SwissCovid FAQ

Thomas Bossard, Lena Csomor, Gian-Luca Frei,
Michael Hartmann, Raphael M. Reischuk

Zühlke Engineering AG

v1.0
June 2020

Abstract

As an additional measure to fight the COVID-19 pandemic, the Swiss government has commissioned the development of a contact-tracing smartphone application called "SwissCovid" [6]. Despite being a desirable goal, it is not always possible to avoid mistrust and uncertainty among the public. To shed more light on frequently discussed concerns, this document aims to answer a number of frequently asked questions in an easily understandable, yet technically precise manner. A thorough and complete security assessment of the source code, the architecture or the protocol is not the subject of this document.

To this end, the authors had a close look at the available resources, in particular at critical points where issues were expected. The work of Vaudenay and Vuagnoux [10] served as additional input.

General limitations of app-based contact tracing (such as medical and societal) are not discussed.

1 Introduction

Citizens generally do not want to be tracked by the government, cybercriminals or nation states. People have therefore been very vocal about their concerns regarding freedom and democracy. After a variety of information channels explained how exactly the tracing itself works [11], and that the government's intent is not to track their citizens' every move, public concerns started to shift. While the intentions of the parties involved in the development of the app might be benign [12], the quality of the protocol, architecture and code that form the app are fundamental for the implementation and fulfilment of these intentions. If the app were not secure and thus vulnerable to abuse, third parties with the right skills could track people, their locations and their identities.

The authors of this FAQ document therefore conducted a broad security code review of the mobile app [3, 4] and a more in-depth security code review of the backend source code [5]. The authors looked at various security concerns, including those raised by researchers [10], and picked the ones considered to be the most relevant to the public. The goal was to make sure that different points of view are taken into account and that answers are easily understood.

2 FAQ

2.1 The crucial question: how secure is SwissCovid?

After having conducted a security review of the mobile app [3, 4] and the backend source code [5], the overall impression of the code base is positive. The implementation is state-of-the-art and appears to be the result of a professional development team. The authors could not find any significant vulnerabilities; other participants of the public security test also only found issues of minor severity [9]. Unexpected functionality was not found.

However, security is an ongoing process, especially in a distributed system with many different endpoints and services. Additional vulnerabilities will likely be found in the future, mainly because the app is still being developed and the source code is continuously changing. In this case, the responsible parties should be able to react quickly and roll out updates in a timely manner.

Furthermore, the overall security level of a complex system depends to a large extent on the security level of each participating component, be it the system infrastructure (including data transport through mobile networks), the smartphones of the users (being jailbroken or otherwise tampered with), the users themselves (with both benign-and-knowledgeable, benign-but-ignorant, and malicious behaviour), or potentially malicious third-party sensors and devices participating in the ecosystem. The core components under the control of the responsible parties can take precautionary measures to best avoid security glitches, which appears to be done sufficiently in the present case. However, security is a joint discipline and thus requires joint effort.

2.1.1 System architecture

The first part of the security review covers the overall system architecture. The SwissCovid app and its backend consist of various decentralized components. The mobile app available on Android and iOS devices serves as the main interaction point for the end users. The app regularly downloads exposure keys from the backend, which are saved in the exposure database. Old exposure keys are deleted regularly. Uploading exposure keys requires a valid security token. Issuing those tokens to the app is only possible when COVID-19 is diagnosed by a certified medical professional and requires a so-called *COVID code* that only an authority can generate. The COVID codes are stored on a separate backend database and are deleted as soon as a user requests the security token to upload their exposure keys.

2.1.2 Database

A closer look has been taken at the exposure database to understand what data is actually being stored about the users. Since the application has a strong emphasis on privacy, the focus was put on personally identifiable information (PII). It appears that only the necessary data is being stored. The time of an exposure upload is rounded to the start of the retention period, which is two hours by default. Thus, within these two hours, all uploaded exposures have the same request timestamp. The only recommendation would be to increase the retention period to 12 hours. Limiting the retention period to two hours or less could potentially lead to correlation of exposure keys, especially when infection numbers are as low as at the time of writing. In our opinion, this correlation could only be used in very unlikely attack scenarios.

2.1.3 Backend

Various classes, endpoints and code fragments that belong to the legacy version of DP3T were found during the review. Those code fragments should be removed, as they make reading the code more difficult and could lead to dead code in the future. In addition, well-implemented security concepts were found, such as fake-exposure-key uploads to obfuscate real uploads and thereby make it much harder for an attacker to determine if a person is infected only by monitoring the encrypted network traffic. The built-in time delay when uploading fake data enhances the efficacy of this feature, since the time taken to respond to fake and real requests is the same.

2.2 Is SwissCovid open-source?

The source code of the mobile apps [3, 4], the backend [5], and the documentation [1] are publicly accessible on GitHub. The main advantage of a publicly accessible source code is that everyone can search for errors and vulnerabilities in the implementation, which is currently undergoing a public security test [9].

The recent article by Serge Vaudenay and Martin Vuagnoux [10] triggered a public debate because the SwissCovid app uses a new operating system functionality (the so-called GAEN API [7]) offered by the operating system manufacturers Apple and Google. (See Section 2.3 for more details on the functionality and its necessity.) The GAEN API, like most parts of the operating systems, is closed-source and cannot be inspected by the public. The corresponding privacy concerns that many people may have, especially when large corporations like Google and Apple are involved, have to be put into perspective, however. Most smartphones today collect an incredible amount of sensitive user data. Not only the number of steps taken or motion data such as GPS data are stored somewhere on the device, but also our pictures, contacts, and passwords. The operating system of a smartphone can – at any time – record and manipulate all data that is being processed on the device without being easily noticed by the end user. This is why using even the most basic functions of today's smartphones requires the user to put a fundamental amount of trust in the manufacturers of the underlying operating systems. With the new GAEN API, the manufacturers of the operating systems do not gain any more control over the device than they already have. As Apple's and Google's operating system code was never officially published, one can state that the Federal Office of Public Health (FOPH, BAG) published as much of the tracing functionality code as possible. SwissCovid should therefore be considered open-source.

Another criticism put forward by Vaudenay and Vuagnoux is that one cannot verify that the published app has indeed been created from the published source code. This argument is valid for all kinds of implementations. While it can be assumed that the Swiss public authorities have shown themselves to be trustworthy with the approaches they have chosen [12, 11, 9], we encourage efforts to reverse-engineer the application and thus deliver a plausible indication that the application works as published. The app does not promise a zero-trust approach (and it is highly questionable whether there is a feasible approach that would work with a zero-trust architecture). Thus, a certain amount of trust in the Swiss public authorities is fundamental at this point.

2.3 Why are Apple and Google involved?

Ensuring the health and overall well-being of its citizens is probably the main responsibility of a government. The representatives must therefore choose their partners carefully, particularly under extreme circumstances such as a quickly spreading pandemic. When it comes to the contact-tracing app released in Switzerland, third parties have been involved [12] and the justification of their participation has to be investigated. This section should shed light on the support and inclusion of the smartphone manufacturers and should help the reader understand the necessity of the decision.

The contact-tracing app relies on Bluetooth connectivity in order to detect proximity to nearby devices. Bluetooth is a good choice, as it exchanges small data packets locally and does not need the users' location information [12]. To this end, Bluetooth needs to run without interruption. However, using Bluetooth over a longer period of time rapidly drains the battery if implemented in the usual way. The reason is that most smartphone operating systems require Bluetooth to be run in the foreground, a decision that was made to protect the users from malicious background activities that would otherwise go unnoticed by them. A foreground process requires user interaction in order to run and cannot start independently. Apple and Google have now decided to offer an API that allows official and certified contact-tracing apps (and only those [8]) to use Bluetooth in the background. The high battery consumption from using Bluetooth would otherwise make the app almost unusable and reduce its public acceptance.

Furthermore, the GAEN API [7] generates the keys (TEK and RPI) for the application in a controlled and unified manner. The seed for the random number generators should therefore only come from the operating system (and could thus be manipulated by Apple and Google if they wanted to; see also Section 2.2).

The seed of a random number generator is the part that should be as truly random as possible [2]. A sequence of pseudorandom numbers is then generated from the seed and used to derive cryptographic keys and identifiers. Having the API generate the keys offers compatibility with contact-tracing apps from other nations. This is crucial, as COVID-19 infections also need to be tracked across borders. It ensures all apps use the same implementation of the same protocol. Moreover, Apple and Google can offer the best support for their own devices, and make it easier to run the app on as many devices as possible and have different apps on different types of devices communicate with each other.

As modifications to the key-derivation process would usually pose a security risk if openly accessible, Apple and Google need to make sure that only trustworthy applications have access to it. Apple and Google have the resources and experience to offer the necessary access control and provide a secure implementation of the API.

Access control to the API from Apple and Google is fundamental. Without access control, third-party apps could jeopardize the user anonymity that the COVID-19 tracing apps aim to guarantee. Third-party apps could use other means of identification or location tracking such as credentials or GPS to map the keys (RPI) of a user to their identity and location, which would pose a violation of the promised anonymity.

A final point to consider is whether or not the public should demand that the code of the API be open-source. As certain functions of the smartphone operating systems are essential in order for the app to work, and the operating system itself is not open-source, it is fair to say that Apple and Google are already considered trusted parties and the trust in the new functions does not exceed the trust already placed in the operating system.

To summarize, it appears to be an appropriate choice to include Apple and Google in a globally and technically coordinated endeavor with potentially unprecedented impact on the health and well-being of our societies.

References

[1] Swiss Admin. PT-System-Documentation. `https://github.com/admin-ch/PT-System-Documents/blob/master/overview.md`, 2020.

[2] Wikipedia contributors. Entropy (computing) — Wikipedia, The Free Encyclopedia.
`https://en.wikipedia.org/wiki/Entropy_(computing)`, 2020. accessed 2020/06/18.

[3] DP3T. dp3t-app-android-ch.
`https://github.com/DP-3T/dp3t-app-android-ch`, 2020.

[4] DP3T. dp3t-app-ios-ch. `https://github.com/DP-3T/dp3t-app-ios-ch`, 2020.

[5] DP3T. dp3t-sdk-backend.
`https://github.com/DP-3T/dp3t-sdk-backend`, 2020.

[6] FOPH. Faktenblatt: Die Swiss PT-App hilft, das Coronavirus in Schach zu halten. `https://www.bag.admin.ch/bag/en/home/krankheiten/ausbrueche-epidemien-pandemien/aktuelle-ausbrueche-epidemien/novel-cov/situation-schweiz-und-international.html#-2097806982`, 2020.

[7] Apple Inc. Privacy-Preserving Contact Tracing.
`https://www.apple.com/covid19/contacttracing`, 2020.

[8] Gerrit de Vynck Mark Gurman. Apple, Google Covid-19 Tool to Be Limited to One App Per Country.
`https://www.bloomberg.com/news/articles/2020-05-04/apple-google-covid-19-tool-to-be-limited-to-one-app-per-country`, 2020.

[9] MELANI. SwissCovid Current reports. `https://www.melani.admin.ch/melani/en/home/public-security-test/current_findings.html`, 2020.

[10] Martin Vuagnoux Serge Vaudenay. Analysis of SwissCovid.
`https://chaosticino.ch/docs/20200605--vaudenay\%2Bvuagnoux--analysis-of-swisscovid.pdf`, 2020.

[11] Jenni Thier. Auch der Nationalrat gibt grünes Licht – was Sie zur Contact-Tracing-App wissen müssen.
`https://www.nzz.ch/technologie/was-sie-zur-contact-tracing-app-wissen-muessen-ld.1555664`, 2020.

[12] Jenni Thier. Kann eine App die Corona-Pandemie stoppen? Mit dieser Frage beginnt eine für die Schweizer Wissenschaft aussergewöhnliche Erfolgsgeschichte. `https://www.nzz.ch/technologie/corona-contact-tracing-die-schweiz-inspiriert-apple-und-google-ld.1559375`, 2020.