Notice the Imposter! A Study on User Tag Spoofing Attack in Mobile Apps

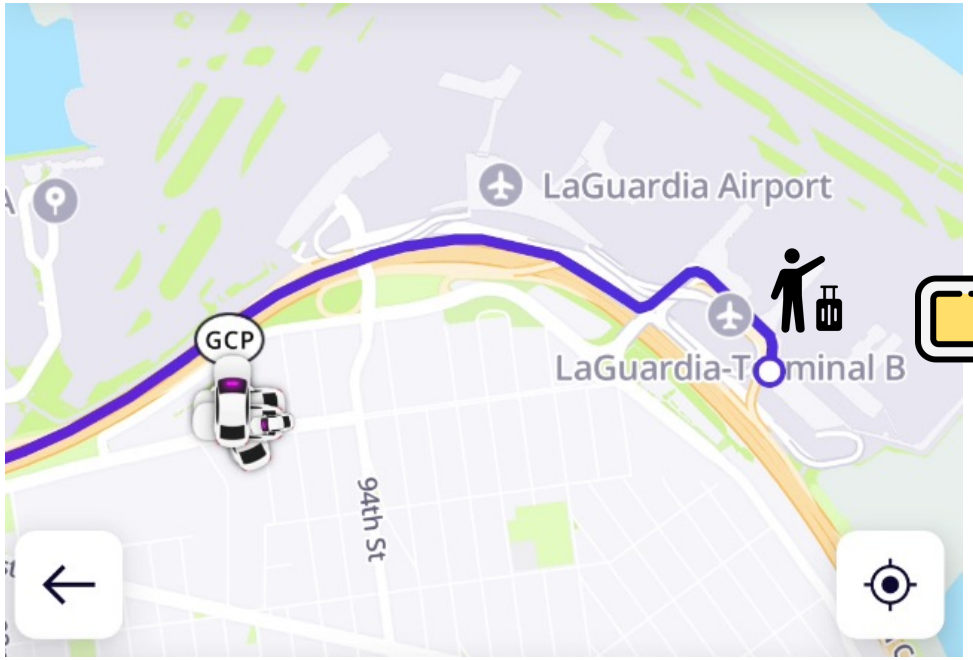
Shuai Li, Zhemin Yang, Guangliang Yang, Hange Zhang,
Nan Hua, Yurui Huang, Min Yang

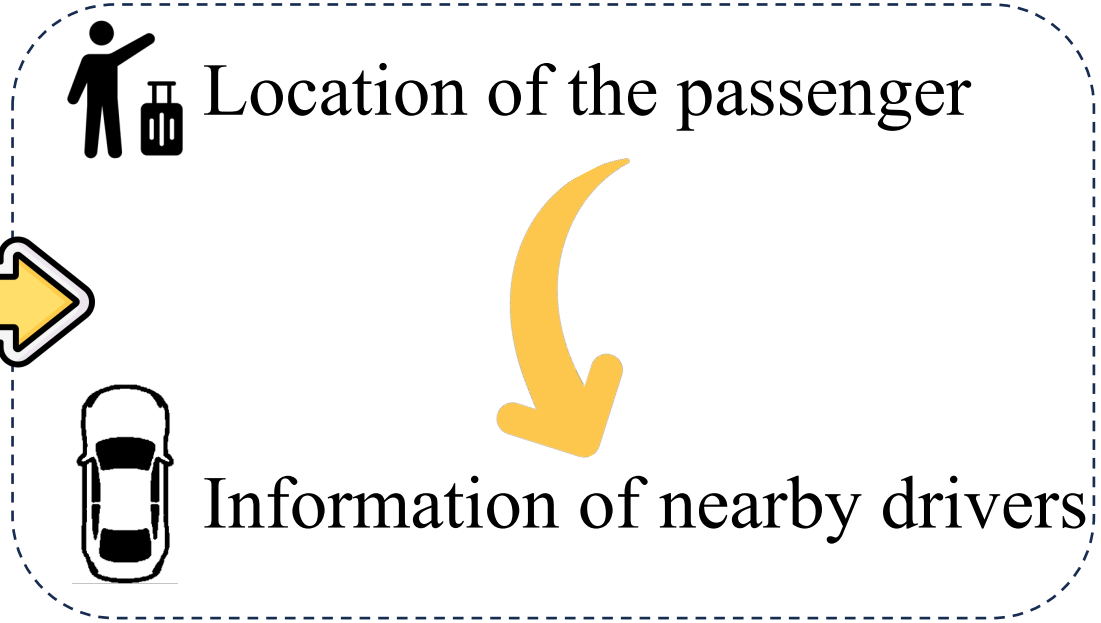
Fudan University

User Tag Sharing

- **An Example**

Mobile Ride-hailing Service



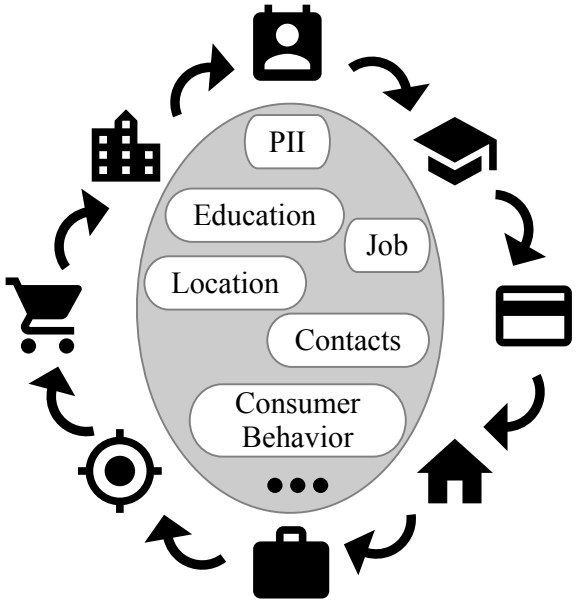


Bound Tag

Free Tags

User Tag Sharing

- **What is Bound Tag, Free Tag and User Tag?**



Digital User Profile

User Tags

- Id: 6374d4bb9afc
- Birthday: 1994-01-01
- Location: 40.69, -74.31
- Height/Weight: 173/100
- Gender: Male
- Income: $100,000/year
- PhoneNumber: xxx
- Contacts: User1, User2
- Preference: Food, Game
- Date: 2023-08-11

...

Bound Tag

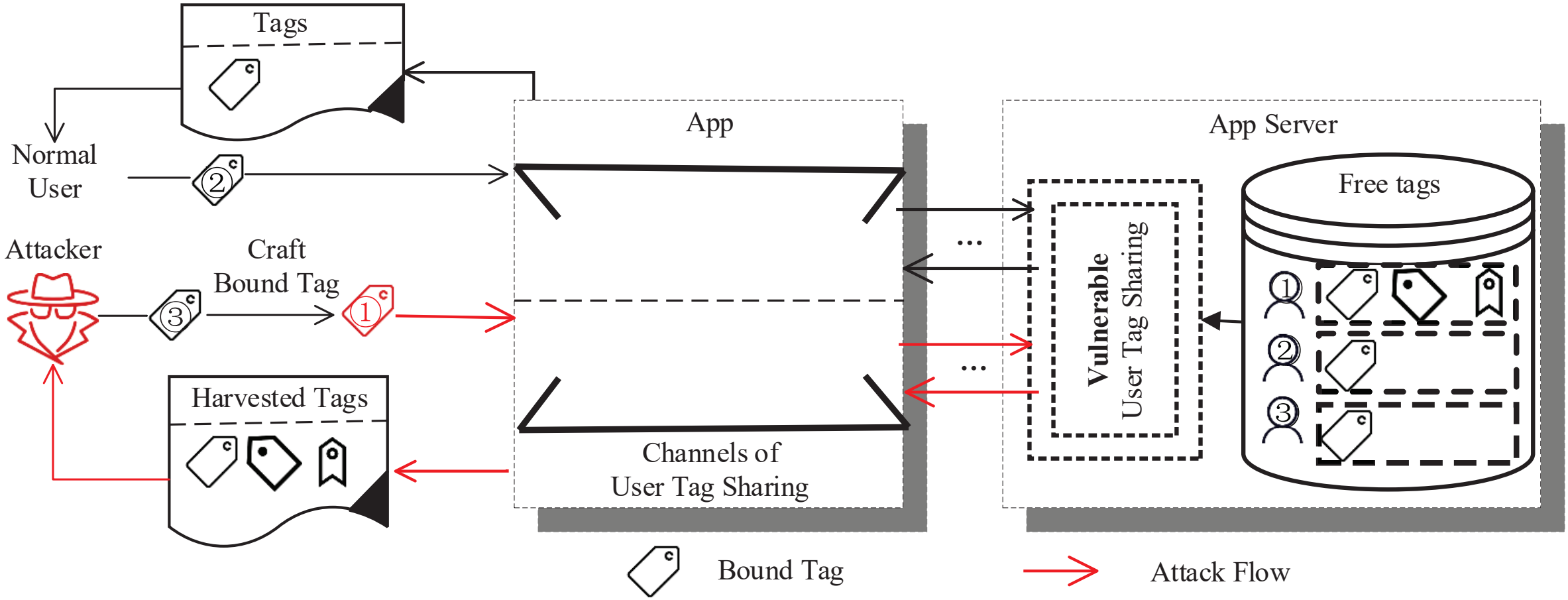
user tags selected as the basis for clustering users

Free Tag

user tags apart from bound tags during user tag sharing

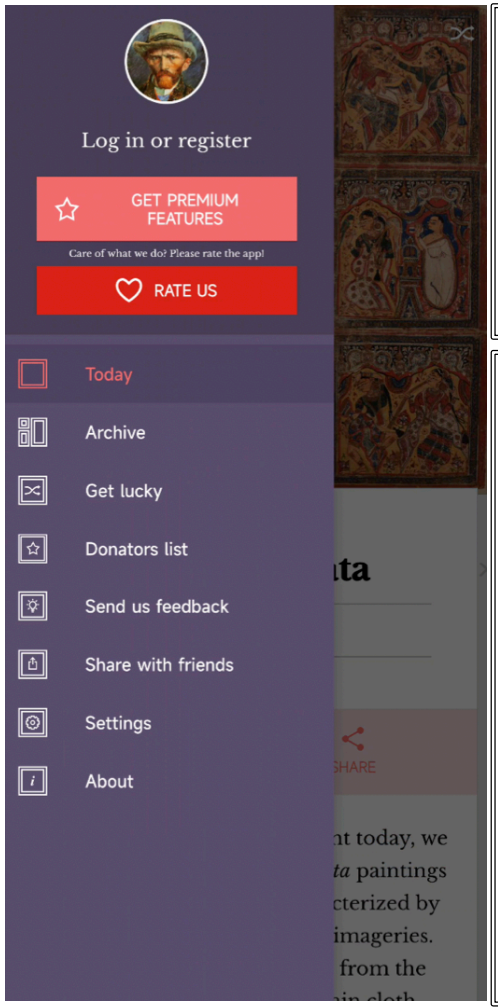
User Tag Spoofing

- **Craft *Bound Tag* to Illegally Access *Free Tag***



User Tag Spoofing

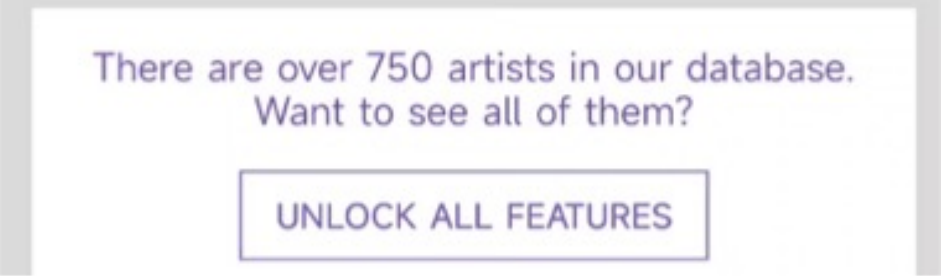
Mobile Service: "Today"

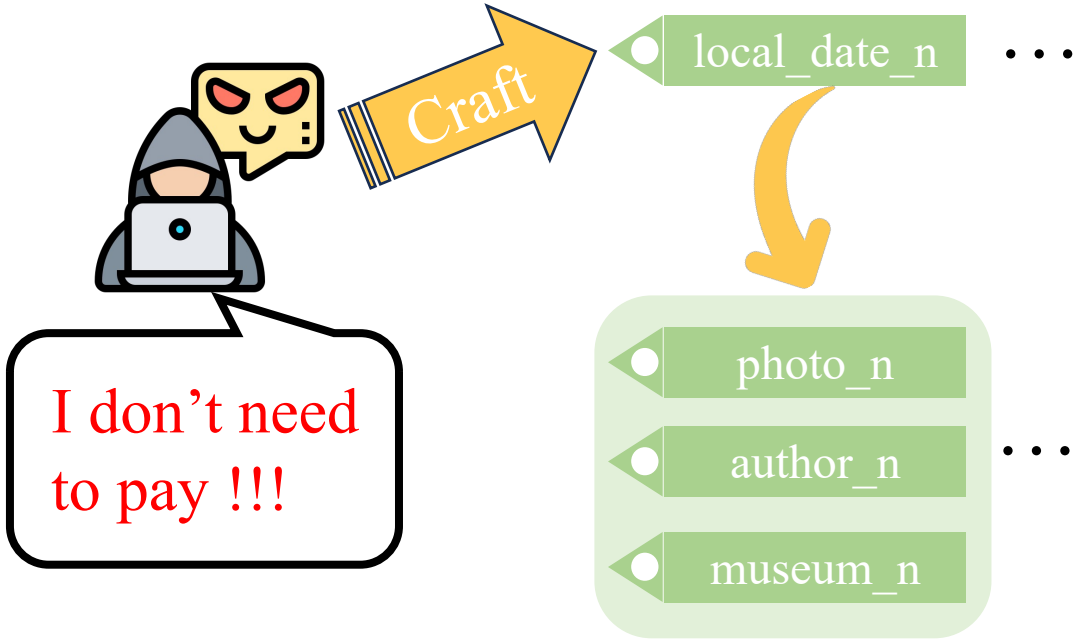



```
POST /APIMobile/artworks/query HTTP/1.1
…

{
   "local_date":"2022-11-17",
   …
}
```

```
HTTP/2 200 OK
…

{
   "entities":[
    {
      photo_related_info,
      author_info,
      museums_info
    },
    …
    ]
   …
}
```

Only Five Artists are available for unsubscribed users





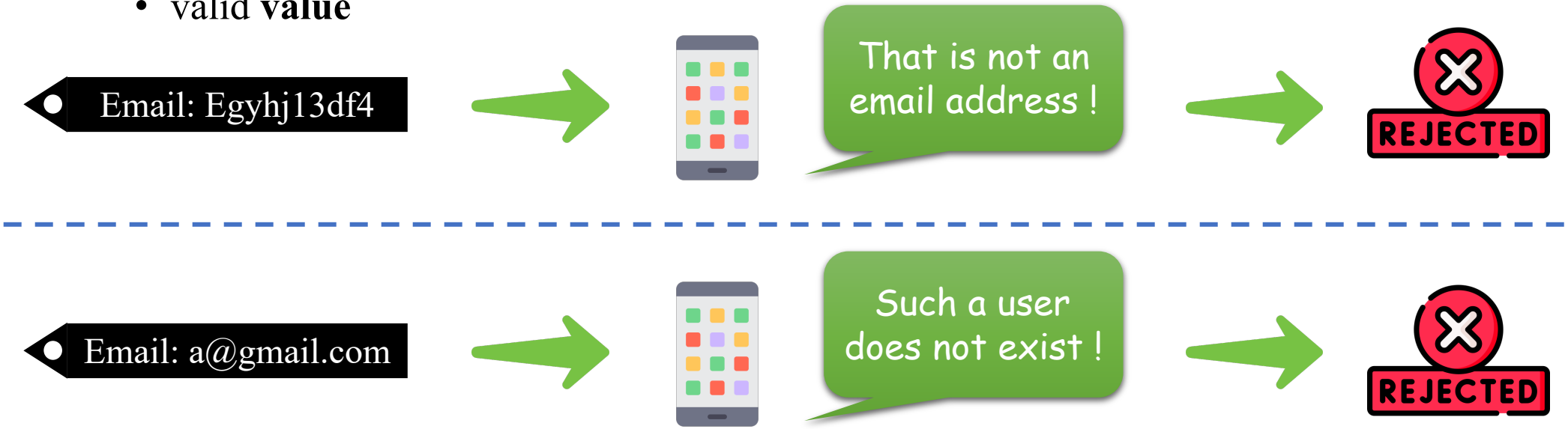
Research Status

- Related Works Focusing on:
  - Privacy Issues Against Personal Data Collection
  - Case Studies Regarding User Tag Spoofing
    - Contact Discovery Service (e.g., All the Numbers are US – NDSS'21)
    - Location Based Service (e.g., Geo-locating Drivers – NDSS'19)
    - …

No idea about the whole picture of user tag spoofing

Approach

- Challenge
  - How to Locate Bound tags?
    - Bound tags have diverse semantics, which **are specific to** concrete mobile services
  - How to craft valid bound tags?
    - valid **format**
    - valid **value**



Approach

- key Insight
  - Bound tags -- differentiate a individual user or a group of users from others

Identify Bound Tag

Bound tags are normally sensitive in semantics

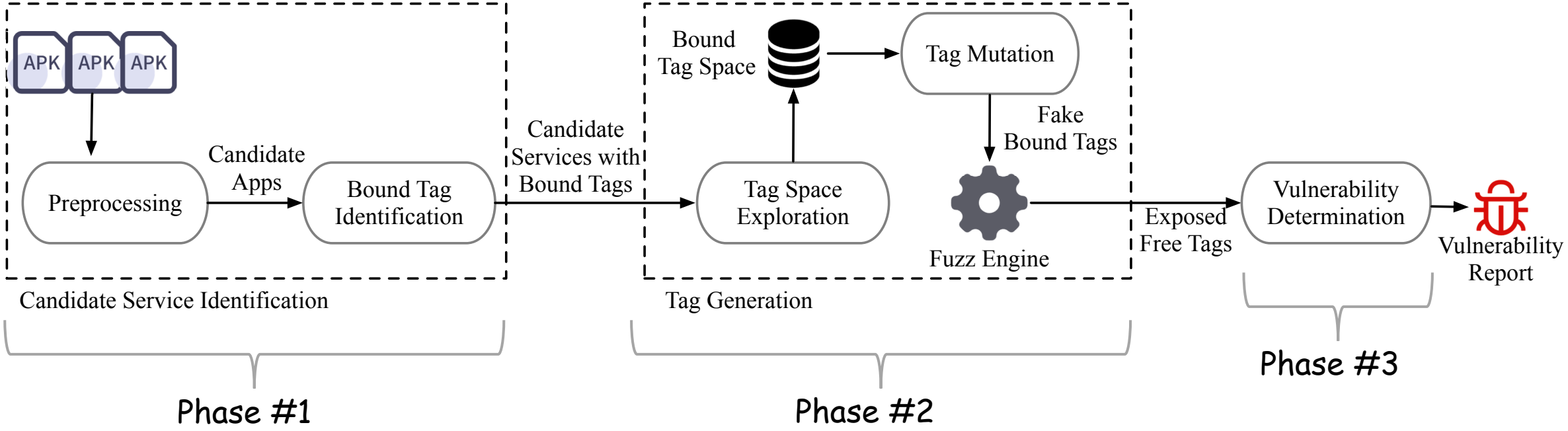
  - Many bound tags are shared back as free tags in mobile services

Craft Valid Bound Tag

Tag values of existing users in the tested service's app are supposed to be valid

Approach

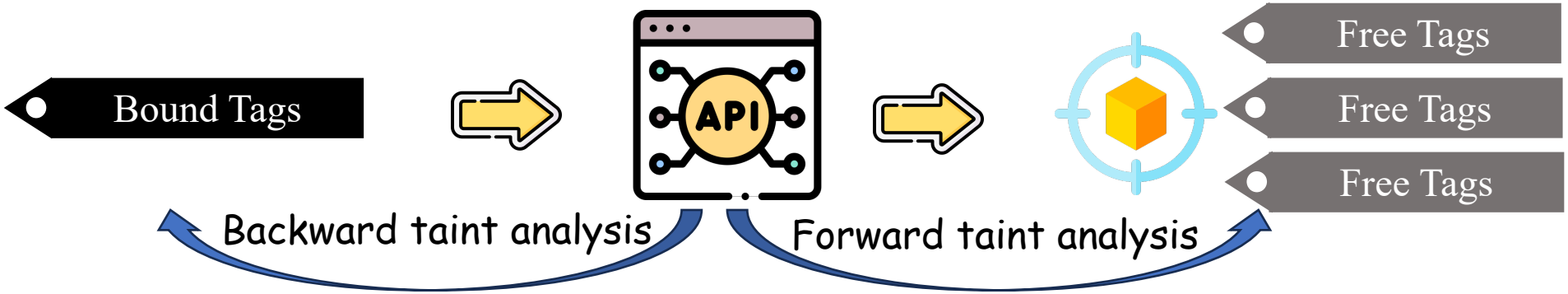
- UTSFuzzer - Architecture
  - Phase #1: Identify User Tag Sharing Services
  - Phase #2: Explore the Value Space of Bound Tag
  - Phase #3: Determine the Existence of Vulnerability



Approach

- Phase #1: Identify User Tag Sharing Services

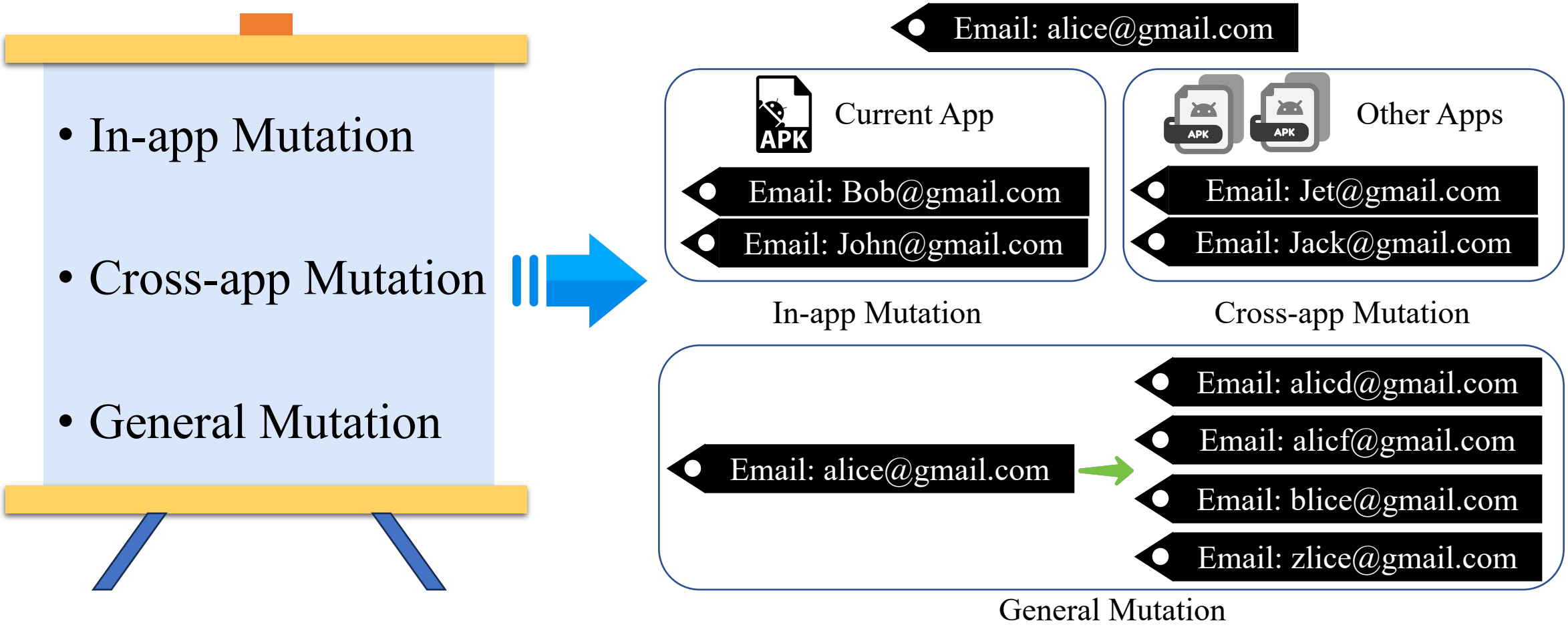
- Preprocessing
  - Code Pattern of User Tag Sharing



- Bound Tag Identification
  - Sensitive Semantic

Approach

- Phase #2: Explore the Value Space of Bound Tag



Approach

- Phase #3: Determine the Existence of Vulnerability
  - $F()$: Mobile service with user tag sharing
  - A_{bound_tag} : Bound tag of a user
  - U_{tags} : Free tags of Other users
  - S : A set of mutated bound tag values

$$U_{tags} = F(A_{bound\_tag}) \qquad U'_{tags} = F(A_{bound\_tag'}),\ A_{bound\_tag'} \in S$$

- Vulnerability exists when:

$$\exists A_{bound\_tag'} \in S\ ,\ U'_{tags} \neq NULL\ \&\ U'_{tags} \neq U_{tags}$$

Evaluation

- Research Questions
  - RQ1: Is UTSFuzzer effective in terms of security detection?
  - RQ2: How many real-world services are impacted by user tag spoofing?
  - RQ3: What attack efforts may be introduced by user tag spoofing?
- Dataset
  - 25,158 popular apps in 30 categories
  - Crawled From Google Play in April 2022

RQ1: Efficacy

- Determine the Existence of Vulnerability
  - Time Cost: 2246 hours
  - 3,257 candidate apps having user tag sharing services
  - **100** apps with **115** mobile services were detected to be vulnerable
  - Precision: **95.00%** / Recall: **98.96%**

	#Num	TP	FP	TN	FN
Candidate Apps VS Non-candidate Apps	438	390	48	-	-
	450	-	-	331	119
Vulnerable Apps VS Secure ones	100	95	5	-	-
	100	-	-	99	1

(Randomly Sampled & Manually Verified)

RQ2: Vulnerability Detection

Bound Tag		Package	#Installs	Service Description	Tag Generation Strategy	Samples of Leaked Free Tags
id	user_id	c**.e***	10M+	Get users' homepage	In-app	job, income, children, education, ethnicity, smoking, alcohol, height
	room_id	c**.m***.t***	500K+	Get owners of chat rooms	General	age, gender, country, language, income
	language_id	c**.f***.c***	1M+	Get users via language	In-app	distance, birthday, date of creation, is_online
	author_id	c**.m***.d***	1M+	Get authors of artworks	In-app	biography, artworks, museums, date of death & birth
	circle_id	c**.g***.f***	10M+	Get users in a circle	General	deeplink, email address, parent_id, device model, phone number
	moment_id	a**.t***.d***	500K+	Get commentators	General	birthday, country, city, email address, phone number
email address		c**.t**.v***	500K+	Get users' homepage	In-app	country, region, birthday, gender, date of creation
country		c**.w***.b***	100K+	Get live streaming users	General	name, country_id, rate, video_id
phone number		c*.h***.m***	10M+	Get users of contacts	Cross-app	real first name, real last name, date of last activity & registration
date		j*.c*.a***.a***	500K+	Get current popular users	General	age, country, login_date, height, weight, distance
location		r*.t***.a**	1M+	Get nearby users	Cross-app	car, birthday, zodiac, region, height, latitude & longitude, date of last activity & creation

- Affected Bound Tags: **11** unique types
- Leaked Tags involve info of:
  - **Demographics**
  - **Device**
  - **Contact**
  - **Education**
  - **Health**
  - **Employment**
  - **…**
- Accumulated Installs: **413** million+
- Existence of User Tag Spoofing in iOS platform

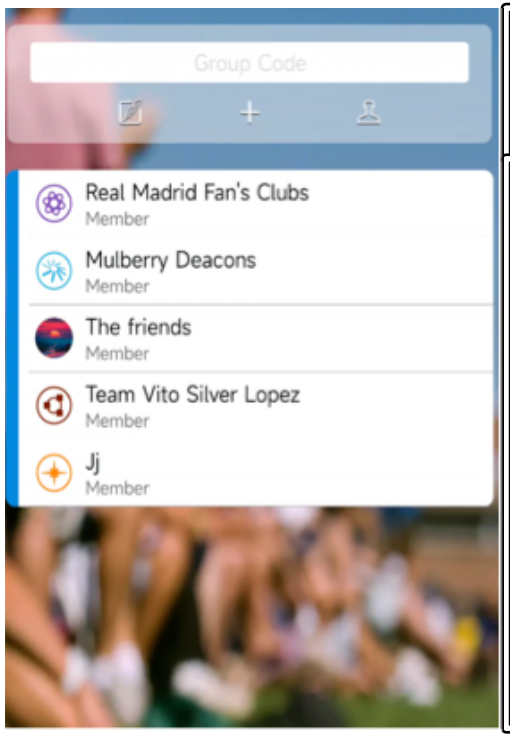
RQ3: Attack Efforts & Case Study

- Generally, user tag spoofing leads to the leakage of various user tags

- But, more than that, …
  - Business Secret Exposure
  - Preservation Mechanisms Breach
  - Economic Loss
  - **User Activities Monitoring**

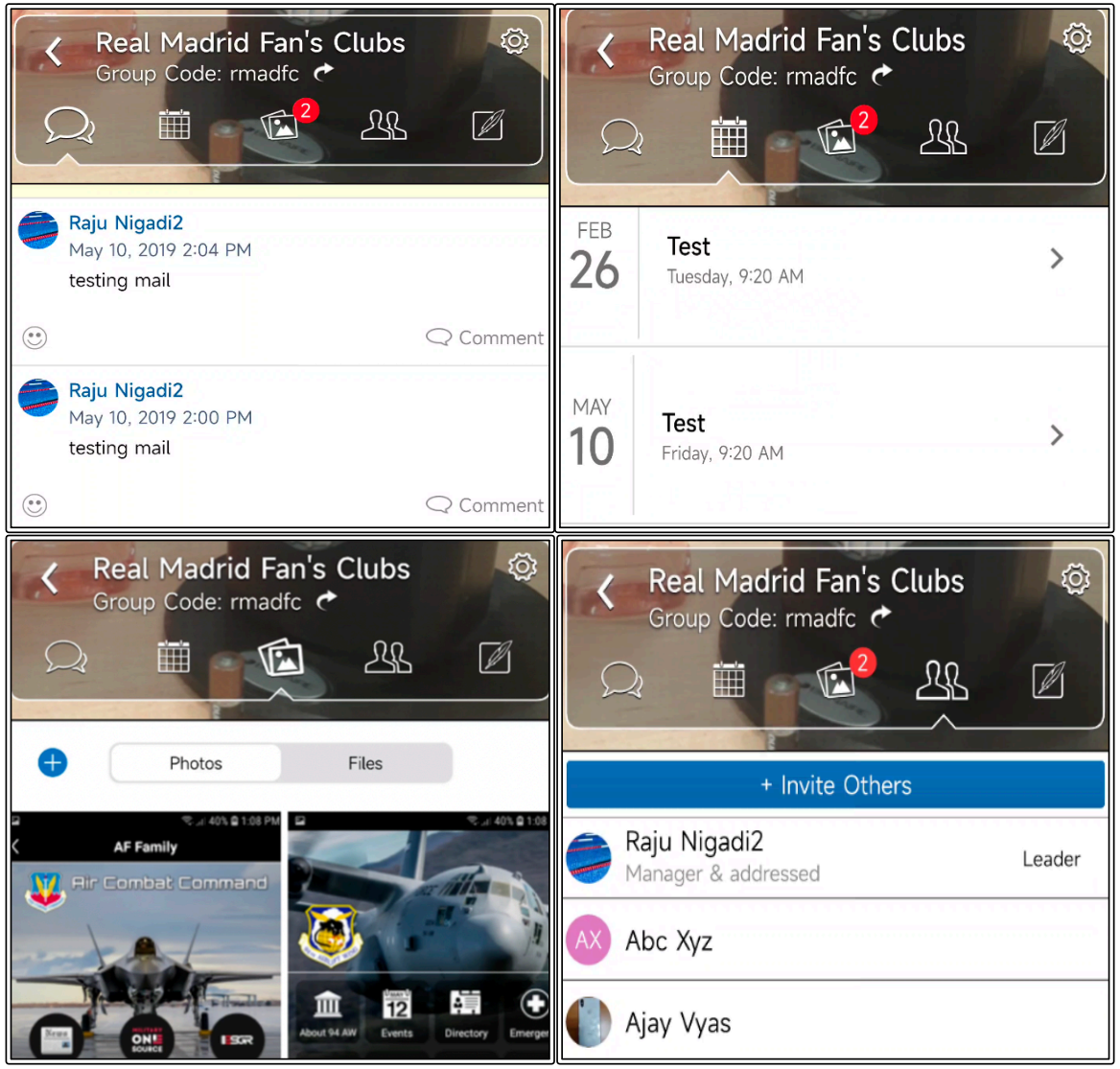
RQ3: Attack Efforts & Case Study

- App: T (anonymized)
- Bound Tag: uid
- Leaked Free Tag: GroupCode




```
GET /361/grouplist_user.php?
xxx&uid=******&xxx
…
```

```
HTTP/1.1 200 OK
…
{
  "Member":[
   {
     "GroupCode": "******",
     "Name": "The friends",
     "LastActivityOn": "1669252927",
     "LeaderInfo":{…},
     …},
   …],
  …
}
```



Summary

- Systematic analysis of user tag spoofing attack in user tag sharing services
- UTSFuzzer: A novel fuzzing based security-vetting tool for automated identification of user tag spoofing risks.
- Revealing the landscape and severity of user tag spoofing attack in the wild & Responsibly notifying app developers to help them fix issues.

Thank you !

Shuai Li
lis19@fudan.edu.cn
August 2023