THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

FOAP: Fine-Grained Open-World Android App Fingerprinting

Jianfeng Li[1], Hao Zhou[1], Shuohan Wu[1], Xiapu Luo[1], Ting Wang[2], Xian Zhan[1], Xiaobo Ma[3]

[1]The Hong Kong Polytechnic University

[2]Pennsylvania State University

[3]Xi'an Jiaotong University

Opening Minds • Shaping the Future
啟迪思維 • 成就未來

Outline

Introduction

Design of FOAP

Evaluation

Introduction

Mobile apps reshape modern lifestyles from information retrieval to instant messaging, and from shopping to entertainment

Despite the adoption of encrypted communication, mobile apps are susceptible to app fingerprinting

App fingerprinting (AF) is a traffic analysis technique

- Inferring user activity using features associated with *packet timing*, *packet direction*, and *packet size*
- No need to inspect packet plaintext payload

Network administrator

- *Censorship*
- *Malware detection*
- *User profiling*

Attacker

- *Reconnaissance attack*
- *User privacy inference*

Introduction

Limitation of existing AF

- Closed-world assumption: cannot handle apps unseen during model training
- Coarse-grained identification: recognizing apps, identifying selected user actions that need manual labeling

Our goal

- *Goal 1*: open-world app fingerprinting

  Challenge: risk of false positives

- *Goal 2*: fine-grained user action identification

  Challenge: automatic user action labeling

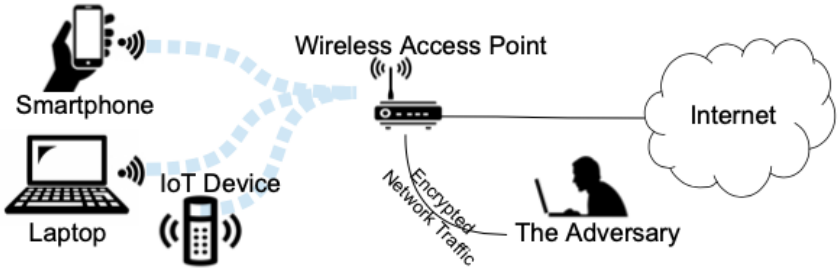
Threat model

The adversary sniffs network traffic on a wireless access point

The adversary *cannot* exploit

i) packet plaintext payload

ii) destination feature of network flows



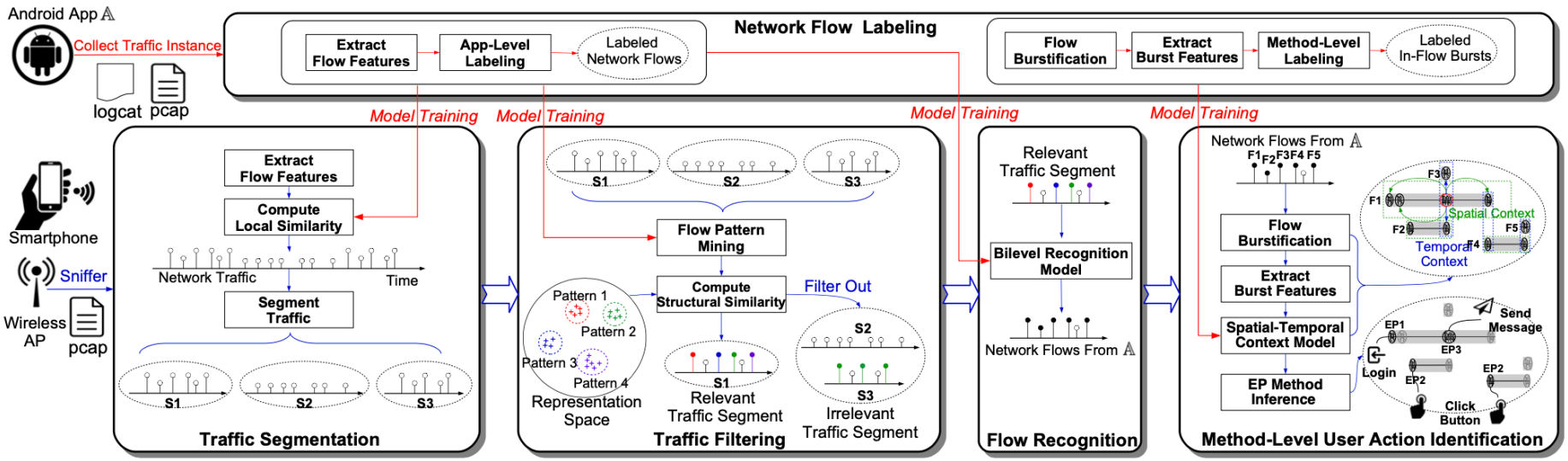
Outline

Introduction

Design of FOAP

Evaluation

System Workflow

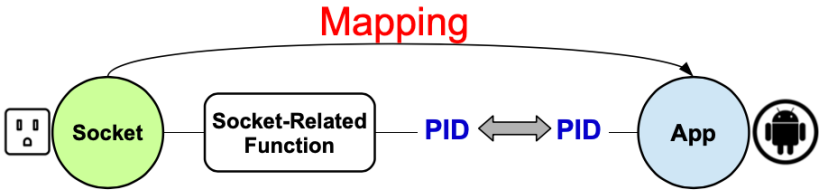


- Training data construction: network traffic labeling
- Open-world app traffic recognition: traffic segmentation, traffic filtering, flow recognition
- Fine-grained app fingerprinting: method-level user action identification

Open-World App Traffic Recognition

App-level labeling

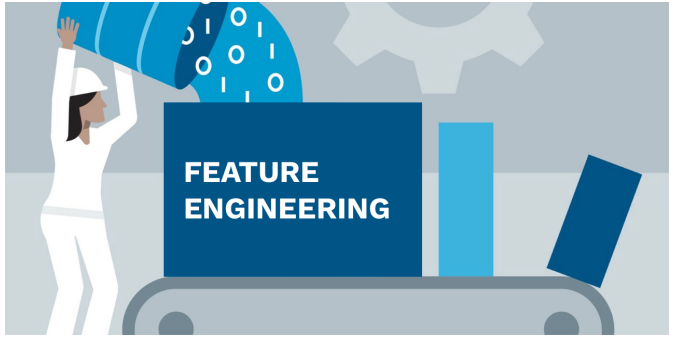
Constructing coarse-grained training data



Extract flow features

123-dimentional feature vector for each flow

- *General characteristic (8 features)*
- *Interactive pattern (20 features)*
- *Packet rate characteristic (5 features)*
- *Temporal characteristic (39 features)*
- *Packet size characteristic (51 features)*



Open-World App Traffic Recognition

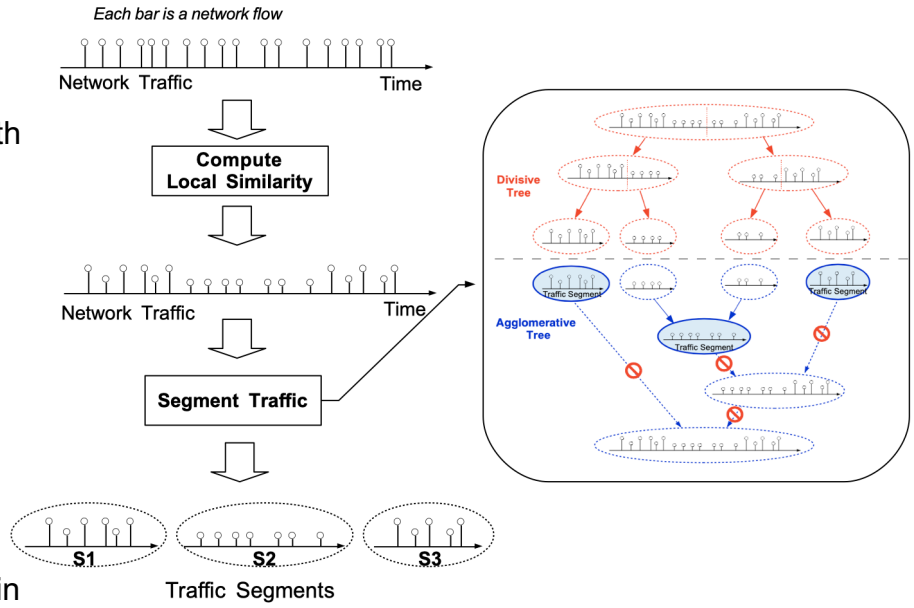
Local similarity (LS): a metric to quantify how a network flow is similar to network flows from the app of interest, say $\mathcal{A}$, in terms of the intrinsic features from itself

Observation:

- Network flows generated by $\mathcal{A}$ have relatively higher LS with $\mathcal{A}$ compared with network flows generated by other apps
- There is obvious difference of LS between time periods dominated by $\mathcal{A}$ and those dominated by other apps

Traffic segmentation

- To locate all possible time periods when $\mathcal{A}$ is active, we conduct traffic segmentation in an unsupervised manner
- Formulate traffic segmentation as a combinational optimization problem
- Propose a divisive-agglomerative tree algorithm to solve it in a greedy fashion



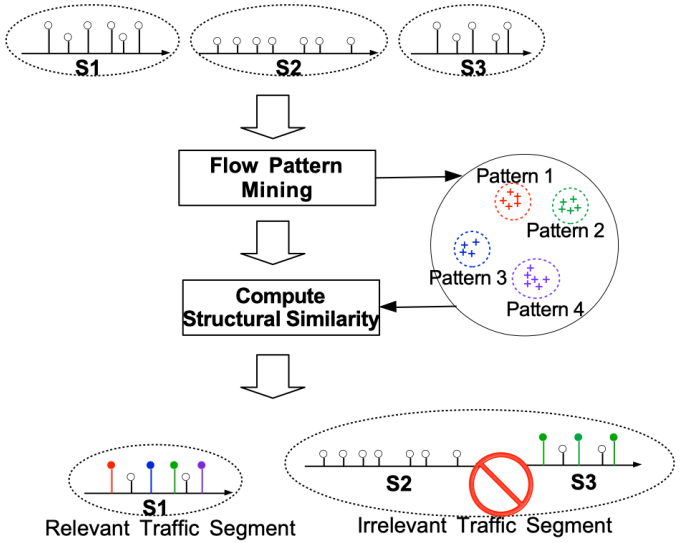
Open-World App Traffic Recognition

Traffic filtering

- If the app of interest is active during a traffic segment, we refer to it as a *relevant traffic segment* and otherwise an *irrelevant traffic segment*.
- Our goal is filtering out all irrelevant traffic segments to effectively reduce risk of false positives

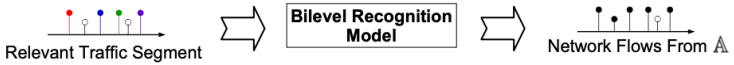
Structural similarity (SS): a metric to characterizes how network flows within a traffic segment are similar to network flows from the app of interest in various patterns

- Mining flow patterns based on metric learning and clustering analysis
- Filtering out irrelevant traffic segments by computing each traffic segment's *SS with the app of interest*



Flow recognition

- Constructing a bilevel recognition model to recognize all network flows generated by the app of interest from relevant traffic segments

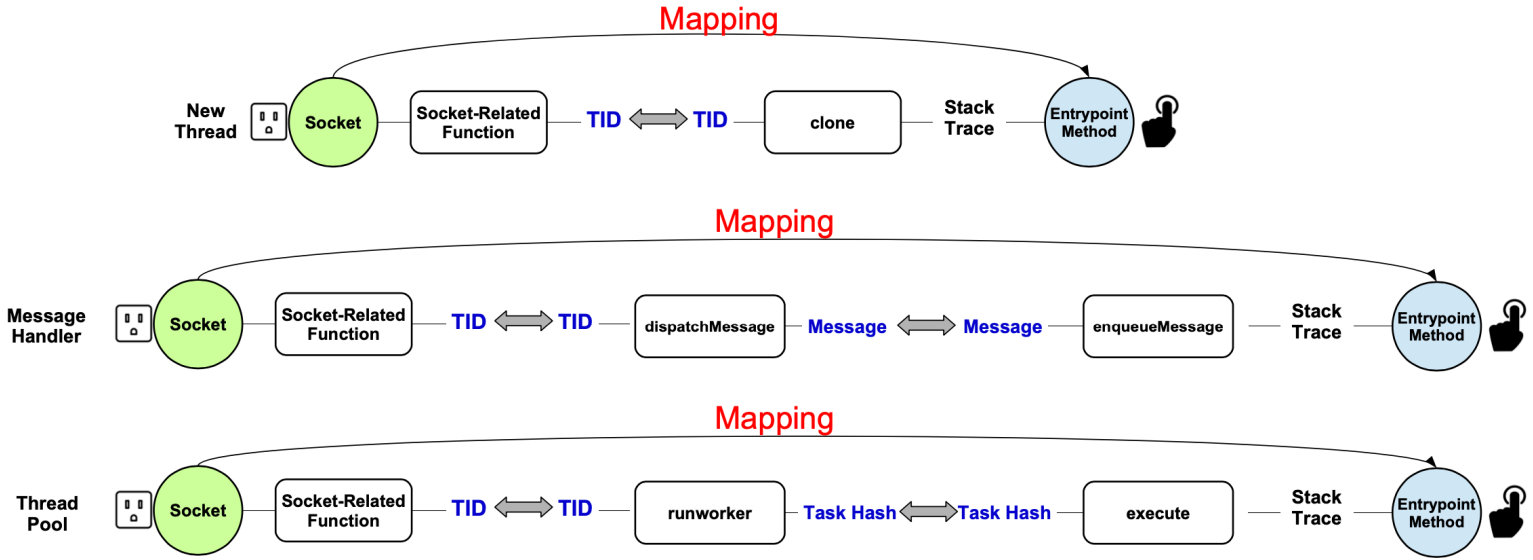


Fine-Grained App Fingerprinting

Method-level labeling

Constructing fine-grained training data

Entrypoint (EP) method-- the component lifecycle or the UI callback



Fine-Grained App Fingerprinting

Method-Level User Action Identification

Flow Burstification

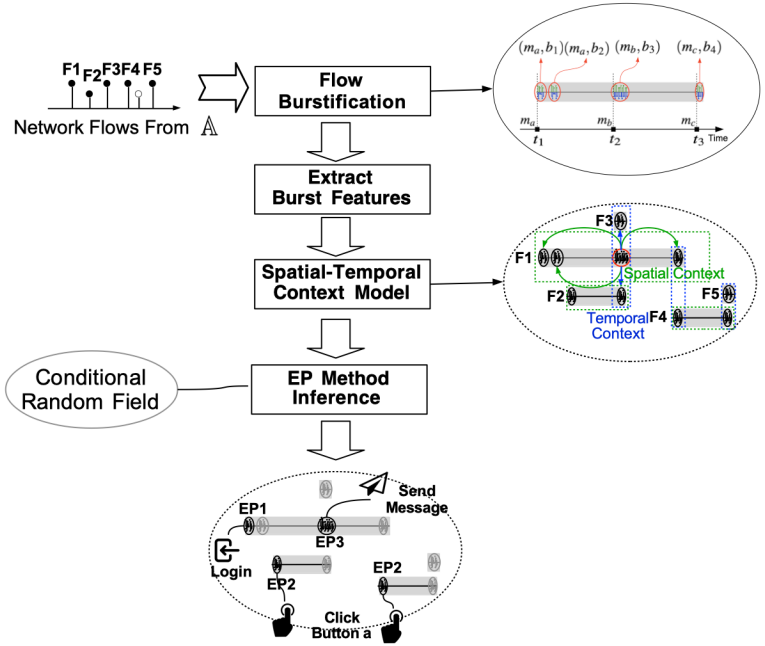
- *Persistent connection*: containing multiple packet bursts associated with sequentially invoked EP methods
- *Short connection*: containing only one packet burst associated with a single EP method

Spatial-Temporal Context Model

- *Spatial Context*: It reflects the contextual relationship between bursts within the same network flow
- *Temporal Context:* An EP method may trigger multiple network flows simultaneously, yielding concurrent in-flow bursts

Iterative Inference of EP Method

- Formulate contextual dependency of in-flow bursts based on conditional random field
- Identify EP Method based on a greedy inference algorithm



Outline

Introduction

Design of FOAP

Evaluation

Evaluation

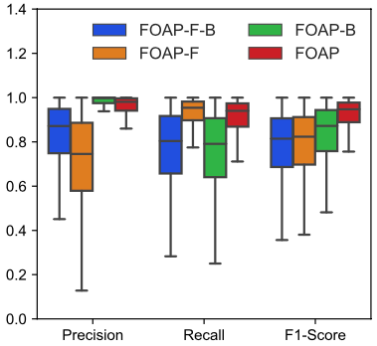
- Open-World App Traffic Recognition

Dataset: 1000 popular apps from Google Play Store

Evaluating FOAP in open-world app traffic recognition (mean $\pm$ standard deviation)

n_T	AppScanner			AppScanner (extended)			FOAP (our approach)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
5	0.315 ± 0.171	0.866 ± 0.105	0.439 ± 0.174	0.307 ± 0.158	0.840 ± 0.111	0.427 ± 0.166	0.759 ± 0.253	$\mathbf{0.925} \pm 0.094$	0.803 ± 0.200
10	0.476 ± 0.183	0.814 ± 0.127	0.582 ± 0.161	0.470 ± 0.178	0.774 ± 0.128	0.568 ± 0.159	0.886 ± 0.170	0.906 ± 0.111	0.881 ± 0.140
15	0.568 ± 0.189	0.782 ± 0.137	0.641 ± 0.158	0.575 ± 0.182	0.729 ± 0.144	0.627 ± 0.155	0.926 ± 0.123	0.897 ± 0.114	0.902 ± 0.111
20	0.641 ± 0.180	0.755 ± 0.146	0.679 ± 0.151	0.646 ± 0.164	0.700 ± 0.148	0.659 ± 0.143	$\mathbf{0.945} \pm 0.102$	0.897 ± 0.126	$\mathbf{0.911} \pm 0.115$

Effect of different modules of FOAP



False positive rate for open-world network traffic recognition (lower is better)

Open-World Traffic	AppScanner	AppScanner (extended)	FOAP (our method)
Mobile Web	2.41×10^{-2}	2.18×10^{-2}	$\mathbf{9.93 \times 10^{-4}}$
IoT	3.39×10^{-2}	2.85×10^{-2}	$\mathbf{4.36 \times 10^{-5}}$
PC	1.21×10^{-2}	2.02×10^{-2}	$\mathbf{3.54 \times 10^{-4}}$
SC-App	3.24×10^{-2}	3.09×10^{-2}	$\mathbf{1.45 \times 10^{-2}}$
DC-App	2.12×10^{-2}	1.99×10^{-2}	$\mathbf{2.64 \times 10^{-3}}$

Evaluation

- ## EP Method Identification

Evaluating FOAP in EP method identification (mean $\pm$ standard deviation)

ε	Precision	Recall	F1-Score
0.5	0.888 ± 0.102	$\mathbf{0.883} \pm 0.105$	$\mathbf{0.885} \pm 0.101$
1	0.886 ± 0.104	0.868 ± 0.114	0.876 ± 0.106
2	0.885 ± 0.111	0.853 ± 0.126	0.866 ± 0.114
5	$\mathbf{0.892} \pm 0.105$	0.828 ± 0.137	0.856 ± 0.116

- ## Cross-Dataset Evaluation

Evaluating FOAP in cross-dataset experimental settings (mean $\pm$ standard deviation)

Transfer Setting	AppScanner			AppScanner (extended)			FOAP (our approach)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
M $\rightarrow$ M	0.625 ± 0.167	0.775 ± 0.120	0.681 ± 0.135	0.634 ± 0.148	0.700 ± 0.136	0.655 ± 0.127	$\mathbf{0.947} \pm 0.074$	$\mathbf{0.886} \pm 0.157$	$\mathbf{0.906} \pm 0.114$
M $\rightarrow$ H	0.506 ± 0.222	0.521 ± 0.195	0.481 ± 0.185	0.436 ± 0.212	0.508 ± 0.201	0.439 ± 0.181	$\mathbf{0.905} \pm 0.163$	$\mathbf{0.760} \pm 0.219$	$\mathbf{0.802} \pm 0.194$
EM $\rightarrow$ H	0.514 ± 0.213	0.649 ± 0.162	0.547 ± 0.180	0.470 ± 0.199	0.656 ± 0.173	0.525 ± 0.179	$\mathbf{0.929} \pm 0.129$	$\mathbf{0.873} \pm 0.169$	$\mathbf{0.882} \pm 0.150$

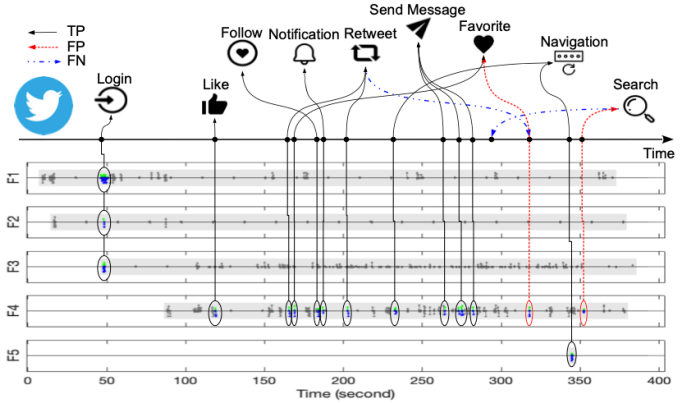
Evaluating EP method identification in cross-dataset experimental settings (mean $\pm$ standard deviation)

Transfer Setting	Precision	Recall	F1-Score
M $\rightarrow$ M	0.879 ± 0.098	0.872 ± 0.093	0.874 ± 0.093
M $\rightarrow$ H	0.771 ± 0.143	0.801 ± 0.146	0.783 ± 0.139
EM $\rightarrow$ H	0.830 ± 0.145	0.834 ± 0.150	0.831 ± 0.143

Evaluation

- Fine-Grained User Activity Inference

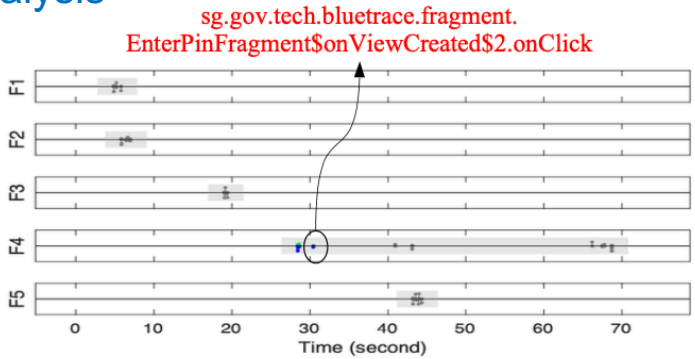
An example of inferring Twitter user activities



EP methods for inferring Twitter user activities

User Action	EP Method
Login	com.twitter.android.p1.onClick
Like	com.twitter.ui.tweet.inlineactions.InlineActionView$b.onAnimationEnd
Favorite	com.twitter.android.va$k.onClick
Retweet	xbb.onClick
Follow	com.twitter.app.profiles.y0.onClick
Notification	tcb.onClick
Send Message	com.twitter.app.dm.widget.DMConversationMessageComposer.onClick
Navigation	com.google.android.material.tabs.TabLayout$i.performClick
Search	km4.onClick

- User Privacy Analysis

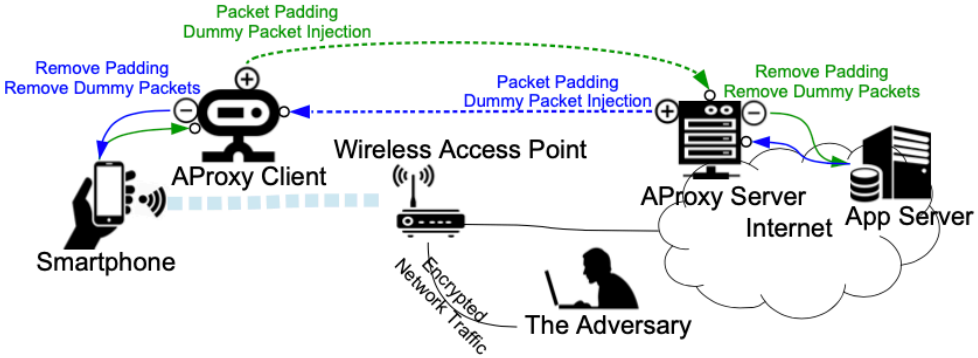


Identifying positive reporting in a COVID-19 contact tracing app (TraceTogether)

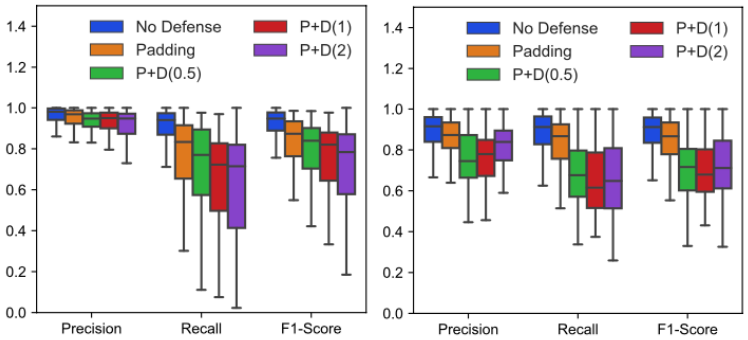
Evaluation

- Defense: packet padding and dummy packet injection

Design of our AProxy countermeasure



Evaluating AProxy against FOAP



(a) Open-world app traffic recognition.

(b) EP method identification.

Summary

- FOAP is the first approach to identifying method-level fine-grained user action of Android apps in the open-world setting
- FOAP effectively reduces the false positive risk in open-world app recognition through adaptive traffic filtering
- FOAP carries out method-level user action identification via synthesizing traffic and binary analysis
- FOAP can be applied in user activity inference and user privacy analysis

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Opening Minds • Shaping the Future
啟迪思維 • 成就未來

Thank You

Email: jfli.xjtu@gmail.com