Secret Stroll

Google Maps

Opinion | THE PRIVACY PROJECT

Twelve Million Phones, One Dataset, Zero Privacy

By Stuart A. Thompson and Charlie Warzel

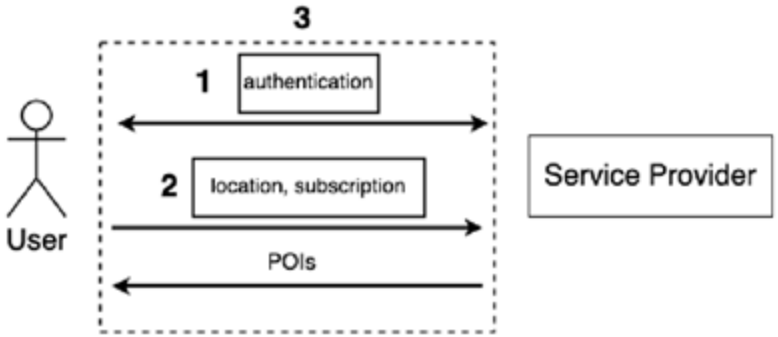
DEC. 19, 2019

Give this article 527

Secret Stroll

Your task is to **build** and **evaluate** a privacy-friendly application for finding points of interest (e.g., gym) around a location. Secret Stroll:

- has no continuous location sharing
  - query-based system
- is subscription-based
  - 1,5.- for gym, 5.- for bars, …
  - no need to implement a payment system! Assume something and work around it.
- (of course) is privacy-oriented
  - minimal leakage possible
    - **3 steps == 3 potential leaks == 3 parts**
  - while preserving functionality



Timeline

Starts 4th April, 2025 (today)
Deadline 30th May, 2025 at 23:59
Duration 8 weeks

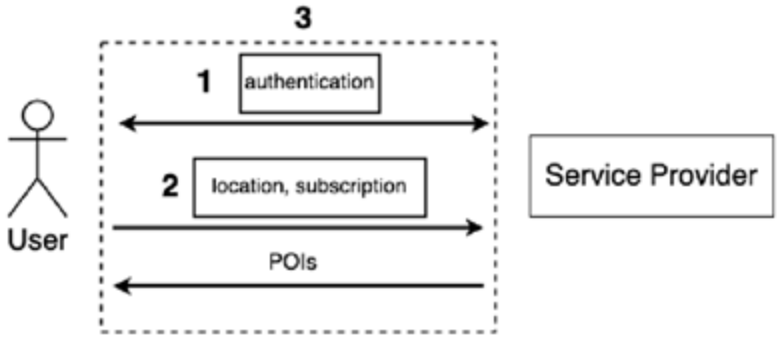
Suggested timeline

1st part Authentication (Weeks 1–3)
2nd part Queries (Weeks 4–5)
3rd part Responses (Weeks 6–7)
(Polish submission on week 8)

Deliverables

Report + code.

1st part: Authentication

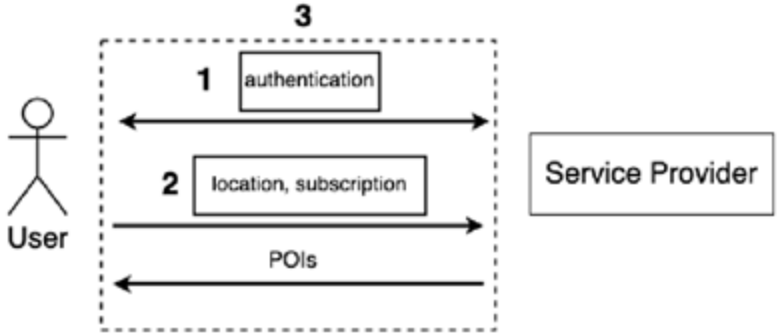


Attribute-Based Credentials (Weeks 1–3)

1. **Implement** the ABC tool: Pointcheval and Sanders
    a. Implementation using petrelic (crypto library)
    b. Implement tests
    c. Code quality is evaluated
2. **Integrate** in Secret Stroll
    a. Research! Bad use of ABC may make the whole thing insecure.
3. **Evaluate** your implementation

Tip for time management: Once you finish part 1, read part 3 (which uses part 1 as black box) and start the data collection.

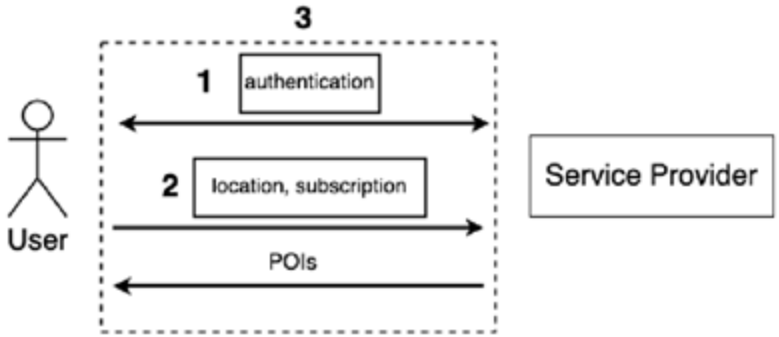
2nd part: Queries



Location Privacy (Weeks 4–5)

1. **Threat modelling** !
2. Using simulated data, **evaluate** the privacy of users towards the service provider.
   a. Simulated data = queries (IPs, subscriptions, …) and responses (POIs, …)
   b. Privacy of users = ?? It can be activity, where they live, …
   c. Attack(s) to be designed and implemented
3. **Propose** and **implement** a defense

3rd part: Responses



Tracking (Weeks 6–7)

1. **Collect** metadata for network fingerprinting
    a. network traffic metadata only, not the content of the reponses
    b. Tip: start early! As soon as you're done with part 1
2. **Infer** the location that a user is querying for
    a. location = cell ID (simple grid)
    b. use a classifier
        i. This is not an ML class, do not worry about getting the best result ever
    c. evaluate your attack
3. **Discuss** counter-measure(s) in the report
    a. no need for implementation

Submission

Finalize code and report (Week 8)

1. Report
    a. There are guidelines in the handout (often formulated as questions).
    b. Remember to add a paragraph saying who did what
2. Code
    a. To be submitted.
    b. Code quality will be evaluated for part 1.

Documentation available to you

Server side:

Open a shell

```
$ cd cs523/secretstroll
$ docker exec -it cs523-server /bin/bash
(server) $ cd /server
(server) $ python3 server.py setup -s key.sec -p key.pub -S restaurant -S bar -S dojo
(server) $ python3 server.py run -D fingerprint.db -s key.sec -p key.pub
```

Client side:

```
Open a shell
$ cd cs523/secretstroll
$ docker exec -it cs523-client /bin/bash
(client) $ cd /client
(client) $ python3 client.py get-pk
(client) $ python3 client.py register -u your_name -S restaurant -S bar -S dojo
(client) $ python3 client.py loc 46.52345 6.57890 -T restaurant -T bar
```

Big project -> lots of things.

1. Code
    a. README
        i. How to use the code (e.g., run part 1)
        ii. Setup VM
        iii. Read it once fully, come back to it if needed
    b. Skeleton
        i. Some to complete, some not to modify (details in next slides)
2. Handout
    a. Report template in the github
    b. Detailed version of this presentation
    c. Guidelines for the reports (questions)
3. ABC guidelines
    a. Detailed version of the slides seen in class
    b. Keep them open when you code

Skeleton

Parts 1 and 3:

1. *credential.py*—Source code that you have to complete.
2. *stroll.py*—Source code that you have to complete.
3. *client.py*—Client CLI calling classes and methods defined in stroll.py.
4. *server.py*—Server CLI calling classes and methods defined in stroll.py.
5. *serialization.py*—Extends the library jsonpickle to serialize python objects.
6. *fingerprinting.py*—skeleton for Part 3.

7. *requirements.txt*—Required Python libraries.
8. *docker-compose.yaml*—docker compose configuration describing how to run the Docker containers.
9. *docker/*—Directory containing Docker configurations for running the client and the server.
10. *tor/*—Intentionally empty folder needed to run a Tor server.
11. *fingerprint.db*—Database containing POI information for Part 3.

The directory *privacy_evaluation* contains files for the Part 2.

Have fun!

Google Maps

Secret Stroll