Deadbolt:

Locking Down Android Disk Encryption

Adam Skillen, David Barrera, and Paul C. van Oorschot

askillen@ccsl.carleton.ca

Carleton UNIVERSITY

Carleton Computer Security Lab
Carleton University
Ottawa, Canada

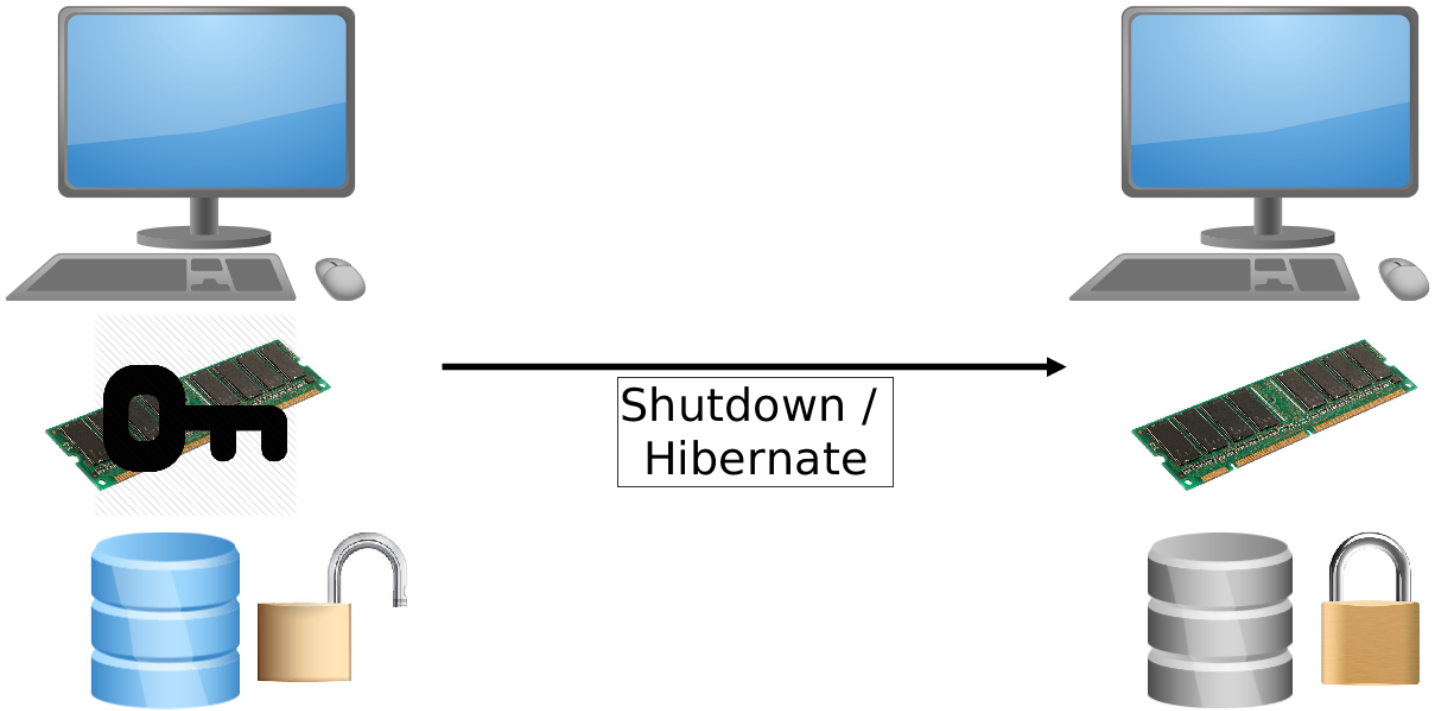
SPSM 2013, Berlin, Germany
November 8, 2013

The problem with Android disk encryption

Carleton UNIVERSITY

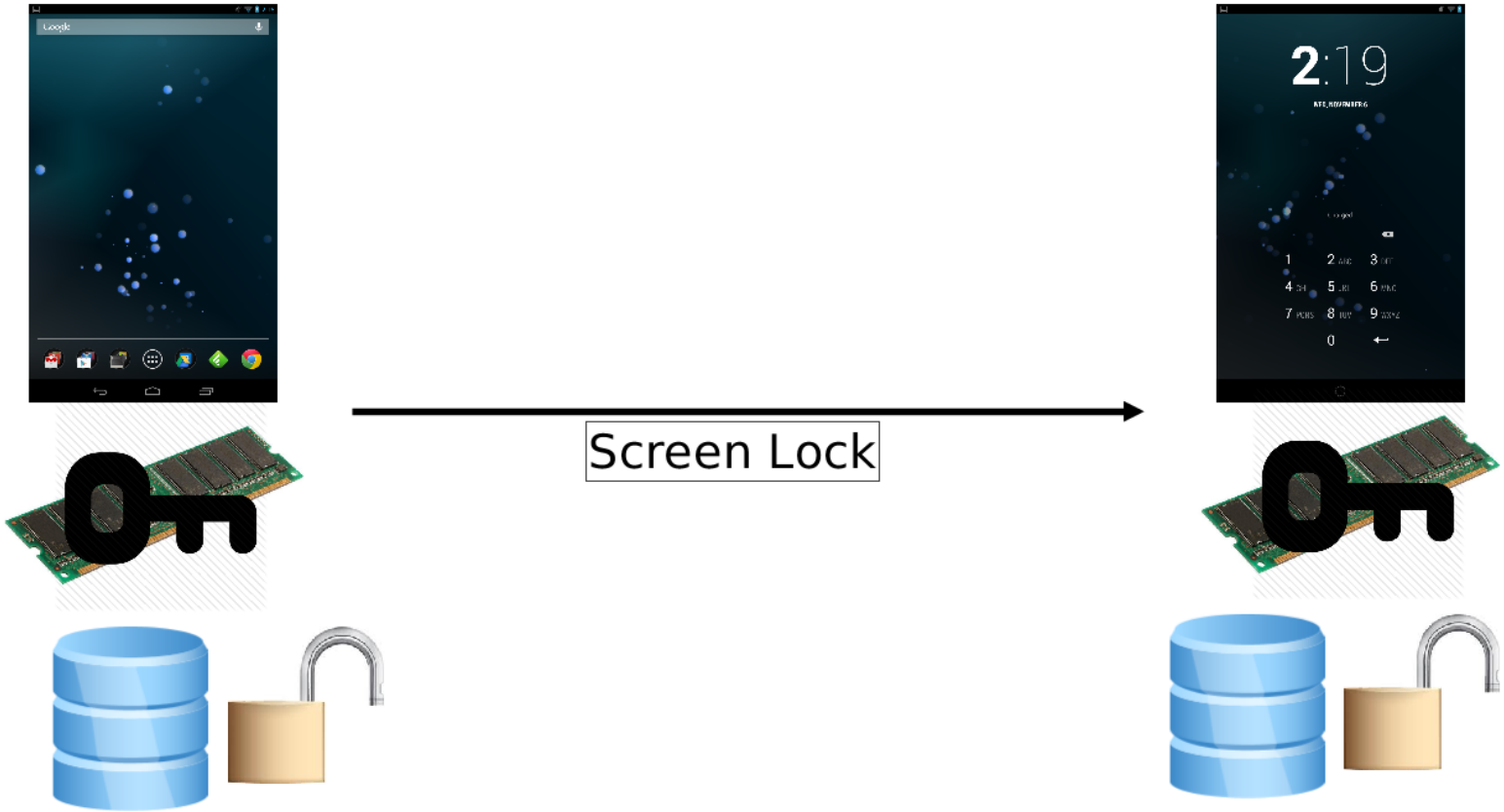
- Android storage encryption uses Full Disk Encryption (FDE).
- Key stays in RAM while *screen-locked*.
- FDE only protects private data when volume is unmapped.
  (e.g., device is shutdown)
- Mobile device *always-on* usage model weakens FDE.
- FDE key and private data are susceptible to cold-boot, lock-screen bypass, and hardware based attacks.

FDE – PC model



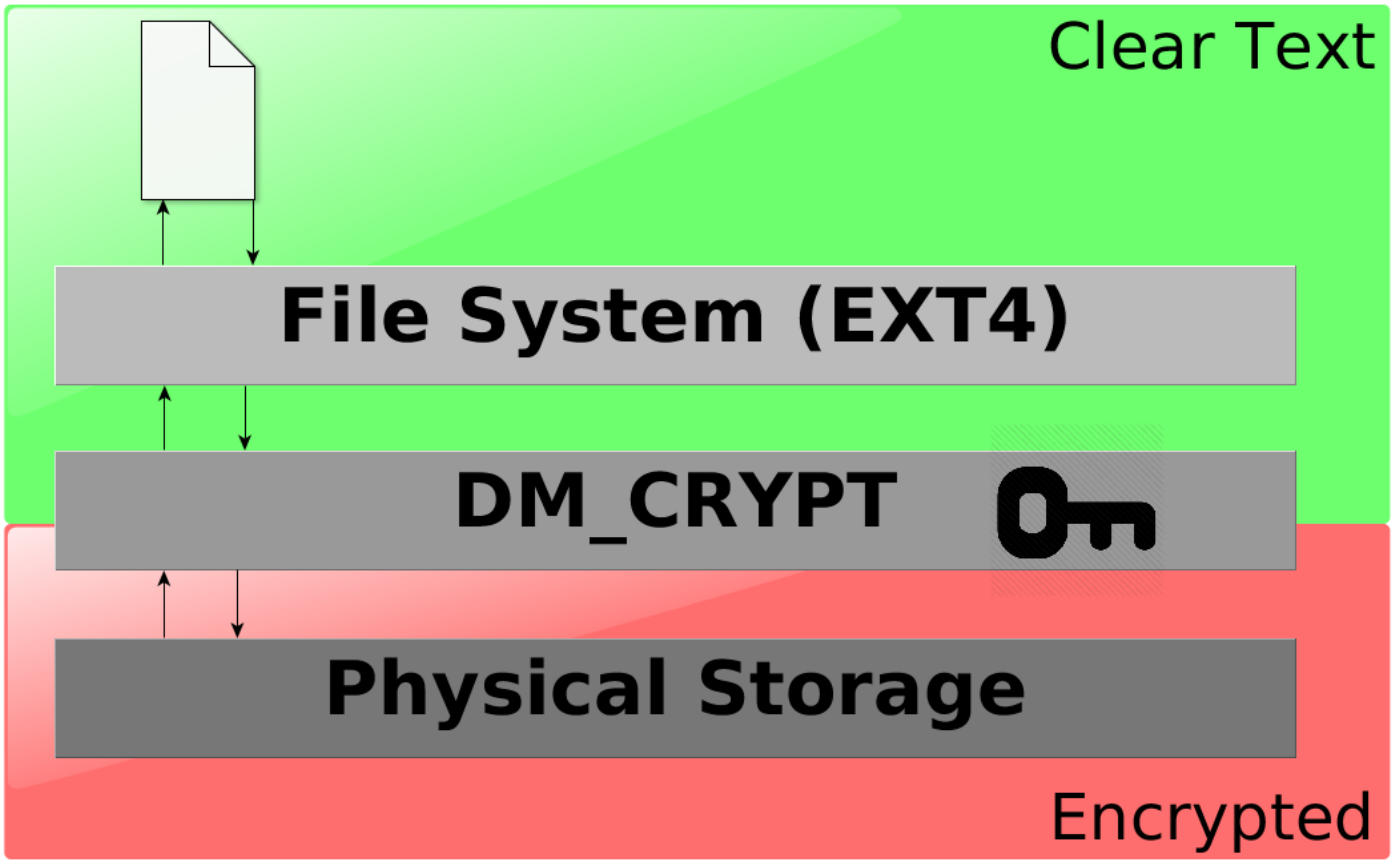
PCs are regularly shut-down or hibernated, effectively securing the encrypted data by removing the key from RAM

FDE – Mobile model



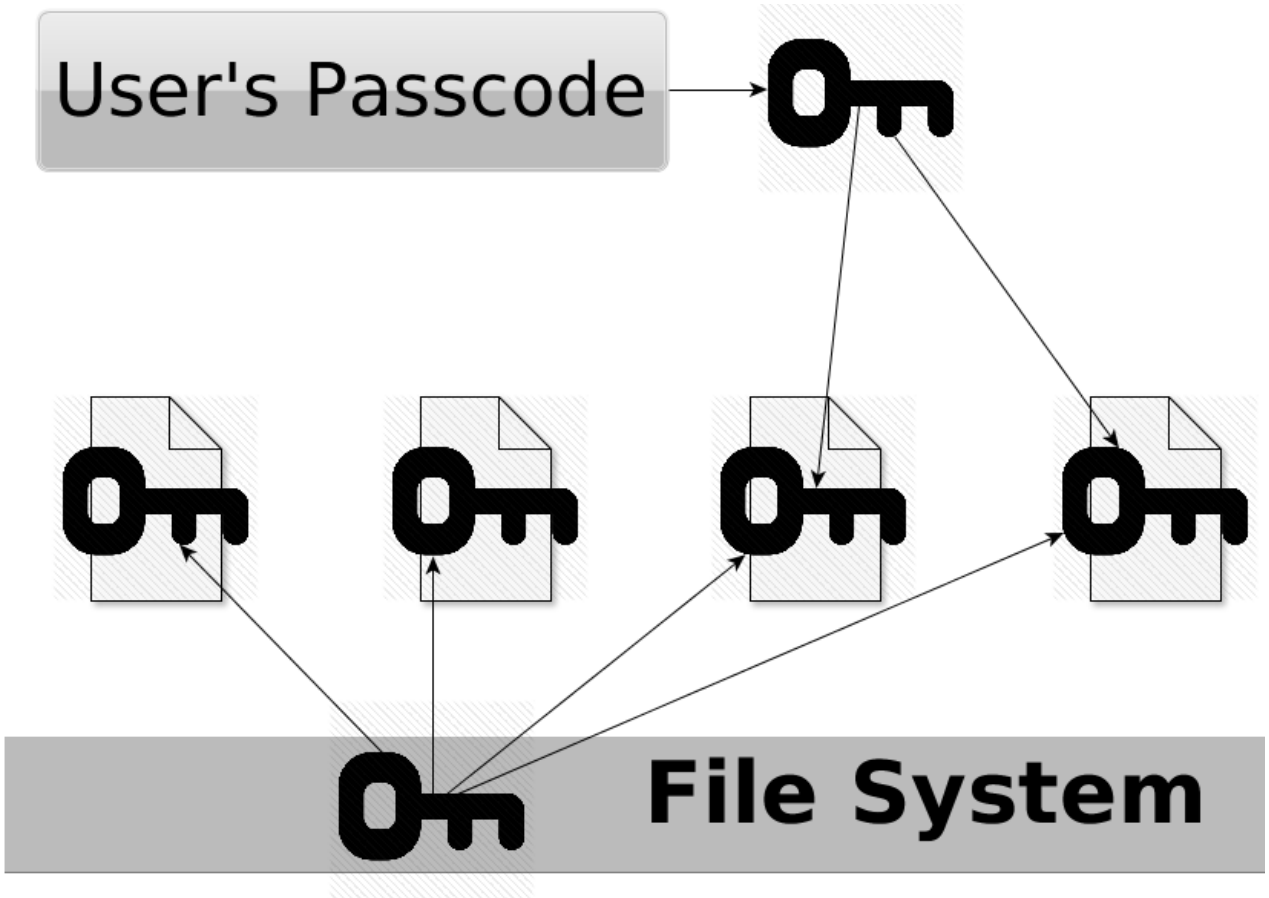
Mobile devices are instead *screen-locked*. The key remains in RAM and volume remains mounted

Android storage encryption



- Implemented through DM_CRYPT
- Block ciphers act on individual disk sectors.
- On-the-fly (transparent to users/apps).

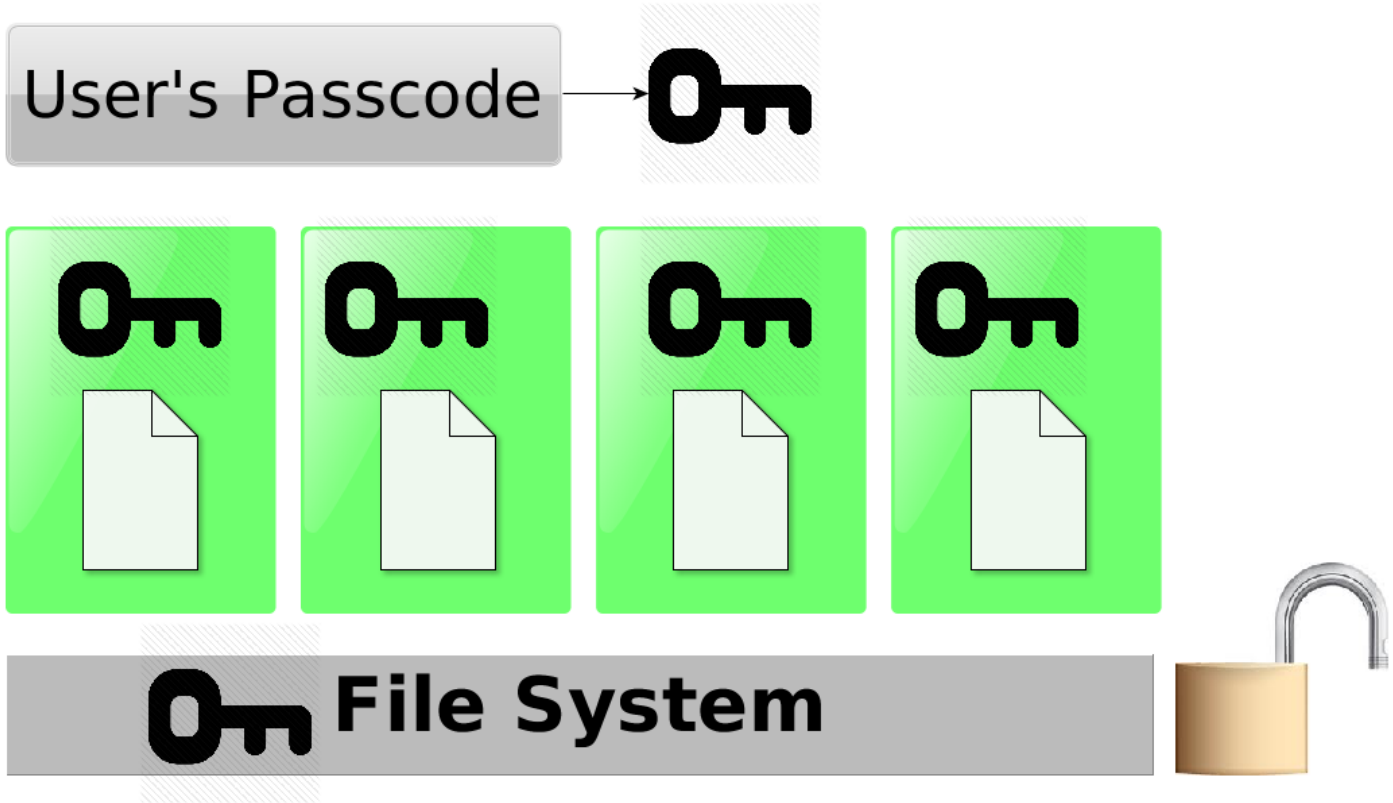
cf. iOS storage encryption



Files are encrypted individually, some keys are removed from RAM when screen-locked

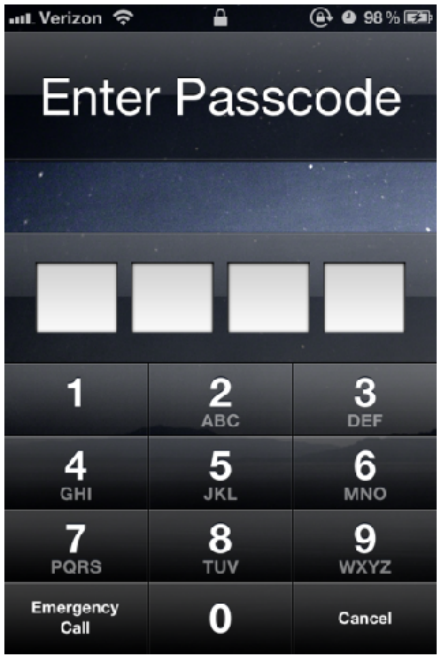
Unlocked iOS device

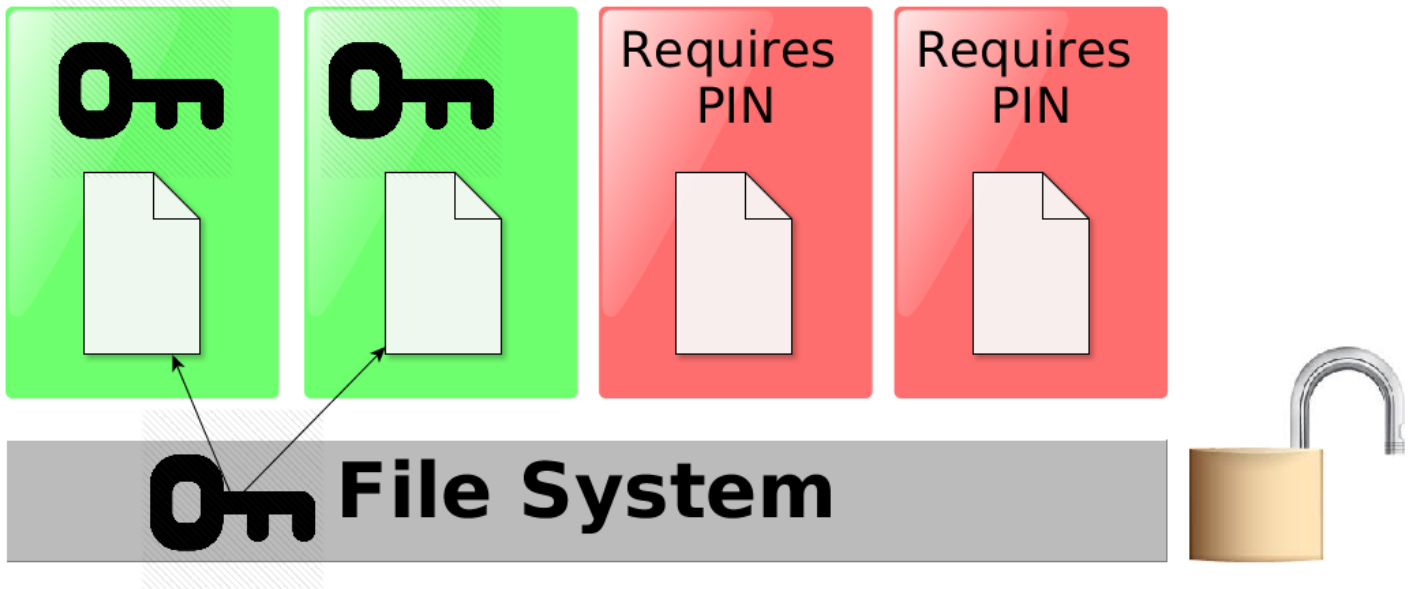
Carleton UNIVERSITY



All keys/files available when screen is unlocked

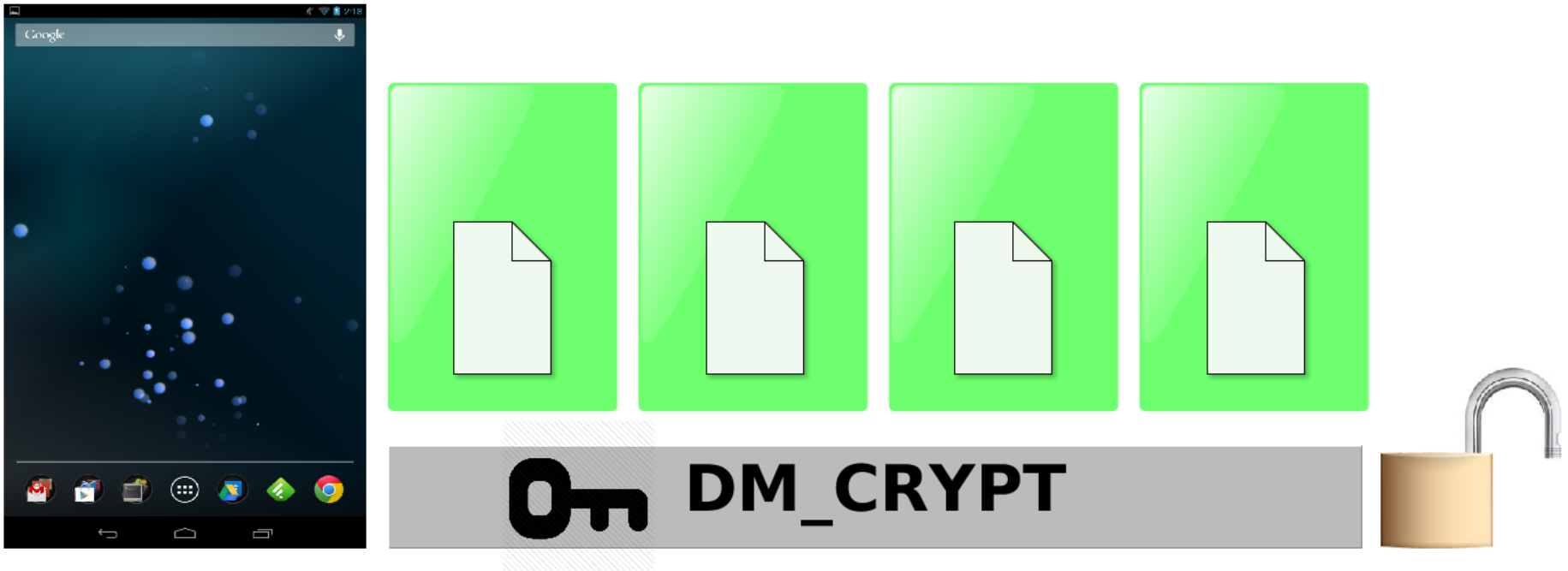
Locked iOS device





Some keys/files available when screen is locked

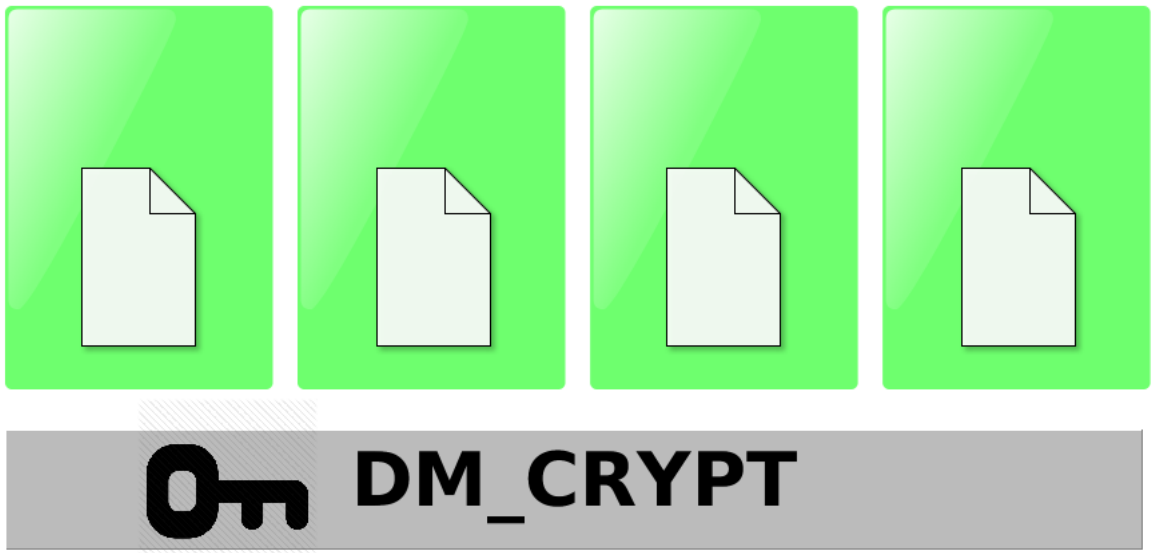
Unlocked Android device



Key and storage available when screen is unlocked

Locked Android device

Carleton UNIVERSITY



Key and storage **remain** available when screen is locked!

Contributions

Carleton UNIVERSITY

1. Software-only method to protect FDE key and encrypted data.
   Resilient to cold-boot and lock-screen bypass while in *Deadbolt* mode.
2. Retains most smart-device functionality.
   (Dialer, SMS, Internet, optionally import some user data).
3. Resuming from Deadbolt mode is faster than a full boot-up.
4. Added benefit of an optional *incognito* environment.
   Logs and activities can be discarded after resuming from Deadbolt.
5. Full design and implementation for use with Android 4.0+
   Source code available from project website.

Use Cases

Carleton UNIVERSITY

- Deadbolt complements the Android lock-screen, for use in high risk situations
  - E.g., travelling, commuting, border-crossing
  - Intended users: anybody that currently uses device encryption
- Optionally, policies could be used to invoke Deadbolt
  - E.g., time-of-day, GPS location
- Incognito mode allows users to perform tasks deniably
  - E.g., phone calls will not show up in logs
- Safe mode allows users to perform potentially hazardous tasks
  - E.g., visit untrusted websites

Threat model

Carleton UNIVERSITY

Assume adversary can obtain physical access to device while in Deadbolt

- **Software vulnerabilities** – lock-screen routinely bypassed (e.g., recent Android Skype bug, iOS 7 bug).
- **Cold boot attack** – keys and intermediate state in RAM, Müller et al. recently demonstrated cold boot on Android [ACNS'13].
- **Hardware attacks** – ARM debug interface, JTAG, etc.

Deadbolt implementation

Carleton UNIVERSITY

Implemented in the Android volume mounting daemon (`vold`)

1. Pause running Android framework (GUI, daemons, etc.)
2. Unmount encrypted `userdata` volume.
3. Zero all key material in RAM.
4. Mount empty `tmpfs` (RAM filesystem) on `/userdata`.
5. Restart Android framework.

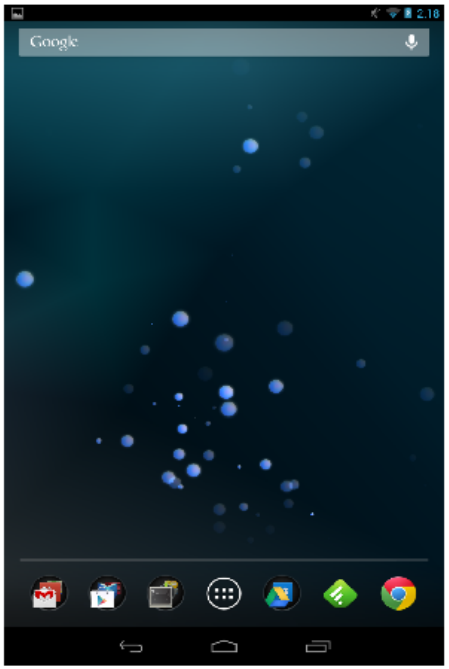
cf. Switching runlevel without restarting kernel.

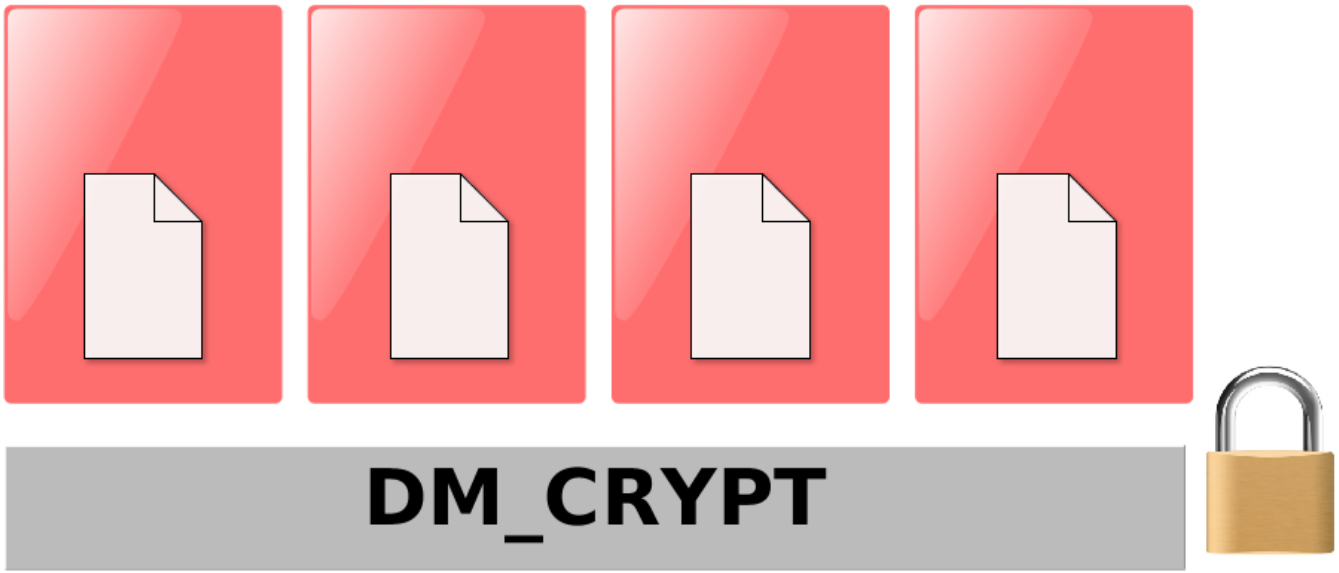
Deadbolt environment

Carleton UNIVERSITY

- Uninitialized environment.
  Default settings, no user data/apps.
- Base system apps (without user data).
  Sufficient for phone, web, texting, maps/GPS.
- `tmpfs` mounted to `userdata` storage.
  Private data inaccessible, all changes must be exported or lost.
- Optionally import certain data.
  E.g., contacts, WiFi passwords, etc.

Deadbolted Android device





Key and storage secured, core smartphone functionality retained

Deadbolt modes

Incognito mode is like a Live-CD environment: no data persists after exiting (Default mode)

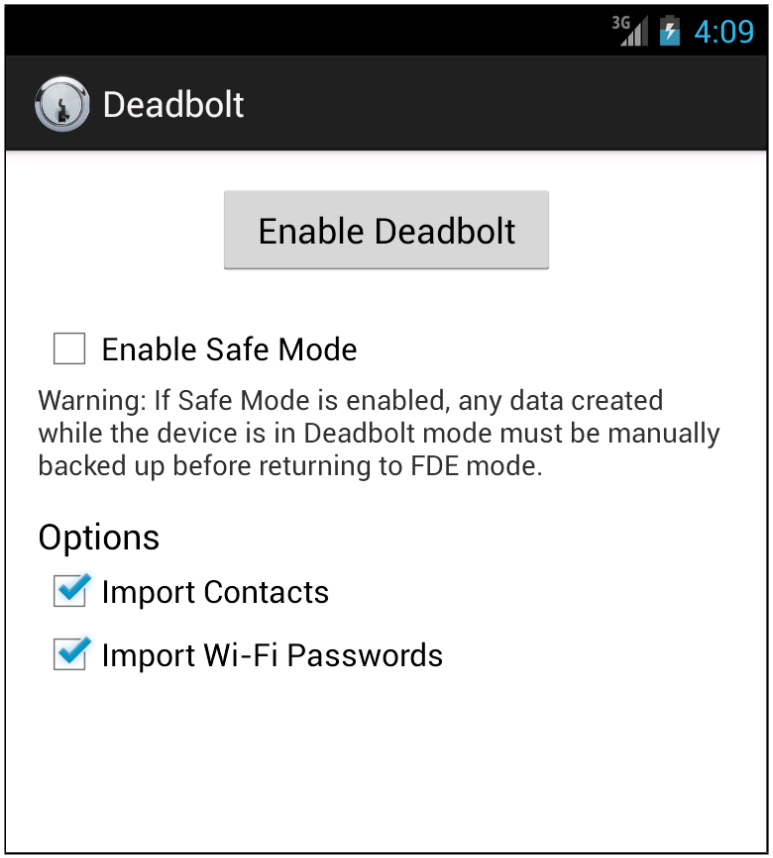
Allows importing/exporting data to encrypted storage

Safe mode allows users to perform potentially dangerous tasks without the risk of disclosing private data

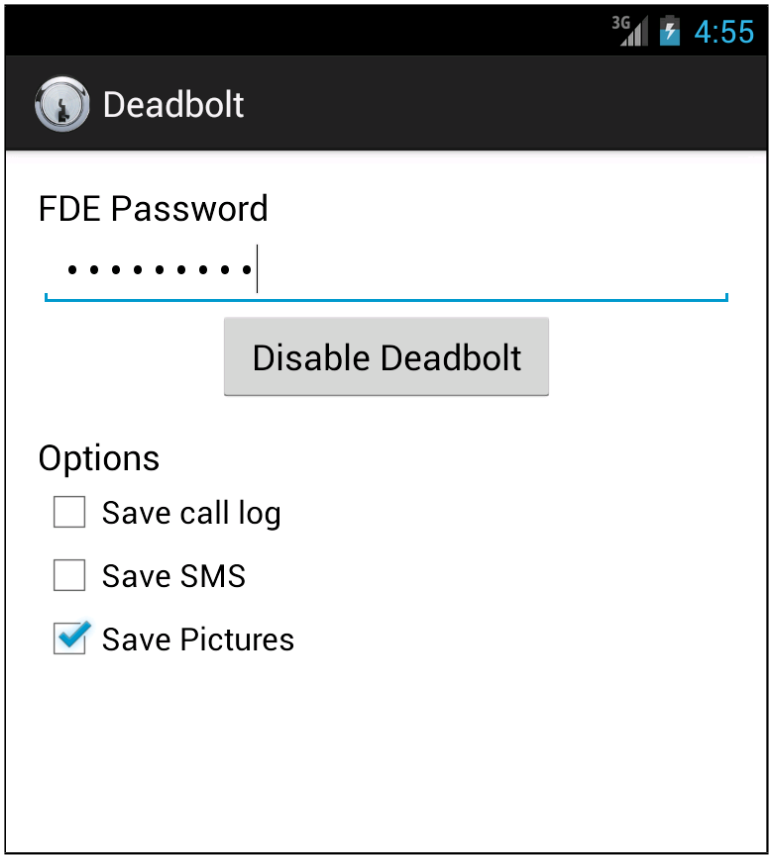
Importing/exporting of private data is disabled

Deadbolt UI

Carleton UNIVERSITY



Enter Deadbolt
(Suspend full environment)



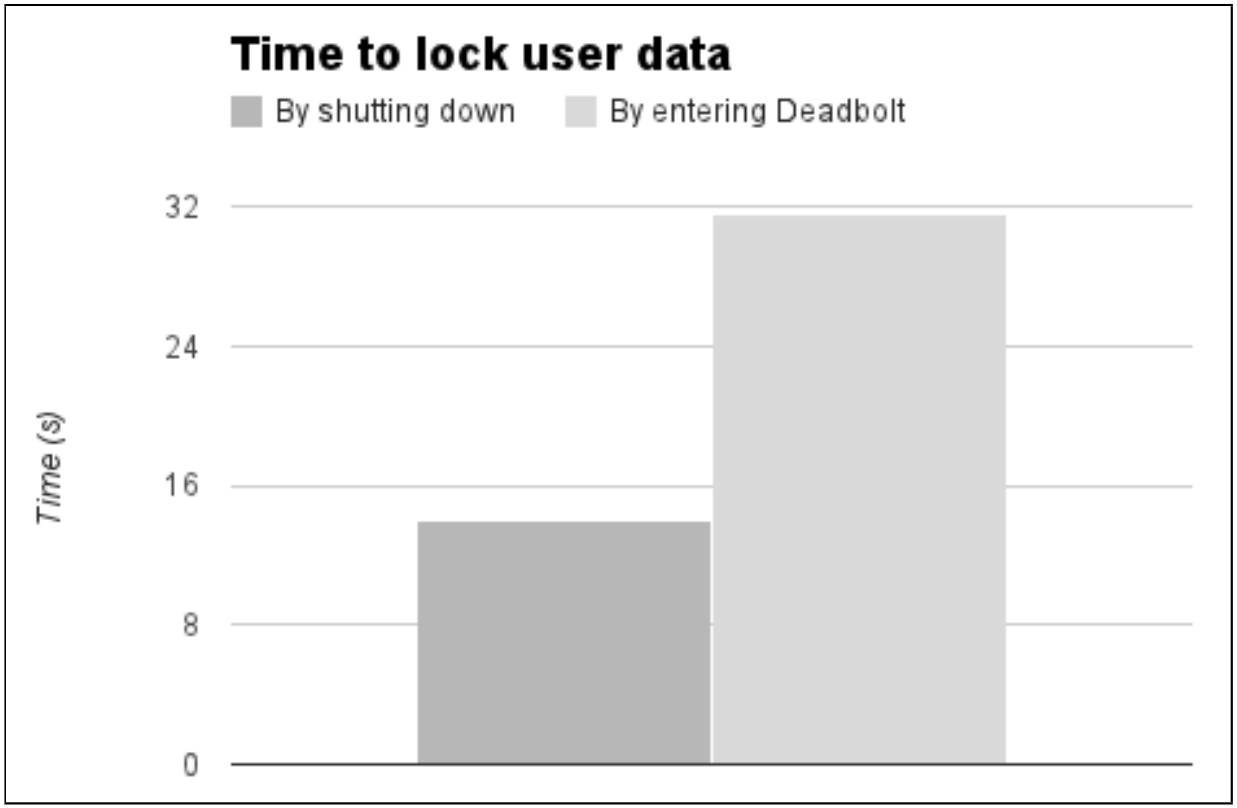
Exit Deadbolt
(Resume full environment)

Deadbolt performance – Locking data

Carleton UNIVERSITY

Tested on Nexus 7 tablet with AOSP 4.2.2

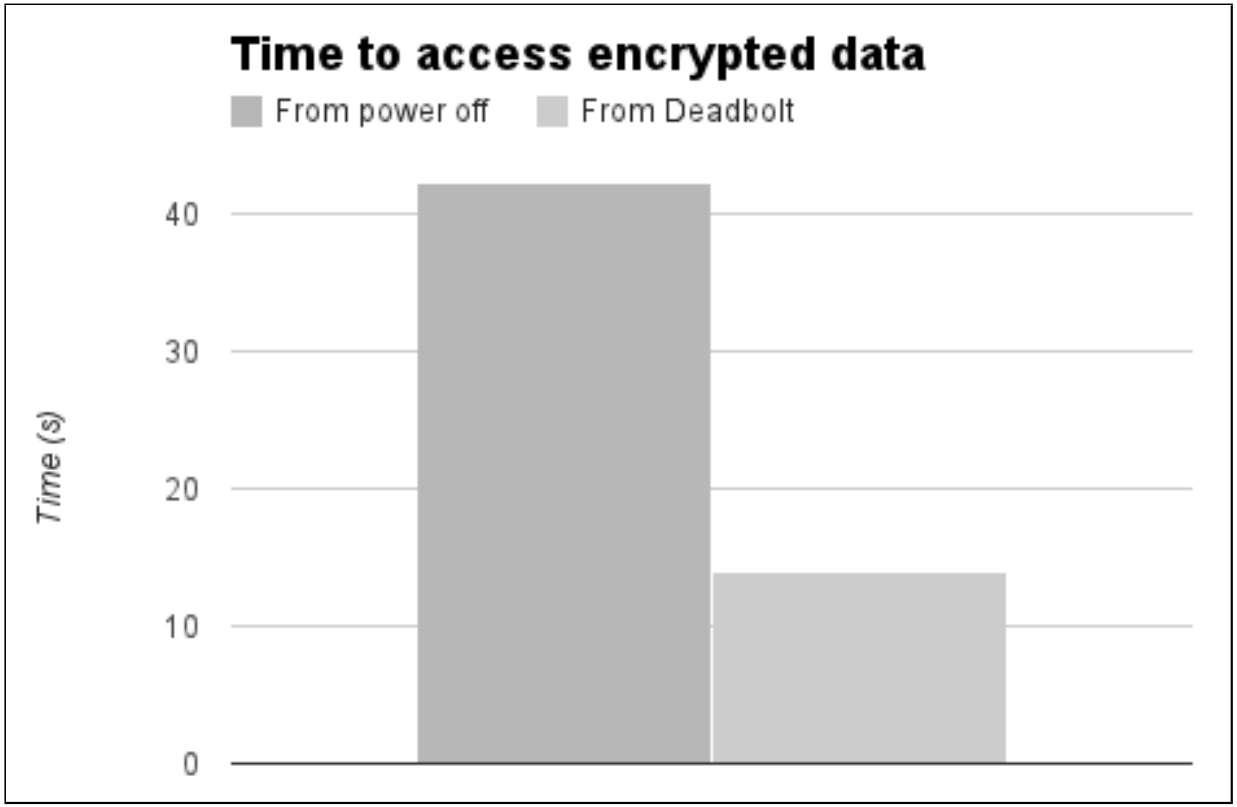
- Power off device:
  $14.03s$ ($\sigma = 0.145$)
- Enter Deadbolt:
  $31.62s$ ($\sigma = 1.235$)



Deadbolt performance – Unlocking data

Carleton UNIVERSITY

- Boot up:
  $42.17s$ ($\sigma = 0.638$)
- Exit Deadbolt:
  $14.00s$ ($\sigma = 0.122$)



Trade increased time to lock for decreased time to unlock, and maintain core functionality

Limitations and usability concerns

Carleton UNIVERSITY

- Absence of user apps and data (e.g., games, email passwords).
- App notifications must use other means (e.g., over SMS).
- Minimum 256 MB RAM (Android 4.0+ devices).
- Cannot be installed after market, must be implemented in OS.
  (Can possibly be made part of default Android OS).
- Private data fragments may remain in RAM.

Summary

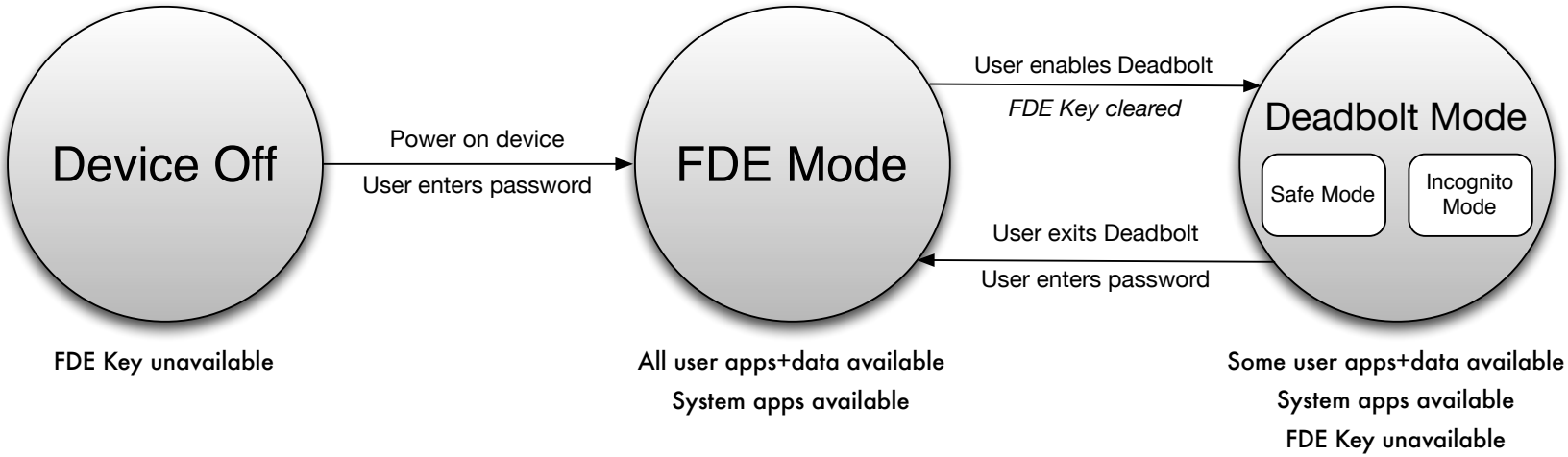
Carleton UNIVERSITY

- With FDE, data remains *unlocked* while device powered on.
- Deadbolt offers security benefits of a powered off device while retaining most mobile functionality.
- Switching to Deadbolt faster than reboot.
- Some usability/security trade-offs.

Deadbolt project website:

`http://www.ccsl.carleton.ca/~askillen/deadbolt`

Deadbolt overview



- Enhanced Android lock-screen.
- All private data encrypted and inaccessible.
- Temporary (empty) Android environment.
- Core phone functionality available.

Deadbolt comparison

Carleton UNIVERSITY

		Lock-screen bypass resilient	Cold-boot resilient	Software only	App notifications	Incognito mode
File	Apple iOS		●		●	
	BlackBerry	●	●	●	●	
FDE	Windows Phone				●	
	Android FDE			●	●	
	Deadbolt	●	●	●	○[a]	●

[a] (e.g., over SMS)

Alternative approaches and future work

Carleton UNIVERSITY

- Exiting Deadbolt is fast (only requires restarting GUI/services)
  - Suspend to disk (likely not an advantage given Android's memory model)
- Entering Deadbolt is slower (requires creating directory structure, unpacking system apps, restart framework)
  - Pre-created disk image could be used with OverlayFS (RO, COW)
  - Trusted execution implementation (key only available inside TEE)

Data transfer between environments

Carleton UNIVERSITY

Copy files and merge SQLite databases while `tmpfs` and FDE storage mounted concurrently.

- **Import** – Optionally import some data into Deadbolt.
  Any imported data is susceptible to disclosure.
  E.g., contacts, WiFi settings/passwords, bookmarks.
- **Export** – Save some data created in Deadbolt.
  E.g., call log, SMS/MMS, photos.

Security evaluation

Carleton UNIVERSITY

- `dm-crypt` uses `kzfree` on key material when unmapped
- We wipe `vold`'s copy of the key/password (using `memset`)
- Used LiME and AESKeyFind to examine memory in Deadbolt
- Plaintext private data fragments may exist in RAM.
- When exiting Deadbolt, we wipe the tmpfs
- Data imported into Deadbolt is subject to disclosure while in Deadbolt

Related Work

Carleton UNIVERSITY

J. Gözfried and T. Müler. ARMORED: CPU-bound encryption for Android-driven ARM devices (ARES 2013).

Key stored in CPU registers rather than RAM. (Defence against cold boot, but still susceptible to physical attack and lock-screen bypass)