Kartik Padave

A022

70362019039

Experiment 9

Aim

Study of Contact Tracing Application and use of cryptology

Abstract

Countries have taken different routes to facilitate contact-tracing: informational check-in apps, location monitors, and modified health-related security and publicity policies. But as countries continue facilitating testing, contact tracing, isolation, and quarantine, there have been public controversies over the methods used to facilitate contact tracing.

Introduction

COVID-19 has devastated the lives of millions of people around the globe. As of June 2021, approximately 176 million people have contracted the virus and about 3.8 million people have died from the virus. While no one of a specific stature or lifestyle is immune to the virus, demographical statistics show healthcare disparities and uneven COVID-19 case distributions across different racial communities. To combat the virus, virologists have recommended several measures one can take, including wearing a mask and social distancing. However, once someone has tested positive for COVID-19, contact tracing can help curtail the spread of the virus. Contact tracing is defined as the process of identifying persons who may have encounter an infected person, and the subsequent collection of further information about these contacts. According to the CDC, “Contact tracing is a key strategy to prevent the further spread of COVID-19”.

The DP-3T algorithm allows for digital contact tracing while protecting users’ privacy. The model of the algorithm presented in this paper is drawn. Apps that utilize this algorithm are currently available on smartphones in Austria, Belgium, Croatia, Germany, Ireland, Italy, the Netherlands, Portugal, and Switzerland.

Theory

We will be explaining the concept using an example, Let Avia and Bai be two people who have downloaded the app. Every day, their phone will generate random keys. Avia and Bai go to a cafe where they sit six feet apart from each other for fifteen minutes. Every ten minutes, their phones communicate, via Bluetooth, exchanging their respective random keys. Both phones will keep track of which keys they heard and spoke. The next day, Bai does not feel very well, and she tests positive for COVID19. She alerts the app, and her phone immediately posts her secret keys on the hospital database. Avia's phone will check this database periodically and compare it against keys it heard. It will see Bai's code and realize Avia met someone with COVID-19. As a result, it will notify Avia that they are a close contact, and they will quarantine. Note that no personal information about Avia or Bai is required throughout the entire process. Furthermore, the use of secret keys allows Bai to keep her COVID-19 diagnosis private from others since she remains completely anonymous to Avia.

First, the algorithm generates random keys or codes and takes note of them. Next, when two smartphones come in close contact with each other for an extended period, they exchange these keys using Bluetooth. During this exchange, the phones keep track of all codes that they say or send to the other smartphone and hear or receive from the other smartphone. If someone tests positive for COVID-19, then all the codes which their phone said are uploaded to a back-end server or hospital database. All smartphones regularly check this database to see if any codes it heard pop up. If a code the phone heard shows up in the database, then the app alerts the user that they are a close contact and should quarantine.

There are three steps in the process of generating random keys. The algorithm for generating these keys inputs the current day and outputs the random keys for that day.

Step 1: Calculating the Secret Key. Let t be the current day. The phone will first produce a random initial daily seed or secret key, SKt, which will be used to produce the ephemeral identifiers (EphIDs). EphIDs are the random messages the algorithm will say to communicate with other smartphones. The value SKt is calculated by the following:

$$SKt = H(SKt-1),$$

where H is a cryptographic hash function that maps SKt−1 to the value SKt in a way that is difficult to guess what SKt−1 is. The initial value, SK0 is generated using a secret key algorithm. These steps ensure that SKt is created securely to prevent the user from being traced.

Step 2: Producing String of EphIDs. Next, the algorithm uses SKt to produce the EphIDs. If the EphID changes every L minute, a total of n = (24 · 60)/L new EphIDs must be produced every day. In Example 2.1, since the devices communicated every 10 minutes, we have L = 10, and each phone would have to create n = (24 · 60)/10, or 144 EphIDs every day. To create all the necessary EphIDs, the algorithm takes SKt as an input and outputs a string with all the EphIDs. The device computes:

$$E = P\,RG(P\,RF(SKt, P)),$$

where P RF is a pseudo-random function that produces random numbers, P RG is a pseudo-random generator that turns a random seed into a longer pseudo random string, and P is a fixed public string. Note that this process is deterministic, so inputting the same value of SKt will produce the same value of E.

Step 3: Turning the String into Individual EphIDs. E, which is a string of all the EphIDs. E has length 16n, and it is split into n 16-bit sections that are the n EphIDs. The device chooses a random order to broadcast the EphIDs for L minutes each. This process allows for the creation of randomized EphIDs, which do not convey any information about the user and thus protect their privacy. Furthermore, EphIDs are changed regularly to prevent the user from being traced by them, protecting their security.

Next, smartphones communicate via Bluetooth Low Energy (Bluetooth LE)one to exchange their respective EphIDs, as shown by Figure 1. The random letters said by each phone in Figure 1 represent EphIDs. Phones send out beacons conveying its EphID to other phones, represented by the speech bubbles in Figure 1. The device receiving the beacon stores: one. The EphID, 2. A measurement of the exposure, and 3. The date. To measure the exposure risk, the phone stores the strength of the Bluetooth signal as an indicator of how far away the other person was. In Example 2.1, this measurement would include

that Avia and Bai sat six feet apart for 15 minutes. Data is stored by EphID to maximize efficiency (since there could be multiple received beacons for each EphID). An estimate of the storage space required is 6.1 MB. Furthermore, each phone stores its SKt value for the past 14 days, the same length as the incubation period for the COVID-19 virus. This ensures that a user can upload all the SKt values from the period they were infectious should they test positive for COVID-19.

Finally, if someone contracts COVID-19, the algorithm has a secure method of reporting infections to preserve privacy.

Step 1: Notify Application of a Positive Result. First, the user must report to the app that they have tested positive for COVID-19. The positive test result is self-reported but see Section 4.1 for a method of verifying the user has COVID.

Step 2: Upload Secret Key Values. Then, the SKt values for all the days when the user was contagious, which could be determined by a medical health expert or the user themselves, are sent to the back-end server.

Step 3: Delete Secret Keys from User Device. After reporting the values of SKt, the user's phone deletes those seeds and restarts the process of generating random keys. Since the next value of SKt is dependent on the previous one, this step prevents an attacker from tracking users by figuring out their current SKt value and EphIDs.

Step 4: Devices Check Back-end Server. The server then reports the values (SKt, t), which all phones with the app download. From this ordered pair, the device can derive the user with COVID-19's EphIDs during the time when they were infectious. The device can compute the EphIDs from the secret key value because the process of creating these EphIDs is deterministic. These EphIDs are then cross-checked to the local database of EphIDs the device heard to determine if the user was a close contact. Note the value t is necessary to report because the app must make sure it met an EphID before SKt was made public to avoid being tricked by an attacker broadcasting those EphIDs after they are published.

Step 5: Alert Close Contacts. Finally, the algorithm alerts the user if they were a close contact and what their exposure measurement is. With this information, the user can then quarantine to mitigate the spread of the virus.

Conclusion

While DP-3T is a highly effective solution for carrying out contact tracing while protecting location privacy, we underscore multiple weaknesses in the algorithm. These weaknesses could cause contact-tracing exchanges or data leakage, which then would result in economic, social, and political troubles at both an individual and national scale. Continual experimentation with these encryption mechanisms and solutions may improve contact tracing systems for COVID-19, as well as for future pandemics or epidemics, in terms of both privacy and security. Technology has allowed us to stay connected despite the 6-feet social distancing barriers that have become a norm during the pandemic era. Our investigations into DP-3T serve as testament to the electronic inter-connectivity and digital health systems that exist in the modern world. Furthermore, DP-3T takes such technologies and attempts to find a middle ground in the spectrum between security and privacy.

References

1. The AES-256 Cryptosystem Resists Quantum Attacks - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Distributing-key-over-quantum-channel_fig3_316284124 [accessed 13 Jun 2021]
2. Aumasson, Jean-Philippe. Serious Cryptography: a Practical Introduction to Modern Encryption. No Starch Press, 2018.
3. Beaufays, Francoise, et al. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf
4. Bennett, Charles H. and Brassard, Gilles, Quantum cryptography: Public key distribution and coin tossing, Theoretical Computer Science, Volume 560, Part 1, 2014, Pages 7-11, ISSN 0304-3975, https://doi.org/10.1016/j.tcs.2014.05.025. (https://www.sciencedirect.com/science/article/pii/S0304397514004241)
5. Choi, Yunseo and Unwin, James. Racial Impact on Infections and Deaths due to COVID-19 in New York City. 9 Jul 2020, https://arxiv.org/pdf/2007.04743.pdf.
6. Chouffani, Reda. “Fortify These 6 Security Layers to Protect Patient Data.” SearchHealthIT, TechTarget, 28 June 2019,

searchhealthit.techtarget.com/tip/ Fortify-these-6-security-layers-to-protect-patient-data.
7. Conikee, Chetan. “Case Study: CVE-2020–15957 Vulnerability Discovery in DP-3T COVID-19 Contact Tracing Backend...” Medium, Medium, 6 Aug. 2020, https://bit.ly/3gkRFFz.
8. “Contact Tracing – CDC’s Role and Approach.” CDC, CDC, 15 Jan. 2021, www.cdc.gov/coronavirus/2019-ncov/downloads/php/ contact-tracing-CDC-role-and-approach.pdf.
9. “COVID-19 Coronavirus Pandemic.” Worldometer, Worldometer, www.worldometers.info/coronavirus/.
10. Craven, Connor. “What is the Advanced Encryption Standard (AES)?” SDxCentral, SDxCentral, 13 May. 2020, https://www.sdxcentral.com/security/definitions/ what-is-advanced-encryption-standard-aes-definition/.
11. Davidson, John. “Quantum Computing 101: What’s Superposition, Entanglement and a Qubit?” Australian Financial Review, 26 Dec. 2019, https://bit.ly/3iE5q3C.