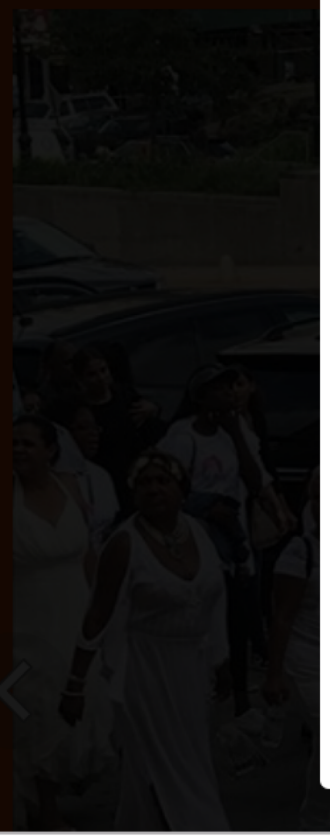
https://www1.nyc.gov/site/ocdv/index.page

End Domestic and Gender-Based Violence | 311 | Search all NYC.gov websites

Mayor's Office to End Domestic and Gender-Based Violence

Kreyòl Ayisyen | Translate | Text-Size

About



Computer Safety Notice

Be aware that abusers can track your visit to this Web page. If you feel that your internet usage may be monitored, please call the 24-hour New York City Domestic Violence Hotline at 1-800-621-HOPE (4673).

NOW

York City
ommunity
o promote
s for
violence.
r DVAM 2018
for
g events, or
tion's DV
ss events to
the calendar of