

# Privacy Guarantees of Bluetooth Low Energy Contact Tracing: A Case Study on COVIDWISE

**Salman Ahmed**, IBM Research

**Ya Xiao, and Taejoong (Tijay) Chung**, Virginia Tech

**Carol Fung**, Virginia Commonwealth University

**Moti Yung**, Google LLC and Columbia University

**Danfeng (Daphne) Yao**, Virginia Tech

*We examine the security, privacy, and reliability of Google and Apple's COVID-19 exposure notification technology, using actual case studies and realistic use cases. Our analysis validates the system, providing piece of mind for adopters of contact tracing and potentially boosting transparency.*

**C**COVID-19 has become the deadliest viral outbreak around the globe since the Spanish influenza pandemic in 1918. At the beginning, with the absence of vaccines, containment and mitigation were the best strategies, and they continue to be when new waves of variants and mutations of the virus appear. Contact tracing can greatly help early containment by linking people who have been exposed to others who are infected and tracking

and notifying them. Advances in computer technology aid the process by following individuals' mobile devices and proximity through GPS<sup>1</sup> and Bluetooth Low Energy (BLE) beacons.<sup>2,3</sup>

To combat COVID-19 and aid governments and health organizations with contact tracing, Google and Apple jointly introduced a BLE technology, Google/Apple Exposure Notification (GAEN), in April 2020.<sup>4</sup> GAEN uses interoperable BLE signals to broadcast Bluetooth beacons from one device to another when Android/iOS users come in close proximity. The beacons help track the distance between users and the duration of users' contact. When a person

is diagnosed as having COVID-19 at the time of contact or within a valid time frame afterward (and only then), the system can notify other users about potential exposure (the infected user's smartphone uploads the generators of its signal, which other users' devices pull from a server).

Researchers have scrutinized contact tracing technology and warned that its adoption could have privacy and security issues,<sup>5-11</sup> thus perhaps advocating against its wide deployment. However, these works primarily engineered attacks based on abstract protocol designs and (theoretically formulated) adversaries, which at times represent an extreme, economically unjustified, and expensive enemy rather than a typical one. Most did not verify the technology through actual investigations (accessing software and experimenting with devices), and none tried to discover scenarios in which the system was robust against typical attacks, which are less expensive and intensive than the breaches they describe.

While scrutiny is always important, none of the earlier works assess the feasibility of attacks under real circumstances (when the system is deployed) in terms of operations and costs versus what an attacker gains beyond a minimal disturbance to the system. Granted, there are cases of heavily invested and massive deployments of devices/readers that can attack the system, and some attacks are extreme due to the ability of the adversary (gaining full access to devices). In this sense, the researchers' attacks were, indeed, good to know about as extreme but unlikely events.

This work, like others<sup>12-15</sup> that evaluate trust, security, privacy, usefulness, traceability, transparency, and

reliability, means to fill the gap and investigate contact tracing in a balanced way by inspecting the actual system (its software and operation) and assessing its strengths as well as its weaknesses (essentially, assuming that the system encounters a typical attack, not one by

most adopted (10.5%) contact tracing app in the United States.<sup>17</sup> Our examination of GAEN is conducted with it.

In the rest of this article, we explain and analyze GAEN's privacy design and experimentally evaluate several BLE-related properties. We confirm

**THERE ARE CASES OF HEAVILY INVESTED AND MASSIVE DEPLOYMENTS OF DEVICES/READERS THAT CAN ATTACK THE SYSTEM, AND SOME ATTACKS ARE EXTREME.**

an adversary that spends extensively to launch a dedicated, targeted attack. The investigation can be useful for understanding the system's resiliency during this and future pandemic outbreaks. Specifically, we perform an analysis of GAEN with two focus points: 1) ensuring that the library code (from Google and Apple) and contact tracing code (from various government and health organizations) protect user privacy and 2) investigating privacy flaws in the design and implementation of GAEN, if any.

COVIDWISE,<sup>16</sup> Virginia's official contact tracing app, uses the GAEN system. Other major GAEN-based contact tracing apps around the globe include COVID Alert (Canada), Corona-Warn-App (Germany), COVID Tracker (Ireland), Swiss-Covid (Switzerland), Immuni (Italy), NHS COVID-19 (United Kingdom), and several U.S. apps, including GuideSafe (Alabama), Covid Watch (Arizona), COVID Alert NY (New York), Care19 Alert (Wyoming), Safer Illinois (University of Illinois), and PocketCare S (State University of New York at Buffalo). As of mid-March 2021, COVIDWISE was the

that GAEN prevents tracking through random Bluetooth addresses, thus providing strong privacy guarantees. We find that iPhones deliver strong privacy protection via the nonresolvable random private address and prevent malicious apps from snooping on users' rolling proximity identifiers (RPIs). We also confirm that RPIs' refresh interval is within the range of 10–20 min<sup>18</sup> and may vary with the distance between devices. We break down assumptions about, and assess the feasibility of, advanced attacks targeting contact tracing apps.

## DESIGN OVERVIEW OF GAEN

GAEN broadcasts and stores BLE beacons without any interaction with the app if the system is turned on. However, a user can turn off the system through the app and the exposure notification settings. GAEN provides application programming interfaces (APIs) to support different operations. Figure 1 illustrates interactions between a user, the exposure notification system, and the app. GAEN uses

BLE due to the technology’s availability on a smartphone needed for applications such as smart homes, proximity tracking, wearables, health care, and fitness that require little data transfer and low latency. BLE’s practical communication range is 10–20 m (33–66 ft), which is sufficient for GAEN. The app also uses BLE to optimize power consumption, requiring significantly less energy than traditional BLE communication for peripherals.<sup>21</sup> Besides, GAEN’s passive power usage (that is, broadcasting only when devices are close to one another) further reduces consumption.

### Temporary exposure keys, Bluetooth beacon, and RPIs

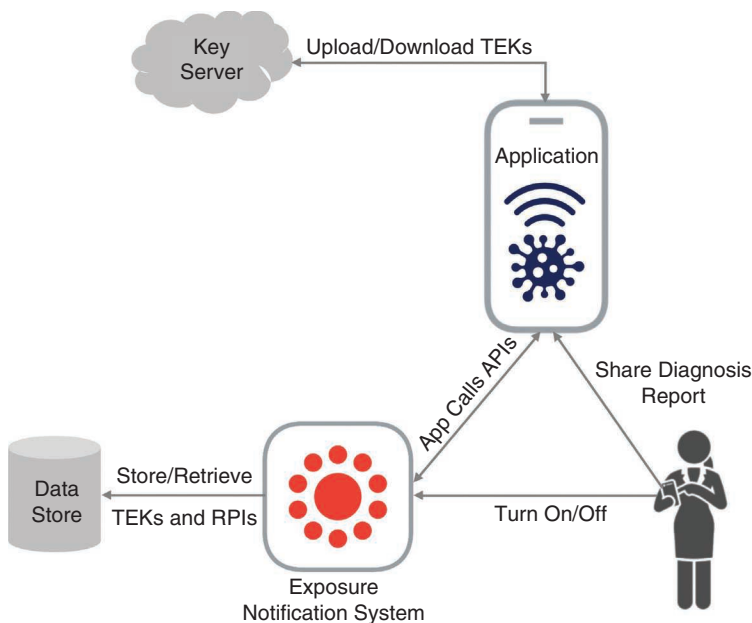
The heart of GAEN is the temporary exposure key (TEK). A TEK is a random number created using a cryptographic

ally secure pseudorandom generator. It is a 16-byte number used to identify a device for a day within its lifetime. The GAEN system generates a new TEK every 24 h to make it hard for attackers to track infected users beyond a one-day period. In addition, the Bluetooth beacon’s payload includes an identifier: the RPI, which is derived from a TEK as an AES encryption key (and the current time indication within the 24-h period as the message). A Bluetooth beacon’s payload also carries metadata such as the protocol version and transmission power, which are encrypted using a key derived from the TEK. The RPI and metadata are expected to change every 10–20 min (see “Exposure Notification Bluetooth Specification”<sup>18</sup>) to prevent attackers from tracking the devices of uninfected users by exploiting Bluetooth beacons that are overly persistent.

When a user is infected, his or her device uploads the TEKs for the relevant period (14 days) to a server. Other users’ devices pull the TEKs of infected people, produce RPIs, and match the results against their stored RPIs to detect exposure. Because TEKs are daily keys, it is impossible to link RPIs between days (when one downloads TEKs from the server, there is no indication which TEKs on other days are coming from the device of a given day’s TEK). In terms of basic privacy, the goal of the system is to relate to TEKs and RPIs, which are random objects, and not to users and devices. This design philosophy was originally shared by the GAEN system and a number of academic groups in an attempt to minimize the loss of privacy while enabling contact tracing (and allowing reasonable storage and computations at devices).

### API and app responsibilities

Contact tracing apps and the underlying GAEN system have different responsibilities. GAEN is responsible for transmitting, receiving, and storing Bluetooth signals. Apps enable people to share positive diagnoses and automatically notify the central server and, eventually via the system, others who were in contact with them. Health authorities (for example, the Virginia Department of Health) set exposure detection thresholds (that is, the minimum distance between users and duration of contact) in the apps. GAEN provides 17 APIs to facilitate interactions with contact tracing apps (for example, COVID-WISE). The key responsibilities of APIs are TEK creation and management, RPI generation and management, BLE broadcasting and scanning, and exposure detection. The apps are responsible for user authorization, downloading TEKs, presenting exposure



**FIGURE 1.** The user and contact tracing app interactions with the exposure notification system. TEK: temporary exposure key. RPI: rolling proximity identifier. API: application programming interface.

notifications, and uploading TEKs. For example, the Virginia Department of Health enables users to share positive COVID-19 test results, and for security, it assigns a six-digit PIN to each patient, who may enter the number in COVIDWISE, as shown in Figure 2. This disclosure is voluntary in Virginia. People who have been in close proximity to a COVID-19-infected person for at least  $T$  minutes in the past 14 days are notified. The Department of Health determines and sets the value of  $T$ .

### Overview of GAEN's privacy design

Out of the 17 GAEN APIs, two—*getTemporaryExposureKeyHistory()* and *provideDiagnosisKeys()* in Android—deal with potentially sensitive information. The first fetches the TEKs from the past 14 days from on-device data storage and provides them to the app for uploading to the key server. Apps use the second API to insert one or more batches of TEKs into on-device storage. These APIs are sensitive because they exchange critical information (that is, TEKs) with the exposure notification system. To ensure the privacy and integrity of the TEKs, the APIs use a specific file format (for example, *export.bin*) and a verification method through signatures (for instance, *export.sig*). On the other hand, contact tracing apps are responsible for securely communicating with the key server and uploading and downloading TEKs. The apps and key server verify the integrity of TEKs through digital signatures. The apps do not use any personally identifiable information (PII), device identifier, or Bluetooth identifier in the process of sharing COVID-19-positive information.

### Threat models and claimed privacy guarantees

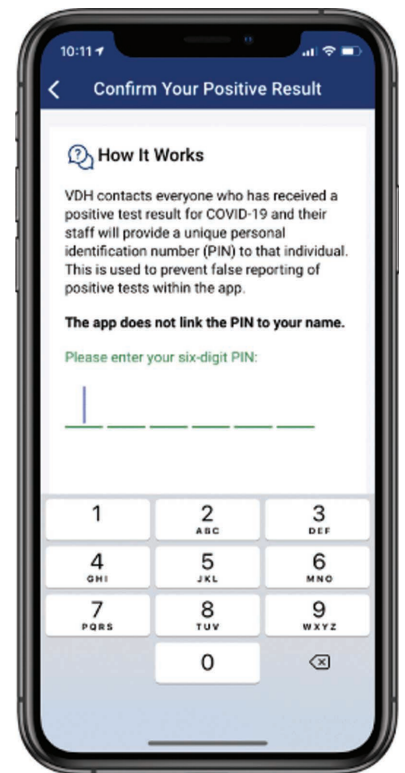
We consider four threat levels to discuss GAEN's privacy guarantees: the 1) the walking trail, 2) your neighbor, 3) stalker, and 4) organized crime models. We define and categorize threat severities based on attackers' privilege levels for accessing RPI beacons in different real-world scenarios. These privilege levels are compatible with the assumptions made in the literature.<sup>5-11</sup> In the walking trail and your neighbor models, an adversary can sniff a limited number of beacons to obtain RPIs. In the stalker model, an adversary can sniff a small number of BLE beacons (for example, using fewer than 10 BLE sniffing devices) to obtain RPIs. In the organized crime model, we assume that an adversary can compromise a smartphone, set up a large-scale infrastructure to sniff BLE beacons, and hack health-care systems to obtain PINs to share information about positive diagnoses. We detail these threat levels with attack scenarios in Table 1.

The GAEN and contact tracing app privacy guarantees include five key aspects: 1) preventing attackers, public health authorities, governments, and Apple/Google from tracking and monitoring users' movements; 2) generating TEKs without using PII or any context (like the geographic location); 3) sharing COVID-19-positive diagnoses without revealing user information; 4) preventing attackers from obtaining PII even if they gain access to TEKs; and 5) enabling users to turn off GAEN at their discretion. Furthermore, based on the principle of least privilege, TEKs never leave users' devices unless there is a positive test.

### BLE AND RPI EXPERIMENTS

We conducted simple experiments to investigate various BLE aspects in

GAEN and COVIDWISE to confirm the privacy guarantees. Using PacketLogger (an extension to the Xcode Apple developer tool) in iOS and Bluetooth system logs in Android, we intercepted and collected Bluetooth beacons to examine whether all the intervals work as expected and whether there are any identifiers (for example, resolvable addresses) in the Bluetooth beacons. We also inspected device storage for keys and identifiers in Android (using a Pixel 4a) and iOS (using an iPhone 7) by measuring the number of Bluetooth beacons sent in 24 h.



**FIGURE 2.** COVIDWISE enables a COVID-19-positive patient to share his or her test result by using a six-digit PIN. (Source: COVIDWISE.)

## Randomness of Bluetooth addresses

We examined the randomness of Bluetooth addresses used in transmitting beacons to observe if a receiving entity can resolve the sender's address. We observed that Android and iOS utilize random addresses to conceal the identity of a sender while transmitting advertisement packets, as expected. Android and iOS apply different types of random addresses. Android phones use

resolvable random private addresses, while iPhones employ nonresolvable random private addresses. The difference is that Android devices enable trusted parties (for example, paired devices) to resolve the random private addresses. However, both operating systems preserve privacy, assuming that paired Bluetooth devices (for instance, a user's AirPods) are trustworthy. It is important to note that contact tracing apps do not require location permission

in the latest version of Android (that is, Android 11). Older versions of Android apps require location settings to be turned on for Bluetooth communication.

## RPI interception

We examined the runtime RPI (the Bluetooth beacon) and metadata by using PacketLogger for iOS devices and Bluetooth Host Controller Interface snoop logs for Android devices. We observed that each device received

**TABLE 1.** GAEN privacy leaks and their severity versus realistic and complex threat models and their assumptions.

ID	Threat level	Attack difficulty	Attack requirement	Attack goal	Information leaked	Severity of leak	Reference
1	Walking trail	Low	Access to one RPI (common scenario)	Any information about a user	None	None	—
2	Your neighbor	Low	Access to zero to five RPIs from three to five victims, considering that neighbors come nearby zero to five times a day (common scenario)	Any information about a user	None	None	—
3	Stalker 1	Low	Access to at least RPIs from five to 10 victims in a 10–20-min window	Estimate the number of GAEN users around an attacker	Approximate number of nearby GAEN users	None	Grünblatt <sup>19</sup>
4	Stalker 2	Medium	<ol style="list-style-type: none"> <li>1. Access RPIs from at least one victim; tracking a victim for 1 h requires all RPIs in that hour</li> <li>2. Maintain continuity of RPI reception from a victim</li> </ol>	Continuously track a user	None (not trackable, based on our observation)	None	Vuagnoux <sup>8</sup>
5	Organized crime 1	High	<ol style="list-style-type: none"> <li>1. Access unlimited RPIs with location data from 10+ victims</li> <li>2. Access published TEKs through jailbroken or rooted phones or imitating a contact tracing app</li> <li>3. Aggregate data for each 10–20-min time window: dates, times, interaction graphs, social graphs, addresses, location types (residential, workplace, library, and so on), surveillance cameras</li> </ol>	Profile movements of infected users and deanonymize them	Imprecise deanonymization (precision decreases with increasing number of profiles)	Medium	Troncoso et al., <sup>2</sup> Baumgärtner et al., <sup>6</sup> and Seiskari <sup>7</sup>
6	Organized crime 2	High	<ol style="list-style-type: none"> <li>1. Access a victim's smartphone through hacking</li> <li>2. Bypass storage protection</li> </ol>	Obtain a victim's infection status	Information about whether a victim is infected	Medium	Chan et al. <sup>20</sup>

a set of advertising payloads around every 4 min in Android and 3.5 min in iOS. Figure 3 presents a raw BLE advertisement packet captured from an Android device (a Pixel 4a). The final 20 bytes are composed of a 16-byte RPI and 4 bytes of metadata. We found that the advertising packet RPI and metadata are zeros for iOS devices. This is because iOS blocks access to avoid malicious third-party apps from snooping on users' RPIs. This mechanism renders attack proposals based on stealing RPIs (for example, as described in Grünblatt<sup>19</sup>) useless in reference to iOS.

20 min). Besides, the distance approximation between may not be precise due to device position (for example, inside pockets) and the presence of glass partitions and personal protective equipment.

interrupted due to movements and obstacles, cutting the total storage to less than 8 MB. We were unable to locate stored TEK values in the logs of the Android and iOS devices, as expected.

### Key and RPI storage

We analyzed RPI storage and found that it occupied around 0.59–0.63 MB per day on Android and iOS devices. Since GAEN stores keys and RPIs for 14 days, a device may devote roughly 8.25–8.8 MB of storage if we consider uninterrupted interactions between phones. In practice, interactions are

### GAEN'S PRIVACY WITH RESPECT TO THREAT MODELS

As with all security solutions, the privacy guarantees of GAEN are relative. There certainly exist extreme scenarios (for example, those in Troncoso et al.,<sup>2</sup> Baumgärtner et al.,<sup>6</sup> Gvili,<sup>7</sup> Vuagnoux,<sup>8</sup> Grünblatt,<sup>19</sup> and Chan<sup>20</sup>) where attackers may learn

### RPI intervals

We intercepted Bluetooth beacons to examine the RPI transmission intervals. In our experiment, we used one Android device (a Pixel 4a) and two iPhones (iPhone 7s). We positioned each device according to the distance in Figure 4. Based on RPIs received on the Android smartphone, we observed that the transmission interval varied with the distance between the devices. However, the intervals satisfied those in the specifications (that is, between 10 and

Time	Bluetooth Address	RPI + Metadata
02:18:07	B7:CD:6B:64:5D:33	<u>3AD310DCA4F810EF2B0A17968BE47CB6</u> <u>EC59B6B6</u>
02:22:06	B7:CD:6B:64:5D:33	<u>3AD310DCA4F810EF2B0A17968BE47CB6</u> <u>EC59B6B6</u>
02:27:00	C6:EB:E9:DA:B2:09	<u>6D5C54D1376E95B7872CFFFC93425903</u> <u>102A1673</u>
02:31:13	D6:2C:59:37:FE:24	<u>0376829E0EBD180E82E5756E52CE7CBD7C465A03</u>
02:35:30	D6:2C:59:37:FE:24	<u>0376829E0EBD180E82E5756E52CE7CBD7C465A03</u>
02:39:32	F0:9A:11:EC:62:11	<u>81E43856A116E224DB876D9D763CAA52</u> <u>62DDAC02</u>
02:43:16	F0:9A:11:EC:62:11	<u>81E43856A116E224DB876D9D763CAA52</u> <u>62DDAC02</u>

FIGURE 3. Twenty bytes of advertised random numbers with an RPI and metadata captured from a Pixel 4a.

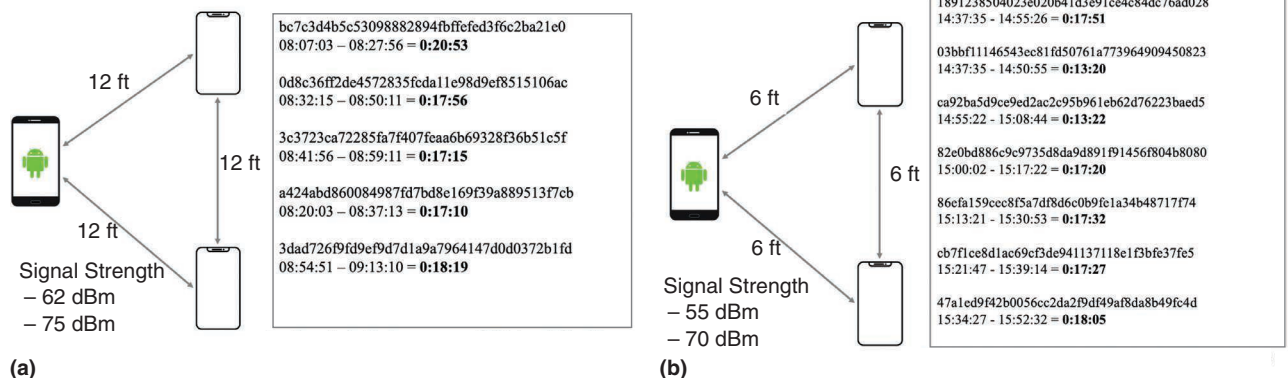


FIGURE 4. An RPI transmission interval experiment. The interval varies with the device distance. (a) A log from an Android phone, with an RPI interval of 18 min, 19 s. (b) A log from an Android phone, with an RPI interval of 16 min, 21 s. dBm: decibels per milliwatt.

additional information. If an adversary has access to RPIs, TEKs, and RPI date-time information for thousands of users, it can profile people's movements.<sup>2,6,7</sup> Table 1 summarizes the attack difficulty and leak severity in GAEN under multiple (increasing) threat categories, including the walking trail causal encounter, your

adversaries in the former receive RPIs normally, while adversaries in the latter deliberately orient themselves (for instance, by changing locations) to intercept RPIs from more victims (for example, five to 10). If successful, the stalker 1 model reveals only the approximate number of nearby GAEN users, which poses no privacy threat.

during three days. We obtained advertising packets from an Android (version 11) device (a Pixel 4a), where the advertising packets were received from two iPhones (an iPhone 7 and iPhone 11 with iOS 15.0.2) placed 6–12 ft apart and performing regular activities. Each Bluetooth address is paired with a unique RPI and vice versa. Using a Python program, we checked for the existence of nonunique pairs by searching for the use of a Bluetooth address with multiple RPIs or an RPI with multiple Bluetooth addresses. We observed no asynchronous changes of the Bluetooth addresses and RPIs. Hence, user privacy is preserved in the stalker 2 model (ID 4). Table 2 provides a few unique Bluetooth addresses and RPI pairs.

Some of the attack scenarios in Table 1 have rather strong assumptions regarding the complexity of the setup and demand huge resources. For example, attackers in the organized crime 1 model (ID 5) require TEKs and aggregated data in each 10–20-min time window to deanonymize infected users.<sup>2,6,7</sup> Aggregated data include public and sensitive information, such as dates, times, interaction graphs, social graphs, addresses, location types (for example, residential buildings, workplaces, and libraries), and surveillance cameras. This requirement for additional side-channel sources of information reduces the feasibility of the attack.

In addition, the organized crime 1 model needs access to published TEKs through a jailbroken/rooted device or by imitating a contact tracing app.<sup>6</sup> While obtaining TEKs through a jailbroken/rooted device might be feasible, imitating a contact tracing app is rather difficult. To mimic one, an attacker must fool or bypass the authorization system—specifically,

**THE APPS DO NOT USE ANY PERSONALLY IDENTIFIABLE INFORMATION, DEVICE IDENTIFIER, OR BLUETOOTH IDENTIFIER IN THE PROCESS OF SHARING COVID-19-POSITIVE INFORMATION.**

neighbor, stalker, and organized crime models. The first three models capture the most typical threat scenarios, in which GAEN leaks no sensitive information.

An adversary in the your neighbor model (ID 2) may occasionally receive beacons from a few (for example, three to five) nearby users. The difference between the your neighbor (ID 2) and stalker 1 (ID 3) models is that

A reported attack<sup>8</sup> relied on an asynchronous change of Bluetooth addresses and RPIs, which is represented in the stalker 2 model (ID 4) in Table 1. However, this attack no longer works, as GAEN requires the Bluetooth address and RPI to change synchronously, which we experimentally confirmed by extracting around 11,000 random Bluetooth addresses and RPI pairs from the advertising packets

**TABLE 2.** The synchronous change of Bluetooth addresses and RPIs in advertising packets.

Bluetooth address	RPI
13:ac:57:35:3c:ea	59c62b86cdace1fe40446bc80689ccbd323588b8
33:5d:64:6b:cd:b7	3ad310dca4f810ef2b0a17968be47cb6ec59b6b6
09:b2:da:e9:eb:c6	6d5c54d1376e95b7872cfff93425903102a1673
24:fe:37:59:2c:d6	0376829e0ebd180e82e5756e52ce7cbd7c465a03
04:2c:4d:b1:93:40	b5f1091b23a3871129a1225a6c3cebf175de28fa

an authorized administrative console, which is designed by GAEN to prevent malicious apps from downloading TEKs. Moreover, a malicious entity cannot fool the contact tracing app into accepting forged TEK export files. To maintain the back-end key server, an authorized contact tracing entity (for example, the Virginia Department of Health) must create a key to sign TEK export files and share the corresponding public version with Google/Apple, ensuring information authenticity.

Google and Apple also restrict app developers' access to GAEN APIs through an approval process. Google added an extra layer of restriction by blocking access to the Android Software Development Kit for regular app developers. These constraints prevent the misuse and abuse of GAEN APIs. The attack represented by the organized crime 2 model in Table 1 (ID 6) is difficult to launch in practice, as it requires hackers to gain access to victims' smartphones.<sup>20</sup> While issues including power and storage drains do not hamper the effectiveness of GAEN, vulnerabilities such as relay-and-replay and trolling attacks have an effect by increasing false positives. These false positives do not have an impact on privacy. Besides, our reported results do not assess the effectiveness of GAEN but focus on privacy issues.

**O**ur findings confirmed that GAEN preserves privacy in a comprehensive collection of typical threat scenarios, including the walking trail causal encounter, your neighbor, organized crime, and stalker models. Compromising user privacy by exploiting GAEN requires a complex attack setup, for example,

compromising a victim's smartphone, mounting many Bluetooth radio devices, correlating additional victim information, and accessing health-care

systems. Besides, the built-in authorization, permission, and policy enforcement mechanisms in GAEN add an extra layer of difficulty to the attacks

## ABOUT THE AUTHORS

**SALMAN AHMED** is a research scientist at IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 10598, USA. His research interests include system security assurance, attack surface quantification, and program analysis. Ahmed received a Ph.D. from Virginia Tech. Contact him at [ahmedms@vt.edu](mailto:ahmedms@vt.edu).

**YA XIAO** is a Ph.D. candidate in the Department of Computer Science, Virginia Tech, Blacksburg, Virginia, 24061, USA, working with professor Danfeng (Daphne) Yao. Her research interests include neural-network-based software security solutions. Xiao received an M.S. from Beijing University of Posts and Telecommunications. Contact her at [yax99@vt.edu](mailto:yax99@vt.edu).


**TAEJOONG (TIJAY) CHUNG** is an assistant professor in the Department of Computer Science, Virginia Tech, Blacksburg, Virginia, 24061, USA. His research interests include Internet security, privacy implications, and Internet measurement. Chung received a Ph.D. in computer science and engineering from Seoul National University. Contact him at [tijay@vt.edu](mailto:tijay@vt.edu).

**CAROL FUNG** is an associate professor at Virginia Commonwealth University, Richmond, Virginia, 23284, USA. Her research interests include network security, mobile and Internet of Things systems, and softwareized and programmable networks. Fung received a Ph.D. in computer science from the University of Waterloo. Contact her at [cfung@vcu.edu](mailto:cfung@vcu.edu).

**MOTI YUNG** is a security and privacy research scientist with Google, Mountain View, California, 94043, USA, and an adjunct research faculty with the Department of Computer Science, Columbia University, New York City, New York, 10027, USA. His research interests include security, privacy, and cryptography. Yung received a Ph.D. from Columbia University in 1988. Contact him at [moti-yung@gmail.com](mailto:moti-yung@gmail.com).

**DANFENG (DAPHNE) YAO** is a professor of computer science at Virginia Tech, Blacksburg, Virginia, 24061, USA. Her research interests include building deployable and proactive cyberdefenses, focusing on detection accuracy and scalability. Yao received a Ph.D. from Brown University. She is an Elizabeth and James E. Turner Jr. '56 Faculty Fellow and a Center for Applied Clinical Investigation Faculty Fellow. Contact her at [danfeng@vt.edu](mailto:danfeng@vt.edu).



proposed in the literature. Our article aims to help people understand and appreciate GAEN's privacy protection and encourage them to adopt GAEN-based contact tracing, which will be extremely powerful, as it will help us effectively manage this and future pandemics and minimize unnecessary casualties due to enhanced automatic contact tracing and its advantages, especially given the initial estimates of effectiveness.<sup>22</sup> 

**ACKNOWLEDGMENT**

This work was supported by the Virginia Commonwealth Cyber Initiative. The opinions and statements in this work are personal and do not necessarily represent those of Google.

**REFERENCES**

1. L. Reichert, S. Brack, and B. Scheuermann, "Privacy-preserving contact tracing of COVID-19 patients," *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 375, 2020.
2. C. Troncoso, M. Payer, and J.-P. Hubaux, "Decentralized privacy-preserving proximity tracing," 2020, arXiv:2005.12273.
3. H. Stevens and M. B. Haines, "Trac-eTogether: Pandemic response, democracy, and technology," *East Asian Sci., Technol. Soc., Int. J.*, vol. 14, no. 3, pp. 523–532, 2020, doi: 10.1215/18752160-8698301.
4. "Exposure notifications: Help slow the spread of COVID-19, with one step on your phone," Google and Apple Inc. [Online]. Available: <https://www.google.com/covid19/exposure-notifications/> (accessed Apr. 14, 2021).
5. Y. Gvili, "Security analysis of the COVID-19 contact tracing specifications by Apple Inc. and Google Inc.," *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 428, 2020.
6. L. Baumgärtner *et al.*, "Mind the GAP: Security & privacy risks of contact tracing apps," in *Proc. 2020 IEEE 19th Int. Conf. Trust, Security Privacy Comput. Commun. (TrustCom)*, pp. 458–467, doi: 10.1109/TrustCom50675.2020.00069.
7. O. Seiskari, "Paparazzi attack PoC," GitHub. [Online]. Available: <https://github.com/oseiskar/corona-sniffer> (accessed Apr. 14, 2021).
8. M. Vuagnoux, "Little thumb attack on SwissCovid," Vimeo. [Online]. Available: <https://vimeo.com/453948863> (accessed Apr. 14, 2021).
9. V. Iovino, S. Vaudenay, and M. Vuagnoux, "On the effectiveness of time travel to inject COVID-19 alerts," *Cryptography ePrint Archive*, 2020/1393, May 2021. [Online]. Available: <https://eprint.iacr.org/2020/1393>
10. A. K. R. Gennaro, A. Krellenstein, and J. Krellenstein, "Exposure notification system may allow for large-scale voter suppression," 2020, arXiv:2005.12273.
11. A. Boutet *et al.*, "Contact tracing by giant data collectors: Opening pandora's box of threats to privacy, sovereignty and national security," Ph.D. dissertation, 2020.
12. Y. J. Park and D. D. Shin, "Contextualizing privacy on health-related use of information technology," *Comput. Human Behav.*, vol. 105, p. 106,204, Apr. 2020, doi: 10.1016/j.chb.2019.106204.
13. Y. J. Park, J. E. Chung, and D. H. Shin, "The structuration of digital ecosystem, privacy, and big data intelligence," *Amer. Behav. Sci.*, vol. 62, no. 10, pp. 1319–1337, 2018, doi: 10.1177/0002764218787863.
14. D.-H. Shin, "The effects of trust, security and privacy in social networking: A security-based approach to understand the pattern of adoption," *Interact. Comput.*, vol. 22, no. 5, pp. 428–438, 2010, doi: 10.1016/j.intcom.2010.05.001.
15. D.-H. Shin, "Ubiquitous computing acceptance model: End user concern about security, privacy and risk," *Int. J. Mobile Commun.*, vol. 8, no. 2, pp. 169–186, 2010, doi: 10.1504/IJMC.2010.031446.
16. "COVIDWISE, official contact tracing app in Virginia," Virginia Department of Health. [Online]. Available: <https://www.vdh.virginia.gov/covidwise/> (accessed Apr. 14, 2021).
17. A. De La Garza, "Contact tracing apps were big tech's best idea for fighting COVID-19. Why haven't they helped?" *Time*. [Online]. Available: <https://time.com/5905772/covid-19-contact-tracing-apps/> (accessed Apr. 14, 2021).
18. "Exposure notification bluetooth specification," Google and Apple Inc. [Online]. Available: [https://blog.google/documents/70/Exposure\\_Notification\\_-\\_Bluetooth\\_Specification\\_v1.2.2.pdf](https://blog.google/documents/70/Exposure_Notification_-_Bluetooth_Specification_v1.2.2.pdf) (accessed Apr. 14, 2021).
19. R. Grünblatt, "Stop Covid detector 3000," GitHub. [Online]. Available: [https://github.com/rgrunbla/Stop\\_Covid\\_Detector\\_3000](https://github.com/rgrunbla/Stop_Covid_Detector_3000) (accessed Apr. 14, 2021).
20. J. Chan *et al.*, "PACT: Privacy sensitive protocols and mechanisms for mobile contact tracing," 2020, arXiv:2004.03544.
21. Ellisys. "Intro to Bluetooth power consumption." Bluetooth. [Online]. Available: <https://www.bluetooth.com/bluetooth-resources/intro-to-bluetooth-power-consumption/> (accessed: Nov. 30, 2021).
22. L. Ferretti, "Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing," *Science*, vol. 368, May 8, 2020. [Online]. Available: <https://www.science.org/doi/10.1126/science.abb6936>