WeTrace

A privacy focused mobile COVID-19 tracing proposal

April 2, 2020

v1.3

Alessandro De Carli
a.decarli@papers.ch

Andreas Gassmann
a.gassmann@papers.ch

Matteo Cortonesi
cortonesi@me.com

***Abstract*— TODO ABSTRACT**

CONTENTS

I Introduction 1
I-A Related Work 1

II Technical Implementation 1
II-A How the system works 1

III Privacy Properties 2
III-A Privacy from Snoopers 2
III-B Privacy from Contacts 2
III-C Privacy from Authorities 2

IV Trade Offs 2

V Discussion 2

I. INTRODUCTION

TODO INTRO

A. Related Work

TODO RELATED WORK

II. TECHNICAL IMPLEMENTATION

The system we propose uses concepts used in privacy-first blockchain projects such as the ZCash project (TODO reference). Using a combination of those concepts in conjunction with Bluetooth Low Energy technology we are able to come up with a system, that has the following properties:

- Data of close contacts stays only on the device, is never shared with a central server.
- Users locally collect close contacts (<2m) on their device, together with the timestamp of when the contact happened and rough geolocation of where the contact happened.
- In case of an infection report, only the users that have been in close contact will be notified, the central server cannot read those messages and has the sole purpose of relaying the message.
- The user reporting the infection can decide whether he/she wants to:
  1) report only the infeciton to the close contacts of the last 14 days
  2) report the above *and* the timestamp of when that close contact has taken place
  3) report the above *and* the geolocation of where the close contact has taken place

A. How the system works

How the system Works is best explained with an example. For this example we have User A with Device A, User B with Device B and User C with Device C.

1) Every device that installs the WeTrace app generates an asymmetric key pair using elliptic curve cryptography. For this examples sake PK_A stands for public key of Device A and SK_A stands for secret key of Device A. So in this step we generated PK_A, PK_B, PK_C and respectively SK_A, SK_B, SK_C.
2) Every device starts broadcasting their PK_* this is also their unique identifier to its surrounding devices.
3) When now 2 devices (i.e. A and B) meet in close contact, Device A knows PK_B and Device B knows PK_A. Besides the PK_*, both devices also store a timestamp and the geolocation of where the encounter happened.
4) When User A is now infected and wants to report it, the Device will go through the list of close Contacts and encrypts a message with the public key of every contact. In our case, it will be encrypted once with PK_B because this was our only contact. All those messages will be sent to the central backend that will relay them to all devices. The messages will contain the data that User A chose to share, so either only the fact that an infection happened, or additionally when or even where it happened. Important: Only the reporting user decides if he/she wants to share this information.
5) Device B and Device C receive from the backend a notification telling them that new reports have happened. Device B will then try to decrypt every message with SK_B and will eventually find out that a message was directed at him/her. Device C will do the same, however because no message was encrypted with PK_C, no message can be decrypted.

Now that the crypto system is outlined the remaining ”privacy issue” that is unsolved is the fact that someone could start tracking a users' location by simply scanning his/her advertising packets. Now the mitigation there is very simple, for simplicity's sake we said that in step 1 the device generates a keypair, however what actually happens is that the users generate a so called master seed. This master seed is used to deterministically derive an unlimited number of keypairs. This means the user will actually be changing the key in a specified period of time (like e.g. 30 min), making him/her only traceable with that public key only for that time frame. The elegance of this system is that the user still only stores 1 master seed and basically tries then with all of his/her keys from the last 14 days to decrypt the message.

III. Privacy Properties

For the privacy properties we are relying on the properties defined by Cho et. Al in there Paper “Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-offs”:

1) Privacy from Snoopers
2) Privacy from Contacts
3) Privacy from Authorities

We are enforcing those privacy property the following way.

A. Privacy from Snoopers

Since the very idea of the system is to broadcast a signal so that others can see the contact, snoopers will also be able to see the public keys being advertised. However since those “identities” will be valid only for a very limited time (that can be further shortened if this is an issue), the user will not be more exposed to snooping than he already is with a wifi enabled device, that is happily broadcasting its mac address without any consent or knowledge of the user (yes even with Mac randomization). Also, only if the snooper was in real close contact he will receive “a” notification of infection, but will not be able to track down from which user this notification was sent. This is because for the notification it does not matter what the snooper collects, but only what the user's device collects and acknowledges as a close contact. So this property is covered in a close to perfect manner by the proposed system.

B. Privacy from Contacts

This is also a very similar argument to what we have in the case of “Privacy from Snoopers” above. Close contacts will receive “a” notification if a user chooses to broadcast his/her infection, however the close contacts will not know from whom this message came. The only way how this can be inferred would be if the user only had a close contact with a single contact during the last 14 days and that contact then broadcasts that

C. Privacy from Authorities

As discussed earlier, due to the fact that only encrypted messages are being sent to the server, the server owner can not get any personal data out of these messages other than the fact that an infection happened.

IV. Trade Offs

There are a couple of tradeoffs to consider. In the end we are trading off central analysis of data for privacy. Central analysis of data can be advantageous if authorities want to detect ”hot spots” of infections, or also perform page ranks on possible next infections. One of the requirements we have is that both devices need to record one-another, otherwise it will not be possible to encrypt the message for the receiver of the encrypted report. This requirement however will be met because by WHO standards someone is defined as close contact if: TODO WHO DEFINITION

V. Discussion

The proposed solution solves all the challenges that have been highlighted by Cho et Al. By these means the solution ticks all boxes. We highlighted the tradeoffs this method has, but believe that if privacy is a requirement the proposed solution is very elegant and easy to implement.

The important bit will be that the community agrees on a standard on how to trace the infections and then make sure that all developers are using the same standard/protocol so that the system can profit from a network effect. It's imperative that this initiative is following a open source philosophy, so that the various app developers can cooperate.