Corona Warn-App – Design, Development and Privacy Considerations

Oliver Layer, Benedikt Jaeger*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: oliver.layer@tum.de, jaeger@net.in.tum.de

***Abstract*—Contact tracing applications can help to reduce the spread of the coronavirus disease 2019 by identifying infection chains. The *Corona-Warn-App* is the official German application. While contact tracing apps require a certain amount of users in the population to be effective, there are privacy, effectiveness and security concerns that may diminish the app's acceptance. In this paper functionality and possible privacy and security attack vectors as well as mitigations for the app are reviewed. Furthermore, the app's architecture is compared with other approaches. The results show that privacy and security measures are in place, limiting possible attacks to be infeasible on a large scale. In contrary, there have been several bugs during the introduction phase of the app which could have put off users.**

***Index Terms*—contact tracing apps, privacy, exposure notification, corona warn app**

1. Introduction

In the beginning of 2020 the world has been struck by a pandemic regarding the *coronavirus disease 2019* (COVID-19). The disease is believed to spread especially in situations where people are in proximity to each other. Using their smartphones and their Bluetooth signals, *contact tracing apps* (CTAs) provide information to users indicating if there was a situation in the past where they were exposed to someone who has already been infected. CTAs can help to identify the chain of infections and can therefore slow down the pandemic by breaking them. In contrast, only relying on manual contact tracing is not suitable on a larger scale and for most situations, such as in public transport.

Throughout the pandemic, several CTAs have been developed using different architectures. The *Corona-Warn-App* (CWA) is the open-source contact tracing app of the German government. It is based on the *Exposure Notification API* (ENA) which has been developed jointly by Google and Apple. The approach builds upon a decentralized architecture with the goal of preserving privacy.

Initially, the German government pursued to follow a centralized approach, which may be more prone to privacy breaches than a decentralized one. After being criticized, the German government instead chose to use the ENA. [1]

Broad usage across the population is important for CTAs to have an impact on the development of the pandemic. The CWA has approximately been downloaded 23 million times as of December 2020. [2] The amount of users having the app currently installed may be less. A study shows that approximately one third of the surveyed did not want to install the CWA for several different reasons. [3, Sec. Results] Therefore, the motivation of this paper is to review the architecture of the CWA regarding privacy, security and some other technical considerations which could prevent users from installing the app, such as bugs in the app or the general transparency of the app. Furthermore, this paper compares the ENA approach with different approaches.

In Section 2, it is explained how the CWA and other related ENA-based CTAs work. Privacy, security and other technical considerations are dealt with in Section 3. The ENA is compared with other approaches in Section 4. Afterwards, related work is summarized in Section 5. A conclusion is given in Section 6.

2. Functionality

There are three parts of the CWA that are essential for contract tracing. The proximity detection is responsible for keeping track of nearby users, while sharing the infection information uploads data of the infected user to the central CWA server. The infection risk calculation consists of getting a list of keys which represent infected users from the CWA server and comparing them with the local data captured by the proximity detection.

2.1. Proximity detection

The proximity detection is part of the ENA and uses *Bluetooth Low Energy* (BLE). Smartphones running ENA-based CTAs broadcast and scan for BLE messages with the ENA service identifier around every 3.5 to 5 minutes. The exact interval is determined by a randomized component to prevent tracking. [4, p. 4] [5, `scanIntervalRandomRangeSeconds()` comment]

The payload of a BLE broadcast consists of the *Rolling Proximity Identifier* (RPI) and the *Associated Encrypted Metadata* (AEM). The RPI serves as a temporary identifier for the sending device and is newly derived every 15 minutes. This happens at the same time the randomly generated Bluetooth MAC address changes. The RPI contains a bucketized version of the Unix timestamp with a bucket size of ten minutes and is AES-128 encrypted using the RPI key. The RPI key itself is derived from the *Temporary Exposure Key* (TEK) using the HKDF function described in RFC5869. [7] The TEK in turn is an identifier that is freshly generated every day using a cryptographic random number generator. [4, p. 3, 4] [6, p. 6]

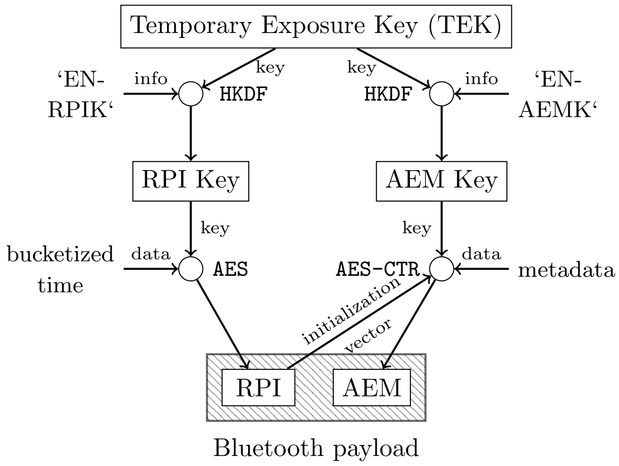

Figure 1: Derivation process of the payload sent in BLE broadcasts. [6, p. 5]

The AEM contains metadata, such as the versioning information about the ENA and the transmit power which was used to send the BLE broadcast. [4, p. 4] This information is later used when determining the infection risk. Similar to the mechanism regarding the RPI, the AEM is encrypted by the AEM key that is derived from the TEK using HKDF. Analogous to the RPI, it also changes every 15 minutes to prevent tracking. As initialization vector for the AES-128-CTR encryption, the current RPI is used. [6, p. 7] The derivation process is visualized in Figure 1.

When scanning and finally receiving a BLE broadcast of another user in proximity, the RPI and the encrypted AEM are stored locally on the smartphone. Decryption of the AEM is only possible with the TEK of the user that initially sent the broadcast. [6, p. 7]

2.2. Sharing infection information

Sharing a positive test result using the CWA allows for other users to check their chance of being infected later in the process of the infection risk calculation.

There are multiple ways on how to mark oneself as infected. Some laboratories print QR codes on the letter that the user receives after conducting the test. The CWA supports scanning this code and will notify users as soon as there is a test result. [8]

Not all laboratories may support this. In this case, the German health authorities may share a code with the infected user that can be used to share their infection status. This code is distributed when the authorities call the users to inform them about the measures they have to take regarding their infection. [8] The user can also take action and phone the CWA call center to receive a code.

As soon as an user is tested positive for COVID-19, there is the possibility of sharing the test result with the CWA server. What is being transmitted to the server in this case are all the TEKs of the last 14 days. The list of TEKs is referred to as diagnosis key. [6, p. 8]

2.3. Infection risk calculation

When installed, the CWA automatically pulls recent diagnosis keys from the CWA server. In previous versions this happened on a daily basis. Recent versions (since v1.7) allow multiple downloads per day, which also means that the infection risk can be invalidated multiple times per day. [9]

Using the downloaded diagnosis keys and the contained TEKs, the CWA can recalculate the RPI keys as well as the AEM keys. Matching locally stored RPIs and AEMs can then be decrypted. For each match, the total encounter time on that particular day is calculated. Additionally, the distance between the smartphones is determined using the signal strength.

If an encounter belonging to the match lasted less than 10 minutes or the distance was larger than 8 meters on average, it is automatically classified as low risk. [10]

For each of the remaining encounters, the total risk score is calculated by multiplying four scaled metrics, ranging from 0 to 8 [11, Sec. Risk Score Calculation], namely:

- days since the exposure has happened
- exposure duration
- signal attenuation
- transmission risk level

The transmission risk level is calculated using an epidemiological model and contained in the uploaded diagnosis key. For example, it can possibly take symptoms entered by the user into account. [12] This particular model is not part of the ENA, but it uses the customizable transmission risk level offered by the ENA.

With taking all exposures into account, a combined risk score is calculated. First, the attenuation levels are grouped into three buckets using predefined thresholds. Each bucket's weight is then multiplied with the sum of the corresponding exposure durations for which the attenuation falls into one of the buckets. The result is called the exposure score. It is multiplied with the normalization of the largest total risk score to finally get the combined risk score. [11, Sec. Risk Score Calculation]

If the combined risk score exceeds a certain threshold, the user is shown a high risk exposure warning.

3. Considerations

CTAs are reliant on a broad acceptance of the population to be effective. The acceptance increases if the app does not interfere with the privacy of the users. For example, this could mean that data that could reveal the users identity is not shared with others.

Moreover security issues, for example attacks that generate fake risks which are shown to the users, can cause uncertainty.

Furthermore different technical considerations could affect the acceptance, such as bugs in the app preventing it from working correctly.

3.1. Privacy

In general, the CWA has been designed with the goal of ensuring as much privacy as possible. Consequently, there have been multiple measures to guarantee privacy, such as frequently changing identifiers, the usage of cryptographic methods for identifier generation / derivation and using a decentralized concept. In contrast to a centralized approach, no information leaves the smartphone, except when sharing a positive test result. Still, the most frequent concerns to not use the CWA are privacy concerns. [3]

Nevertheless, there have been successful attempts demonstrated in literature to circumvent these privacy measures. The resulting privacy threats are mainly deanonymization and movement tracking of users. If exploited, these threats could lead to loss of acceptance in the population.

Movement tracking of users sharing their positive test result is practically possible for all apps using the ENA, such as the CWA. As described in Section 2.2, sharing the result requires the users to upload their TEKs of the last 14 days. The TEKs can then be queried from the CWA backend by anyone. Using BLE sniffers deployed at central locations, such as train stations or supermarkets, one can trace the movement of infected users for at least one day by deriving RPIs from a particular TEK and comparing them with the RPIs picked up by the sniffers. Tracing the movement for longer than one day is also possible, if one manages to match multiple uploaded TEKs using the users movement behavior. Using the movement information, one can possibly also deanonymize users. [13, Sec. III]

Limitations of this approach are certainly not being able to trace users who did not share their test result. In addition it requires the deployment of BLE sniffers. For tracing people in a city with a population of around 160 000 people, approximately 430 strategically placed sensing stations would be necessary. [13, Sec. III] This amount of sniffers needed makes this approach infeasible on a Germany-wide scale. Especially the government, as publisher of this CTA, has access to more suitable methods for tracking users, such as using the data from the mobile networks.

On a smaller scale, deanonymization is also possible using another attack with a BLE device capturing signals at multiple locations. One can then store the RPIs and the signal strength at each location. Observing the locations when capturing the signals establishes a connection between the captured signals and the observed person. Another similar attack is to approach a person and track the RPIs sent by the persons smartphone. When there are not many other signals around, one can likely identify the RPIs belonging to the approached person. If the person is now tested positive for COVID-19 and shares the infection status, one has gained the information that the person is infected. This can be done by deriving the RPIs from the uploaded TEKs and comparing them with the previously picked up RPIs. [14, Sec. 4.2]

These attacks require personal proximity to the victims or camera surveillance and are therefore only possible on a small scale. There are mitigation proposals for both attacks, such as varying signal strength when sending BLE broadcasts. [14, Sec. 4.2]

As seen in the limitations, all presented attacks require a significantly large effort to be able to track users on a large scale. Nevertheless, they are feasible when tracking users on a smaller scale.

3.2. Security

Besides privacy issues, security issues can lead to attacks that stop the app to work in the desired way. For example this could be generating fake risks that lead to warnings for users.

Literature has shown that wormhole attacks (also *relay and replay attacks*) are possible for ENA-based apps. Such an attack picks up a BLE signal at some location, preferably a crowded one. Then the attacker uses a second device at a different location. The second device receives BLE messages that the first device picked up. This is done with the help of a tunnel built by the attacker between the two devices. Now the second device broadcasts these messages and all devices will receive BLE broadcasts originated from the first location, while actually being at the second location. [13, Sec. IV]

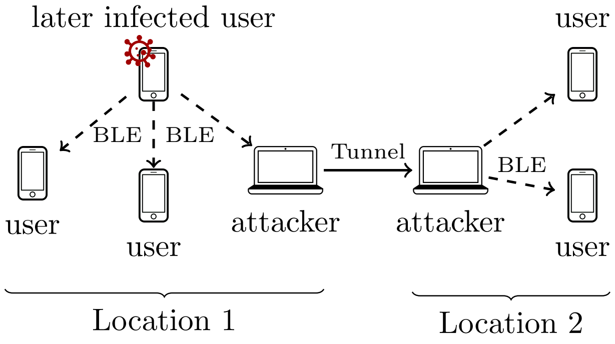

Figure 2: Example setup of a wormhole attack. [13, Figure 7]

Figure 2 shows an example setup of this attack. All users present at location 2 will receive broadcasts from the later infected user at location 1, although this user is not necessarily in proximity in reality.

This attack can be used to generate fake risk contacts, which may tempt users to conduct a test or go into quarantine without a real risk being present. Possible mitigations require either a handshake mechanism or additional verification using the GPS location or the cellular network. [14, Sec. 3.2] A limitation of this attack is that physical presence of some kind (e.g. a smartphone or a microcontroller) is required at the locations where the attack should be performed.

Another possible attack is called power and storage drain attack. It is a denial of service attack, in which the attacker broadcasts a large amount of BLE messages. Devices in the proximity will pick up and process these messages. A large amount of messages to process will result in a higher power consumption. If the attacker manages to generate valid messages, they will also use space on the smartphones storage, as the RPIs and the AEMs are persisted. [14, Sec. 3.1] While this attack may be less severe than the relay and replay attacks, the users' acceptance will decrease if such an attack occurs at her smartphone. A mitigation for this attack is also proposed in literature. [14]

Both presented attacks are hard to apply at a large scale, because they require physical devices at the attacked locations. Nevertheless, anyone exploiting these attacks will certainly lead to the app not working as intended.

3.3. Technical

Since the CWA launched in June 2020, several bugs were discovered. Some of these bugs were preventing users from utilizing the app.

One problem that occurred in early versions was that the CWA was not refreshing the infection risk value on some Android devices without the user manually opening the app. Broadcasting and receiving RPIs would work, but in case an user had an encounter with a later positive diagnosed person, the user would not get a notification without opening the app. A fix deployed later added a setting which, when enabled, lets the CWA run in the background even on Android systems which stop apps running in the background for battery saving reasons. [15]

Another bug appeared for iOS users with the update to version 1.2.0 released in early August 2020. Some users were not able to start the app any more after the update. This was quickly fixed in a follow-up update released five days later. [16]

In September 2020, an additional bug was found which affected smartphones running iOS 13.7. The bug caused the computation of the risk value to be faulty, and would ultimately result in displaying a too high risk for some users. [17]

An additional reason for not using the CWA may be the power consumption of the app. Using the app could lead to a decreased runtime of the smartphone. There is no relevant literature that investigates battery consumption of the ENA or CWA. Nevertheless, BLE was chosen as it is explicitly designed for usage in environments with battery constraints, such as smart home applications for example.

Another technical aspect is transparency. Transparency certainly leads to higher confidence of users that the app contains what is being promised. The CWA is completely open-source and reproducible builds are currently worked on, which then gives certainty that the code in the GitHub repository belongs to the deployed binaries in the app stores. [18]

4. Comparison

Knowing the functionality and issues of the CWA makes it interesting to compare it to other approaches. An overview of selected other approaches is shown in Table 1.

Architecture	Concept	Country
Decentralized	ENA	Germany, Denmark [19], Brazil [20], Italy [21], Spain [22], United Kingdom [23], United States *(partly)* [24], Canada *(partly)* [25]
Partially-centralized	BlueTrace	Singapore [26], Australia [27]
	ROBERT	France [28]
	other	Iceland, India

TABLE 1: Architectures, theoretical concepts and corresponding deployment location of selected CTAs

The underlying concepts of most CTAs can be grouped into two categories regarding their architecture. There are *decentralized* and *partially-centralized* architectures. *Partially-centralized* architectures generally require more interaction of the users with a central server. Examples of an interaction may be an initial registration with personal contact information or an upload of encounters to the server, depending on the implementation. In contrast, for *decentralized* architectures the only transmission of user data to the server may possibly happen when sharing a positive test result, which is not mandatory for using the app.

The most prominent concept using a *decentralized* architecture is the already discussed ENA, which is used by many western countries as seen in Table 1. On the other hand. there are different concepts using a *partially-centralized* architecture, such as *BlueTrace* or *ROBERT*.

In the following subsections selected concepts using a *partially-centralized* architecture are compared with the ENA.

4.1. BlueTrace

BlueTrace has been developed by the government of Singapore. [29, Sec. Abstract] As seen in Table 1, it is currently used in Australia and Singapore.

To use the app, users have to register using their phone number. An account is then created on the backend side, containing the phone number and a randomly generated user identifier. [29, Sec. 4]

Similar to the ENA, the proximity detection uses BLE broadcasts with frequently changing temporary identifiers. In contrast to the ENA, the temporary identifiers are not generated by the user but by the central server. After receiving them from the server, they are broadcasted by the user's smartphone. A temporary identifier includes the user identifier and time information and is encrypted on the server using symmetrical encryption, which enables only the health authority to decrypt it. Analogously to the ENA, received broadcasts are stored on the local smartphone storage. [29, Sec. 4]

If users are tested positive, they upload their locally saved encounters to the central server. The health authority can then decrypt the temporary identifiers and contact the encounters using their phone number saved in the server's database. [29, Sec. 4]

In contrary to the ENA, BlueTrace is only affected by wormhole attacks (see Section 3) to a limited extent. Firstly, this is the case because the broadcasts contain an expiry timestamp, which the server verifies upon uploading the encounter history. Therefore, the broadcast of a user can only be rebroadcasted for a maximum of 15 minutes. Secondly, human operators also verify the locations of both, the infected user and potentially infected users, via phone. [29, Sec. 8] This does not completely rule out wormhole attacks, but may limit their effectiveness.

Bluetooth sniffer attacks by third parties as in Section 3 are not applicable to BlueTrace, assuming a third party cannot decrypt the broadcasts of users and is therefore not able to track them beyond the 15 minutes refresh interval of identifiers. However, such an attack concerning all users could be performed by the health authority, as they are able to decrypt all temporary identifiers.

BlueTrace may also be more effective when it comes to risk classification, as employees of the health authority can decide to contact encounters based on some additional context given by the infected person during the phone call.

Nevertheless, the main weakness of BlueTrace is that if somebody manages to obtain the secret key of the health authority, every temporary identifier could be decrypted. With additional access to the database of the server, every temporary identifier could be connected with the users' phone numbers. This is not possible using the ENA, because the data that is sent in the broadcasts is encrypted with keys generated by the users themselves. Additionally, no personal data of users is stored in context of the ENA.

4.2. ROBERT

The French CTA uses a concept called *ROBust and privacy-presERving proximity Tracing* (ROBERT), which is in turn built upon the *Pan-European Privacy-Preserving Proximity Tracing* (PEPP-PT). The German government initially pursued to implement a CTA based on PEPP-PT. [1]

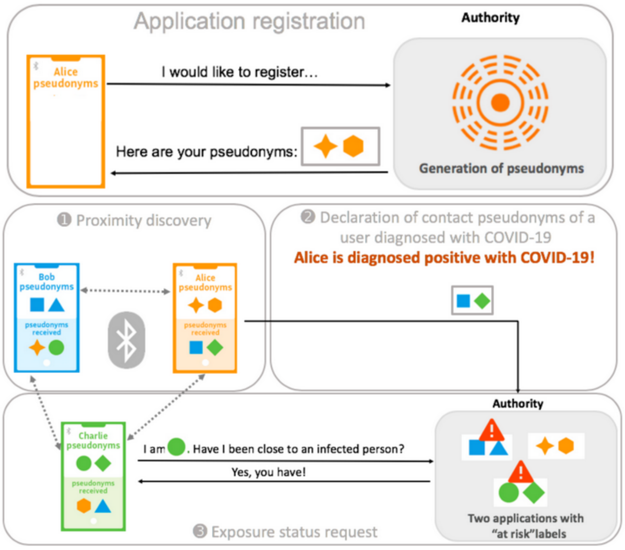
Application registration
Authority
Alice pseudonyms
I would like to register...
Here are your pseudonyms:
Generation of pseudonyms
❶ Proximity discovery
Bob pseudonyms
pseudonyms received
Alice pseudonyms
pseudonyms received
❷ Declaration of contact pseudonyms of a user diagnosed with COVID-19
Alice is diagnosed positive with COVID-19!
Authority
Charlie pseudonyms
pseudonyms received
I am ●. Have I been close to an infected person?
Yes, you have!
Two applications with "at risk"labels
❸ Exposure status request

Figure 3: Tracing flow of ROBERT. [30, p. 3]

Similar to BlueTrace, ROBERT relies on temporary identifiers that are broadcasted via BLE and generated by a central server. [31, Sec. 4, 5.1] The user regularly gets these identifiers from the server and uses them for broadcasts. Received broadcasts are saved to the local smartphone storage, analogously to the ENA and BlueTrace. [31, Sec. 5.2]

If a user is tested positive and wants to share the infection status, the encounter history is uploaded to the central server. The server then calculates the encounter times and adds them to the database entry belonging to the encountered user. [31, Sec. 6]

Every user regularly sends a request to the server with recently used identifiers. The server checks if there has been an encounter and returns the result to the client. [31, Sec. 7]

The tracing flow of ROBERT is depicted in Figure 3.

The main difference to BlueTrace is that the user does not have to send personal information, e.g. the phone number, to the server.

Compared to the ENA, firstly ROBERT uses identifiers generated at the server and not at the local smartphone and secondly relocates the logic of risk calculation to the server. While wormhole attacks may still be viable, sniffer attacks as described in Section 3 become impossible for third parties, as there is no publicly accessible list of infected identifiers. On the other hand, ROBERT makes it possible to perform sniffing attacks for all users when having access to the key used by the server.

4.3. Other approaches

There are other approaches which do not use BLE for proximity detection. The Icelandic CTA uses the locations services (e.g. GPS) of the smartphone's operating system. Only when sharing the infection status or upon request of the authority, the location history can be uploaded to a server. [32] The authority can then take measures, for example by warning people regularly being present at one of the locations.

The CTA of India uses yet another approach. It combines both, Bluetooth and GPS data for proximity detection. Additionally, it requires users to register themselves by providing personal information, such as their name, their age or their phone number. If an infection happens, the Bluetooth encounter history is uploaded to the server together with the location information. [33]

5. Related Work

Most of the literature focuses on one particular aspect of the CWA. The analysis in [14] gives a theoretical, detailed overview of security and privacy issues in the ENA, while [13] contains two case studies demonstrating security and privacy issues of the CWA. Similar to this paper, both papers give a short overview about the functionality.

While being technical, the influence of issues on the apps acceptance is not discussed. Most literature discussing reasons why not to use the CWA are not technical, but rather only conduct representative surveys of the population, like it is the case for [3].

There is no literature that explicitly looks at the population's concerns about the app and compares them with the technical background. This paper tries to fill this gap.

6. Conclusion

There are privacy issues in the CWA that could lead to deanonymization and tracking of users in the worst-case. In addition, wormhole attacks can decrease the usefulness of the app by generating fake risk warnings. Also the user experience was cumbersome especially in the beginning, as some users were not able to correctly use the app due to bugs. Nevertheless, the most critical bugs were fixed in the meantime.

While there exist these privacy and security issues and real world attacks may be possible for single cases, they are not feasible for a large scale. Even on a small scale, a large amount of effort is required. In fact, the CWA provides a decentralized architecture which ensures that no sensitive data leaves the smartphone. Information like

doi: 10.2313/NET-2021-05-1_11

the location or identity are in no means transmitted to the server, instead, a design of locally generated and frequently changing identifiers is used. For other approaches, such as the partially-centralized ones, this mostly is not the case. In their case, this could possibly lead to more drastic worst-case privacy breaches than it is the case for the decentralized approach.

Especially in regards to previous software projects developed by the government, the CWA seems to be an good example in terms of privacy and transparency.

To conclude, the privacy and security measures of the CWA are good enough for attacks only to have a limited impact on a large scale. Users with privacy concerns may not know about the effort of the measures taken to ensure this level of privacy and security.

References

[1] K. Becker and C. Feld, "Corona-Tracing: Bundesregierung denkt bei App um," *Tagesschau*, April 2020, https://www.tagesschau.de/inland/coronavirus-app-107.html, [Online; accessed 10-December-2020].

[2] Robert Koch-Institut, "Kennzahlen zur Corona Warn App," December 2020, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/WarnApp/Archiv_Kennzahlen/Kennzahlen_04122020.pdf, [Online; accessed 10-December-2020].

[3] K. T. Horstmann, S. Buecker, J. Krasko, S. Kritzler, and S. Terwiel, "Short report: Who does or does not use the "Corona-Warn-App" and why?" *European Journal of Public Health*, 12 2020, ckaa239. [Online]. Available: https://doi.org/10.1093/eurpub/ckaa239

[4] Apple/Google, "Exposure Notification Bluetooth® Specification," April 2020, https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-BluetoothSpecificationv1.2.pdf, [Online; accessed 12-December-2020].

[5] Google, "Exposure Notifications Internals - ContactTracingFeature.java," August 2020, https://github.com/google/exposure-notifications-internals/blob/main/exposurenotification/src/main/java/com/google/samples/exposurenotification/features/ContactTracingFeature.java#L367, [Online; accessed 07-January-2021].

[6] Apple/Google, "Exposure Notification Cryptography Specification," April 2020, https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-CryptographySpecificationv1.2.pdf, [Online; accessed 12-December-2020].

[7] H. Krawczyk and P. Eronen, "Hmac-based extract-and-expand key derivation function (hkdf)," Internet Requests for Comments, RFC Editor, RFC 5869, May 2010, http://www.rfc-editor.org/rfc/rfc5869.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5869.txt

[8] Corona-Warn-App Team, "Software Design Verification Server," July 2020, https://github.com/corona-warn-app/cwa-verification-server/blob/master/docs/architecture-overview.md, [Online; accessed 14-December-2020].

[9] ——, "Android App Releases," December 2020, https://github.com/corona-warn-app/cwa-app-android/releases, [Online; accessed 24-December-2020].

[10] ——, "Risk Assessment," December 2020, https://github.com/corona-warn-app/cwa-documentation/blob/master/cwa-risk-assessment.md, [Online; accessed 24-December-2020].

[11] ——, "Solution Architecture," December 2020, https://github.com/corona-warn-app/cwa-documentation/blob/master/solution_architecture.md, [Online; accessed 24-December-2020].

[12] ——, "Epidemiological Motivation of the Transmission Risk Level," October 2020, https://raw.githubusercontent.com/corona-warn-app/cwa-documentation/master/transmission_risk.pdf, [Online; accessed 24-December-2020].

[13] L. Baumgärtner, A. Dmitrienko, B. Freisleben, A. Gruler, J. Höchst, J. Kühlberg, M. Mezini, R. Mitev, M. Miettinen, A. Muhamedagic, T. D. Nguyen, A. Penning, D. F. Pustelnik, F. Roos, A.-R. Sadeghi, M. Schwarz, and C. Uhl, "Mind the gap: Security & privacy risks of contact tracing apps," 2020, https://arxiv.org/abs/2006.05914.

[14] Y. Gvili, "Security analysis of the covid-19 contact tracing specifications by apple inc. and google inc." Cryptology ePrint Archive, Report 2020/428, 2020, https://eprint.iacr.org/2020/428.

[15] A. Wilkens, "Corona-Warn-App: SAP erläutert Problem mit der Hintergrundaktualisierung," heise online, July 2020, https://www.heise.de/news/Corona-Warn-App-SAP-erlaeutert-Problem-mit-der-Hintergrundaktualisierung-4851648.html, [Online; accessed 29-December-2020].

[16] L. Becker, "Corona-Warn-App öffnet sich nicht mehr auf iPhones: Update soll helfen," heise online, August 2020, https://www.heise.de/news/Corona-Warn-App-oeffnet-sich-nicht-mehr-auf-iPhones-Update-soll-helfen-4869676.html, [Online; accessed 29-December-2020].

[17] J. Hoerdt, "Problems with iOS 13.7," September 2020, https://www.coronawarn.app/en/blog/2020-09-10-ios-13-bug, [Online; accessed 29-December-2020].

[18] Corona-Warn-App Team, "F-Droid release and reproducible builds," December 2020, https://github.com/corona-warn-app/cwa-app-android/issues/1483, [Online; accessed 29-December-2020].

[19] Smittestop, "About the app," https://smittestop.dk/about-the-app/, [Online; accessed 11-February-2021].

[20] Governo do Brasil, "Coronavírus-SUS: aplicativo alerta contatos próximos de pacientes com Covid-19," August 2020, https://www.gov.br/casacivil/pt-br/assuntos/noticias/2020/agosto/coronavirus-sus-aplicativo-alerta-contatos-proximos-de-pacientes-com-covid-19, [Online; accessed 11-February-2021].

[21] Presidenza del Consiglio dei Ministri, "Immuni - Domande Frequenti," https://www.immuni.italia.it/faq.html, [Online; accessed 15-February-2021].

[22] RadarCOVID Team, "RadarCOVID iOS App Repository," https://github.com/RadarCOVID/radar-covid-ios, [Online; accessed 15-February-2021].

[23] NHS, "I got an "Exposure Check Complete" notification from the app. What does this mean?" https://faq.covid19.nhs.uk/article/KA-01319, [Online; accessed 15-February-2021].

[24] The Stanford Daily, "CA Notify app offers COVID-19 exposure alerts for Stanford community," December 2020, https://www.stanforddaily.com/2020/12/28/ca-notify-app-offers-covid-19-exposure-alerts-for-stanford-community/, [Online; accessed 15-February-2021].

[25] Canadian Digital Service, "COVID Alert Mobile App Repository," https://github.com/cds-snc/covid-alert-app, [Online; accessed 15-February-2021].

[26] Bluetrace, "BlueTrace Protocol," https://bluetrace.io/, [Online; accessed 15-February-2021].

[27] Australian Government, "Technology behind COVIDSafe," https://www.covidsafe.gov.au/technology.html, [Online; accessed 15-February-2021].

[28] INRIA, "TousAntiCovid Repository," https://gitlab.inria.fr/stopcovid19/accueil, [Online; accessed 15-February-2021].

[29] J. Bay, J. Kek, A. Tan, C. Sheng Hau, L. Yongquan, J. Tan, and T. Anh Quy, "BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders," 2020, https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf, [Online; accessed 27-February-2021].

[30] Inria and Fraunhofer AISEC, "ROBERT: ROBust and privacy-presERvingproximity Tracing Summary," April 2020, https://raw.githubusercontent.com/ROBERT-proximity-tracing/documents/master/ROBERT-summary-EN.pdf, [Online; accessed 28-February-2021].

[31] ——, "ROBERT: ROBust and privacy-presERvingproximity Tracing," May 2020, https://raw.githubusercontent.com/ROBERT-proximity-tracing/documents/master/ROBERT-specification-EN-v1_1.pdf, [accessed 28-February-2021].

[32] The Directorate of Health of Iceland and The Department of Civil Protection and Emergency Management of Iceland, "Rakning C-19," 2020, https://www.covid.is/app/en, [Online; accessed 28-February-2021].

[33] FTI Consulting Asia Pacific, "A Review of India's Contact-tracing App, Aarogya Setu ," September 2020, https://www.lexology.com/library/detail.aspx?g=f54419a1-4823-404c-92f3-c5e4f193b733, [Online; accessed 28-February-2021].