ARSpy: Breaking Location-based Multi-player Augmented Reality Application for User Location Tracking

Presented by

Si Chen (schen@wcupa.edu)
Liu Cui (lcui@wcupa.edu)

WEST CHESTER UNIVERSITY OF PENNSYLVANIA 1871

Information Security Center https://cs.wcupa.edu/ISC

Augmented reality (AR) applications

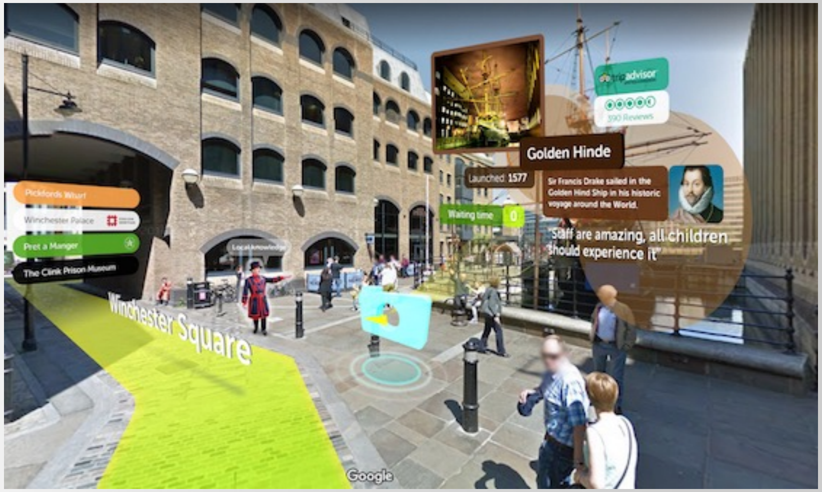
Augmented reality (AR) applications connect the physical world and the cyber world by overlaying digitally generated information on a user's perception of the real world.

Common AR applications use a **marker** to trigger AR content.

Location-based AR applications

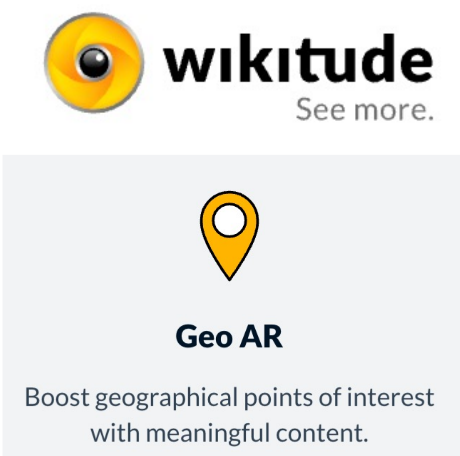
Location-based AR applications, in contrast, heavily rely on users' physical locations.

Typically, they use GPS (BLE beacons for the indoor environment) and simultaneous localization and mapping (SLAM) techniques to determine a user's location and to detect a device's orientation.



Location-based AR Software Development Kit (SDK)

Many **third-party AR services** such as Wikitude and Motive.io, provide a full-featured software development kit (SDK) that allows developers to build location-based AR applications without concern for technical details like *motion tracking, proximity calculation, or scale estimation*.

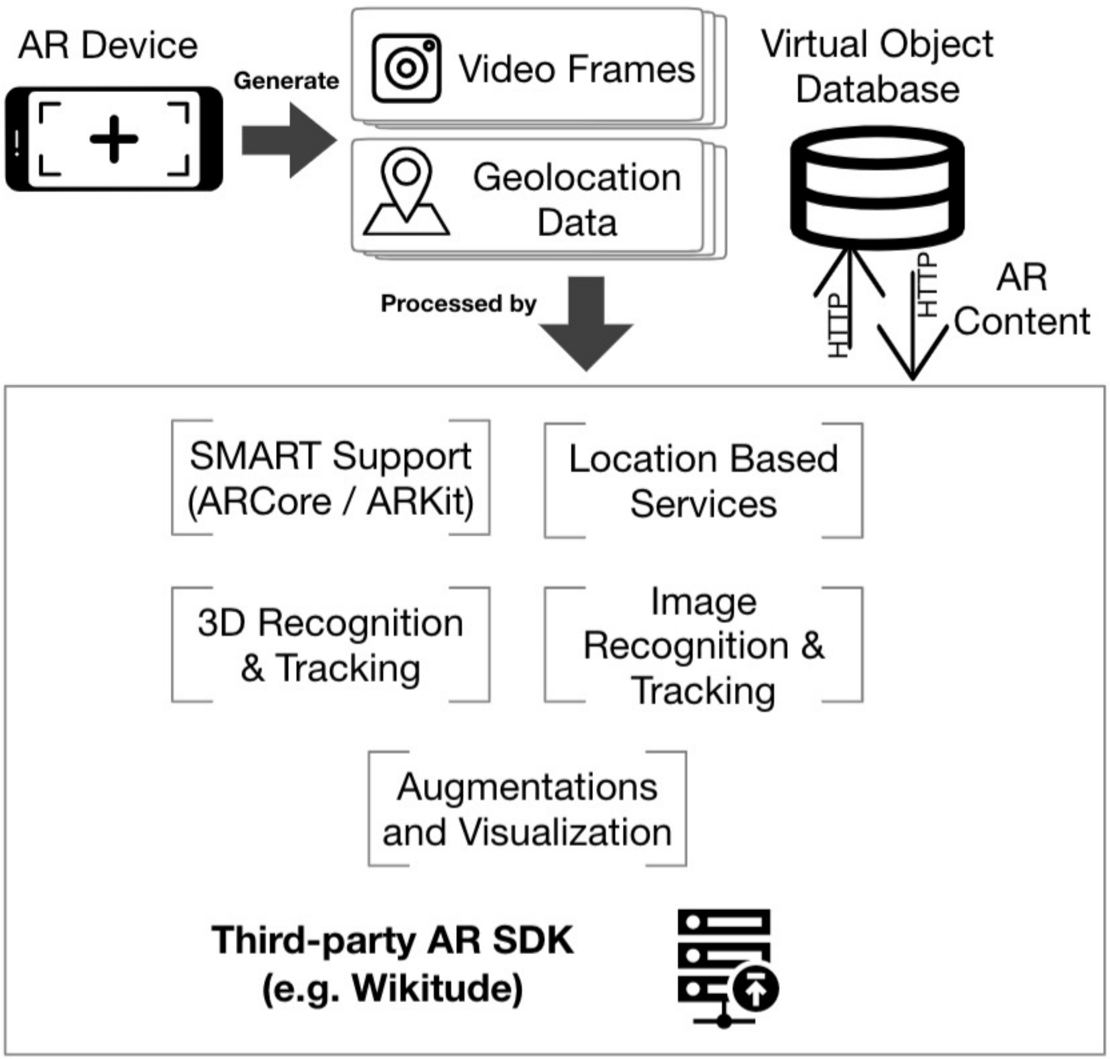


Location-based AR Software Development Kit (SDK)

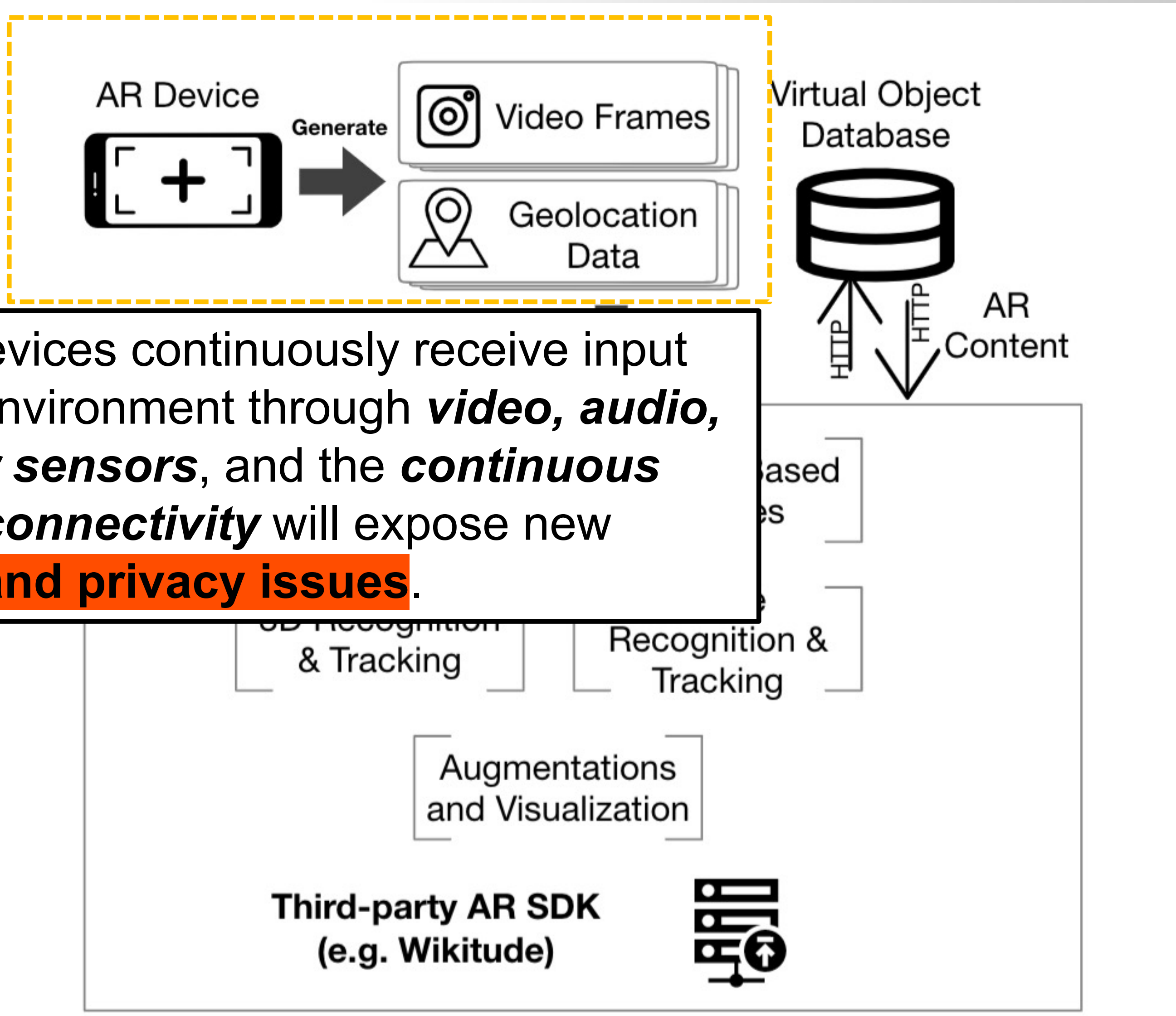
	Geo API	GPS	Content API	Cloud API	Cost
ARCore	✓	✓	✓	✓(Cloud Anchors)	Free
ARkit2	✓	✓	✓	✓	Free
AR Studio	✓	✓	✓	✓	Free
ARcrowd	✓	✓	-	✓	Free + Commercial SDK option
ARmedia	✓	✓	-	-	Free + Commercial SDK option
ARPA	✓	-	-	-	Free + Commercial SDK option
Metaio SDK (now Apple inc)	✓	✓	✓	✓	Free + Commercial SDK option
DroidAR	✓	✓	OpenGL or jMonkey Engine	-	Free + Commercial SDK option
HoloBuilder	✓	✓	✓	✓	Free + Commercial SDK option
Kudan AR Engine	-	-	✓	-	Free + Commercial SDK option
Vuforia	-	✓	with Vuforia Cloud	✓	Free + Commercial SDK option
Wikitude	✓	✓	with Wikitude Studio and Cloud Recognition	✓	Free + Commercial SDK option
Motive.io	with Unity	✓	with Unity	-	Free +Commercial SDK option
EasyAR	-	-	-	-	Free

Third-party AR SDK feature comparison

A location-based AR application with third-party SDK



A location-based AR application with third-party SDK

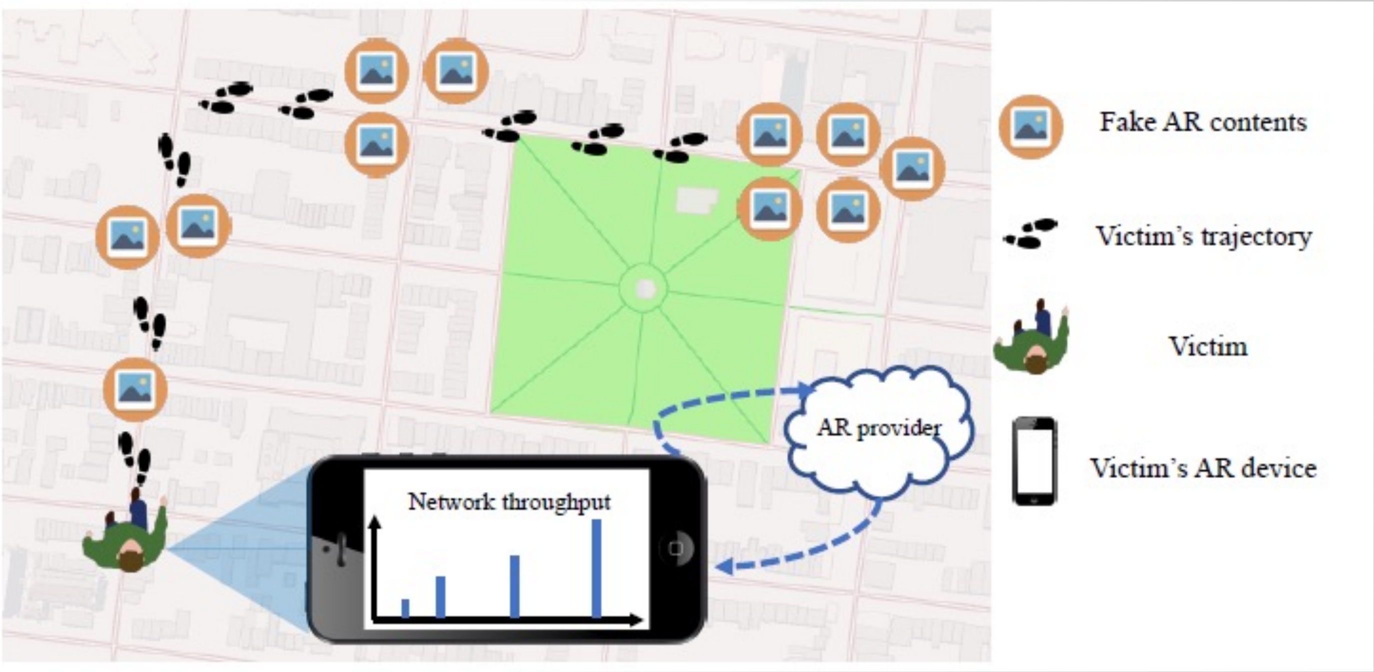


The AR devices continuously receive input from the environment through ***video, audio, and other sensors***, and the ***continuous network connectivity*** will expose new **security and privacy issues**.

Key Insight

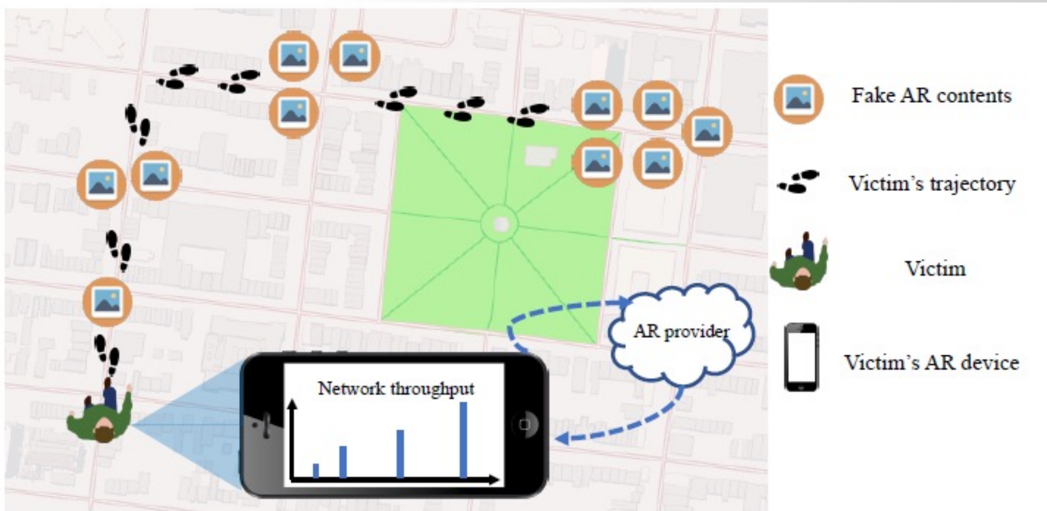
We explore the security threat model of AR devices and demonstrate a new **side-channel threat** caused by location-based AR applications' unique combination of a high volume of real-time data and outsourced geolocation processing.

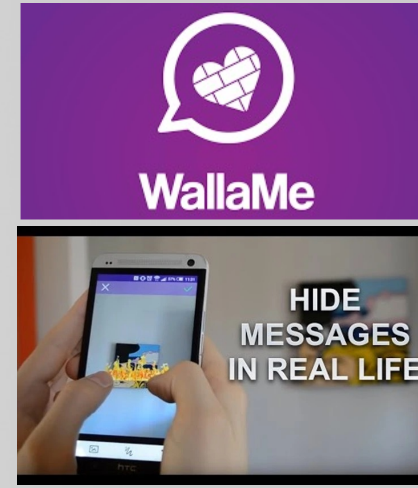
In short: If the downloading or uploading job are location-related, the location information may be leaked



Information Security Center

Adversary Model





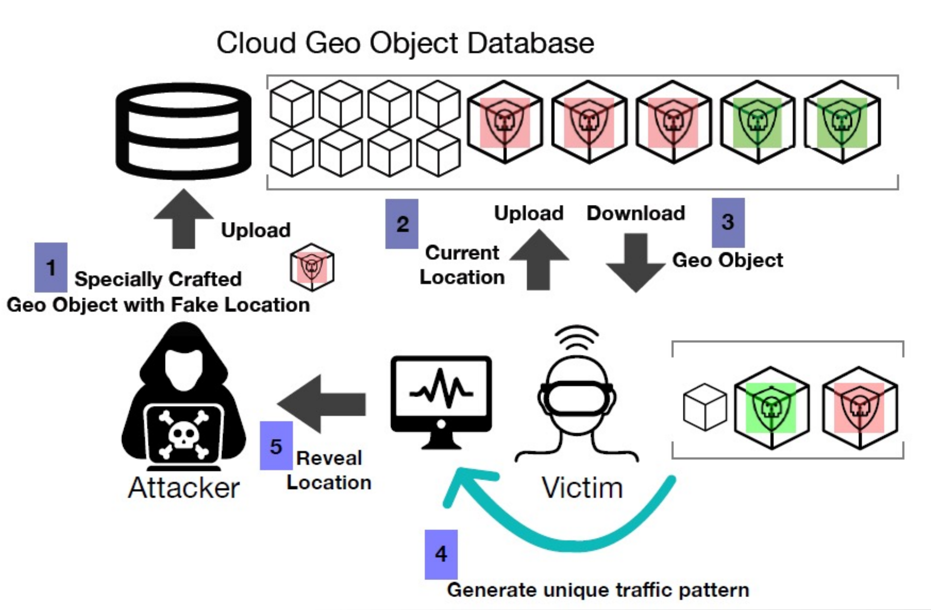
In this study, we consider a capability-restricted attacker that is aiming at **revealing the location of users** of some AR applications. These AR applications allow users to **publish or delete AR contents at any geolocation**. The attacker's capability is restricted in the following senses:

1. It only has the access right no more than that of a standard user -- except that it can manipulate (aka spoof) its geolocation
2. It can trick victims into installing its malicious applications that only require non-sensitive permissions.

Information Security Center

West Chester University

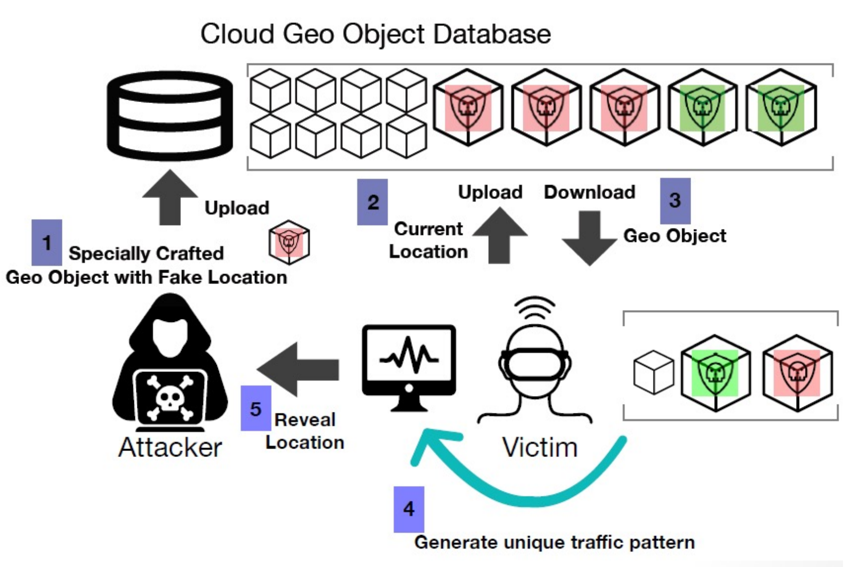
Overview of the Attack

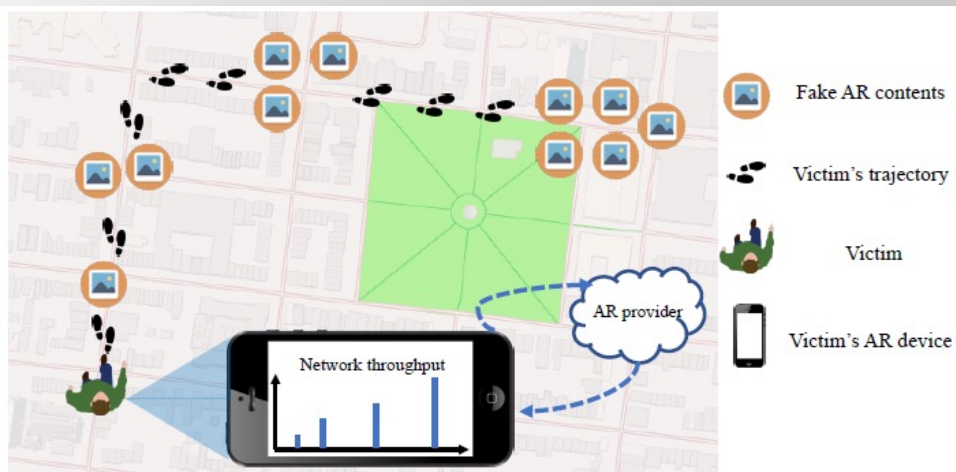


- There are three parts to the location-based side channel attack:
  - AR users (victim),
  - AR cloud database
  - Malicious user (attacker).

Overview of the Attack

- A complete attack can be divided into five steps.
  1. The attacker uploads several specially crafted geo-objects with a fake location to the cloud database.
  2. The victim posts his/her current location to query the database.
  3. The database returns several geo-objects back to the victim including the crafted objects.
  4. The victim downloads these objects and creates a unique traffic pattern.
  5. The attacker utilizes the malicious application to keep monitoring victim's **traffic pattern** and uses the reported pattern to reveal the location of the user.





Experiment -- Feasibility study

- Attacker needs to….
  - Trick victims into installing its malicious applications
  - Require non-location-related permissions to monitor network throughput of the targeted AR application
    - Permission needed: read phone status and identity
  - Manipulate its geolocation and deploy fake AR contents

Application	Number of installation
Tmall	1,000,000+
Youku	10,000,000+
Facebook	1,000,000,000+
Twitter	500,000,000+
Uber	100,000,000+

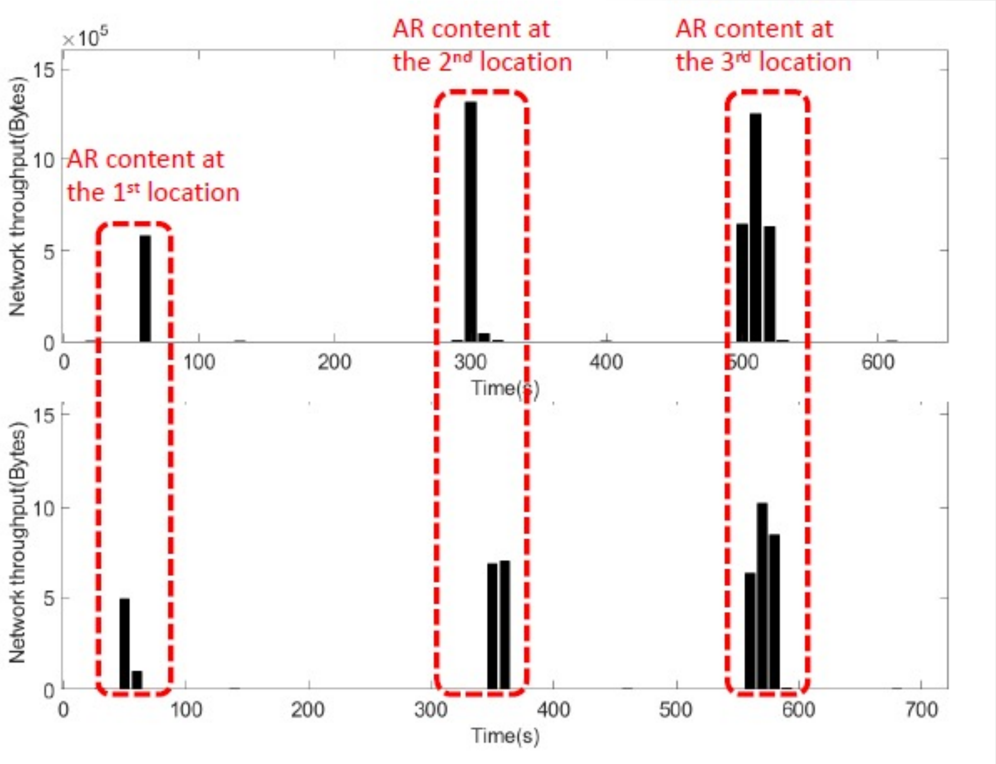
Popular mobile applications that ask for “readphone status and identity” permission.

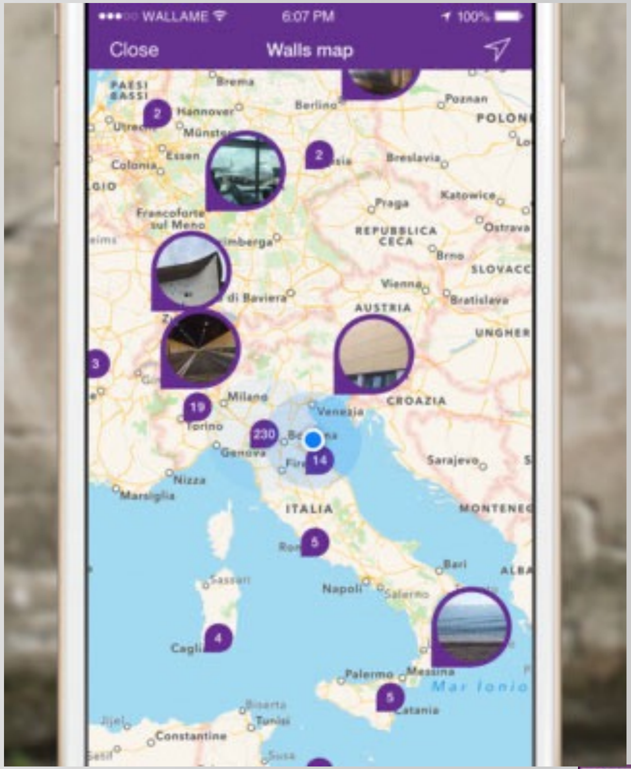
Experiment -- Feasibility study

- We first studied several state-of-the-art AR applications and SDKs (e.g., Google AR and Wikitude) and found that these AR applications and SDKs send local information (e.g., locations and images) to the server using a simple HTTP(S) GET requests.
- After getting the requests from the client, the AR server serializes returned information into a structured data using HTTP protocol and returns it to the AR application.
- Extended studies show that all existing AR applications and AR SDKs are based on the same mechanism.

Experiment -- Feasibility study

- Ask volunteers to use our self-made AR applications and walking among three locations



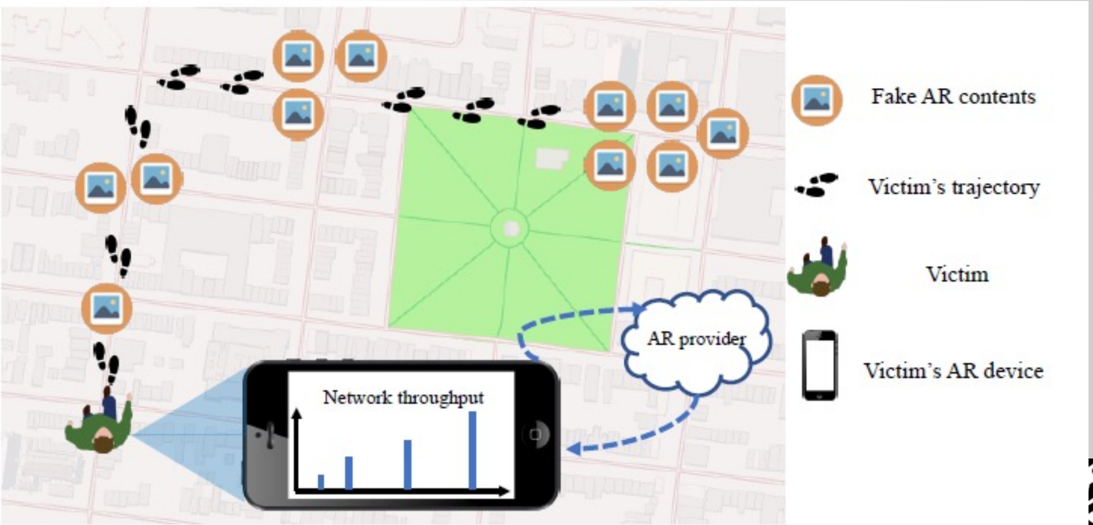


Information Security Center

West Chester University

Coarse-grained location detection.

- To locate the victim in a detected region, the basic idea is to cut the region into several non-overlapped areas.
  - Each area is a circle whose center is the location of AR contents and radius is the searching range of the AR application.
  - The size of AR content in one area is distinct from that in any other areas.
- It has two key limitations.
  - First, it cannot cover all locations in the small region since the searching area of each AR content is a circle.
  - Second, the localization granularity is relatively coarse.

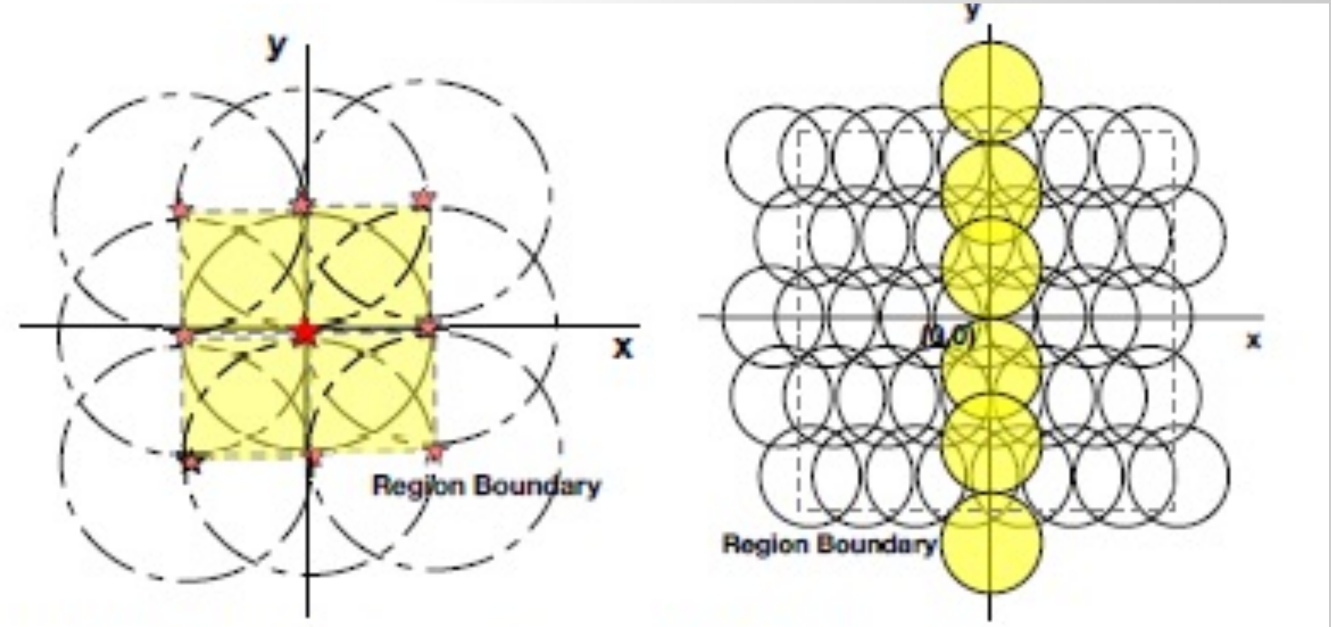


Information Security Center

West Chester University

Fine-grained location detection.

- AR contents deployment
  - Space partition & coverage
    - Divide the target area into four non-overlapping regions
    - Pinpoint the victim in the space to precisely one of the regions



(a) Illustration of space partition. (b) Strip-based space coverage strategy.

Fine-grained location detection.

- Attack procedure: AR contents deployment

  AR content size generation algorithm

    The sizes of AR contents at different geolocations $\mathrm{W} = (w_1, w_2, \dots w_n)$ be a super increasing sequence

$$w_k > w_{k-1} + \cdots + w_2 + w_1, \text{ for all } 2 \le k \le n$$

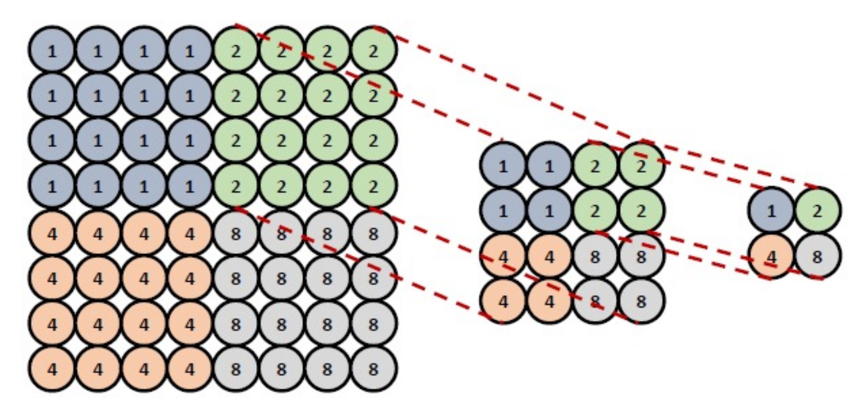
  Recursive region detection

    The size rises rapidly with a large n

Algorithm 2 AR content generator

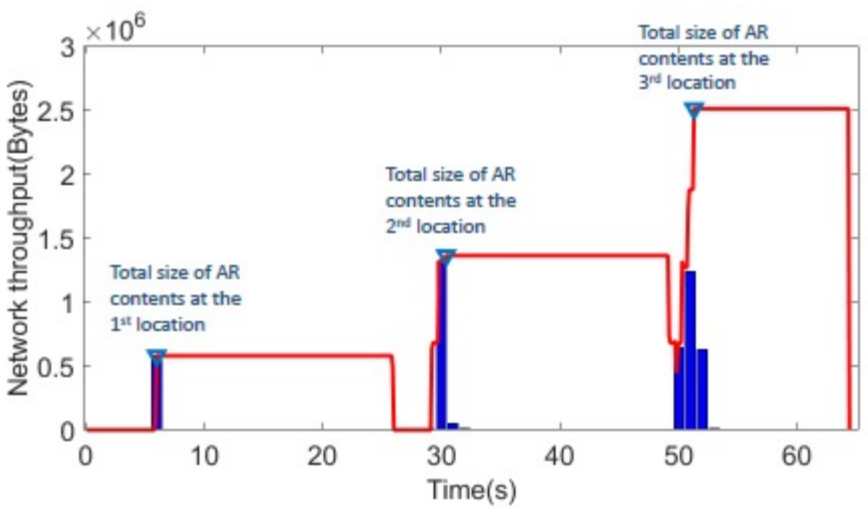
In: Size n, Constant number c
Out: Set W

1: **for** i `in range(1, ` n`)` **do**
2: $\quad w_i \leftarrow sum(w_0, w_1, \dots, w_{i-1}) + random(1, c)$



Fine-grained location detection.

- Attack procedure: Network traffic processing
  - Noise removal and throughput accumulation
  - Eliminate small traffic that cannot be caused due to AR contents
  - Accumulate the network throughput within a moving time window



Algorithm 3 Localization algorithm

In: A local maxima in accumulated throughput sequence $\mathbf{S}$, generated AR content size set $\mathbf{W}$

Out: Inferred location $\mathbf{X}$

$\mathbf{X} = \emptyset$
$n \leftarrow sizeOf(\mathbf{W})$
for i `in range(n, 1)` **do**
  if $S > w_i$ **then**
    $\mathbf{X} = \mathbf{X} \cup x_i$
    $S \leftarrow S - w_i$
  else
    $x_i \leftarrow 0$

Implementation

- We built a real testbed in order to effectively evaluate the attack methods we propose.
  - An Android application for monitoring network traffic
  - An Android application for imitating the behaviors of AR applications
  - A customized location provider for location spoofing
  - A back-end AR server

- Simple graphical user interfaces (GUI) are designed to help subjects to collect data.

Implementation

- **Network traffic monitoring:** The core part of our system is accurately monitoring the network traffic of a specific application.
- To achieve this goal, we studied the feasibility of monitoring network traffic on Android platform:
  - ***NetworkStatsManager*** can provide access to network usage history and statistics of other applications, which enables an attacker to implement a listener in another application on a victim's device.
  - We created a **background service** that can log the total network usage every second. The throughput of each second was acquired by calculating the difference between neighboring entries in the log file.

Implementation

- **Location spoofing:** Location spoofing is used to generate mock locations, so that the attack can deploy fake AR contents at any location without physically being there.
- We found it is possible to add customized location -- the attacker needs to enable ``Allow mock location" option in the developer options of their Android device before getting access to the mock location API.

Information Security Center

West Chester University

Performance Evaluation

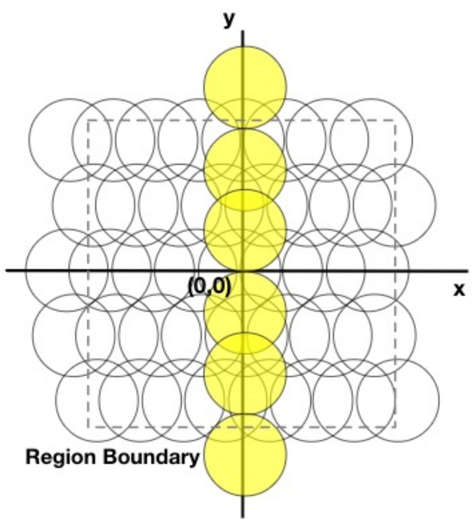
To evaluate the performance of our attack model, we conducted various experiments on our testbed on a campus,

For coarse -grained location detection:
On the path:

- We uniformly picked 8 locations on the map.
- The distance between neighboring locations was about 60 meters.
- The searching range of each location was set to different values to evaluate the performance of our attack strategies

Performance Evaluation

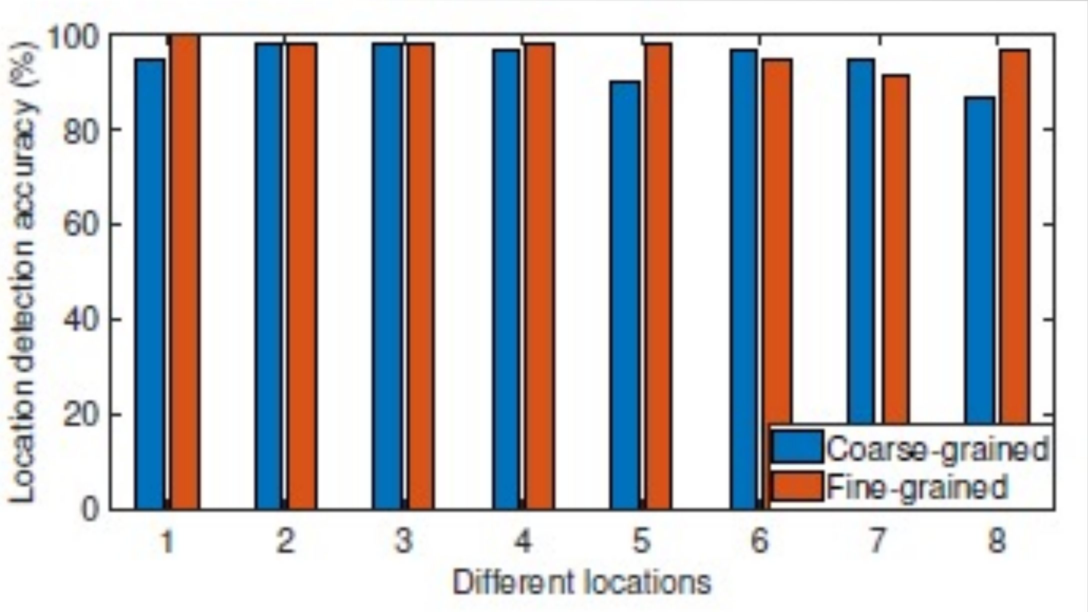
To evaluate the performance of our attack model, we conducted various experiments on our testbed on a campus,



For fine-grained location detection:

- the searching radius is about 45 meters
- We deployed AR contents whose total sizes follow the rule of super increasing sequence at each location.
- The minimal size of deployed AR contents was also 1 KB.

Performance Evaluation



Performance of a single location detection.

Evaluation results show that

- Our coarse-grained location detection strategy can locate the victim in non-overlapped areas with a mean accuracy of about 94.6%
- Fine-grained strategy can provide a better average accuracy of about 97.1%

Performance Evaluation

There is a trade-off in how densely the attacker should deployed the AR contents in a small region

- If we deploy AR contents at many different locations, we can estimate victim's locations with a better granularity
- But the location detection accuracy may not be good due to inaccurate GPS coordinates

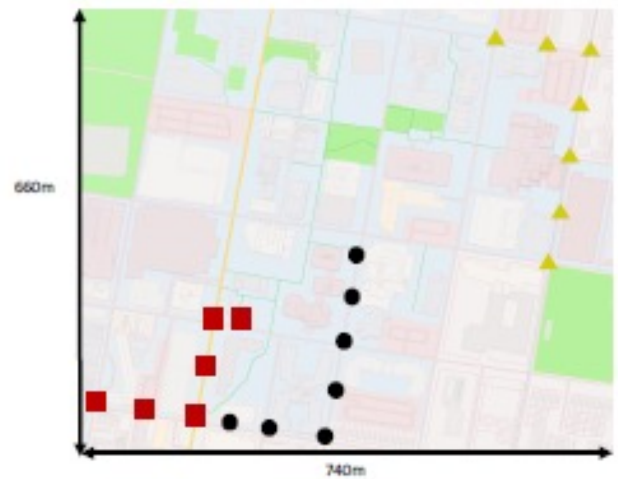
The fewer locations where we deploy AR contents, the better the detection accuracy is expected to be, but more details of the victim's trajectory are lost

Distance (meter)	70	27	13
Accuracy	100%	98%	60%

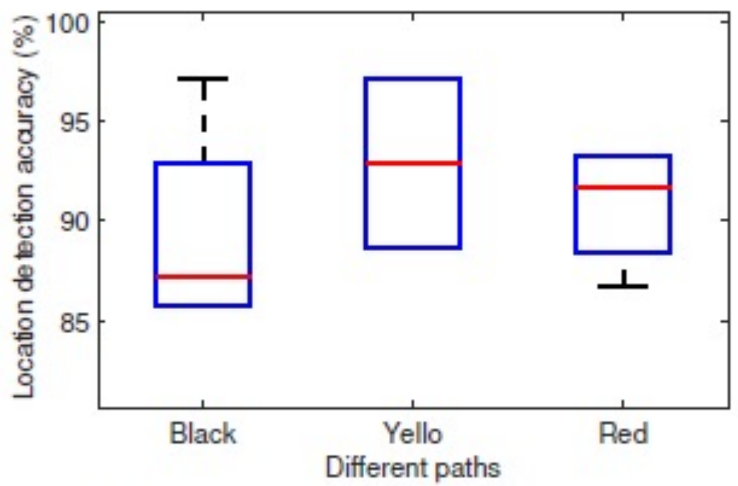
Location-detection accuracies with different distances between neighboring locations.

Performance Evaluation

- Performance on different paths

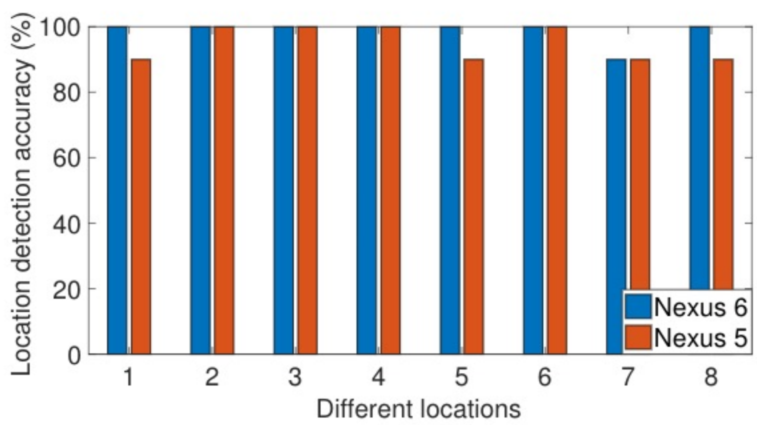


(a) Deployment of AR contents at three paths.



(b) Average location detection accuracy.

Performance Evaluation



Location-detection accuracies on two devices

We found that Nexus 6 has a better performance than Nexus 5.

- **The sampling rate of GPS data** -- every 1 second vs. more than 1 second
- **Maximal delay of receiving the next GPS coordinate --** 12 vs. 22 seconds

Performance Evaluation

Percentage of top N locations

The top N locations inferred from human mobility data can be used to reveal the identity of a user

Our attack method is able to deduce at least top 4 locations for more than 50% of the user

Achieve 86% detection rate for the top two (and above) locations. → home and workplace

Mitigations

1. SDK providers or developers can deploy and maintain an active cache with variable size to store the AR contents on the client side.
2. Add limitation → Any AR user cannot deploy too many AR contents at a single location. Meanwhile, the size of each AR contents should not be too large.
3. Another method is to further limit the permission of network traffic monitoring on the victim's devices, which means third-party applications cannot get the network traffic information.

Future Directions

- Protecting Visual Information in Augmented Reality

  Visual AR applications raise serious privacy concerns and incur new challenges to protect user privacy

  The original visual information in the scene can be covered by malicious virtual objects

Information Security Center

West Chester University

Q & A