DISSECTING AND DETECTING MOBILE RANSOMWARE

Nicolò Andronio
Stefano Zanero, Federico Maggi

NESCTLab
Politecnico di Milano

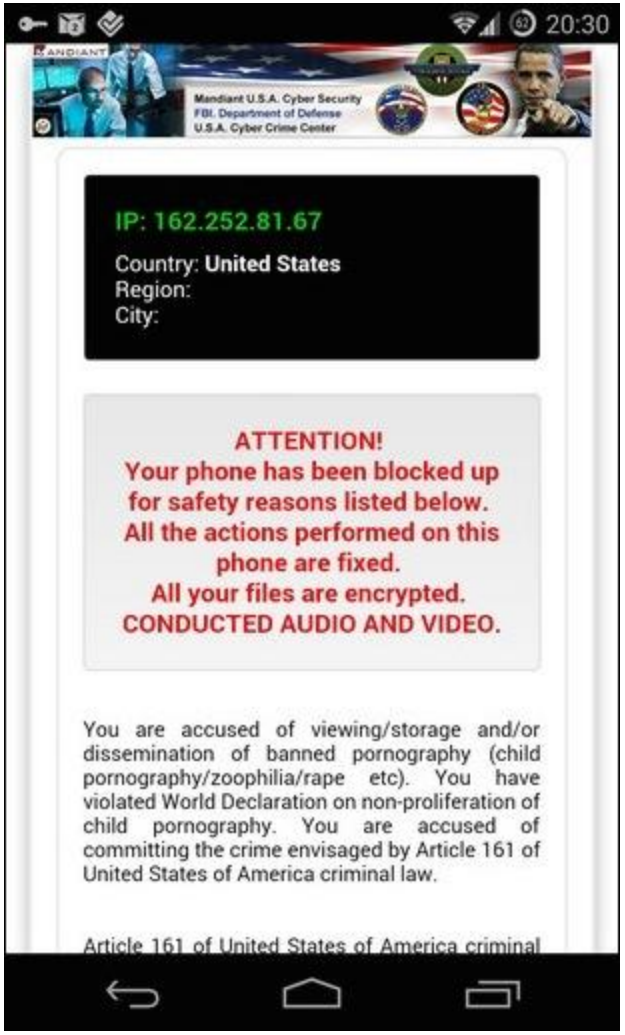
RANSOMWARE

- sub-class of malicious software
- typical behavior
  - threaten victims
    - social-engineering, embarrassing photos
  - encrypt data on device
  - makes the device unusable

- **goal:** extort money
  - the victims pay a "ransom"

- variant: scareware (no malicious actions)


FBI
FEDERAL BUREAU OF INVESTIGATION
FBI Criminal Investigation
#356440047053168
US
Prohibited content
This device is locked due to the violation of the federal laws of the United States of America:
• Article 161
• Article 148
• Article 215
• Article 301
• of the Criminal Code of U.S.A.
Your device was used to visit websites containing pornography.
Following violations were detected:


20:30
Mandiant U.S.A. Cyber Security
FBI. Department of Defense
U.S.A. Cyber Crime Center
IP: 162.252.81.67
Country: United States
Region:
City:
ATTENTION!
Your phone has been blocked up for safety reasons listed below.
All the actions performed on this phone are fixed.
All your files are encrypted.
CONDUCTED AUDIO AND VIDEO.
You are accused of viewing/storage and/or dissemination of banned pornography (child pornography/zoophilia/rape etc). You have violated World Declaration on non-proliferation of child pornography. You are accused of committing the crime envisaged by Article 161 of United States of America criminal law.
Article 161 of United States of America criminal

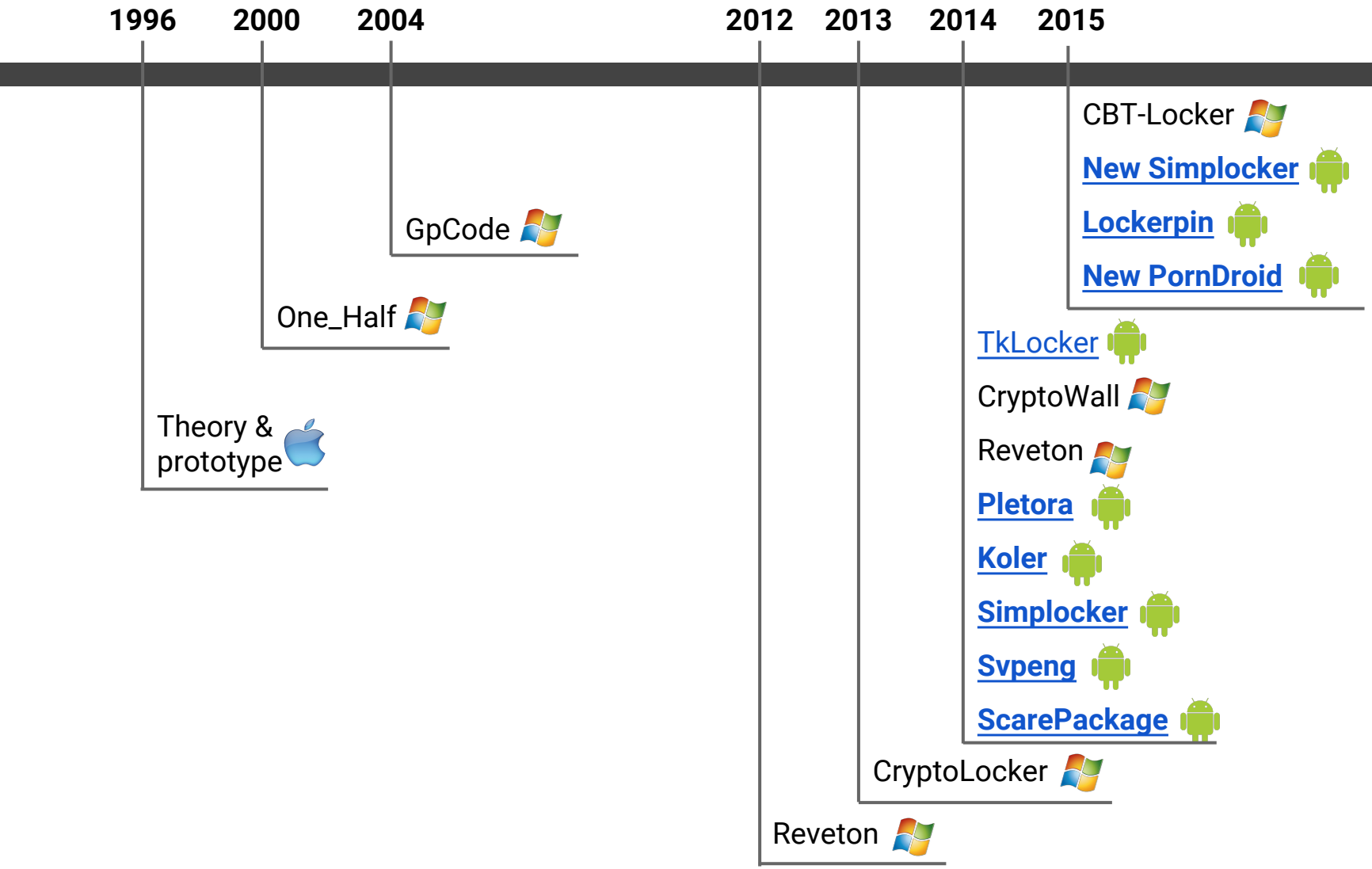

green dot MoneyPak
Amount of fine is $200.
You can settle the fine with MoneyPak express Packet vouchers.
As soon as the money arrives to the Treasure account, your device will be unblocked and all information will be decrypted in course of 24 hours.
We made a photo with your camera, it will be added to the investigation.
All your contacts are copied. If you do not pay the fine, we will notify your relatives and colleagues about the investigation.
Поддержка абонентов
88001007337
Поддержка абонентов
88001007337
Поддержка абонентов
88001007337

POPULARITY

- **generic** ransomware (desktop OSs)
  - 2,000,000+ **samples** (in 2 years by McAfee)
  - Nov 2014

- **mobile** ransomware (Android)
  - 900,000+ **victims** infected in 1 month
  - Aug 2014

TIMELINE AND ANDROID FAMILIES



RESEARCH GAP

- ample research on Android malware
- state-of-the-art tool (DREBIN) detected only 48.7% of the known ransomware
  - we asked the authors to perform the test
- commercial tools
  - lack of generality: well known limitation
- **bottom line:**
  - work reactively
  - need constant updates
  - easy to evade
  - cannot detect new variants

GOALS

- systematize the **state of the art** of **ransomware** families that target Android

- devise robust **indicators of compromise**
  - characteristics
    - feature based (as opposed to signature based)
    - accurate (almost zero false negatives)
    - adaptable (without code modifications)

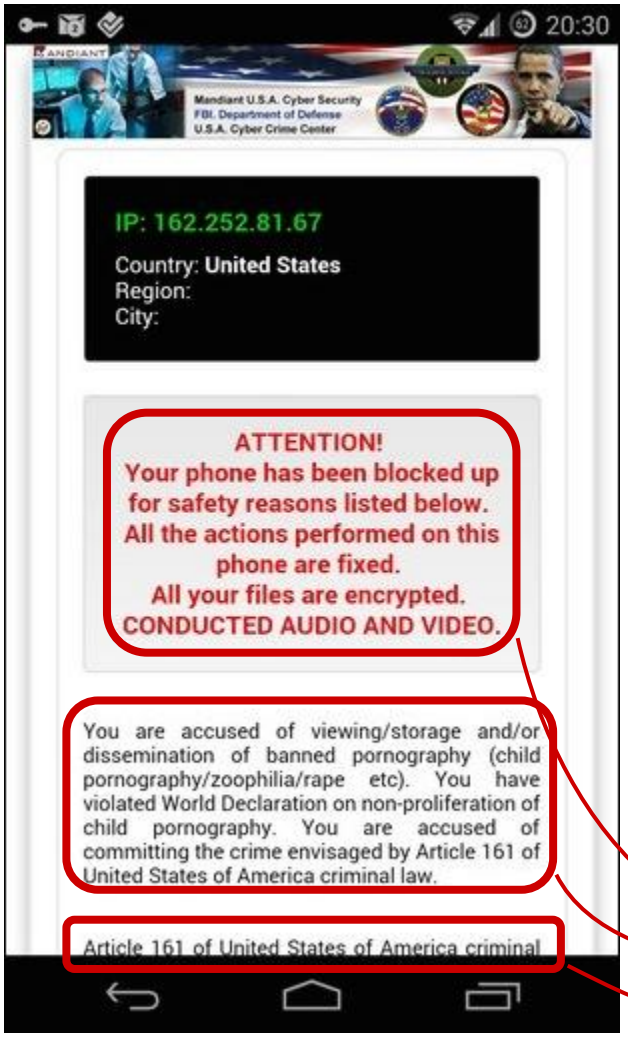
DETAILED ANALYSIS

- we reverse engineered a few samples for each family

COMMON CHARACTERISTICS

UNAVOIDABLE FOR ANY RANSOMWARE

THREATENING TEXT



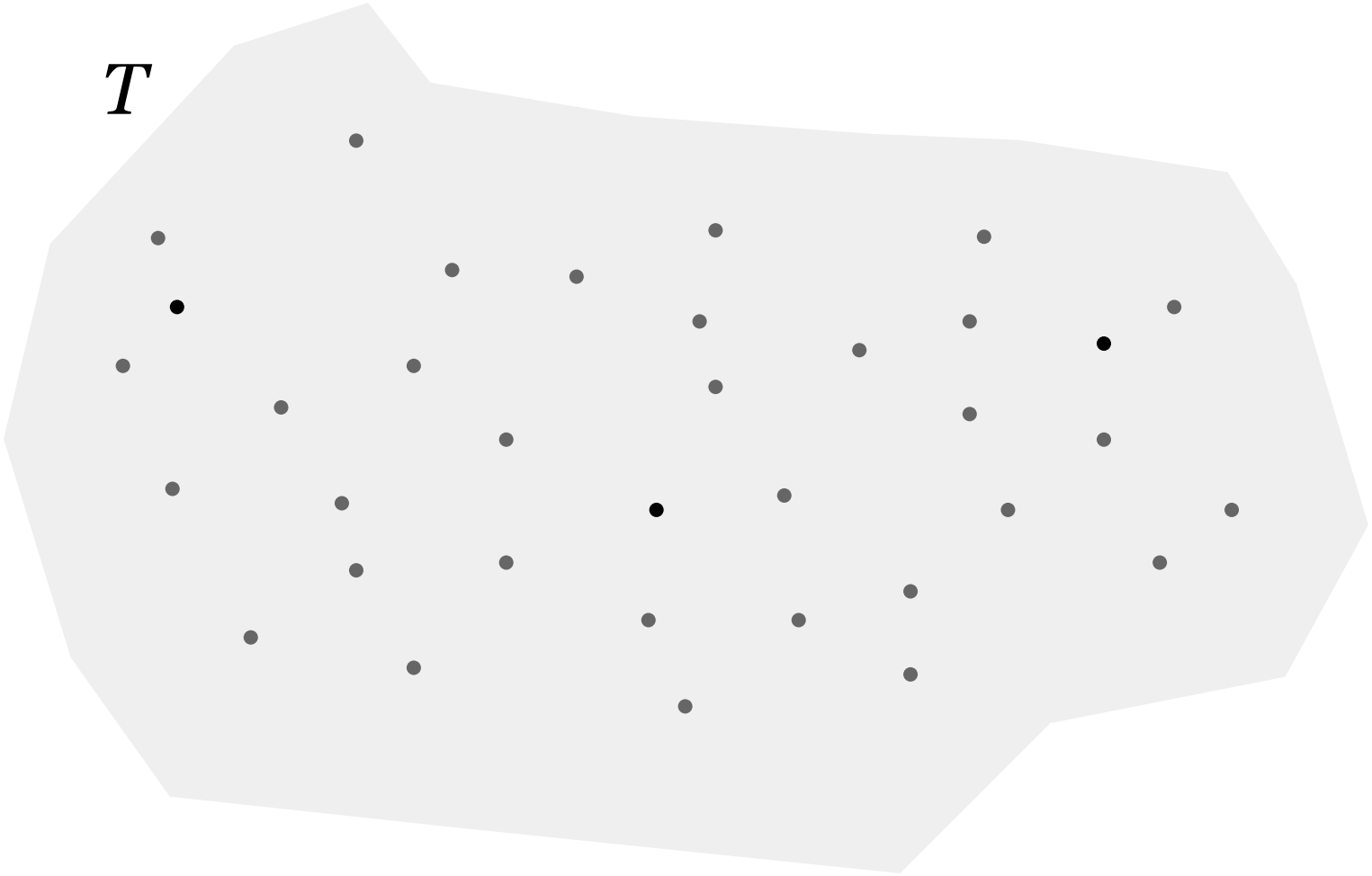
- **must be clear**, understandable and **convincing**
- **coercion** techniques
  - refer to **law codes**
  - various **accusations**
    - **copyright** violation
    - **illegal** content found
    - **prohibited** sites visited
- detailed **payment instructions**

text analysis & classification

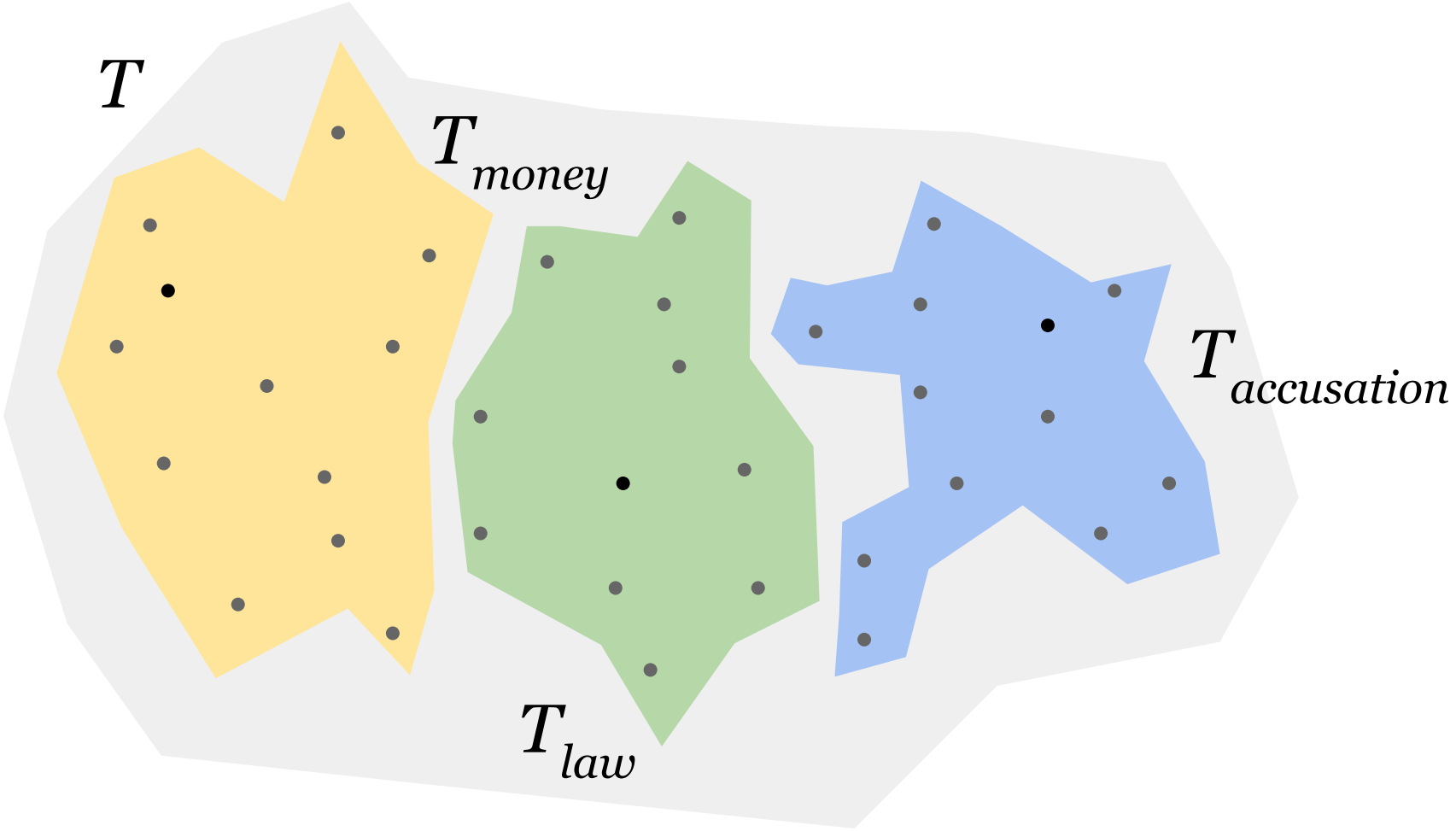
TEXT ANALYSIS: PREPARATION

1. **Language detection**
   - frequency-based analysis (e.g., English, French)
2. **Segmentation**
   - **"This device has been locked for safety reasons"**
   - **"All actions performed are fixed"**
3. **Stop-words removal**
   - **"~~This~~ device ~~has been~~ locked ~~for~~ safety reasons"**
   - **"~~All~~ actions performed ~~are~~ fixed"**
4. **Stemming**
   - **"~~This~~ device ~~has been~~ lock~~ed~~ ~~for~~ safe~~ty~~ reason~~s~~"**
   - **"~~All~~ action~~s~~ perform~~ed~~ ~~are~~ fix~~ed~~"**
5. **Stem vector**
   - presence/absence of each word in a binary vector

TRAINING



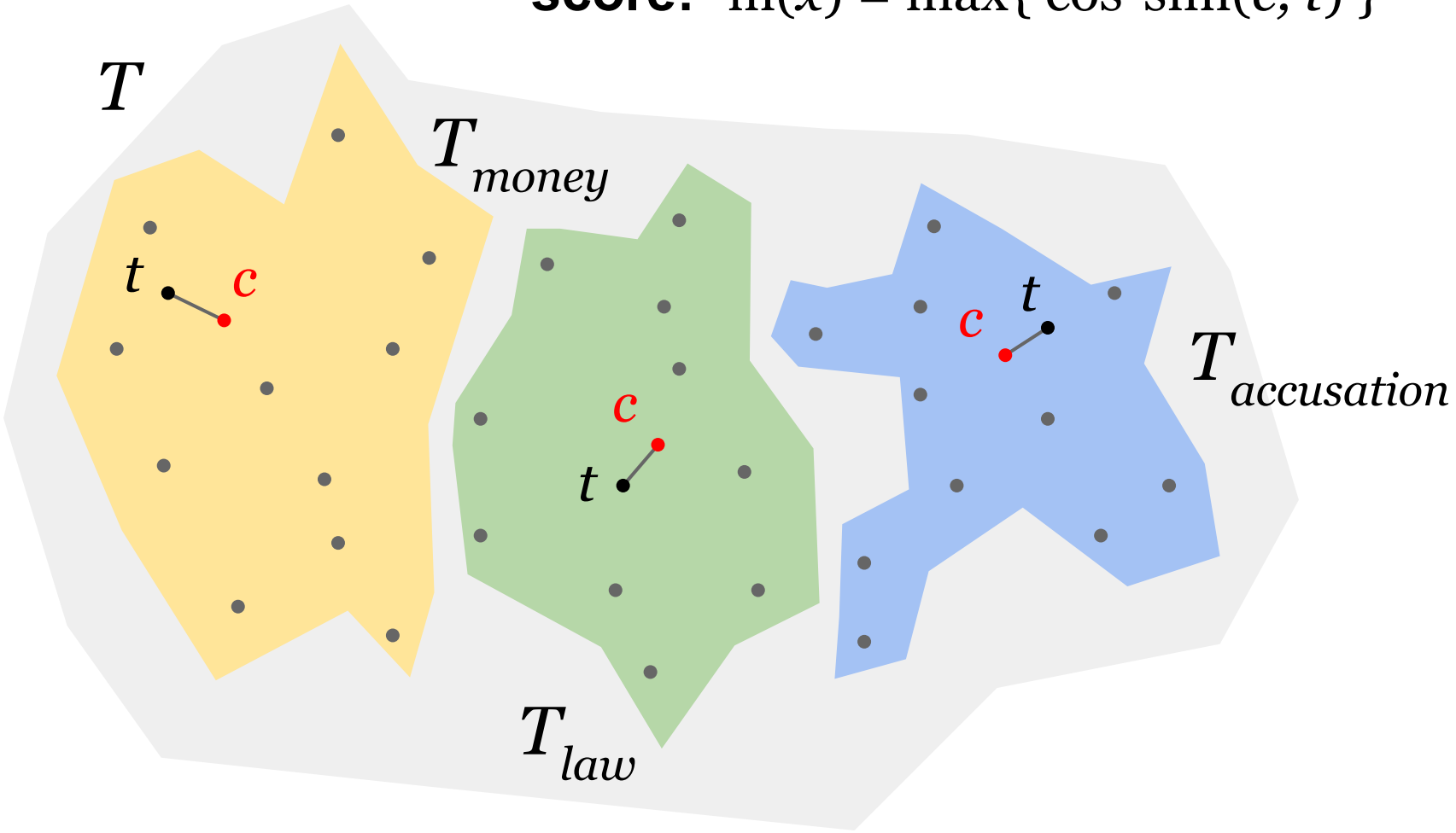
TRAINING



SCORING

text: $x = \{c_1, c_2, \ldots, c_n\}$

score: $\mathrm{m}(x) = \max\{\ \text{cos-sim}(c, t)\ \}$



decision thresholds: minimum to detect known ransomware

DECISION

if (*best score* in "money")

    could be **ransomware**

if (*best score* in "accusation" or "law")

    could be **scareware**

Note: adding new categories and building new decision criteria in the future would require only text samples.

LOCKING TECHNIQUES

- Request **device administration privileges** and use the `lockNow` API to lock the device
- **Immortal activity:**
  - **fill screen** with an activity
  - **inhibit navigation** with home/back keys
    - cover/hide the software-defined keys
    - intercept `onKeyDown`/`onKeyUp` and do nothing
- **Immortal dialog:**
  - create a dialog that cannot be closed using the `setCancelable(false)` API

EXAMPLE of LOCKING DETECTION


```
.method public onKeyDown(ILandroid/view/KeyEvent;)Z
  .locals 1

  # p1 = integer with the key code associated to the pressed key.

  const/4 v0, 0x4         # 4 = back button
  if-ne p1, v0, :cond_0
  iget-object v0, p0, Lcom/android/x5a807058/ZActivity;->q:Lcom/android/zics/ZModuleInterfac

  if-nez v0, :cond_0
  iget-object v0, p0, Lcom/android/x5a807058/ZActivity;->a:Lcom/android/x5a807058/ae;

  # we track function calls as well
  invoke-virtual {v0}, Lcom/android/x5a807058/ae;->c()Z
  :cond_0

  const/4 v0, 0x1     # True = event handled -> do not forward
  return v0
.end method
```

on "back" or "home" key pressed

return true -> event handled -> screen locked

Detection based on custom Smali emulation.

ENCRYPTION USAGE DETECTION

TYPICAL SEQUENCE

a. **read** from the filesystem (e.g., external SD card)
b. call some **encryption** API function
c. **write** to the filesystem (and **delete**)

Note: adding new flows is a configuration option.

EXAMPLE of ENCRYPTION DETECTION

```
invoke-static {},
  Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
move-result-object v0
invoke-virtual {v0}, Ljava/io/File;->toString()Ljava/lang/String;
move-result-object v0
new-instance v1, Ljava/io/File;
invoke-direct {v1, v0}, Ljava/io/File;-><init>(Ljava/lang/String;)V
```

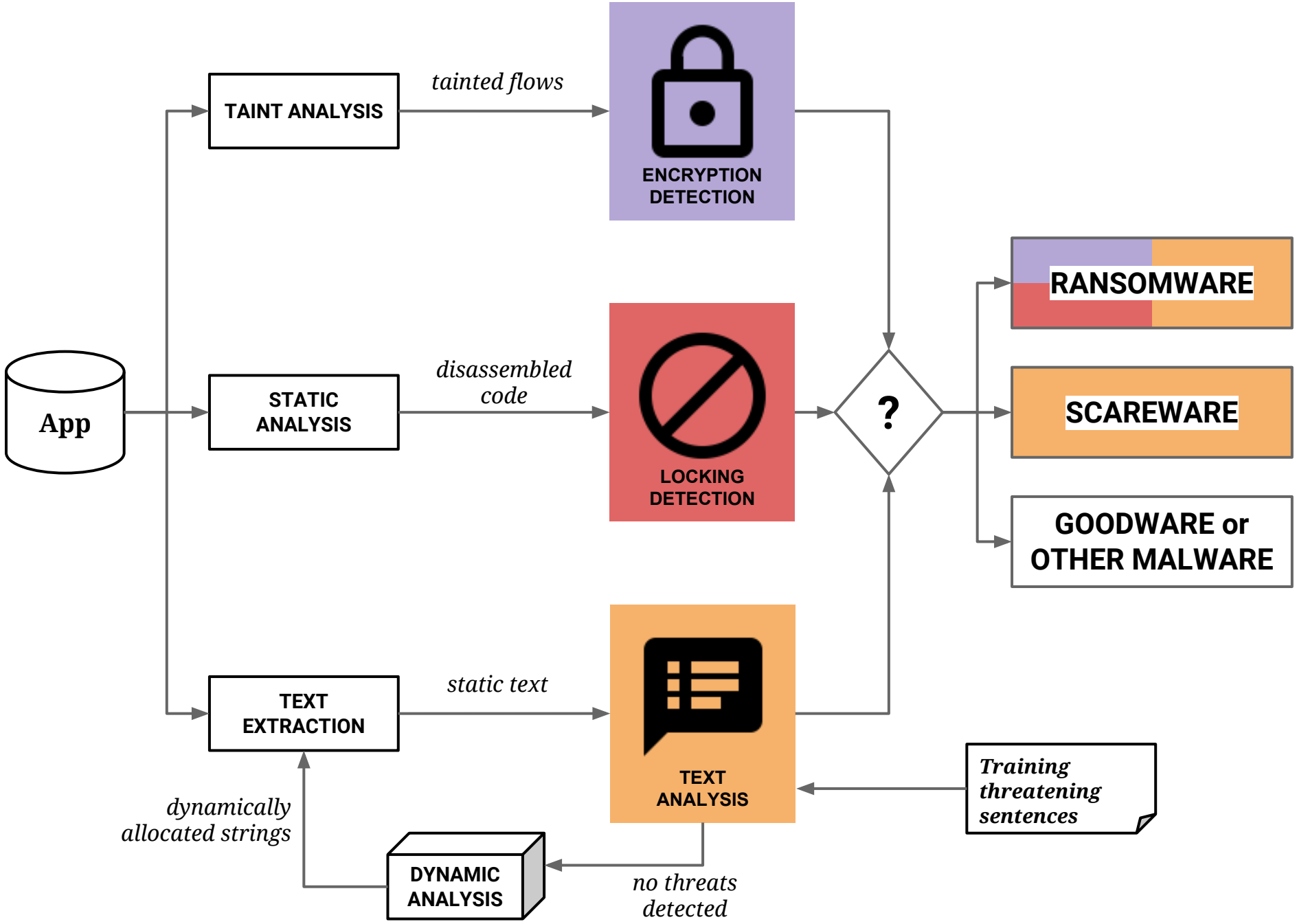
```
invoke-virtual {v2, v0, v4},
  Lcom/free/xxx/player/a;->a(Ljava/lang/String;Ljava/lang/String;)V

new-instance v4, Ljava/io/File;
invoke-direct {v4, v0}, Ljava/io/File;-><init>(Ljava/lang/String;)V
invoke-virtual {v4}, Ljava/io/File;->delete()Z
```

```
invoke-direct {v1, p2}, Ljava/io/FileOutputStream;-><init>(Ljava/lang/String;)V
iget-object v2, p0, Lcom/free/xxx/player/a;->a:Ljavax/crypto/Cipher;

const/4 v3, 0x1
iget-object v4, p0,
  Lcom/free/xxx/player/a;->b:Ljavax/crypto/spec/SecretKeySpec;

iget-object v5, p0,
  Lcom/free/xxx/player/a;->c:Ljava/security/spec/AlgorithmParameterSpec;
```


App
TAINT ANALYSIS
tainted flows
ENCRYPTION DETECTION
STATIC ANALYSIS
disassembled code
LOCKING DETECTION
?
RANSOMWARE
SCAREWARE
GOODWARE or OTHER MALWARE
TEXT EXTRACTION
static text
TEXT ANALYSIS
Training threatening sentences
dynamically allocated strings
DYNAMIC ANALYSIS
no threats detected

IMPLEMENTATION DETAILS

- language analysis:
  - OpenNLP
  - Stop-words Project
  - Snowball stemmer

- static flow analysis
  - FlowDroid

- dynamic analysis (based on TraceDroid):
  - only if no threatening text is found statically
  - OCR text (tesseract): tested its technical feasibility

EXPERIMENTAL VALIDATION

Implemented and made public through a REST API, on top of which we developed a client-side analyzer (see http://ransom.mobi).

SOURCE	SIZE	USE
AndRadar	172,174	**EXPERIMENT 1:** **Malware + Goodware (false positive eval.)**
AndroTotal.org	12,842	
Malware Genome	1,260	
Generic Malware	400	
Known ransomware	207	**Text-analysis training (manually vetted)**
Unseen ransomware	**443**	**EXPERIMENT 2:** **Detection evaluation**

EXPERIMENT 1: FALSE POSITIVES

- 9 (0.07%) out of 12,842 false positives

- 7 flagged as scareware
  - 6 benign apps
  - 1 adware app
  - manual analysis: large portions of law- or copyright-related text (e.g., terms of service)

- 2 flagged as ransomware
  - malicious, non-ransomware apps
    - locking behavior correctly detected

EXPERIMENT 2: DETECTION

- **internal validation**
  - can detect all known ransomware: OK, as expected

- **prediction (443 unseen ransomware)**
  - **375** correctly detected
  - **49** not detected: false negatives? not really:
    - mislabeled by VirusTotal!
  - **4** disarmed/non-working
  - **4** manually analyzed => no evidence found
  - **11** language unsupported (e.g., Spanish, Russian)

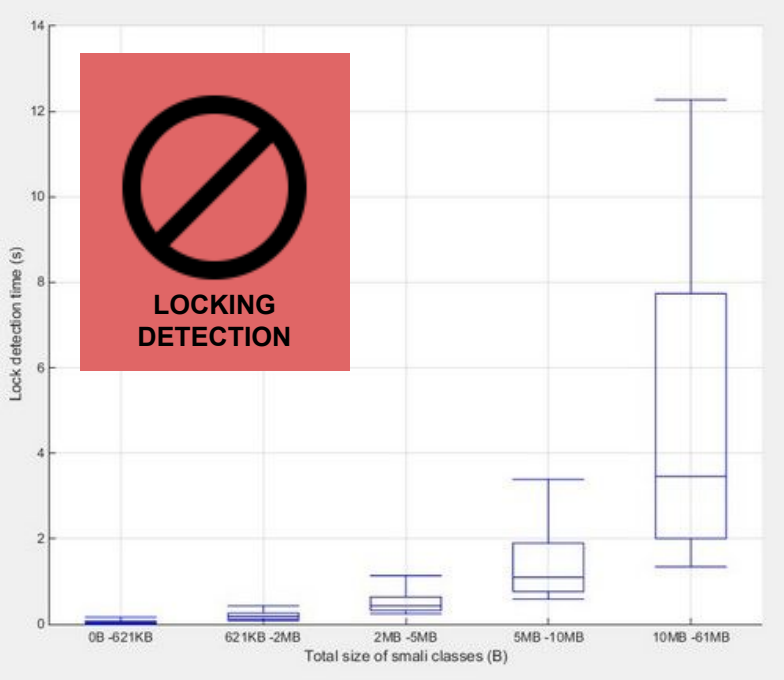
ADDING NEW LANGUAGES

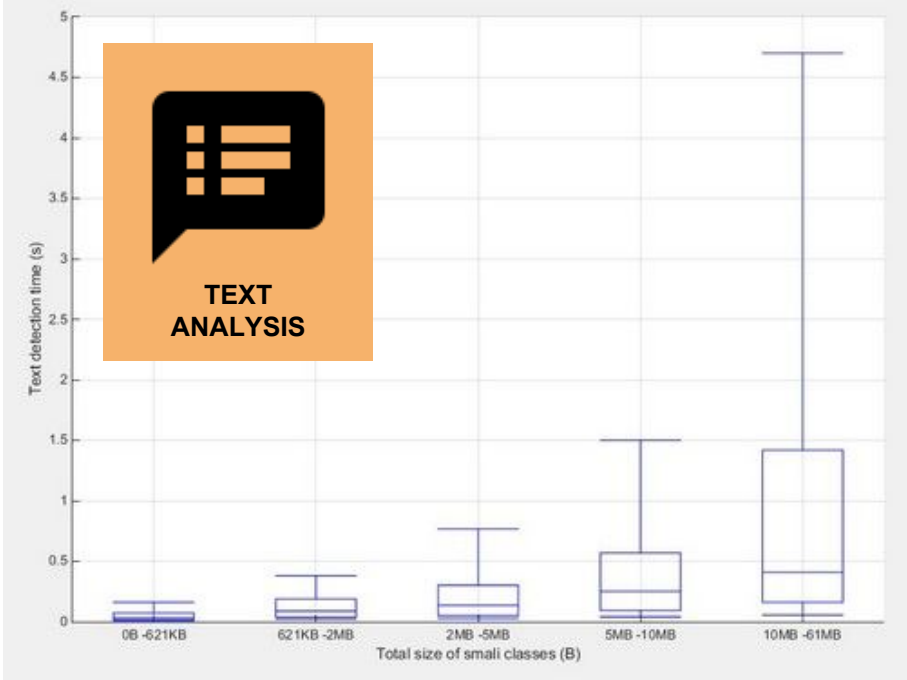
- localized ransomware campaigns
  - translated from English by the malware authors

- re-train our text classifier on translated text
  - increased prediction capabilities
  - **no localized sample needed!**
    - crucial for anticipating new campaigns

- burden: ~30 minutes of manual work per language

EXPERIMENT 3: SPEED





Encryption usage detection takes about 10^{-2} seconds.

LIMITATIONS (=> FUTURE WORK)

- **portability**
  - other than Android
  - "custom" encryption
- **evasion**
  - taint obfuscation (heavy use of reflection)
  - text analysis evasion: images, videos, out of band
- **internationalization**
  - non-Romance languages
    - Chinese
    - Japanese
    - Korean

CONCLUSIONS

- presented the **first systematic analysis** of Android ransomware characteristics

- proposed a first set of mobile-specific **indicators of compromise**

- released a **prototype** and client at http://ransom.mobi (please be gentle with API requests ;-)