CS-523 SecretStroll Report

Wicky Simon, Nunes Silva Daniel Filipe

Abstract—**This project consists of three parts, all related to a central topic : Privacy in a location based service. The first part makes use of Attributes based credentials (ABC) and zero-knowledge proof to anonymously query a server for nearby Points of Interests (PoIs), while proving the ownership of a subscription. The second part we evaluate the loss of privacy induced by IP-level metadata. The last part of this project is dedicated to trying to infer location of query author, based solely on metadata.**

I. INTRODUCTION

The aim of the first part of this project is to build a privacy-friendly location based service, SecretStroll, which allows users to query a database about nearby PoIs while remaining anonymous. Since this service is not free, every query must provide the proof of an active subscription. Each PoIs is part of a category such as *Restaurant*, *Cinema*, *Train station*, etc. Each of these category are part of a subscription and a user can have many of them, which naturally leads to ABC. A user joining the service first register himself with the category he paid for and then sign each of his query anonymously, revealing only the categories he want to get PoIs from.

II. ATTRIBUTE-BASED CREDENTIAL

The base of the ABC scheme for this work is described in Section 6 of [1]. This scheme has been used as follows :

- The messages mentionned in [1] are attributes, representing categories of PoIs. The server creates a public key large enough to encode all of these attributes.
- When a user wants to register, she commits to attributes she paid for, that will remain hidden. She uses a zero-knowledge proof (ZKP) to prove that she did so correctly. The server checks the proof, and sign this commitment that she will then use as credential.
- When a user makes a query, she choose the attributes (i.e the categories of PoIs) that she want. She then makes use of her credential to prove she has a subscription for the queried attributes, using a ZKP. She also uses this credential to sign a message representing her current location using the Fiat-Shamir heuristic described in [2] and detailed later.

In order to make the ZKP non-interactive, we use the Fiat-Shamir heuristic. The challenge that the prover would send is replaced by a cryptographic hash of all known values. During the registration, the challenge is composed of the committed attributes, the commitment used for the ZKP and the server's public. To sign the message for a query, it is simply added to the other values constituting the challenge.

This overall approach leaks only the number of attributes chosen, but it is necessery for the server to correctly compute the ZKP. Otherwise, hidden attributes cannot be inferred and every credential is unique.

A. Test

The system is tested with six different tests :

1) *test_generate_ca()* : The server's certificate generation is tested with a simple structure test, to see if all keys have the right lenghth.
2) *test_credentials_difference()* : Two requests are created with the same input parameters, a registration is perforemd with both of them and two credentials are created. To preserve anonymity, the requests and the built credential should be different.
   Two credentials created from the same request should also be different.
3) *test_tampered_credential()* : A request with invalid credential should be rejected. In this test, a valid credential is tampered to be invalid.
4) *test_correct_credential()* : A request with a correct credential should be accepted.
5) *test_correct_credential_no_attributes()* : A request revealing no attributes should be valid.
6) *test_wrong_revealed_attr()* : A request revealing unobtained attributes should be invalid.

The effectiveness of these tests could be assessed using a standard metric such as branch coverage or coverage. However, these metrics are not complete and will not discover every bug present in the code.

B. Evaluation

Evaluate your ABC: report communication and computation stats (mean and standard deviation). Report statistic on key generation, issuance, signing, and verification.

III. (DE)ANONYMIZATION OF USER TRAJECTORIES

A. Privacy Evaluation

We evaluate the privacy risks using simulated data of two hundred users who made use of the application hundred times each in average over twenty days. We assume that no mechanism to hide any kind of data is used, i.e. data is sent in cleartext inside standard IP packets. Any malicious adversary could hack the application servers or sniff the network between a user and the server in order to retrieve similar datasets which include IP addresses, locations, query types, timestamps and responses. According to [3], the IP address or a set or IP addresses are relevant attributes because they can be linked to a given user since they are persistent for a certain duration. Moreover, combining IP addresses with location and time data makes sense since users often keep their habits that they may

share with very few people [4], i.e. they may work during the day at their office, be back home in the evening and do an activity at some specific location. Therefore, we deduce that this implementation leak senstive informations about the users. As a proof of concept for breaching users privacy, we try to infer work and home locations of users as well as habits in their activities.

Provide a privacy analysis of the dataset. You should explicitly state your assumptions, adversary models, methods, and findings.

B. Defences

Propose a defence that users of the service could deploy to protect their privacy. You should state your assumptions, adversary models, and provide an experimental evaluation of your defences using the datasets and the grid specification. You should also discuss the privacy-utility trade-offs of your defence.

IV. CELL FINGERPRINTING VIA NETWORK TRAFFIC ANALYSIS

A. Implementation details

Provide a description of your implementation here. You should provide details on your data collection methods, feature extraction, and classifier training.

B. Evaluation

Provide an evaluation of your classifier here – the metrics after 10-fold cross validation.

C. Discussion and Countermeasures

Comment on your findings here. How well did your classifier perform? What factors could influence its performance? Are there countermeasures against this kind of attack?

REFERENCES

[1] D. Pointcheval and O. Sanders, "Short randomizable signatures," Cryptology ePrint Archive, Report 2015/525, 2015, https://eprint.iacr.org/2015/525.

[2] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," vol. 263, 03 1999.

[3] V. Mishra, P. Laperdrix, A. Vastel, W. Rudametkin, R. Rouvoy, and M. Lopatka, "Don't count me out: On the relevance of ip addresses in the tracking ecosystem," *HAL*, 01 2020.

[4] P. Golle and K. Partridge, *On the Anonymity of Home/Work Location Pairs*, 2009.