Embedded System Security

What security property is needed?

What security property is needed?

- **Confidentiality**
  - component of privacy that implements to protect our data from unauthorized viewers

What security property is needed?

- **Confidentiality**
  - component of privacy that implements to protect our data from unauthorized viewers
- **Integrity**
  - data cannot be modified in an unauthorized or undetected manner

What security property is needed?

- **Confidentiality**
  - component of privacy that implements to protect our data from unauthorized viewers
- **Integrity**
  - data cannot be modified in an unauthorized or undetected manner
- **Availability**
  - preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks, such as a flood of incoming messages to the target system, essentially forcing it to shut down

What security property is needed?

- **Confidentiality**
  - component of privacy that implements to protect our data from unauthorized viewers
- **Integrity**
  - data cannot be modified in an unauthorized or undetected manner
- **Availability**
  - preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks, such as a flood of incoming messages to the target system, essentially forcing it to shut down
- **Non-repudiation**
  - one party of a transaction cannot deny having received a transaction, nor can the other party deny having sent a transaction

What security property is needed?

- **Confidentiality**
  - component of privacy that implements to protect our data from unauthorized viewers
- **Integrity**
  - data cannot be modified in an unauthorized or undetected manner
- **Availability**
  - preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks, such as a flood of incoming messages to the target system, essentially forcing it to shut down
- **Non-repudiation**
  - one party of a transaction cannot deny having received a transaction, nor can the other party deny having sent a transaction
- **Authenticity**
  - make sure that you really communicate with the partner you want to

What security property is needed?

- **Confidentiality**
  - component of privacy that implements to protect our data from unauthorized viewers
- **Integrity**
  - data cannot be modified in an unauthorized or undetected manner
- **Availability**
  - preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks, such as a flood of incoming messages to the target system, essentially forcing it to shut down
- **Non-repudiation**
  - one party of a transaction cannot deny having received a transaction, nor can the other party deny having sent a transaction
- **Authenticity**
  - make sure that you really communicate with the partner you want to
- Privacy related properties: **Anonymity, Unlinkability** ....

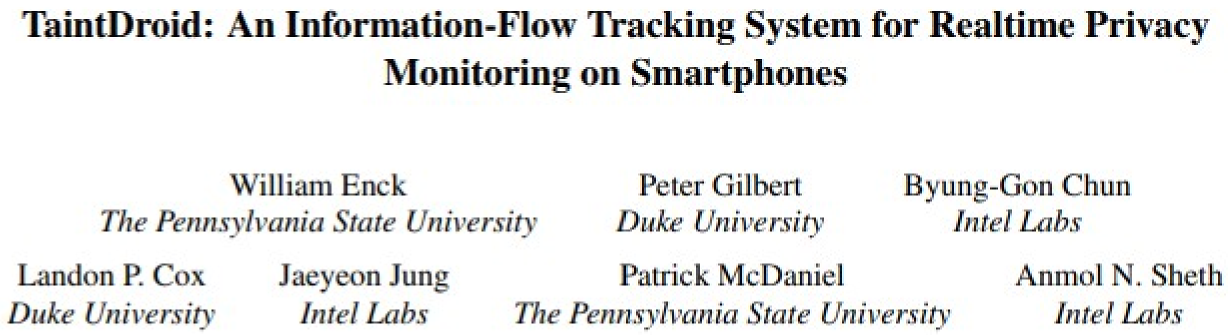
Threat Model or Trust Model

Threat Model or Trust Model

- **Does Alice trust Bob with whom she is communicating, but is wary of third parties like apps downloaded from playstore?**

Threat Model or Trust Model

- **Does Alice trust Bob with whom she is communicating, but is wary of third parties like apps downloaded from playstore?**



TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones

William Enck
The Pennsylvania State University

Peter Gilbert
Duke University

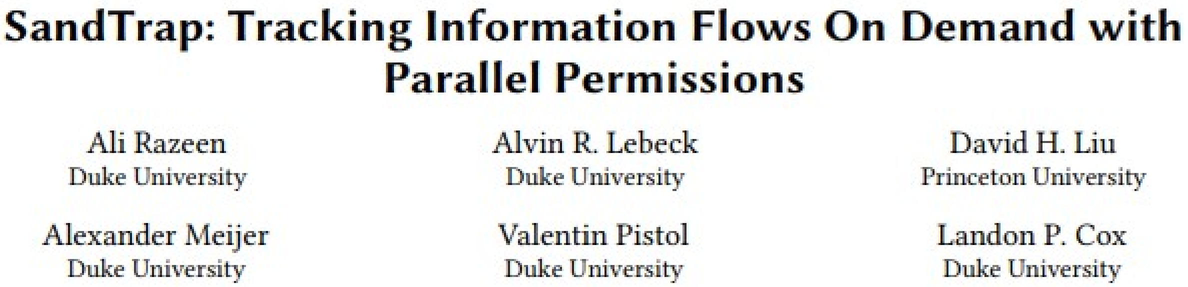
Byung-Gon Chun
Intel Labs

Landon P. Cox
Duke University

Jaeyeon Jung
Intel Labs

Patrick McDaniel
The Pennsylvania State University

Anmol N. Sheth
Intel Labs



SandTrap: Tracking Information Flows On Demand with Parallel Permissions

Ali Razeen
Duke University

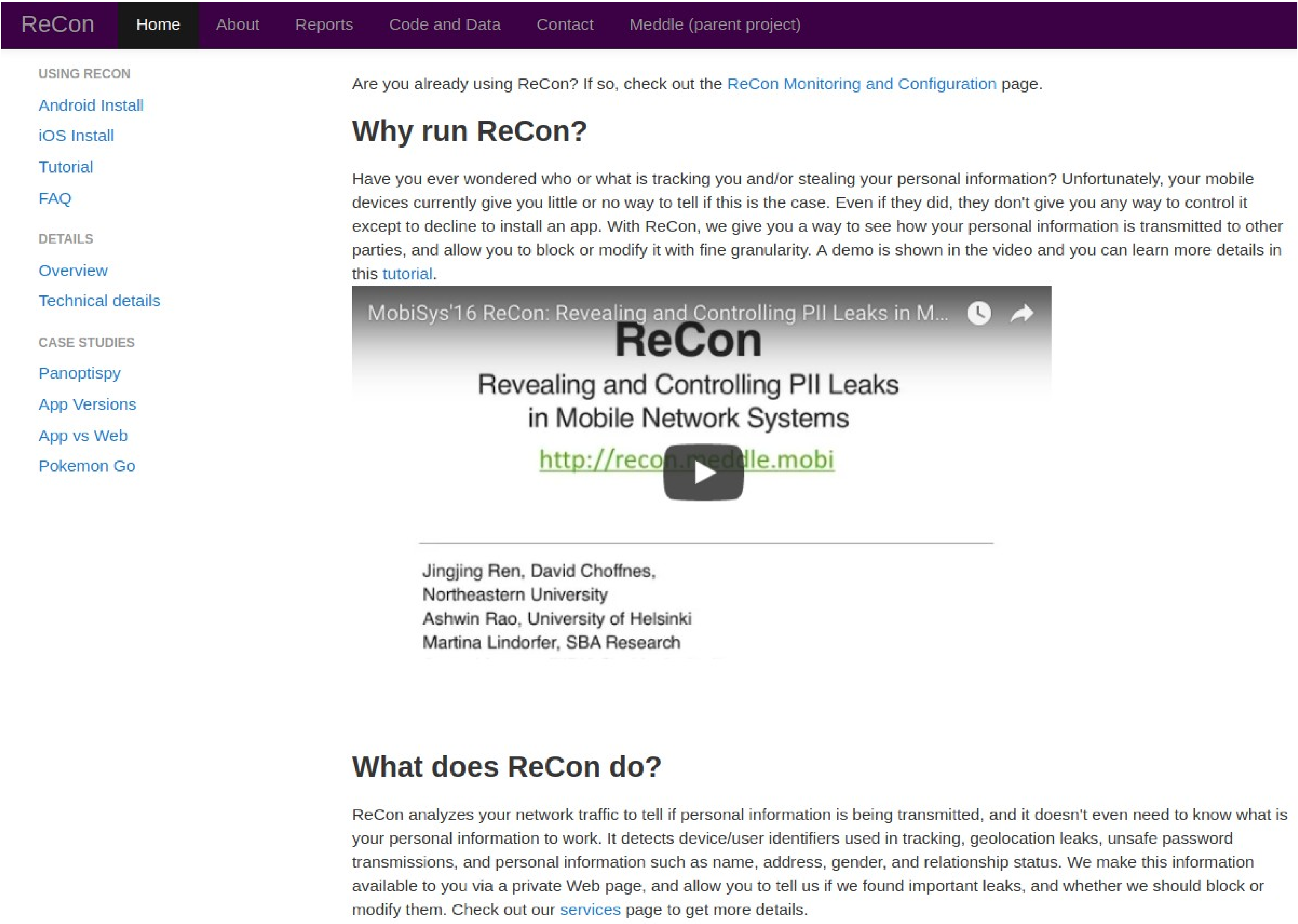
Alvin R. Lebeck
Duke University

David H. Liu
Princeton University

Alexander Meijer
Duke University

Valentin Pistol
Duke University

Landon P. Cox
Duke University



Are you already using ReCon? If so, check out the ReCon Monitoring and Configuration page.

Why run ReCon?

Have you ever wondered who or what is tracking you and/or stealing your personal information? Unfortunately, your mobile devices currently give you little or no way to tell if this is the case. Even if they did, they don't give you any way to control it except to decline to install an app. With ReCon, we give you a way to see how your personal information is transmitted to other parties, and allow you to block or modify it with fine granularity. A demo is shown in the video and you can learn more details in this tutorial.

MobiSys'16 ReCon: Revealing and Controlling PII Leaks in M...

ReCon

Revealing and Controlling PII Leaks
in Mobile Network Systems

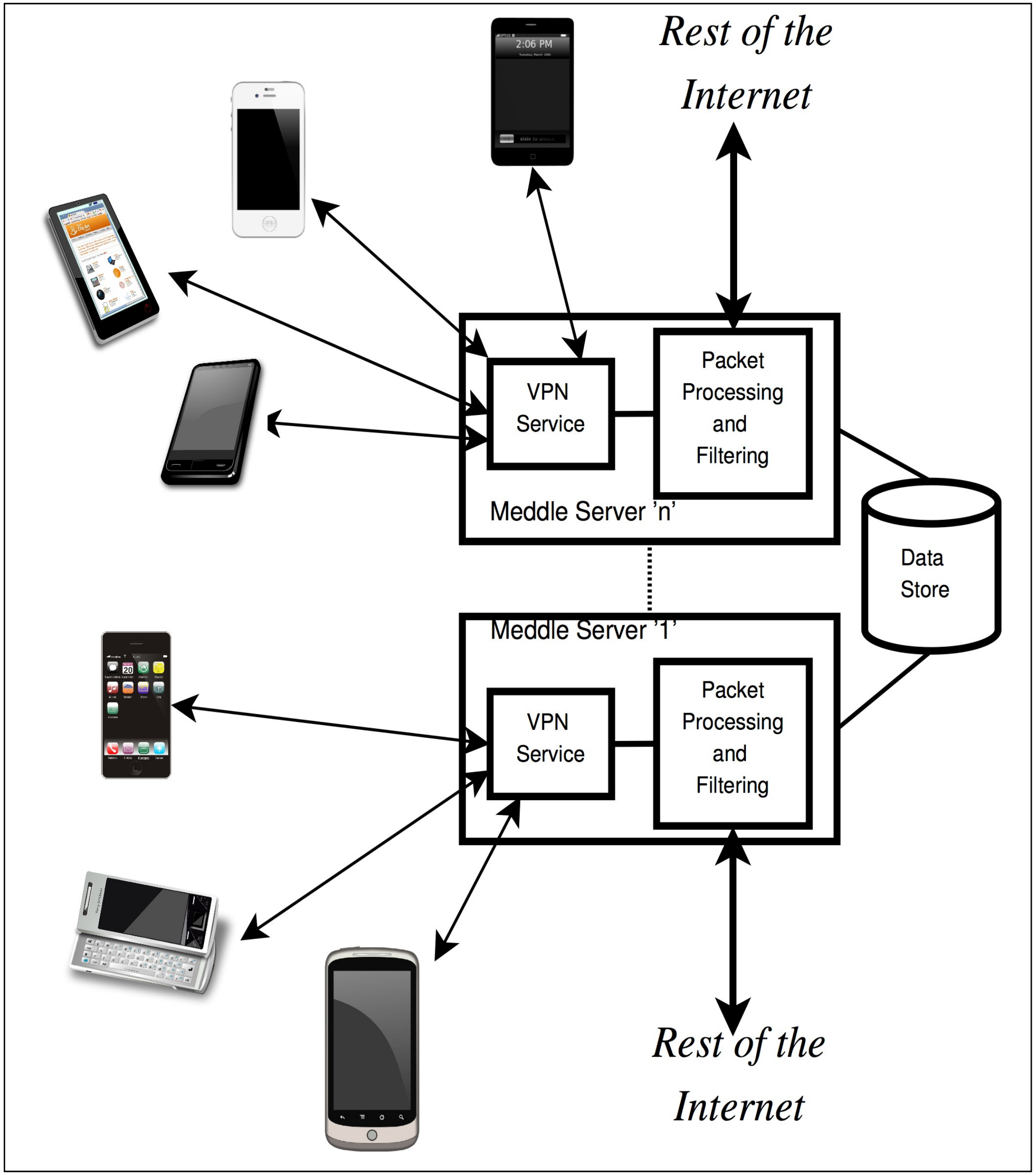
http://recon.meddle.mobi

Jingjing Ren, David Choffnes,
Northeastern University
Ashwin Rao, University of Helsinki
Martina Lindorfer, SBA Research

What does ReCon do?

ReCon analyzes your network traffic to tell if personal information is being transmitted, and it doesn't even need to know what is your personal information to work. It detects device/user identifiers used in tracking, geolocation leaks, unsafe password transmissions, and personal information such as name, address, gender, and relationship status. We make this information available to you via a private Web page, and allow you to tell us if we found important leaks, and whether we should block or modify them. Check out our services page to get more details.

Applied Machine Learning on Network Traffic



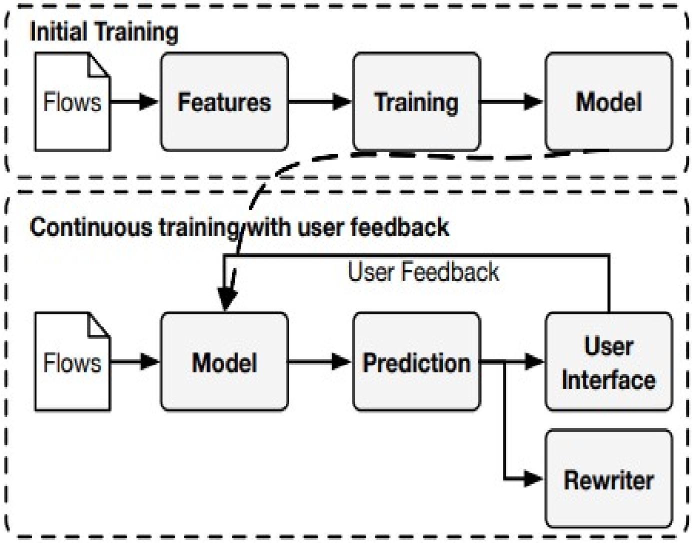
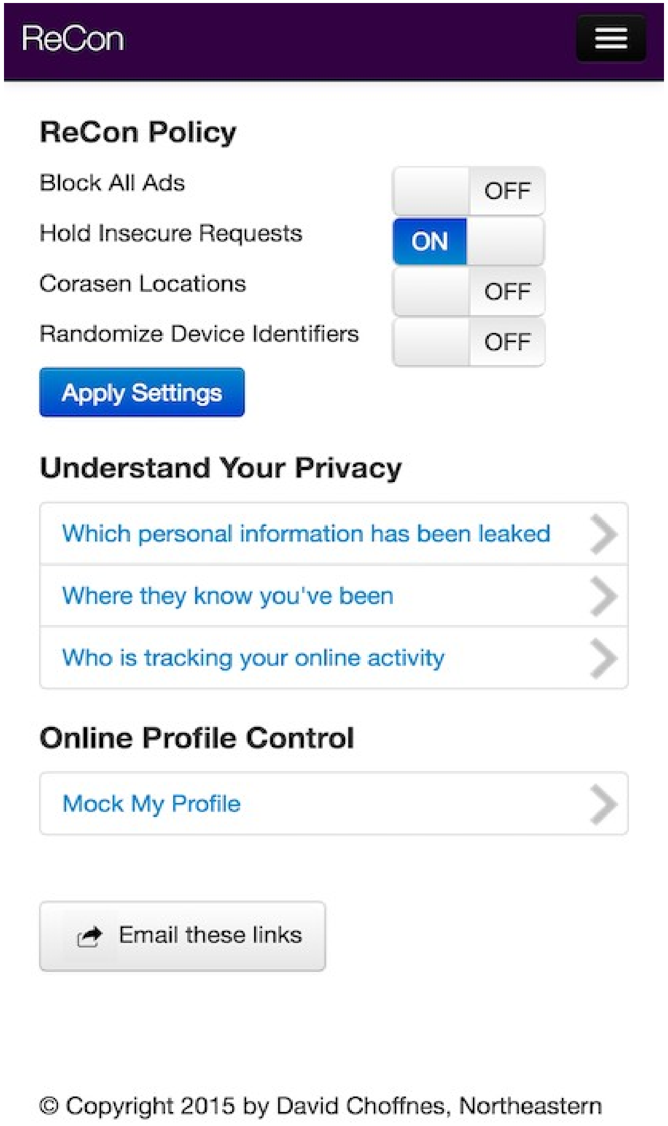


Figure 1: *ReCon* **architecture**. *We initially select features and train a model using labeled network flows (top), then use this model to predict whether new network flows are leaking PII. Based on user feedback, we retrain our classifier (bottom). Periodically, we update our classifier with results from new controlled experiments.*

Transparency



ReCon

ReCon Policy

Block All Ads OFF

Hold Insecure Requests ON

Corasen Locations OFF

Randomize Device Identifiers OFF

Apply Settings

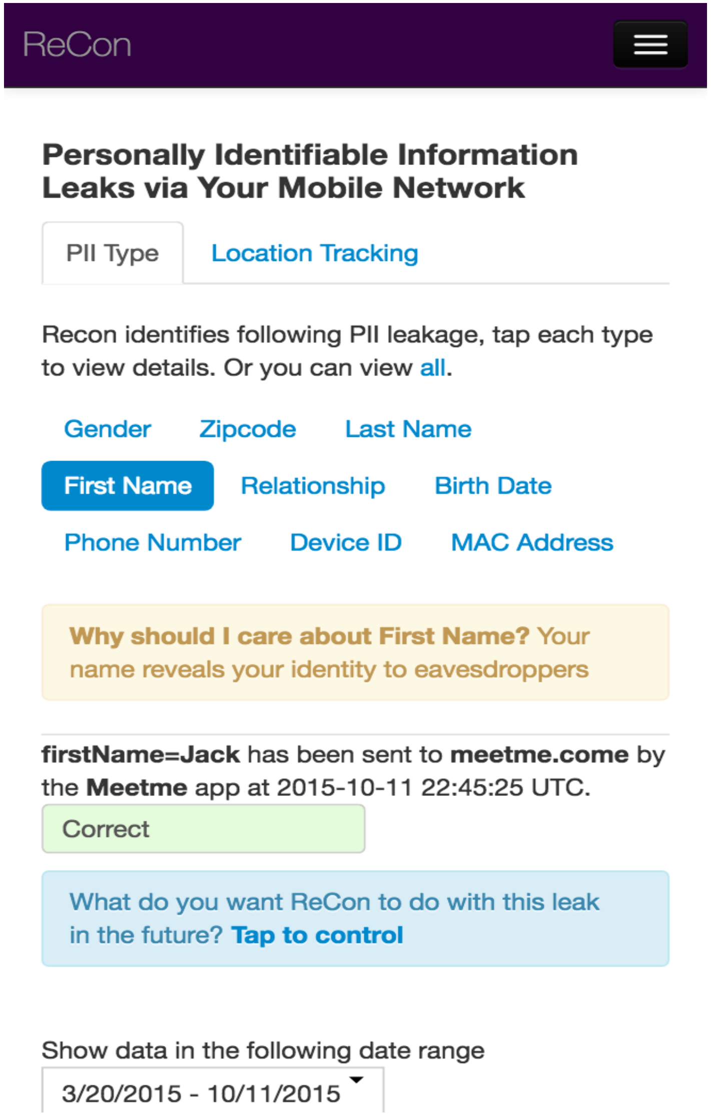
Understand Your Privacy

- Which personal information has been leaked
- Where they know you've been
- Who is tracking your online activity

Online Profile Control

- Mock My Profile

Email these links



ReCon

Personally Identifiable Information Leaks via Your Mobile Network

PII Type | Location Tracking

Recon identifies following PII leakage, tap each type to view details. Or you can view all.

Gender Zipcode Last Name

First Name Relationship Birth Date

Phone Number Device ID MAC Address

Why should I care about First Name? Your name reveals your identity to eavesdroppers

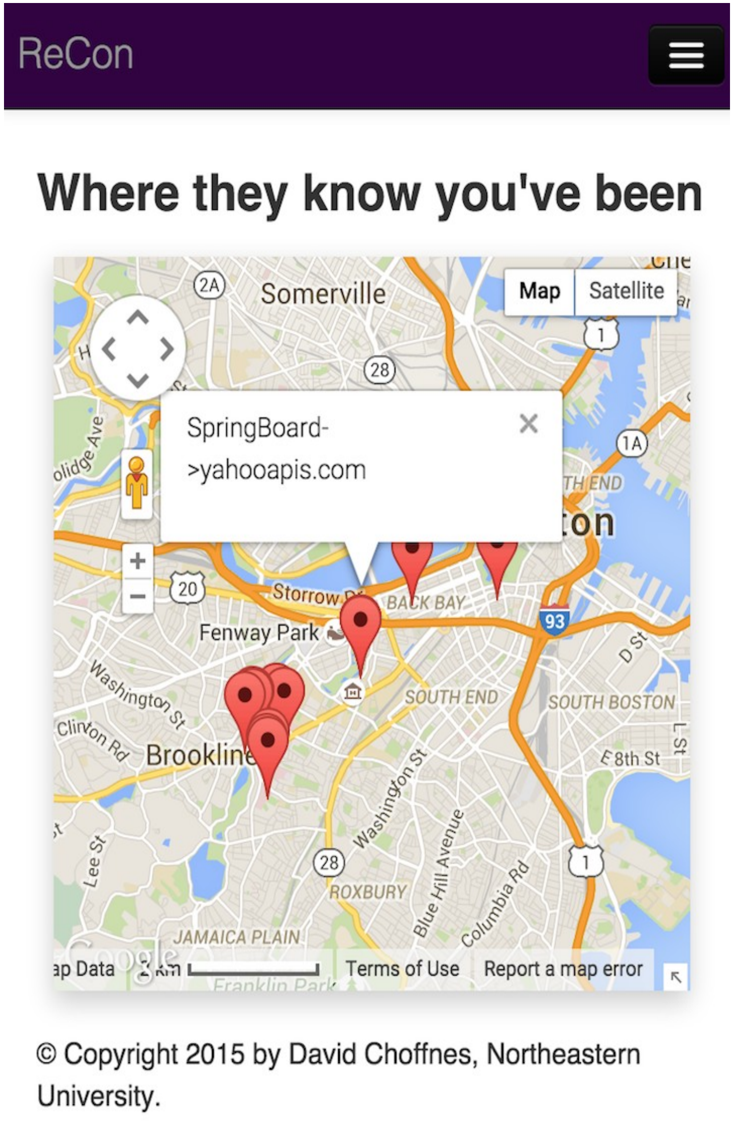
firstName=Jack has been sent to **meetme.come** by the **Meetme** app at 2015-10-11 22:45:25 UTC.

Correct

What do you want ReCon to do with this leak in the future? **Tap to control**

Show data in the following date range

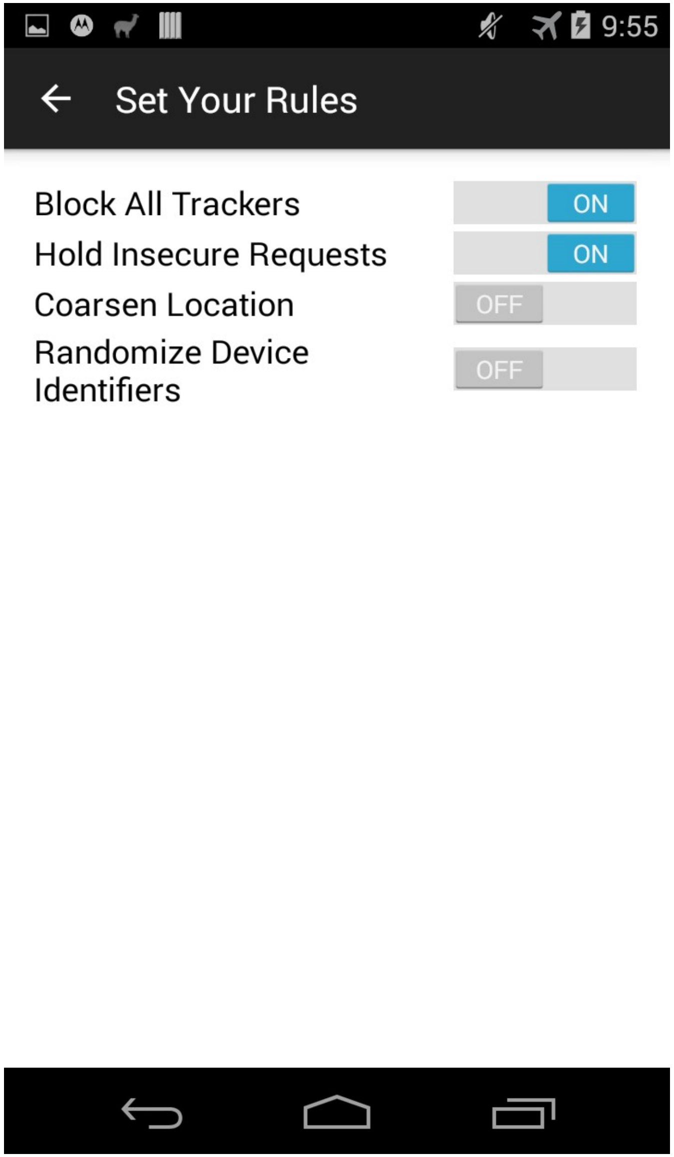
3/20/2015 - 10/11/2015



ReCon

Where they know you've been

And Control



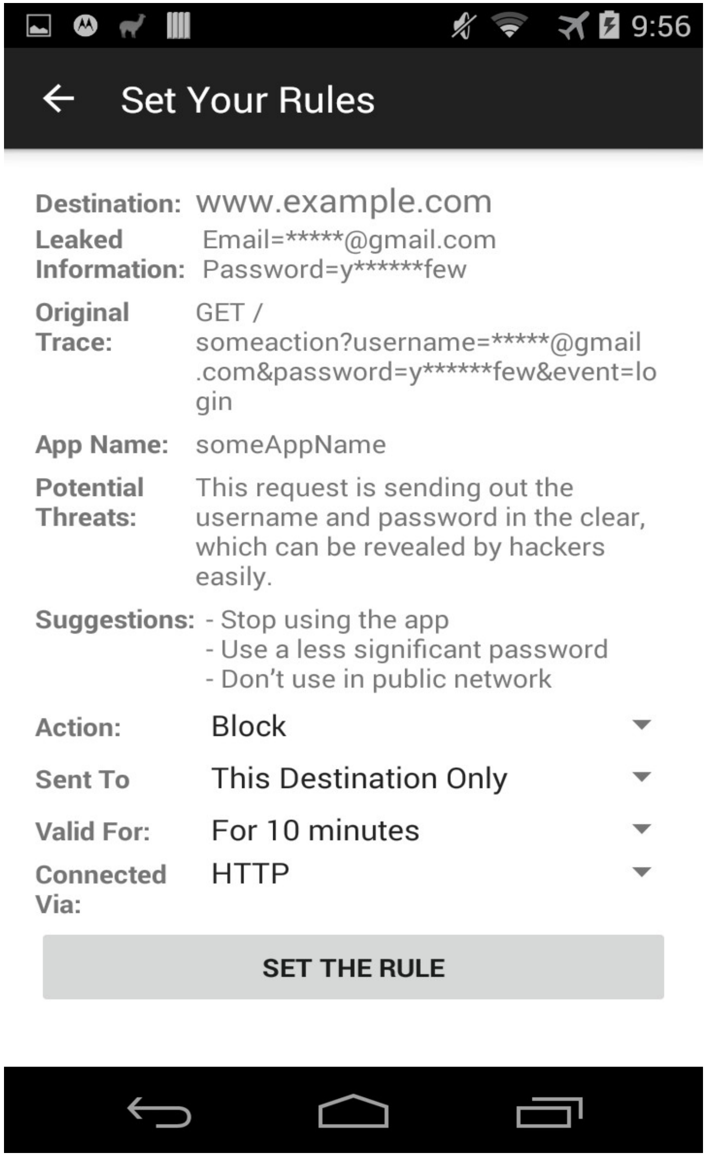
Set Your Rules

Block All Trackers: ON

Hold Insecure Requests: ON

Coarsen Location: OFF

Randomize Device Identifiers: OFF



Set Your Rules

Destination: www.example.com

Leaked Information: Email=*****@gmail.com
Password=y******few

Original Trace: GET / someaction?username=*****@gmail.com&password=y******few&event=login

App Name: someAppName

Potential Threats: This request is sending out the username and password in the clear, which can be revealed by hackers easily.

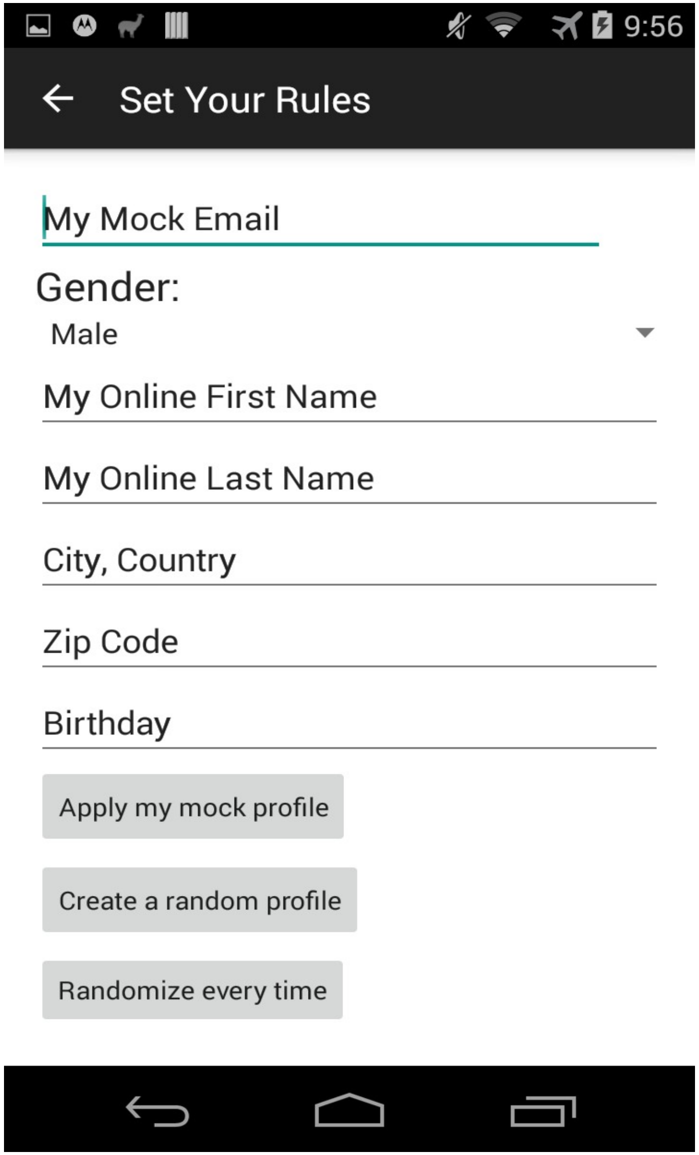
Suggestions:
- Stop using the app
- Use a less significant password
- Don't use in public network

Action: Block

Sent To This Destination Only

Valid For: For 10 minutes

Connected Via: HTTP



Set Your Rules

My Mock Email

Gender:
Male

My Online First Name

My Online Last Name

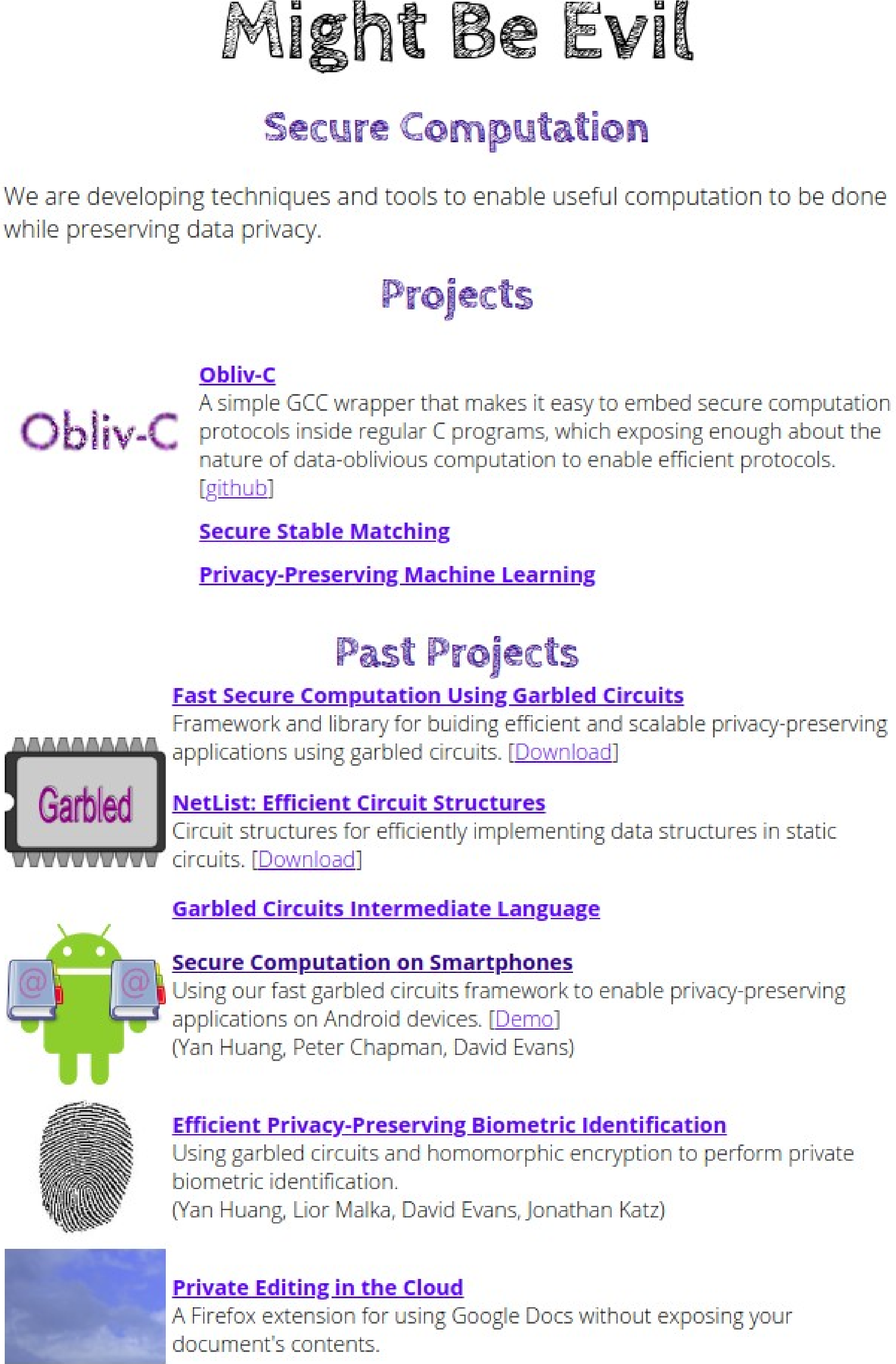
City, Country

Zip Code

Birthday

Threat Model or Trust Model

- Does Alice trust Bob with whom she is communicating, but is wary of third parties like apps downloaded from playstore?
- **Does Alice mistrust Bob, but still needs to compute some joint function with him for certain applications?**



Might Be Evil

Secure Computation

We are developing techniques and tools to enable useful computation to be done while preserving data privacy.

Projects

Obliv-C
A simple GCC wrapper that makes it easy to embed secure computation protocols inside regular C programs, which exposing enough about the nature of data-oblivious computation to enable efficient protocols. [github]

Secure Stable Matching

Privacy-Preserving Machine Learning

Past Projects

Fast Secure Computation Using Garbled Circuits
Framework and library for buiding efficient and scalable privacy-preserving applications using garbled circuits. [Download]

NetList: Efficient Circuit Structures
Circuit structures for efficiently implementing data structures in static circuits. [Download]

Garbled Circuits Intermediate Language

Secure Computation on Smartphones
Using our fast garbled circuits framework to enable privacy-preserving applications on Android devices. [Demo]
(Yan Huang, Peter Chapman, David Evans)

Efficient Privacy-Preserving Biometric Identification
Using garbled circuits and homomorphic encryption to perform private biometric identification.
(Yan Huang, Lior Malka, David Evans, Jonathan Katz)

Private Editing in the Cloud
A Firefox extension for using Google Docs without exposing your document's contents.

Threat Model or Trust Model

- Does Alice trust Bob with whom she is communicating, but is wary of third parties like apps downloaded from playstore?
- Does Alice mistrust Bob, but still needs to compute some joint function with him for certain applications?
- **Does Alice mistrust her own operating system, interpreters, system libraries, third party libraries?**

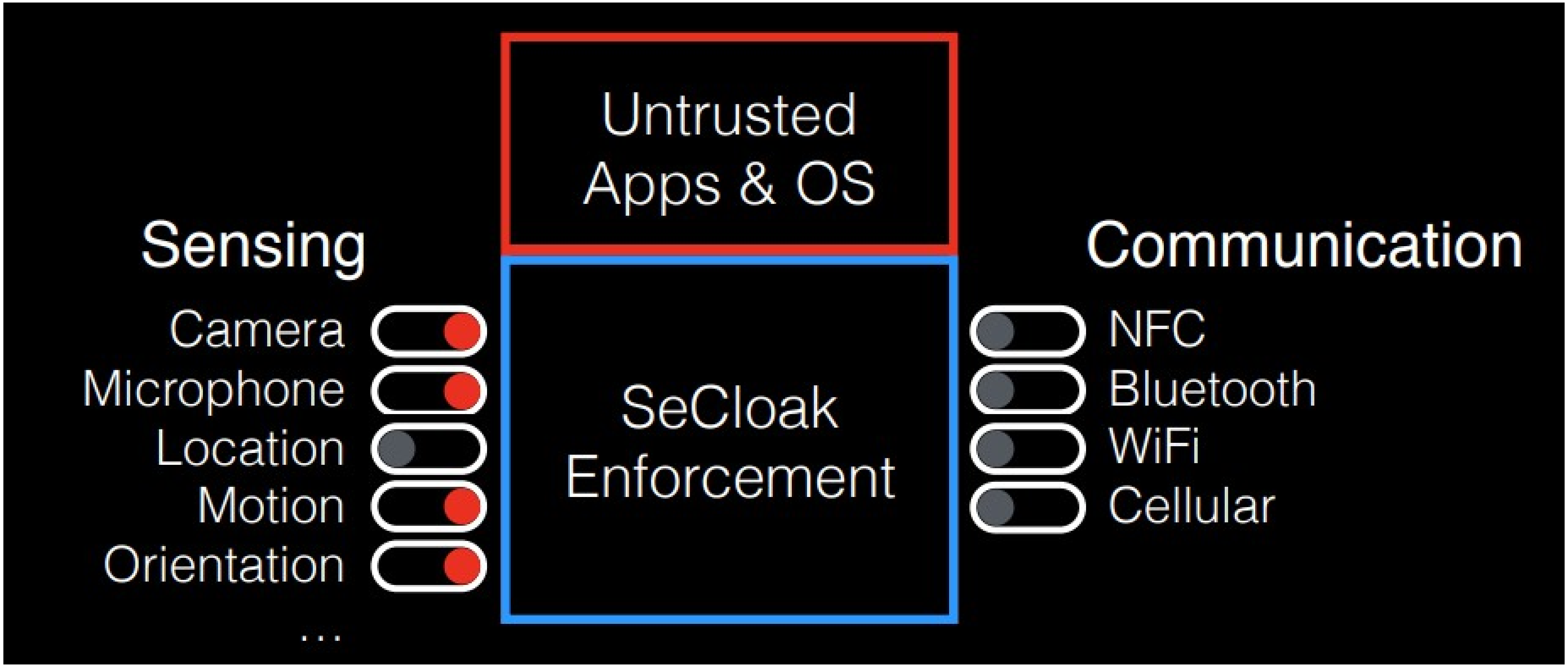
SeCloak: ARM Trustzone-based Mobile Peripheral Control

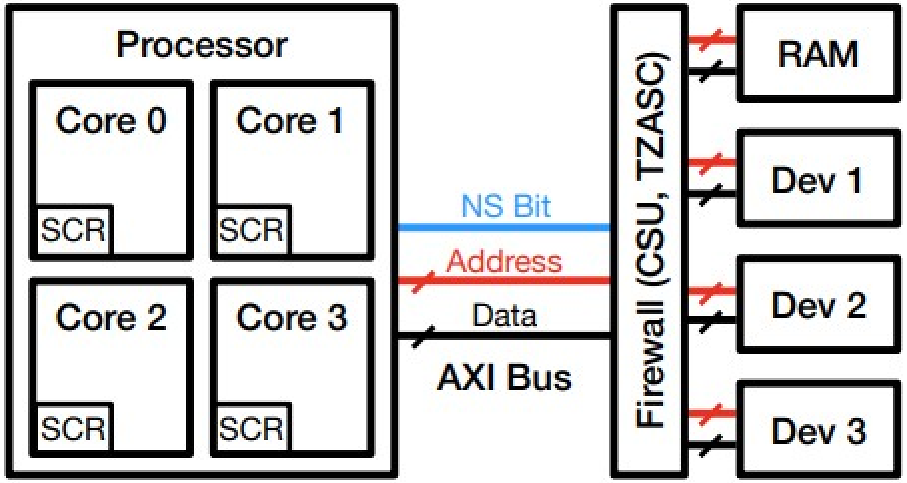
Matthew Lentz
University of Maryland
mlentz@cs.umd.edu

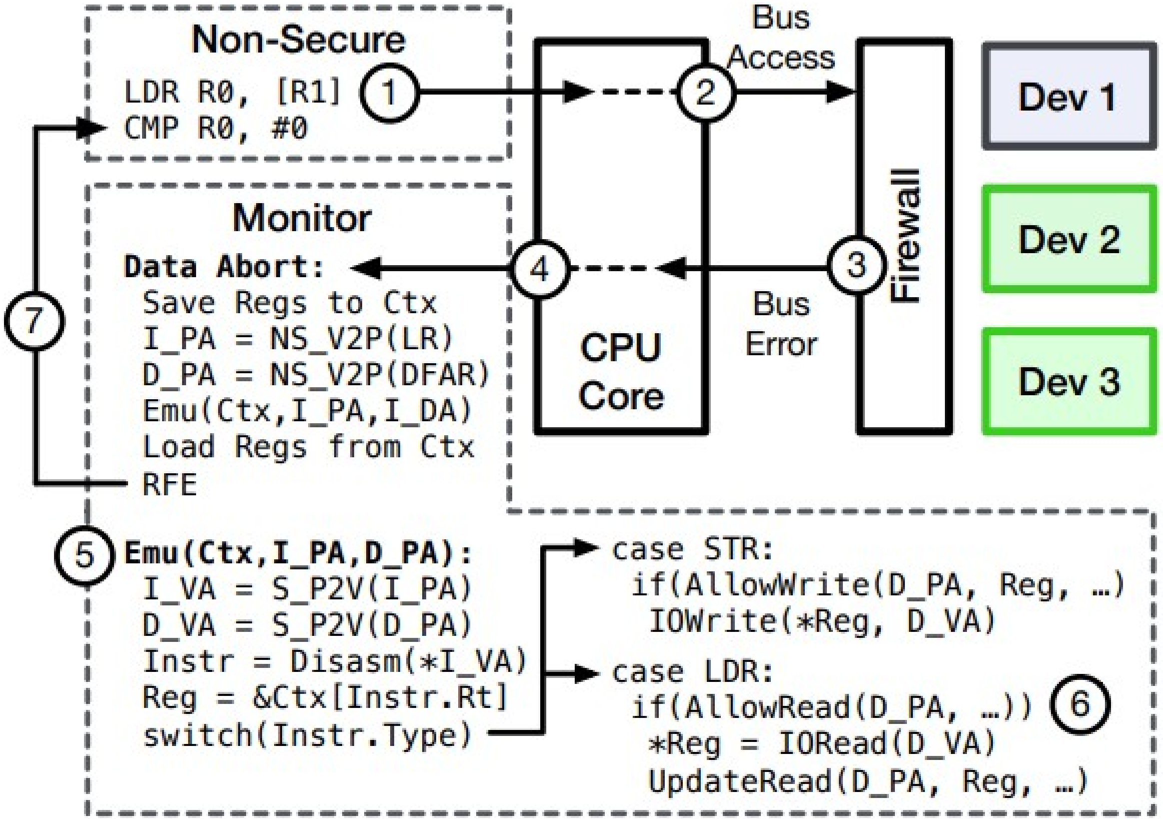
Rijurekha Sen
Max Planck Institute for Software Systems
rijurekha@mpi-sws.org

Peter Druschel
Max Planck Institute for Software Systems
druschel@mpi-sws.org

Bobby Bhattacharjee
University of Maryland
bobby@cs.umd.edu







```
LDR R0, [R1]
CMP R0, #0
```

```
Data Abort:
 Save Regs to Ctx
 I_PA = NS_V2P(LR)
 D_PA = NS_V2P(DFAR)
 Emu(Ctx,I_PA,I_DA)
 Load Regs from Ctx
 RFE
```

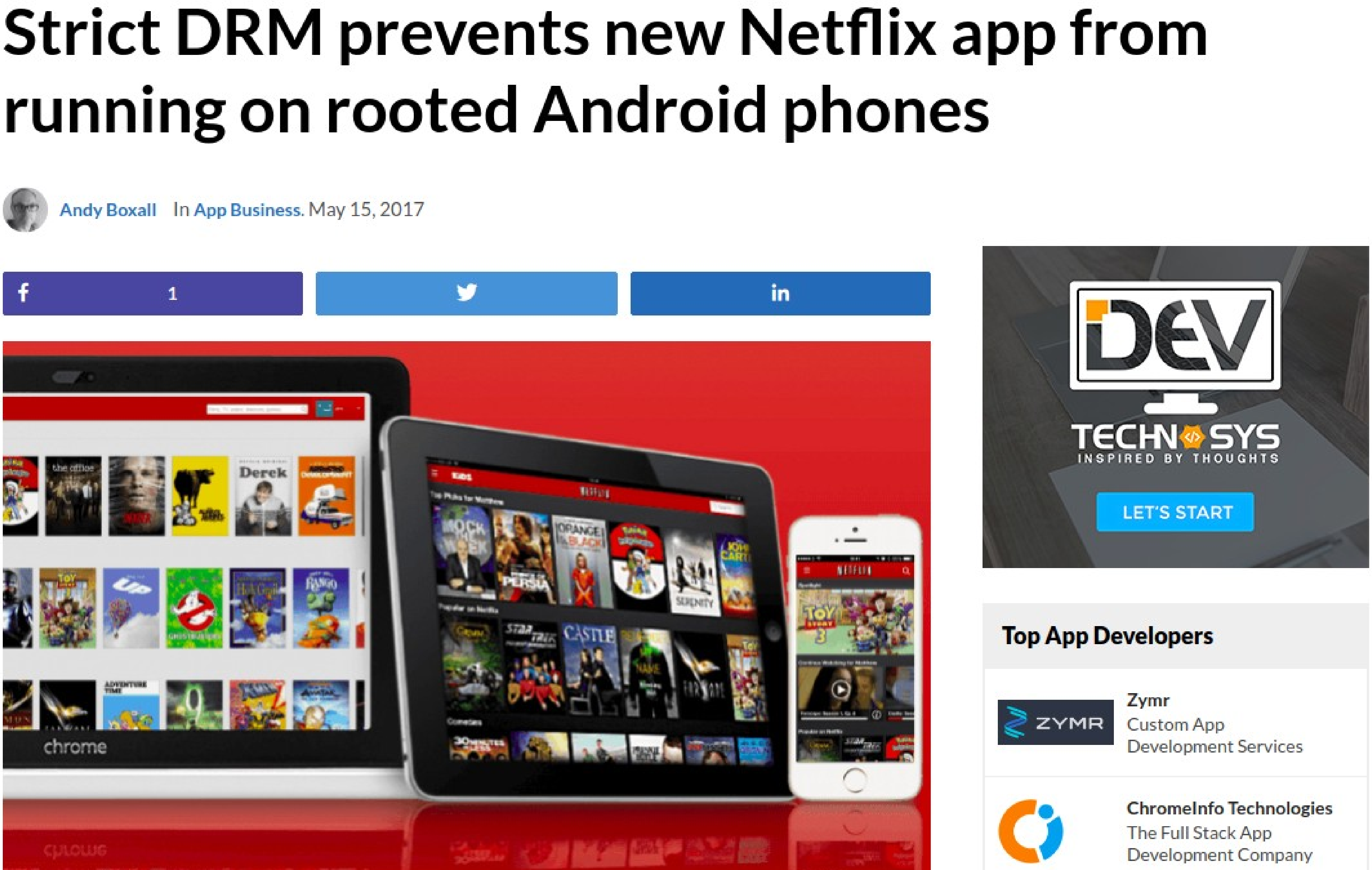
```
Emu(Ctx,I_PA,D_PA):
 I_VA = S_P2V(I_PA)
 D_VA = S_P2V(D_PA)
 Instr = Disasm(*I_VA)
 Reg = &Ctx[Instr.Rt]
 switch(Instr.Type)
```

```
case STR:
 if(AllowWrite(D_PA, Reg, …)
  IOWrite(*Reg, D_VA)
case LDR:
 if(AllowRead(D_PA, …))
  *Reg = IORead(D_VA)
  UpdateRead(D_PA, Reg, …)
```

Threat Model or Trust Model

- Does Alice trust Bob with whom she is communicating, but is wary of third parties like apps downloaded from playstore?
- Does Alice mistrust Bob, but still needs to compute some joint function with him for certain applications?
- Does Alice mistrust her own operating system, interpreters, system libraries, third party libraries?
- **Is Alice herself mistrusted?**

Digital Rights Management (DRM)



Strict DRM prevents new Netflix app from running on rooted Android phones

Andy Boxall In App Business. May 15, 2017

Top App Developers

Zymr
Custom App Development Services

ChromeInfo Technologies
The Full Stack App Development Company

Root the phone and flash TWRP

Install Xposed Framework (XposedInstaller APK and Flash Xposed APK using TWRP)

Install SSL Unpinning Module for Xposed

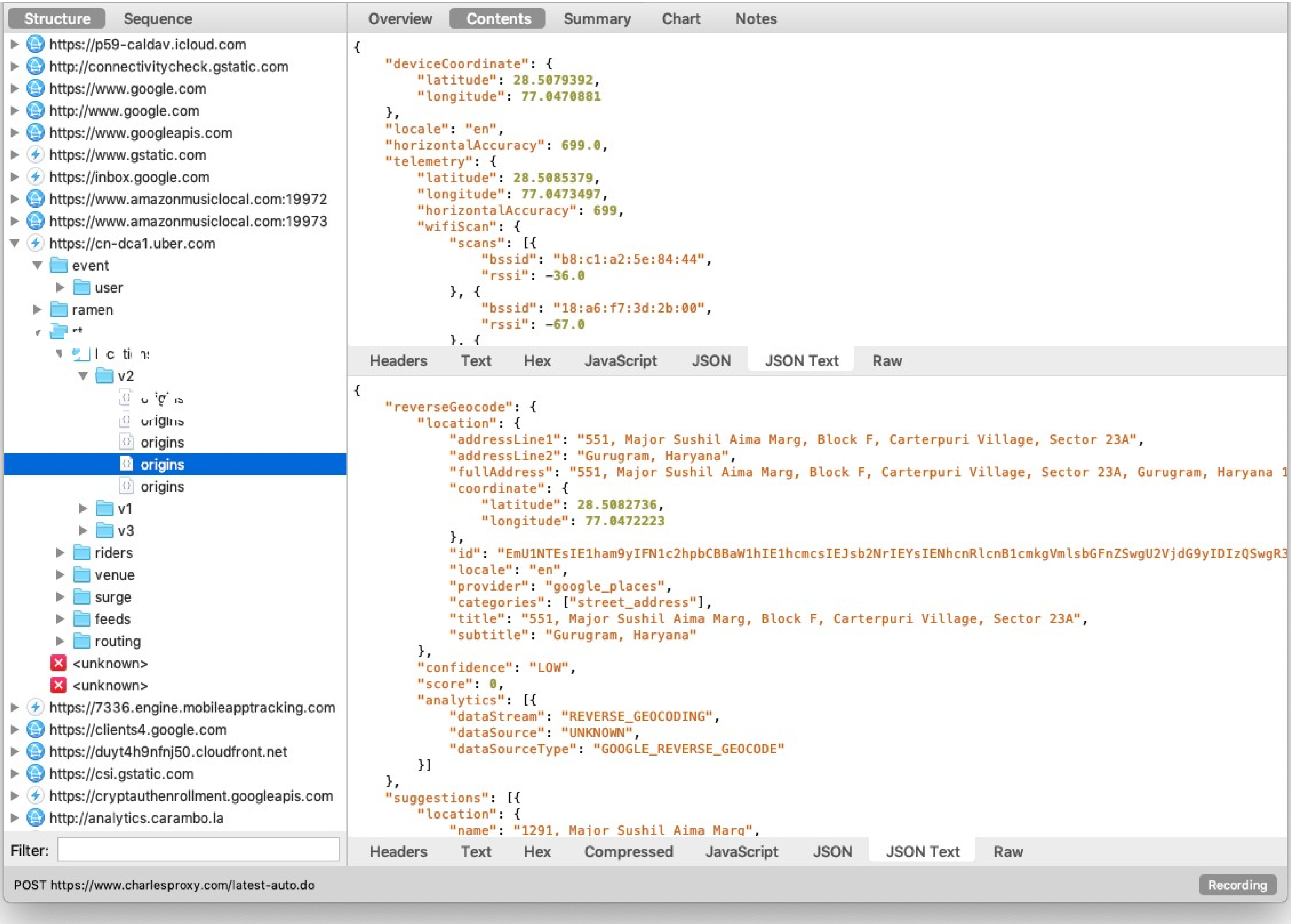
From the module: Select the App for which you want to unpin SSL and sniff data

Install Charles on your laptop

Add Charles’ Certificate on your Android device

In your Android WiFi settings, add proxy, routing your traffic through your laptop (using its IP)

Sniff!


Structure
Sequence
https://p59-caldav.icloud.com
http://connectivitycheck.gstatic.com
https://www.google.com
http://www.google.com
https://www.googleapis.com
https://www.gstatic.com
https://inbox.google.com
https://www.amazonmusiclocal.com:19972
https://www.amazonmusiclocal.com:19973
https://cn-dca1.uber.com
event
user
ramen
rt
locations
v2
origins
origins
origins
origins
origins
v1
v3
riders
venue
surge
feeds
routing
<unknown>
<unknown>
https://7336.engine.mobileapptracking.com
https://clients4.google.com
https://duyt4h9nfnj50.cloudfront.net
https://csi.gstatic.com
https://cryptauthenrollment.googleapis.com
http://analytics.carambo.la
Filter:
Overview
Contents
Summary
Chart
Notes

```
{
    "deviceCoordinate": {
        "latitude": 28.5079392,
        "longitude": 77.0470881
    },
    "locale": "en",
    "horizontalAccuracy": 699.0,
    "telemetry": {
        "latitude": 28.5085379,
        "longitude": 77.0473497,
        "horizontalAccuracy": 699,
        "wifiScan": {
            "scans": [{
                "bssid": "b8:c1:a2:5e:84:44",
                "rssi": -36.0
            }, {
                "bssid": "18:a6:f7:3d:2b:00",
                "rssi": -67.0
            }, {
```

```
{
    "reverseGeocode": {
        "location": {
            "addressLine1": "551, Major Sushil Aima Marg, Block F, Carterpuri Village, Sector 23A",
            "addressLine2": "Gurugram, Haryana",
            "fullAddress": "551, Major Sushil Aima Marg, Block F, Carterpuri Village, Sector 23A, Gurugram, Haryana 1
            "coordinate": {
                "latitude": 28.5082736,
                "longitude": 77.0472223
            },
            "id": "EmU1NTEsIE1ham9yIFN1c2hpbCBBaW1hIE1hcmcsIEJsb2NrIEYsIENhcnRlcnB1cmkgVmlsbGFnZSwgU2VjdG9yIDIzQSwgR3
            "locale": "en",
            "provider": "google_places",
            "categories": ["street_address"],
            "title": "551, Major Sushil Aima Marg, Block F, Carterpuri Village, Sector 23A",
            "subtitle": "Gurugram, Haryana"
        },
        "confidence": "LOW",
        "score": 0,
        "analytics": [{
            "dataStream": "REVERSE_GEOCODING",
            "dataSource": "UNKNOWN",
            "dataSourceType": "GOOGLE_REVERSE_GEOCODE"
        }]
    },
    "suggestions": [{
        "location": {
            "name": "1291, Major Sushil Aima Marg",
```

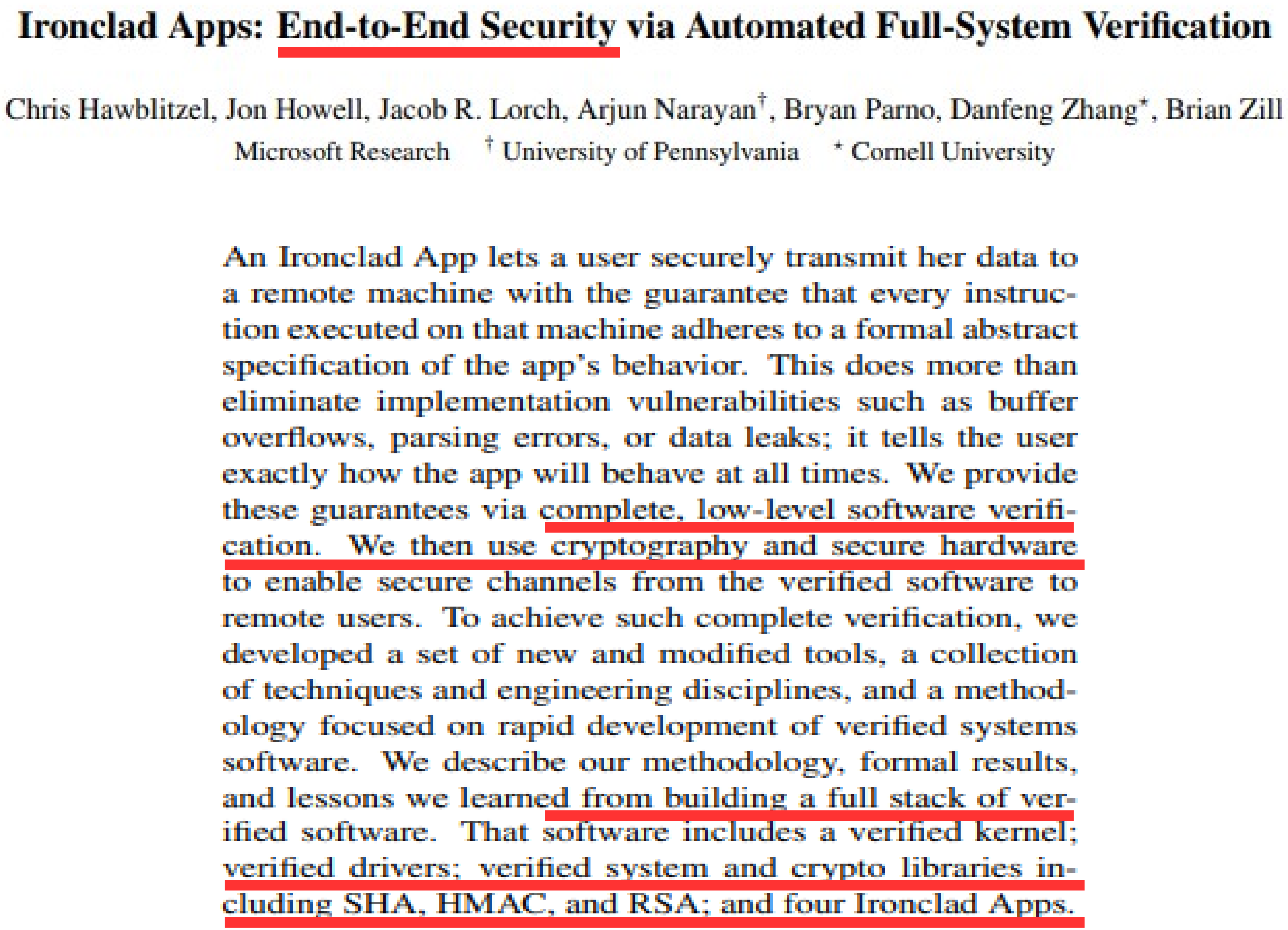
POST https://www.charlesproxy.com/latest-auto.do

Threat Model or Trust Model

- Does Alice trust Bob with whom she is communicating, but is wary of third parties like apps downloaded from playstore?
- Does Alice mistrust Bob, but still needs to compute some joint function with him for certain applications?
- Does Alice mistrust her own operating system, interpreters, system libraries, third party libraries?
- Is Alice herself mistrusted?
- ....
- **Malicious attackers? Inadvertent attackers? Honest but curious attackers? ....**

Interdisciplinary Computer Science

- **Software verification**
  - Logic, programming languages, static analysis, automata ....
- **Applied cryptography**
  - Number theory, protocol design, random number generation ....
- **Hardware security**
  - Computer architecture, new device drivers
- **Ethical hacking**
- **Network middleware, applied ML ....**
- End to end system design vs. building a part



Ironclad Apps: End-to-End Security via Automated Full-System Verification

Chris Hawblitzel, Jon Howell, Jacob R. Lorch, Arjun Narayan[†], Bryan Parno, Danfeng Zhang[*], Brian Zill
Microsoft Research [†] University of Pennsylvania [*] Cornell University

An Ironclad App lets a user securely transmit her data to a remote machine with the guarantee that every instruction executed on that machine adheres to a formal abstract specification of the app's behavior. This does more than eliminate implementation vulnerabilities such as buffer overflows, parsing errors, or data leaks; it tells the user exactly how the app will behave at all times. We provide these guarantees via complete, low-level software verification. We then use cryptography and secure hardware to enable secure channels from the verified software to remote users. To achieve such complete verification, we developed a set of new and modified tools, a collection of techniques and engineering disciplines, and a methodology focused on rapid development of verified systems software. We describe our methodology, formal results, and lessons we learned from building a full stack of verified software. That software includes a verified kernel; verified drivers; verified system and crypto libraries including SHA, HMAC, and RSA; and four Ironclad Apps.

Interdisciplinary Computer Science

- **Software verification**
  - Logic, programming languages, static analysis, automata ....
- **Applied cryptography**
  - Number theory, protocol design, random number generation ....
- **Hardware security**
  - Computer architecture, new device drivers
- **Ethical hacking**
- **Network middleware, applied ML ....**
- End to end system design vs. building a part

Papers in wide range of conferences. For applied cryptography, CRYPTO, EUROCRYPT etc. will have mathematical proofs, NDSS, Oakland, S&P will have cryptographic protocols, SOUPS, PETS and CHI will have user studies interacting with cryptographic systems ...

Additional Challenges for Embedded Systems

- Low computational capabilities causing latency
- Energy constraints
- Memory and storage constraints

Handled with

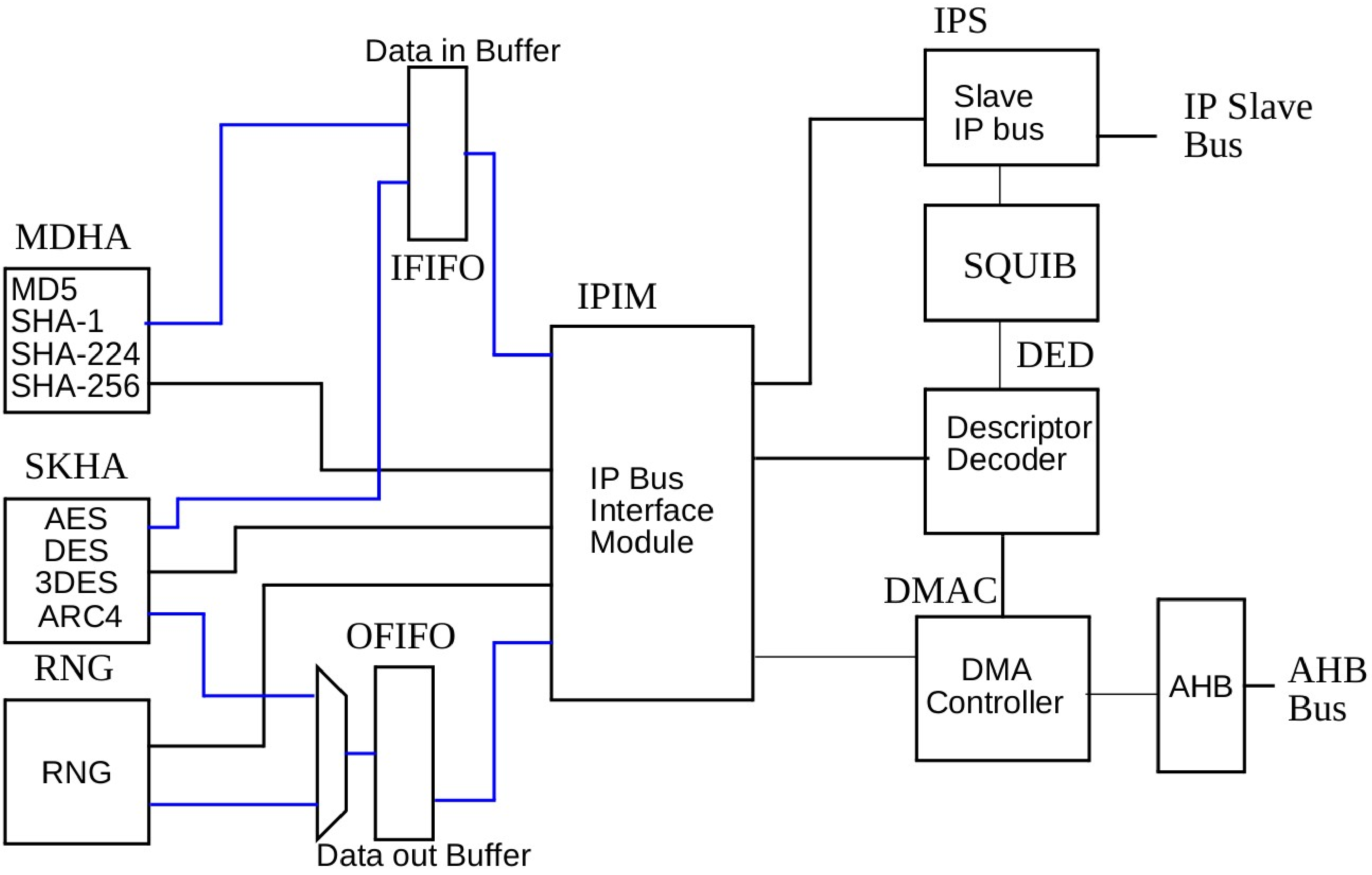
  - Split computations between cloud and device, taking care not to introduce new security threats
  - Efficient data structures
  - Hardware cryptographic accelerators
  - ....

Security Accelerator or Co-Processor

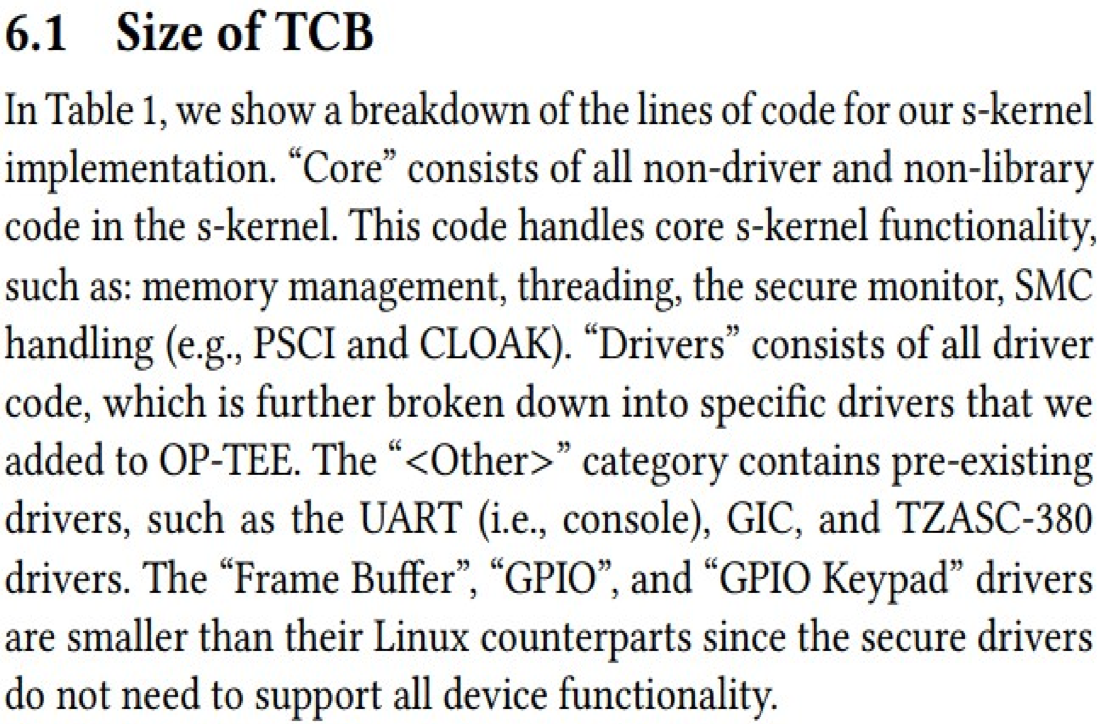
- E.g.- The Symmetric/Asymmetric Hashing and Random Accelerator (SAHARA) is a security coprocessor present in IMX53 SoC

- SAHARA implements
  - block encryption algorithms, (AES, DES, and 3DES),
  - hashing algorithms (MD5, SHA-1, SHA-224, and SHA-256),
  - a stream cipher algorithm (ARC4),
  - and a hardware random number generator.
  -
- It has a slave IP bus interface for the host to write configuration and command information, and to read status information.
- It also has a DMA controller, with an AHB bus interface, to reduce the burden on the host to move the required data to and from memory.

Block Diagram

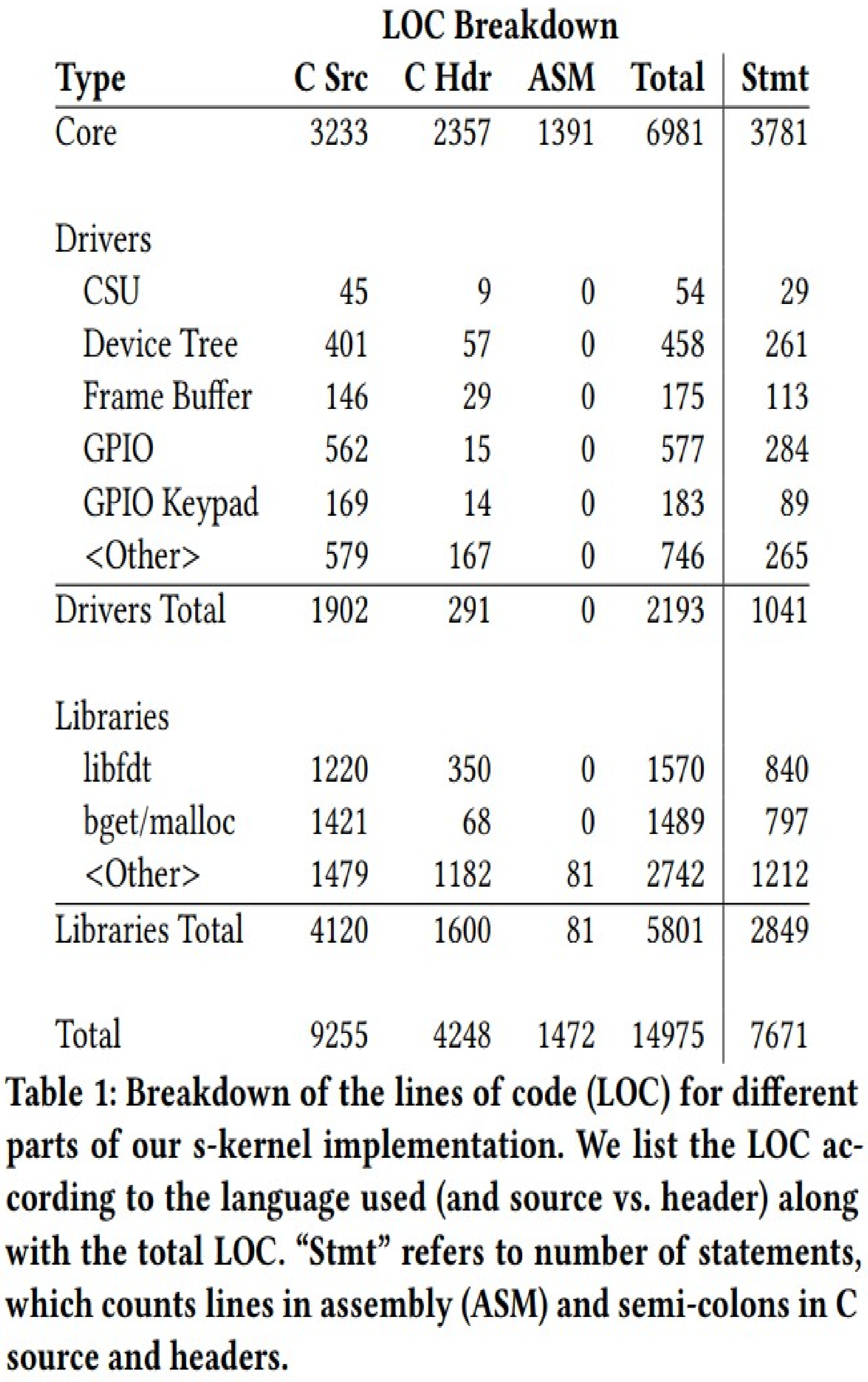


Small Trusted Code Base



6.1 Size of TCB

In Table 1, we show a breakdown of the lines of code for our s-kernel implementation. “Core” consists of all non-driver and non-library code in the s-kernel. This code handles core s-kernel functionality, such as: memory management, threading, the secure monitor, SMC handling (e.g., PSCI and CLOAK). “Drivers” consists of all driver code, which is further broken down into specific drivers that we added to OP-TEE. The “<Other>” category contains pre-existing drivers, such as the UART (i.e., console), GIC, and TZASC-380 drivers. The “Frame Buffer”, “GPIO”, and “GPIO Keypad” drivers are smaller than their Linux counterparts since the secure drivers do not need to support all device functionality.



	LOC Breakdown				
Type	C Src	C Hdr	ASM	Total	Stmt
Core	3233	2357	1391	6981	3781
Drivers					
CSU	45	9	0	54	29
Device Tree	401	57	0	458	261
Frame Buffer	146	29	0	175	113
GPIO	562	15	0	577	284
GPIO Keypad	169	14	0	183	89
<Other>	579	167	0	746	265
Drivers Total	1902	291	0	2193	1041
Libraries					
libfdt	1220	350	0	1570	840
bget/malloc	1421	68	0	1489	797
<Other>	1479	1182	81	2742	1212
Libraries Total	4120	1600	81	5801	2849
Total	9255	4248	1472	14975	7671

Table 1: Breakdown of the lines of code (LOC) for different parts of our s-kernel implementation. We list the LOC according to the language used (and source vs. header) along with the total LOC. “Stmt” refers to number of statements, which counts lines in assembly (ASM) and semi-colons in C source and headers.

Interdisciplinary Computer Science

- **Software verification**
  - Logic, programming languages, static analysis, automata ....
- **Applied cryptography**
  - Number theory, protocol design, random number generation ....
- **Hardware security**
  - Computer architecture, new device drivers
- **Ethical hacking**
- **Network middleware, applied ML ....**
- End to end system design vs. building a part

Papers in wide range of conferences. For applied cryptography, CRYPTO, EUROCRYPT etc. will have mathematical proofs, NDSS, Oakland, S&P will have cryptographic protocols, SOUPS, PETS and CHI will have user studies interacting with cryptographic systems ...