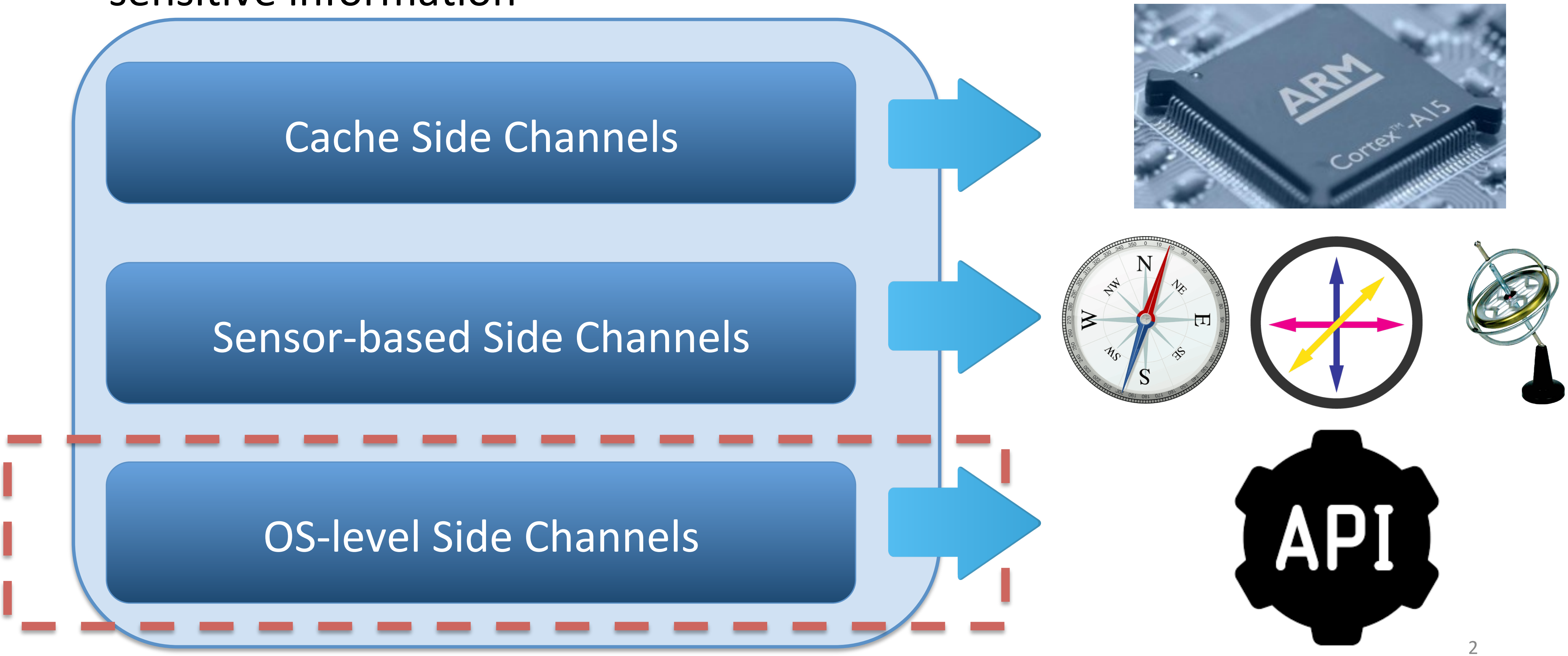
OS-level Side Channels without Procfs: Exploring Cross-App Information Leakage on iOS

Xiaokuan Zhang[1], Xueqiang Wang[2], Xiaolong Bai[3],
Yinqian Zhang[1] and XiaoFeng Wang[2]

[1]The Ohio State University, [2]Indiana University Bloomington, [3]Tsinghua University

Mobile Side-Channel Attacks

- Side-channel Attack: make use of seemingly harmless information to infer sensitive information



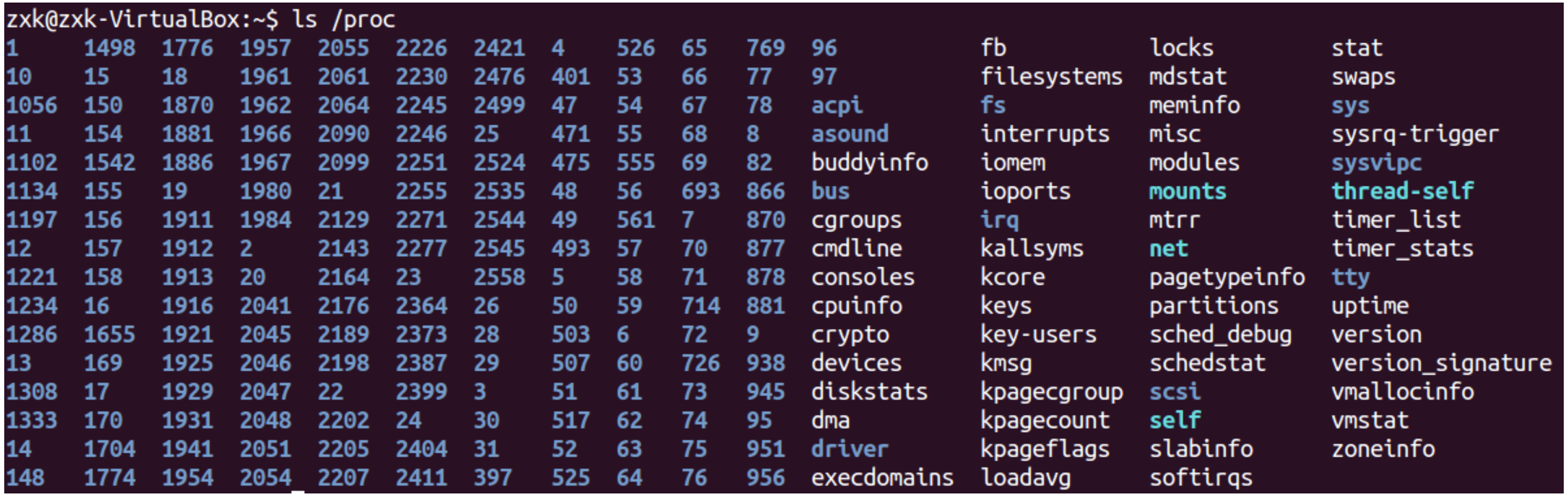
Cache Side Channels

Sensor-based Side Channels

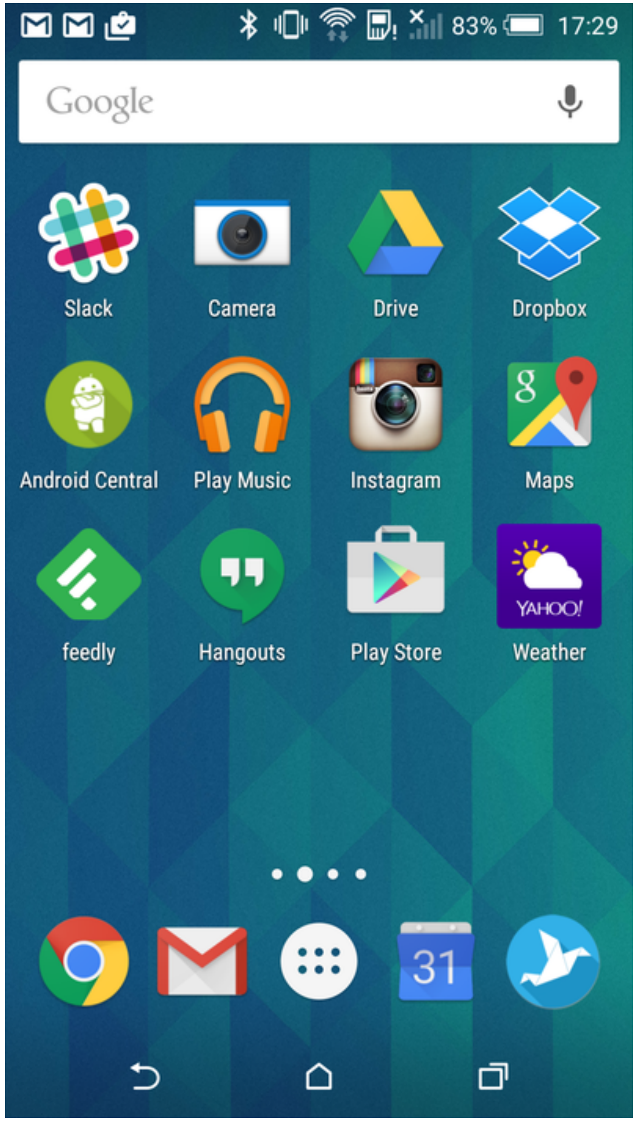
OS-level Side Channels

OS-level Side-Channel Attacks on Android

- Malicious app running in the background, calling APIs
- Procfs: system statistics
  - virtual/physical memory, network traffic, CPU usage info, …

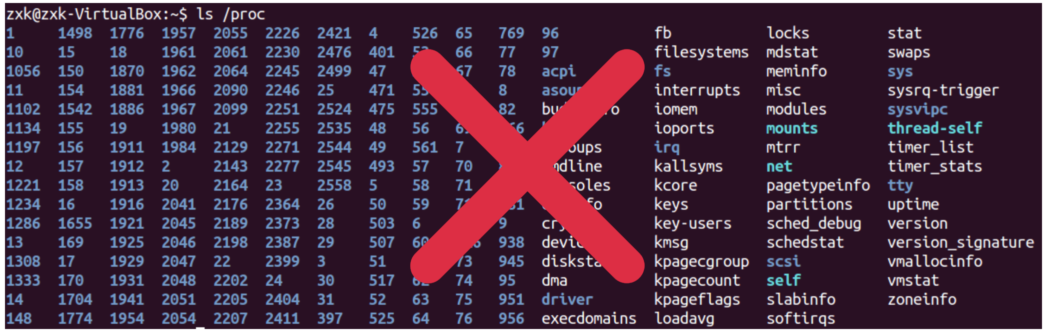


```
zxk@zxk-VirtualBox:~$ ls /proc
1     1498  1776  1957  2055  2226  2421  4    526  65   769  96           fb           locks         stat
10    15    18    1961  2061  2230  2476  401  53   66   77   97           filesystems  mdstat        swaps
1056  150   1870  1962  2064  2245  2499  47   54   67   78   acpi         fs           meminfo       sys
11    154   1881  1966  2090  2246  25    471  55   68   8    asound       interrupts   misc          sysrq-trigger
1102  1542  1886  1967  2099  2251  2524  475  555  69   82   buddyinfo    iomem        modules       sysvipc
1134  155   19    1980  21    2255  2535  48   56   693  866  bus          ioports      mounts        thread-self
1197  156   1911  1984  2129  2271  2544  49   561  7    870  cgroups      irq          mtrr          timer_list
12    157   1912  2     2143  2277  2545  493  57   70   877  cmdline      kallsyms     net           timer_stats
1221  158   1913  20    2164  23    2558  5    58   71   878  consoles     kcore        pagetypeinfo  tty
1234  16    1916  2041  2176  2364  26    50   59   714  881  cpuinfo      keys         partitions    uptime
1286  1655  1921  2045  2189  2373  28    503  6    72   9    crypto       key-users    sched_debug   version
13    169   1925  2046  2198  2387  29    507  60   726  938  devices      kmsg         schedstat     version_signature
1308  17    1929  2047  22    2399  3     51   61   73   945  diskstats    kpagecgroup  scsi          vmallocinfo
1333  170   1931  2048  2202  24    30    517  62   74   95   dma          kpagecount   self          vmstat
14    1704  1941  2051  2205  2404  31    52   63   75   951  driver       kpageflags   slabinfo      zoneinfo
148   1774  1954  2054  2207  2411  397   525  64   76   956  execdomains  loadavg      softirqs
```



OS-level Side-Channel Attacks on iOS

- No Procfs providing system stat



```
zxk@zxk-VirtualBox:~$ ls /proc
1     1498  1776  1957  2055  2226  2421  4    526  65   769  96           fb           locks         stat
10    15    18    1961  2061  2230  2476  401  5    66   77   97           filesystems  mdstat        swaps
1056  150   1870  1962  2064  2245  2499  47        67   78   acpi         fs           meminfo       sys
11    154   1881  1966  2090  2246  25    471  55        8    asou         interrupts   misc          sysrq-trigger
1102  1542  1886  1967  2099  2251  2524  475  555       82   bu     o     iomem        modules       sysvipc
1134  155   19    1980  21    2255  2535  48   56   6     6                ioports      mounts        thread-self
1197  156   1911  1984  2129  2271  2544  49   561  7           oups       irq          mtrr          timer_list
12    157   1912  2     2143  2277  2545  493  57   70          ndline     kallsyms     net           timer_stats
1221  158   1913  20    2164  23    2558  5    58   71          soles      kcore        pagetypeinfo  tty
1234  16    1916  2041  2176  2364  26    50   59   7     1         fo     keys         partitions    uptime
1286  1655  1921  2045  2189  2373  28    503  6         9    cr           key-users    sched_debug   version
13    169   1925  2046  2198  2387  29    507  60     6  938  devi         kmsg         schedstat     version_signature
1308  17    1929  2047  22    2399  3     51        73   945  disksta      kpagecgroup  scsi          vmallocinfo
1333  170   1931  2048  2202  24    30    517  62   74   95   dma          kpagecount   self          vmstat
14    1704  1941  2051  2205  2404  31    52   63   75   951  driver       kpageflags   slabinfo      zoneinfo
148   1774  1954  2054  2207  2411  397   525  64   76   956  execdomains  loadavg      softirqs
```

- No unauthorized cross-app query

Is it possible to conduct OS-level side-channel attacks on iOS?

Outline

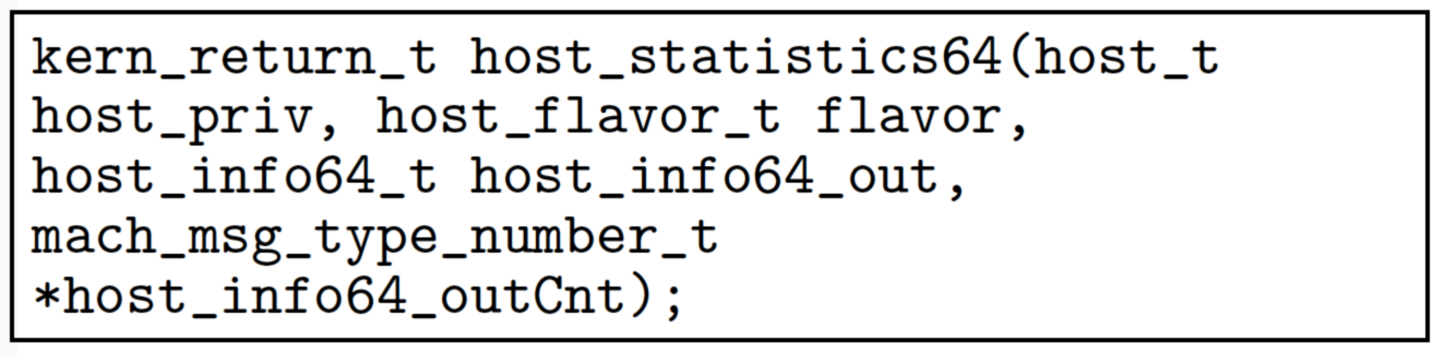
1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion

Threat Model

- Monitoring app:
  - User downloads it from App Store
  - Audio player

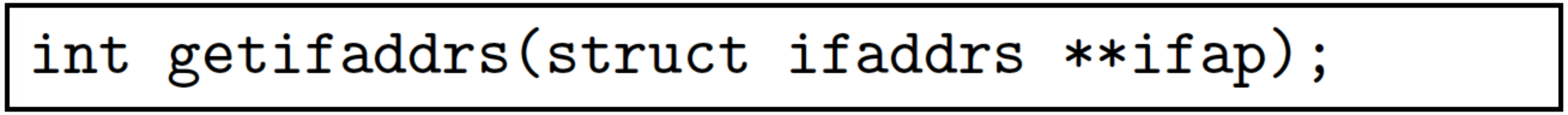
New Attack Vectors

- Host_statistics64(): Gl



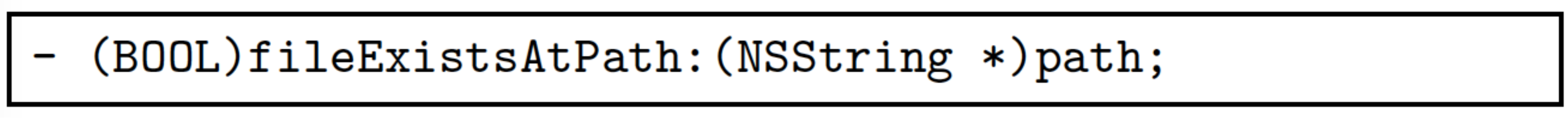
```
kern_return_t host_statistics64(host_t
host_priv, host_flavor_t flavor,
host_info64_t host_info64_out,
mach_msg_type_number_t
*host_info64_outCnt);
```

- Getifaddrs():



```
int getifaddrs(struct ifaddrs **ifap);
```

- [NSFileManager fileExistsAtPath:]: The existence of a file/directory



```
- (BOOL)fileExistsAtPath:(NSString *)path;
```

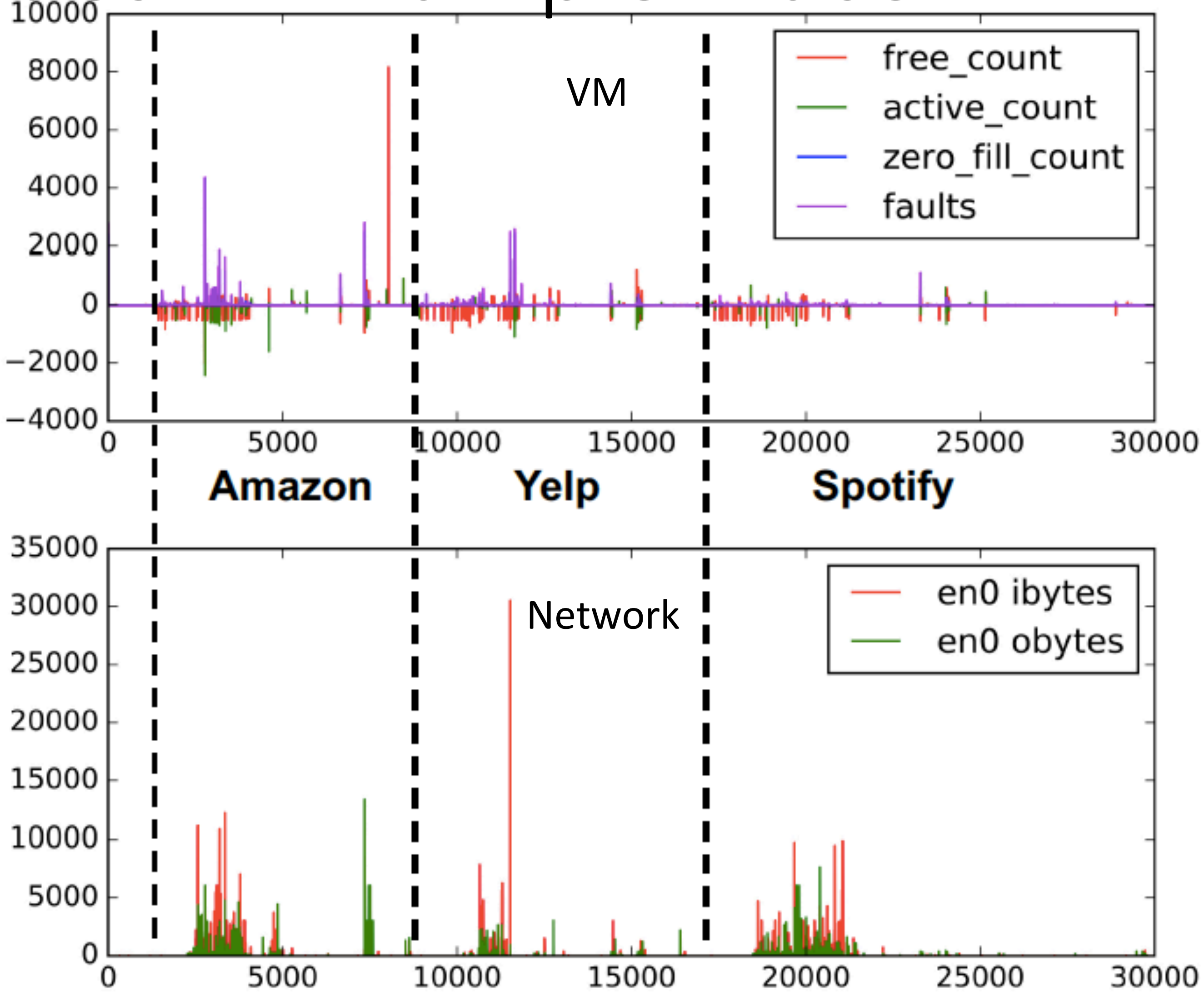
Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion

Classifying User Activities --- Example Trace

amazon
yelp

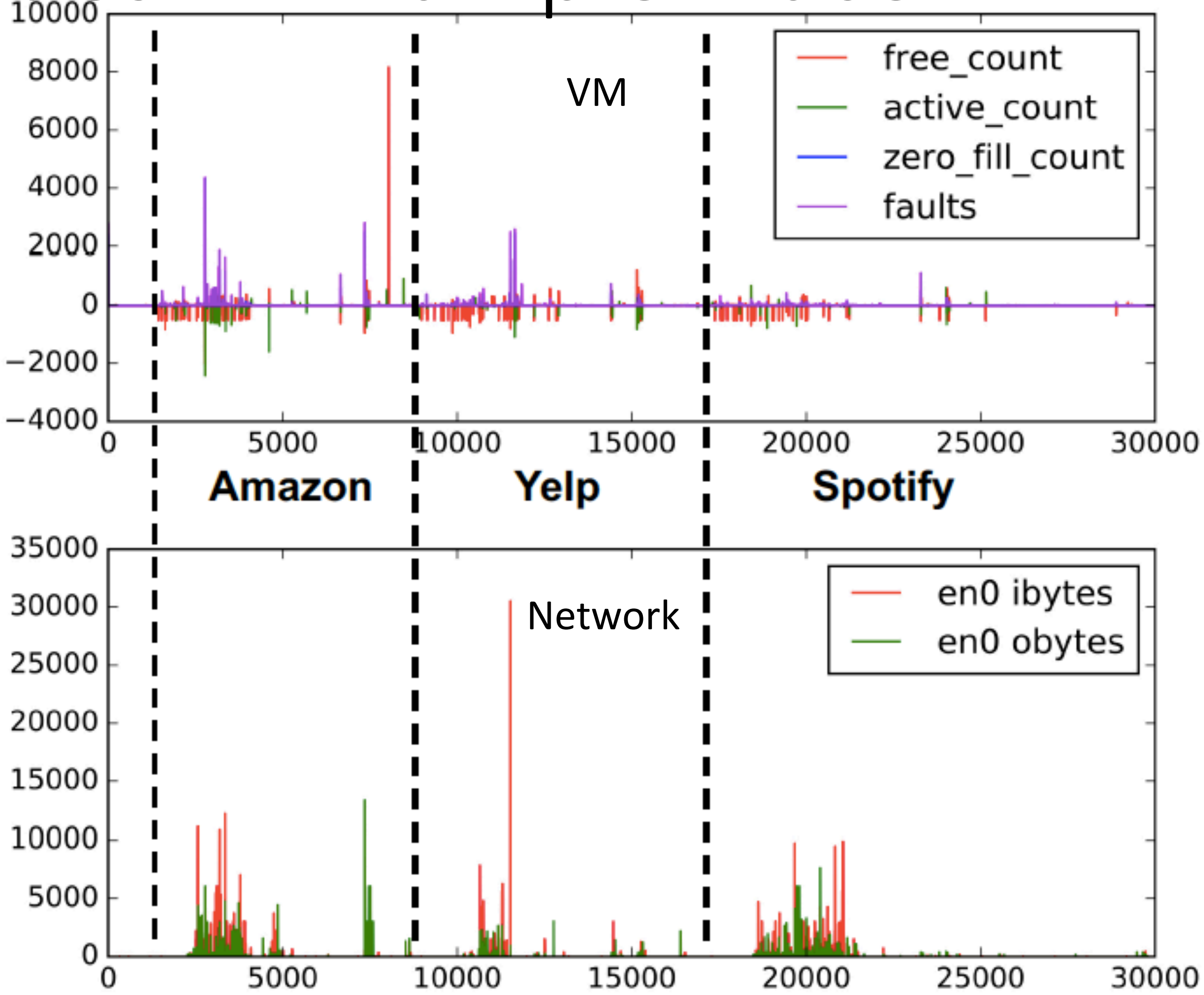
- Calling APIs to get time series A
  - Host_statistics64()
  - Getifaddrs()
- Plotting diff series: A[i] – A[i-1]



Time series leak information!!!

Classifying User Activities --- Example Trace

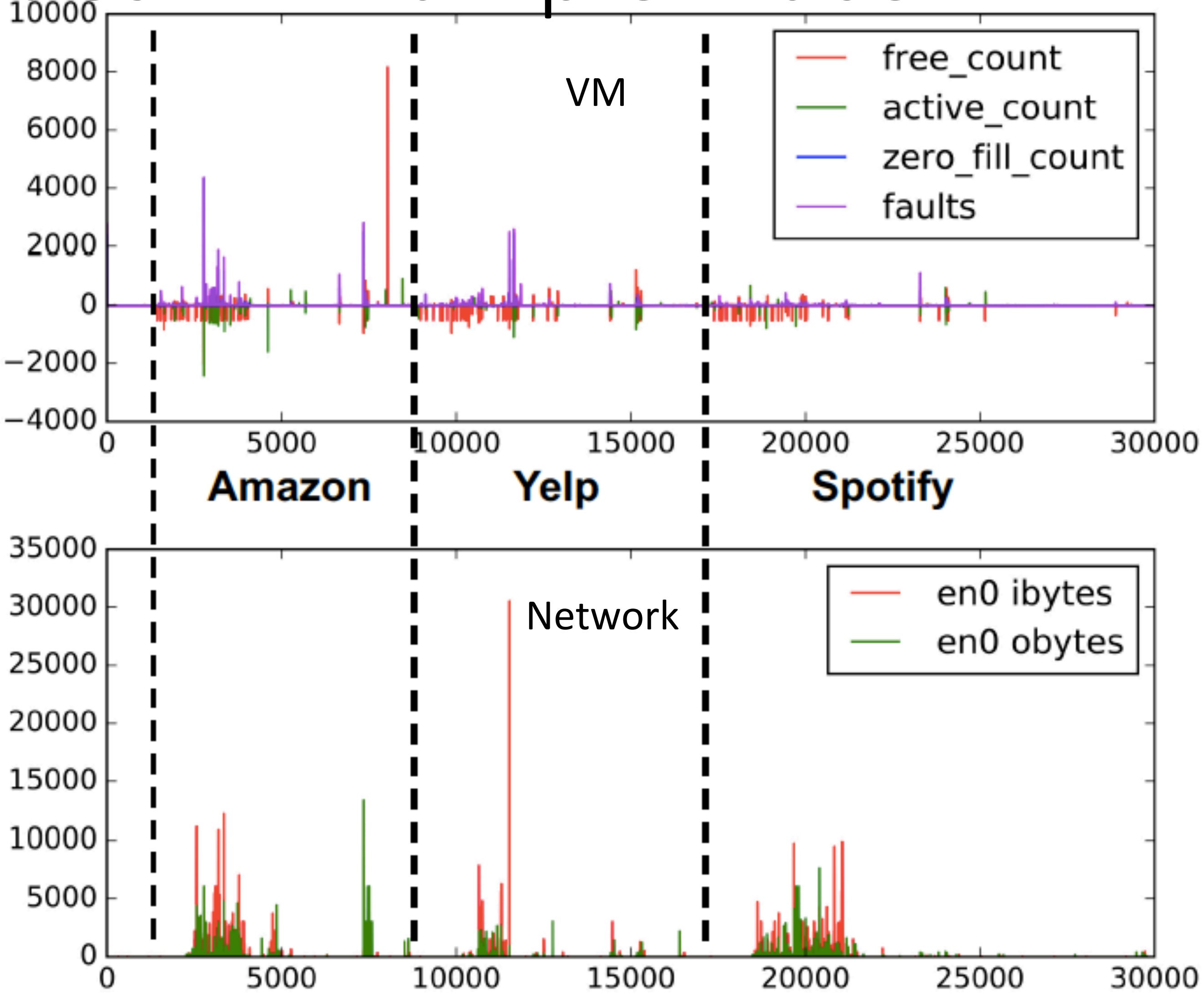
How to combine multiple time series to perform inference attacks?



Classifying User Activities --- Example Trace

How to combine multiple time series to perform inference attacks?

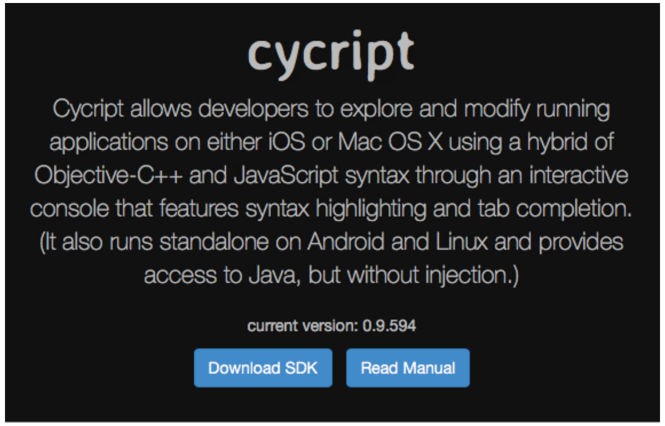
- Requirements:
  - Combining multiple time series
  - Reducing the dimension
- Major components:
  - SAX (Keogh et al., 2002)
  - BOP (Lin et al., 2009)
  - LibSVM (Chang et al., 2011)



Classifying User Activities --- Case Studies

- Device: jailbroken iPhone 7 with iOS 10.1.1

- Automated using Cycript



cycript

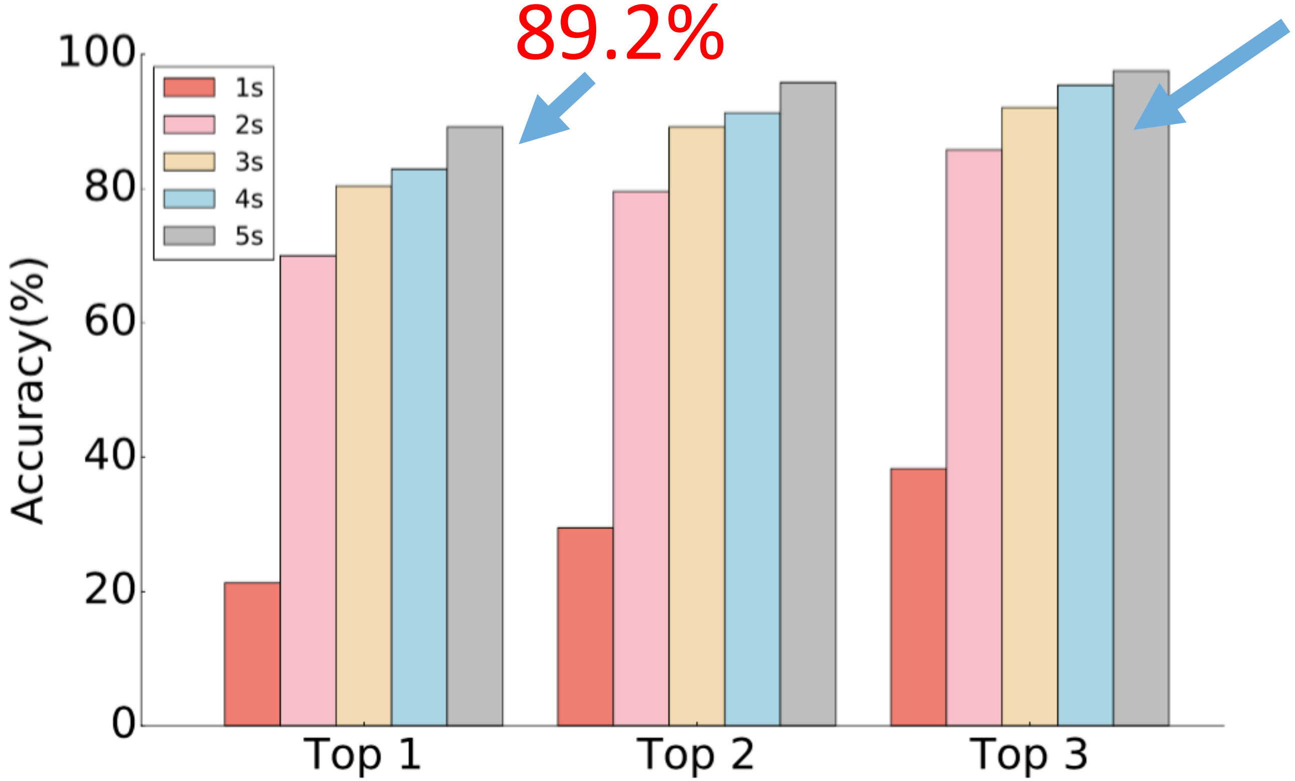
Cycript allows developers to explore and modify running applications on either iOS or Mac OS X using a hybrid of Objective-C++ and JavaScript syntax through an interactive console that features syntax highlighting and tab completion. (It also runs standalone on Android and Linux and provides access to Java, but without injection.)

current version: 0.9.594

- Monitoring app:
  - running in the background
  - calling APIs at a rate of 1000/s

Classifying User Activities --- Case Studies

- Foreground Apps:
  - 100 apps from Top Charts + 20 pre-installed apps
  - Top N accuracy: the percentage of the test samples being correctly labeled by one of the top N predicted classes by the classifier

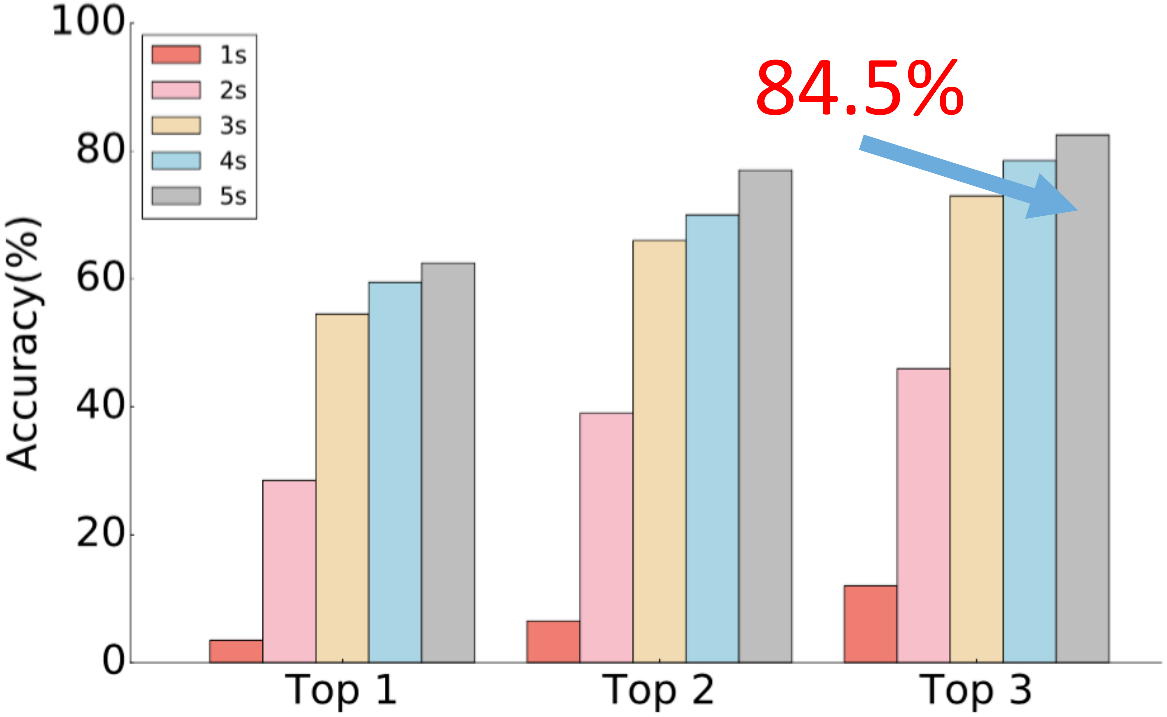


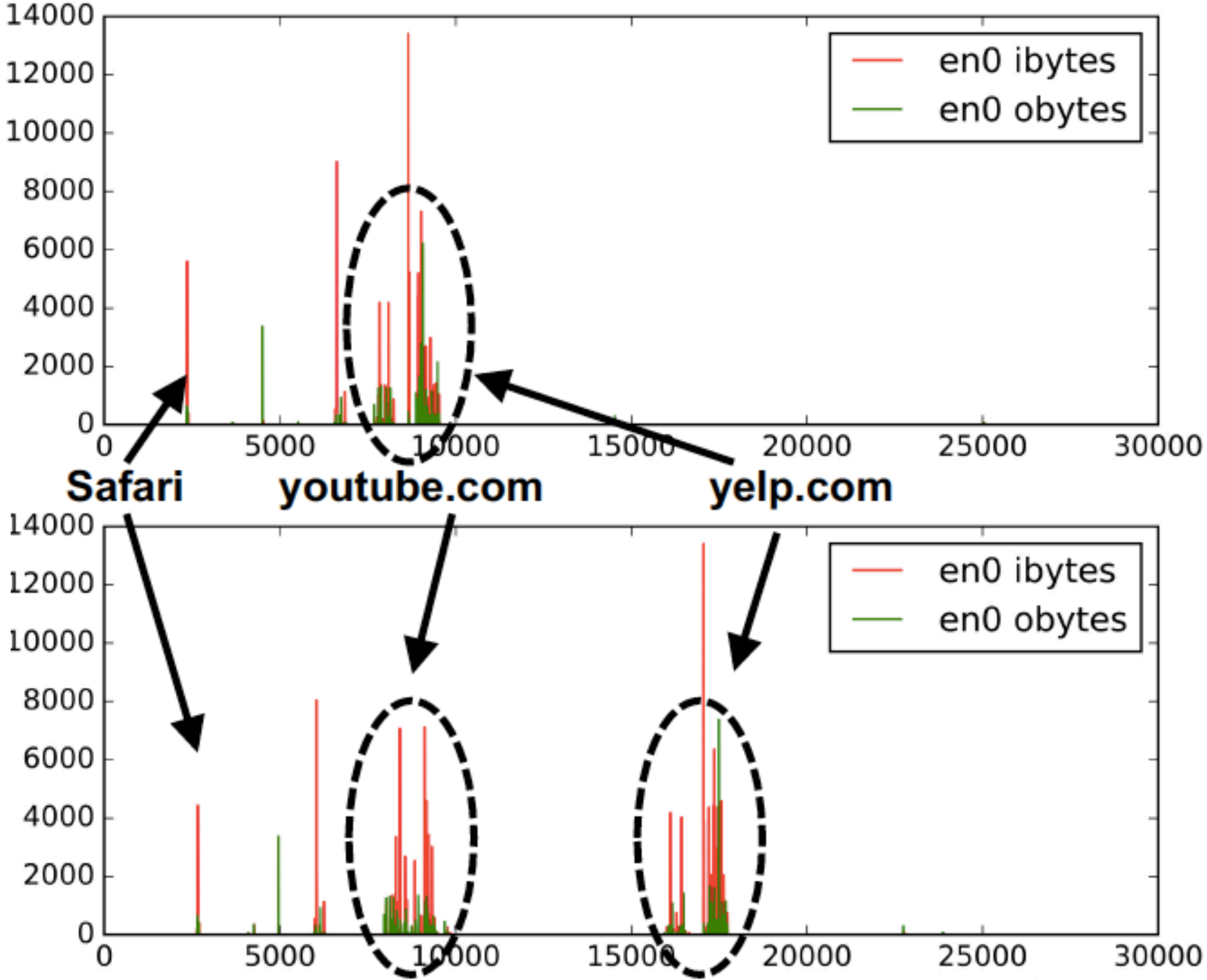
89.2%

97.5%

Classifying User Activities --- Case Studies

- Safari Websites

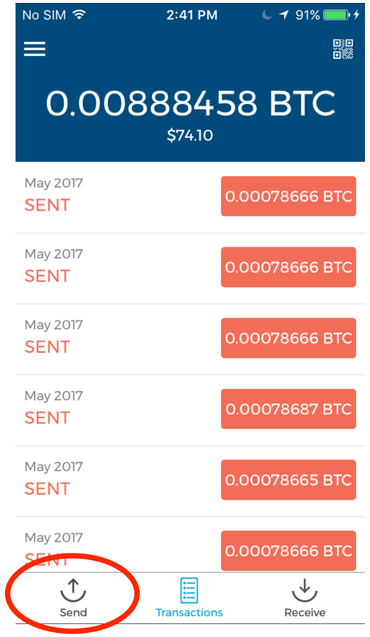




Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion

Detecting Sensitive In-App Activities

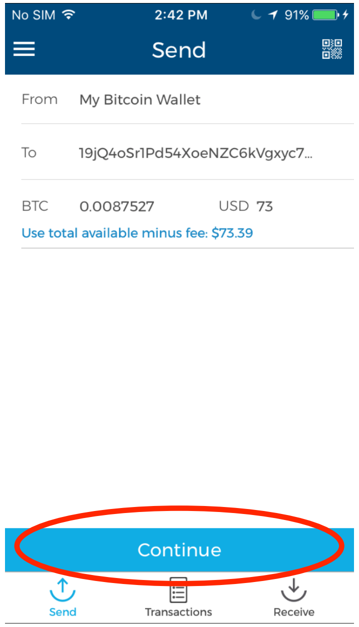


No SIM 2:41 PM 91%

0.00888458 BTC

$74.10

May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078687 BTC
May 2017 SENT	0.00078665 BTC
May 2017 SENT	0.00078666 BTC



No SIM 2:42 PM 91%

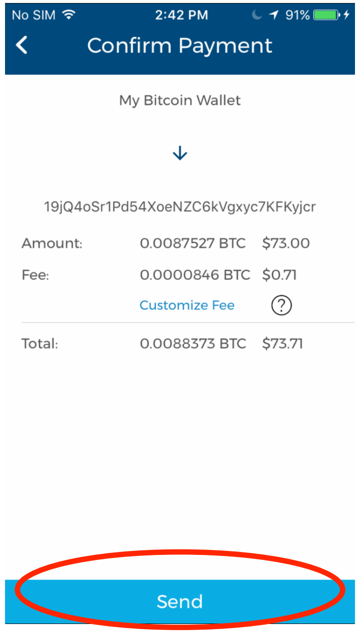
Send

From My Bitcoin Wallet

To 19jQ4oSr1Pd54XoeNZC6kVgxyc7...

BTC 0.0087527 USD 73

Use total available minus fee: $73.39



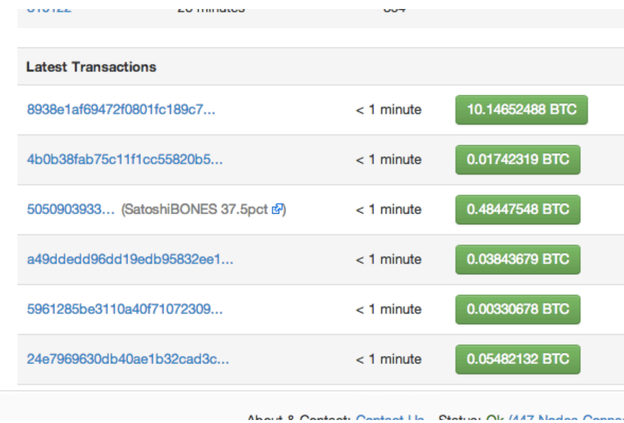
No SIM 2:42 PM 91%

Confirm Payment

My Bitcoin Wallet

19jQ4oSr1Pd54XoeNZC6kVgxyc7KFKyjcr

Amount:	0.0087527 BTC	$73.00
Fee:	0.0000846 BTC	$0.71
	Customize Fee	
Total:	0.0088373 BTC	$73.71



Latest Transactions

8938e1af69472f0801fc189c7...	< 1 minute	10.14652488 BTC
4b0b38fab75c11f1cc55820b5...	< 1 minute	0.01742319 BTC
5050903933... (SatoshiBONES 37.5pct)	< 1 minute	0.48447548 BTC
a49ddedd96dd19edb95832ee1...	< 1 minute	0.03843679 BTC
5961285be3110a40f71072309...	< 1 minute	0.00330678 BTC
24e7969630db40ae1b32cad3c...	< 1 minute	0.05482132 BTC

Blockchain.info

Detecting Sensitive In-App Activities --- Attack Methods

- Identify critical events

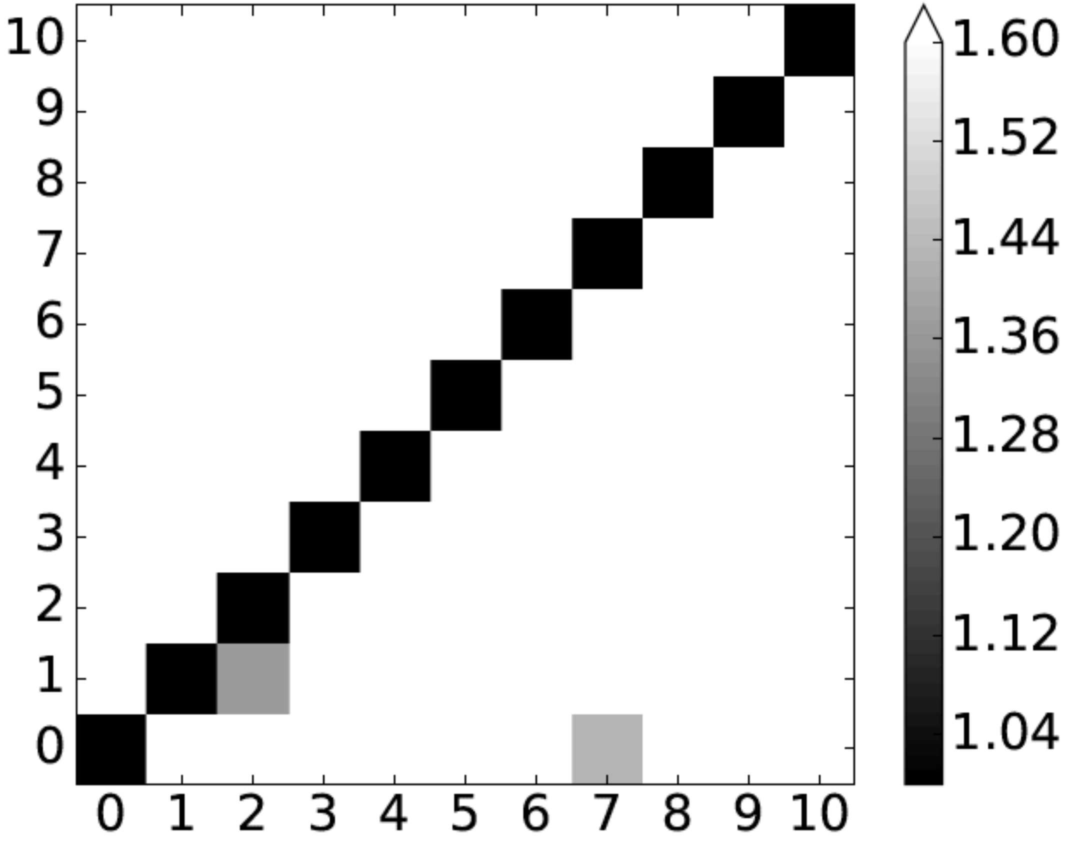
- Correlates with public records

Detecting Sensitive In-App Activities --- Case Studies

- Target: *Blockchain Wallet* App

- Goal: identify *payment* event (idx: 0)

$$d(\vec{X}_t, \vec{S}_t) = \sum_{k=1}^{l} \frac{1}{w_k} \cdot \mathrm{DTW}\big(\vec{X}_t^k, \vec{S}_t^k\big)$$



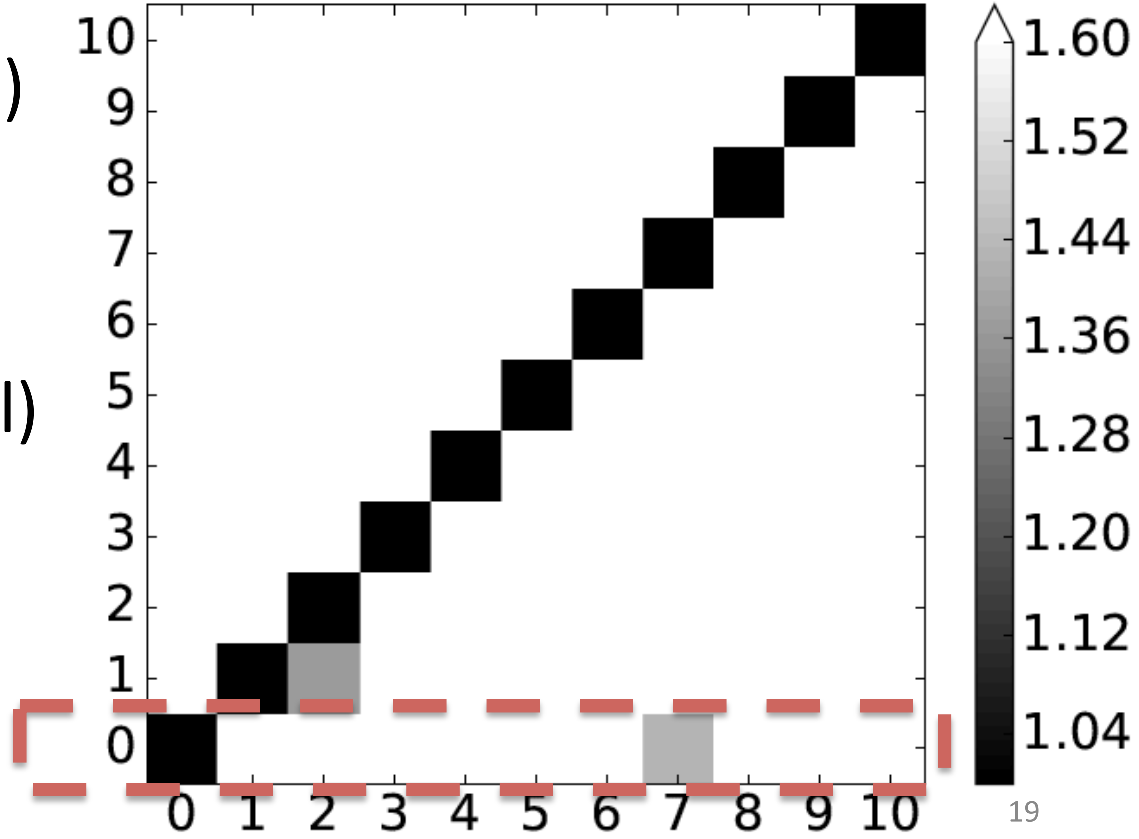
Detecting Sensitive In-App Activities --- Case Studies

- Target: *Blockchain Wallet* App

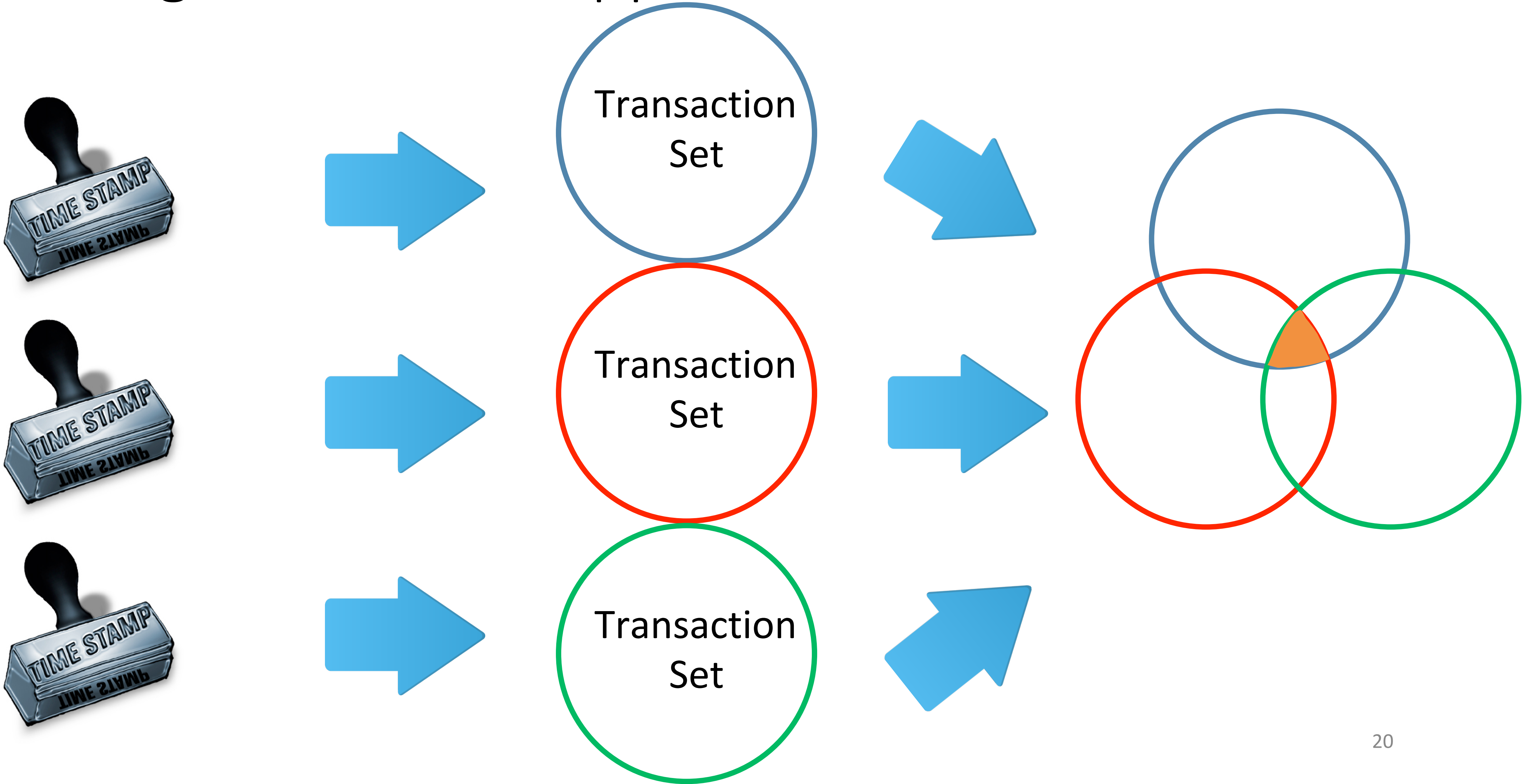
- Goal: identify *payment* event (idx: 0)

- Normalize the distance per row
  using cell(i,i) as the base (diagonal)

$$d(\vec{X}_t, \vec{S}_t) = \sum_{k=1}^{l} \frac{1}{w_k} \cdot \mathrm{DTW}\left(\vec{X}_t^k, \vec{S}_t^k\right)$$



Detecting Sensitive In-App Activities --- Case Studies

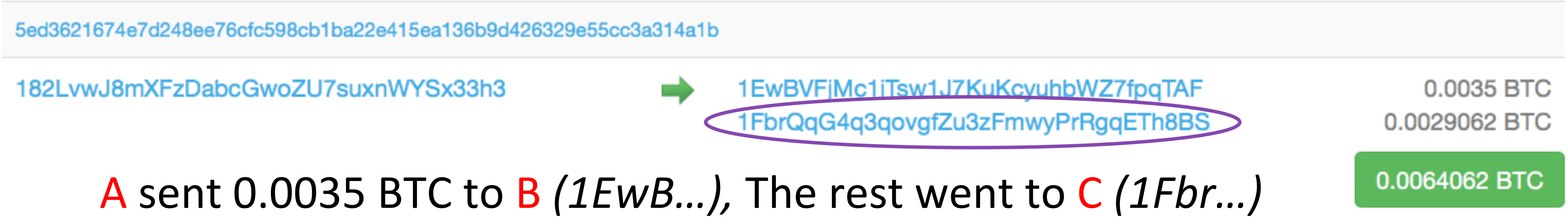


Transaction Set

Transaction Set

Transaction Set

Detecting Sensitive In-App Activities --- Case Studies



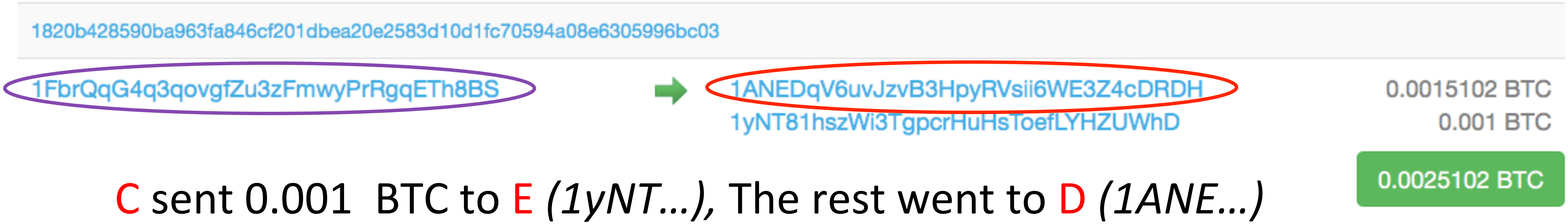
5ed3621674e7d248ee76cfc598cb1ba22e415ea136b9d426329e55cc3a314a1b

182LvwJ8mXFzDabcGwoZU7suxnWYSx33h3 → 1EwBVFjMc1iTsw1J7KuKcyuhbWZ7fpqTAF 0.0035 BTC

1FbrQqG4q3qovgfZu3zFmwyPrRgqETh8BS 0.0029062 BTC

0.0064062 BTC

A sent 0.0035 BTC to B *(1EwB…)*, The rest went to C *(1Fbr…)*



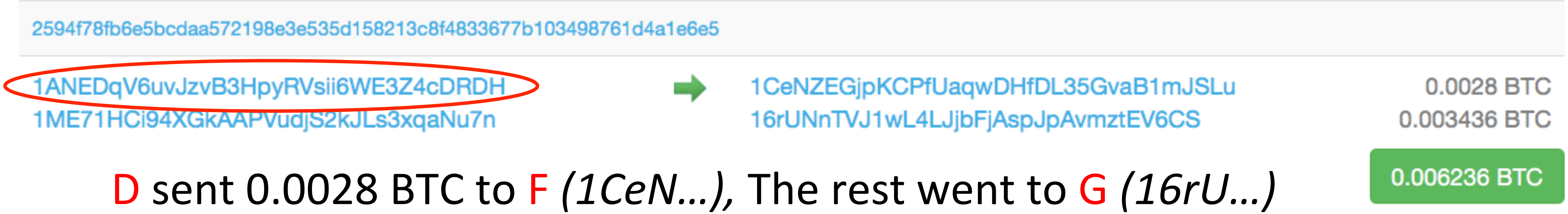
1820b428590ba963fa846cf201dbea20e2583d10d1fc70594a08e6305996bc03

1FbrQqG4q3qovgfZu3zFmwyPrRgqETh8BS → 1ANEDqV6uvJzvB3HpyRVsii6WE3Z4cDRDH 0.0015102 BTC

1yNT81hszWi3TgpcrHuHsToefLYHZUWhD 0.001 BTC

0.0025102 BTC

C sent 0.001 BTC to E *(1yNT…)*, The rest went to D *(1ANE…)*



2594f78fb6e5bcdaa572198e3e535d158213c8f4833677b103498761d4a1e6e5

1ANEDqV6uvJzvB3HpyRVsii6WE3Z4cDRDH → 1CeNZEGjpKCPfUaqwDHfDL35GvaB1mJSLu 0.0028 BTC

1ME71HCi94XGkAAPVudjS2kJLs3xqaNu7n 16rUNnTVJ1wL4LJjbFjAspJpAvmztEV6CS 0.003436 BTC

0.006236 BTC

D sent 0.0028 BTC to F *(1CeN…)*, The rest went to G *(16rU…)*

Detecting Sensitive In-App Activities --- Case Studies

- Other Targets: *Venmo / Twitter*

Outline

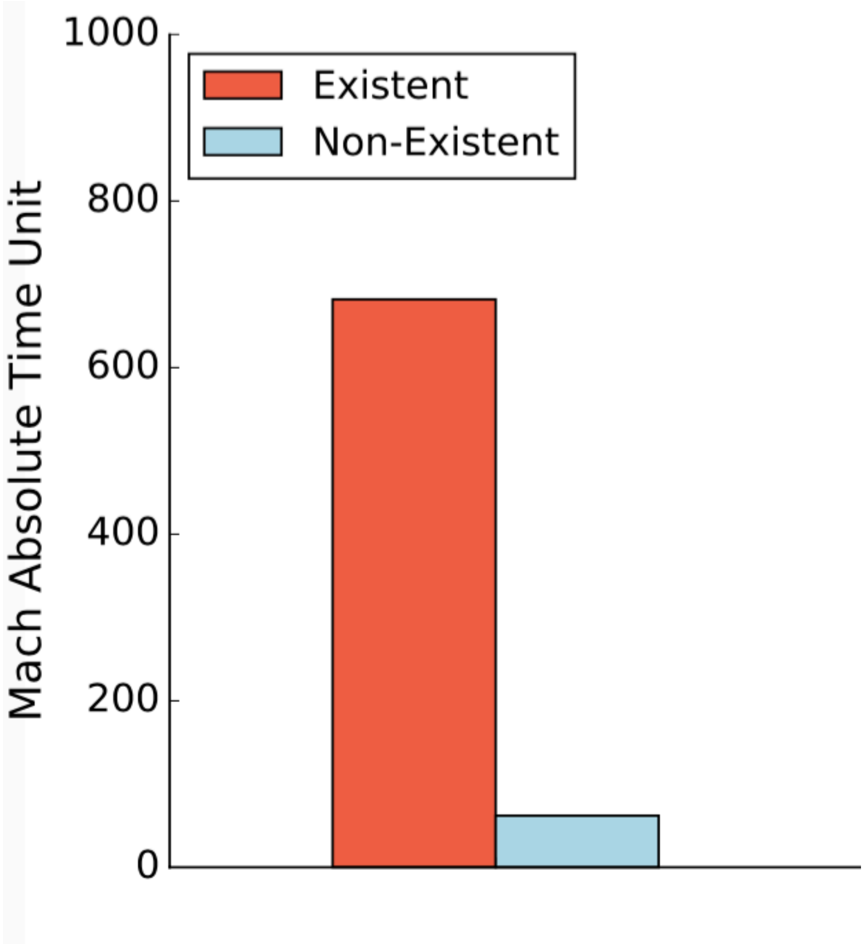
1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion

Bypassing Sandbox Restrictions --- Attack Methods

- Device: non-jailbroken iPhone 7 with iOS 10.2.1

- Execution time of FileExistAtPath

Huge Difference!!!



Bypassing Sandbox Restrictions --- Case Studies

- Detect whether an app has been installed

DivorceForce

AsthmaMD

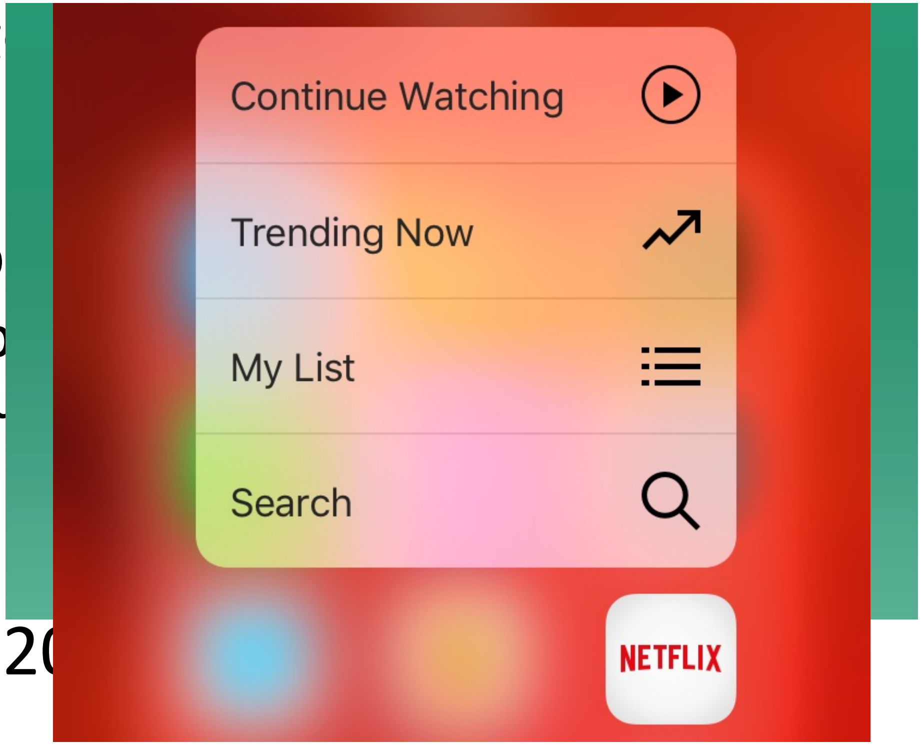
Pregnancy+

Sugar Sense

Bypassing Sandbox Restrictions --- Case Studies

- Push notifications:
  - .pushstore file with the bundle identifier as its name will be created in a specific directory
  - (/var/mobile/Library/SpringBoard/PushStore/com.g Gmail app)

- Dynamically registered home screen quick actio
  - .plist file with the bundle identifier as its name will b var/mobile/Library/SpringBoard/Application Shortcu Gmail app)



- Top 150 apps in App Store's "Top Charts" (Aug. 2(
  - Push notification: 67 (44.7%)
  - dynamically registered home screen quick actions: 44 (31.3%)

Bypassing Sandbox Restrictions --- Case Studies

- Other cases: number of photos/memos

- Generic approach to detect files

Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion

Practical Issues

- App Store Vetting
  - Disguised as an *Audio Player*
  - Passed the vetting

App Store

- Power Consumption
  - Device: jailbroken iPhone 7 with iOS 10.1.1
  - 60 min: 5% battery was consumed

Practical Issues --- Cross-device Attack Feasibility

training device: Device A

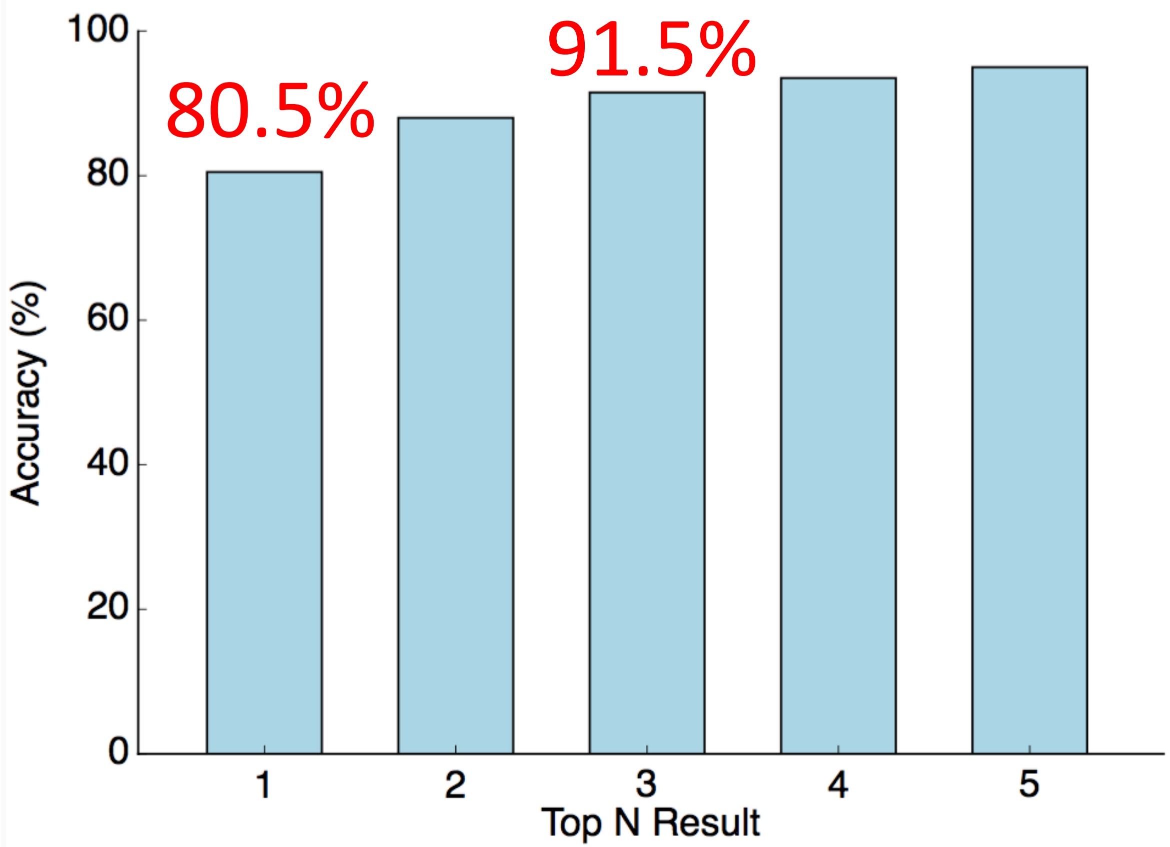
iOS 10.1.1

testing device: Device B

Non-jailbroken iOS 10.2.1

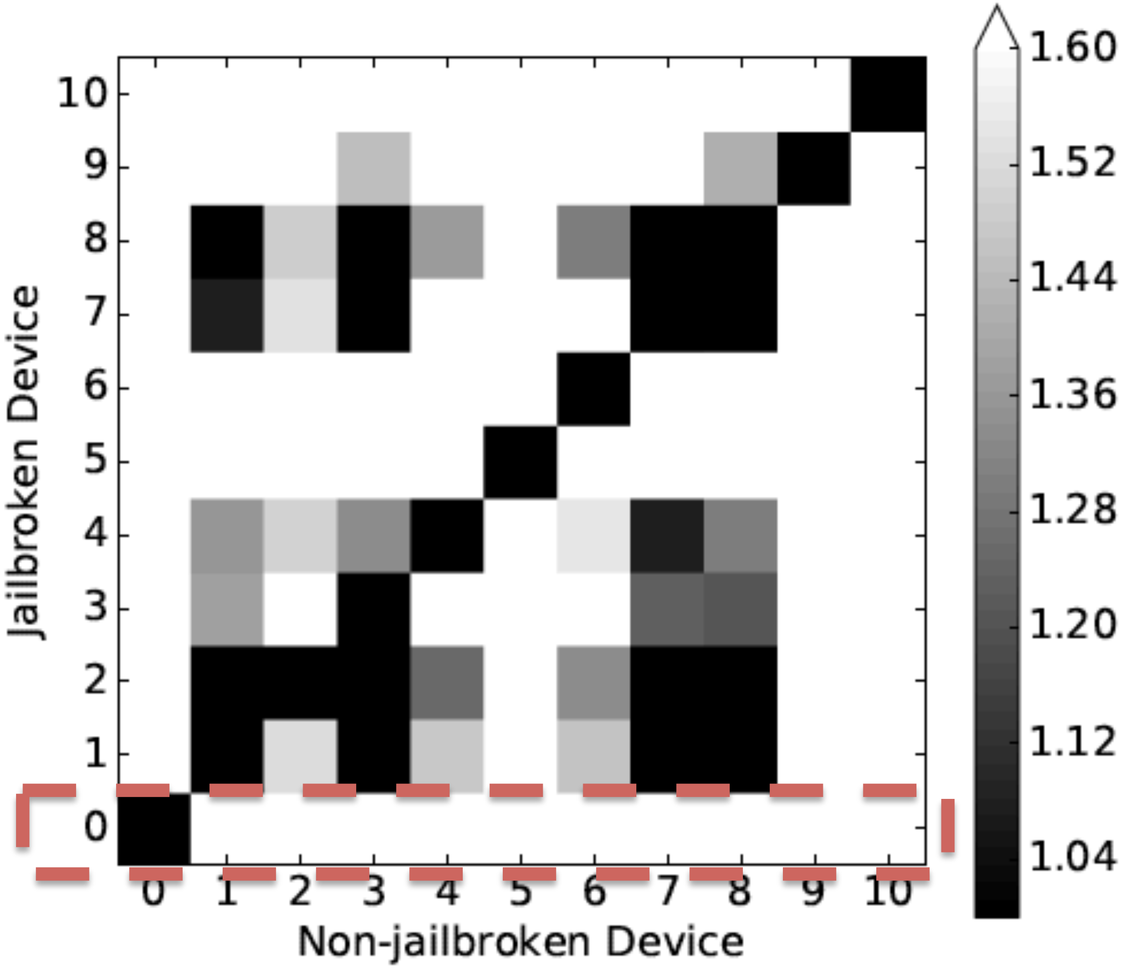
Practical Issues --- Cross-device Attack Feasibility

- Test set: Randomly select 20 third-party apps
- Redo Foreground Apps Experiment



Practical Issues --- Cross-device Attack Feasibility

- Target: *Blockchain Wallet*

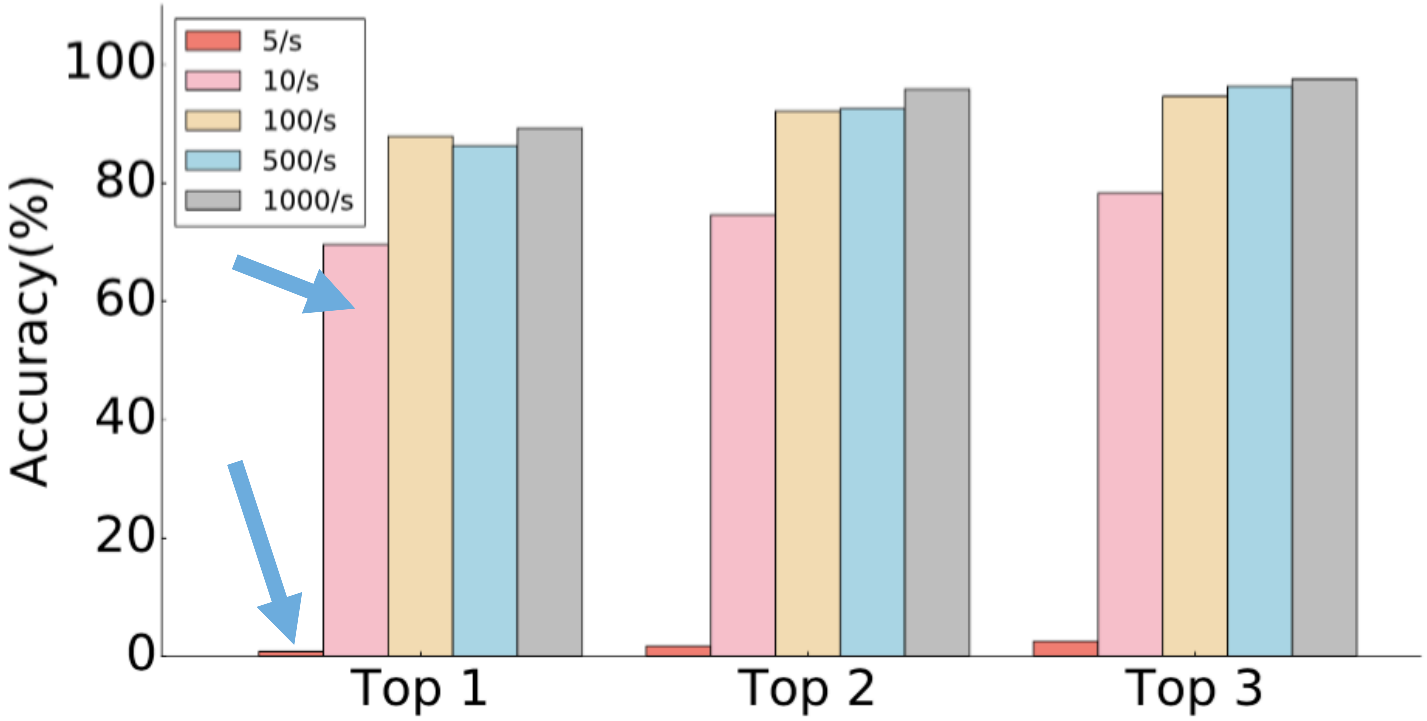


Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion

Countermeasures

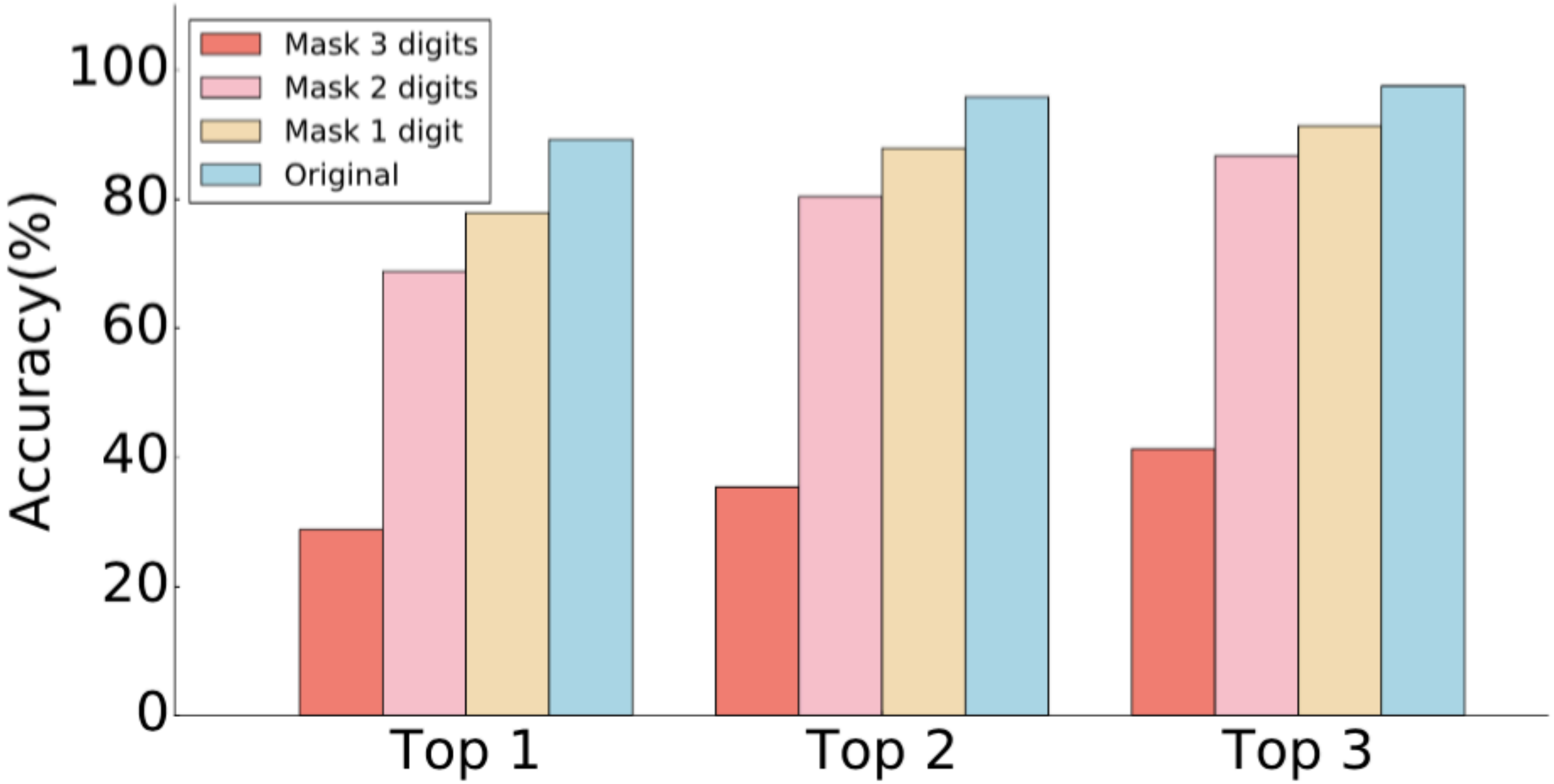
- Rate Limiting: limit the sampling rate
  - Filter the data and only keep every *(1000/N)th* data point
  - Re-evaluate the foreground app classification



Implemented in iOS 11.1 for host_statistics64(): 2/s

Countermeasures

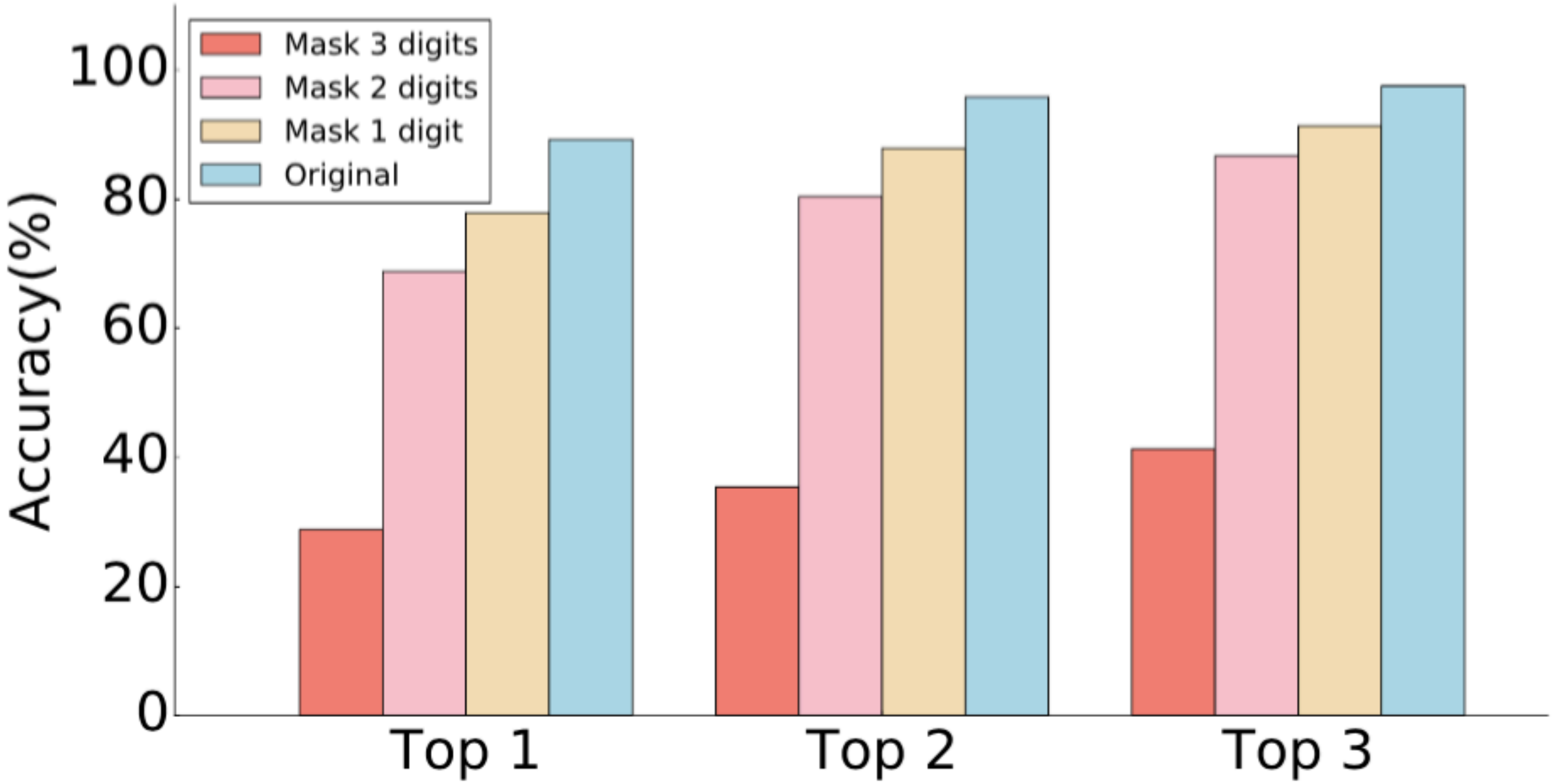
- Coarse-grained return values: masking the digits of return values
  - Mask 1/2/3 digits of all 6 features
  - Re-evaluate the foreground app classification



Original:	1234
Mask 1 digit:	1230
Mask 2 digits:	1200
Mask 3 digits:	1000

Countermeasures

- Coarse-grained return values: masking the digits of return values
  - Mask 1/2/3 digits of all 6 features
  - Re-evaluate the foreground app classification



Implemented in iOS 11 for getifaddrs(): Round to 1KB

Countermeasures

- Eliminating the attack vectors
- Runtime detection
- Privacy-preserving statistics reporting
- Removing the fileExistsAtPath timing channel

fileExistsAtPath timing channel has been eliminated in iOS 11

Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion

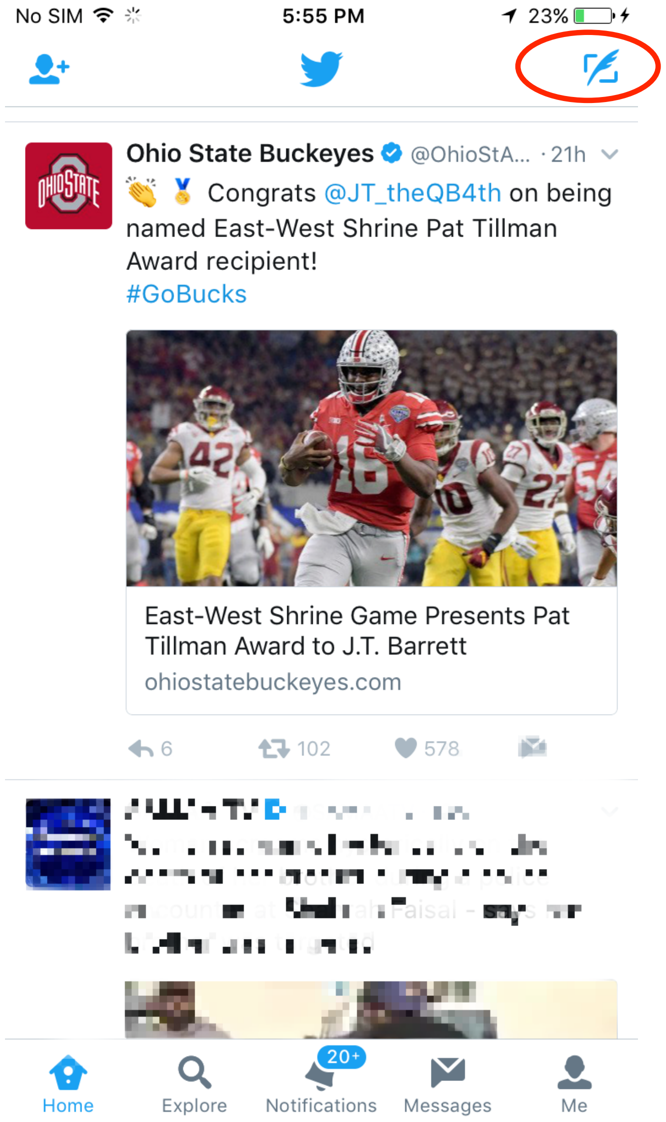
Conclusion

- First exploration of OS-level side channels on iOS
- Three categories of side-channel attacks
- Proposed countermeasures integrated in iOS and MacOS

Xiaokuan Zhang
zhang.5840@osu.edu

Detecting Sensitive In-App Activities --- Attack Methods

- Time is short (<0.5s)
- Difference is subtle



Ohio State Buckeyes @OhioStA... · 21h

👏 🏅 Congrats @JT_theQB4th on being named East-West Shrine Pat Tillman Award recipient!
#GoBucks

East-West Shrine Game Presents Pat Tillman Award to J.T. Barrett
ohiostatebuckeyes.com

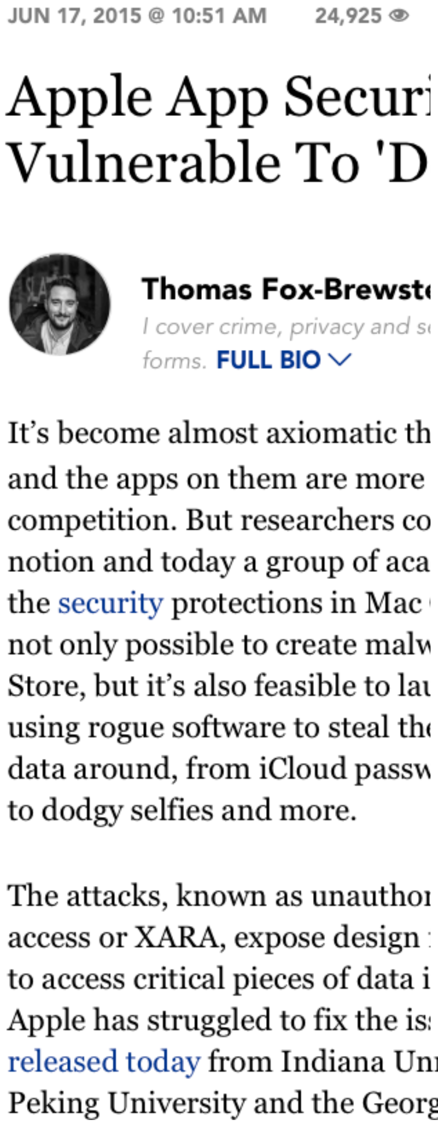
[redacted]

Detecting Sensitive In-App Activities --- Attack Methods

- Pattern Matching: compare two multi-dimensional data traces
  - Sample: $\vec{X}_t = \{\vec{X}_t^1, \vec{X}_t^2, \cdots, \vec{X}_t^l\}$, where $\vec{X}_t^i = (X_{t_1}^i, X_{t_2}^i, \cdots, X_{t_{n_i}}^i)$
  - Signature: $\vec{S}_t \quad = \quad \{\vec{S}_t^1, \vec{S}_t^2, \cdots, \vec{S}_t^l\}$
  - Goal: measure the distance $d(\vec{X}_t, \vec{S}_t)$
  - Extended DTW (DTW_I): (w_k: normalization factor)

$$d(\vec{X}_t, \vec{S}_t) = \sum_{k=1}^{l} \frac{1}{w_k} \cdot \mathbf{DTW}\left(\vec{X}_t^k, \vec{S}_t^k\right)$$

iOS Attacks



JUN 17, 2015 @ 10:51 AM 24,925

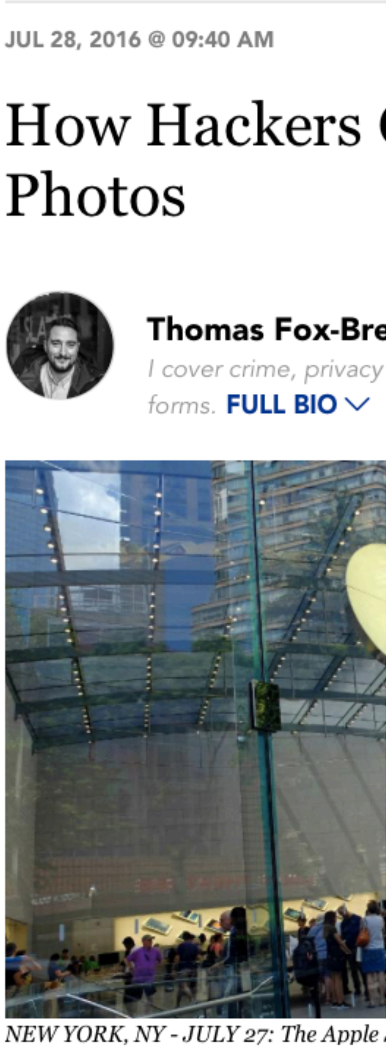
Apple App Securi
Vulnerable To 'D

Thomas Fox-Brewst

I cover crime, privacy and s
forms. **FULL BIO**

It's become almost axiomatic th and the apps on them are more competition. But researchers co notion and today a group of aca the security protections in Mac not only possible to create malw Store, but it's also feasible to lau using rogue software to steal th data around, from iCloud passw to dodgy selfies and more.

The attacks, known as unauthor access or XARA, expose design to access critical pieces of data i Apple has struggled to fix the is released today from Indiana University Bloomington, Peking University and the Georgia Institute of Technology.



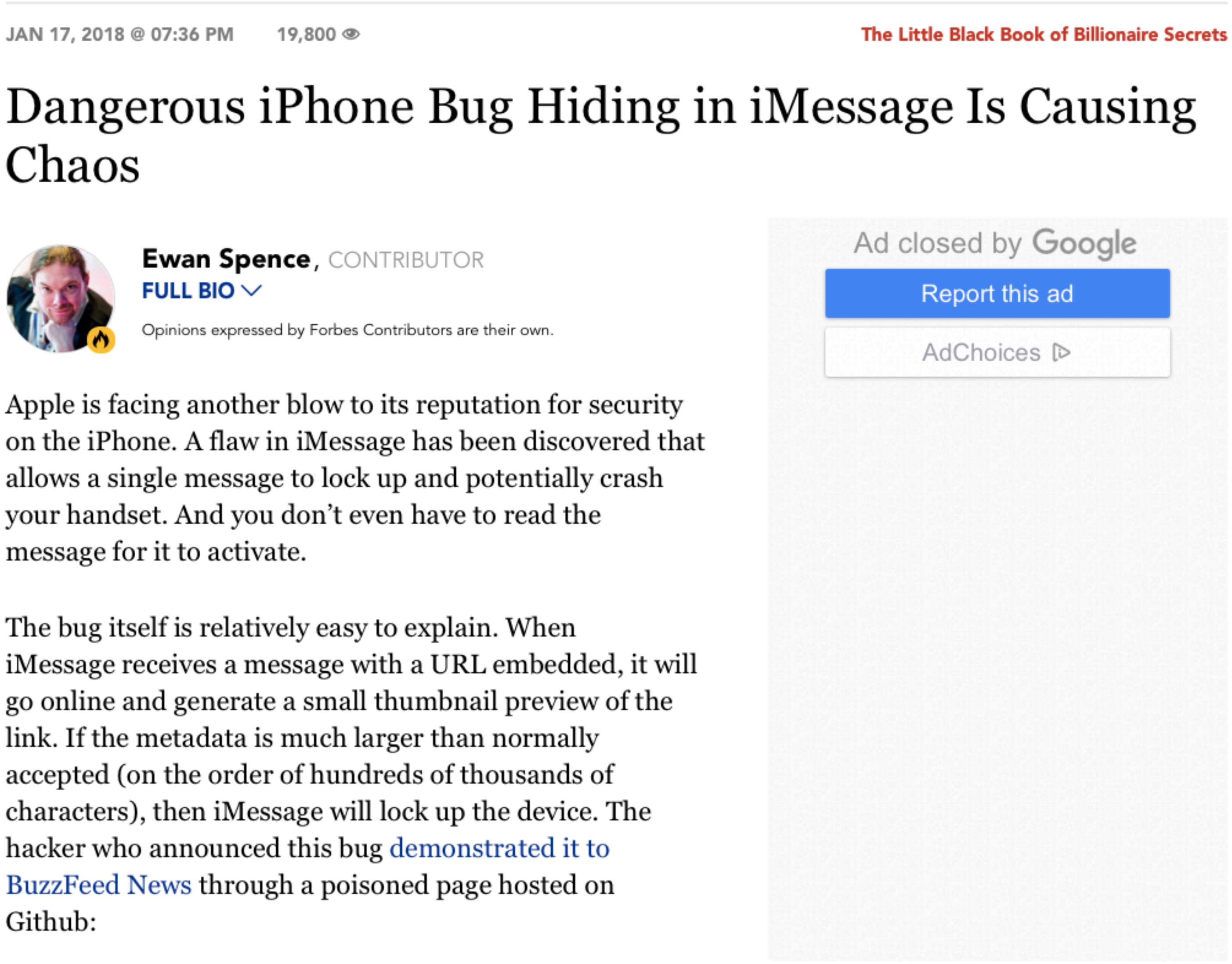
JUL 28, 2016 @ 09:40 AM

How Hackers
Photos

Thomas Fox-Bre

I cover crime, privacy
forms. **FULL BIO**

NEW YORK, NY - JULY 27: The Apple



JAN 17, 2018 @ 07:36 PM 19,800

The Little Black Book of Billionaire Secrets

Dangerous iPhone Bug Hiding in iMessage Is Causing Chaos

Ewan Spence, CONTRIBUTOR
FULL BIO

Opinions expressed by Forbes Contributors are their own.

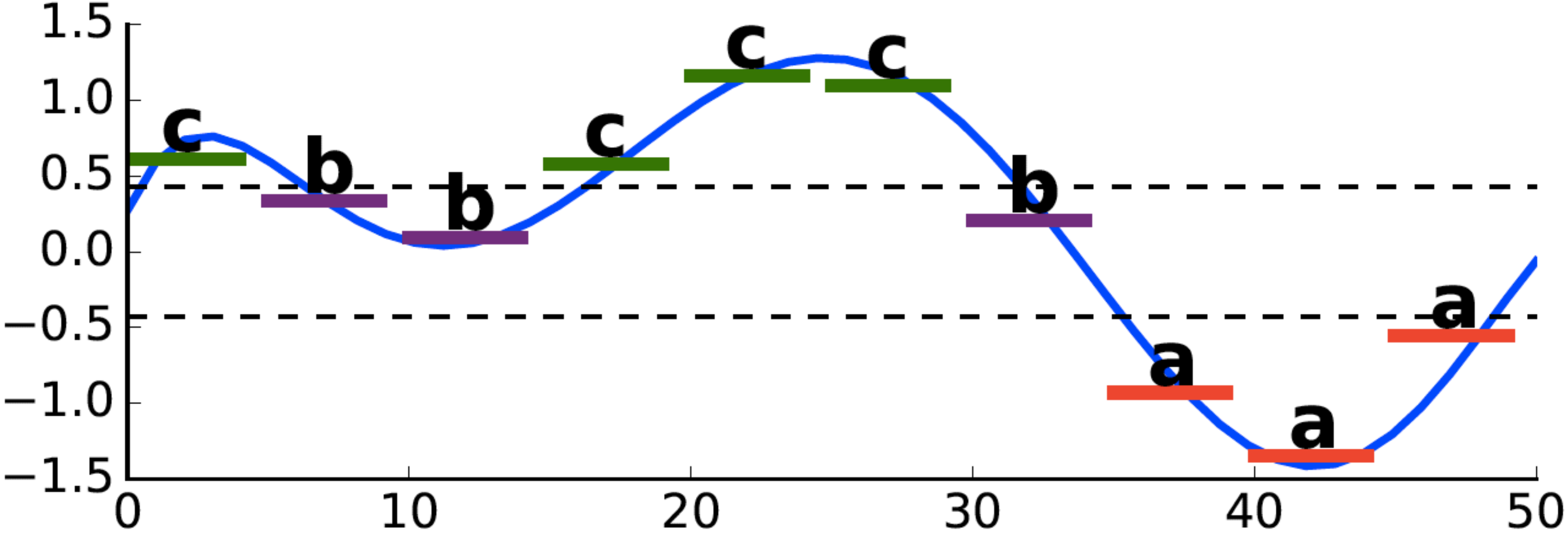
Apple is facing another blow to its reputation for security on the iPhone. A flaw in iMessage has been discovered that allows a single message to lock up and potentially crash your handset. And you don't even have to read the message for it to activate.

The bug itself is relatively easy to explain. When iMessage receives a message with a URL embedded, it will go online and generate a small thumbnail preview of the link. If the metadata is much larger than normally accepted (on the order of hundreds of thousands of characters), then iMessage will lock up the device. The hacker who announced this bug demonstrated it to BuzzFeed News through a poisoned page hosted on Github:

Paper	Vector	Impact
Chen et al., Security’14	/proc/pid/ statm	UI inference attacks (stealing login credentials, photos)
Diao et al., Oakland’16	/proc/ interrupts	Interrupt timing analysis (cracking unlock patterns)

Classifying User Activities --- Attack Methods

- Requirements:
  - Combining
  - Reducing t
- Major comp
  - Symbolic
  - Bag-of-Pa
  - Support Vector Machine (LibSVM) (Chang et al., 2011)



{cbb:1, bbc:1, bcc:1, ccc:1,
ccb:1, cba:1, baa:1, aaa:1}

Classifying User Activities --- Case Studies

- Top N Accuracy Example

Sample	True Class	SVM Prediction (Probability Model)		
A	1	4	2	1
B	2	2	5	4
C	3	3	1	2
D	4	1	4	2
E	5	5	2	4

Classifying User Activities --- Case Studies

- Top N Accuracy Example

Sample	True Class	SVM Prediction (Probability Model)		
A	1	4	2	1
B	2	2	5	4
C	3	3	1	2
D	4	1	4	2
E	5	5	2	4

Classifying User Activities --- Case Studies

- Top N Accuracy Example

Sample	True Class	SVM Prediction (Probability Model)		
A	1	4	2	1
B	2	2	5	4
C	3	3	1	2
D	4	1	4	2
E	5	5	2	4

Top 1 Accuracy: 3/5 = 60%

Classifying User Activities --- Case Studies

- Top N Accuracy Example

Sample	True Class	SVM Prediction (Probability Model)		
A	1	4	2	1
B	2	2	5	4
C	3	3	1	2
D	4	1	4	2
E	5	5	2	4

Classifying User Activities --- Case Studies

- Top N Accuracy Example

Sample	True Class	SVM Prediction (Probability Model)		
A	1	4	2	1
B	2	2	5	4
C	3	3	1	2
D	4	1	4	2
E	5	5	2	4

Top 2 Accuracy: (3+1)/5 = 80%

Classifying User Activities --- Case Studies

- Top N Accuracy Example

Sample	True Class	SVM Prediction (Probability Model)		
A	1	4	2	1
B	2	2	5	4
C	3	3	1	2
D	4	1	4	2
E	5	5	2	4

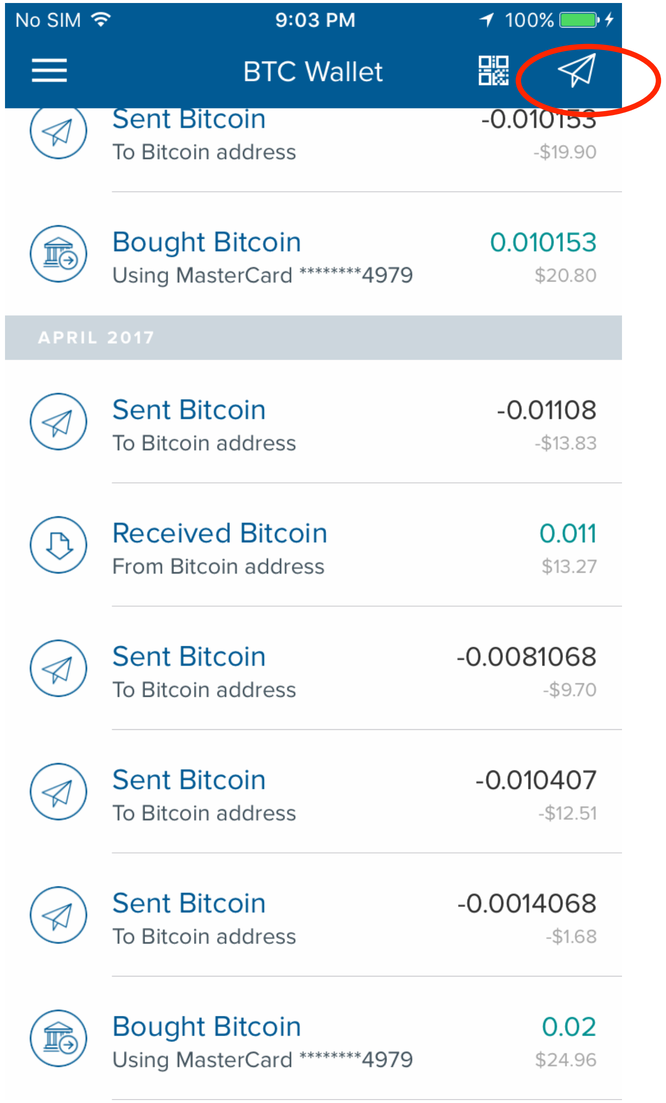
Classifying User Activities --- Case Studies

- Top N Accuracy Example

Sample	True Class	SVM Prediction (Probability Model)		
A	1	4	2	1
B	2	2	5	4
C	3	3	1	2
D	4	1	4	2
E	5	5	2	4

Top 3 Accuracy: (2+1+2)/5 = 100%

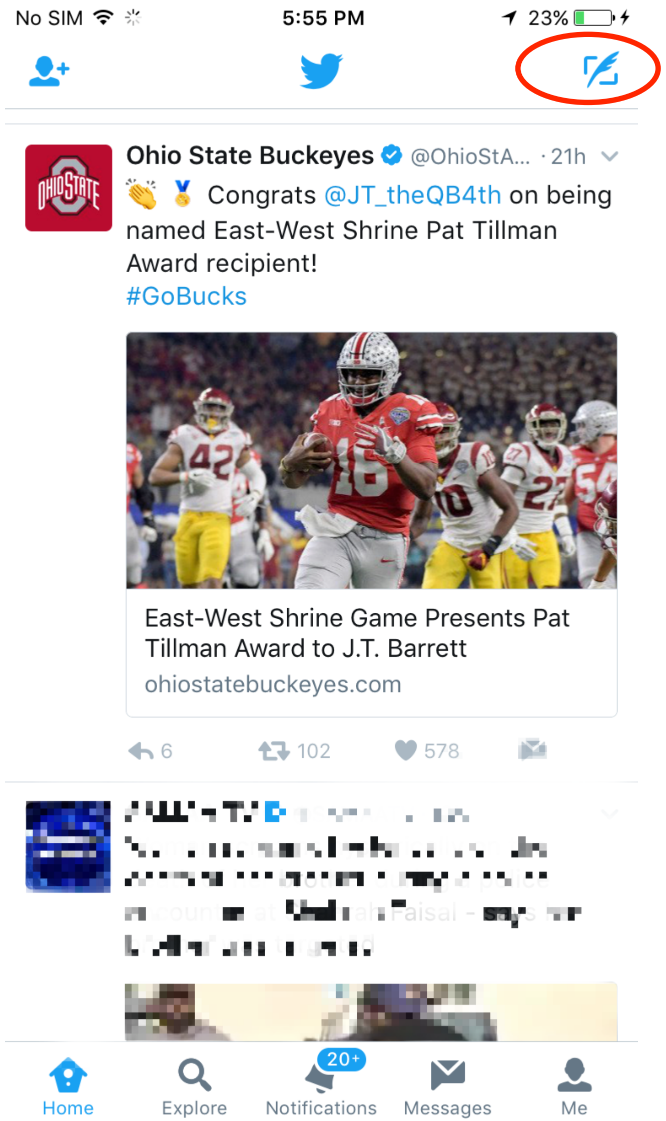
Detecting Sensitive In-App Activities



No SIM 9:03 PM 100%

BTC Wallet

Sent Bitcoin To Bitcoin address	-0.010153 -$19.90
Bought Bitcoin Using MasterCard ********4979	0.010153 $20.80
APRIL 2017	
Sent Bitcoin To Bitcoin address	-0.01108 -$13.83
Received Bitcoin From Bitcoin address	0.011 $13.27
Sent Bitcoin To Bitcoin address	-0.0081068 -$9.70
Sent Bitcoin To Bitcoin address	-0.010407 -$12.51
Sent Bitcoin To Bitcoin address	-0.0014068 -$1.68
Bought Bitcoin Using MasterCard ********4979	0.02 $24.96



No SIM 5:55 PM 23%

Ohio State Buckeyes @OhioStA... · 21h

👏 🏅 Congrats @JT_theQB4th on being named East-West Shrine Pat Tillman Award recipient! #GoBucks

East-West Shrine Game Presents Pat Tillman Award to J.T. Barrett
ohiostatebuckeyes.com

6 102 578

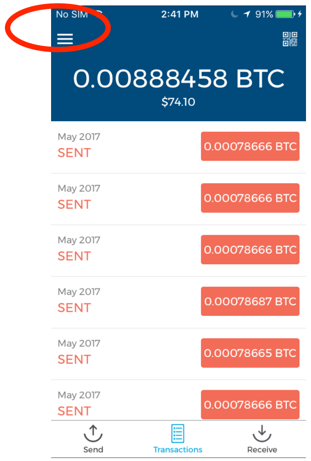
[redacted]

Detecting Sensitive In-App Activities --- Attack Methods

- Identify critical events

- Correlates with public records

Detecting Sensitive In-App Activities

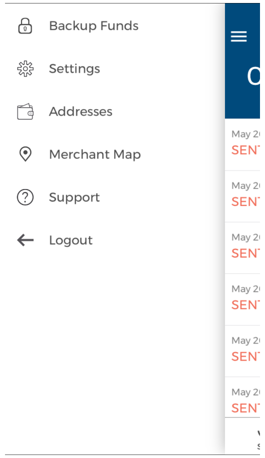


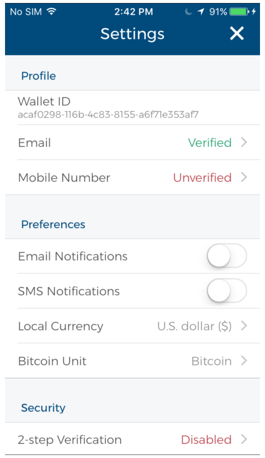
No SIM 2:41 PM 91%

0.00888458 BTC

$74.10

May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078687 BTC
May 2017 SENT	0.00078665 BTC
May 2017 SENT	0.00078666 BTC





No SIM 2:42 PM 91%

Settings

Profile

Wallet ID
acaf0298-116b-4c83-8155-a6f71e353af7

Email Verified

Mobile Number Unverified

Preferences

Email Notifications ○

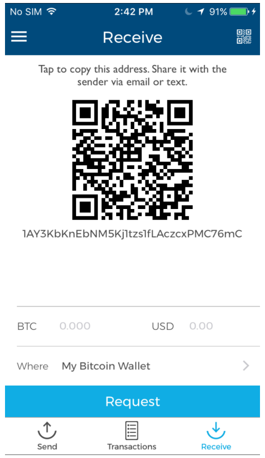
SMS Notifications ○

Local Currency U.S. dollar ($)

Bitcoin Unit Bitcoin

Security

2-step Verification Disabled



No SIM 2:42 PM 91%

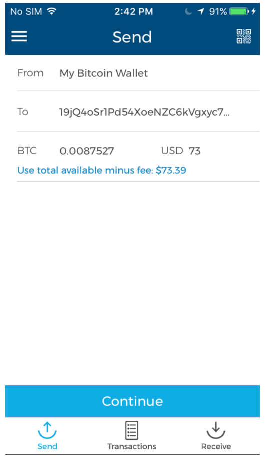
Receive

Tap to copy this address. Share it with the sender via email or text.

1AY3KbKnEbNM5Kj1tzs1fLAczcxPMC76mC

BTC 0.000 USD 0.00

Where My Bitcoin Wallet



No SIM 2:42 PM 91%

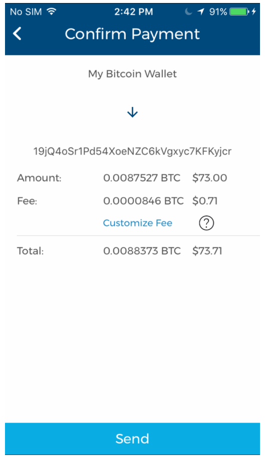
Send

From My Bitcoin Wallet

To 19jQ4oSr1Pd54XoeNZC6kVgxyc7...

BTC 0.0087527 USD 73

Use total available minus fee: $73.39



No SIM 2:42 PM 91%

Confirm Payment

My Bitcoin Wallet

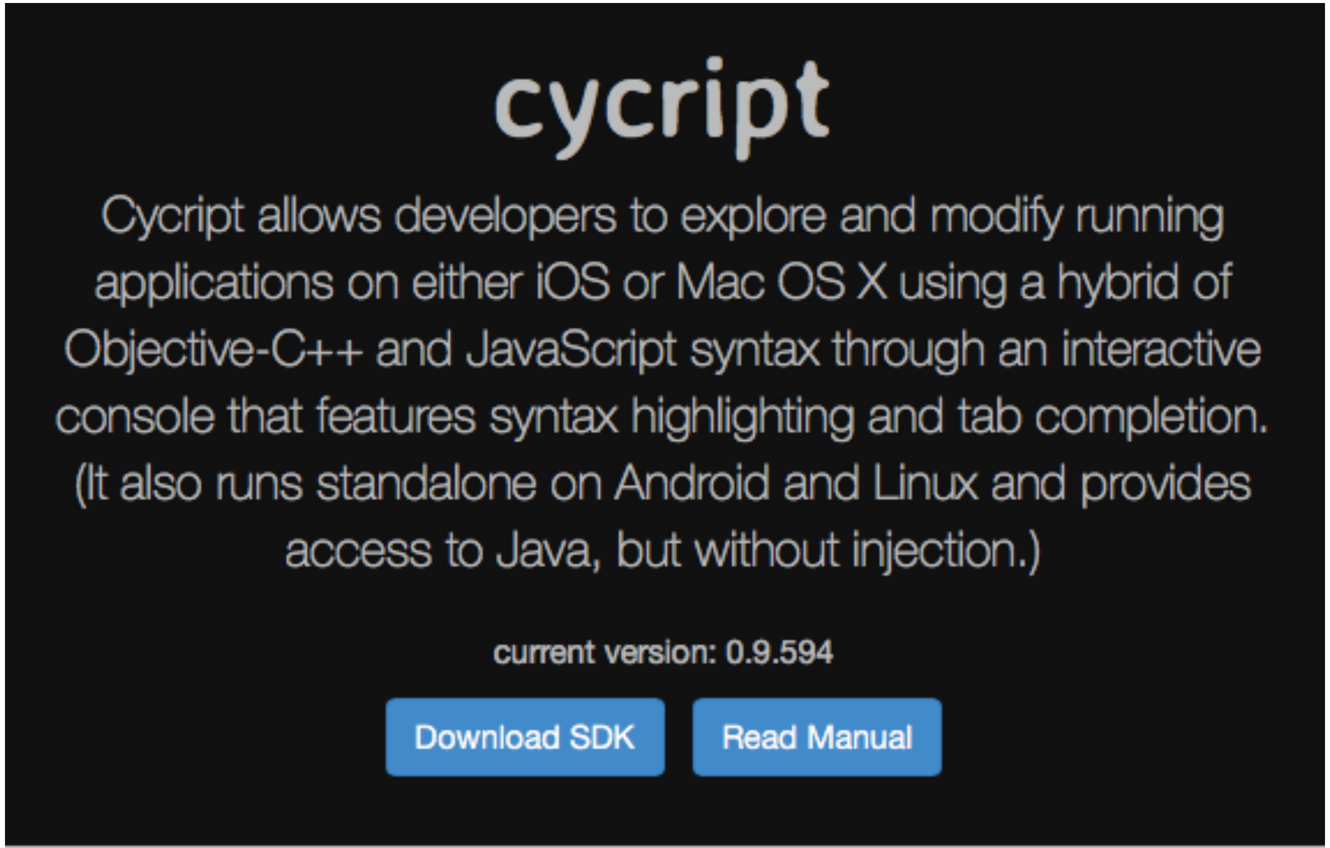
↓

19jQ4oSr1Pd54XoeNZC6kVgxyc7KFKyjcr

Amount:	0.0087527 BTC	$73.00
Fee:	0.0000846 BTC	$0.71
	Customize Fee	?
Total:	0.0088373 BTC	$73.71

Classifying User Activities --- Case Studies

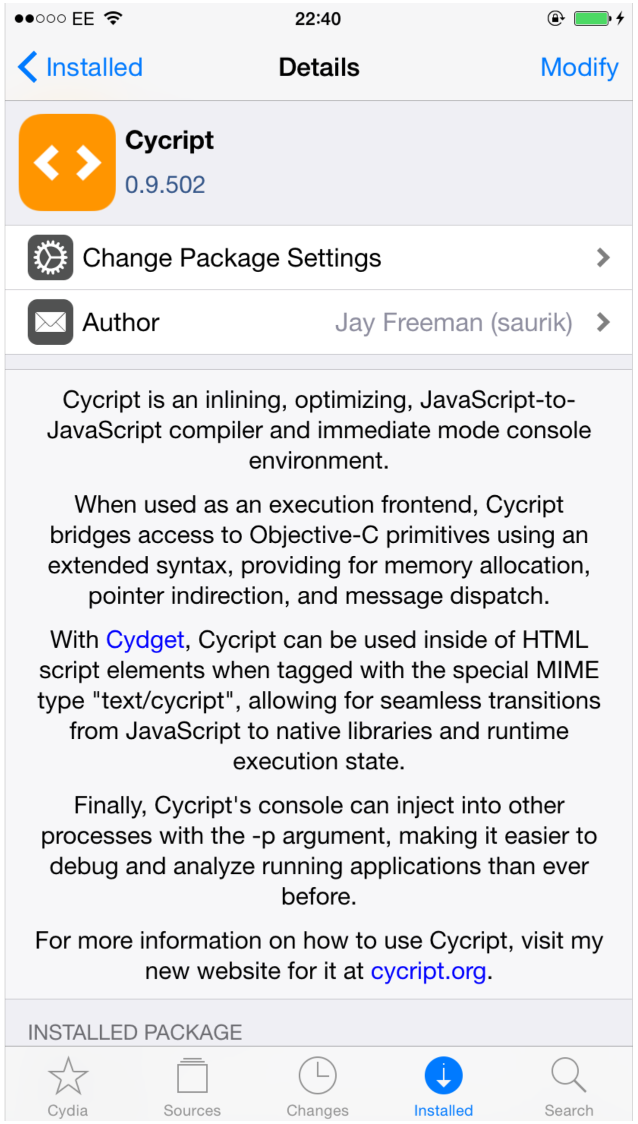
- Device: jailbroken iPhone 7 with iOS 10.1.1
- Automated using Cycript



cycript

Cycript allows developers to explore and modify running applications on either iOS or Mac OS X using a hybrid of Objective-C++ and JavaScript syntax through an interactive console that features syntax highlighting and tab completion. (It also runs standalone on Android and Linux and provides access to Java, but without injection.)

current version: 0.9.594



Cycript
0.9.502

Change Package Settings

Author — Jay Freeman (saurik)

Cycript is an inlining, optimizing, JavaScript-to-JavaScript compiler and immediate mode console environment.

When used as an execution frontend, Cycript bridges access to Objective-C primitives using an extended syntax, providing for memory allocation, pointer indirection, and message dispatch.

With Cydget, Cycript can be used inside of HTML script elements when tagged with the special MIME type "text/cycript", allowing for seamless transitions from JavaScript to native libraries and runtime execution state.

Finally, Cycript's console can inject into other processes with the -p argument, making it easier to debug and analyze running applications than ever before.

For more information on how to use Cycript, visit my new website for it at cycript.org.

INSTALLED PACKAGE

Why global stat can work?

- iOS itself suspends apps when they run in the background, unless the app specially requests background permissions

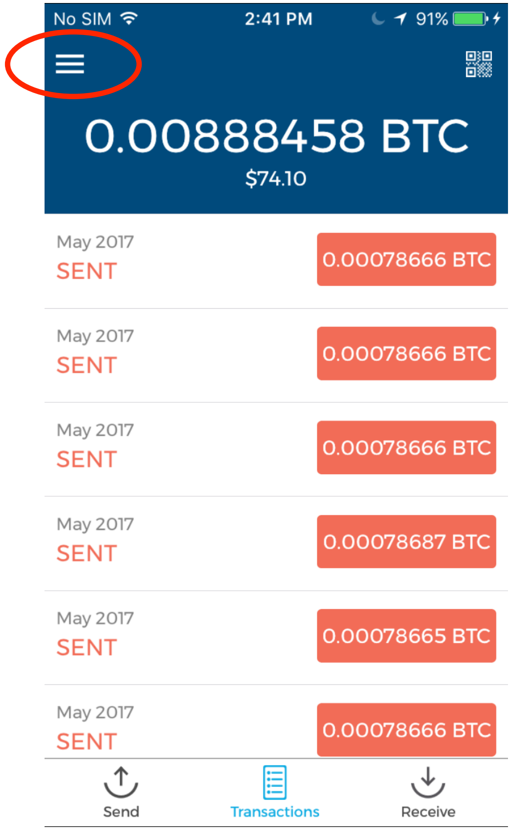
- iOS is relatively quieter than Android, which greatly facilitates side-channel attacks

Run Background Apps on iOS

- *AUDIO* background mode

- [NSTimer scheduledTimerWithTimeInterval: target: selector: userInfo: repeats:]

Detecting Sensitive In-App Activities

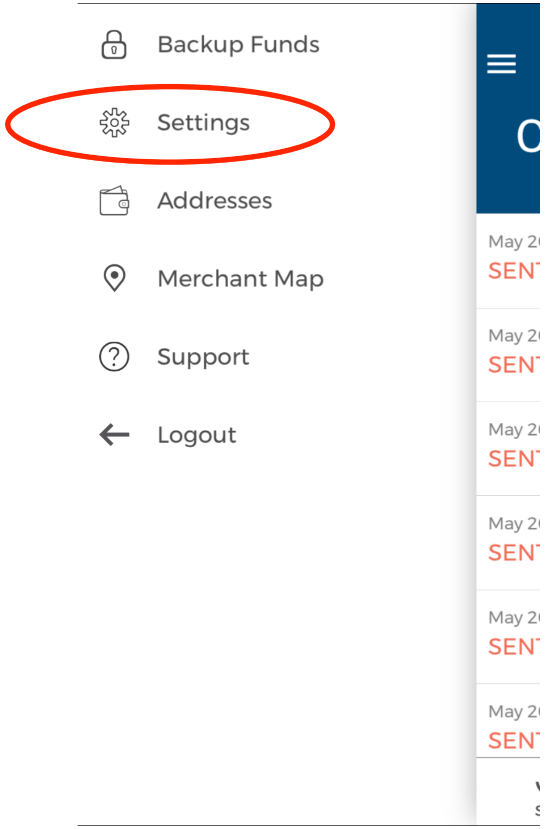


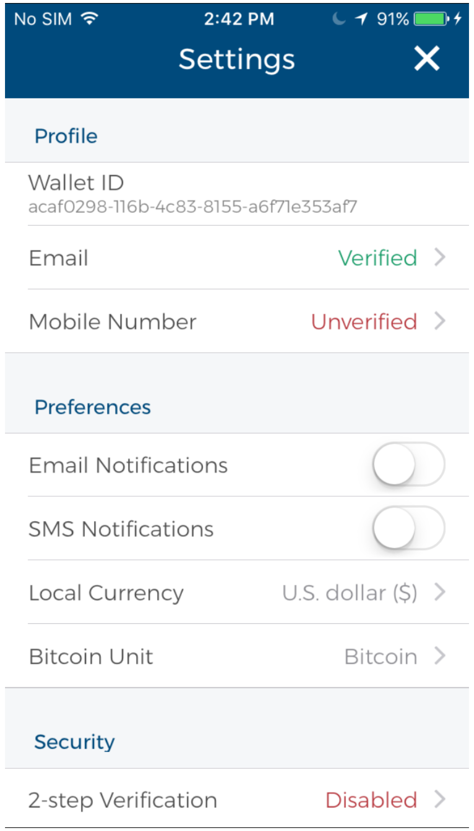
No SIM 2:41 PM 91%

0.00888458 BTC

$74.10

May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078666 BTC
May 2017 SENT	0.00078687 BTC
May 2017 SENT	0.00078665 BTC
May 2017 SENT	0.00078666 BTC





No SIM 2:42 PM 91%

Settings

Profile

Wallet ID
acaf0298-116b-4c83-8155-a6f71e353af7

Email Verified

Mobile Number Unverified

Preferences

Email Notifications ○

SMS Notifications ○

Local Currency U.S. dollar ($)

Bitcoin Unit Bitcoin

Security

2-step Verification Disabled