



# 隐私消亡的移动互联时代

杨 珉

教授 国家973首席科学家

# 王珞丹住哪里？



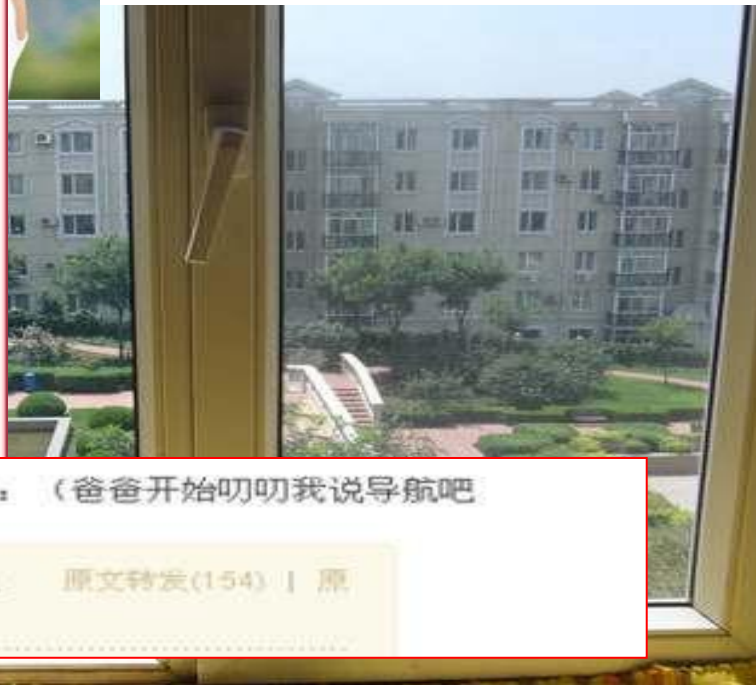
四环堵死了！我联排迟到了



2010-3-6 13:46

光顾着看围脖留言忘记给老爸指路！都开到中关村了。（爸爸开始叨叨我说导航吧

@王珞丹V：爸爸送我和小6去给<无人驾驶>配音的路上 原文转发(154) | 原文评论(310)



# 王珞丹住哪里？



# 顶级信息安全专家“中招”

- 吕述望 教授，博士生导师
  - 中科院信工所，信息安全国家重点实验室。

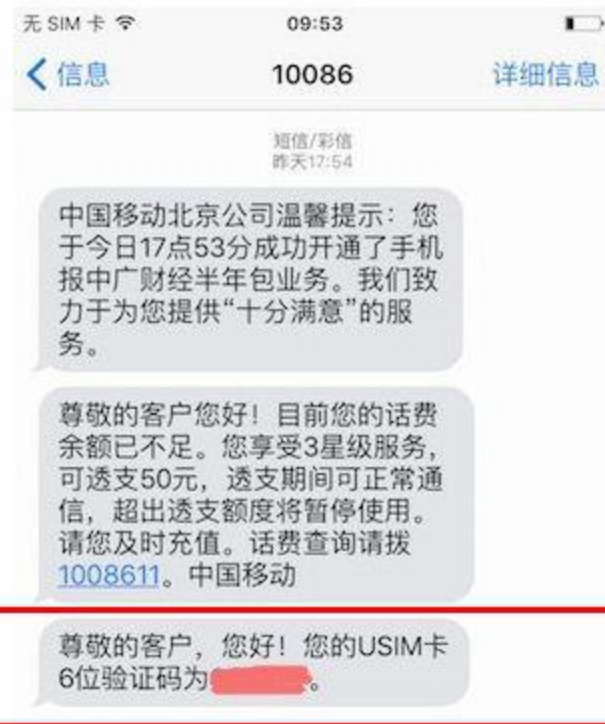
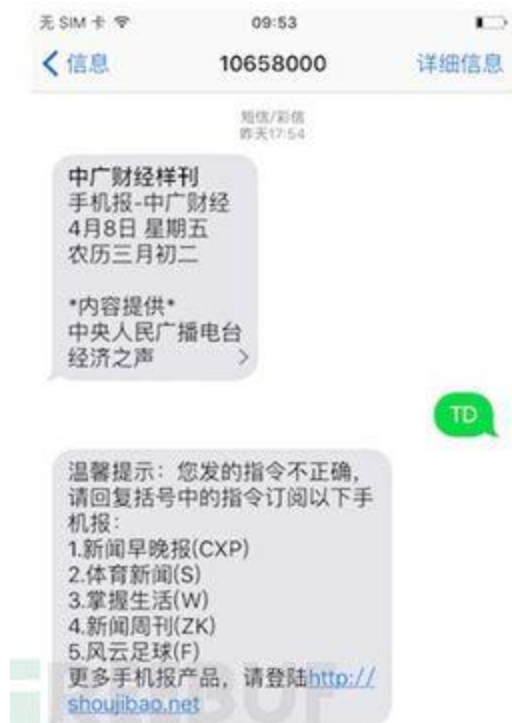


老吕，刚才你给我发过这个短信息吗？

“严明，我是吕述望，这是我帮你拍的小视频 [df.tc/sgND9J](http://df.tc/sgND9J)”。我看着可疑，没有打开它。

# 一条短信，支付宝沦陷了

- 2016年4月11日，“中国移动，请你告诉我，为什么一条短信就能骗走我所有的财产？”





# 移动互联生态系统：光怪陆离

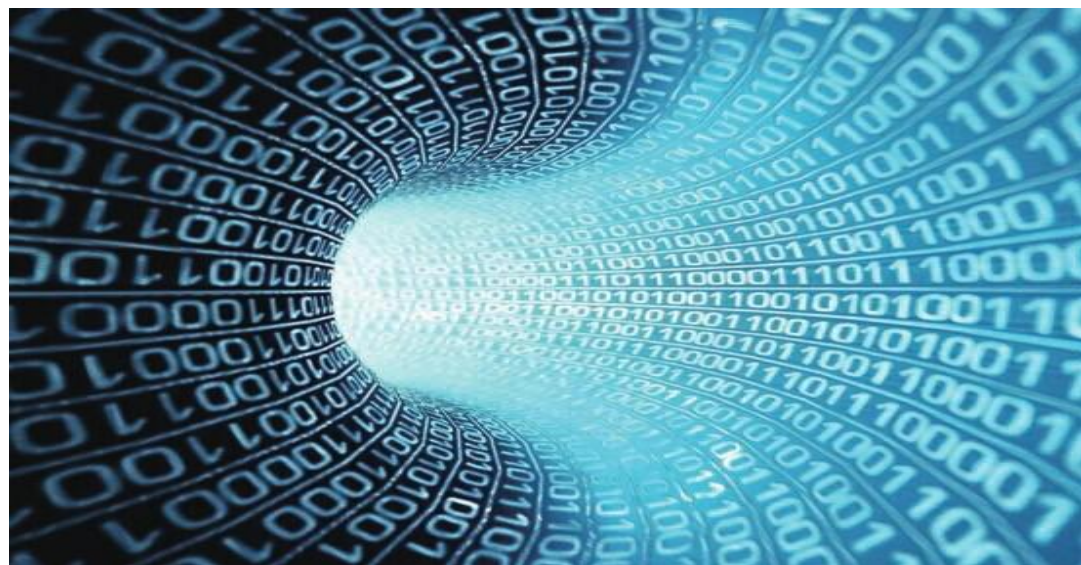


- 隐私：我是谁？我在哪里？我做了什么？

# 泛在智能的平行空间



实体空间



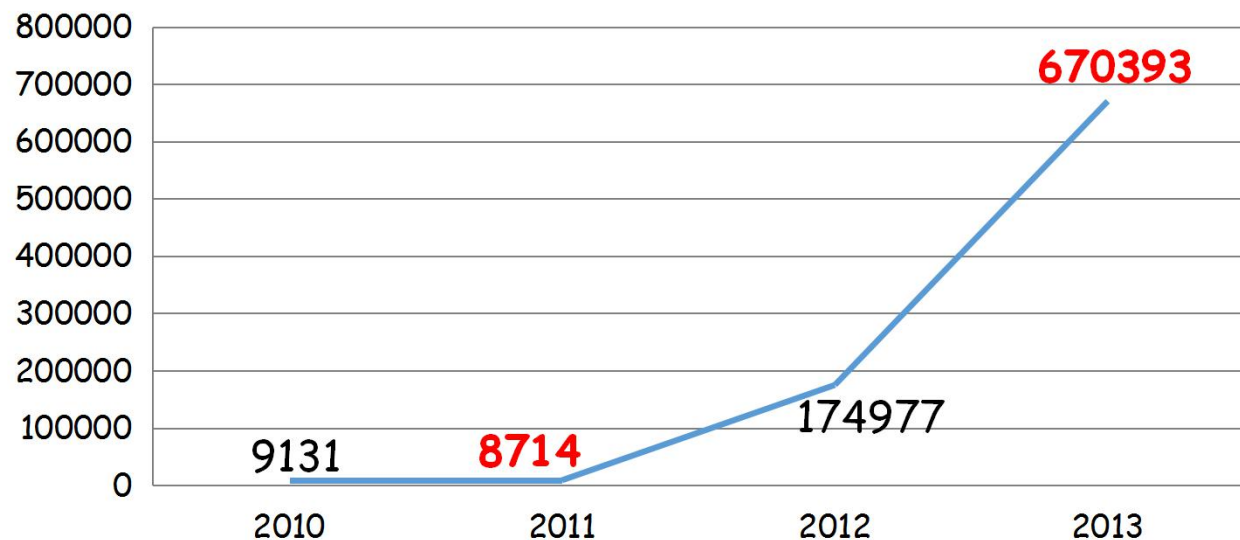
网络空间





# 移动恶意软件泛滥

- 智能终端安全问题严重，成为社会热点问题。



来源: McAfee Labs, 2013 Internet Threat's Report

2013年，美国直接经济损失 100 亿美元。

来源: IDC, Mobile Security Threat Report, 2014

2013年，中国直接经济损失超过 1000 亿元。

来源: 中国互联网协会, 2013年中国互联网产业发展综述

数百万款恶意软件

数万款恶意软件

数千款恶意软件







# Science: The End of Privacy

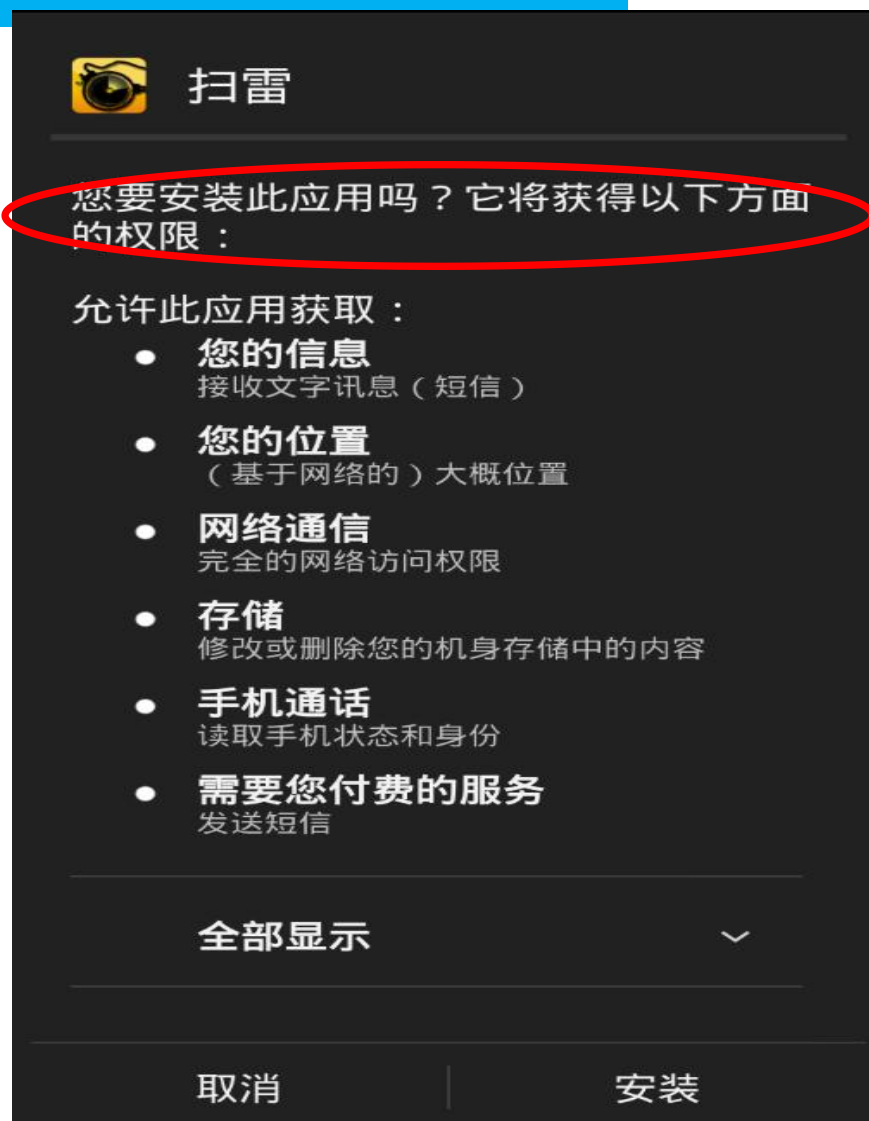


## • 超学科的研究范畴

- Unmasked
- When your voice betrays you
- Breach of trust
- Risk of exposure
- Could your pacemaker be hackable?
- Trust me, I'm a medical researcher
- Control use of data to protect privacy
- Privacy and human behavior in the age of information
- Balancing privacy versus accuracy in research protocols
- What the "right to be forgotten" means for privacy in a digital age
- .....

# 移动平台的隐私保护机制

- 权限 (Android、iOS)
  - 授权软件可**访问**和**使用**敏感数据
- 恶意行为检测方法
  - 静态、动态、动静态
  - Google Bouncer/iOS Vetting Process
- 安全软件
  - 腾讯、百度、360、安天、LBE





# 应用软件隐私泄露严重

泄露内容 数目统计	商城			厂商		运营商	平台	汇总
	1	2	3	1	2			
IMEI	42	27	22	24	20	11	36	182
IMSI	14	7	5	7	1	4	2	40
ICCID	4	1	1	2	0	1	2	11
Contact	10	2	5	2	1	3	1	24
Phone Number	7	1	3	5	0	1	0	17
SMS	2	0	3	2	6	1	1	14
Location	0	1	0	2	1	0	0	4
样本数目	70	50	40	50	50	20	50	330
泄露数目	47	29	25	25	16	13	36	191
泄露比率	67%	58%	62%	50%	32%	65%	72%	58%

国内Android应用商城中程序隐私泄露分析, 第五届信息安全漏洞分析与风险评估大会, 2012年

AppIntent (CCS 2013), VetDroid (CCS 2013), IEEE Trans. On Information Forensics and Security等多篇CCF A类论文





# UIPicker: User-Input Privacy Identification in Mobile Applications

Usenix Security 2015 (信息安全顶级会议)

# Motivation



## Privacy leakage

- App's insecure implementation
- Vulnerabilities in system
- Millions of Malware

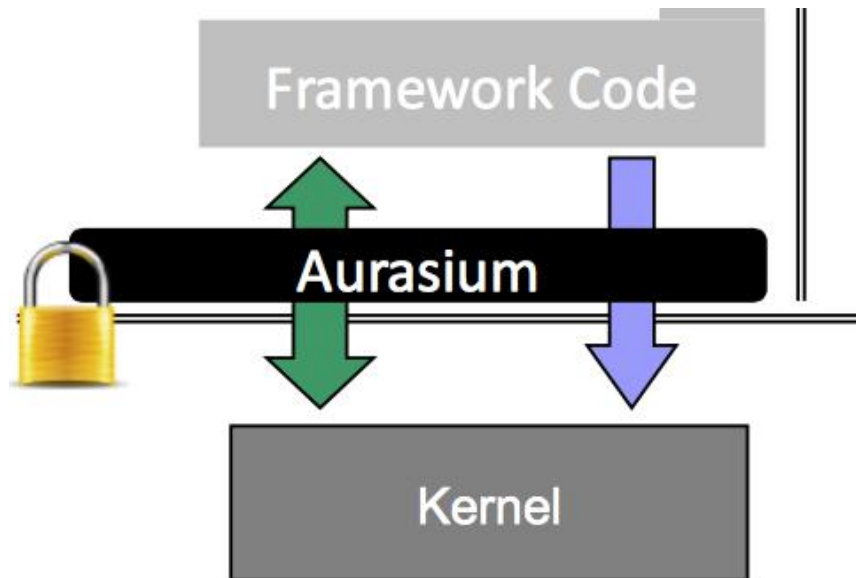


# Motivation

## Existing Works

Dynamic/Static taint analysis

- TaintDroid [OSDI'12]
- FlowDroid [PLDI' 14]

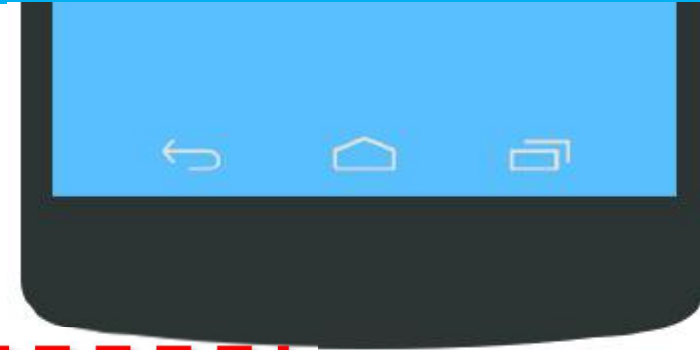


```
c = taint_source()
...
a = b + c
...
network_send(a)
```

Access control mechanisms

- SELinux on Android
- Auriasium [SECURITY'12]
- FineDroid [SecureComm'15]

# But wait ...



- Location ✓ `LocationManager.getLastKnownLocation()`
- Contact ✓ `ContentResolver.query(CONTACT_URI)`
- SMS ✓ `SmsMessage.getMessageBody ()`
- Phone Number ✓ `TelephonyManager.getLine1Number()`
- ... ✓ ...

## System Centric Privacy Data



# But wait ...

Account Credentials  
& Profiles

Full name  
First name, last name

Skype Name  
6-32 characters

Repeat password  
6-20 characters

Email  
someone@example.com

Mobile number  
Phone number (e.g. +44 1234567)

Yes, send me Skype news and promotions.

amazon

Or enter a new shipping address  
Be sure to click "Ship to this address" when done.

Full Name:

Address Line 1:  
Street address, P.O. box, company name, c/o

Address line 2:  
Apartment, suite, unit, building, floor, etc.

City:

State/Province/Region:

ZIP:

Location

银行卡支付

中国银行(信用卡)  
6227 6121 4583

03/20

369

bob

身份证  
310109196509

138 1839

727114

同意《协议》并保存卡信息, 支付更便捷

支付金额¥95 立即支付

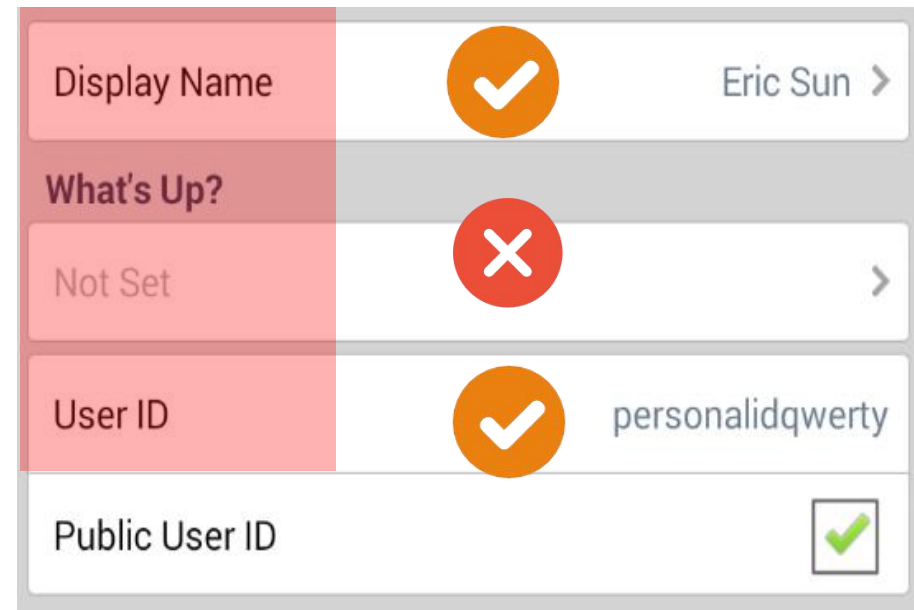
Financial

## User-Input Centric Privacy Data (UIP Data)

# Challenges

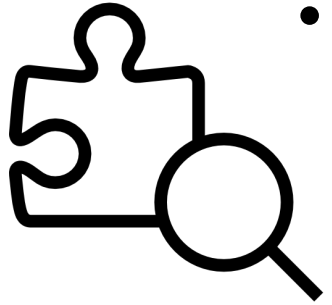
- UIP data cannot be found without parsing the context and semantic of UIs

- Label all input as sensitive
  - Large number of false positives



- Manually specify such contents need to be protected
  - Intensive human intervention

# Our Work



- UIPicker

- Novel framework for automatic, large-scale User-Input Privacy (UIP data) identification within Android apps.

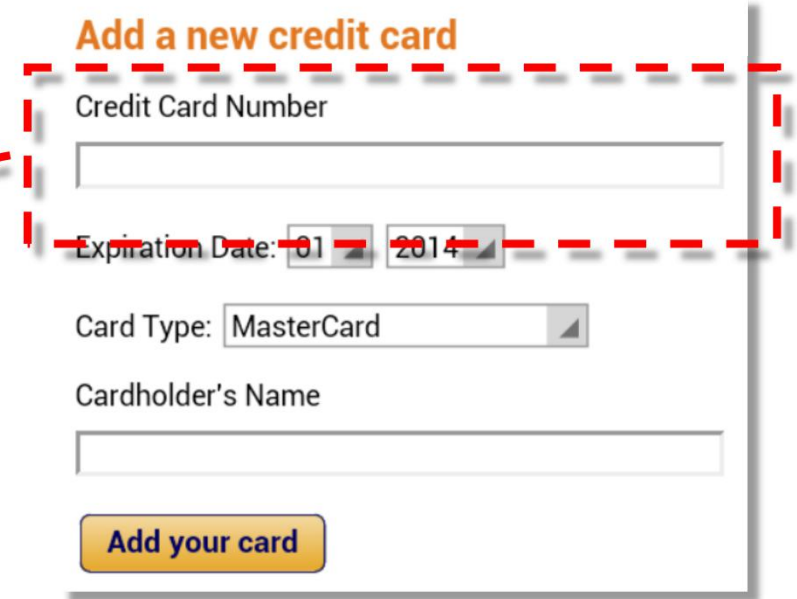
- Runtime Security-Enhancement Mechanism

- Protect insecure UIP data transmission in app's runtime



# Key Observation

- Privacy-related semantics exist in
  - UI Screens
  - Layout resource files



The screenshot shows a form titled "Add a new credit card". It contains several input fields: "Credit Card Number" (a text input field), "Expiration Date" (two dropdown menus showing "01" and "2014"), "Card Type" (a dropdown menu showing "MasterCard"), and "Cardholder's Name" (a text input field). A yellow button labeled "Add your card" is at the bottom. A red dashed box highlights the "Credit Card Number" field and the "Expiration Date" field. A red dashed arrow points from the bottom of this box towards the code block below.

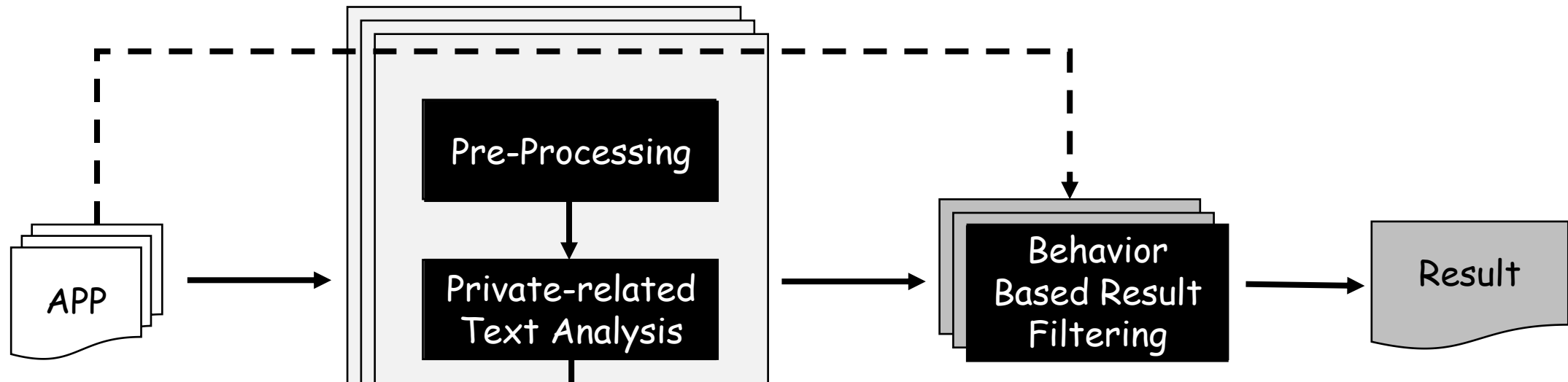
```
<TextView android:text="@string/opl_new_payment_credit_card_number" />  
<EditText android:id="@id/opl_credit_card_number" android:inputType="number" />
```



# Technical challenges

- How to get complete privacy-related texts
  - Highly unstructured semantics
    - E.g. password, pass, pwd, ...
  - Sparsely distributed in different UIs
    - In practical for manual collecting
- To what extent, a layout element could be UIP data
  - Keyword based search? Very imprecise results
    - E.g. Split "Username" into "user" & "name"

# UIPicker Overview



1. Resource Extraction
2. Cluster privacy-related texts describing UIP data
3. Identify UIP data from textual semantics
4. Confirm UIP data with application code behaviors

# Pre-Processing

- Resource Extraction
- Word splitting
  - Delimiter/Letter-separated words
- Redundant content removal
  - Non-English strings/Stop words
- Stemming
  - Reduce the number of texts
  - E.g. Change "secured", "security" into "secure"

1 UI Texts

## Add a new credit card

Credit Card Number

Expiration Date: 01 2014

Card Type: MasterCard

Cardholder's Name

Add your card

2 Layout Descriptions

@id/opl\_credit\_card\_number

@string/opl\_new\_credit\_card\_expiration\_date\_month

@string/opl\_new\_credit\_card\_save\_button

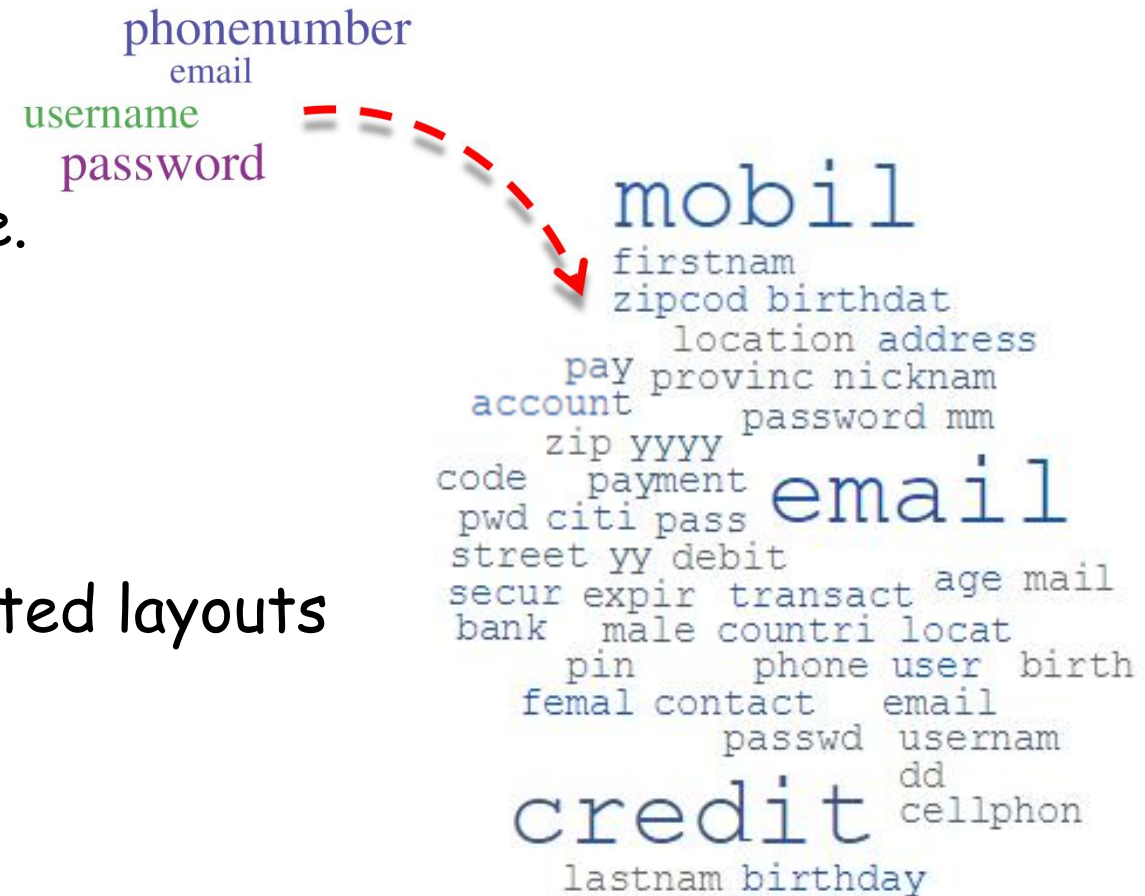
# Privacy-related Texts Analysis

- Privacy-related layout Selection

- Subset of UIP related layouts.
- E.g. Login, Registration, settings page.
- Initial seeds: username, password

- Privacy-related texts clustering

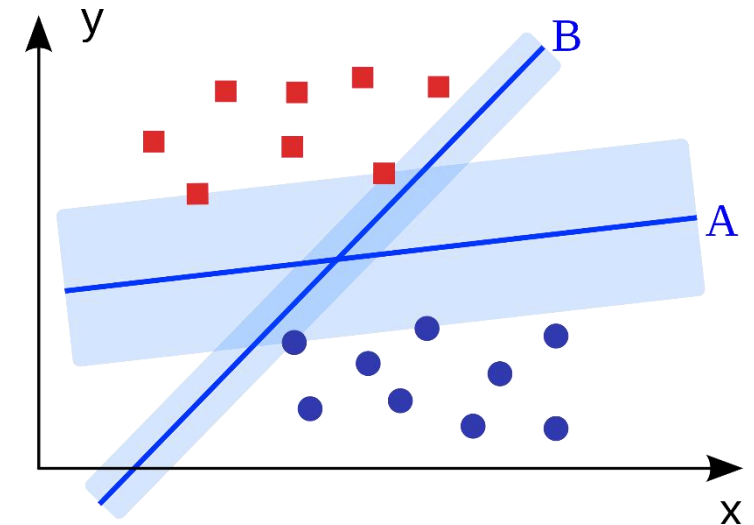
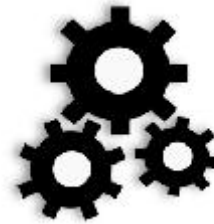
- Words appeared most in privacy-related layouts rather than normal layouts.
- By applying Chi-Square test.





# UIP Data Identification

- Classifier: Support Vector Machine
- Features
  - Privacy-related texts
  - Sibling elements
    - E.g. Short phrase for describing the input field.
- Training Data
  - Subset of UIP data element
    - Text fields with sensitive **inputtype**.



```
<EditText android:id="@id/password_edittext" android:inputType="textPassword" />
```

# Behavior Based Result Filtering

- Matching UIP data behaviors reflected in application's program code
- Filter out **static labels which only contain privacy-related semantics**
- Recover Input fields other than "EditText"
  - Implicit user input
    - E.g. Drop down list
  - Customized input
    - E.g. `com.alipay.inputBox`

The screenshot shows a mobile app sign-up form with several elements highlighted by red dashed boxes and icons:

- A checkbox labeled "Upload my address book to connect me with my friends." is checked.
- A "Sign Up" button is highlighted.
- A text block containing "By tapping 'Sign Up' above, you are agreeing to the [Terms of Service](#) and [Privacy Policy](#), including [Cookies use](#)." is highlighted.
- A text block containing "Your email address and phone number may be used to connect you with others, but will not be" is highlighted.
- A red circle with a white 'X' icon is positioned to the right of the highlighted text blocks.
- An orange circle with a white checkmark icon is positioned to the right of the form's bottom section.

The bottom section of the form includes:

- Expiration Date: 01 2014
- Card Type: MasterCard

# Behavior Based Result Filtering

## 1. Locating UIP candidate & its layout

```
Activity.setContentView(R.layout.add_credit_card.xml)
```

```
Void onCreate(Bundle bundle){
```

```
    InputBox IB = findViewById(2131231511);
```

```
    submitBtn = findViewById(2131623982);
```

```
    submitBtn.setOnClickListener(new addCardListener());
```

```
}
```

## 2. User-Triggered Event handling

```
addCardListener.onClick(View v) {
```

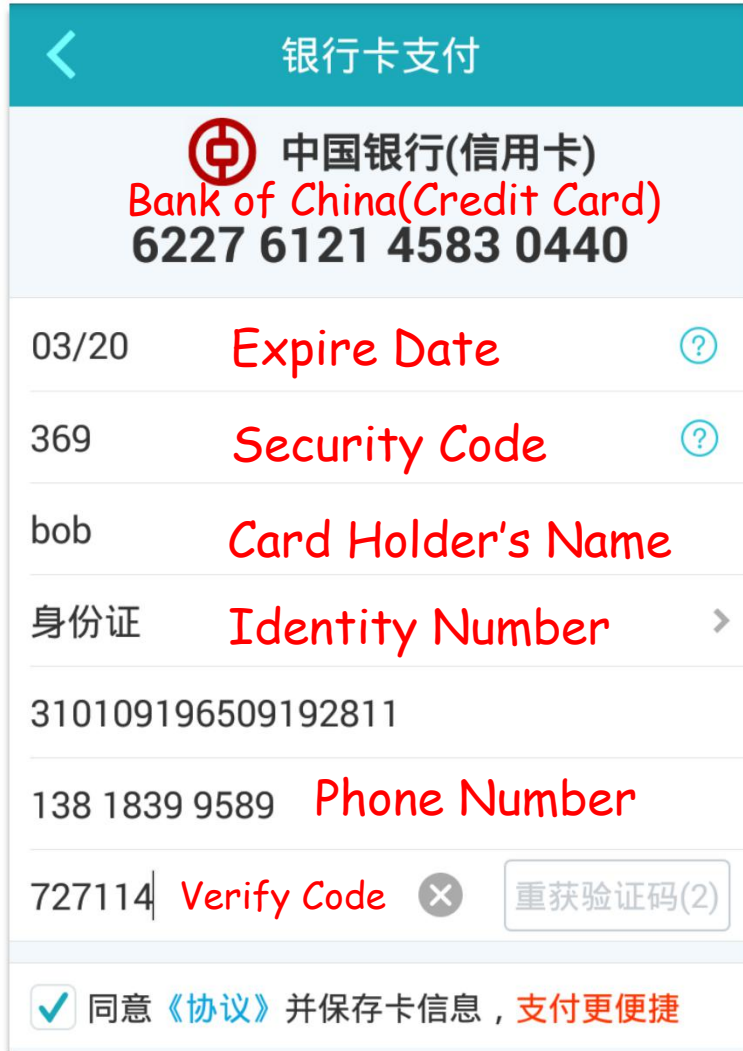
```
    .....  
    creditCard = IB.getText();  
    sendContent(creditCard)
```

```
    .....
```

```
}
```

The screenshot shows a form titled "Add a new credit card". The form contains the following fields: "Credit Card Number" (with a red box around it and the text "UIP Data Elements" next to it), "Expiration Date" (with two dropdown menus showing "01" and "2014"), "Card Type" (with a dropdown menu showing "MasterCard"), and "Cardholder's Name" (with an empty text box). At the bottom of the form is a yellow button labeled "Add your card". A red hand icon is pointing at the button. Dashed red arrows originate from the "Credit Card Number" field and the "Add your card" button, pointing towards the code snippets on the right.

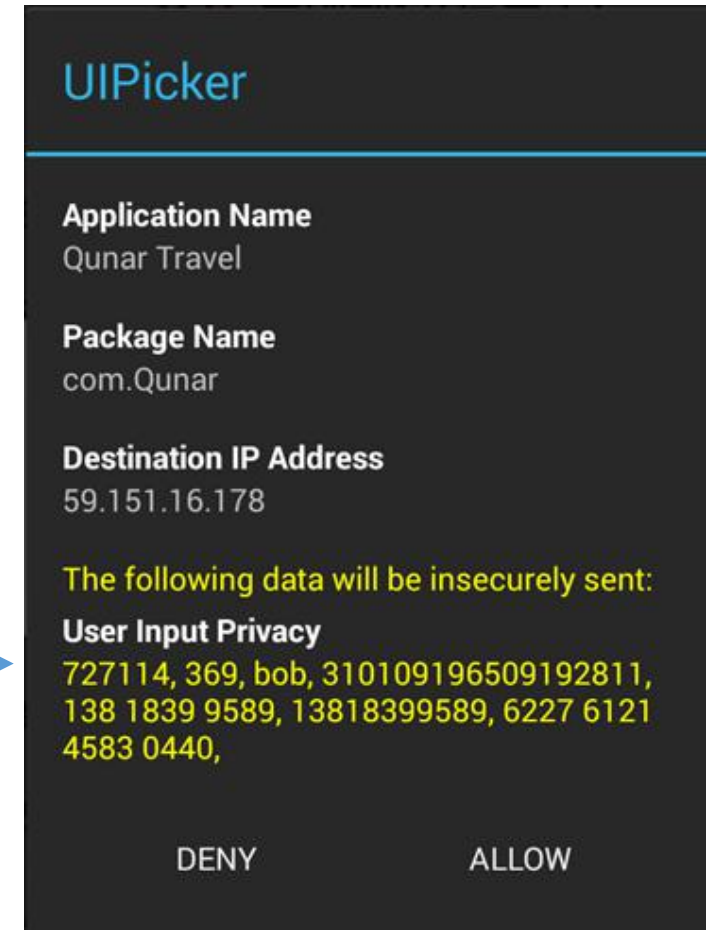
# Runtime Security Enhancement



Taint Tracking based on UIPicker



```
c = taint_source()  
...  
a = b + c  
...  
network_send(a)
```



- Plain Text transmission
  - Runtime checking
- Insecure SSL implementation
  - Offline analysis with MalloDroid

# UIPicker Evaluation

- Dataset
  - 17,425 apps crawled from Google-Play in Oct.2014
  - 35 categories, 500 apps for each

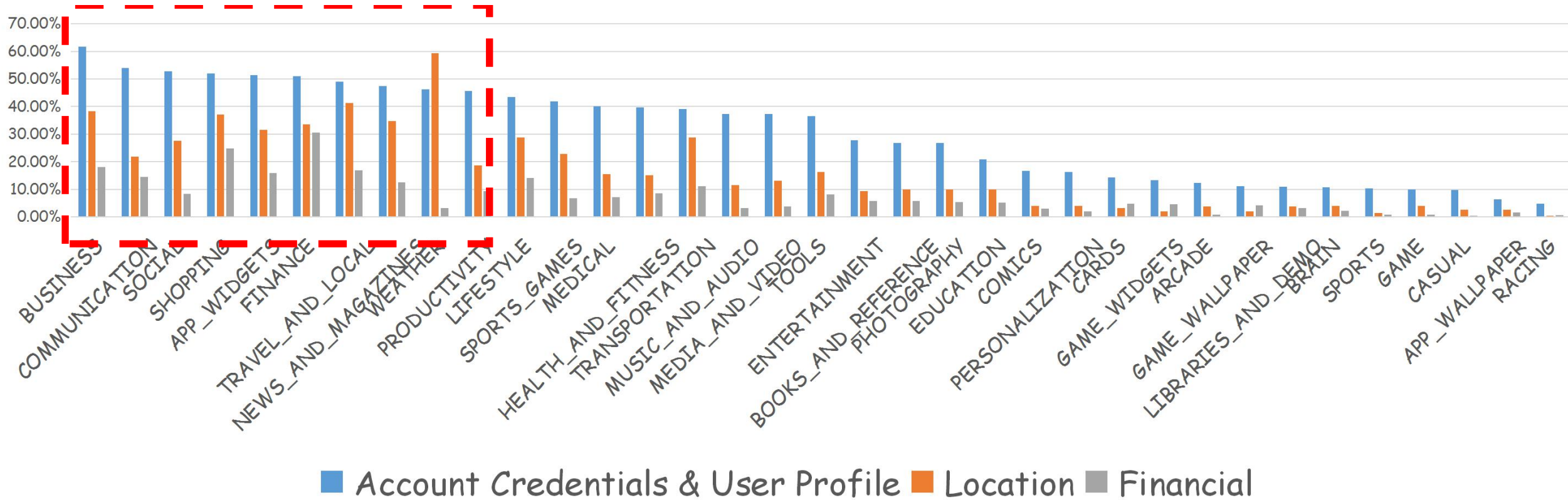


# Performance

- Identification Process
  - 32 core Debian server with 64GB memory
  - Proceeded in 32 threads concurrently
  - Finished in 30.5 hours (7.6 apps per minute) for the dataset
- Run-Time Security Enhancement Mechanism
  - Only provides additional UIP data sources
  - No performance overhead compare to TaintDroid

# UIP Data Distribution

- 6,179 (35.46%) contain UIP data in 17,425 Apps
- Exist in more than half of apps in 9 out of 35 categories.



# UIP Data Distribution

- Categories like BUSINESS, FINANCE, SHOPPING, COMMUNICATION and SOCIAL are more likely to request Account Credentials and User Profile information
- SHOPPING category contains many location-related elements
- Both FINANCE and SHOPPING apps require many financial-related sensitive inputs

# Effectiveness

- Compare with System Defined APIs (# Apps)

TelephonyManager.getLine1Number()  
AccountManager.getAccounts()  
  
LocationManager.getLastKnownLocation()  
Location.getLongitude()  
Location.getLatitude()

Category	System Defined APIs	UIPicker	Overlap
Account Credentials & User Profile	4,900	5,330	1,340
Location	15,221	2,883	2,282
Financial	-	1,318	-
Total	15,632	6,179	-

# Effectiveness

- Compare with Sensitive Attributes(# Elements)

textEmailAddress textPersonName  
textPassword textVisiblePassword  
password/email/phoneNumber

textPostalAddress

Category	InputType	UIPicker	Incremental
Account Credentials & User Profile	24,021	46,227	26,087
Location	941	14,311	13,370
Financial	-	6,353	-
Total	24,962	71,224	46,262

Annotations: Red dashed boxes highlight the 'InputType' and 'UIPicker' columns. Red dashed arrows point from the 'textPostalAddress' box to the 'UIPicker' cell for 'Location' and from the 'Sensitive Attributes' box to the 'InputType' cell for 'Account Credentials & User Profile'. Red arrows on the right indicate a 15x increase for 'Location' and a 2x increase for 'Total'.



# Effectiveness

- Compare with Sensitive Attributes(# Elements)

Type	# Elements	% in UIP Data
TextView	10,582	14.86%
Customized	5,075	7.13%
Spinner	1,962	2.75%
Others	784	1.10%
Total	18,403	25.84%

# Precision

- Manual Validation
  - 200 random selected apps in 10 categories (20 in each)
  - False Positives: 67/1042 elements
  - False Negatives: 107 elements
  - Overall: **93.6% Precision and 90.1% Recall**

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

# Conclusion

- UIPicker: Identify UIP data based on novel combination of NLP, machine-learning approach, and static analysis techniques
- Runtime security enhancement based on UIPicker
- Easily be deployed for other existing mechanisms for privacy analysis/protection.

谢谢！

m\_yang@fudan.edu.cn