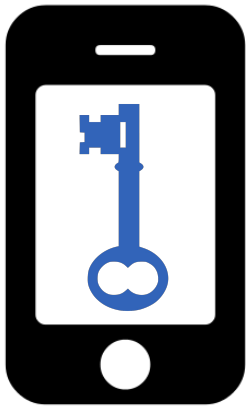
Shatter: Using threshold cryptography to protect single users with multiple devices

Erinn Atwater and Urs Hengartner

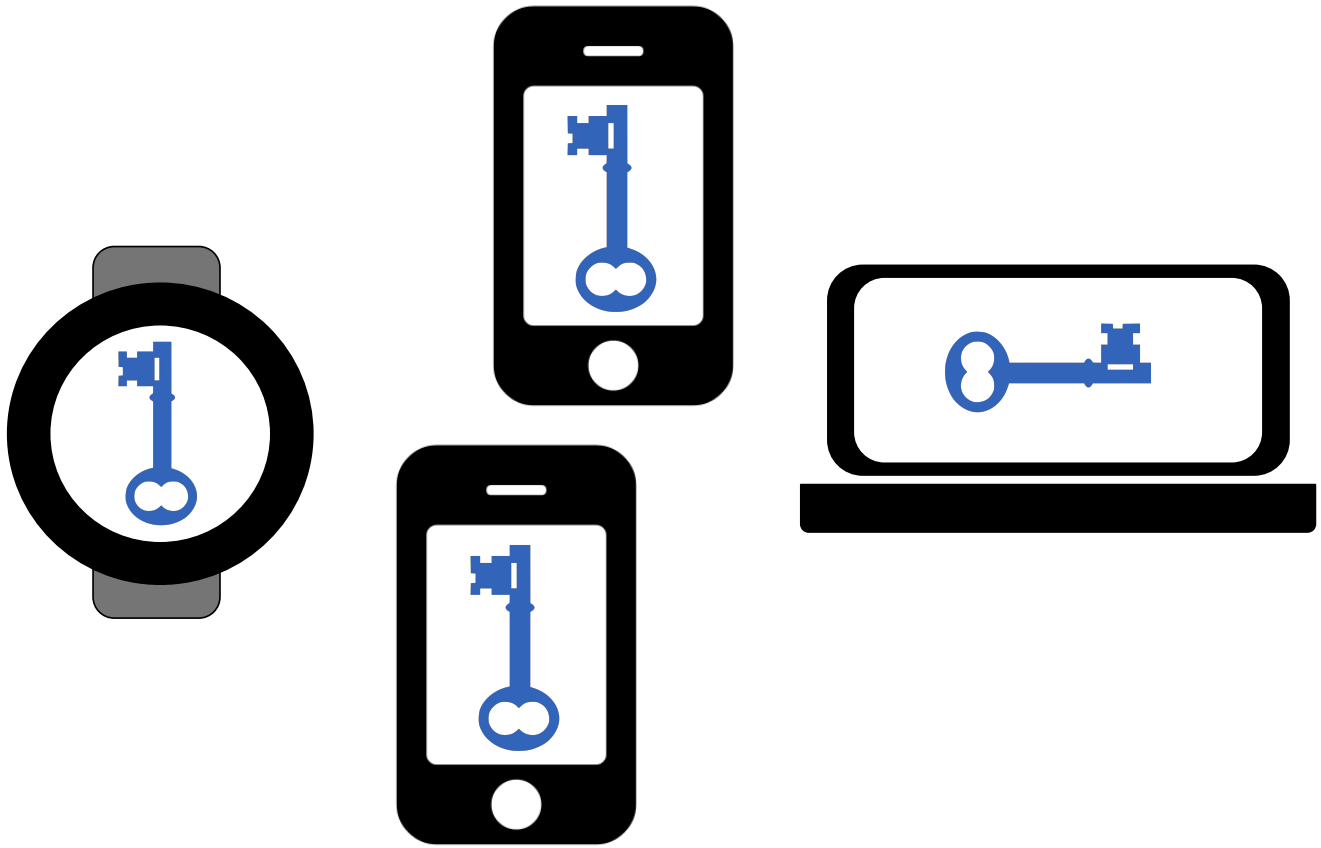
University of Waterloo, Ontario, Canada

The multi-device problem

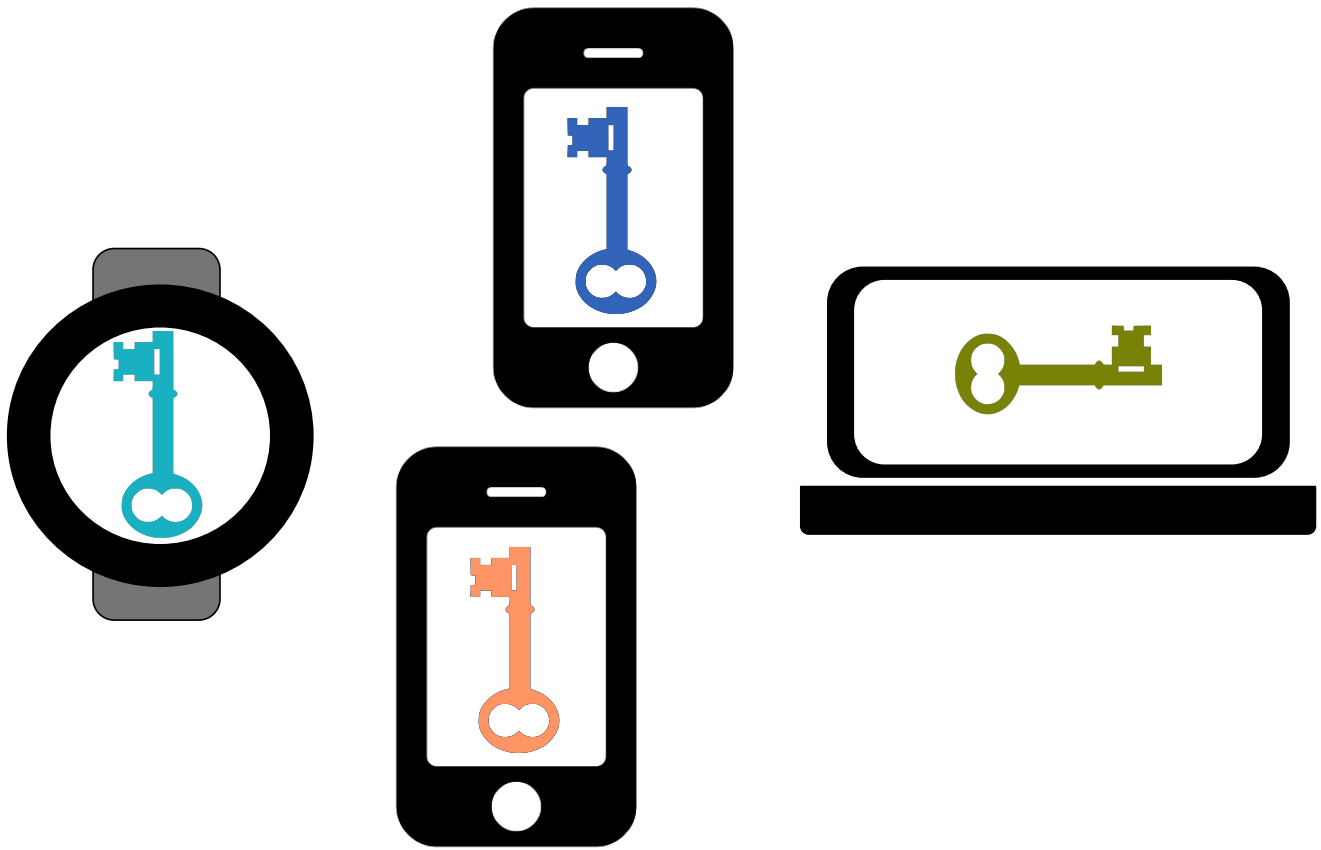
The multi-device problem



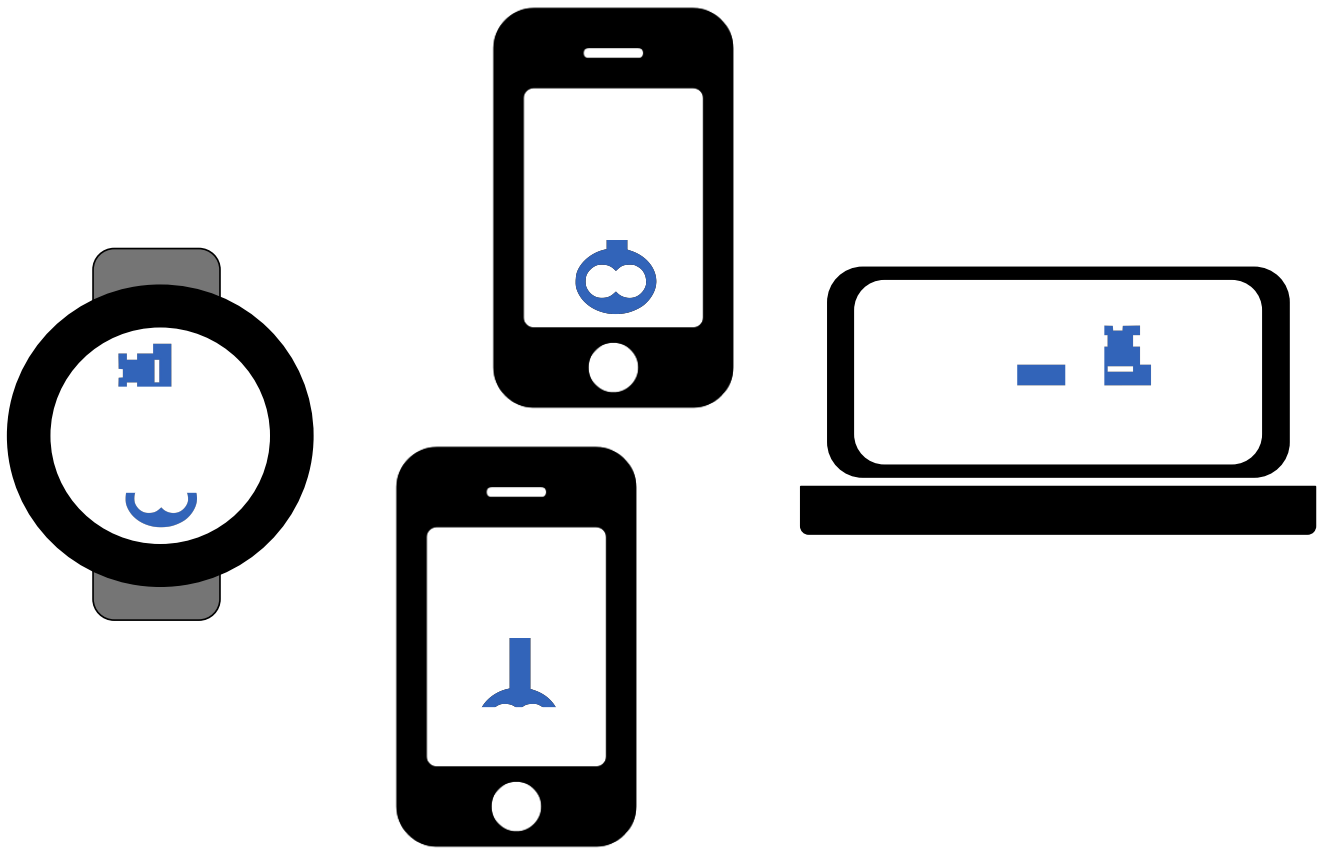
The multi-device problem: Key sync



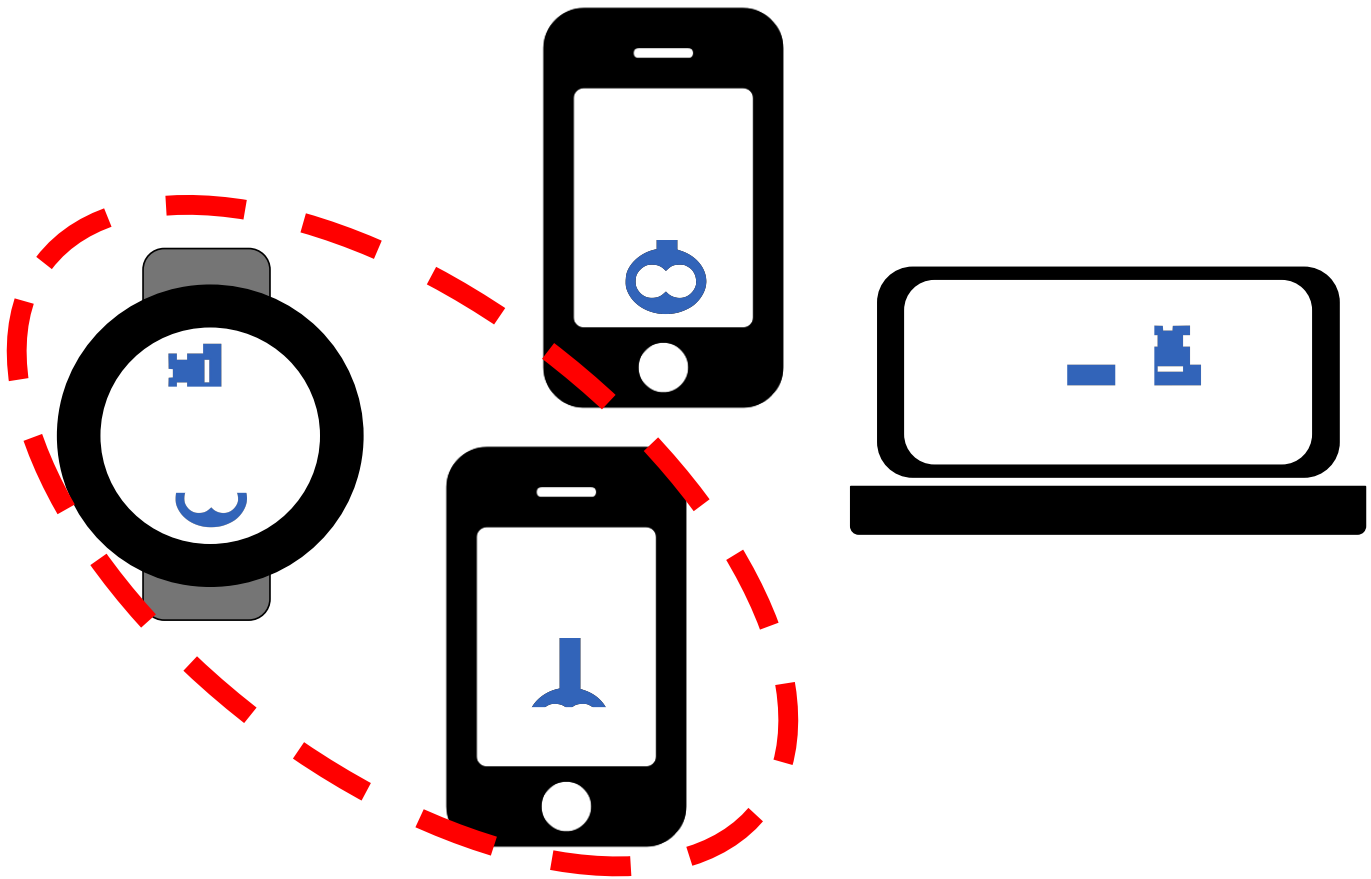
The multi-device problem: Separate keys



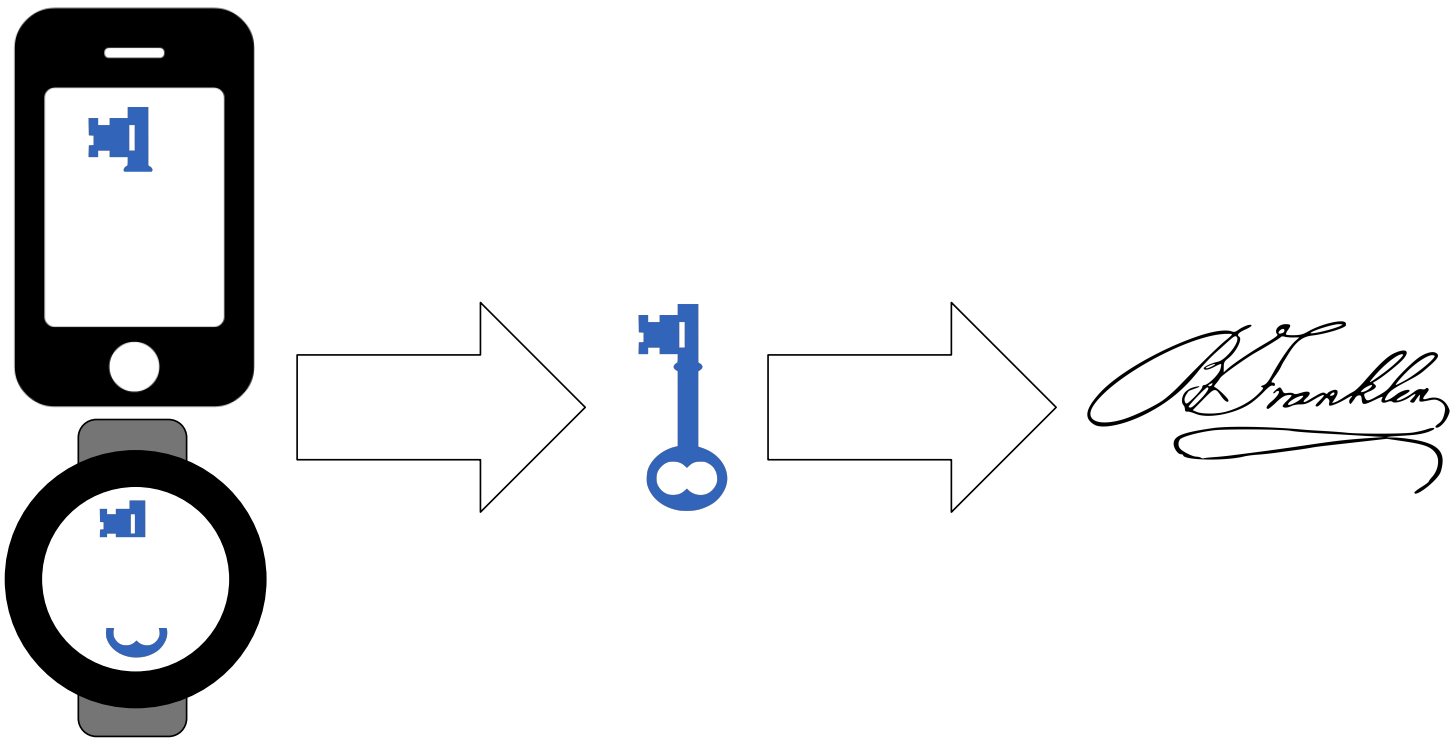
The multi-device problem: Threshold crypto



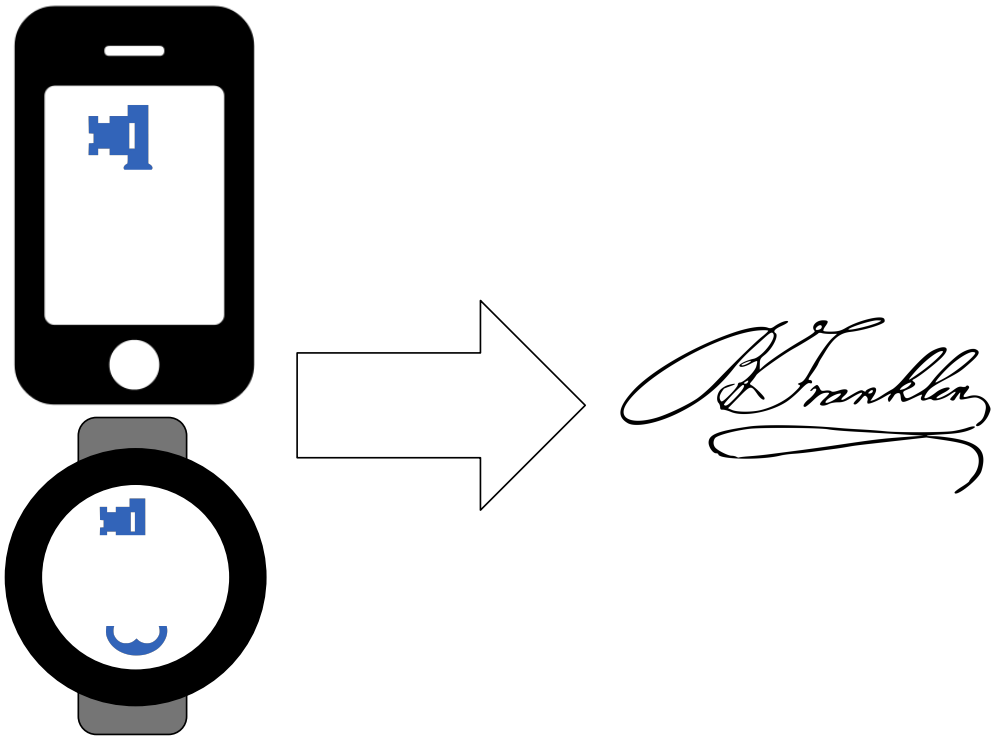
The multi-device problem: Threshold crypto



The multi-device problem: Secret sharing



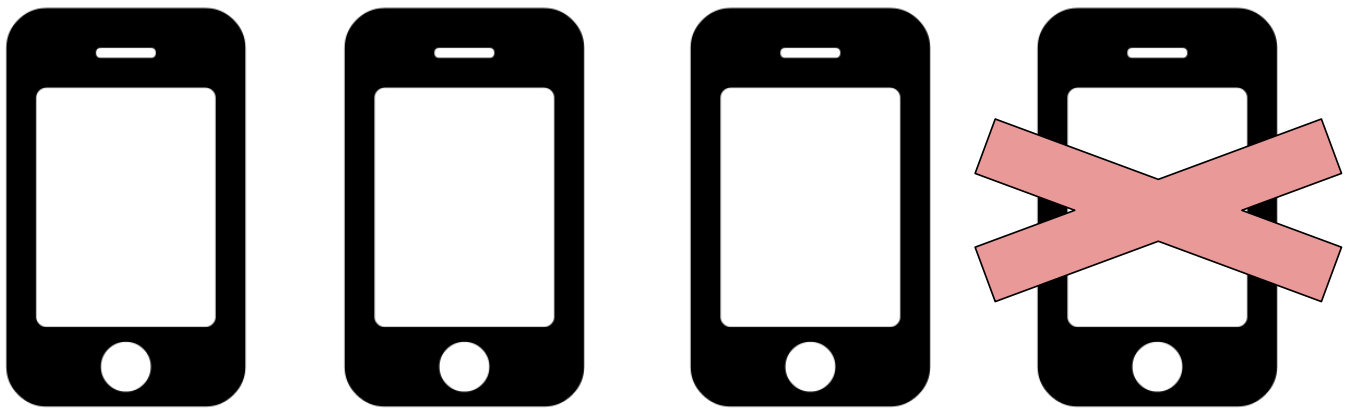
The multi-device problem: Threshold signatures



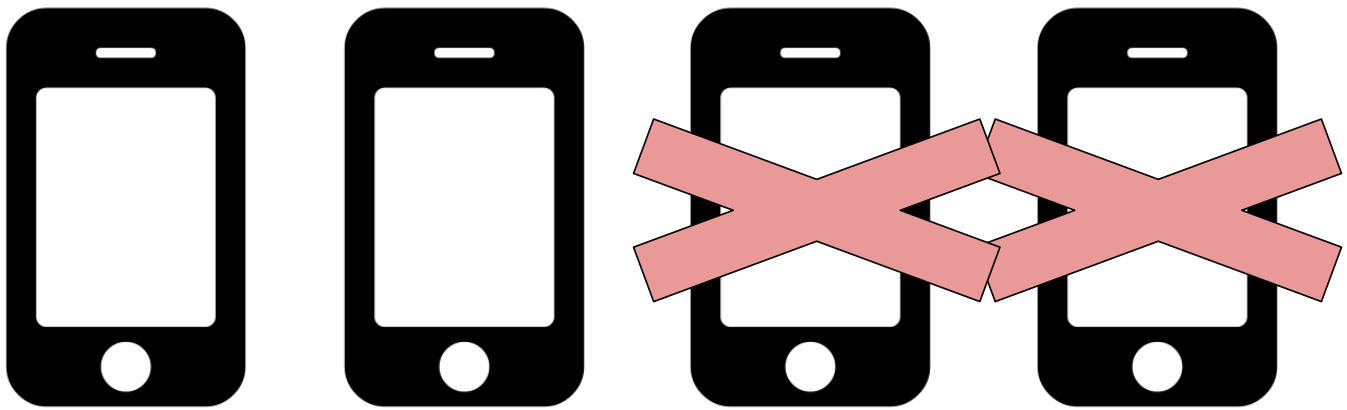
Contribution

- We created **Shatter**, a library and suite of apps for protecting user data distributed across multiple devices with threshold cryptography.
- A user with n devices picks a threshold $2 \leq t \leq n$
  - t devices must work together to perform a cryptographic operation
  - participation does not result in generating the private key
  - $x < t$ compromised devices reveals no information about the private key

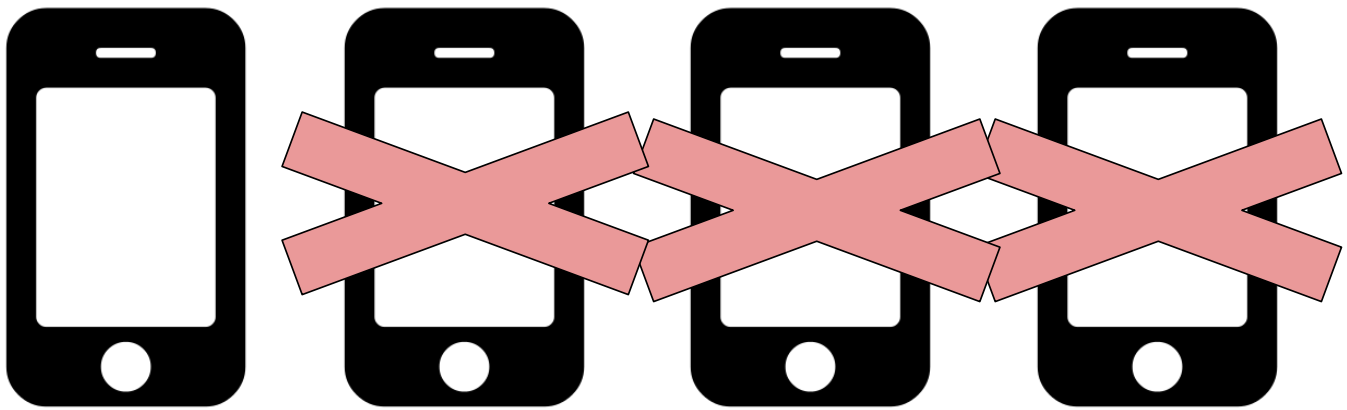
Threat model



Threat model



Threat model



Goals for an ideal solution

Backwards compatibility The scheme can be used to communicate with people who are using unmodified software

Device anonymity The remote party cannot distinguish which of the user's devices were used to perform an operation

Theft resistance goals

Weak theft resistance If only $0 < x < t$ devices are compromised, the long-lived private key remains uncompromised and does not need to be revoked (as long as devices do not automatically participate in requests from other devices).

Strong theft resistance If only $0 < x < t$ devices are compromised, the long-lived private key remains uncompromised and does not need to be revoked (even if devices do automatically participate in requests from other devices).

Device handling Goals

Single public key Remote parties only see a single public key representing the user

No master device / CA All the devices are treated equally; there is no device that acts as a single point of failure

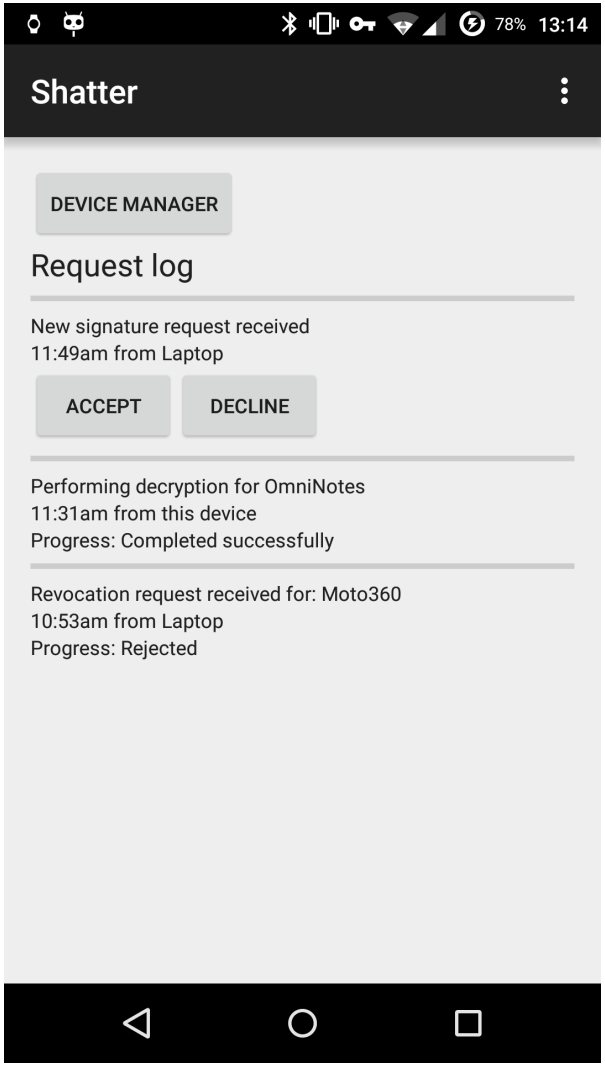
Only one active device The scheme does not require multiple devices to be powered on and in communication with each other in order to perform a necessary operation

One active device

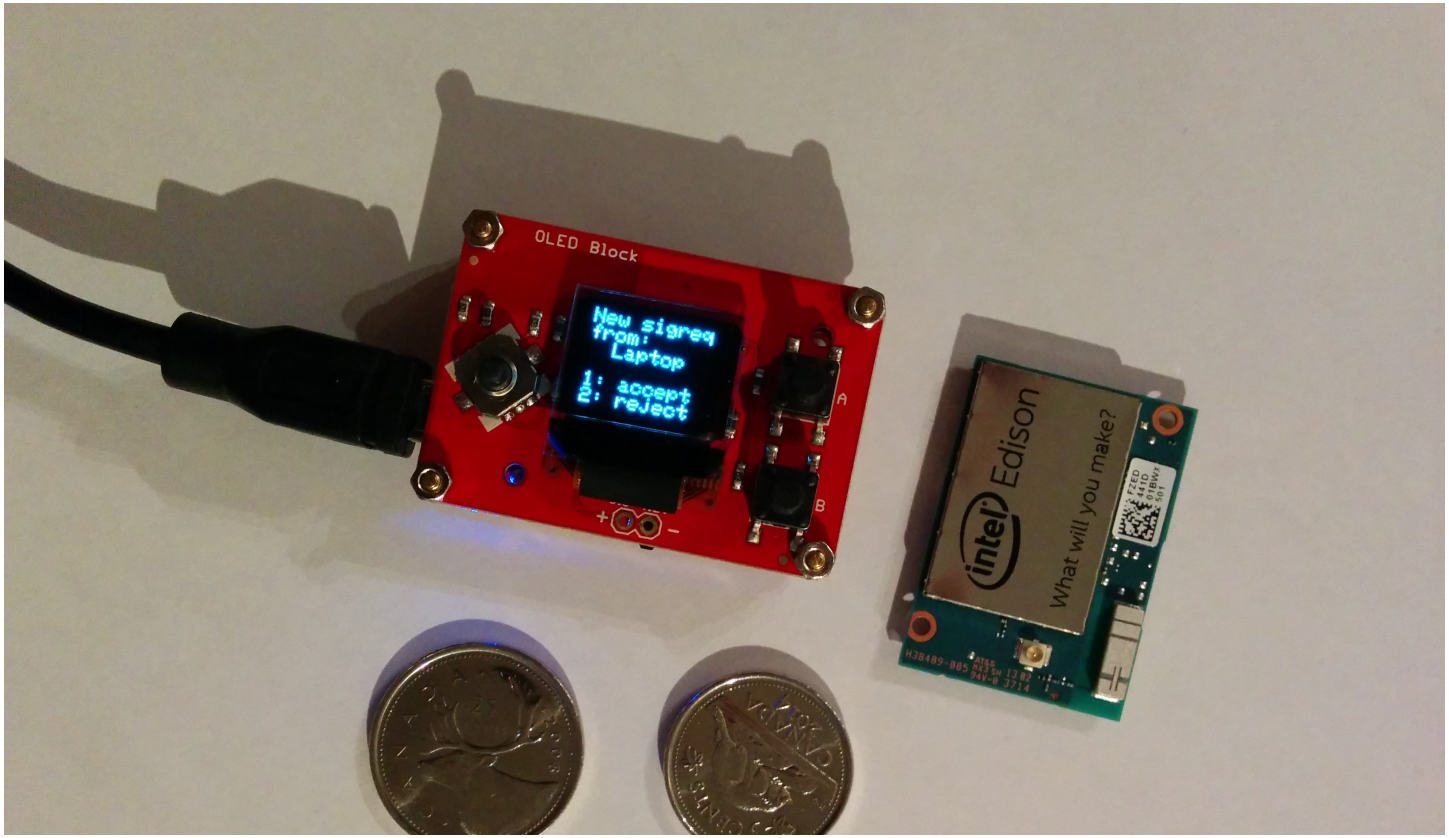
Shatter

- **Shatter** is a core library plus set of supporting applications designed to extend the protections of threshold cryptography to users with multiple devices
- User installs Shatter client on each of their devices – currently we support Windows/OSX/Linux, Android, and Android Wear
- App developers use libShatter to request cryptographic operations be performed using the user's enrolled devices

Shatter Client on Android



Shatter Client on an Intel Edison

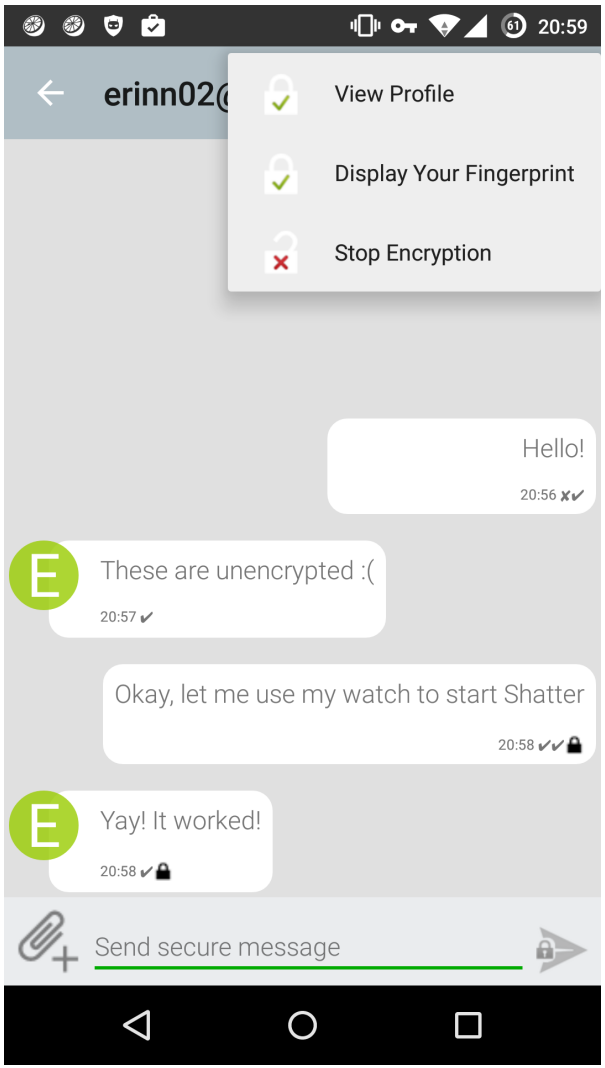


Shatter Clients on a Moto 360

Shatter app: ChatSecure

- First algorithm: Threshold (EC)DSA, by Gennaro et al.
- Allows any app to request a threshold DSA signature from Shatter
  - DSA is the signature algorithm specified in Off-the-Record Messaging (OTR)
- We modified **ChatSecure**, an Android client that supports OTR over arbitrary XMPP servers, to request DSA signatures from a Shatter client installed on the device

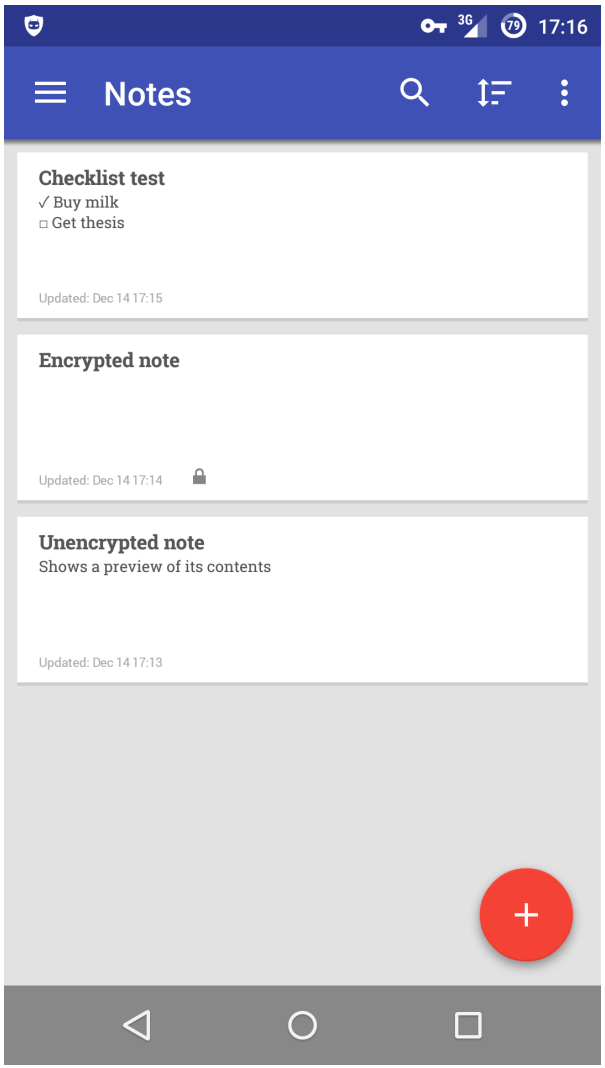
Shatter app: ChatSecure



Shatter app: Omni-Notes

- Next algorithm: Threshold Paillier encryption
- Added support in Shatter for encrypting/decrypting arbitrary plaintexts
- Modified **Omni-Notes**, the most popular open-source Android app for taking notes, to ask Shatter to encrypt notes before storing on disk
  - Decryption is a threshold operation, done on loading

Shatter app: Omni-Notes



Evaluation

- Evaluated threshold DSA and Paillier
- Desktop: Ubuntu 14.04, AMD FX-6100 3.3GHz 6-core CPU
- Phone: Nexus 5, Android 5.1.1
- Watch: Moto 360, Android Wear 5.0.1

Evaluation

Device, role	**time**
Signature, 3 desktops	11
Signature, 3 phones	12
Signature, all unique	13
Decryption, 3 desktops	0.82
Decryption, 3 phones	1.9
Decryption, all unique	2.5

Computation time for $(3, n)$-threshold schemes, in seconds

Evaluation

Device, role	Proofs	No proofs
Signature, 3 desktops	11	0.24
Signature, 3 phones	12	0.51
Signature, all unique	13	0.54
Decryption, 3 desktops	0.82	0.12
Decryption, 3 phones	1.9	0.25
Decryption, all unique	2.5	0.26

Computation time for $(3, n)$-threshold schemes, in seconds

Evaluation

- Bandwidth: One DSA signature takes $6t$ messages; decryption takes $2t$
  - Messages are sent in parallel, so overall time is constant
- Messages range from 4–16 kB

Conclusion

- Shatter protects users with multiple devices using threshold cryptography, offering theft resistance, multi-device key management, and more
- Enables near-seamless end-to-end cryptography on multiple devices
- Algorithms already available for signatures and encryption
- Shatter-aware versions of ChatSecure and OmniNotes
- Free and open source:

https://crysp.uwaterloo.ca/software/shatter/