CS 5472 - Advanced Topics in Computer Security

Topic 5: Deniable Encryption (1)

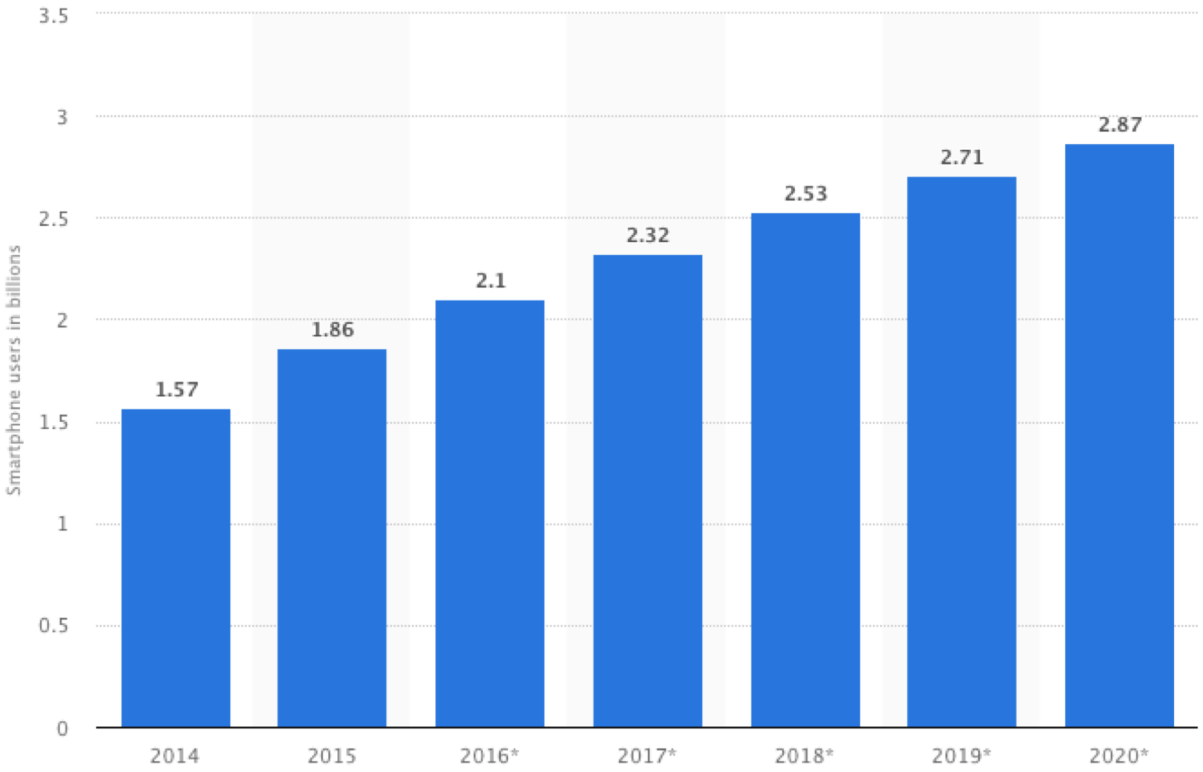
Spring 2021 Semester
Instructor: Bo Chen
bchen@mtu.edu
https://cs.mtu.edu/~bchen
https://snp.cs.mtu.edu

Review: IoT Security

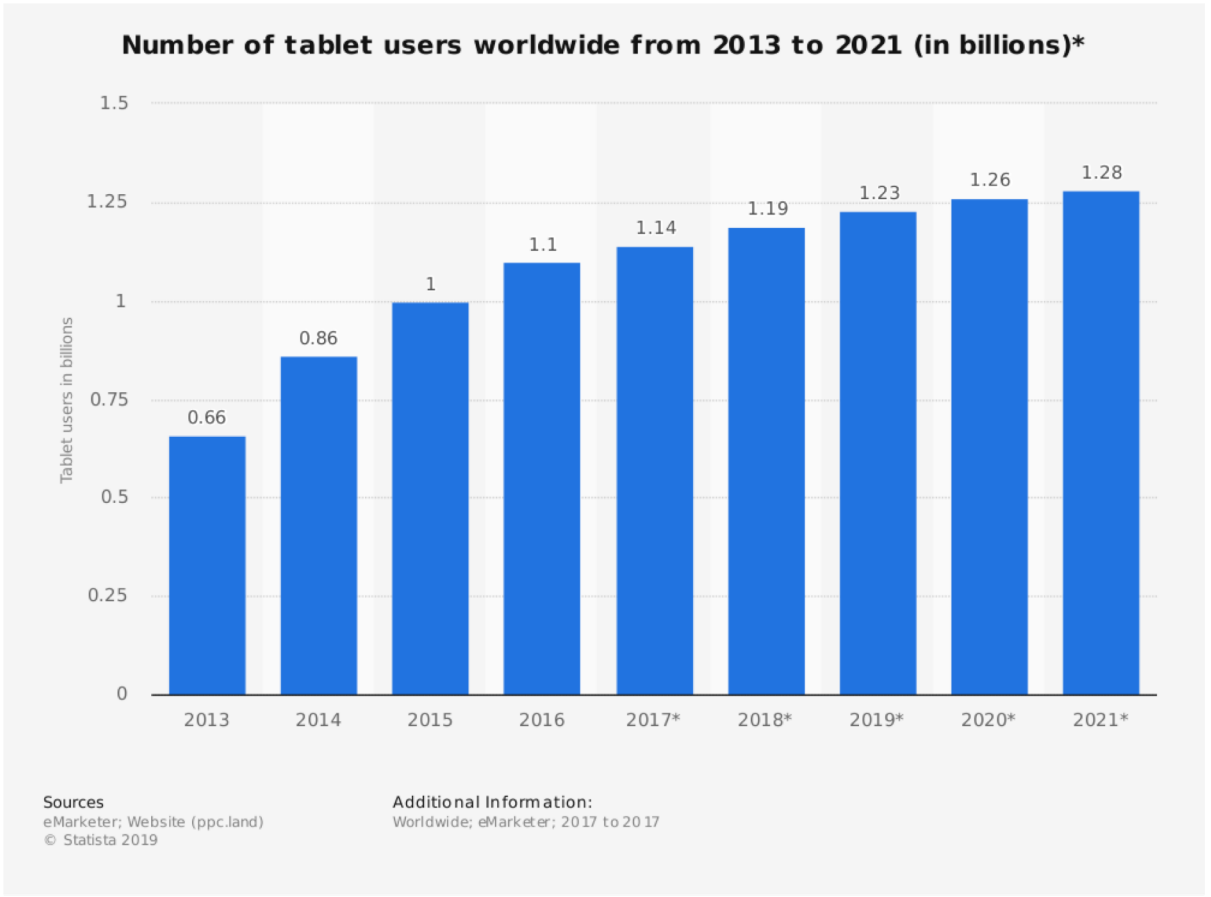
- Internet of Things (IoT)
- Smart home

IoT will not be possible without the mobile devices/IoT devices

Mobile Devices are Turning to Mainstream Computing Devices

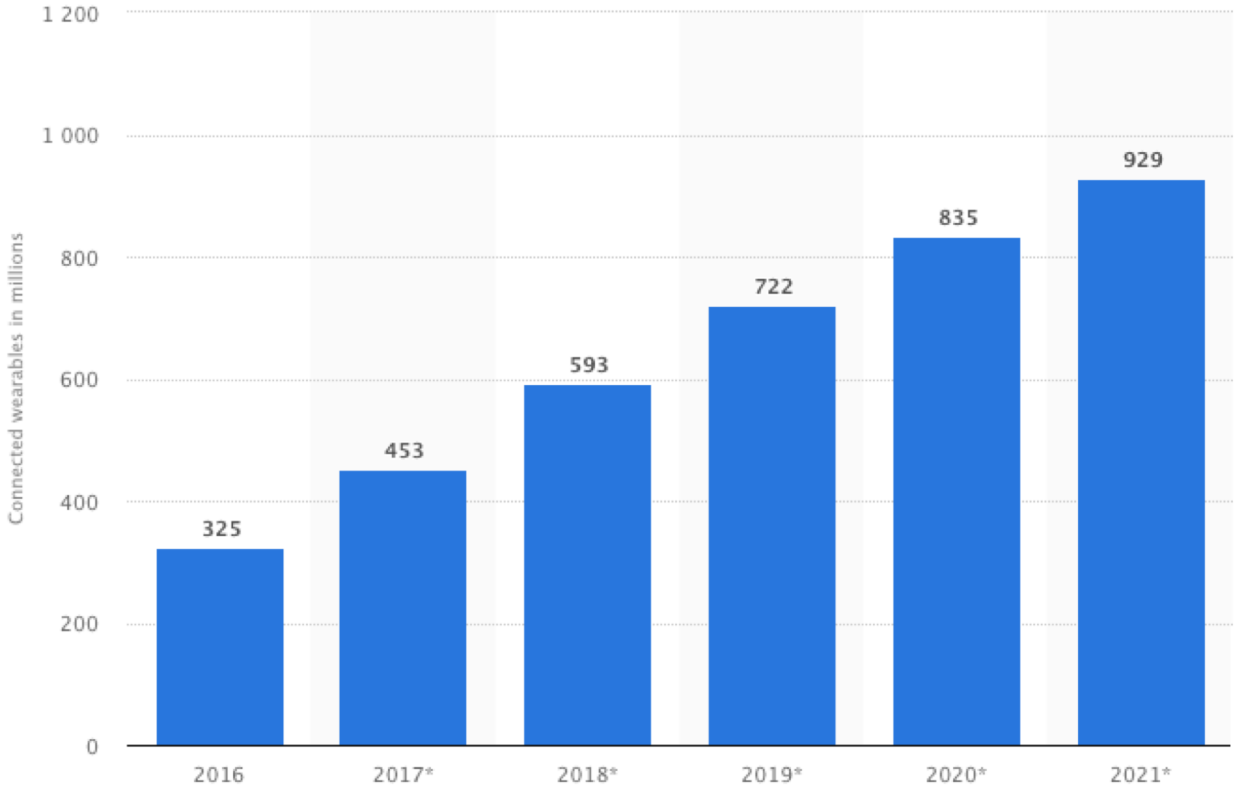


Number of smartphone users worldwide from 2014 to 2020 (in billions)



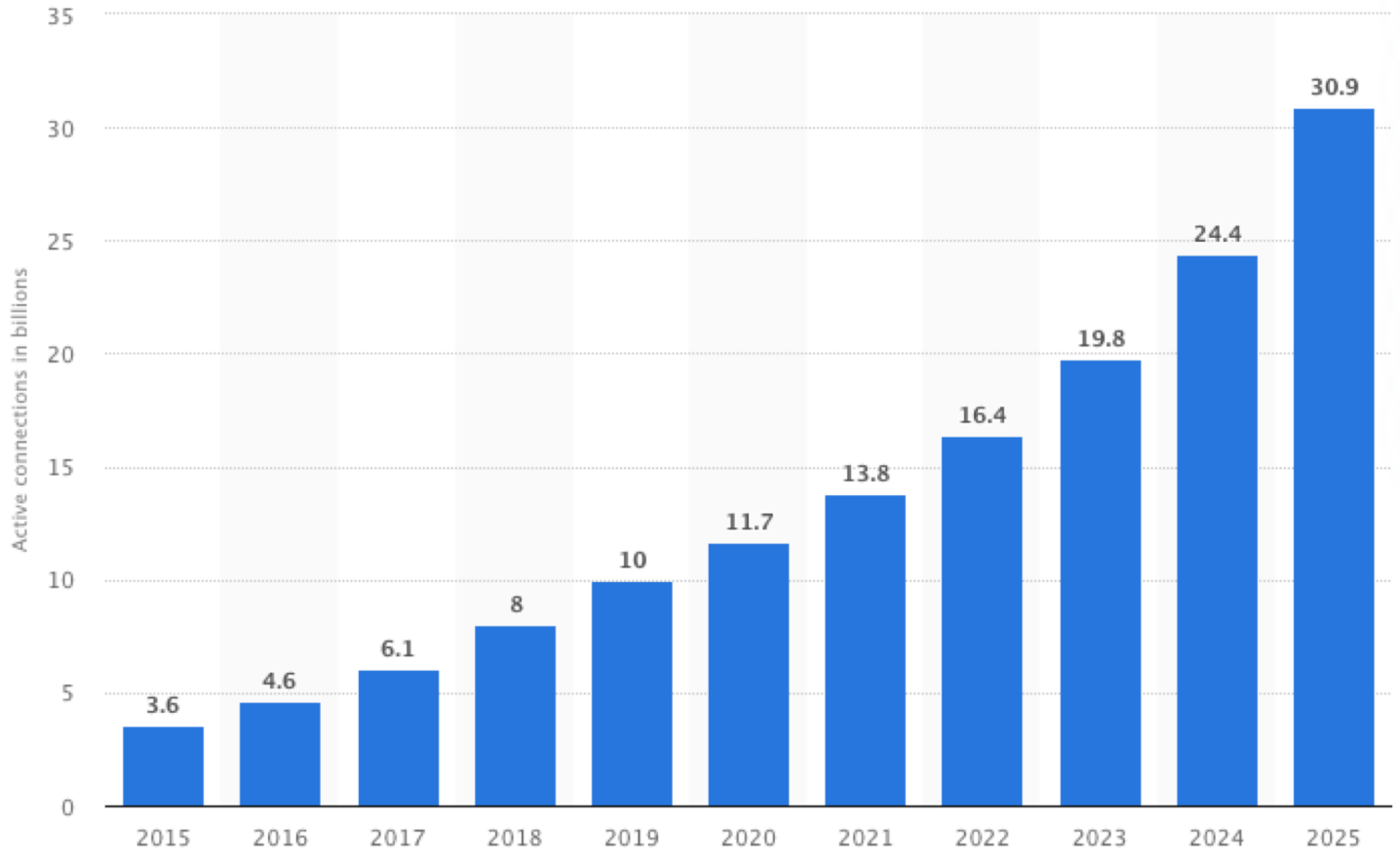
Number of tablet users worldwide from 2013 to 2021 (in billions)

Mobile Devices are Turning to Mainstream Computing Devices (cont.)



Number of connected wearable devices worldwide from 2016 to 2021 (in millions)

Mobile Devices are Turning to Mainstream Computing Devices (cont.)



Internet of Things (IoT) connected devices

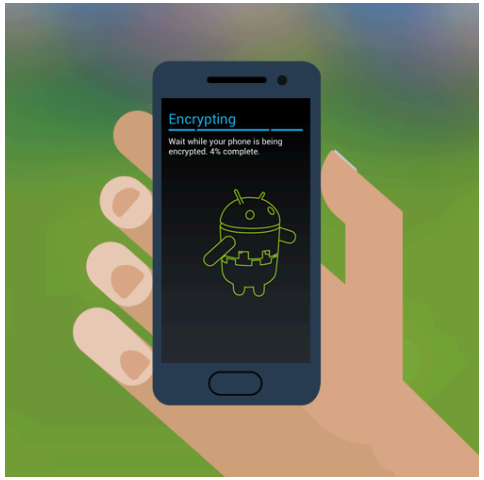
Mobile Devices are Increasingly Used for Critical Applications

- Mobile devices are increasingly used to handle sensitive data
  - Online banking
  - Ecommerce
  - Cryptocurrency/stock trading
  - Naked photos
  - A human rights worker collects evidence of atrocities in a region of oppression
  - Etc.

- Security issues in mobile computing devices
  - Confidentiality
  - Integrity and recoverability
  - Authentication
  - Access control
  - Malware detection and removal

How to Ensure Confidentiality of Data in Mobile Devices

- Main-stream mobile devices usually integrate full disk encryption (FDE)
  - FDE is available in Android phones since Android 3.0
  - Since iPhone 3G S, Apple has consistently built 256-bit AES encryption into iOS devices.



  - Other popular disk encryption tools: TrueCrypt/VeraCrypt, BitLocker (MicroSoft), FileVault (Apple), LUKS

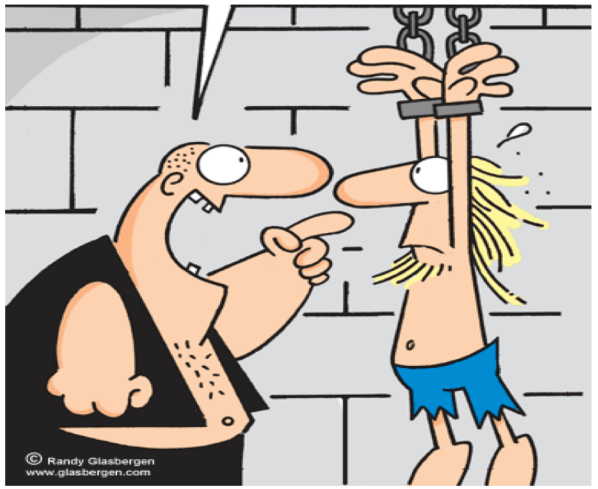
Disk Encryption Is Vulnerable to Coercive Attacks

- Disk encryption usually relies on symmetric encryption (relying on a secret key)
  - AES
  - 3DES

- Conventional encryption is vulnerable to a coercive attack

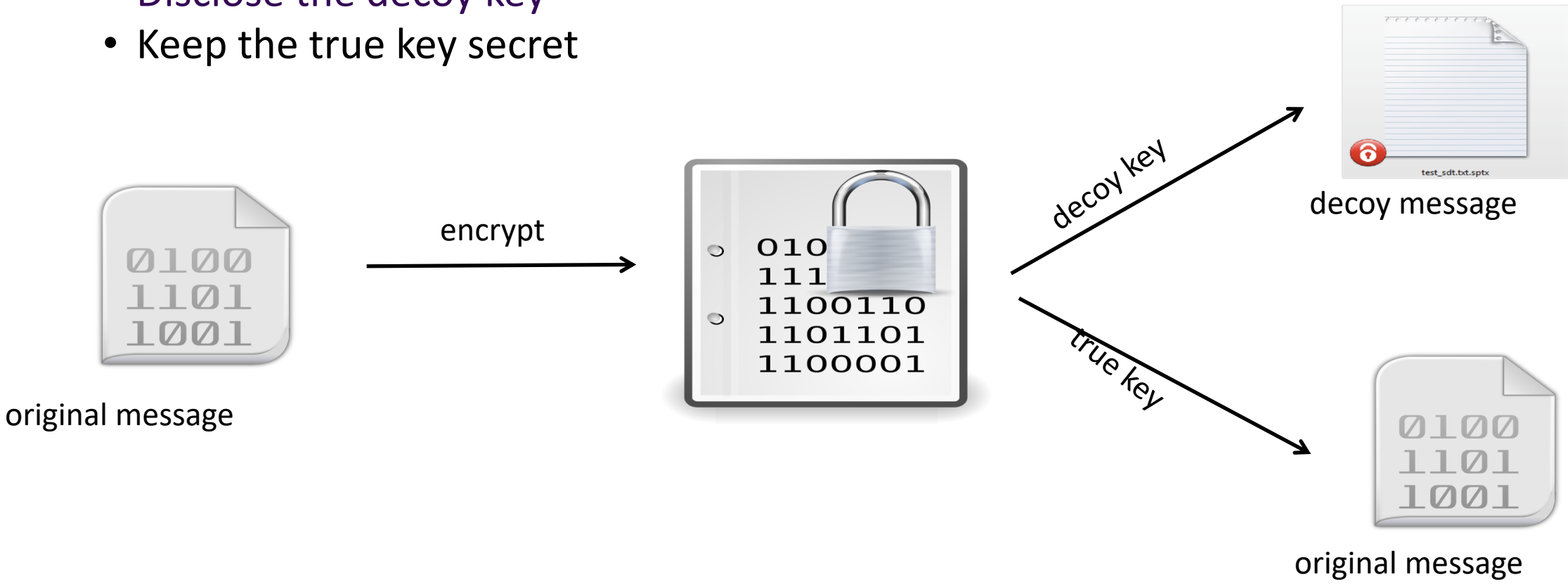
An attacker forces the device's owner to disclose the decryption key

TELL ME YOUR KEY!!!

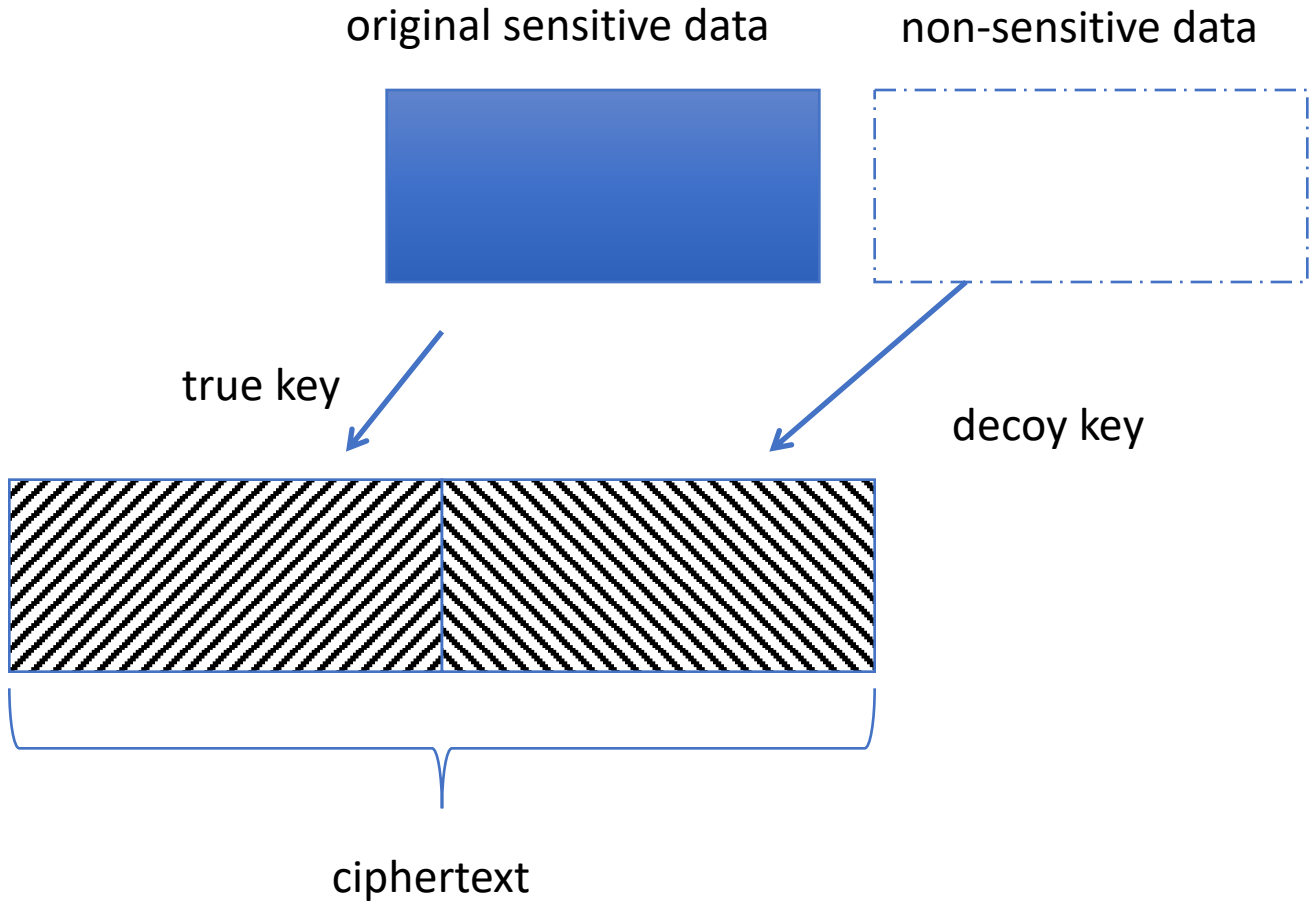


Plausible Deniable Encryption (PDE)

- Plausible Deniable Encryption (PDE) [Canetti et al., CRYPTO ’97]: a crypto primitive designed for mitigating coercive attacks
  - Disclose the decoy key
  - Keep the true key secret



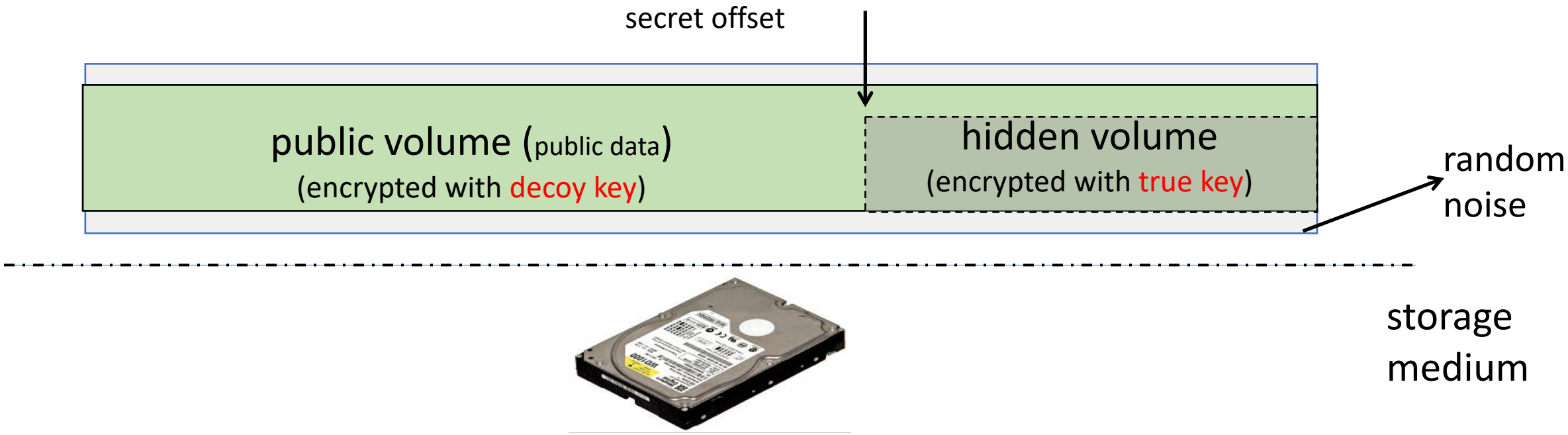
Instantiate PDE in Cryptography



- Issues: the size of ciphertext is increased. Deniability is easily compromised

Implementing PDE in Systems - Hidden Volume

- Hidden volume [TRUECRYPT '04] realizes the concept of PDE in systems
  - Only the decoy key will be disclosed
  - The encrypted hidden volume cannot be differentiated from the random noise



Implementing PDE in Systems – Steganographic File Systems

- Option 1:
  - A few cover files in the systems, and the hidden file is an XOR of these cover files
- Option 2:
  - The file system is initially filled completely with blocks of random data. The file blocks of the hidden file are hidden amongst this random data

A common limitation is, the hidden file may be over-written by the regular files, and we need to store a few redundant copies across the disk.

Storage Architecture in a Mobile Device

Applications layer

Files, APPs

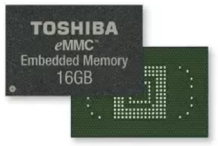
Mobile file system layer

EXT4, EXT3, EXT2, etc.
Implement system calls like open, read, write, etc

Manage the mappings between the applications' view and the block device's view

Block device layer

Flash memory layer



eMMC Chip

MMC Card

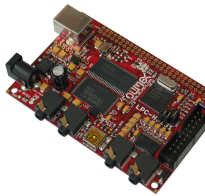
Research Problems

- How to incorporate PDE concept into real-world mobile devices to allow the device's owner to survive when facing coercive attacks?
  - Smart phones (e.g., Android phones)
  - Wearable devices (e.g., Android wear smart watches)

- What need to be achieved
  - Security: provide deniability against a coercive adversary who can capture the device owner and the device
    - No deniability leakages in memory/external storage media
    - Defend against a multiple-snapshot adversary
  - Multiple deniability levels: allow different levels of data protection
  - Fast mode switching: can fast switch to the hidden operating mode
  - Compatibility: compatible with different file systems
  - Efficiency: mobile devices are usually light-weight (limited computational power and battery)
  - Etc.

The Efforts of My Research Group on Building PDE Systems for Mobile Devices

- Niusen Chen, **Bo Chen**, and Weisong Shi. MobiWear: A Plausibly Deniable Encryption System for Wearable Mobile Devices. The First EAI International Conference on Applied Cryptography in Computer and Communications(AC3 ’21), Xiamen, China, May 2021.
- Bing Chang, Fengwei Zhang, **Bo Chen**, Yingjiu Li, Wen Tao Zhu, Yangguang Tian, Zhan Wang, and Albert Ching. MobiCeal: Towards Secure and Practical Plausibly Deniable Encryption on Mobile Devices. The 48th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '18), June 2018.
- Bing Chang, Yao Cheng, **Bo Chen**, Fengwei Zhang, Wen Tao Zhu, Yingjiu Li, and Zhan Wang. User-Friendly Deniable Storage for Mobile Devices. *Elsevier Computers & Security*, vol. 72, pp. 163-174 January 2018.
- Shijie Jia, Luning Xia, **Bo Chen**, and Peng Liu. DEFTL: Implementing Plausibly Deniable Encryption in Flash Translation Layer. 2017 ACM Conference on Computer and Communications Security (*CCS '17*), Dallas, Texas, USA, Oct 30 - Nov 3, 2017.
- Bing Chang, Zhan Wang, **Bo Chen**, and Fengwei Zhang. MobiPluto: File System Friendly Deniable Storage for Mobile Devices. 2015 Annual Computer Security Applications Conference (*ACSAC '15*), Los Angeles, California, USA, December 2015.
- Xingjie Yu, **Bo Chen**, Zhan Wang, Bing Chang, Wen Tao Zhu, and Jiwu Jing. MobiHydra: Pragmatic and Multi-Level Plausibly Deniable Encryption Storage for Mobile Devices. The 17th Information Security Conference (*ISC '14*), Hong Kong, China, Oct. 2014.



Paper Presentation

- On Implementing Deniable Storage Encryption for Mobile Devices

- Presented by Joe

On Implementing Deniable Storage Encryption for Mobile Devices

Paper by: Adam Skillen, Mohammad Mannan

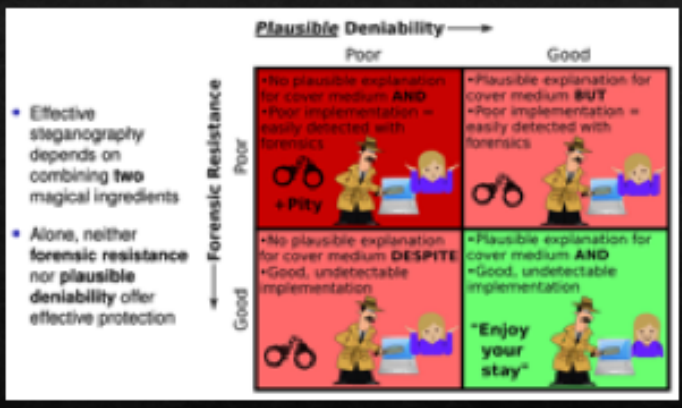
Presentation by: Joseph Muhle

Overview

- Goal and Motivation
- Threat Model and Assumptions
- Mobiflage Design
- Mobiflage Implementation
- Precautions against Colluding Carriers
- Sources of Compromise
- Security Analysis
- Performance Evaluation
- Thoughts

Goal and Motivation

- For some data, encryption is not enough
  - Users can be "coerced" to disclosing key
- Need a way to hide data completely
  - Hidden data appears **no different than random bits**
  - Adversary cannot prove existence of some plaintext data
- Need a scapegoat in emergencies
  - Provide decoy keys that produce **reasonable but innocuous** plaintexts



- These are the ideas behind Plausibly Deniable Encryption (PDE)
  - "Deniable encryption serves to undermine an attacker's confidence either that data is encrypted, or that the person in possession of it can decrypt it and provide the associated plaintext." - Wikipedia

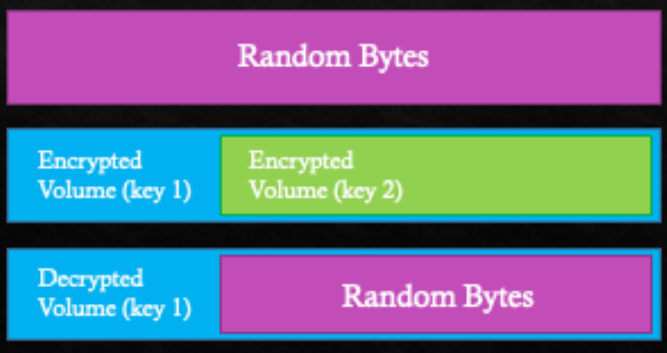
Threat Model and Assumptions

- This may be for
  - Human rights workers in areas of conflict
  - Journalists
  - Citizens under oppressive regime
- Phones must use default (or common) hardware/software
- Requires FAT32 SD card (physical or simulated)
- User's phone is free of malware/viruses, and OS/apps are trusted

- Adversary has
  - full knowledge of Mobiflage, but lacks key
  - some means of coercion
  - full access to phone (root access, internal/external storage/etc.)
  - No access to periodic snapshots of phone
  - Wireless Carrier and/or ISP
  - No direct access to phone's Plausibly Deniable Encryption (PDE) mode

Mobiflage Design

- Phone has 2 operating modes:
  - Standard – normal phone operation, accessed with decoy password
  - PDE – when secret data needs to be stored, accessed with secret password
- Hide volumes in empty space on external storage
  - Fill with random noise
  - Create 2 volumes: userdata (applications and settings), and auxiliary (documents, photos, etc.)
- Tangent – Steganography or Hidden Volumes?
  - Steganography – hidden message inside another innocuous message
    - Inefficient use of disk space, possible data loss, increased IO operations = no good
  - Hidden Volumes
    - No altered file system drivers required, IO is as efficient as standard encryption, Place files near end of storage which helps prevent overwriting data



Mobiflage Design (Contd.)

- Entire phone encrypted with decoy key (keys generated from passwords)
- Then, additional filesystems created at different offsets on disk and encrypted with different keys (in theory…)
  - **Hidden among random noise**
  - Offset based on password
    - Offset must be generated and not stored

$$E_{K1}(Vol_1)||E_{K2}(Vol_2)||...||E_{Kn}(Vol_n)$$

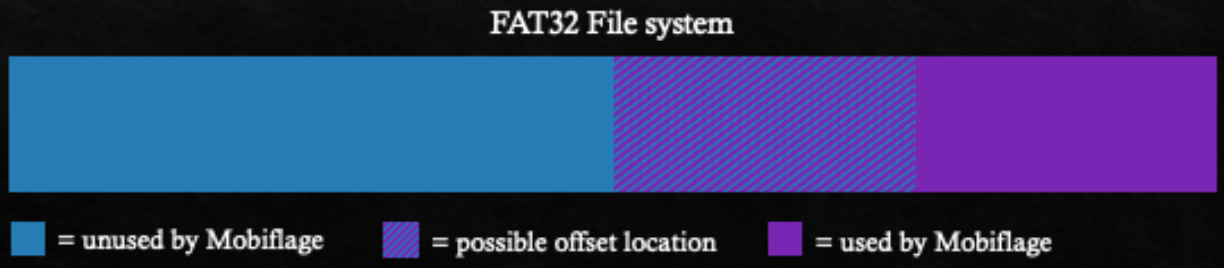
$$D_{K1}(Vol_1||Vol_2||...||Vol_n) = Vol_1||Rand$$

Mobiflage Design (Contd.)

vlen - # of 512-byte sectors on logical block storage device)

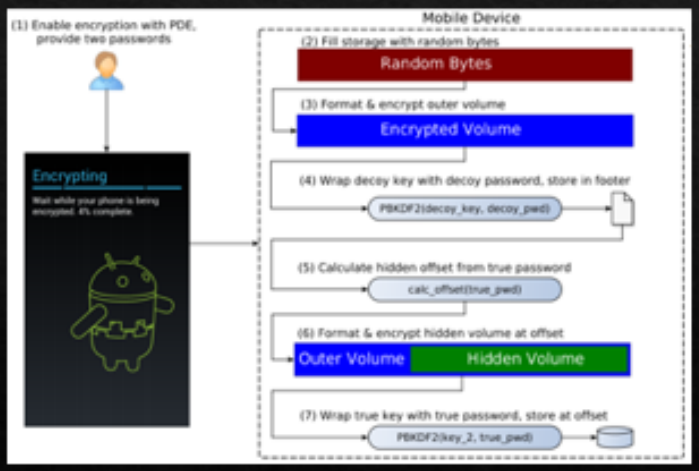
$$offset = 0.75 \times vlen - (\mathrm{H}(pwd||salt) \bmod (0.25 \times vlen))$$

- The generated offset is greater than one half and less than three quarters of the disk
- Hidden volume size is between 25-50% of total disk size
- **User should never fill standard-mode storage more than 50%**
- Note: User has no say in size of hidden volumes – once again, generated from password
  - Users however can request some size (more on this in a bit)

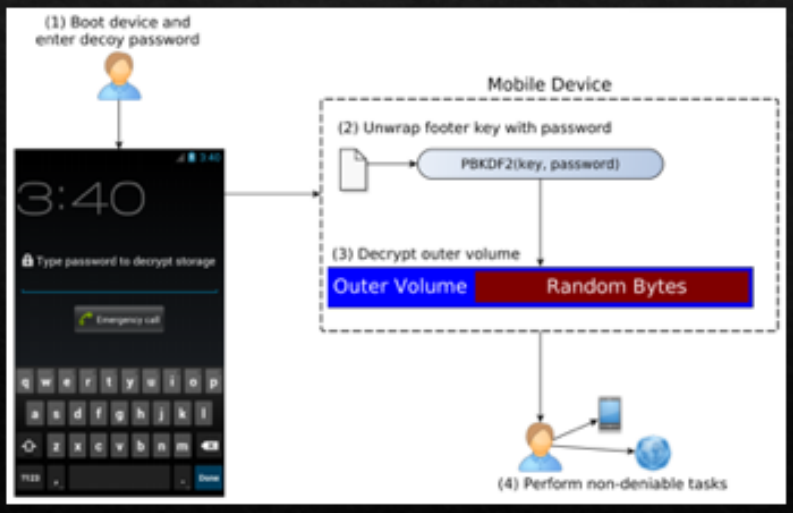


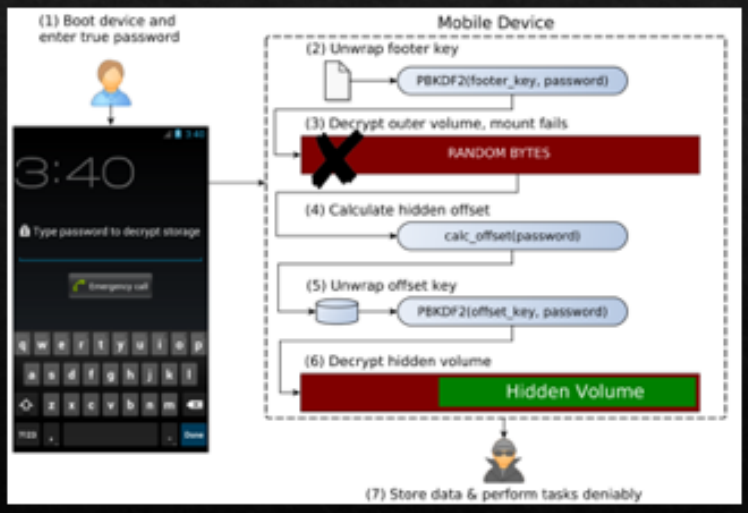
Mobiflage Design (Contd.)

- Usage of Mobiflage
  - User enables encryption PDE on phone
    - This erases data on external storage
  - User enters decoy and true passwords
  - Mobiflage creates hidden volumes and encrypts them
    - Initializing external storage with noise slows down process a bit
  - User enters decoy password for day-to-day usage in standard mode
    - Important to use standard mode – create a paper trail
  - User reboots and enters true password for PDE mode
    - Transfer documents, take photos, use hidden apps, etc.
      - System logs discarded from PDE mode
    - Precautions for PDE mode in a bit



Mobiflage Design (Contd.)





Mobiflage Implementation

- Not going to get super technical
- Hidden volumes should be on FAT32 file system
  - Ext4 is default for mobile devices internal storage, FAT32 default for external storage
    - Ext4 more complicated, harder to hide things, distributes things across whole disk rather than sequentially
- On startup, device will try to mount userdata volume
  - If no valid Ext4 file system found, user is prompted for password
  - System will attempt to mount volume with stored key
  - On failure, system will search for file system at offset derived from password
    - Success – mount userdata and external volumes
    - Failure – prompt to reenter password

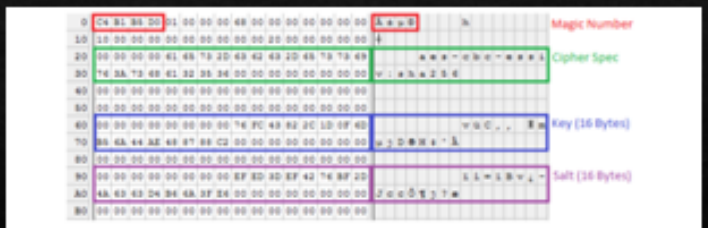


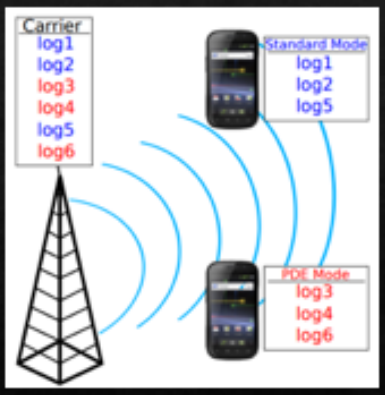
Figure 1: Android FDE footer (note that the cipher specification field is stored in clear text)

Mobiflage Implementation (Contd.) - Limitations

- External FAT32 storage partition required (SD of eMMC) OR internal FAT32 partition
- Hidden volume size random… kinda
  - Size is derived from user password (to avoid storing it)
  - Users can request a size, and system will iterate the hash until close to requested size
    - Storage of iterations not needed
    - system will try several hash offsets of password until filesystem is found (or max count is reached)
    - This can slow down boot process a bit
- At time of writing, only one hidden volume offset is supported
- No support for transferring files from standard mode to PDE mode
- Adversary can just wipe confiscate device
- Only 50% of SD can be safely used

Precautions against Colluding Carriers

- Carriers could be working with adversaries
- Your device is almost always connected to wireless carrier
- Use of PDE mode can cause discrepancies between phone log and carrier's logs.
  - Ex. Carrier sees a phone call from user device go through when it was in PDE mode
  - Adversary sees user phone shows no such log and concludes user is hiding something
- Best practices while in PDE mode
  - Device should be in airplane mode with SIM card removed
  - Anonymous SIM should be used, and phone info spoofed if connected to mobile network
  - Use public Wi-Fi if you can. IP should be spoofed, obscured (Tor), or VPN should be used
  - Web services (email, social networking) shouldn't be used unless under pseudonym reserved only for PDE mode
    - Ex. User logs onto email with normal account (imjohnsmith@hotmail.com)



Sources of Compromise

- Pseudo Random Number Generators – disk must appear as random data, stand up to statistical analysis
- Encryption Modes – Must be secure (duh)
- Flash Storage – must be used with wear-leveling mechanisms in place (finite reads/writes)
  - Storage is not linear, old fragments of encrypted data could be left behind
  - Adversary could use this to discover changes made to the disk outside of usual decoy volume

Security Analysis

- User passwords are used for offset encryption key → user should have good password
- Using a higher number of hash iterations helps slow down offline dictionary attacks
- When Android devices need more RAM, it writes background app state data to userdata partition (rather than writing RAM contents).
  - Not an issue since userdata partitions of modes are separated
- Log files are not an issue, logs from standard mode are preserved, PDE mode logs are deleted
- Caching is not an issue, in PDE mode we can use temporary filesystem cache
- In the threat model, we assume the adversary doesn't have access to past snapshots of the device
- Multiple hidden volumes may be beneficial to the user if the user cannot be held indefinitely
  - For indefinite holding, adversary may expect additional hidden volumes
  - Could have password to make hidden data inaccessible

Performance Evaluation

- Authors tested Mobiflage on Nexus S and Motorola Xoom
- Compared results of
  - No encryption
  - Android default encryption
  - PDE
- Standard definition audio, video, and mobile apps are not an issue
  - Blu-ray and very large apps could see performance issues

	Nexus S	**Moto Xoom**
Internal Storage	1 GB	32 GB
External Storage	15GB eMMC	8 GB SD

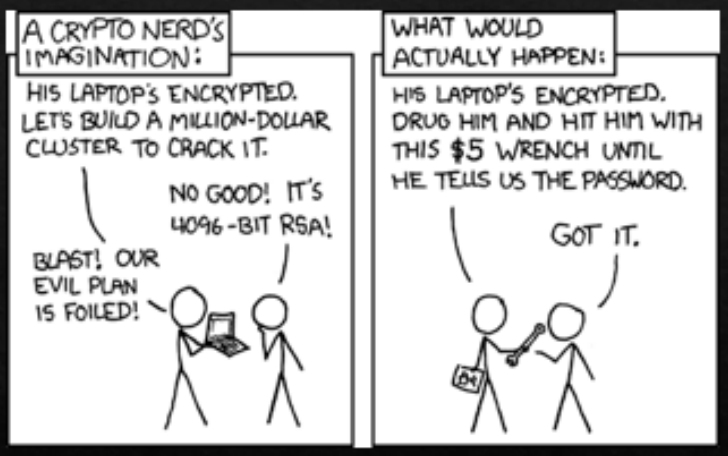
Cipher-spec	Key-length (bits)	Speed (KB/s)		Speed reduction	
		Nexus S	Xoom	Nexus S	Xoom
Unencrypted	N/A	5880±260	4767±238	-	-
AES-CBC-ESSIV (Android 4.x)	128	5559±76	4168±186	5.46%	12.57%
AES-XTS-Plain64 (Mobiflage)	512 (256+256)	5288±69	3929±146	10.07%	17.58%

	Nexus S	Moto Xoom
Std. Encryption	1 hr 5 min	1 hr 15 min
Mobiflage	< 2 hr	2 hr 28 min

Thoughts

- This paper is relatively old
  - 2013, using android 4.x (we are currently on android 11)
- Thorough in their methodology
  - Lots of technical details
- Threat model tips the scales to the adversary's benefit (better than being too lax)

Questions?



https://xkcd.com/538/

Sources

- https://www.ndss-symposium.org/wp-content_uploads_2017_09_Presentation06_3.pdf
- https://www.ndss-symposium.org/wp-content/uploads/2017/09/06_3_0.pdf