Fully ANoNYmous and decentralized pandemic witness protocol (FANNY)

Samir Amellal[a], Nicolas Guy[b], Vincent Bouillon[c], Benoit Fleuriaud[d]

[a]*M.Sc in Data and Digital Science Skema Business School/University of Lille*
[b]*Ph.D in Robotic and Automation ISAE-SUPAERO France*
[c]*Ph.D in Philosophy University of Paris X France*
[d]*Engineer in IT and Security École Nationale Supérieure d'Ingénieurs de Caen France*

Abstract

In a context of global pandemic, many countries are looking for a reliable solution to alert people that they had a narrow contact with a Covid-19 positive patient. This solution should respect the privacy and it should be compatible with a wide range of devices. We propose a fully decentralized tested solution based on a peer to peer protocol with local storage and processing. It offers a direct private and anonymous communication between two devices up to 15 days after a narrow contact [8].

Keywords: localStorage, WebRTC, Token, SHA256 hash function, peer to peer, anonymous, privacy, decentralization, Covid-19

1. Our motivation

The main goals of this approach are :

1. The anonymity,
2. The transparency,
3. Make sure that it works without any user file or data base,
4. Make it impossible to identify who was ill,
5. Make it impossible to rebuild a social graph,
6. Make the tractability impossible,
7. Make the approach reliable and inter-operable,
8. Make the approach accessible,
9. Make the scale up possible.

Email addresses: s.amellal@gmail.com (Samir Amellal), guy@soyhuce.fr (Nicolas Guy), vincent.bouillon@vib.solutions (Vincent Bouillon), benoit.fleuriaud@soyhuce.fr (Benoit Fleuriaud)

May 5, 2020

2. Introduction

The main objective is to develop a solution that provides a witness of a narrow contact between two people while strongly respecting their privacy. The second one is to make it accessible for a very large part of people. For this reason, the protocol described below has been built so as to be compatible with a large number of mobile phones in a hybrid application, as this category of devices is hugely adopted around the world.
It is currently possible to observe two kinds of technical approaches. The first one is a centralized solution that uses storing, processing, management and dispatching of data from a server. The second one is based on the usage of the Bluetooth technology [5, 6, 7]. This is yet not the most reliable solution to establish a peer to peer connection, especially between two devices working with different operating systems. Furthermore the devices that use old Bluetooth versions [1] are limited regarding the number of simultaneous network connection they can handle.
The resurgence of a situation of a global pandemic or of a local epidemic could be more frequent in the upcoming years and a global long-term solution could become necessary to save many lives [9, 4]. This solution would have to respect conditions of decentralization, privacy, accessibility, compatibility, low cost of development and implementation. Above all the acceptance of the population will be critical.
The right protocol must be the one that offers the best compromise between accessibility, efficiency and development cost. As the pandemic question is intrinsically global, a reliable medium and long term solution cannot be local and designed by one country or a small group of countries. The goal here is to propose an easy and accessible solution for a generalized use that could be implemented at a global scale.
A narrow contact between two people increases significantly the risk of a virus contamination and propagation. The definition of a narrow contact in this document considers the distance and the duration of the interaction, the shorter the distance and the longer the duration, the narrower the contact.

3. Witness of a narrow interaction

The following process is based on the WebRTC API [3] and a local processing and storage. Its originality relies on an unlimited distance peer to peer standard in association with a deported processing and storage client side, without user tracking [10]. In this context, the server contribution is minored because it just acts as an intermediate, as known as a broker, to establish a direct communication channel between two mobile phones.

[1] On oldest versions of Bluetooth the piconet adhoc network restrain the number of simultaneous connected devices. Some security questions could rise the untrustness of Bluetooth technology.

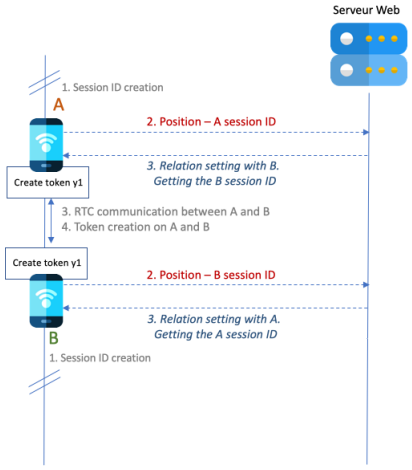

Figure 1: Decentralized creation of a witness of narrow interaction

Detailed protocol of decentralized creation of a witness of narrow interaction (Fig. 1) :
Step 1 - Users A and B both download the application on their mobile phones. When they launch the app, a temporary session ID and a contact ID are locally created using the UUIDv4 [1]. Those session ID and contact ID is deleted and forgotten every time the application is turned off.
Step 2 - A and B regularly push their own position (latitude/longitude) to a server in order to check if another phone is nearby. This position is not stored by the server and does not feed any positions history. This last point ensures that no geographical tracking could be done afterwards and participates in privacy enhancement. The mobile phone location can be provided directly by the device using a position request [GeolocationPosition] from a Geolocation API.
Step 3 - A and B are close to each other (meaning in a predefined radius), the server pushes the correspondent respective session IDs to both A and B. The first one receiving its correspondent session ID (let's say A) is arbitrarily in charge of establishing the WebRTC communication channel between the two mobile phones[2].
Step 4 - The one in charge of the channel establishment is also in charge of crafting a token (Fig. 2). Such a token designates a narrow contact between two people and must therefore be unique, as every geographical and temporal contact is unique in real life. This token is crafted by concatenating the two contact IDs (in alphabetical order) and then applying a SHA256 [2] hash function on it.Everything is done client side, the contact ID don't transit by the server, they exchanged and processed directly by mobiles phones.

[2]The WebRTC direct connection mandatory acceptation - peer to peer - between mobiles phone A and B is validated by the acceptation of each new app session launched on the mobile phone. As described above it uses the session ID to establish a direct connection.

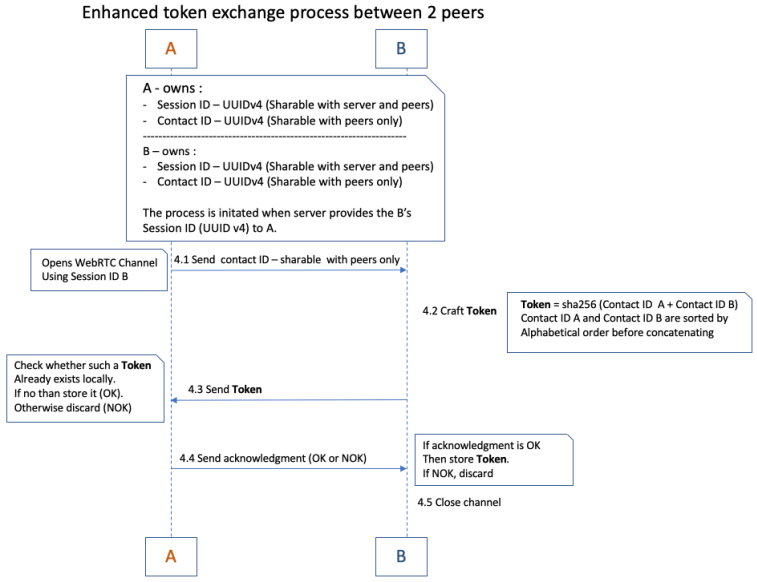

Figure 2: Representation of the Token creation process

Step 5 - Once transferred to B, the token is immediately stored locally by B. An acknowledgment is returned to A who can now store the same token locally. This way, we ensure that both mobile phones have the same information of a narrow contact. Note that no one saves the UUIDv4 of its channel correspondent. If B was initiating a WebRTC channel with A at the same time (since channel establishment and token crafting are not instantaneous), trying to store a token that already exists is not possible and the process is aborted.
Step 6 - This shared token is stamped with an expiration date of 15 days [8] beyond which it is simply removed from the mobile phone, as would do an expired cookie in a web browser[3].

4. The *a posteriori* and anonymous contamination risk alerting

The second feature of this approach is to inform a user when he had a narrow contact with another user declared *a posteriori* virus positive.
In order for a positive patient to inform other users that were effectively in a narrow contact with him that they might have been exposed, this positive patient must release his own tokens for a comparison. If at least one identical token is found on another mobile phone, it means that he has been exposed.
In this context, the server is in charge of gathering the tokens of the positive declared patient. The main goal is to do it anonymously, and the server can't trace the origin of the tokens and it can't know the destination regarding the process of token creation described above.

[3]The hour to delete an expired token must be always the same in order make the exact moment of the narrow contact impossible to deduct. The date of expiration is not an exact delay of 15 days, it must be 15 days plus the number of hour, minutes and seconds necessary to achieve the fixed suppression hour.

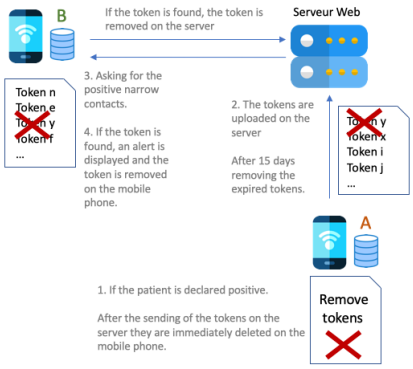

Figure 3: Anonymous contamination risk alerting

Detailed process of anonymous contamination risk alerting (Fig. 3) :
Step 1 - The holder of the mobile phone A declares, after a few days, that he was probably positive and contagious when few days ago some narrow contacts triggered the token creations.
Step 2 - Each token stored in mobile phone A (positive) is uploaded on the server and immediately deleted on the mobile phone A. Each token is associated with its expiration date.
Step 3 - If a user B wants to know if he has been in a narrow contact with a positive and contagious other user, he can start the app and ask the server.
Step 4 - The user B receives the positive tokens from the server in order to compare them to his stored tokens. If he finds an identical token his phone, an alert is displayed and the tokens are deleted. The token's comparison is carried out directly on the B mobile phone. The server must not be able to reconcile a narrow contact. The servers could be sleeted and managed by big area like, for example, a city or region in order to minimize the volume of data to compare. This area must be populated enough to guarantee anonymity.

N.B. : When a user declares his situation as positive and contagious, his tokens cannot be simultaneously on more than one mobile phone because tokens are immediately deleted after the transfer.

If a positive contagious token reaches the expiration date before any comparison or merge 5 with another token, it must be deleted anyway.
The decentralization ensures that the data needed to extract a tractable information about an interaction cannot be on the server at the same time. And when someone was contaminated it is impossible to know by who because this anonymous information does not exist in parallel on the emitter of the information and the receptor. When the receptor gets the information, the transmitter no longer has any trace of the information.

5. Idea to investigate

To improve this last feature it could be interesting to encrypt the token before the transfer on the server. Encryption key must be stored on the mobile when the token is created. The receiver mobile phone must have the decryption key. It's could be created in parallel of the token creation on the mobile phone.
To further reduce centralization, the app could take the service from different servers managed by different organizations, cities, departments, central state's, NGO's, hospitals, companies, areas ... A signature of compliance could be required before accepting to take and provide the service between the app and the server.
The server can require the signature of compliance form the app before accepting to provide the service. Reciprocally if the app respects the rules, the server can check it with a complience signature before accepting to provide the service. This way, the solution can be stretched beyond any kind of borders.

6. Method

This approach was firstly built theoretically with keeping in mind a strong focus on the anonymity, the accessibility, the scale up and the transparency. The decentralization is the main strategy to solve the challenge.
Each technology included in the theoretical phase was tested one by one and independently. The last step consisted in building a prototype that can run together each technology regarding the basic rules described above. This prototype is open sourced and every contribution, improvement or fork is welcome.

7. Conclusion

The goal of this approach is to propose a frame to develop an accessible solution reliable and respectful against the user privacy as much as possible. This approach could be improved, especially regarding the decentralized aspect. As it is already written above, the risk of living a multiplication in the future of pandemic or epidemic situations is strong and it is important to have and improve this kind of tool in the future.

An open source codebase will be released in a few days.

References

[1] , 2005. RFC 4122 - A Universally Unique IDentifier (UUID) URN Namespace. IETF.
URL https://tools.ietf.org/html/rfc4122#section-4.4

[2] , 2006. RFC 4634 - US Secure Hash Algorithms (SHA and HMAC-SHA). IETF.
URL https://tools.ietf.org/html/rfc4634

[3] , 2019. WebRTC 1.0: Real-time Communication Between Browsers. W3C.
URL https://www.w3.org/TR/webrtc/

[4] Andrea Apolloni, Chiara Poletto, J. J. R.-P. J. . V. C., 2014. Metapopulation epidemic models with heterogeneous mixing and travel behaviour. Theoretical Biology and Medical Modelling.
URL https://tbiomed.biomedcentral.com/articles/10.1186/1742-4682-11-3

[5] Bluetooth SIG, I., 2020. Profiles make bluetooth technology interoperable.
URL https://www.bluetooth.com/specifications/profiles-overview/

[6] Group, W. B. C., 2020. Draft community group report.
URL https://webbluetoothcg.github.io/web-bluetooth/f

[7] INRIA france, F. A. G., 2020. Robert: Robust and privacy- preserving proximity tracing.
URL https://github.com/ROBERT-proximity-tracing/documents/blob/master/ROBERT-summary-EN.pdf

[8] Lauren Tindale, Michelle Coombe, J. E. S., 2020. Transmission interval estimates suggestpre-symptomatic spread of covid-19. medRxiv.
URL https://www.medrxiv.org/content/10.1101/2020.03.03.20029983v1

[9] Toph Allen, Kris A. Murray, C. Z.-T., 2017. Global hotspots and correlates of emerging zoonotic diseases. Nature.
URL https://www.nature.com/articles/s41467-017-00923-8

[10] Xavier Bonnetain, Anne Canteaut, V. C., 2020. Le traçage anonyme, dangereux oxymore.
URL https://risques-tracage.fr/docs/risques-tracage.pdf